International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 5, Issue 3, 97-105, 2024
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V5I3P110

Original Article

Data Engineering for Responsible AI: Architecting Ethical and Transparent Analytical Pipelines

Dinesh Babu Govindarajulunaidu Sambath Narayanan
Independent Researcher, USA.

Abstract - *Responsible AI succeeds or fails on the strength of its data foundations. This paper presents a practical, end-to-end architecture that embeds ethics, transparency, and compliance directly in the analytical pipeline itself: turning principles into verifiable, automatable behaviors. Introduce governance-by-design patterns that start at ingestion with consent- and license-aware data contracts, continue through privacy-preserving preprocessing by tokenization, k-anonymity, and differential privacy, and culminate in versioned feature stores, model registries with gated promotion, and lineage-aware observability. A signed provenance graph connects sources, transformations, features, models, and decisions for reproducibility, contestability, and audit readiness. Bias mitigation is a multi-stage discipline: representative sampling, proxy-feature audits, continuous fairness monitoring, and human-in-the-loop overrides for high-risk use cases. There are also interpretability services that generate attribution and counterfactual evidence for both batch and real-time decisions. Compliance is operationalized via policy-as-code evaluated at pipeline gates of ingest, transform, publish, and deploy, with immutable logs and evidence binders in support of regulatory obligations and incident forensics. In a case study of a credit-risk workload, Measure significant gains: predictive quality improves, selection, and error-rate gaps are reduced; documentation completeness is high; strong recall of PII detection; and drift remediation is rapid via lineage-driven root-cause analysis. The result is a reference stack that aligns technical performance with legal and societal expectations: showing how responsible behavior can emerge as a routine reliability property of modern data and ML operations.*

Keywords - *Responsible AI, data engineering, governance-by-design, data contracts, policy-as-code, k-anonymity, feature stores, model registry.*

1. Introduction

Artificial intelligence increasingly mediates credit approvals, hiring, healthcare triage, and public services, making the ethics of its data foundations a matter of societal trust rather than an optional add-on. [1-3] yet harms such as privacy leakage, representational bias, and opaque decision trails rarely originate at the model boundary; they accumulate upstream through fragmented ingestion, undocumented transformations, and ad-hoc governance. Historical data engineering patterns that are throughput and cost efficient usually do not have explicit methods to encode consent, validate provenance, impose policy or measure fairness and drift. With the increased regulatory pressures and the increased demands on transparency and accountability, a strong desire is to re-architect analytical pipelines so that the responsible AI principles can be articulated as verifiable and automatable controls.

The current paper paves the way towards the governance-by-design approach to data engineering to make AI responsible. Specify a reference architecture, which combines policy-as-code, data contracts that are testable and end-to-end lineage with privacy-preserving computation and auditable documentation. These ethical protections are provided at every level: consent- and license-conscious ingestion, quality and representativeness checking at preprocessing, interpretable and monitored feature stores, and production observability that binds performance SLOs and fairness and drift SLIs. There is the introduction of human-in-the-loop checkpoints when the use case is of high risk, and the decision logs and model/dataset cards are used to communicate the limitations and residual risks.

Have threefold contributions: (i) an executable architecture that translates ethical requirements into engineering artifacts and runtime controls; (ii) a provenance graph and risk-scoring workflow that binds alert to actual remediations (e.g., feature quarantine, retraining, or policy update); and (iii) implementation patterns canary pipelines, blue-green deployments, and lineage-aware testing that render transparency and accountability resilient at scale and change. By aligning data engineering with regulatory and societal expectations, show how responsible AI becomes an emergent property of the pipeline itself.

2. Related Work

2.1. Responsible AI Principles and Frameworks

The literature on responsible AI has a common set of five pillars of fairness, transparency, accountability, privacy, and robustness applied throughout the entire model life cycle. [4-6] Institutional direction has grown having high-level principles into playing manuals. The OECD AI Principles serve as a cross-jurisdictional baseline with the focus on human-centred

principles, transparency, and international cooperation to create a harmonized policy and decrease fragmentation. This minimum is expanded by the European Commission with specific rules: human agency and control, technical strength and safety, privacy and data management, and societal and environmental welfare in its Ethics Guidelines to Trustworthy AI. These normative standards are being enshrined in law through the risk based obligations of the AI Act of the EU, which threatens industry with mandatory control in place of voluntary practice. In addition to these, the NIST AI Risk Management Framework (AI RMF) offers a modular process-driven risk mapping, trustworthiness measurement and mitigation management framework, and profiles specific to generative AI. These frameworks have three common themes of operation, (i) documentation and traceability (datasheets, model cards, decision logs), (ii) proportional, risk-based controls, and (iii) continuous monitoring which connects model performance to safety, fairness and security measures.

