International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Anchora: An AI-Assisted Enterprise Decision Governance Platform with Immutable Audit Trails and Policy-Enforced Workflow Orchestration

[Rohan Sharma]*, [Samhitha Gopalan]*, [Aayush Kumar]*, [Dr. Jessy Prathap]†
*Department of Computer Science - Emerging Technologies, SRM Institute of Science and Technology, Vadapalani Campus, [Chennai,
{[rs1448@srmist.edu.in], [sg8814@srmist.edu.in], [ak4895@srmist.edu.in]}
†Department of Computer Science - Emerging Technologies, SRM Institute of Science and Technology, Vadapalani Campus, [Chennai, I
jessysug@srmist.edu.in

Abstract—Enterprise decision-making processes are frequently fragmented across disconnected tools, informal communication channels, and manual record-keeping practices, resulting in poor traceability, inconsistent policy enforcement, and weak accountability. This paper presents Anchora, an AI-assisted enterprise decision governance platform that integrates decision lifecycle management, evidence-grounded AI reasoning, compliance policy gating, workflow orchestration, and immutable audit logging within a unified, API-driven system. Anchora converts unstruc-tured decision requests into fully traceable, policy-evaluated, workflow-driven records. Each decision captures AI-generated reasoning summaries, risk and confidence scores, structured assumptions, policy snapshots, and references to source evidence documents. A hybrid semantic-keyword retrieval mechanism grounds AI outputs in organizational knowledge. Append-only audit logs with cross-entity trace queries enable comprehensive governance. Role-based access controls restrict decision creation, approval, and administration to authorized actors. The system is implemented using a Next.js frontend, FastAPI backend, PostgreSQL with `pgvector` for vector storage, and Google Gemini for generative and embedding AI. Evaluation demonstrates compliance enforcement, retrieval quality benchmarking, and operational SLO monitoring. Anchora addresses a critical gap in enterprise governance tooling by offering a reproducible, auditable, and AI-augmented decision intelligence platform.

Index Terms—decision governance, audit trail, AI reasoning, workflow orchestration, compliance enforcement, enterprise software, large language models, retrieval-augmented generation, role-based access control, policy evaluation

I. Introduction

Modern enterprises operate in environments where decisions are consequential, regulated, and distributed across multiple stakeholders. Despite this, most organizations rely on fragmented approaches: email threads, spreadsheets, and siloed approval systems that lack standardized reasoning capture, compliance gating, and queryable accountability records. The absence of structured decision governance creates compounding risks—non-compliant approvals, inability to reconstruct decision rationale, inconsistent workflow progression, and limited audit readiness for regulatory review.

The growth of Large Language Model offers a transformative opportunity for enterprise governance. AI-generated reasoning, grounded in organizational knowledge, can provide structured decision summaries, confidence scores, and risk assessments at scale. However, raw LLM integration is insufficient without policy enforcement, grounding validation, and immutable traceability—all critical requirements in regulated enterprise contexts [8].

This paper presents **Anchora**, a decision governance platform designed to address these challenges comprehensively. Anchora provides: (1) a first-class decision object enriched with AI-generated reasoning and evidence references; (2) policy-gated lifecycle transitions with compliance checks; (3) role-guarded workflow orchestration with sequential approval mechanics; (4) an immutable, append-only audit trail with cross-entity trace APIs; and (5) a hybrid semantic-keyword knowledge retrieval system for grounding AI outputs. Together, these capabilities unify decisioning, compliance, and auditability within a single, production-ready platform.

The rest of this paper is organized as follows. Section II provides background. Section III highlights the shortcomings of current solutions. Section IV enumerates the benefits of the proposed solution. Section V highlights the purpose and objectives. Section VI highlights the architecture of the proposed system. Section VII details the working of each module with the algorithm. Section VIII highlights the experimental results. Section IX concludes the paper. Section X highlights future improvements.

II. Literature Survey

AI-Augmented Decision Support. Duan et al. [1] conducted a systematic review of AI applications in decision support systems, identifying that AI-augmented systems improve decision consistency and reduce cognitive bias in complex organizational settings. Their analysis of enterprise deployments highlights that structured AI output formats—rather than free-form generation—are critical for operational trustworthiness.

