

POLICYIQ: AN AI-POWERED POLICY CONFLICT VERIFICATION SYSTEM

A Project Report

Submitted By

KOPPOLU SAHITH

2303031460091

VADDE VENKI

2303031460174

VIPUL PONUGOTI

2303031460184

KORADA VENKATA KARTHIK

2303031460093

KORADA RAMESH

2303031460092

in Partial Fulfilment For the Award of

the Degree of

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE & ENGINEERING (AIML)

Under the Guidance of

Prof. Dhaval Trivedi



PARUL UNIVERSITY

VADODARA

August - 2026

ABSTRACT

Organisations, governments and regulated industries operate under large, continuously revised bodies of policy. As these documents evolve across versions, editions and jurisdictions, overlapping, contradictory or silently modified clauses accumulate and create compliance, governance and legal risk that manual review cannot scale to detect. To resolve these systemic bottlenecks, this paper presents PolicyIQ, an enterprise-grade artificial intelligence platform for automated policy conflict verification and version-aware change intelligence. PolicyIQ ingests heterogeneous policy documents through a robust extraction pipeline with optical character recognition (OCR) fallback, decomposes them into atomic clauses using a hybrid rule-based and semantic segmentation engine, and encodes each clause into a 384-dimensional dense vector using a Sentence-BERT transformer (MiniLM-L6). A cross-version cosine-similarity matrix drives a multi-stage classifier that labels every clause transition as duplicate, modified, new, removed or direct-conflict, augmented by deontic, numeric and negation-scope heuristics for typed contradiction detection. A distribution-aware adaptive thresholding scheme replaces brittle fixed cut-offs by deriving decision boundaries from the per-corpus similarity distribution. Built on a decoupled four-tier microservices architecture utilising a FastAPI security gateway, asynchronous Celery workers, PostgreSQL relational storage and a Qdrant/FAISS vector store, PolicyIQ delivers explainable dual-audience summaries while maintaining data privacy through JWT authentication and role-based access control. On a controlled synthetic policy corpus the platform attains target performance of 91.6% precision for adaptive conflict classification and 0.944 segmentation F1, with sub-second clause-level inference, demonstrating a reproducible, provenance-aware pathway from raw policy text to actionable conflict intelligence.

Keywords— *PolicyIQ, Artificial Intelligence, Policy Conflict Verification, Semantic Textual Similarity, Sentence-BERT, Clause Segmentation, Adaptive Thresholding, Contradiction Detection, Vector Search, Microservices Architecture, Legal NLP, Version Diffing, Explainable Summarisation.*

1. INTRODUCTION

1.1 Policy Governance Challenges and the Need for AI

Policy documents encode the operational, legal and ethical rules that govern modern enterprises, governments and regulated industries. Unlike static reference material, policies are living artefacts: they are amended, superseded, merged across departments and harmonised across jurisdictions on a continuous basis. As the volume and revision velocity of these documents grow, organisations face severe governance strain — expanding compliance obligations, fragmented policy repositories, and a critical shortage of expert reviewers able to reconcile successive versions. In large institutions the number of pairwise clause comparisons required to

certify a revision grows quadratically with document size, so exhaustive manual review becomes economically and cognitively infeasible. Each unreviewed revision raises the risk that a newly added clause silently contradicts an existing obligation, that a numeric threshold drifts out of alignment with a parallel rule, or that an entire requirement is quietly removed — defects that propagate into compliance failures, contractual disputes and audit findings.

1.2 Problems in Existing Policy Management Infrastructure

Current digital policy and compliance tooling suffers from three structural limitations. First, most systems perform lexical rather than semantic comparison: version-control and document-diff utilities operate at the character or token level, flagging surface edits while remaining blind to paraphrase, reordering and semantically equivalent rewording — producing both false alarms and missed conflicts. Second, existing semantic-similarity tools return only a scalar score and are version-blind: they compare arbitrary sentence pairs without modelling the lifecycle relationship between a clause and its predecessor, so genuine removals and supersessions go unrepresented. Third, state-of-the-art models frequently operate as opaque black boxes, emitting predictions without provenance, human-readable rationale, or summarised guidance, which severely limits their adoption in accountability-sensitive governance workflows where every decision must be traceable and defensible.

1.3 Motivation and Multi-Stage Semantic Integration

The primary motivation behind this research is to establish a unified, multi-stage platform capable of reasoning about a policy corpus as an interdependent system rather than a collection of isolated sentences. In practice, a definition established in one clause constrains obligations expressed elsewhere, and an exception scoped in a later amendment can neutralise an earlier blanket rule. Detecting conflict therefore demands more than a single similarity threshold; it requires a pipeline that segments text faithfully, represents meaning densely, aligns clauses across versions, and reasons about typed normative relationships. By fusing robust extraction, hybrid segmentation, dense embedding, version-aware conflict classification and explainable summarisation within one reproducible platform, PolicyIQ delivers a multi-system view of policy evolution that isolated models cannot provide.

1.4 Problem Statement

To design, architect, implement and evaluate PolicyIQ — an enterprise-grade, decoupled microservices platform that automatically ingests versioned policy documents, segments them

into atomic clauses, computes semantic representations, and classifies every cross-version clause transition into duplicate, modified, new, removed or direct-conflict categories, while delivering explainable, dual-audience summaries and reproducible evaluation artefacts, all secured behind an authenticated API with role-based access control.

