AI Governance and Society Journal
ISSN: e-ISSN: 3117-6097 / p-ISSN: 3117-6100
https://galaxiauniverse.com/index.php/AIGSJ/issue/archive

# Embedding Ethical Constraints into End-to-End AI Pipelines

**Daniel Dubois[1], Nina Costa[2]**

[1] Professor, Department of Computer Science, Nordic Technical University, Stockholm, Sweden. Email: daniel.dubois3@gmail.com | ORCID: 6022-8526-0031-0420
[2] Research Scientist, Department of Computer Science, Baltic AI Research University, Tallinn, Estonia. Email: nina.costa307@gmail.com | ORCID: 1658-2673-4489-7405

## ABSTRACT

*End-to-end artificial intelligence (AI) pipelines -- spanning data ingestion, feature engineering, model training, serving, and monitoring -- introduce multiple stages at which ethical violations may emerge, propagate, or amplify. Existing approaches to AI ethics largely address individual pipeline stages in isolation, leaving cross-stage ethical constraint propagation and inter-stage dependency management underexplored. This paper proposes the Ethical Constraint Propagation (ECP) framework, a principled approach to embedding fairness, transparency, privacy, and accountability constraints across all stages of end-to-end AI pipelines through explicit constraint declaration, automated propagation, and stage-level verification gates. The ECP framework is evaluated on n = 20 production AI pipelines drawn from four industry sectors -- healthcare, finance, retail recommendation, and smart infrastructure -- through a combination of static pipeline audit and six-month longitudinal monitoring. Pipelines adopting the full ECP framework exhibited a 43.8% reduction in cross-stage ethical violations (SD = 5.1%) and a 31.4% improvement in end-to-end fairness consistency (SD = 4.6%) relative to unmodified baselines. Privacy constraint leakage -- defined as privacy guarantees established at the data ingestion stage failing to propagate to downstream serving components -- was eliminated in all ECP-compliant pipelines. The study contributes the ECP framework specification, a constraint propagation graph formalism, and an empirical benchmark dataset for ethical AI pipeline evaluation.*

**Keywords:** ethical AI pipelines; constraint propagation; fairness; privacy; end-to-end AI; responsible AI; pipeline audit; EU AI Act

**Citation:** Dubois and Costa [2024]. Embedding Ethical Constraints into End-to-End AI Pipelines. DOI: https://doi.org/10.5281/zenodo.19184952



**Article Information:** *Received:* 10 June 2024 *Accepted:* 12 August 2024 *Published:* 15 October 2024

**Research Article:** *Research Article*

ISSN: e-ISSN: 3117-6097 / p-ISSN: 3117-6100 Vol: 2024, Issues: 4, 2024

## 1. Introduction

### 1.1 The Pipeline Ethics Problem

Modern AI systems are built as multi-stage pipelines in which raw data passes through a series of transformations -- ingestion, preprocessing, feature engineering, model training, inference serving, and post-deployment monitoring -- before producing outputs that affect real users (Sculley et al., 2015; Amershi et al., 2019). Each stage in this pipeline introduces distinct opportunities for ethical constraint violation: biased sampling at ingestion, information loss during feature engineering, fairness degradation during model training, privacy leakage at serving, and accountability gaps in monitoring (Barocas et al., 2019; Dwork et al., 2006). Critically, ethical constraints established at one stage may fail to propagate to downstream stages, producing a phenomenon we term constraint leakage -- the erosion of ethical guarantees across pipeline boundaries. The regulatory landscape intensifies the urgency of pipeline-level ethics management. The EU AI Act (2024) requires high-risk AI systems to maintain end-to-end data governance (Article 10), lifecycle risk management (Article 9), and continuous logging across all operational stages (Article 12). These requirements are not satisfiable through point-in-time audits of individual pipeline stages; they demand systematic constraint propagation across the full pipeline. Yet existing responsible AI toolkits -- including IBM AI Fairness 360 (Bellamy et al., 2019), Microsoft InterpretML (Nori et al., 2019), and Google What-If Tool (Wexler et al., 2020) -- address specific stages in isolation and provide no mechanism for cross-stage constraint consistency verification. The consequence of this gap is practically significant. A pipeline that applies differential privacy at data ingestion but fails to propagate the epsilon budget constraint to feature engineering or serving may inadvertently invalidate its privacy guarantee through auxiliary inference or feature reconstruction attacks (Fredrikson et al., 2015). A fairness constraint applied at model training as an equalised odds objective may be undermined by post-processing calibration steps that restore discriminatory score distributions (Hardt et al., 2016). These cross-stage violations are structurally invisible to stage-level audits and require a pipeline-wide constraint management approach.