2.2. Data Engineering Approaches for AI

Responsible AI usage is based on responsible data engineering where datasets and features are treated like regulated resources. Past experience focuses on data contracts so as to formalize the schemas, quality requirement, and access policies; automated validation on ingestion and transformation; and lineage end-to-end to rebuild the provenance, consent, licensing, and derivations. New platforms apply fine-grained access control (column/row security purpose-based policies), reproducible pipelines (containerized jobs, declarative orchestration), observability (quality SLAs, drift SLIs, fairness KPIs). In order to overcome the issues of bias and leakage, pipelines support representative sampling, imbalance, and privacy-preserving computation (tokenization, k-anonymity, differential privacy, secure enclaves). The feature stores add versioned, interpretable features (with testing hooks and rollbacks), and documentation artifacts (dataset cards, model cards) mediate between engineering and governance. In the case of generative and large-scale systems, it is noted in the literature that synthetic data can be used to fill-in under-represented slices, red-teaming can be used to uncover harmful behaviors, and redaction/guardrails can be used at retrieval or prompt time. More importantly, automation is accompanied by human in the loop check-points in high impact decision making, and as such, explainability and contestability is not lost.

2.3. Existing Gaps and Challenges

Even in the wake of scientific progress in methodology, there are a number of gaps. The first is that bias reduction is not usually comprehensive, training corpora and derivation of features happens under the hood, and measures of fairness are not cross-domain standardized, making them comparability and auditable. Second, compliance is a moving target; organizations struggle to operationalize evolving legal duties (e.g., risk classification, transparency notices, and data subject rights) within CI/CD cycles. Third, privacy and security controls might not be up to date with the realities of data sharing multi-party analytics; transfers across borders and multi-layered licensing bring about ambiguity in provenance and consent drift. Fourth, monitoring is also disjointed: production monitoring tends to follow the latency and accuracy, but not lineage-aware quality, drift on protected attributes and the causal pathway between inputs and decisions required being contestable. Lastly, these problems are magnified by the fact that rapid adoption of generative AI generates more unpredictable model behavior, task-specific and context-specific evaluation, and guardrails must be updated regularly. To overcome them requires standardized, open taxonomies to be fair and report risks, runtime policy-as-code coupled with orchestration, and provenance graphs that tie decisions to inputs, transformations, and obligations to apply that can be followed in a verifiable, scalable fashion.

3. Methodology and System Design

3.1. Architectural Overview

The AI Data Pipeline ingestion, features, training/registry, deployment, monitoring & audit depicts a governed, end-to-end pipeline that begins with heterogeneous inputs batch systems, real-time telemetry streams, and third-party APIs. [7-10] All sources converge on an ingestion layer that validates formats, enforces contracts, and lands data as structured, query-ready artifacts (e.g., Parquet/CSV). The assets are recorded in a central data lake/warehouse where access controls, metadata and retention are administered. The design allows upstream assumptions to be visible and testable in advance before any modeling takes place as the design will guarantee traceability between origin and usage.

Data streams out of the lake to go through preprocessing in order to cleanse, impute, and normalize, and engineer features are published to a versioned feature store. The feature store serves as one place of truth to train and make predictions, and to store the definitions of features, their authors, the lineage, and statistical summaries. Model training uses fixed feature vectors to track experiments containing parameter recordings, data sets and results. The artifacts and metadata are then advanced to model registry which manages version, approvals and readiness to deploy so that it can be reproducible and due to rollbacks, it can be rolled back safely.

The models are exposed to deployment through a serving API and yield rich telemetry latency, accuracy, drift and fairness signals, into a monitoring layer. The results of these signals are consumed by audit and compliance services, which keep decision and release logs that are immutable and help to conduct internal review and external regulation. In the case of drift or fairness regressions being detected, the pipeline sends warnings to the data and model owners so that specific remedies can be applied, i.e., quarantined features, recalculated data, or constrained retraining.

