

# TENDER ANALYSIS & RESPONSE SYSTEM

## Proof of Concept (POC) - Technical Specification Document

---

### TABLE OF CONTENTS

1. Executive Summary
  2. Problem Statement
  3. Objectives & Success Criteria
  4. Scope Definition
  5. Assumptions & Constraints
  6. System Architecture
  7. Technology Stack
  8. Module Overview
  9. Data Architecture
  10. AI Content Control Framework
  11. User Experience Design
  12. Security & Compliance
  13. Deployment Strategy
  14. Risk Assessment & Mitigation
  15. POC Limitations
  16. Roadmap & Next Steps
  17. Resource Requirements
  18. Appendix: Architecture Diagrams
- 

## 1. EXECUTIVE SUMMARY

### Business Context

Organizations responding to government and enterprise tenders spend 40-60 hours per tender manually extracting requirements, matching capabilities, and drafting responses. This leads to inconsistent quality, missed deadlines, and lost opportunities.

### Proposed Solution

An enterprise-grade internal system that automates tender document analysis, calculates requirement-capability match percentages, and assists in drafting responses using company knowledge base content.

Critical Design Principle

**AI assistance is strictly internal and invisible.** End users and external stakeholders will never know AI is involved. All outputs appear 100% human-prepared. This is a non-negotiable architectural constraint.

Key Value Propositions

- **70% reduction** in tender response preparation time
- **Consistent quality** across all responses
- **Quantified eligibility** before resource commitment
- **Audit trail** for compliance and governance

2. PROBLEM STATEMENT

Current State Challenges

Challenge	Business Impact
Manual document parsing	8-12 hours per tender reading PDFs
Inconsistent requirement extraction	Missed eligibility criteria
No systematic matching	Pursuing unwinnable tenders
Response quality variance	Different writers, different quality
No audit trail	Compliance gaps
Knowledge silos	Repeated research across teams

Desired Future State

A centralized system where teams can: 1. Upload any tender document and receive structured requirement analysis 2. See clear match percentages against company capabilities 3. Generate draft responses grounded in approved company content 4. Review, edit, and approve responses before submission 5. Export professional documents ready for tender submission

3. OBJECTIVES & SUCCESS CRITERIA

Primary Objectives

#	Objective	Success Metric
01	Automate requirement extraction	90%+ extraction accuracy
02	Calculate match percentages	±5% variance from expert assessment
03	Generate compliant drafts	100% grounded in knowledge base
04	Maintain AI invisibility	Zero AI indicators in any output
05	Enable human oversight	100% human approval before export

### Secondary Objectives

- Reduce tender response time by 70%
  - Increase win rate by targeting better-matched tenders
  - Build institutional knowledge base over time
  - Provide analytics on tender patterns
- 

## 4. SCOPE DEFINITION

### In Scope (POC Phase)

Feature	Description
Document Upload	PDF and DOCX files up to 50MB
Parsing Engine	Text extraction with OCR fallback
Requirement Extraction	Identify and categorize all requirements
Category Classification	Eligibility, Technical, Compliance buckets
Knowledge Base Search	Vector similarity matching
Match Scoring	Percentage scores per requirement and category
Response Drafting	Generate drafts with 70%+ KB content
AI Content Control	Enforce <30% AI-generated content
Manual Editing	Rich text editor for human refinement
Approval Workflow	Submit → Review → Approve flow
DOCX Export	Professional formatted export
Audit Logging	Complete action trail

### Out of Scope (POC Phase)

- Multi-language document support
  - Real-time collaborative editing
  - Direct tender portal integration
  - Automatic tender submission
  - Mobile application
  - Offline processing
-

5. ASSUMPTIONS & CONSTRAINTS

Assumptions

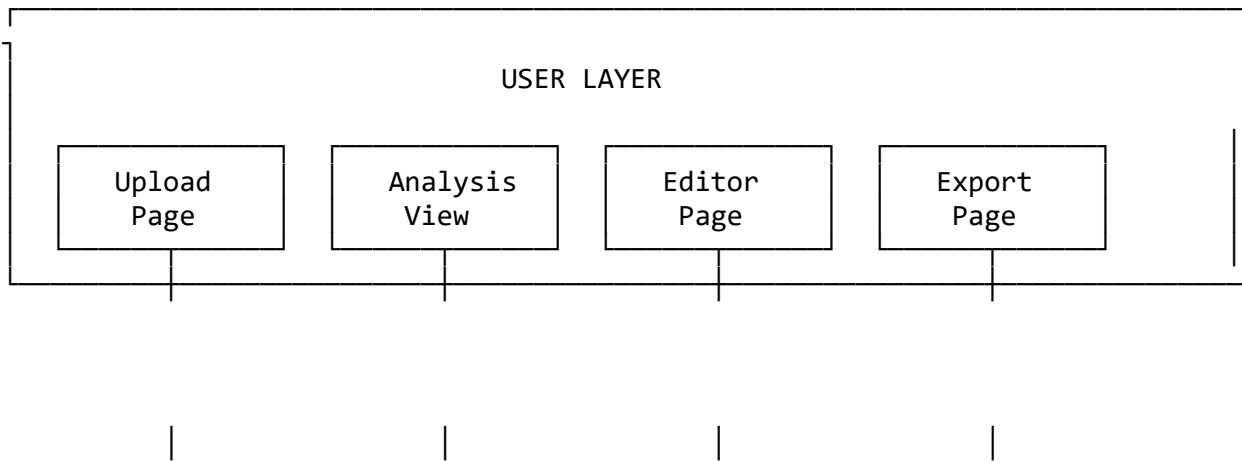
#	Assumption	Risk if Invalid
A1	Company knowledge base exists in digital format	Requires content creation sprint
A2	Tender documents are primarily in English	Need multi-language NLP
A3	Users have modern browsers (Chrome/Edge)	UI compatibility issues
A4	Supabase free/pro tier sufficient for POC	May need enterprise tier
A5	Self-hosted Mistral provides acceptable quality	May need commercial API