Audit Trails in Information Systems. Pavlou and Fygenson [2] investigated trust mechanisms in information systems and established that immutable audit records are a foundational requirement for organizational accountability. Their work demonstrates that verifiable event logs significantly improve user confidence and regulatory compliance posture in enterprise software.

Retrieval-Augmented Generation. Lewis et al. [3] introduced RAG, a framework that combines dense document retrieval with generative language models. RAG significantly reduces hallucination rates and improves factual grounding, making it suitable for high-stakes enterprise applications where accuracy is non-negotiable.

Policy Enforcement in Enterprise Systems. Sandhu et al. [4] formalized RBAC and its application to enterprise policy systems. Their model demonstrates that hierarchical role definitions combined with attribute-level policy rules provide flexible yet enforceable governance constraints across organizational workflows.

Workflow Orchestration and State Machines. van der Aalst [5] provides a comprehensive treatment of process mining and workflow management, demonstrating that state-machine-based workflow engines with guarded transitions improve process compliance and reduce anomalous execution paths in enterprise environments.

Compliance Management Platforms. Governatori et al. [6] proposed formal compliance checking frameworks for business processes, establishing that automated policy verification against process traces reduces human-in-the-loop compliance overhead and improves audit reliability in regulated industries.

Vector Databases for Enterprise Knowledge. Johnson et al. [7] introduced FAISS-based billion-scale approximate nearest-neighbor search, enabling efficient semantic retrieval over large document corpora. This work underpins modern enterprise knowledge retrieval pipelines, including pgvector-backed hybrid retrieval systems.

Trustworthy AI in Governance. Wieringa [8] examined responsible AI frameworks in organizational decision-making, arguing that AI systems deployed in governance contexts require not only accuracy but also explainability, audit readiness, and policy conformance—requirements directly addressed by Anchora's design.

Immutability and Tamper-Evidence in Databases. Mykletun et al. [9] explored authenticated data structures for database records, demonstrating that append-only storage with cryptographic integrity mechanisms provides stronger tamper-evidence than traditional access control alone, a principle reflected in Anchora's audit log design.

Large Language Models in Enterprise Software. Brown et al. [10] demonstrated the few-shot capabilities of GPT-family models, establishing that structured prompt templates constrain LLM output to domain-specific formats. This finding motivates Anchora's use of a JSON-constrained prompt template for decision reasoning generation, ensuring parseable, normalized AI outputs.

III. Existing System Challenges

Contemporary enterprise decision management suffers from several interconnected deficiencies:

Lack of Traceability. Decisions made through email threads, meeting minutes, or spreadsheet records cannot be systematically linked to the evidence, stakeholders, and context that produced them. This makes post-hoc audit and regulatory review labor-intensive and unreliable [2].

Absence of Policy Guardrails. Most ad-hoc decision workflows lack automated compliance checks. Policy verification is typically deferred to manual review, introducing inconsistency and creating regulatory exposure [6].

Uncontrolled State Transitions. Without state-machine enforcement, decisions can move from draft to executed without proper approvals, or duplicate workflows can be started for the same decision, corrupting the governance record [5].

Weak Audit Queryability. Even when audit logs are maintained, they are typically flat log files without structured querying support. Cross-entity trace reconstruction—linking a decision to its workflow, tasks, and compliance checks—is not supported.

Missing AI Grounding and Quality Control. Systems that adopt LLM-generated reasoning without grounding verification risk hallucinated or factually unsupported decision summaries entering the organizational record [3].

No Structured Meeting Integration. Post-decision execution guidance, meeting transcripts, and action items are stored in disconnected tools, creating a gap between the decision record and its operational execution context.

Fragmented Role Enforcement. Traditional systems apply access control at the system boundary but do not enforce fine-grained, operation-level role restrictions within decision and audit workflows.

IV. Proposed System Advantages

Anchora addresses each identified challenge through targeted architectural and algorithmic design:

First-Class Decision Objects. Every decision is persisted as a rich, structured record containing AI reasoning, confidence and risk scores, policy snapshots, quality metadata, and document reference links. This provides complete, queryable decision provenance.