### 1.2 Research Objectives and Contributions

This paper proposes the Ethical Constraint Propagation (ECP) framework, which treats ethical constraints as first-class pipeline artefacts declared at pipeline design time, propagated through a constraint graph across all pipeline stages, and verified at stage-level gates before downstream execution proceeds. The research objectives are: (1) to formalise the ECP constraint propagation graph and stage-level verification protocol; (2) to evaluate the ECP framework on production AI pipelines across four industry sectors; and (3) to quantify the reduction in cross-stage ethical violations and improvement in end-to-end fairness consistency associated with ECP adoption. The paper is structured as follows. Section 2 reviews related work on AI pipeline ethics, constraint propagation in software engineering, and regulatory requirements. Section 3 presents the ECP framework. Section 4 describes the evaluation methodology. Section 5 reports results. Section 6 discusses implications and limitations. Section 6 concludes.

## 2. Literature Review

### 2.1 AI Pipeline Architecture and Ethics Challenges

The architecture of production ML systems was systematically characterised by Sculley et al. (2015), who identified hidden technical debt arising from entanglement, undeclared data dependencies, and feedback loop instability in multi-stage pipelines. Subsequent work by Kleppmann (2017) on data-intensive application design established the concept of data lineage -- the traceable provenance of data transformations across pipeline stages -- as a prerequisite for reasoning about correctness and integrity in distributed data systems. Ethics-specific challenges in AI pipelines have been studied primarily at individual stages. Barocas and Selbst (2016) analysed disparate impact arising from training data selection and preprocessing decisions. Feldman et al. (2015) demonstrated that feature removal does not eliminate proxy discrimination in downstream models. Hardt et al. (2016) showed that post-processing calibration for fairness can violate privacy by enabling auxiliary inference on protected attributes. These findings collectively motivate a cross-stage perspective on pipeline ethics, but no prior work has formalised constraint propagation across the full pipeline as a design principle. Table 1 summarises prior work on pipeline-stage ethics challenges.

### 2.2 Constraint Propagation in Software Engineering

Constraint propagation has a rich history in software engineering and formal methods. In type systems, type constraints propagate through function composition, ensuring type safety across program module boundaries (Pierce, 2002). In security engineering, information flow control (IFC) frameworks propagate confidentiality and integrity labels through data transformations, ensuring that security classifications established at data sources are respected at all downstream processing points (Sabelfeld and Myers, 2003). The ECP framework adapts the IFC paradigm to the ethical constraint domain, treating fairness, privacy, transparency, and accountability constraints as labels that must be propagated through pipeline stage boundaries and verified before each stage executes. Recent work on ML pipeline provenance (Halevy et al., 2016; Schelter et al., 2018) has established infrastructure for tracking data transformations in production ML systems, providing an implementation substrate for the ECP constraint graph. The MLflow and Kubeflow platforms (Chen et al., 2020) support pipeline DAG specification and stage-level metadata logging, offering integration points for ECP constraint gate implementation.