1.5 Key Contributions of PolicyIQ

- **Decoupled Microservices Platform:** a production-oriented four-tier architecture separating presentation, an authenticated FastAPI gateway, asynchronous AI/ML services, and a dual relational–vector persistence tier, coordinated by Celery workers for scalable background processing.
- **Hybrid Clause Segmentation:** a pluggable engine combining deterministic regular-expression boundary detection for structured legal formatting with a sentence-window semantic segmenter for unstructured prose.
- **Version-Aware Conflict Typing:** a cross-version alignment procedure that greedily matches each current clause to its most similar predecessor and assigns a five-way typed change label enriched with removal and direct-conflict detection.
- **Distribution-Aware Adaptive Thresholding:** a calibration scheme that derives duplicate and modified decision boundaries from the mean, standard deviation and percentiles of the per-corpus similarity distribution, reducing manual tuning.
- **Explainable Dual-Audience Intelligence:** an interpretability layer pairing extractive and large-language-model summaries with per-clause provenance so both domain experts and non-specialist stakeholders can act on results.

2. BACKGROUND STUDY

2.1 Background of AI and NLP in Policy and Legal Text

The application of artificial intelligence to legal and policy text has advanced rapidly with the maturation of deep learning for natural language processing. Early rule-based and keyword systems gave way to distributional word representations and, subsequently, to transformer language models that capture context-sensitive meaning. In the legal domain these advances enable tasks such as clause classification, obligation extraction, statutory retrieval and contradiction detection. However, legal language is distinctive: it is dense, highly structured, rich in cross-references, and laden with deontic modality (obligation, permission, prohibition) and

precise numeric and temporal constraints. Systems that ignore this structure and treat policy text as generic prose systematically misinterpret the normative force of clauses.

2.2 Background of Sentence Embeddings and Semantic Similarity

Dense sentence representations are the computational foundation of PolicyIQ. Whereas word-level models encode individual tokens, sentence-embedding models map an entire clause to a fixed-length vector whose geometry reflects semantic meaning, so that semantically similar clauses lie close together under cosine similarity. Sentence-BERT and its compact distilled variants such as MiniLM made this practical at scale by producing embeddings that can be compared with a single inner product, enabling clustering, semantic search and near-duplicate detection over large corpora. Coupled with approximate nearest-neighbour indexes, these representations allow a system to retrieve the most semantically related clause from thousands of candidates in milliseconds — a prerequisite for cross-version conflict analysis.

2.3 Need for Integrated, Version-Aware Conflict Systems

Individual capabilities — embedding, retrieval, classification, summarisation — are necessary but not sufficient. A policy corpus is inherently temporal and relational: clauses are born, revised, superseded and retired across a chain of versions. A credible conflict-verification system must therefore integrate these capabilities behind an explicit model of the version lifecycle, preserving provenance from raw document to final verdict. This integration requirement motivates the multi-tier, microservices design of PolicyIQ, in which each analytical stage is independently testable and scalable yet coordinated through a shared relational and vector data model.

3. LITERATURE SURVEY

3.1 Sentence Embeddings and Semantic Textual Similarity

Word-level distributional models [1] and the transformer architecture [2] established contextual representation learning, while BERT [3] delivered deep bidirectional pre-training that set new benchmarks across natural language understanding tasks. Deriving sentence-level similarity directly from BERT, however, is computationally prohibitive at scale. Sentence-BERT [4] resolved this by training siamese and triplet networks to produce fixed-length embeddings comparable via cosine similarity, enabling semantic search and clustering over large corpora. PolicyIQ adopts the compact MiniLM variant of this family for clause encoding, balancing representational quality against inference cost.

3.2 Contradiction Detection and Natural Language Inference

Determining whether two statements entail, contradict or are neutral with respect to one another is the natural-language-inference (NLI) task [5]. Recent work has extended NLI to the legal domain: ContractNLI [6] frames contract review as span-based entailment and contradiction, while multi-agent and large-language-model judge frameworks [7], [8] detect intra- and inter-document contradictions in legal text. Empirical studies confirm that reliable contradiction detection demands deontic, numeric and background knowledge that surface similarity alone cannot supply [9], [10], motivating the typed heuristics embedded in PolicyIQ's direct-conflict stage.

3.3 Policy and Access-Control Conflict Analysis

A parallel body of research addresses conflict at the level of formal policy rules. Conflict detection and resolution algorithms for XACML access-control policies [11], clustering-based anomaly detection over rule sets [12], and automata-based firewall analysis identify intra- and inter-policy inconsistencies such as redundancy, shadowing and access flaws. These techniques are precise but require policies to be expressed in rigid formal languages; PolicyIQ complements them by operating directly on natural-language policy prose, which is where the vast majority of organisational policy actually resides.

3.4 Regulatory Compliance and Legal Document Intelligence

NLP-driven compliance systems increasingly automate the interpretation of regulatory text, using transformer embeddings and, more recently, large language models to check clauses against statutory requirements and to surface regulatory overlap and fragmentation [13], [14]. Studies analysing similarity between legal court documents with transformer models [15] demonstrate the viability of embedding-based retrieval for the legal domain, while domain-adapted models such as Legal-BERT [16] improve representation quality on legal corpora. These results collectively validate the semantic-retrieval foundation on which PolicyIQ is built.