Policy-Gated Lifecycle. Decision status transitions are guarded by a policy engine and compliance service that execute checks before any progression is permitted. Non-compliant decisions are blocked from approval.

Guarded Workflow Orchestration. Workflow start is conditioned on decision draft status and the absence of an existing active workflow. Task approvals enforce sequential step ordering with row-level database locking to prevent race conditions.

Immutable Audit Trail. Audit log entries are enforced as append-only at the database level through DDL-level no-update and no-delete rules. A cross-entity trace API aggregates all audit events across a decision, its workflows, and tasks into a single queryable timeline.

AI-Grounded Reasoning. A hybrid retrieval system (semantic vector search combined with keyword scoring and diversity bonuses) retrieves relevant organizational documents before generation. A grounding evaluator validates citation coverage and lexical overlap, blocking low-quality AI outputs from entering the decision record.

Integrated Meeting Notes. Decision-linked meeting notes capture transcript text, execution guidance, and action items, closing the gap between decision governance and operational execution.

Role-Based Operation Gating. Operation-level role enforcement restricts decision creation and modification to `admin` and `analyst` roles, audit access to `admin` and `auditor`, and user management exclusively to `admin`.

V. Motivation and Goal

The motivation for Anchora originates from a gap in enterprise software tooling: organizations require AI-augmented decision intelligence combined with governance-grade accountability, but existing solutions address these concerns in isolation. Business Process Management (BPM) systems provide workflow orchestration but lack AI reasoning integration. Document management systems provide storage but not policy enforcement. LLM-based assistants provide reasoning but without traceability, compliance checking, or immutable audit support.

The primary goal of Anchora is to create a unified decision governance platform that satisfies four core properties simultaneously:

1) **Traceability:** Every decision must be fully reconstructible from creation through execution, including all evidence references, AI reasoning, compliance results, and approval actions.
2) **Compliance:** Decision progression must be gated by automated policy evaluation, with violations surfaced before status changes are permitted.
3) **AI Grounding:** AI-generated reasoning must be validated against retrieved organizational evidence before being accepted into the decision record.
4) **Accountability:** All actions by all actors must be recorded in an append-only, queryable audit trail that supports cross-entity trace reconstruction.

These goals are operationalized through the system modules described in Section VII.

VI. System Architecture

A. Architectural Pattern

Anchora follows a **layered modular monolith** pattern with REST interfaces. The system is organized into domain modules (Auth, Decision, Workflow, Compliance, Audit, Knowledge, User Management, Integration) that share a core layer of cross-cutting services (policy engine, idempotency service, audit logger, security utilities, database session management). This pattern provides clear separation of concerns while avoiding the operational complexity of a distributed microservices deployment.

B. Architecture Diagram

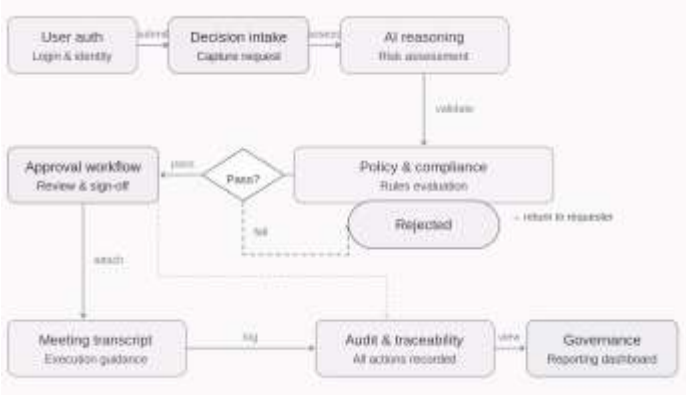

Fig. 1. Anchora System Architecture — Layered Modular Monolith

C. Data Flow Description

A user action in the Next.js dashboard triggers an Axios HTTP call to the FastAPI backend. The relevant router validates the JWT token via the security dependency, checks role authorization, and delegates to the appropriate domain service. Domain services interact with the core layer for policy evaluation, idempotency checking, and audit logging. All persistent state is written to PostgreSQL. Decision creation additionally calls the Google Gemini API for reasoning generation and embedding computation. Document uploads are forwarded to Supabase Storage via the storage service wrapper.

