GLOBAL MULTIDISCIPLINARY PERSPECTIVES JOURNAL

End-to-End Observability for Customer AI: Tracing Data, Features, and Predictions Across Systems

Achuta Krishna Kishore Varma Alluri
Salesforce CRM Lead, Informa Support Services Inc, Illinois, USA

* Corresponding Author: **Dr. Priyanka Appasaheb Jadhav**

Article Info

ISSN (online): 3107-3972
Volume: 01
Issue: 05
September-October 2024
Received: 18-09-2024
Accepted: 20-10-2024
Page No: 67-70

Abstract
Customer-facing AI systems increasingly span heterogeneous components such as event ingestion, streaming and batch feature computation, feature stores, model training, online inference, experimentation, and downstream decision services. Failures across these layers can silently degrade customer experience through feature staleness, training-serving skew, or distribution drift. Traditional microservice observability often stops at service telemetry and does not preserve the semantic lineage required to explain why a prediction occurred. This paper proposes an end-to-end observability approach for Customer AI that unifies distributed tracing, data lineage, feature provenance, and prediction monitoring into a single correlation fabric. We present a reference architecture that couples runtime telemetry with lineage events, binds features and predictions to immutable identifiers, and enables cross-system diagnosis workflows such as identifying which upstream data changes impacted a cohort of predictions. The resulting blueprint supports privacy-aware attribution, governance expectations, and operational actionability for production AI.

DOI: https://doi.org/10.54660/GMPJ.2024.1.5.67-70

Keywords: AI observability, data lineage, feature provenance, distributed tracing, MLOps, concept drift, feature store, governance, monitoring, customer AI

1. Introduction

Customer AI is embedded in high-frequency product experiences such as personalization, fraud screening, ranking, recommendations, and proactive customer support, where even minor degradations can trigger measurable customer and business impact. [2]

Organizations have adopted distributed tracing to debug latency and failure propagation across microservices, but most traces do not capture ML-specific semantics such as feature provenance or model versioning. [1]

Open standards for telemetry collection and context propagation make it feasible to correlate operational signals across services and platforms at enterprise scale. [3]

This paper targets the practical gap between microservice observability and AI lifecycle traceability by defining what must be observed, how it is correlated, and which workflows it should enable.

2. Background and Related Work

End-to-end observability for deployed ML pipelines has been framed as a unified loop of detection, diagnosis, and reaction to silent ML failures across pipeline stages. [6]

Standardized trace context propagation over HTTP enables consistent request identification across heterogeneous services, which is a prerequisite for correlating inference calls with upstream feature retrieval and downstream decisions. [7]

Lineage specifications are increasingly used to represent datasets, runs, and their relationships as first-class events, enabling platform-agnostic reasoning over data flow beyond logs. [9]

Data quality directly impacts ML reliability, motivating structured approaches that define data quality dimensions and practices specific to ML workloads. [19]

Monitoring research highlights that production ML introduces verification and validation challenges that differ from classical software because model behavior depends on shifting data and delayed labels. [17]

MLOps tool-support studies emphasize fragmentation across experimentation, deployment, and monitoring, reinforcing the need for a common observability and metadata backbone. [12]

Preprocessing and join logic can introduce distribution bugs and technical bias that are difficult to detect without transformation-aware inspection, motivating lineage-driven debugging techniques. [13]

3. Requirements for Customer AI Observability

A Customer AI observability system must enable cross-system correlation from customer requests to predictions, feature values, feature computation runs, and the exact data snapshots used during training. [16]

It must also support AI-specific risk management by preserving evidence for accountability, traceability, and audit readiness throughout the lifecycle. [5]

Operational overhead must be controlled through sampling, attribute discipline, and scalable storage, because customer-facing inference often runs at high throughput and strict latency budgets. [8]

Finally, observability must incorporate cybersecurity and information security requirements, since telemetry and lineage can expose sensitive operational and customer signals if not governed properly. [4]

4. Unified Observability Graph Architecture

We propose a two-plane architecture that produces a single Observability Graph by merging runtime telemetry with lineage and metadata into a correlated entity graph.

The runtime plane instruments inference services, feature retrieval, and decision services using spans and events, while preserving request context in a portable manner. [3]

The lineage plane emits dataset and run relationships, feature set definitions, and model registry artifacts so that pipeline semantics are queryable independent of the underlying compute platform. [9]

Feature stores provide a natural boundary for attaching feature identifiers and transformation fingerprints, while also reducing duplicated feature logic across teams. [15]

To prevent missing lineage during rapid feature iteration, feature definitions should be managed in a shared platform that supports reuse, offline training joins, and online inference access patterns. [14]

5. Instrumentation and Data Model

Every online inference should produce a stable prediction identifier and minimal metadata record linking the request trace to the serving model version and feature vector signature, enabling later cohort analysis without storing raw feature values.