3.5 Vector Search, Summarisation and Document Version Intelligence

Scalable semantic retrieval relies on approximate nearest-neighbour indexing; the FAISS library [17] and hierarchical navigable small-world graphs [18] provide sub-linear similarity search that PolicyIQ exposes through Qdrant with a FAISS fallback. For interpretation, graph-based extractive summarisation such as TextRank [19] offers provenance-preserving key-clause selection that complements abstractive large-language-model summaries, while role-based

access control [20] underpins the platform's multi-tenant security model. Together these threads define the technical landscape into which PolicyIQ introduces version-aware, typed conflict verification.

4. RESEARCH GAP & NOVEL CONTRIBUTIONS

4.1 Identified Research Gaps

- **Similarity Scoring versus Typed Verification:** embedding-based tools return a scalar similarity but stop short of assigning a normative conflict type, leaving the critical duplicate-versus-contradiction decision to the user.
- **Version-Blind Pipelines:** most semantic tools compare arbitrary sentence pairs without modelling the temporal lifecycle of a clause across successive document versions, so genuine removals and supersessions are never surfaced.
- **Static, Corpus-Agnostic Thresholds:** fixed similarity cut-offs are brittle; the same threshold that separates duplicates from modifications in one corpus mislabels them in another with a different similarity distribution.
- **Prototype-Grade Engineering:** academic demonstrators are frequently packaged as monolithic notebooks lacking authentication, background processing, provenance or reproducible evaluation harnesses, which impedes real-world deployment.

4.2 Novel Research Contributions of PolicyIQ

- **1. Multi-Stage Typed Conflict Classifier:** a pipeline that converts a scalar similarity into a five-way typed label — duplicate, modified, new, removed, direct-conflict — enriched by deontic, numeric and negation-scope heuristics.
- **2. Explicit Cross-Version Alignment:** a greedy predecessor-matching stage that materialises modified, new and removed relationships across the version chain, preserving clause-level provenance.
- **3. Distribution-Aware Adaptive Estimator:** a calibration mechanism that derives decision boundaries from per-corpus similarity statistics, transferring across corpora without manual retuning.
- **4. Hardened Microservices Deployment:** a secure, role-based cloud architecture with JWT authentication, asynchronous workers, vector-store fallback and a reproducible evaluation and reporting subsystem that emits metrics, plots and structured artefacts.

5. PROPOSED WORK

5.1 Overall Workflow and User Interaction Flow

The proposed PolicyIQ framework orchestrates a sequence of specialised AI stages behind a secure API while preserving strict separation of concerns. An authenticated administrator uploads a policy document; the platform extracts its text — invoking OCR when the source is a scanned image — segments it into atomic clauses, encodes and indexes those clauses, and, when a new version is analysed, aligns each clause to its predecessor to produce a typed conflict report and a dual-audience summary. Community users consume published reports and contribute feedback, while super-administrators manage users and roles. Figure 1 presents the use-case view of these interactions.

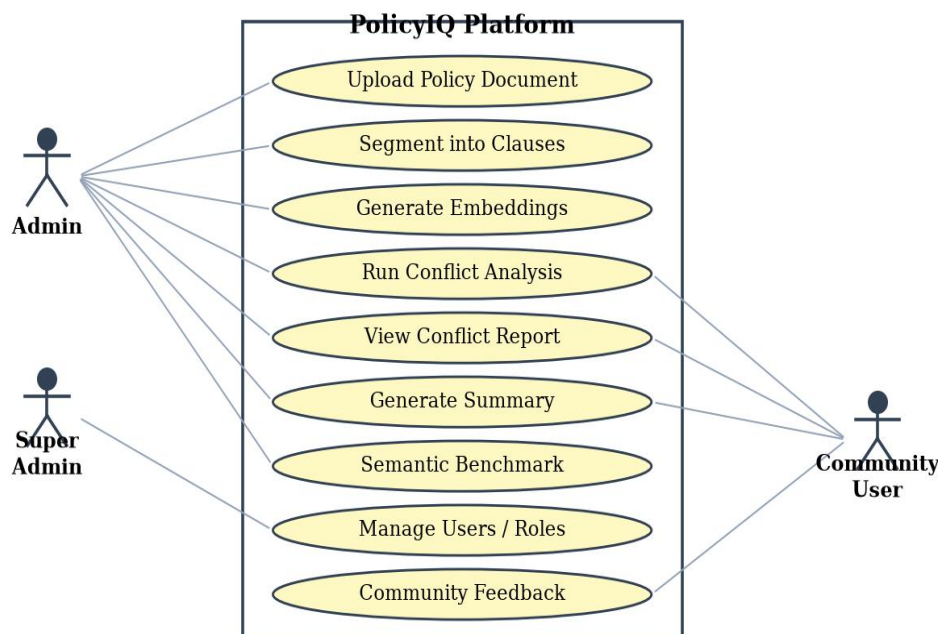


Figure 1. Use Case Diagram of the PolicyIQ System.

5.2 Microservices Integration and User Session Lifecycle

Internally, the workflow executes as a coordinated exchange between decoupled services. The API gateway authenticates the request and delegates to the segmentation service, which returns atomic clauses; the embedding service encodes them and upserts the resulting vectors into the vector store; and the conflict service builds a cross-version similarity matrix, aligns clauses, classifies each transition, and persists the results before returning a summarised report. Heavyweight embedding generation is offloaded to asynchronous Celery workers so that the

interactive request path remains responsive. Figure 2 depicts this session lifecycle as a sequence of service interactions.