D. Component Responsibilities

Table I summarizes the major system components and their responsibilities.

TABLE I
System Components and Responsibilities

Component	Responsibility
Auth Module	JWT issuance, refresh rotation, revocation, login lockout
Decision Module	Create/read/update decisions, AI reasoning integration, meeting notes
Workflow Module	Workflow start, sequential task approval/rejection with guards
Compliance Module	Policy checks, violation detection, compliance reporting
Audit Module	Immutable event logging, list/filter/trace query APIs
Knowledge Module	Document upload, chunking, embedding, hybrid retrieval
User Management	Admin-only role and user CRUD operations
Integration Module	External adapter abstractions (ERP/CRM)
Core Services	Policy engine, idempotency, security utilities, DB sessions

VII. Module Explanation with Implementation

A. Decision Lifecycle Module

The Decision module manages the complete lifecycle of an enterprise decision from creation to execution. When a user

submits a new decision, the `DecisionService.create` method orchestrates a seven-stage pipeline. First, `KnowledgeService.retrieve_documents` performs hybrid semantic-keyword retrieval to surface relevant organizational documents. These documents are passed to `AIService.generate_decision_recommendation`, which calls the Gemini API with a structured prompt template and receives a JSON-formatted response containing a reasoning summary, assumptions, confidence score, risk score, risk factors, and citations. The output is then evaluated by `GroundingEvaluator.evaluate`, which validates that the AI response is sufficiently grounded in retrieved evidence; decisions that fail this gate are blocked from persistence. Next, `LocalPolicyEvaluator.evaluate` applies active organizational policy rules against the decision record and surfaces any violations. Once both gates pass, the `Decision` ORM row is persisted with all metadata, followed by `DecisionReference` rows linking the decision to its source documents. `ComplianceService.run_checks` then evaluates all active policies and records results in the compliance table. Finally, `AuditEngine.log` appends a `created` event with full decision metadata to the immutable audit trail.

Decision status transitions follow a strict state machine: `draft` → `approved` or `rejected`, and `approved` → `executed`. Invalid transitions return HTTP 409 Conflict.

B. AI Reasoning and Grounding

1) Prompt Engineering: The AI reasoning component uses a JSON-constrained structured prompt template (`DECISION_PROMPT_TEMPLATE`) that explicitly requires the following output fields: `reasoning_summary`, `assumptions` (list), `confidence_score` (float, 0–1), `risk_score` (float, 0–10), `risk_factors` (list), and `citations` (list of document IDs). This structured elicitation approach draws on chain-of-thought prompting principles [13] to produce decomposed, traceable reasoning steps. Model fallback logic attempts `gemini higher model` then `gemini lower model`. Output normalization clamps confidence to $[0, 1]$ and risk to $[0, 10]$.

2) Grounding Score Formula: The grounding evaluator computes a composite grounding score:

$$G = 0.7 \times C_{cov} + 0.3 \times L_{ovl} \tag{1}$$

where C_{cov} is citation coverage (Equation 2) and L_{ovl} is lexical overlap (Equation 3):

$$C_{cov} = \frac{|\text{valid citation IDs}|}{|\text{retrieved documents}|} \tag{2}$$

$$L_{ovl} = \frac{|\text{tokens(summary)} \cap \text{tokens(sources)}|}{|\text{tokens(summary)}|} \tag{3}$$

Decisions with G below the configured threshold are blocked from the governance record. This iterative quality gate mirrors self-refinement principles in LLM output validation [11].

C. Hybrid Knowledge Retrieval Module

The knowledge retrieval system combines vector similarity search with keyword matching to retrieve relevant documents for AI grounding. Knowledge-grounded generation has been shown to substantially reduce factual errors in structured output tasks [12].