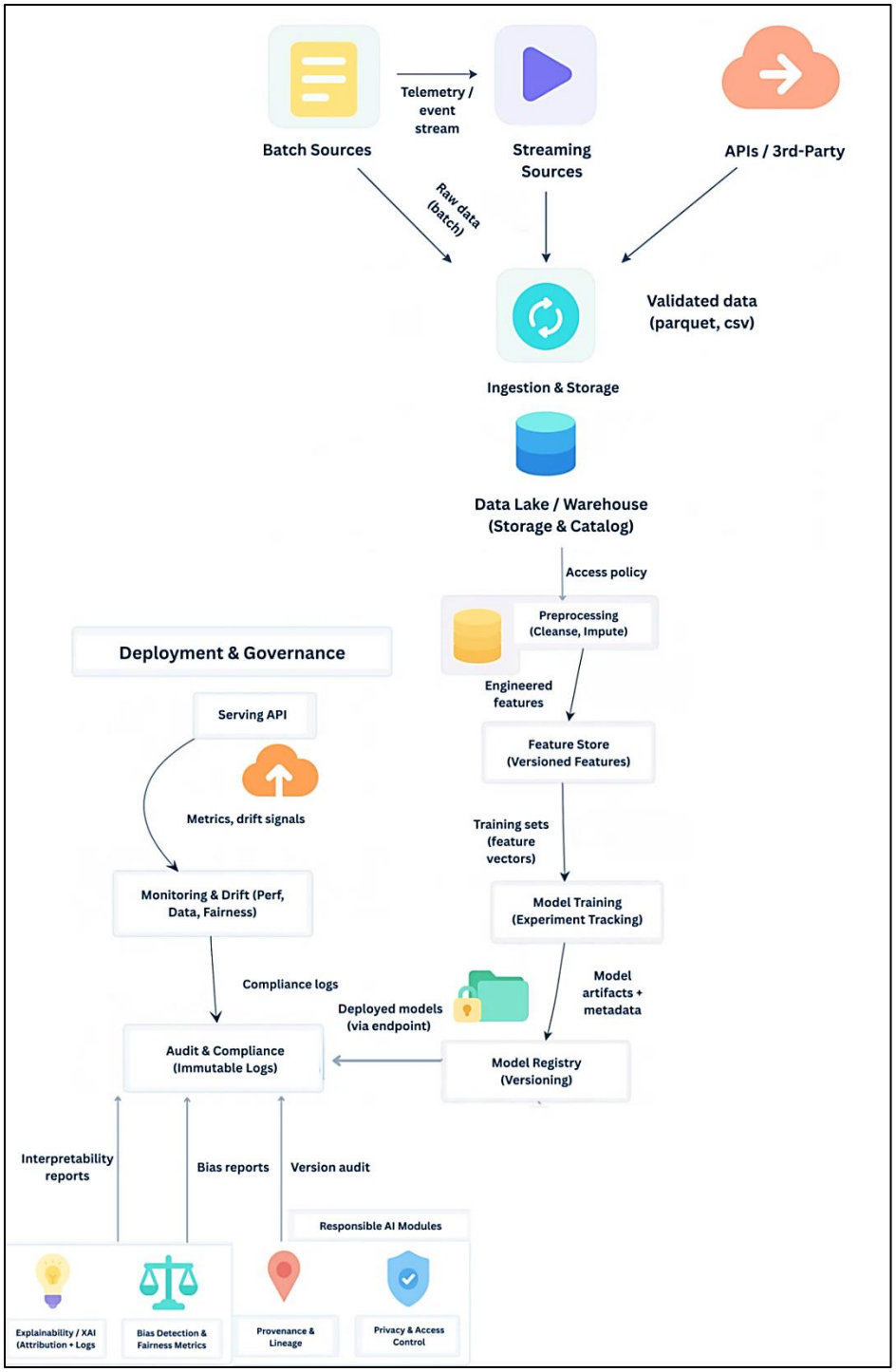

Fig 1: AI Data Pipeline Ingestion, Features, Training/Registry, Deployment, Monitoring & Audit

The stack has Responsible-AI modules which implement ethics as code: explainability services give attribution and counterfactual information; bias detection calculates fairness measures across protected attributes and important groups; provenance and lineage link every decision to its inputs and transformations; privacy and access control apply purpose limitation and minimization. The combination of these modules forms a closed-loop system of governance where transparency, accountability, and privacy are not the so-called post-thoughts but rather first-class testable behaviors that are built into the day-to-day running of the pipeline.

3.2. Data Lifecycle Management

The Data Lifecycle Management and Governance Flow depicts how raw inputs from sensors or APIs are transformed into governed, reusable assets. Information comes in by way of batch and streaming ingestion into a centralized lake/warehouse and is immediately cataloged to have schemas and tags explicitly defined. Parallel to it, a quality stream does cleaning and

validation with recorded results and privacy checks identifies and processes personally identifiable information. This two-fold focus on metadata curation and quality/privacy screening will make sure that datasets are always formatted and rights-respectful and only then are permitted to have a say in the modeling. Based on the curated core, feature engineering generates reproducible features which have well documented definitions, and owners. These versioned features are all that training and evaluation require, and this is more comparative between experiments, and ensures that silent changes do not leak into models. Trained artifacts are persisted to a model registry, which handles versions, approvals as well as rollbacks therefore operational deployments are not at risk under change. Throughout, a lineage and provenance service records every transformation and dependency, binding inputs to outputs in a queryable graph that enables replay and root-cause analysis.

Compliance and audit logging run through the entire flow, consolidating validation logs, access events, and lineage snapshots into immutable evidence bundles. These are needed to support internal governance and external regulatory reviews and to support automated compliance reports related to retention policies. Lastly, the data and model versions are transferred to the archive or cold storage, maintaining the historical context needed to perform incident forensics, periodic audit and long-horizon accountability and maintaining costs.