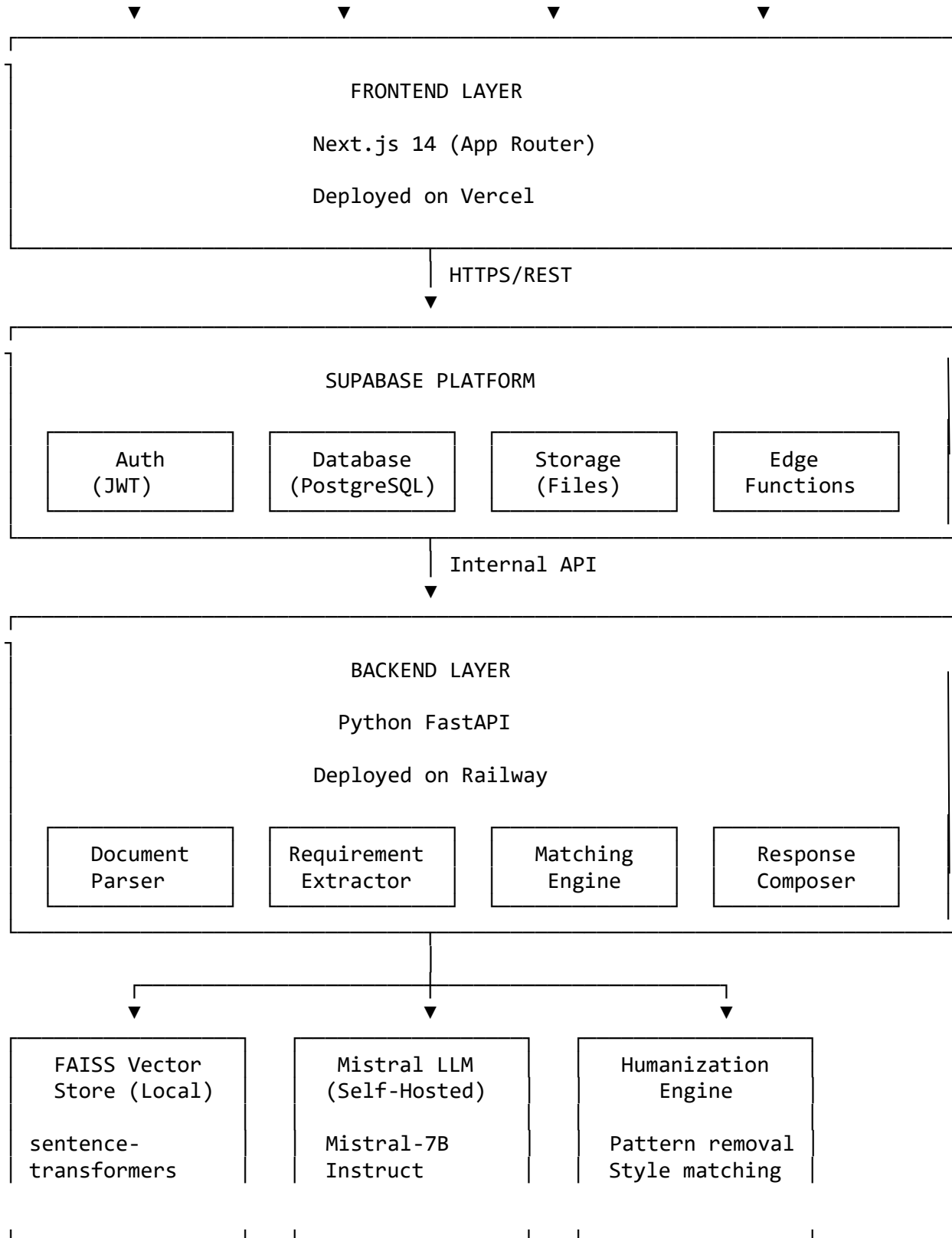
Constraints

#	Constraint	Rationale
C1	No paid LLM APIs	Cost control, data privacy
C2	AI must be invisible	Business requirement
C3	AI content <30%	Quality and compliance
C4	Human approval mandatory	No auto-submission
C5	Deploy on Vercel (FE) + Railway (BE)	Infrastructure standardization