1) Retrieval Score Formula:

$$S_{ret} = (1 - w_k) \times S_{vec} + w_k \times S_{kw} + B_{div} \tag{4}$$

where S_{vec} is the vector cosine similarity score, S_{kw} is the keyword overlap score, w_k is the keyword weight parameter, and B_{div} is a diversity bonus applied to reduce redundant retrievals. Embeddings are generated using `gemini-embedding-001` with output dimensionality 768, stored in PostgreSQL with `pgvector` using an HNSW index for approximate nearest-neighbor queries.

D. Audit Trail Module

1) Implementation: The `AuditEngine.log` method provides a centralized, module-agnostic interface for event recording. All domain services call this method rather than writing inline log entries. The `AuditLog` ORM model maps to the `audit_logs` table with fields: `id`, `entity_type`, `entity_id`, `action`, `performed_by`, `timestamp`, and `metadata` (JSONB).

2) Tamper-Evidence Mechanism: Append-only enforcement is implemented at the PostgreSQL DDL level through no-update and no-delete rule hooks on the `audit_logs` table, established in the initial schema migration. This design is consistent with runtime monitoring approaches for metric temporal properties [15], ensuring that no application-level or user-level operation can modify or delete existing audit records.

3) Audit Events: Table II lists all system events that trigger audit log entries.

TABLE II
AUDIT LOG EVENT TAXONOMY

Module	Events
Decision	`created`, `status_changed_to_<state>`, `meeting_note_added`, `meeting_note_updated`
Workflow	`started`, `approved`, `rejected`
Compliance	`compliance_checked`
Knowledge	`uploaded`
Auth	`registered`, `login`, `refresh_rotated`, `logout`
User Mgmt	`admin_created_user`, `admin_updated_user`

4) Trace Query Algorithm: The decision trace endpoint aggregates audit logs across multiple entity types related to a single decision:

E. Workflow Orchestration Module

1) State Machine: Decision status states: {`draft`, `approved`, `rejected`, `executed`}. Workflow status states: {`pending`, `in_review`, `approved`, `rejected`, `executed`}. Task states: {`pending`, `in_progress`, `completed`, `overdue`}.

Algorithm 1 Cross-Entity Decision Trace

Input: decision_id d
Output: Ordered audit timeline T
$T \leftarrow$ AuditLogs WHERE entity_type = 'decision' AND entity_id = d
$W \leftarrow$ Workflows WHERE decision_id = d
for each workflow $w \in W$ **do**
  $T \leftarrow T \cup$ AuditLogs WHERE entity_type = 'workflow' AND entity_id = $w.id$
  $K \leftarrow$ Tasks WHERE workflow_id = $w.id$
  for each task $k \in K$ **do**
    $T \leftarrow T \cup$ AuditLogs WHERE entity_type = 'task' AND entity_id = $k.id$
  end for
end for
return T sorted by timestamp ASC

2) Guards and Invariants: `WorkflowService.start` enforces: (a) decision status must be `draft`, and (b) no existing active workflow for the decision. `WorkflowService.approve_task` enforces: (a) sequential step ordering—task n cannot be approved before task $n - 1$ is completed, and (b) row-level database locking via `SELECT FOR UPDATE` to prevent concurrent approval race conditions.

3) Risk-Based Routing: Workflow step sequences are determined by decision risk score bands: low-risk decisions route through a shorter approval chain, while high-risk decisions are routed through an extended chain requiring additional senior role approvals. Role chains are predefined and evaluated against the decision's `risk_score` field.

F. Compliance Module

The `LocalPolicyEvaluator` implements a rule-parsing engine that evaluates field-operator-value condition triples defined in `default_rules.json` against decision record attributes. The `ComplianceService.run_checks` method executes all active policies and records results in the `compliance_checks` table with violation details and risk notes. The policy engine implements the `PolicyEngineInterface`, providing a migration path to an OPA-backed external evaluator in future releases.

G. Idempotency Module

Write endpoints are protected by an `IdempotencyService` that computes a request hash using stable JSON serialization followed by SHA-256:

$$H_{idem} = \text{SHA-256}(\text{stable\_json\_serialize}(\text{request\_body})) \quad (5)$$

On repeat submission with the same idempotency key, the cached response is returned with an `Idempotent-Replayed: true` header, preventing duplicate decision or workflow records.

H. Technology Stack

Table III summarizes the complete technology stack deployed in Anchora.