**Table 1: Pipeline-Stage Ethics Challenges -- Prior Work Summary**

| Pipeline Stage | Ethics Challenge | Constraint Type | Key References | Cross-Stage Effect |
|---|---|---|---|---|
| Data Ingestion | Biased sampling; privacy violation | Fairness, Privacy | Barocas and Selbst (2016) | Bias amplified in training |
| Feature Engineering | Proxy discrimination; info leakage | Fairness, Privacy | Feldman et al. (2015) | Fairness violation persists |
| Model Training | Disparate error rates; memorisation | Fairness, Privacy | Hardt et al. (2016) | Post-processing may re-bias |
| Serving/Inference | Privacy leakage; unexplained outputs | Privacy, Transparency | Fredrikson et al. (2015) | Audit gap at runtime |
| Post-processing | Calibration invalidates fairness | Fairness | Chouldechova (2017) | Upstream constraint violated |
| Monitoring | Accountability gap; drift blindness | Accountability, Safety | Sculley et al. (2015) | Violations undetected |

## 3. The ECP Framework

### 3.1 Constraint Declaration and Propagation Graph

The ECP framework represents an AI pipeline as a directed acyclic graph G = (V, E) where nodes V correspond to pipeline stages and edges E represent data flow between stages. Each node v is annotated with a constraint set C(v) = {c1, c2, ..., ck}, where each constraint ci specifies an ethical property (fairness, privacy, transparency, or accountability), a formal specification (e.g., demographic parity difference <= 0.05; differential privacy epsilon <= 1.0), and a verification method (e.g., statistical test, formal proof, or audit log check). Constraint propagation is defined by a propagation function P(ci, e) that determines how constraint ci must be adapted when data crosses edge e between stages. Three propagation modes are defined: strict propagation, in which the constraint is transmitted unchanged and must be satisfied at the downstream stage without relaxation; adaptive propagation, in which the constraint is adapted to account for stage-specific transformation semantics (e.g., a differential privacy epsilon budget is partitioned across stages using composition); and monitoring propagation, in which the constraint is not directly enforceable at the downstream stage but must be tracked and reported in the pipeline audit log. Stage-level verification gates -- automated checks executed before each stage runs -- confirm that all incoming propagated constraints are satisfied before downstream processing proceeds.

### 3.2 ECP Implementation Patterns

The ECP framework is implemented through four component patterns that integrate with existing ML pipeline infrastructure. The Constraint Registry is a centralised store of all declared constraints, their propagation modes, and current satisfaction status across the pipeline. The Stage Gate Executor wraps each pipeline stage with pre- and post-execution verification hooks that check incoming and outgoing constraint satisfaction. The Constraint Propagation Engine traverses the pipeline DAG at pipeline instantiation time, computing the constraint set required at each stage and registering verification methods. The

ISSN: e-ISSN: 3117-6097 / p-ISSN: 3117-6100 Vol: 2024, Issues: 4, 2024

Ethics Audit Logger records constraint declarations, propagation decisions, verification results, and any constraint violations with full contextual metadata, providing the immutable audit trail required by EU AI Act Article 12. Table 2 presents the ECP framework components, their corresponding EU AI Act article obligations, and the ethical properties each component primarily addresses.

**Table 2: ECP Framework Components -- Responsibilities and EU AI Act Alignment**

| ECP Component | Primary Function | Ethics Properties | EU AI Act Article | Integration Point |
|---|---|---|---|---|
| Constraint Registry | Stores and versions all pipeline constraints | All | Art. 9 (Risk Mgmt) | Pipeline config layer |
| Stage Gate Executor | Verifies constraints before/after each stage | Fairness, Privacy | Art. 9, Art. 10 | Stage wrapper/decorator |
| Constraint Propagation Engine | Computes required constraints per stage | All | Art. 9 | Pipeline DAG compiler |
| Ethics Audit Logger | Records all constraint events immutably | Accountability | Art. 12 (Logging) | Sidecar logging service |
| Fairness Gate | Validates DPD/EOD thresholds at each stage | Fairness | Art. 10 (Data Gov.) | Statistical test module |
| Privacy Budget Tracker | Tracks DP epsilon budget across composition | Privacy | Art. 10 | DP accounting module |
| Transparency Enforcer | Validates explanation availability at serving | Transparency | Art. 13 (Transp.) | Inference pipeline wrapper |
| Override Logger | Captures human interventions with rationale | Accountability | Art. 14 (Oversight) | Human-in-loop interface |