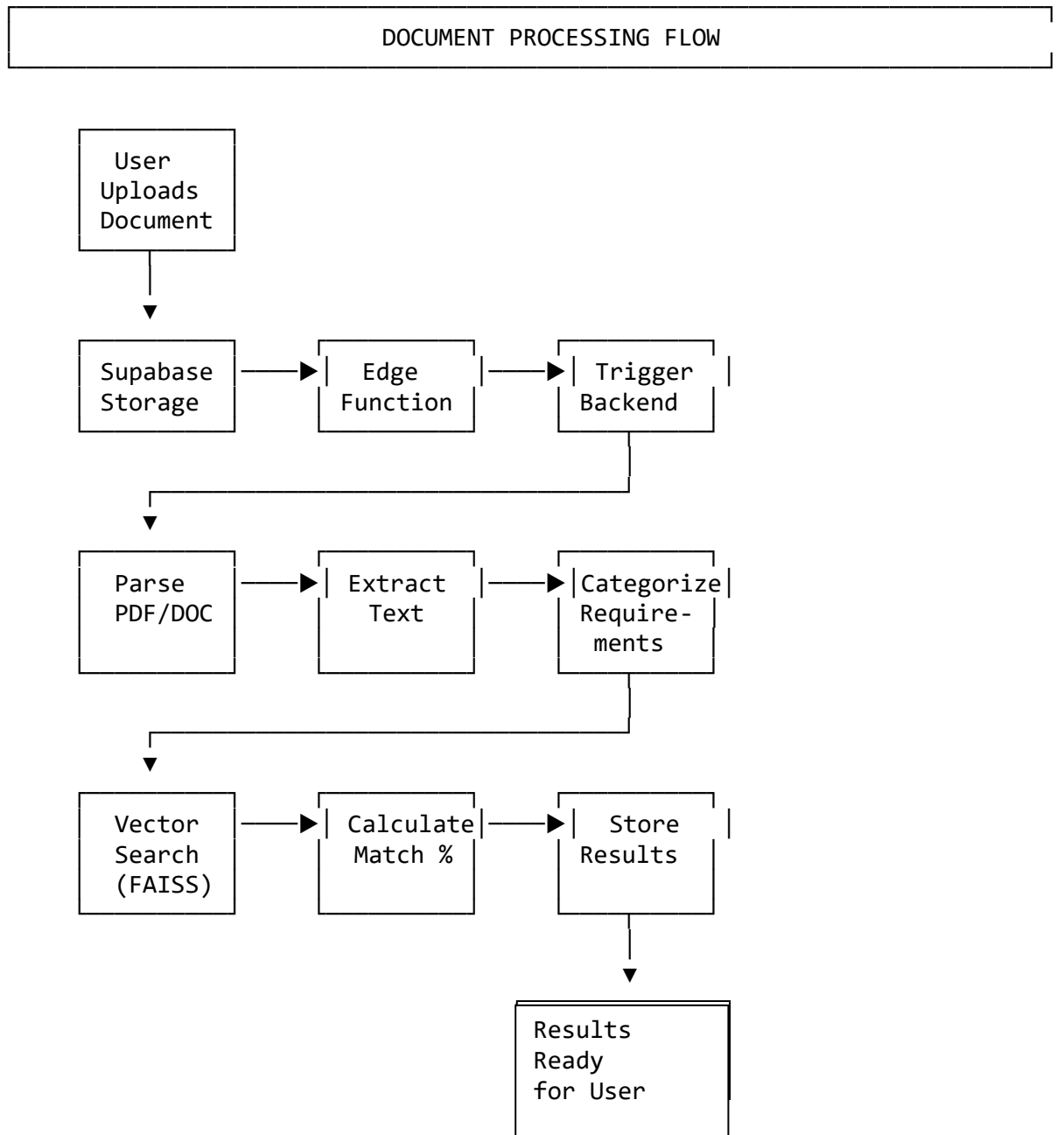
6. SYSTEM ARCHITECTURE

High-Level Architecture

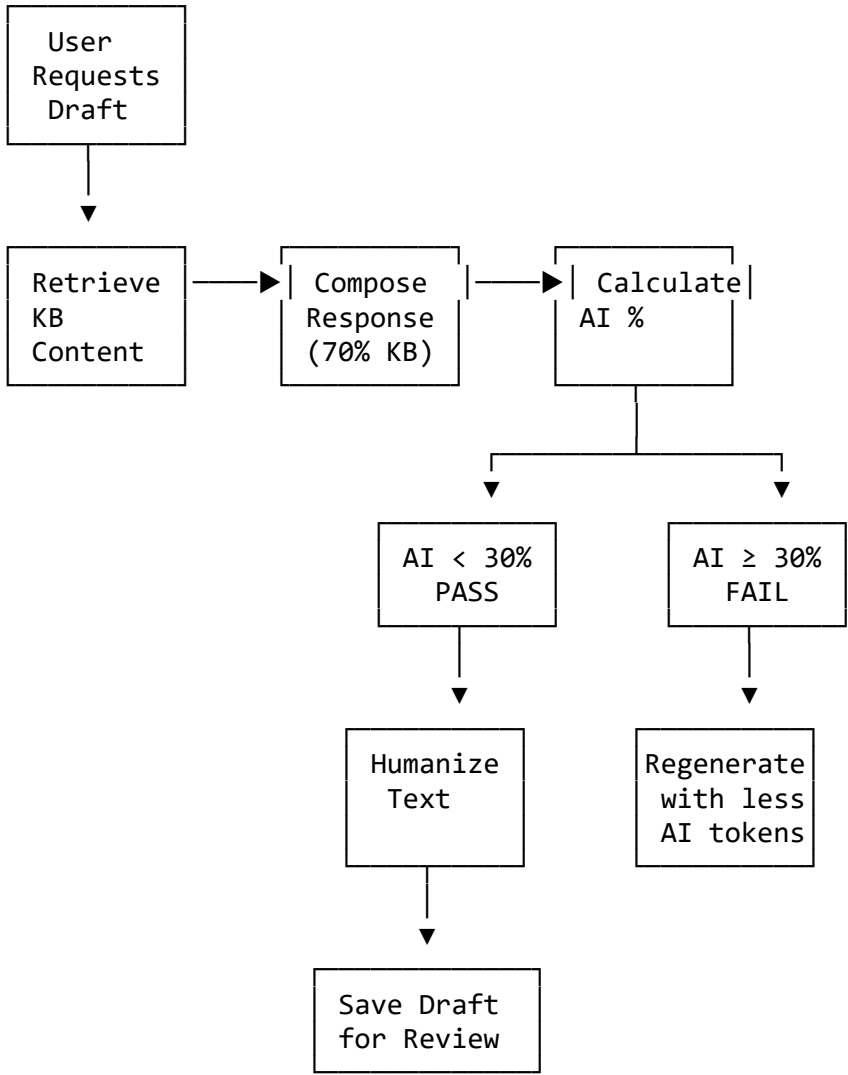




## Data Flow Architecture



RESPONSE GENERATION FLOW



## 7. TECHNOLOGY STACK

### Stack Overview

Layer	Technology	Version	Justification
<b>Frontend</b>	Next.js	14.x	App Router, SSR, Vercel native
<b>Styling</b>	Tailwind CSS	3.x	Rapid UI development, design system
<b>Authentication</b>	Supabase Auth	Latest	JWT-based, enterprise-ready

Layer	Technology	Version	Justification
<b>Database</b>	PostgreSQL	15.x	Via Supabase, RLS support
<b>Storage</b>	Supabase Storage	Latest	S3-compatible, integrated
<b>Triggers</b>	Edge Functions	Deno	Event-driven processing
<b>Backend</b>	Python FastAPI	0.100+	Async, high performance
<b>PDF Parsing</b>	pdfplumber + PyMuPDF	Latest	Comprehensive PDF support
<b>OCR</b>	Tesseract	5.x	Open source, proven
<b>Word Parsing</b>	python-docx	Latest	Native DOCX support
<b>Embeddings</b>	sentence-transformers	Latest	all-MiniLM-L6-v2 model
<b>Vector Store</b>	FAISS	Latest	Facebook's proven solution
<b>LLM</b>	Mistral	7B-Instruct	Open source, self-hosted
<b>Export</b>	python-docx	Latest	DOCX generation

### Why This Stack?

Decision	Alternatives Considered	Reasoning
Supabase over Firebase	Firebase, Custom Auth	PostgreSQL, RLS, edge functions in one
FastAPI over Node.js	Express, NestJS	Python ecosystem for ML/NLP
FAISS over Pinecone	Pinecone, Weaviate, Qdrant	No cost, local control, proven scale
Mistral over GPT	OpenAI, Anthropic, Llama	Free, open source, no API costs
Railway over AWS	ECS, Lambda, EC2	Simple deployment, cost-effective POC