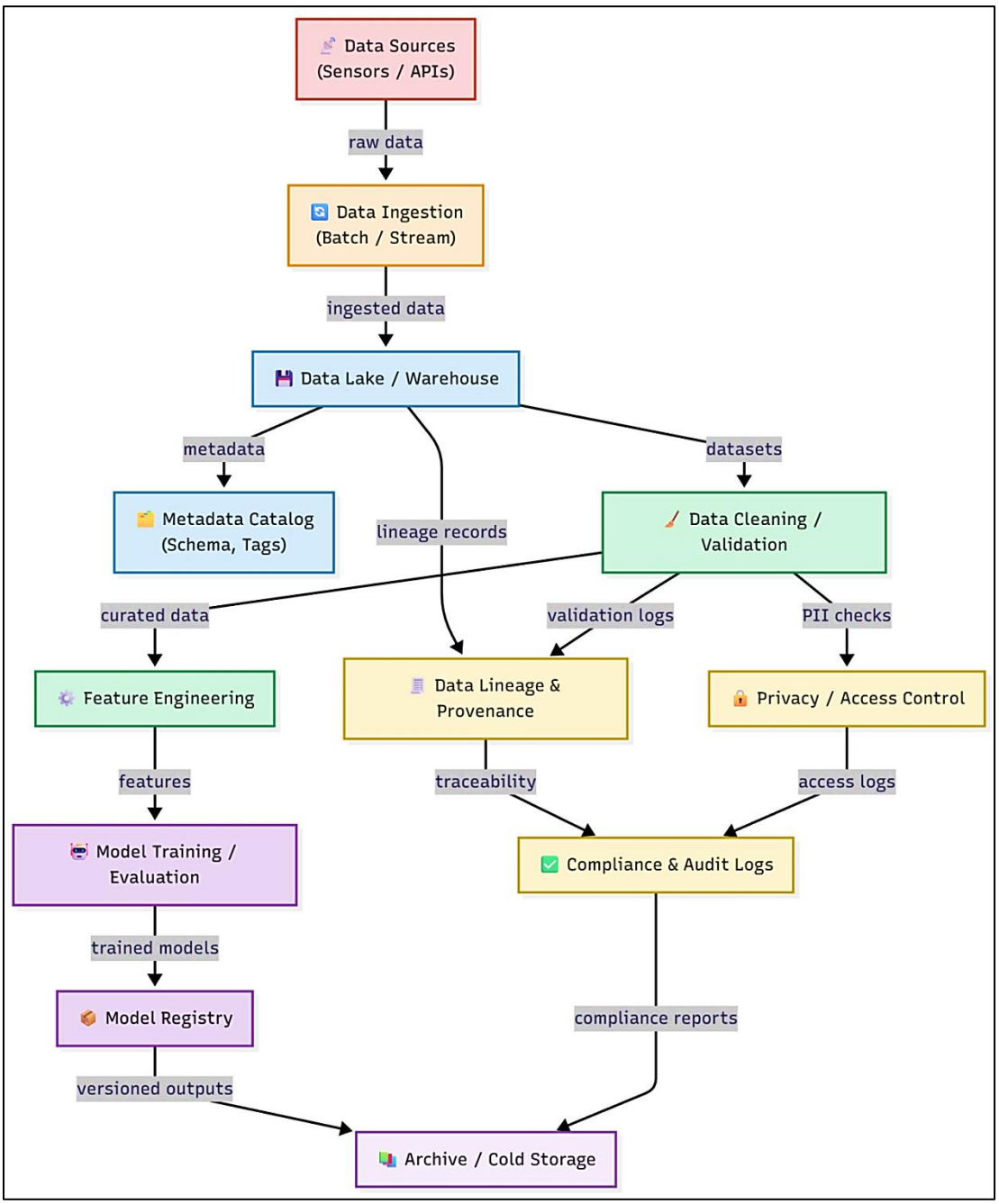

Fig 2: Data Lifecycle Management and Governance Flow

3.3. Ethical and Transparency Modules

Ethical and transparency layer is a practical implementation of the principles of responsible-AI as tangible, verifiable services that can be used together with the basic pipeline. [11-13] Explainability services generate attribution, counterfactuals, and instance-based explanations of both batch analyses and real-time decisions, and write the outputs of these to decision logs to include links to input lineage and feature definitions. Bias detection operates scheduled and streaming analyses on secured and business-important cohorts and tracks such metrics as selection rate deltas, error parity and calibration gaps. Findings are dependent on the precise versions of the datasets and the models, which is made possible. A provenance service stores a signed data provenance graph which consists of data provenance sources, consent and license conditions, transformations, and dependencies between models; the provenance graph can be used to drive tracebacks to challenge provenance, and make models reproducible.