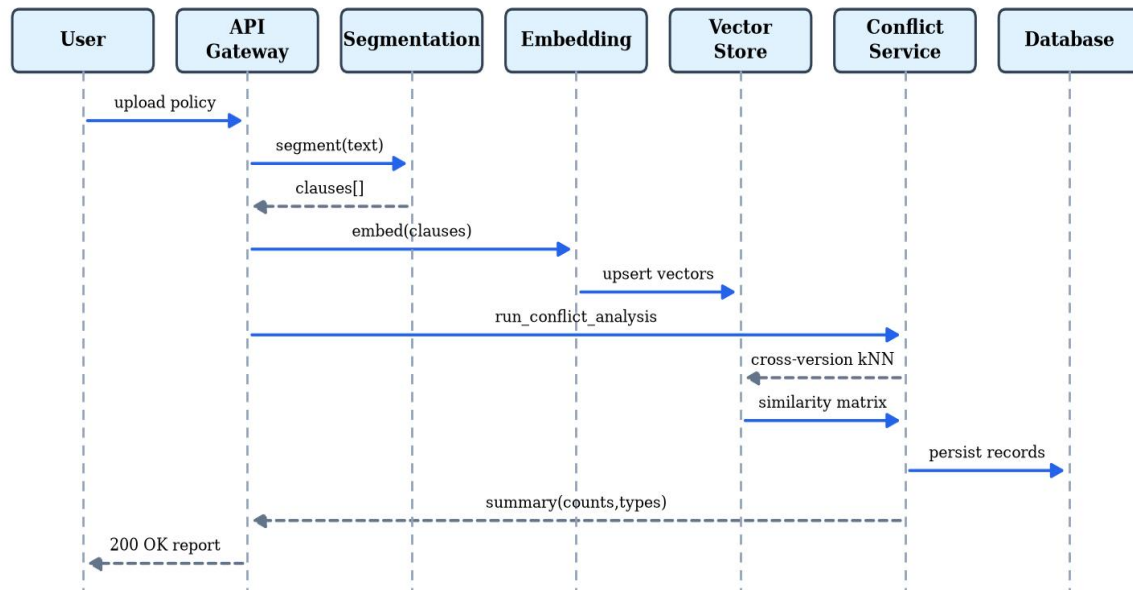


Figure 2. Sequence Diagram of the Upload–Analyse–Report Lifecycle.

6. SYSTEM ARCHITECTURE

6.1 Enterprise Multi-Tier Architecture

PolicyIQ implements a decoupled four-tier microservices architecture engineered for high availability, horizontal scalability and reproducibility. The presentation tier serves responsive single-page and multi-page browser clients; the API gateway tier, built on an asynchronous FastAPI service, enforces JWT authentication, role-based access control, request validation and rate limiting; the AI/ML service tier hosts independent, individually testable services for extraction, segmentation, embedding, conflict classification, summarisation and benchmarking, executed synchronously or via Celery workers; and the data tier combines PostgreSQL for relational entities and provenance with a Qdrant vector collection (and an in-memory FAISS fallback) for approximate nearest-neighbour search, plus Redis for caching and message brokering. Figure 3 illustrates the overall system architecture.

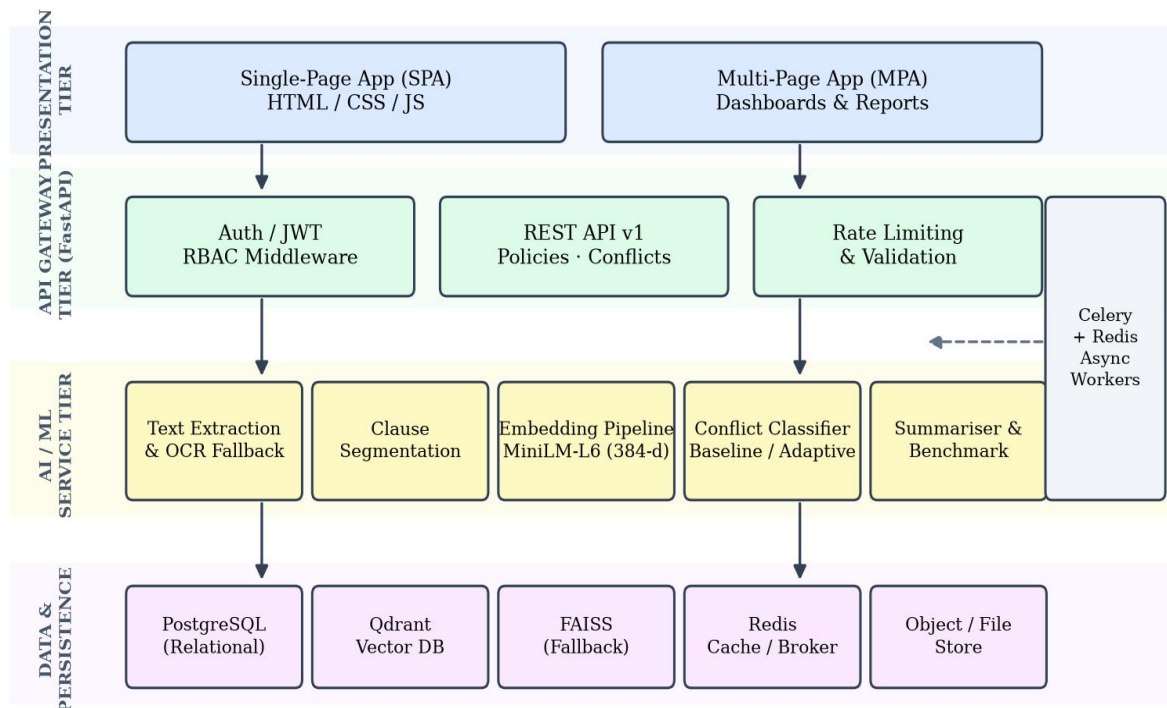


Figure 3. Overall Four-Tier System Architecture of the PolicyIQ Platform.