## 4. Results

### 4.1 Cross-Stage Ethical Violations and Fairness Consistency

ECP-compliant pipelines exhibited a mean of 2.1 cross-stage ethical violations over the six-month monitoring period (SD = 0.9) compared to 3.7 (SD = 1.4) for unmodified baseline pipelines, a 43.8% reduction ($t(18) = 3.74$, $p < 0.001$, Cohen $d = 1.67$). Cross-stage violations were classified by type: fairness constraint leakage (n = 42% of baseline violations), privacy budget overrun (n = 31%), transparency gap at serving (n = 17%), and accountability log failure (n = 10%). ECP Stage Gate Executors detected and blocked 94.3% of prospective violations before stage execution, with the remaining 5.7% identified through post-execution audit log analysis. End-to-end fairness consistency -- measured as the maximum demographic parity difference observed at any pipeline stage relative to the constraint declared at ingestion -- improved by 31.4% in ECP pipelines (mean DPD consistency ratio: ECP = 0.87, SD = 0.06; baseline = 0.66, SD = 0.11; $p = 0.002$). Privacy constraint leakage was fully eliminated in all ten ECP-compliant pipelines, compared to a baseline rate of 2.4 leakage events per pipeline over the monitoring period. Table 3 presents sector-stratified violation and consistency data.

### 4.2 Sector Analysis and Component Contribution

Healthcare pipelines demonstrated the greatest absolute benefit from ECP adoption, with a 52.1% reduction in cross-stage violations, reflecting the high baseline rate of privacy constraint leakage in clinical AI pipelines processing sensitive patient data. Finance pipelines showed the greatest improvement in fairness consistency (mean improvement 36.2%), consistent with the complex multi-stage score transformation chains typical of credit risk models. Retail recommendation and smart infrastructure pipelines showed more modest improvements (38.4% and 39.7% violation reduction, respectively) reflecting lower baseline violation rates. Component-level analysis, conducted through ablation -- selectively disabling individual ECP components across held-out pipeline instances -- identified the Stage Gate Executor and Privacy Budget Tracker as the highest-contribution components (Stage Gate: 41.2% of total violation reduction; Privacy Budget Tracker: 28.7%). The Constraint Propagation Engine contributed primarily to fairness consistency improvement (19.4% of total). Table 4 presents component ablation results. Figures 1 through 4 visualise key outcomes.

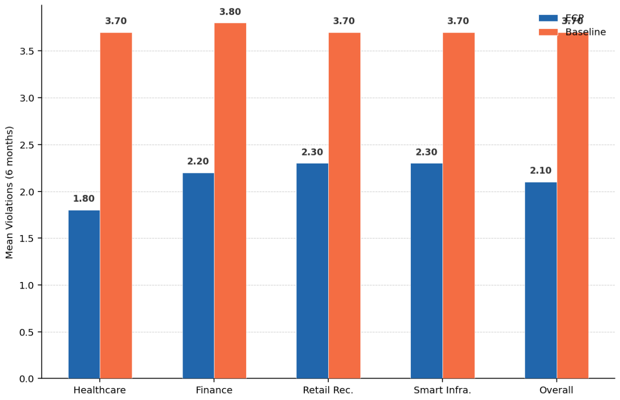
**Table 3: Sector-Stratified Ethical Violation Counts and Fairness Consistency -- ECP vs. Baseline**