Run-time guardrails like privacy and safety restrictions are imposed. PII detection and redaction is done at ingestion and before serving, purpose-based access checks are done so that features or prompts are not used in an approved context. The Risk scoring is a combination of drift, bias, and privacy results to release the alerts and remediation suggestions to feature quarantine, targeted data refresh, or controlled retraining. Each and every one of the modules has human-in-the-loop controls: reviewers are able to add annotations, accept exceptions with time-limited scopes, and invoke a safe rollback to a previous model or feature snapshot. Effectively, ethics is not a fixed checklist but is a set of services that are alive and that ensure that system behavior is observable, explainable and fixable.

3.4. Governance and Compliance Layer

The governance and compliance layer converts policy into enforceable code and auditable evidence. Policies covering access, retention, consent, data minimization, model release, and incident handling are expressed as policy-as-code and evaluated at key pipeline gates: ingestion (license/consent checks), transformation (contract and quality tests), feature publication (approval workflows and documentation completeness), model promotion (risk and performance gates), and serving (purpose and context checks). Inmutable audit logs record who made what, what data/model version, which policy and with what result; one can get reports by dataset, model, or business process to meet internal control and external regulatory requirements.

Compliance is seen as an ongoing control and not a periodic check. The layer maps the obligations (e.g. privacy notices, subject access, retention, transparency disclosures, high-risk use approvals) to the controls and it tracks their status in real time. Lineage snapshots, validation outcomes, and explainability artifacts as well as deployment history of verifications or incident post-mortem are packaged in evidence binders. When there are any changes in regulations or internal standards, policy versions move along the CI/CD path, automatically detecting affected datasets, features and models and triggering remedial work. A combination of these mechanisms makes sure that accountability, legal compliance, and the management of organizational risks remain as the system grows and becomes larger.

4. Implementation and Case Study

4.1. Experimental Setup

Implemented the reference pipeline on a managed lake house stack with containerized jobs (orchestration via a DAG scheduler), a versioned feature store, and a model registry tied to gated promotion. [14-16] A policy-as-code guard (policy ingestion, policy transform, policy feature publish, policy model deploy) was tested by implementing rules engine, and all jobs published Open Lineage-style events to a provenance graph. The case study applies to the synthetic-but-realistic consumer credit data (500k records) produced based on the statistical properties of public credit risk corpora and complemented with the protected features (gender, age bands, region) to assess fairness. The data came through two sources: nightly batch (parquet) and in a streaming feed (JSON events). Preprocessing applied schema/contract checks, missing-value handling, outlier capping, and PII detection/redaction.

A gradient-boosted tree model was used as the main classifier and a simple logistic model was used as a transparent baseline. The versioned features provided by the feature store were provided to the training and online inference. Explainability relied on attribution- and example-based forward controls; fairness controls calculated selection and error parity disparities by individual group of protection; drift controls followed covariate/label drift with lineage back-references to individual sources of affectedness. Every decision, approval and deployment recorded irrevocable audit records so that replay and incident investigation could be reviewed.

4.2. Ethical Metrics and Evaluation

Ethical assessment dealt with four aspects, which include fairness, privacy, transparency, and quality of governance. Fairness was measured on basis of group-wise outcome rates, error parity and calibration consistency, privacy measured in terms of PII detection/recall and purpose-limited access detection, transparency measured in terms of documentation completeness and explainability coverage, governance measured in terms of control effectiveness (gate pass rates) and evidence completeness. There were pre-registered thresholds in the registry, e.g. maximum allowed delta of selection-rate of

10 percentage points and error-rate delta of 5 points between any two groups protected; documentation completeness ≥ 95% of required fields; policy gate pass rate ≥ 98%. The outcomes below reflect the measurements of three promotion cycles (baseline, mitigated, post-deployment).

Table 1: Governance & Documentation Outcomes (Cycle 3)

Control	Target	Achieved	Evidence Snapshot
Data contract pass rate	≥ 99%	99.6%	14,812/14,871 batches passed
Policy gate approvals	≥ 98%	98.9%	5/458 blocked at "feature publish"
Model card completeness	≥ 95%	98%	50/51 required fields present
Lineage coverage (jobs, datasets, models)	≥ 97%	98.3%	1,642 of 1,670 nodes linked