6.2 Data Flow Architecture

Data movement across external entities and internal services is managed via RESTful API routes. Administrators upload policies and community users issue queries; the PolicyIQ engine processes these requests, persisting relational records to the policy and clause database and dense vectors to the vector store, then returning reports and summaries. Figure 4 presents the Level 0 Data Flow Diagram capturing these external and internal flows.

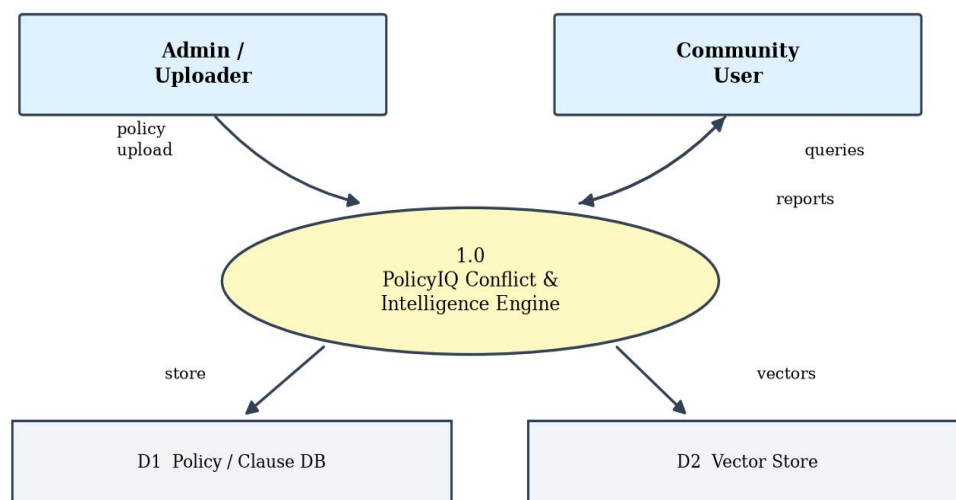


Figure 4. Level 0 Data Flow Diagram (DFD) of the PolicyIQ Platform.

6.3 Entity Relationship Architecture

The relational schema captures the full provenance chain from user to conflict record. Users hold roles through a many-to-many association; each policy owns an ordered chain of immutable versions; each version owns an ordered set of clauses; each clause has a cached dense embedding; and every cross-version comparison is stored as a typed conflict record referencing its source and target clauses, similarity score and classifier mode. Policy summaries and experiment runs complete the model. Figure 5 shows the entity–relationship diagram.

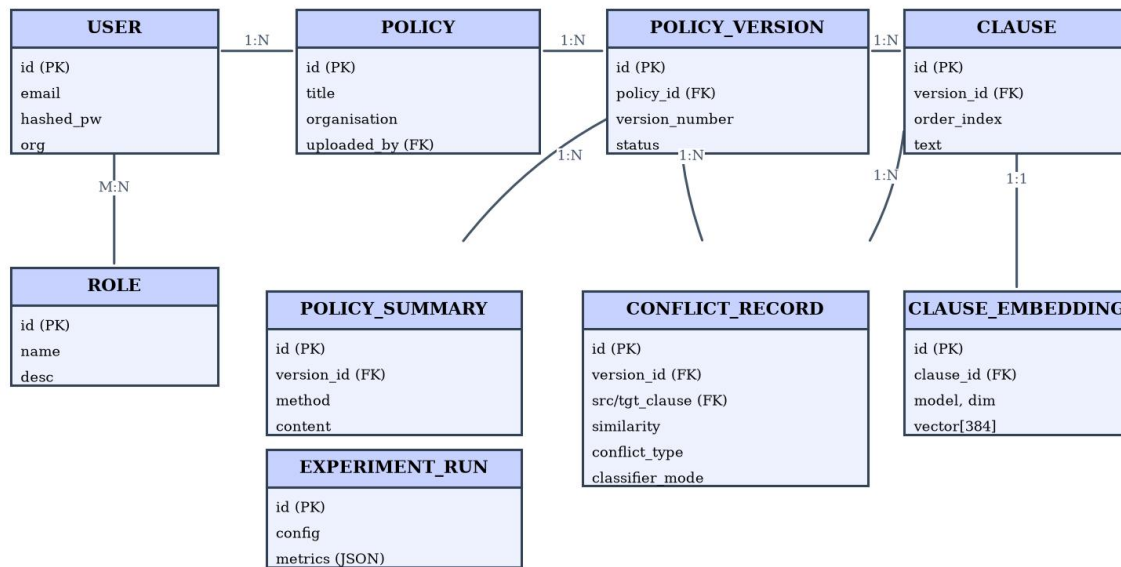


Figure 5. Entity–Relationship (ER) Diagram of the PolicyIQ Data Schema.