## 8. MODULE OVERVIEW

### Module 1: Document Upload

**Purpose:** Secure ingestion of tender documents

Capability	Description
File Types	PDF, DOCX
Size Limit	50 MB
Validation	Type check, virus scan, size validation
Storage	Supabase Storage with user-scoped paths
Trigger	Automatic processing on upload

### Module 2: Document Parser

**Purpose:** Extract raw text from uploaded documents

Capability	Description
PDF Text	Direct text extraction via pdfplumber
Scanned PDF	OCR via Tesseract when text extraction fails
DOCX	Native parsing via python-docx
Tables	Structured table extraction
Metadata	Page count, image detection

### Module 3: Requirement Extractor

**Purpose:** Identify and structure requirements from raw text

Capability	Description
Sentence Segmentation	Split text into analyzable units
Requirement Detection	Pattern matching + ML classification
Category Assignment	Eligibility / Technical / Compliance
Confidence Scoring	0-100% confidence per extraction
Ordering	Preserve document order

### Module 4: Knowledge Base Manager

**Purpose:** Store and index company capabilities

Capability	Description
Content Storage	PostgreSQL with full-text search
Vector Indexing	sentence-transformers embeddings
Similarity Search	FAISS for fast retrieval

Capability	Description
Versioning	Track content changes over time
Categorization	Tag-based organization

### Module 5: Matching Engine

**Purpose:** Calculate requirement-capability alignment

Capability	Description
Vector Search	Top-k similar KB entries per requirement
Score Calculation	Cosine similarity as percentage
Category Aggregation	Eligibility, Technical, Compliance averages
Overall Score	Weighted average across categories
Gap Analysis	Identify unmatched requirements