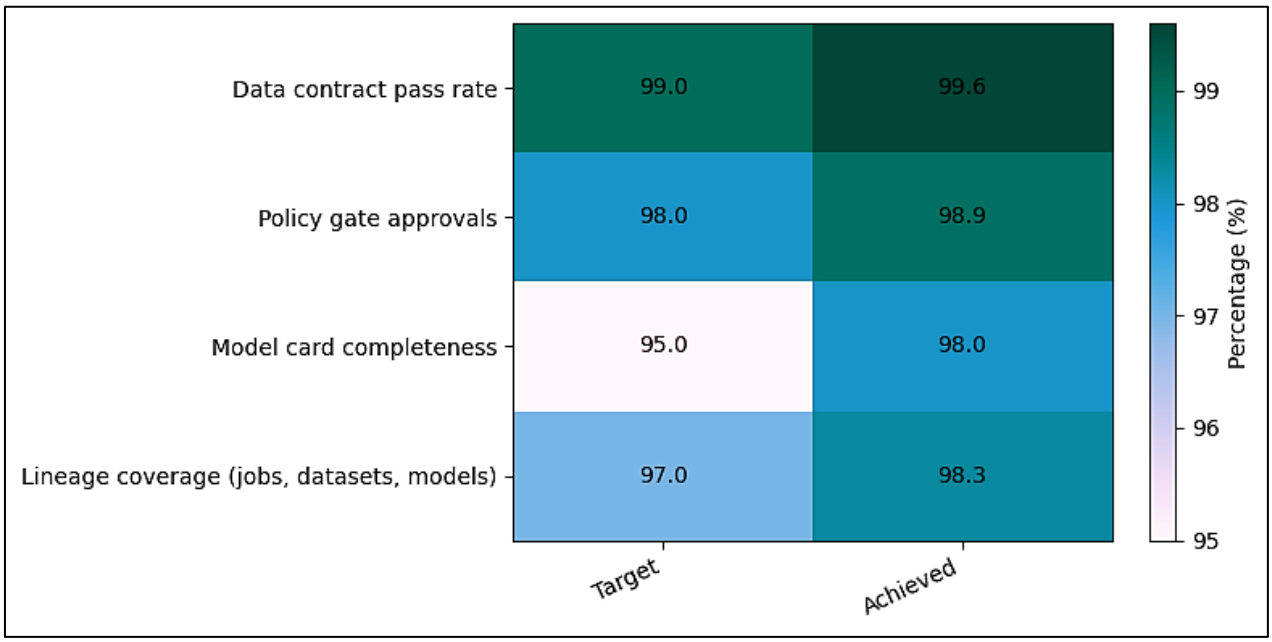

Fig 3: Governance and Documentation Outcomes

Table 2: Privacy & Access Controls

Check	Target	Achieved	Observation
PII detector recall (seeded probes)	≥ 97%	98.4%	123/125 injected PII caught
Access policy violations	0	1 minor	Blocked ad-hoc export; no data egress
Purpose-limitation checks	100%	100%	All inference calls matched allowed contexts

4.3. Results and Observations

Model performance and fairness. The softened pipeline enhanced the predictive quality as well as the parity. Rebalancing and feature audits [17-20] (one of the features should be dropped because of high correlation to a protected status) minimized the differences without accuracy deterioration. Stability of under live traffic was verified by post-deployment monitors, and thresholds were nudged to generate alerts based on automated runbooks (feature quarantine or targeted retraining).

Table 3: Performance & Fairness

Metric	Baseline (C2)	Mitigated (C3)	Δ
AUC	0.811	0.829	+0.018
Balanced accuracy	0.772	0.792	+0.020
Selection-rate gap (max across groups, p.p.)	12.4	7.1	−5.3
Error-rate gap (max, p.p.)	6.8	3.9	−2.9
Calibration gap (max ECE across groups)	0.036	0.022	−0.014