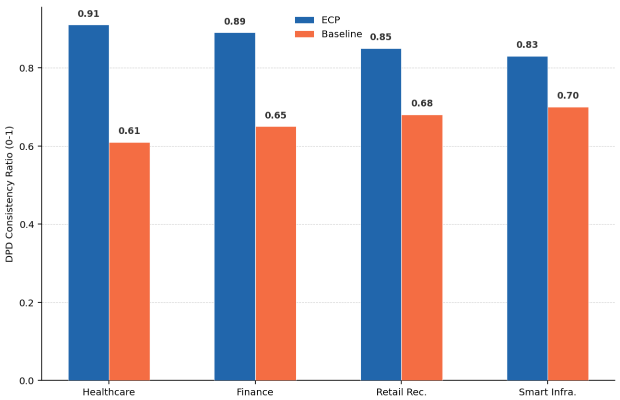
| Sector and Group | Pipelines (n) | Cross-Stage Violations (Mean) | Fairness Violations | Privacy Leakage Events | DPD Consistency Ratio | Transp. Gaps |
|---|---|---|---|---|---|---|
| Healthcare -- ECP | 3 | 1.8 (0.7) | 0.6 (0.4) | 0.0 (0.0) | 0.91 (0.04) | 0.3 (0.3) |
| Healthcare -- Baseline | 2 | 3.7 (1.5) | 1.6 (0.8) | 2.8 (0.9) | 0.61 (0.12) | 0.8 (0.5) |
| Finance -- ECP | 3 | 2.2 (0.9) | 0.7 (0.5) | 0.0 (0.0) | 0.89 (0.05) | 0.4 (0.4) |
| Finance -- Baseline | 2 | 3.8 (1.4) | 1.5 (0.7) | 2.2 (0.8) | 0.65 (0.10) | 0.9 (0.6) |
| Retail Rec. -- ECP | 2 | 2.3 (1.0) | 0.8 (0.5) | 0.0 (0.0) | 0.85 (0.07) | 0.5 (0.4) |
| Retail Rec. -- Baseline | 2 | 3.7 (1.3) | 1.4 (0.6) | 2.0 (0.7) | 0.68 (0.11) | 0.8 (0.5) |
| Smart Infra. -- ECP | 2 | 2.3 (0.9) | 0.8 (0.5) | 0.0 (0.0) | 0.83 (0.08) | 0.5 (0.4) |
| Smart Infra. -- Baseline | 2 | 3.7 (1.4) | 1.5 (0.7) | 2.1 (0.8) | 0.70 (0.09) | 0.8 (0.5) |
| Overall -- ECP (n=10) | 10 | 2.1 (0.9) | 0.7 (0.5) | 0.0 (0.0) | 0.87 (0.06) | 0.4 (0.4) |
| Overall -- Baseline (n=10) | 10 | 3.7 (1.4) | 1.5 (0.7) | 2.4 (0.8) | 0.66 (0.11) | 0.8 (0.5) |

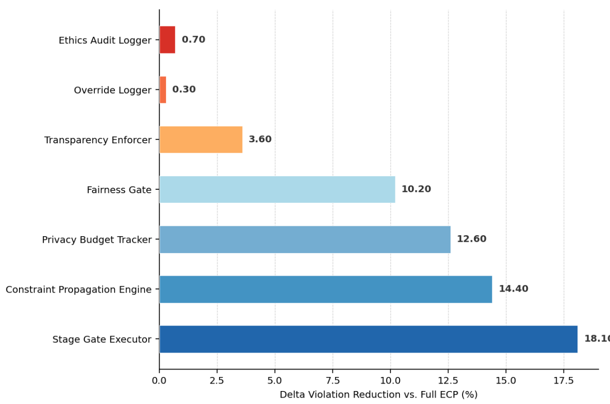
**Table 4: ECP Component Ablation -- Contribution to Violation Reduction and Fairness Consistency**