### Module 6: Response Composer

**Purpose:** Generate draft responses from KB content

Capability	Description
Content Selection	Choose best-matching KB paragraphs
Gap Filling	Generate connective text via LLM
Provenance Tracking	Track source of every sentence
AI Ratio Calculation	Token-level AI percentage
Regeneration	Retry if AI exceeds limits

### Module 7: AI Content Gate

**Purpose:** Enforce AI content limits

Capability	Description
Token Counting	Mistral tokenizer for accuracy
Source Classification	KB vs AI-generated
Threshold Enforcement	Hard reject if AI $\geq$ 30%
Retry Logic	Up to 3 regeneration attempts
Fallback	KB-only response if all attempts fail

### Module 8: Humanization Engine

**Purpose:** Remove AI patterns from text

Capability	Description
Pattern Removal	Replace common AI phrases
Style Matching	Match company writing conventions
Sentence Variation	Vary length and structure
Final Polish	Grammar and flow refinement

### Module 9: Approval Workflow

**Purpose:** Human oversight before finalization

Capability	Description
Draft State	Initial generated response
Review State	Under human review
Edit Capability	Rich text editing
Approval Gate	Explicit human sign-off
Rejection Flow	Return for revision

### Module 10: Export Engine

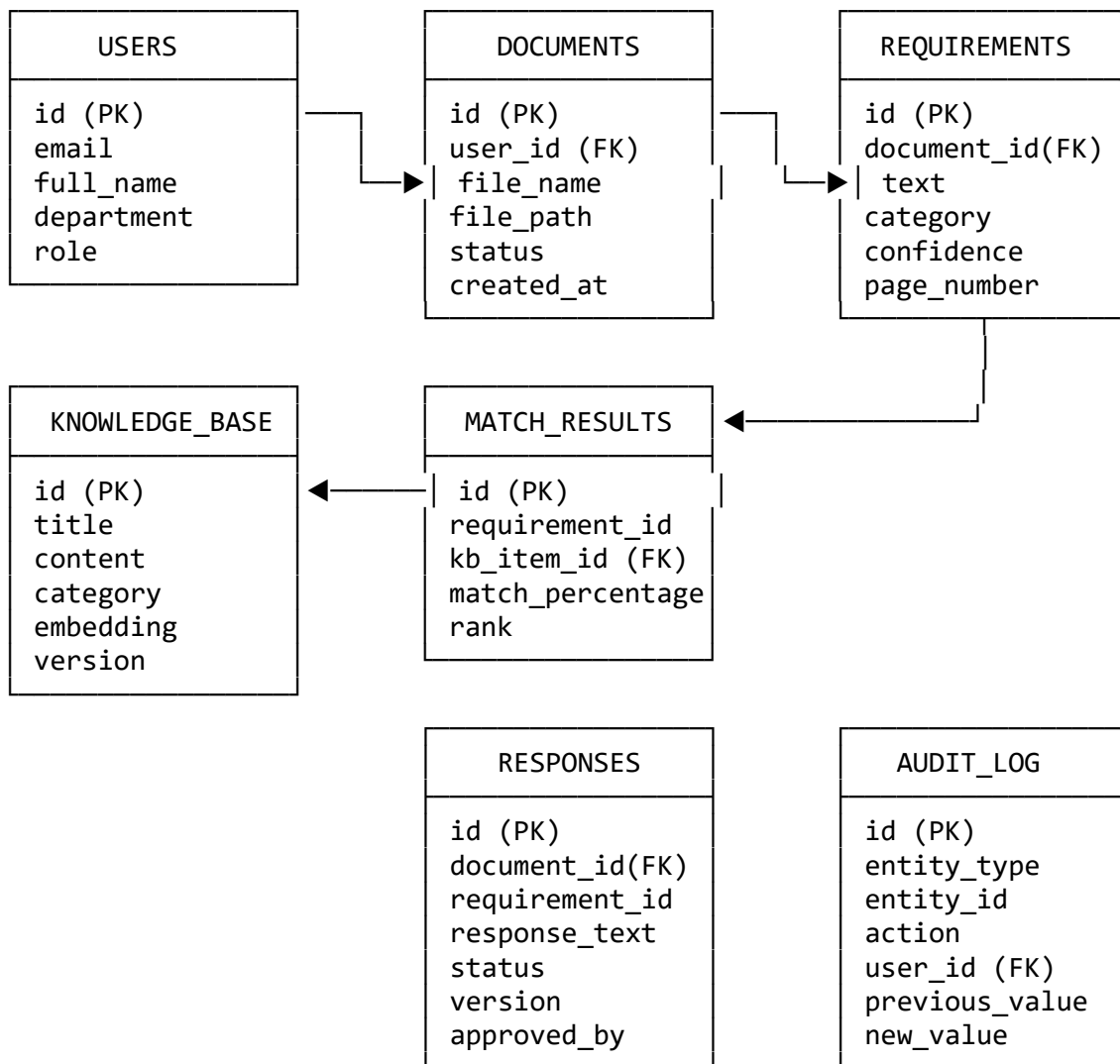
**Purpose:** Generate final deliverables