Training and feature pipeline runs must log metadata such as dataset version identifiers, feature set identifiers, and transformation hashes to support reproducibility and rollback analysis. [18]

ML metadata stores help capture lineage across artifacts, executions, and contexts, allowing operators to connect deployed model versions back to the exact training run and data snapshot. [21]

Because many organizations operate in multiple clouds or hybrid stacks, metadata capture should align with widely adopted libraries that support lineage retrieval and querying across systems. [18]

6. Monitoring Signals for Data, Features, and Predictions

Data and feature monitoring should combine schema checks, freshness checks, distributional tests, and semantic constraints that are evaluated continuously in both batch and streaming pipelines. [28]

Continuous data profiling can reduce the time-to-detection for subtle pipeline regressions by embedding profiling into the transformation loop rather than treating it as an occasional manual step. [22]

Drift detection is essential because labels are frequently delayed, so monitoring must rely on unsupervised or weakly supervised signals that detect distributional changes and representation shifts. [24]

For operational robustness, telemetry should include tail latency, error rates, cache hit rates, and feature store SLA indicators to distinguish data-quality faults from infrastructure faults.

7. Incident Response Workflows Enabled by End-to-End Observability

When customer impact is detected, operators should be able to pivot from a prediction cohort to the exact feature distributions that shifted and then traverse provenance to the upstream pipeline run and dataset version that produced those features. [6]

If preprocessing introduced cohort-specific bias or unexpected distribution changes, transformation-aware inspection can reveal the exact operator (e.g., join, filter, imputation) that created the shift. [13]

Automation ROI studies reinforce that observability should be evaluated as a time-to-detection and time-to-recovery accelerator, not only as a monitoring dashboard, because operational savings compound at scale. [19]

Decision intelligence frameworks for agile governance can be integrated with observability to route incidents into standardized remediation playbooks, automated testing gates, and controlled promotions. [26]

8. Security, Privacy, and Governance Controls

Telemetry and lineage data must be governed with role-based access control, attribute allowlists, and retention policies that minimize the risk of exposing sensitive customer or business information.

Security guidance for smart infrastructure emphasizes that systems supporting critical services must implement defense-in-depth and continuous monitoring to ensure resilience under attack and failure conditions. [11]

Online safety perspectives further motivate privacy-aware observability designs that prevent telemetry from becoming a mechanism for unintended inference about users or sensitive cohorts. [29]

ISO AI management system standards provide an organizational framework for implementing AI governance, including lifecycle controls, documentation, and continual improvement processes that observability can operationalize. [23]

9. Implementation Considerations and Performance

Sampling strategies should balance forensic value and cost by increasing sampling during incidents, new deployments, or drift alerts while keeping baseline sampling low for steady-state traffic.

Kernel-level observability approaches can complement application traces by capturing low-overhead network and system signals that help diagnose performance regressions in feature retrieval and inference paths. [25]

Automation practices should ensure that changes to feature definitions, training code, and model configurations automatically emit lineage and deployment metadata to prevent gaps that break correlation.

Operational evaluation should report coverage (fraction of predictions traceable end-to-end), correctness (lineage consistency), and actionability (mean time to detect and recover) alongside cost metrics.

10. Conclusion

Customer AI systems require observability that extends beyond microservice telemetry to include data lineage, feature provenance, and lifecycle traceability for predictions and decisions.

By unifying runtime telemetry with lineage metadata into an Observability Graph, teams can answer high-impact questions such as which upstream data change caused a prediction shift and which feature transformation introduced skew.

Future work includes standardized schemas for feature and prediction events, privacy-preserving cohort analysis techniques, and automated remediation loops that connect observability signals to controlled retraining and rollback pipelines.