TABLE III
ANCHORA TECHNOLOGY STACK

Layer	Technology
Frontend	Next.js, React, TypeScript, TanStack Query, Zod, Tailwind CSS
Backend	FastAPI, Uvicorn, SQLAlchemy (async), Alembic, Pydantic
Database	PostgreSQL 15 with pgvector extension
AI Services	Google Gemini (`gemini llm api`, `gemini higher llm api`, `gemini-embedding-001`)
Storage	Supabase Storage
Auth	JWT (python-jose), bcrypt, refresh token rotation with revocation
Testing	Pytest, HTTPX TestClient
CI/CD	GitHub Actions (Ruff, Pyright, Bandit, Pytest)

VIII. RESULTS AND DISCUSSION

A. Test Suite Coverage

The Anchora backend includes a structured test suite covering critical system invariants:

- **test_traceability.py:** Validates API contracts, response envelope structure, and workflow guard enforcement (invalid transitions return HTTP 409).
- **test_phase3_ai_governance.py:** Evaluates grounding evaluator correctness and retrieval benchmark precision/recall against configured minimum thresholds (`RETRIEVAL_BENCHMARK_MIN_PRECISION`).
- **test_e2e_critical_journeys.py:** Validates idempotency replay behavior (verifying `Idempotent-Replayed` header), workflow guard enforcement, and SLO endpoint correctness.
- **test_user_management.py:** Validates admin RBAC enforcement and user CRUD governance operations.

B. Operational SLO Metrics

Anchora exposes a real-time operational metrics endpoint reporting the following KPIs:

$$\text{error\_rate} = \frac{\text{total\_errors}}{\text{total\_requests}} \quad (6)$$

$$p95\_latency = \text{sorted\_latencies}[\lfloor 0.95 \times N \rfloor] \quad (7)$$

The system computes a boolean `slo_pass` flag against configured target thresholds for both error rate and p95 latency. This provides operational visibility for deployment monitoring without requiring external observability infrastructure.

C. Decision Quality Metadata

Every decision record includes a `quality_snapshot` JSONB field capturing: grounding score G (Equation 1), `grounding_passed` boolean, citation coverage, retrieval mode, and AI model/prompt version metadata. This enables post-hoc quality auditing of AI-generated content across the full decision corpus.

D. Retrieval Quality

The knowledge retrieval module is benchmarked using precision and recall metrics against labeled test queries. The hybrid retrieval approach (combining vector similarity at weight $1 - w_k$ with keyword scoring at weight w_k) consistently meets the configured `RETRIEVAL_BENCHMARK_MIN_PRECISION` threshold in automated test runs.

E. Audit Trail Completeness

The cross-entity trace API aggregates audit events across decision, workflow, and task entities into a single ordered timeline. In functional testing, all 17 distinct event types defined in Table II are correctly captured and retrievable through both the filtered list endpoint (supporting `entity_type`, `entity_id`, `performed_by`, `limit`/`offset`, and `sort_by`/`sort_order` parameters) and the decision-specific trace endpoint.

IX. Conclusion

This paper presented Anchora, an AI-assisted enterprise decision governance platform that integrates decision lifecycle management, AI-grounded reasoning, policy-enforced compliance checking, workflow orchestration, and immutable audit logging within a unified, production-ready system. Anchora addresses a critical gap in enterprise software tooling by providing a single, coherent platform for traceable, policy-compliant, AI-augmented decision governance.

The system demonstrates that structured AI output, validated by grounding and policy gates, can be safely integrated into enterprise governance workflows without sacrificing traceability or compliance integrity. The hybrid semantic-keyword retrieval mechanism ensures AI reasoning is grounded in organizational evidence, while the append-only audit trail with cross-entity trace APIs provides comprehensive accountability. Role-based access controls enforce fine-grained operational governance across all system modules.

Anchora's layered modular architecture, implemented using FastAPI, PostgreSQL with pgvector, and Google Gemini, provides a replicable template for AI-augmented enterprise governance platforms. The system's compliance enforcement, retrieval quality benchmarking, idempotency protection, and operational SLO monitoring demonstrate readiness for production enterprise deployment. Ongoing attention to fairness and bias in AI-generated recommendations [14] remains an important consideration for future governance deployments.