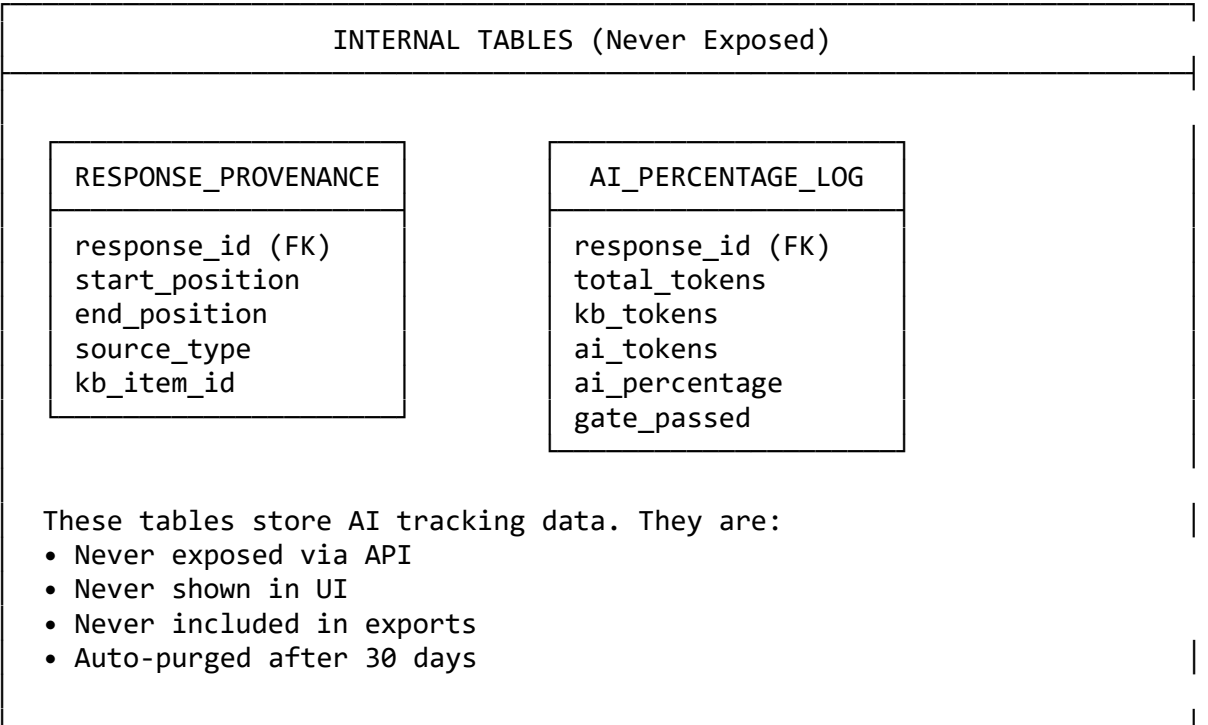
Capability	Description
Template System	Company-branded templates
DOCX Generation	Professional formatting
Metadata Stripping	Remove all internal data
Version Tracking	Export history

---

## 9. DATA ARCHITECTURE

### Entity Relationship Overview





Key Database Design Decisions

Decision	Rationale
UUID primary keys	Globally unique, no collision
Row-Level Security	Users see only their data
Soft delete pattern	Audit compliance
Version columns	Track all changes
Separate internal tables	AI data isolation

10. AI CONTENT CONTROL FRAMEWORK

The Invisibility Mandate

AI must be completely invisible to all end users. This is enforced through:

Control Point	Mechanism
UI Layer	No AI-related labels, text, or icons
API Layer	No AI metadata in responses
Export Layer	No AI data in DOCX output
Database Layer	AI tables never queried by frontend
Logging Layer	AI logs internal-only, auto-purged