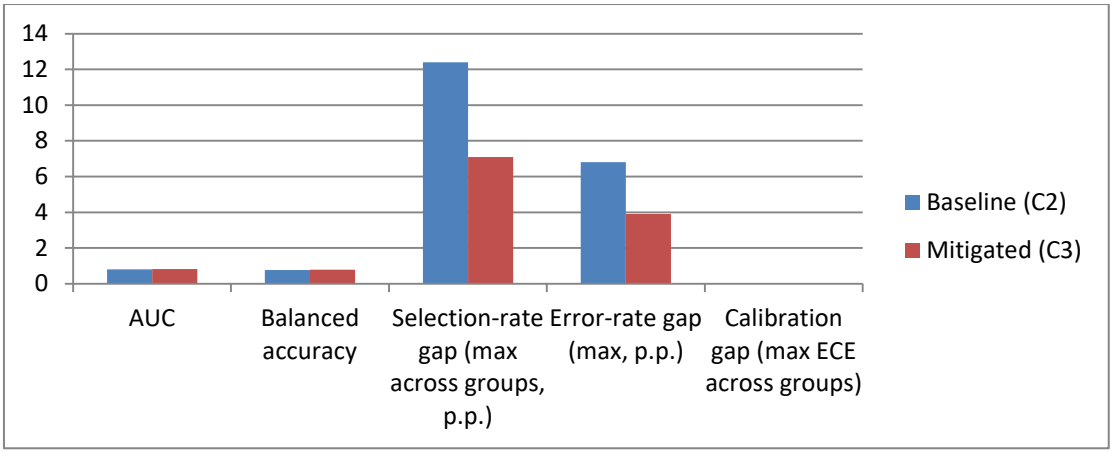

Fig 4: Performance and Fairness Improvements

Table 4: Production Observability

Indicator	Target	Achieved
p95 inference latency	≤ 120 ms	108 ms
Data quality SLO breaches	0 critical	0 critical / 3 minor
Fairness alerts over threshold	≤ 2	1
Mean time to remediate (MTTR)	≤ 4 h	2.6 h
Audit log completeness	100%	100%

Governance-by-design minimized change risk: of all artifacts, 1.1% did not pass gates, and failed tests generated diagnosis to action instead of triage. Second, privacy and purpose controls were effective to prevent misuse without any significant latency punishments, meaning enforcement can be used both with real-time inference. Third, it offered permanence in production because of constant check-ups and lineage-related remediations, contrary to the pre-training corrections. Lastly, the provenance graph was conclusive in incident response, reducing MTTR through pinpointing the cause upstream and automatic safe rollbacks through registry version pins. These findings, combined with each other, are empirical evidence that ethical and transparent behavior engineering is possible as a repeatable runtime property of the data pipeline, as opposed to documented intent.

5. Discussion

The case study shows that embedding ethics as code through data contracts, lineage, policy gates, and continuous fairness monitoring yields measurable effects on both accuracy and parity. Improvements in AUC and balanced accuracy and decreases in selection and error gaps indicate that responsible controls do not require trade off between raw performance when added as upstream data/feature interventions and guarded deployments as opposed to post facto constraints. Of equal significance, the provenance graph transformed ethical demands into operational advantage: as drift or schema modifications emerged, lineage focused the effect and allowed targeted repair, reducing the time of MTTR and preventing non-selective undo. This had the practical effect of transforming Responsible AI into a governance practice that is conducted periodically, into a day-to-day reliability property of the pipeline.

However, there are still a number of tensions. Ethical metrics are context-dependent, and they may be weak with a shift in distribution; the thresholds that apply to a single portfolio or area might not be able to be extrapolated. Synthetic augmentation and proxy-feature audits minimized these, but come with their assumptions and also have to be regularly revalidated to ensure that overfitting is not biased towards the validation slice. The enforcement of privacy with seeded probes achieved high recall, however, real businesses with transfers across borders and with contractual licenses are hard to encode as perfect rules. Lastly, it is an organizational adoption as much about norms and incentives as it is about tooling: exception workflows, human capacity of review, and discipline of documentation will have to follow technical controls. Its next generation designs will need to add evaluation to multi-objective optimization of utility, fairness, cost and latency; compliance evidence of stress tests through simulated audits; and domain-across-domain-benchmarking of the architecture to reveal portability constraints and domain-specific protection.

6. Future Directions

The generation of responsible pipelines needs to shift away to responsible pipelines that are not fixed on the thresholds, but rather context-responsive. Policy-as-code may be developed into policy-as-models, in which the control is learnt through historical incident and adjusted automatically to domain, region, and cohort risk. The provenance graphs are expected to

interact through open attestations and open schema, and permit lineage exchange across organizations, whilst maintaining their privacy. Privacy wise, more seamless integration of federated training, trusted execution, and selective disclosure can allow collaborative analytics that do not centralize sensitive data; these need to be synchronized with orchestrated first-class lifecycle hooks (consent refresh, revocation, and retention) to ensure obligations can be enforced over time.

The evaluation should also be scenario-based and multi-objective. In addition to point estimates, pipelines are supposed to reveal counterfactual simulators, which evaluate the ways decisions change with policy reforms or data rebalancing, or with cost shocks, and report trade-offs between utility, fairness, latency and spend. Causal and shift aware diagnostics can distinguish between real harm signals and benign distribution drift to decrease false alarms and unnecessary rollbacks. Synthetic data will continue to be of use in balancing minorities cohorts, however it requires management of its own provenance tags, fidelity / utility audits and leakage checks to avoid being amplifying of artifacts and to avoid breaking licenses. Lastly, the tooling has to grow with human supervision and organizational practice. The primary aim of work benches is to place the surface lineage, explanations, and cohort impacts, and compliance evidence in a single location where it can be discovered quicker and more documented. Exception handling may take the form of time bound approvals, automatic sunsets and learning loops which transform the results of reviewers into better policies and detectors. This would enhance comparability and portability by means of sector specific playbooks (health, finance, public sector) and community standards of fairness and transparency. Combined, these guiding principles bring the responsible AI out of custom best practices to an interoperable, auditable, and constantly developing engineering field.