References

1. Bowen Li, Xin Peng, Qilin Xiang, Hui Wang, Tao Xie, Jian Sun, Xuandong Liu. Enjoy your observability: an industrial survey of microservice tracing and analysis. Empir Softw Eng. 2022;27(1):25. doi:10.1007/s10664-021-10063-9.
2. Kacheru G. Revolutionizing Healthcare: The Role of Artificial Intelligence in Clinical Practice. J Comput Anal Appl. 2023;31(4):1546–1544. Available from: https://eudoxuspress.com/index.php/pub/article/view/3270
3. OpenTelemetry. OpenTelemetry Specification [Internet]. 2024 [cited 2026 Mar 2[8]]. Available from: https://github.com/open-telemetry/opentelemetry-specification
4. Pittala SK, Ashok VKC. A new era in security: Bridging information security and cybersecurity. Int J Multidiscip Futur Dev. 2023;4(1):69–72. doi:10.54660/IJMFD.2023.4.1.69-72
5. ISO/IEC 23894:2023. Artificial intelligence — Guidance on risk management. Geneva: International Organization for Standardization; 2023. Available from: https://www.iso.org/standard/77304.html
6. Shreya Shankar, Aditya G Parameswaran. Towards observability for production machine learning pipelines. Proc VLDB Endow. 2022;15(13):4015–4022. doi:10.14778/3565838.3565853
7. W3C. Trace Context. W3C Recommendation. 23 Nov 2021. Available from: https://www.w3.org/TR/trace-context/
8. Gunda SK. Comparative Analysis of Machine Learning Models for Software Defect Prediction. 2024 Int Conf Power Energy Control Transm Syst (ICPECTS); 2024; Chennai, India. p. 1–6. doi:10.1109/ICPECTS62210.2024.10780167
9. OpenLineage. Announcing OpenLineage 1.0 [Internet]. 2023 Aug 4 [cited 2026 Mar 2[8]]. Available from: https://openlineage.io/blog/1.0-release/
10. Gupta N, Mujumdar S, Patel H, Masuda S, Panwar N, Bandyopadhyay S, *et al*. Data quality for machine learning tasks. In: Proc. KDD ’21. 2021. p. 4040–4041. doi:10.1145/3447548.3470817
11. Ashok VKC. Cybersecurity for smart infrastructure and public utilities. Int J Multidiscip Res Growth Eval. 2023;4(2):947–949. doi:10.54660/.IJMRGE.2023.4.2.947-949
12. Hewage N, Meedeniya D. Machine learning operations: A survey on MLOps tool support. arXiv:2202.10169 [Preprint]. 2022 [cited 2026 Mar 2[8]]. Available from: https://arxiv.org/abs/2202.10169
13. Grafberger S, Groth P, Stoyanovich J, Schelter S. Data distribution debugging in machine learning pipelines. VLDB J. 2022;31(5):1103–1126. doi:10.1007/s00778-021-00726-w
14. Lin H, Mo J. Open sourcing Feathr – LinkedIn’s feature store for productive machine learning [Internet]. LinkedIn Engineering Blog; 2022 Apr 12 [cited 2026 Mar 2[8]]. Available from: https://www.linkedin.com/blog/engineering/open-source/open-sourcing-feathr--linkedin-s-feature-store-for-productive-m
15. de la Rúa Martínez J, Buso F, Kouzoupis A, Ormenisan AA, Niazi S, Bzhalava D, *et al*. The Hopsworks Feature Store for Machine Learning. SIGMOD ’24; 2024 Jun 9–15.
16. NIST. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. 2023. Available from: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf
17. Schröder T, Schulz M. Monitoring machine learning models: A categorization of challenges and methods. Data Sci Manag. 2022;5(3):213–222. Available from: https://www.sciencedirect.com/science/article/pii/S2666764922000303
18. TensorFlow Extended (TFX). ML Metadata (MLMD) Guide [Internet]. 2024 Sep 6 [cited 2026 Mar 2[8]]. Available from: https://www.tensorflow.org/tfx/guide/mlmd
19. Kacheru G, Bajjuru R, Arthan N. The ROI of software automation: Measuring time and cost savings. Int J Commun Netw Inf Secur. 2023;15(4):774–785.
20. Pittala SK. Cybersecurity and online safety: A critical asset in the information era. J Front Multidiscip Res. 2023;4(1):576–579. doi:10.54660/.jfmr.2023.4.1.576-579
21. Google Cloud. Vertex AI ML Metadata: Introduction [Internet]. 2024 [cited 2026 Mar 2[8]]. Available from: https://docs.cloud.google.com/vertex-ai/docs/ml-metadata/introduction
22. Epperson W, Moritz D, Heer J. Dead or Alive: Continuous data profiling for interactive data science. IEEE Trans Vis Comput Graph. 2024. doi:10.1109/TVCG.2023.3327367
23. ISO/IEC 42001:2023. Artificial intelligence

management system requirements. Geneva: International Organization for Standardization; 2023. Available from: https://www.iso.org/standard/42001

24. Hinder F, Meyer T, *et al*. One or two things we know about concept drift: a survey on unsupervised concept drift detection. Front Artif Intell. 2024 [cited 2026 Mar 2[8]. Available from: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1330257/full
25. Soldani D, *et al*. eBPF: A new approach to cloud-native observability, networking and security for current 5G and future mobile networks (6G and beyond). IEEE Access. 2023 [cited 2026 Mar 2[8]. Available from: https://iris.polito.it/retrieve/6af0ca2f-0b07-496e-9ec9-ba781c535002/eBPF_A_New_Approach_to_Cloud-Native_Observability_Networking_and_Security_for_Current_5G_and_Future_Mobile_Networks_6G_and_Beyond.pdf
26. Sivva SD, Thalakanti RR, Bandari SSG, Yettapu SDR. AI-driven decision intelligence for agile software lifecycle governance: An architecture-centered framework integrating machine learning defect prediction and automated testing. Int J Eng Tech Comput Sci Inf Technol. 2023 Dec 30;4(4):167–172. Available from: https://www.ijetcsit.org/index.php/ijetcsit/article/view/554
27. Gunda SK. Fault prediction unveiled: Analyzing the effectiveness of random forest, logistic regression, and KNeighbors. 2nd Int Conf Self Sustain Artif Intell Syst (ICSSAS); 2024; Erode, India. p. 107–113. doi:10.1109/ICSSAS64001.2024.10760620
28. Ustunboyacioglu I, Genc G, *et al*. Data quality assessment in the wild: Findings from GitHub. In: Proc. ESEC/FSE 2024. 2024. doi:10.1145/3661167.3661213