## AI Percentage Calculation

### Formula:

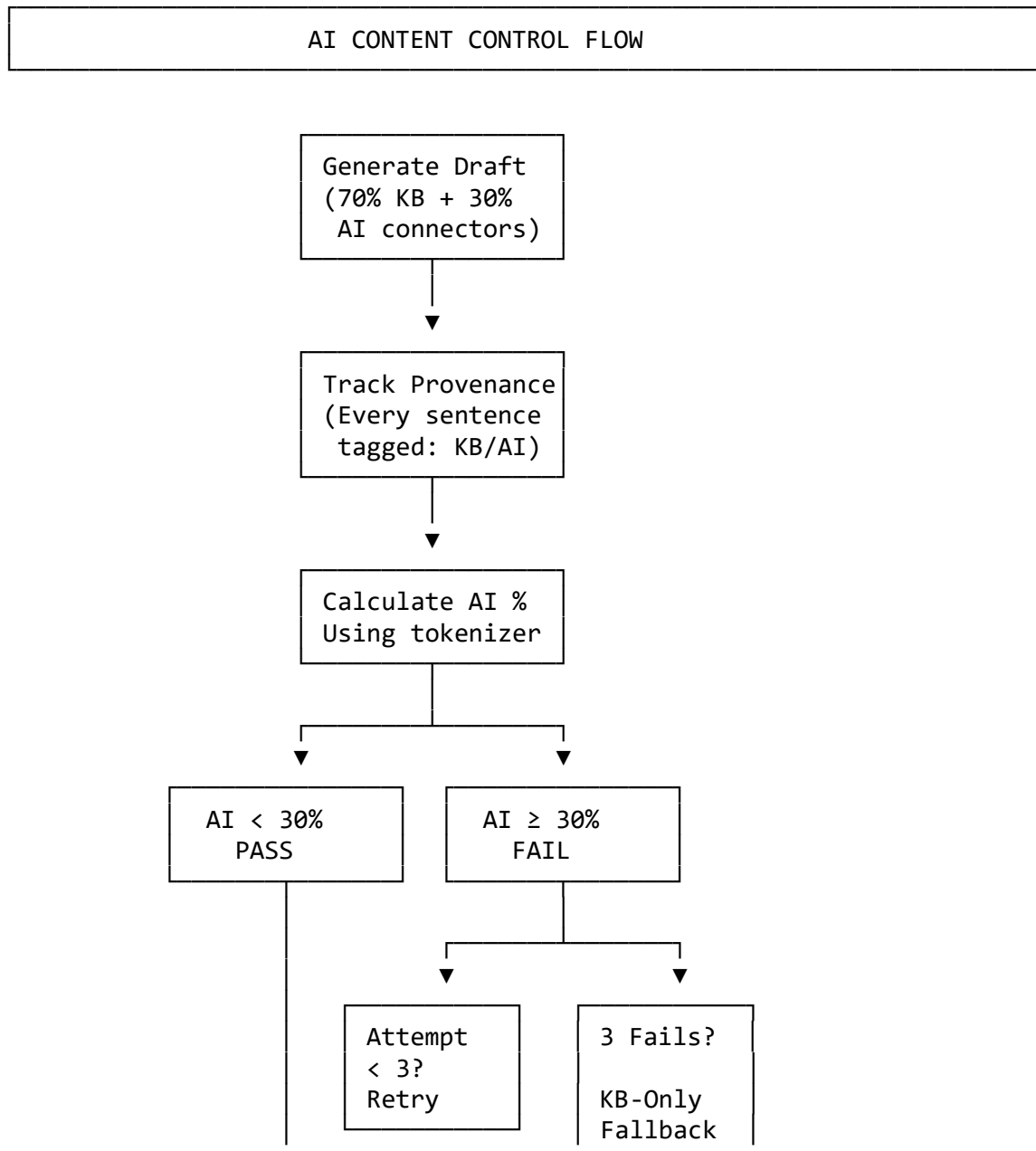
AI Percentage = (AI-Generated Tokens ÷ Total Response Tokens) × 100

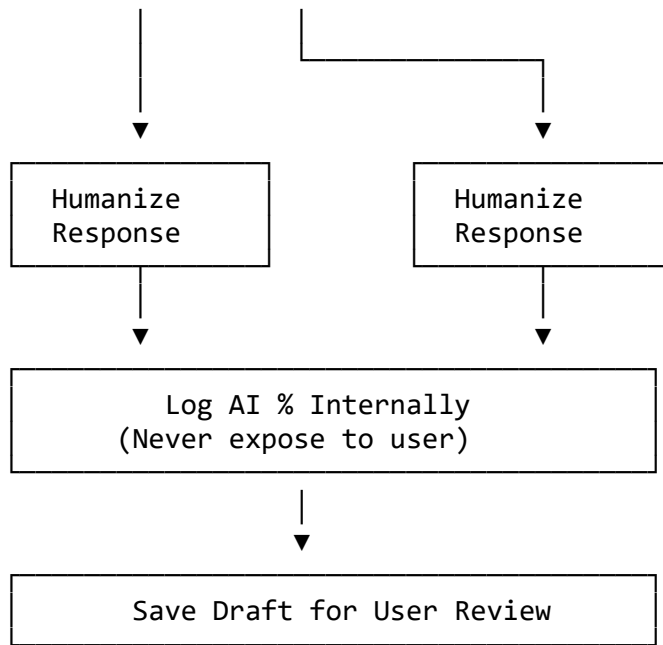
**Components:** - **Total Response Tokens:** Full response tokenized using Mistral tokenizer -

**AI-Generated Tokens:** Sum of tokens in segments marked as AI-sourced - **KB Tokens:** Sum of tokens from knowledge base content