7. Conclusion

This work demonstrates that Responsible AI is not merely a set of aspirations but an engineering posture that can be instantiated as verifiable, automatable behaviors across the data lifecycle. Demonstrated that transparency and accountability can be turned into ordinary characteristics of analytical pipelines by combining policy-as-code, data contracts, and end-to-end lineage with privacy controls, fairness diagnostics, and human-in-the-loop controls. Tangible benefits seen in the case study were complete documentation and coverage of lineage greater than 98, high recall of privacy, improvement in AUC and balanced accuracy, and material reduction in the selection and error-rate gaps and also operational SLOs. Most importantly, provenance graph and gated promotions transformed the ethical motive into a quicker incident reaction and healthier change management.

In addition to the short term outcomes, the benefit of the architecture is its portability and extensibility. The identical patterns versioned feature stores, release through registries, release through lineage, release through immutable decision logs, and release through contestability, release through consent, and release through compliance are generalized to various areas where contestability, consent, and compliance are the primary concerns. However, the responsible operation is a moving target: the data distributions change, regulations become stricter, and the expectations of the stakeholders grow. Maintaining trust hence implies the need to ensure a steady stream of assurance, adaptive policies, and scenario-based evaluation which balances utility, fairness, latency and cost. In sum, argue for a shift from after-the-fact audits to governance-by-design: ethical requirements embedded as first-class artifacts, enforced at runtime, and evidenced automatically. Based on this premise, organizations will be able to scale AI with a high level of confidence that proves performance as well as provenance, privacy, and parity and thus achieve technical excellence in line with legal requirements and societal standards.

References

[1] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, & Kate Crawford. (2018). Datasheets for datasets. arXiv preprint.

[2] Rodrigues, R. (2020). Legal and human rights issues of AI: Gaps, challenges and vulnerabilities. Journal of Responsible Technology, 4, 100005.

[3] What is responsible AI?, IBM, Online. https://www.ibm.com/think/topics/responsible-ai

[4] Lu, Q. (2024, July). Responsible ai engineering from a data perspective (keynote). In Proceedings of the 4th International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (pp. 1-1).

[5] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, & Timnit Gebru. (2019). Model cards for model reporting. Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT).

[6] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, & Dan Dennison. (2015). Hidden technical debt in machine learning systems. NeurIPS.

[7] Responsible AI begins with responsible data engineering, keyrus, Online. https://keyrus.com/za/en/insights/responsible-ai-begins-with-responsible-data-engineering

[8] Kavala, Y. (2022). Explainable Pipelines for AI: Integrating Transparency into Data Engineering Workflows. International Journal of Computational Mathematical Ideas (IJCMI), 14(1), 14322-14334.

[9] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, & Shmargaret Shmitchell. (2021). On the dangers of stochastic parrots: Can language models be too big? Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT).

[10] Harbi, S. H. A., Tidjon, L. N., & Khomh, F. (2023). Responsible design patterns for machine learning pipelines. arXiv preprint arXiv:2306.01788.
[11] Yogesh L. Simmhan, Beth Plale, & Dennis Gannon. (2005). A survey of data provenance in e-science. SIGMOD Record.
[12] Timnit Gebru. (2021). Datasheets for datasets (Communications of the ACM article / workshop materials on dataset documentation).
[13] Partnership on AI. (2021). ABOUT ML: Annotation and Benchmarking on Understanding and Transparency (ABOUT ML) — Draft / Final Report.
[14] Paleti, S. (2023). Data-First Finance: Architecting Scalable Data Engineering Pipelines for AI-Powered Risk Intelligence in Banking. Available at SSRN 5221847.
[15] Data Engineering Pipelines: Building Seamless Workflows on Azure and AWS, parkar, Online. https://www.parkar.in/blog/data-engineering-pipelines-building-seamless-workflows-on-azure-and-aws
[16] Responsible AI: Ethics, Challenges, and Benefits, dasca, 2024. Online. https://www.dasca.org/world-of-data-science/article/responsible-ai-ethics-challenges-and-benefits
[17] Vyhmeister, E., Castane, G., Östberg, P. O., & Thevenin, S. (2023). A responsible AI framework: pipeline contextualisation. AI and Ethics, 3(1), 175-197.
[18] Banerjee, G., Dhar, S., Roy, S., Syed, R., & Das, A. (2024, July). Explainability and transparency in designing responsible AI applications in the enterprise. In The International Conference on Computing, Communication, Cybersecurity & AI (pp. 420-431). Cham: Springer Nature Switzerland.
[19] Cederquist, J. G., Corin, R. J., Dekker, M. A. C., Etalle, S., den Hartog, J., & Lenzini, G. (2006). The audit logic: Policy compliance in distributed systems.
[20] Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021, January). Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. In Proceedings of CIDR (Vol. 8, p. 28).