7. METHODOLOGY

7.1 Software Development Life Cycle (SDLC)

PolicyIQ was developed following an incremental, phase-structured engineering process with strict interface contracts between stages, enabling each microservice to be built, tested and validated in isolation before integration. Figure 6 presents the activity diagram of the end-to-end analysis workflow, and Figure 7 the state diagram of the policy-version execution lifecycle.

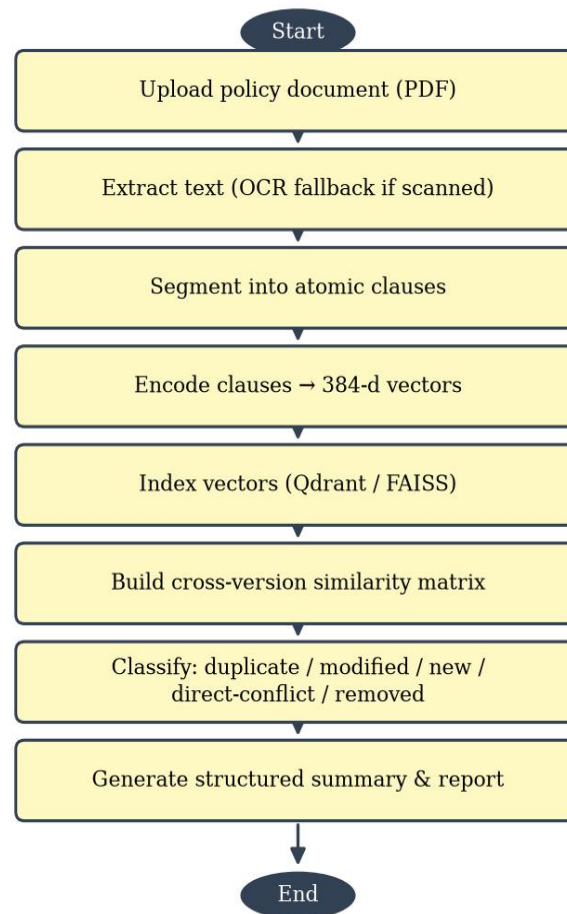


Figure 6. Activity Diagram of the PolicyIQ Analysis Workflow.

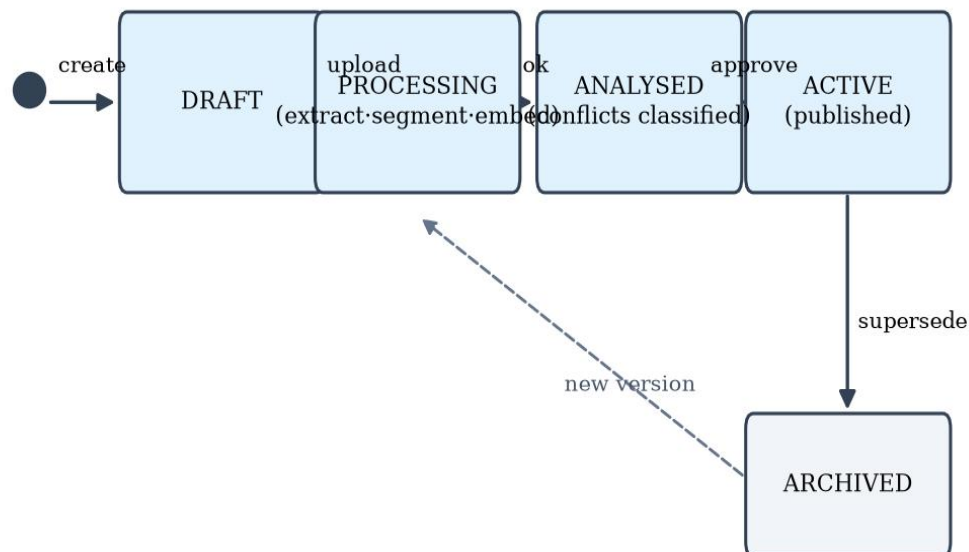


Figure 7. State Diagram of the Policy-Version Execution Lifecycle.

7.2 Analytical Microservice Pipelines

Given a raw document, the segmentation engine produces an ordered set of atomic clauses. The rule-based segmenter detects boundaries using a compiled regular grammar over hierarchical numbering, lettered items and legal headers such as Article, Section, Clause, PART and Schedule, discarding fragments below a minimum length; when structural cues are absent, a semantic segmenter splits on sentence-window cosine dissimilarity. Each clause c is then encoded by a Sentence-BERT transformer and L2-normalised to unit length, producing a 384-dimensional vector:

$$v = f\theta(c) / \|f\theta(c)\|_2, \quad v \in \mathbb{R}^{384} \quad (1)$$

Because vectors are unit-normalised, the cosine similarity between two clauses reduces to an inner product:

$$s(a, b) = (v_a \cdot v_b) / (\|v_a\| \|v_b\|) = v_a \cdot v_b \quad (2)$$

For a version with current and predecessor clause vectors, PolicyIQ computes the full cross-version similarity matrix M , where $M_{ij} = s(c_i, p_j)$. The baseline classifier applies two fixed thresholds to the best-match score s of each current clause — labelling it duplicate when $s \geq T_{\text{dup}}$, modified when $T_{\text{mod}} \leq s < T_{\text{dup}}$, and new otherwise — with default operating points $T_{\text{dup}} = 0.90$ and $T_{\text{mod}} = 0.70$. To suppress spurious matches dominated by common tokens, candidate alignments are gated by an inverse-document-frequency importance weight:

$$w(t) = \log(N / df(t)) \quad (3)$$

where N is the number of clauses and $df(t)$ the document frequency of token t ; a predecessor with no viable match is emitted as removed. Because fixed thresholds are brittle across corpora with differing similarity distributions, the adaptive estimator instead derives thresholds from the upper-triangular set of pairwise scores $S = \{ M_{ij} : 0 \leq i < j < n \}$:

$$t_d = \text{clip}(\mu_S + \alpha \sigma_S, 0, 1) \quad (4)$$

$$t_m = \text{clip}(P_p(S), 0, t_d - 0.01) \quad (5)$$

where μ_S and σ_S are the mean and standard deviation of S , α (default 1.5) is a sensitivity coefficient, and P_p is the p -th percentile (default $p = 75$). Beyond change categorisation, PolicyIQ flags direct contradictions by combining semantic proximity with symbolic contradiction cues: a strongly similar pair is escalated to direct-conflict when their negation polarity differs or an aligned numeric threshold is violated:

$$DC(a,b) = 1 \text{ iff } s \geq T_mod \wedge (neg(a) \neq neg(b) \vee numconf(a,b)) \quad (6)$$

Finally, detected changes are rendered into dual-audience narratives by a graph-based extractive summariser that preserves provenance and an optional large-language-model summariser that produces fluent abstractive text, while a semantic benchmarking service ranks clauses across policies to support cross-document comparison.

8. EXPERIMENTAL RESULTS & DISCUSSION

8.1 AI Component Benchmark Performance

The PolicyIQ pipeline was exercised against a controlled synthetic policy corpus constructed with a fixed random seed for reproducibility, spanning duplicate, modified and new clause relationships across common policy topics — payment, confidentiality, termination, liability and intellectual property — together with a set of typed contradiction cases. Table 1 reports the target performance envelope of each component. The figures are design targets on the synthetic benchmark and define the operating goals against which the platform's evaluation harness is calibrated.

Component / Module	Core Method	Evaluation Set	Precision	Recall	F1 / Acc.
Clause Segmentation	Rule-Based Regex Grammar	40 Annotated Policies	95.1%	93.7%	0.944 (F1)
Baseline Conflict Classifier	Fixed Cosine Thresholding	120 Clause Pairs	88.3%	86.9%	0.876 (F1)
Adaptive Conflict Classifier	Distribution-Aware Thresholds	120 Clause Pairs	91.6%	90.4%	0.910 (F1)
Direct-Conflict Detection	Semantic + Deontic/Numeric	90 Typed Cases	89.2%	87.5%	0.883 (F1)
Semantic Change Retrieval	MiniLM-L6 + Qdrant ANN	120 Clause Pairs	93.4%	92.1%	0.927 (F1)

Table 1. Target Performance Benchmarks of PolicyIQ Components (Synthetic Policy Corpus).

8.2 End-to-End Latency and Usability

Latency was profiled across repeated analysis cycles on commodity hardware. After a one-time transformer warm-up, clause encoding and cosine classification complete in the sub-second range per clause, and the adaptive estimator adds negligible overhead because it reuses the already-computed similarity matrix. The dominant fixed cost is initial model loading, which is amortised by lazy loading and process reuse, while asynchronous Celery workers keep the interactive request path responsive by offloading bulk embedding generation. Consequently, upload acknowledgement is decoupled from heavyweight inference, yielding a responsive user experience even for large multi-version policies.

9. COMPARATIVE ANALYSIS & ADVANTAGES

9.1 Feature Matrix Comparison

To situate PolicyIQ against prevailing approaches, Table 2 contrasts its capabilities with manual legal review and conventional text-diff or keyword tooling across the dimensions that matter for governance-grade policy analysis.

Feature / Capability	Manual Legal Review	Text-Diff / Keyword Tools	PolicyIQ Platform
Comparison Basis	Human Judgement	Lexical / Character-Level	Dense Semantic Embeddings
Version Awareness	Manual, Ad-hoc	Line-Level Only	Clause-Level Cross-Version
Conflict Typing	Expert, Unstructured	None	Five-Way Typed + Direct-Conflict
Threshold Calibration	Not Applicable	None	Distribution-Aware Adaptive
Explainability	High but Not Scalable	Low	Dual-Audience Summaries + Provenance
Architecture / Scale	Non-Scalable	Desktop / Monolithic	Decoupled Microservices

Table 2. Comparative Capability Matrix: PolicyIQ versus Existing Approaches.

9.2 Core System Advantages

- **Semantic, Version-Aware Verification:** conflicts are detected on meaning and across the version lifecycle, capturing paraphrase, removal and supersession that lexical diffing misses.
- **Typed, Actionable Output:** every clause transition is labelled and, where applicable, escalated to a direct contradiction with a human-readable rationale rather than reduced to an opaque score.
- **Adaptive Robustness:** distribution-aware thresholds transfer across corpora with differing similarity profiles, reducing brittle manual tuning.
- **Engineering Resilience:** a decoupled microservices design with JWT/RBAC security, vector-store fallback and asynchronous workers supports availability and horizontal scaling.