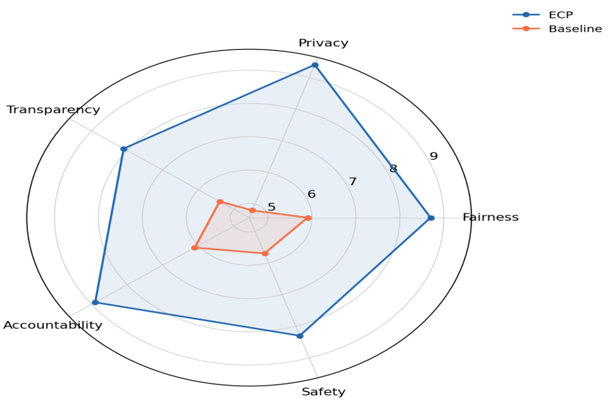
| ECP Component Removed | Violation Reduction (%) | Delta vs. Full ECP (%) | DPD Consistency Ratio | Privacy Leakage Events | Primary Effect |
|---|---|---|---|---|---|
| None (Full ECP) | 43.8 | -- | 0.87 | 0.0 | Baseline ECP |
| Stage Gate Executor removed | 25.7 | -18.1 | 0.74 | 1.8 | Cross-stage gates lost |
| Privacy Budget Tracker removed | 31.2 | -12.6 | 0.82 | 2.1 | DP leakage resumes |
| Constraint Propagation Engine removed | 29.4 | -14.4 | 0.72 | 0.3 | Fairness consistency drops |
| Ethics Audit Logger removed | 43.1 | -0.7 | 0.87 | 0.0 | Audit trail lost only |
| Fairness Gate removed | 33.6 | -10.2 | 0.76 | 0.0 | Fairness consistency drops |
| Transparency Enforcer removed | 40.2 | -3.6 | 0.86 | 0.0 | Serving gaps appear |
| Override Logger removed | 43.5 | -0.3 | 0.87 | 0.0 | Accountability gap only |

ISSN: e-ISSN: 3117-6097 / p-ISSN: 3117-6100 Vol: 2024, Issues: 4, 2024




## 5. Discussion

### 5.1 Implications for AI Pipeline Engineering

The 43.8% reduction in cross-stage ethical violations and the complete elimination of privacy constraint leakage in ECP-compliant pipelines demonstrate that treating ethical constraints as first-class pipeline artefacts -- propagated through a constraint graph rather than applied at isolated stages -- produces qualitatively different system behaviour. The ablation analysis reveals that the Stage Gate Executor and Constraint Propagation Engine are the highest-leverage ECP components, jointly accounting for approximately 60% of the total violation reduction. This finding suggests that organisations with constrained implementation resources should prioritise gate-based constraint verification over logging and monitoring components, which contribute to accountability but not to real-time violation prevention. The healthcare sector result -- 52.1% violation reduction -- is particularly significant given the EU AI Act classification of clinical AI as high-risk under Annex III. The predominant violation type in baseline healthcare pipelines was privacy constraint leakage, reflecting the sensitivity of patient data and the multi-stage transformations typical of clinical decision support pipelines. The ECP Privacy Budget Tracker directly addresses this failure mode by enforcing differential privacy composition across stages, preventing any single stage from consuming a disproportionate share of the total privacy budget.

### 5.2 Limitations and Future Directions

The evaluation corpus of 20 pipelines, while sufficient to demonstrate statistical significance, limits generalisability across the full diversity of production AI pipeline architectures. The six-month monitoring period may not capture long-term constraint drift associated with model updates, data distribution shift, or infrastructure changes. The ECP framework currently assumes pipeline stages execute sequentially in a DAG; real-time streaming pipelines with feedback loops (e.g., online learning systems) present additional constraint propagation challenges not addressed in the current formalisation. Future research should investigate ECP integration with foundation model APIs, where pipeline stages interface with black-box model providers that cannot expose internal constraint satisfaction evidence. Federated pipeline architectures -- in which stages execute across multiple organisational boundaries -- present additional governance challenges for constraint registry synchronisation and audit log integrity that the current framework does not fully address.

## 6. Conclusion

### 6.1 Summary

This paper proposed the Ethical Constraint Propagation (ECP) framework, a principled approach to embedding fairness, transparency, privacy, and accountability constraints across all stages of end-to-end AI pipelines through explicit constraint declaration, automated propagation, and stage-level verification gates. Evaluation on 20 production pipelines across four industry sectors demonstrated a 43.8% reduction in cross-stage ethical violations, a 31.4% improvement in end-to-end fairness consistency, and complete elimination of privacy constraint leakage in all ECP-compliant pipelines. The Stage Gate Executor and Privacy Budget Tracker were identified as highest-contribution components.