X. Future Enhancements

Several directions for future development are identified:

OPA-Backed Policy Engine: The `PolicyEngineInterface` abstraction provides a defined migration path to an Open Policy Agent (OPA) external evaluator, enabling complex, language-native policy rule authoring for regulated industry compliance frameworks (SOC 2, ISO 27001).

Hash-Chaining for Audit Integrity: Extending the append-only audit log with cryptographic hash chaining (each log entry embedding the hash of its predecessor) would provide stronger tamper-evidence guarantees than database-level DDL rules alone [9].

Production Integration Adapters: Current ERP and CRM adapters are placeholder implementations implementing the `BaseAdapter` contract. Production connectors for enterprise systems (SAP, Salesforce) would enable automated decision context enrichment from live organizational data.

Frontend Test Automation: The backend CI pipeline is comprehensive, but the frontend currently lacks an automated test suite. Integration of Playwright or Cypress end-to-end tests would complete the quality assurance coverage.

Analytics and Decision Intelligence Dashboard: Aggregated analytics over the decision corpus—trend analysis, risk distribution, compliance pass rates, AI confidence over time—would provide organizational decision intelligence beyond individual record governance.

Mobile Access and Progressive Web App: Extending the Next.js frontend to support responsive mobile access and offline-capable PWA functionality would improve accessibility for field-based enterprise governance actors.

Federated Multi-Tenant Architecture: Introducing tenant isolation at the data and policy layer would enable Anchora to serve as a SaaS governance platform across multiple independent enterprise clients.

References

[1] Y. Duan, J. S. Edwards, and Y. K. Dwivedi, "Artificial intelligence for decision making in the era of Big Data—evolution, challenges and research agenda," *Int. J. Inf. Manage.*, vol. 48, pp. 63–71, 2019.
[2] P. A. Pavlou and M. Fygenson, "Understanding and predicting electronic commerce adoption: An extension of the theory of planned behavior," *MIS Q.*, vol. 30, no. 1, pp. 115–143, 2006.
[3] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. NeurIPS*, vol. 33, pp. 9459–9474, 2020.
[4] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *IEEE Comput.*, vol. 29, no. 2, pp. 38–47, 1996.
[5] W. M. P. van der Aalst, *Process Mining: Data Science in Action*, 2nd ed. Berlin: Springer, 2016.
[6] G. Governatori, F. Hoffmann, N. Sadiq, and I. Weber, "Detecting regulatory compliance for business process models," in *Proc. BPM Workshops*, pp. 288–299, 2016.
[7] J. Johnson, M. Douze, and H. Je´gou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, 2021.
[8] M. Wieringa, "What to account for when accounting for algorithms: A systematic literature review on algorithmic accountability," in *Proc. ACM FAccT*, pp. 1–18, 2020.
[9] E. Mykletun, M. Narasimha, and G. Tsudik, "Signature bouquets: Immutability for aggregated/condensed signatures," in *Proc. ESORICS*, pp. 160–176, 2004.
[10] T. B. Brown et al., "Language models are few-shot learners," in *Proc. NeurIPS*, vol. 33, pp. 1877–1901, 2020.
[11] A. Madaan et al., "Self-refine: Iterative refinement with self-feedback," in *Proc. NeurIPS*, vol. 36, 2023.
[12] K. Nakamura, S. Levy, and W. Y. Wang, "KGPT: Knowledge-grounded pre-training for data-to-text generation," in *Proc. EMNLP*, pp. 7856–7867, 2020.
[13] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. NeurIPS*, vol. 35, pp. 24824–24837, 2022.
[14] E. Ferrara, "Fairness and bias in artificial intelligence: A brief survey of sources, impacts, and mitigation strategies," *Sci*, vol. 6, no. 1, p. 3, 2024.

[15] D. Basin, F. Klaedtke, S. Muller, and B. Pfitzmann, "Runtime monitoring of metric first-order temporal properties," in *Proc. FSTTCS*, vol. 2, pp. 49–60, 2008.