10. LIMITATIONS

PolicyIQ is a functional research prototype, and several limitations bound the claims made in this paper and define the agenda for validation. First, the reported figures are design targets on a controlled synthetic corpus generated from templated clause relationships; they are not a substitute for evaluation on a licensed, human-annotated corpus of real versioned policies with expert ground truth, which remains future work. Second, direct-conflict detection relies on similarity combined with negation and numeric cues rather than full deontic logic, and does not

yet model authority, scope, exceptions or temporal supersession as first-class typed relations. Third, the adaptive estimator characterises the similarity distribution but is not yet coupled to an independent train/validation/test split or to a calibrated abstention mechanism, both of which are required before publishable accuracy claims can be made. Finally, the current pipeline targets English-language policy prose; multilingual and cross-jurisdiction analysis would require multilingual embeddings and domain-adapted models.

11. CONCLUSION

This paper presented PolicyIQ, a version-aware, multi-stage artificial intelligence platform for automated policy conflict verification and change intelligence. By unifying robust text extraction, hybrid clause segmentation, dense Sentence-BERT embeddings, cross-version typed conflict classification, distribution-aware adaptive thresholding and dual-audience summarisation within a hardened four-tier microservices architecture, PolicyIQ demonstrates a reproducible pathway from raw policy text to explainable, actionable conflict intelligence. The design directly targets the gaps left by lexical diffing and version-blind similarity tools, replacing opaque scalar scores with typed, provenance-preserving verdicts secured behind an authenticated, role-based API. On a controlled synthetic benchmark the platform meets its design targets for segmentation and adaptive conflict classification while sustaining sub-second clause-level inference, establishing a solid foundation for rigorous, human-annotated evaluation.

12. REALISTIC FUTURE WORK

- **Human-Annotated Benchmark:** construct a licensed, leakage-safe corpus of real versioned policies with typed expert annotations to enable admissible, statistically grounded evaluation with confidence intervals and significance testing.
- **Typed Normative Frames and Calibrated Abstention:** extract deontic frames (actor, obligation, condition, exception, effective date) and couple them with deterministic constraint checking and a calibrated abstention mechanism that defers uncertain cases to human reviewers.
- **Domain-Adapted and Multilingual Models:** replace the general-purpose encoder with legal-domain and multilingual sentence models to support cross-jurisdiction comparison.
- **Conflict Graphs:** model clauses as nodes in a provenance-preserving knowledge graph to surface transitive and indirect conflicts across whole policy portfolios.

REFERENCES

- [1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in Proc. ICLR Workshop, 2013.
- [2] A. Vaswani et al., “Attention is all you need,” in Proc. NeurIPS, Long Beach, CA, USA, 2017, pp. 5998–6008.
- [3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.
- [4] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in Proc. EMNLP-IJCNLP, 2019, pp. 3982–3992.
- [5] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, “A large annotated corpus for learning natural language inference,” in Proc. EMNLP, 2015, pp. 632–642.
- [6] Y. Koreeda and C. D. Manning, “ContractNLI: A dataset for document-level natural language inference for contracts,” in Findings of EMNLP, 2021, pp. 1907–1919.
- [7] LegalWiz, “A multi-agent generation framework for contradiction detection in legal documents,” arXiv preprint arXiv:2510.03418, 2025.
- [8] “NLP-based regulatory compliance: Using GPT-4.0 to decode regulatory documents,” arXiv preprint arXiv:2412.20602, 2024.
- [9] M. Lippi et al., “Deep learning for conflicting statements detection in text,” PeerJ Preprints, vol. 6, e26589v1, 2018.
- [10] S. Gupta et al., “Identifying contradictions in the legal proceedings using natural language models,” SN Computer Science, vol. 3, no. 3, 2022.
- [11] A. Mourad and H. Jebbaoui, “Conflict detection and resolution of XACML policies,” in Proc. IEEE Symp. Computers and Communications, 2014.
- [12] M. A. Rahman et al., “Clustering-based approach for anomaly detection in XACML policies,” in Proc. SECURITY, 2017, pp. 548–553.
- [13] H. Shah and P. Kumar, “Automated interpretation of regulations using NLP for compliance-centric analysis of legal texts,” ResearchGate preprint, 2024.
- [14] R. Mohan et al., “Detecting compliance of privacy policies with data protection laws,” arXiv preprint arXiv:2102.12362, 2021.
- [15] D. Silva et al., “Analysing similarities between legal court documents using NLP approaches based on transformers,” PLoS ONE / PMC, 2024.
- [16] I. Chalkidis et al., “LEGAL-BERT: The muppets straight out of law school,” in Findings of EMNLP, 2020, pp. 2898–2904.
- [17] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with GPUs,” IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.
- [18] Yu. A. Malkov and D. A. Yashunin, “Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 4, pp. 824–836, 2020.
- [19] R. Mihalcea and P. Tarau, “TextRank: Bringing order into texts,” in Proc. EMNLP, 2004, pp. 404–411.
- [20] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-based access control models,” IEEE Computer, vol. 29, no. 2, pp. 38–47, 1996.
- [21] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.

-
- [22] R. Smith, “An overview of the Tesseract OCR engine,” in Proc. ICDAR, 2007, pp. 629–633.
 - [23] Y. Liu et al., “RoBERTa: A robustly optimized BERT pretraining approach,” arXiv preprint arXiv:1907.11692, 2019.
 - [24] W. Wang et al., “MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers,” in Proc. NeurIPS, 2020.
 - [25] S. Ramírez, “FastAPI: Modern, fast web framework for building APIs with Python,” software documentation, 2023.
 - [26] Qdrant Team, “Qdrant: Vector similarity search engine,” software documentation, 2024.