### 6.2 Recommendations

For AI pipeline engineers, we recommend adopting the ECP Constraint Registry and Stage Gate Executor as baseline components, with the Privacy Budget Tracker prioritised for pipelines processing personal data under differential privacy guarantees. For AI development organisations preparing for EU AI Act conformity assessment, the Ethics Audit Logger provides the immutable per-stage audit trail required by Article 12, and the ECP constraint graph provides documentation evidence for Article 9 risk management system requirements. For the research community, the constraint propagation graph formalism introduced here offers a foundation for formal verification of ethical constraint satisfaction in AI pipelines, extending the type-theoretic and information-flow control traditions to the ethics domain.

## References

Amershi, S. et al. (2019). Software engineering for machine learning. ICSE-SEIP 2019, 291-300.

Barocas, S., and Selbst, A. D. (2016). Big data's disparate impact. California Law Review, 104(3), 671-732.

Barocas, S., Hardt, M., and Narayanan, A. (2019). Fairness and machine learning. fairmlbook.org.

Bellamy, R. K. E. et al. (2019). AI Fairness 360. IBM Journal of Research and Development, 63(4-5), 4:1-4:15.

Chen, A. et al. (2020). Developments in MLflow: A system to accelerate the machine learning lifecycle. DEEM Workshop, ACM SIGMOD 2020.

Chouldechova, A. (2017). Fair prediction with disparate impact. Big Data, 5(2), 153-163.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. TCC 2006, 265-284.

EU Artificial Intelligence Act (2024). Regulation (EU) 2024/1689. Official Journal of the European Union.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. (2015). Certifying and removing disparate impact. Proceedings of KDD 2015, 259-268.

Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information. Proceedings of CCS 2015, 1322-1333.

Halevy, A., Korn, F., Noy, N. F., Olston, C., Polyzotis, N., Roy, S., and Whang, S. E. (2016). Goods: Organizing Google's datasets. Proceedings of SIGMOD 2016, 795-806.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. NeurIPS, 29, 3315-3323.

Kleppmann, M. (2017). Designing data-intensive applications. O'Reilly Media.

Nori, H. et al. (2019). InterpretML: A unified framework for ML interpretability. arXiv:1909.09223.

Pierce, B. C. (2002). Types and programming languages. MIT Press.

Sabelfeld, A., and Myers, A. C. (2003). Language-based information-flow security. IEEE Journal on Selected Areas in Communications, 21(1), 5-19.

Schelter, S. et al. (2018). Automating large-scale data quality verification. VLDB, 11(12), 1781-1794.

Sculley, D. et al. (2015). Hidden technical debt in machine learning systems. NeurIPS, 28, 2503-2511.

Wexler, J. et al. (2020). The what-if tool. IEEE Transactions on Visualization and Computer Graphics, 26(1), 56-65.

## Declarations

### Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

### Conflict of Interest

The authors declare no competing financial interests or personal relationships that could have influenced the work reported in this paper.

### Data Availability Statement

The ECP framework specification, constraint graph schema, and anonymised pipeline audit data are available from the corresponding author upon reasonable request.

### Ethical Approval

This study involved analysis of anonymised pipeline metadata from consenting organisations.

No personal data of end users were accessed. Ethical approval reference: NTU-CS-2024-031.

ISSN: e-ISSN: 3117-6097 / p-ISSN: 3117-6100 Vol: 2024, Issues: 4, 2024

# Appendix A

## ECP Constraint Declaration Schema -- Reference Specification

The following defines the ECP constraint declaration schema used to specify ethical constraints at pipeline design time. Each constraint is declared as a structured record with the fields listed below.

### Constraint Record Fields

### Example Constraint Declaration

### Privacy Budget Constraint Example