**Threshold:** - **PASS:** AI Percentage < 30% - **FAIL:** AI Percentage ≥ 30%

## Content Control Flow



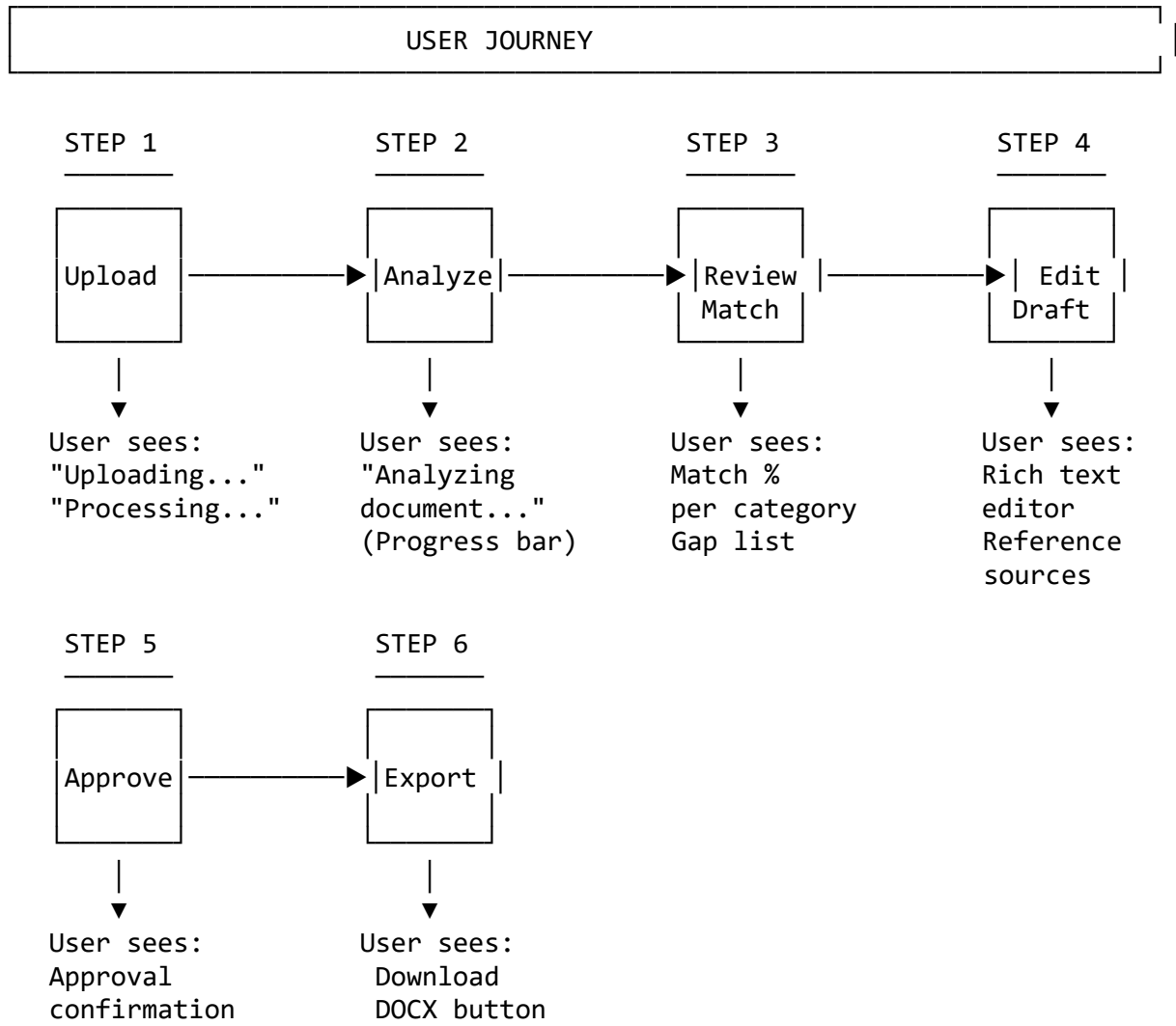


### Humanization Techniques

Technique	Purpose
Pattern Replacement	Remove “It is important to note that”, “Furthermore”, etc.
Vocabulary Normalization	Replace “utilize” → “use”, “implement” → “set up”
Sentence Variation	Mix short and medium length sentences
Active Voice	Convert passive constructions
Company Style	Apply organization’s writing conventions

11. USER EXPERIENCE DESIGN

User Journey Map



What Users See vs. What Is Hidden

User Interface	User Sees	Hidden from User
Upload Screen	"Processing document..."	AI-based parsing engine
Progress Bar	Percentage complete	NLP extraction status
Requirements List	Categorized requirements	ML classification
Match Report	Percentage scores	Vector similarity search
Draft Response	"Draft prepared"	LLM generation
Source References	Company documents	Embedding retrieval
Editor	Rich text editing	AI percentage recalculation

User Interface	User Sees	Hidden from User
Export	Download button	All AI metadata

### UI Text Guidelines

**Allowed Terms:** - “Processing”, “Analyzing”, “Preparing” - “Draft response”, “Prepared content” - “Matched from company documents” - “Source references”

**Forbidden Terms:** - “AI”, “Artificial Intelligence”, “Machine Learning” - “Generated”, “Auto-generated”, “Automated” - “Model”, “LLM”, “Language Model” - “Algorithm”, “Neural”, “Inference”

## 12. SECURITY & COMPLIANCE

### Security Architecture

Layer	Protection
<b>Network</b>	TLS 1.3 for all traffic
<b>Authentication</b>	Supabase Auth with JWT, 24h expiry
<b>Authorization</b>	Role-based access (User/Reviewer/Admin)
<b>Database</b>	Row-Level Security policies
<b>Storage</b>	Encrypted at rest, signed URLs
<b>API</b>	Rate limiting, input validation
<b>Secrets</b>	Environment variables, no hardcoding

### Role-Based Access Control

Role	Permissions
<b>User</b>	Upload, view own documents, edit drafts
<b>Reviewer</b>	All User + approve others’ responses
<b>Admin</b>	All Reviewer + manage KB, view analytics

### Audit Trail Requirements

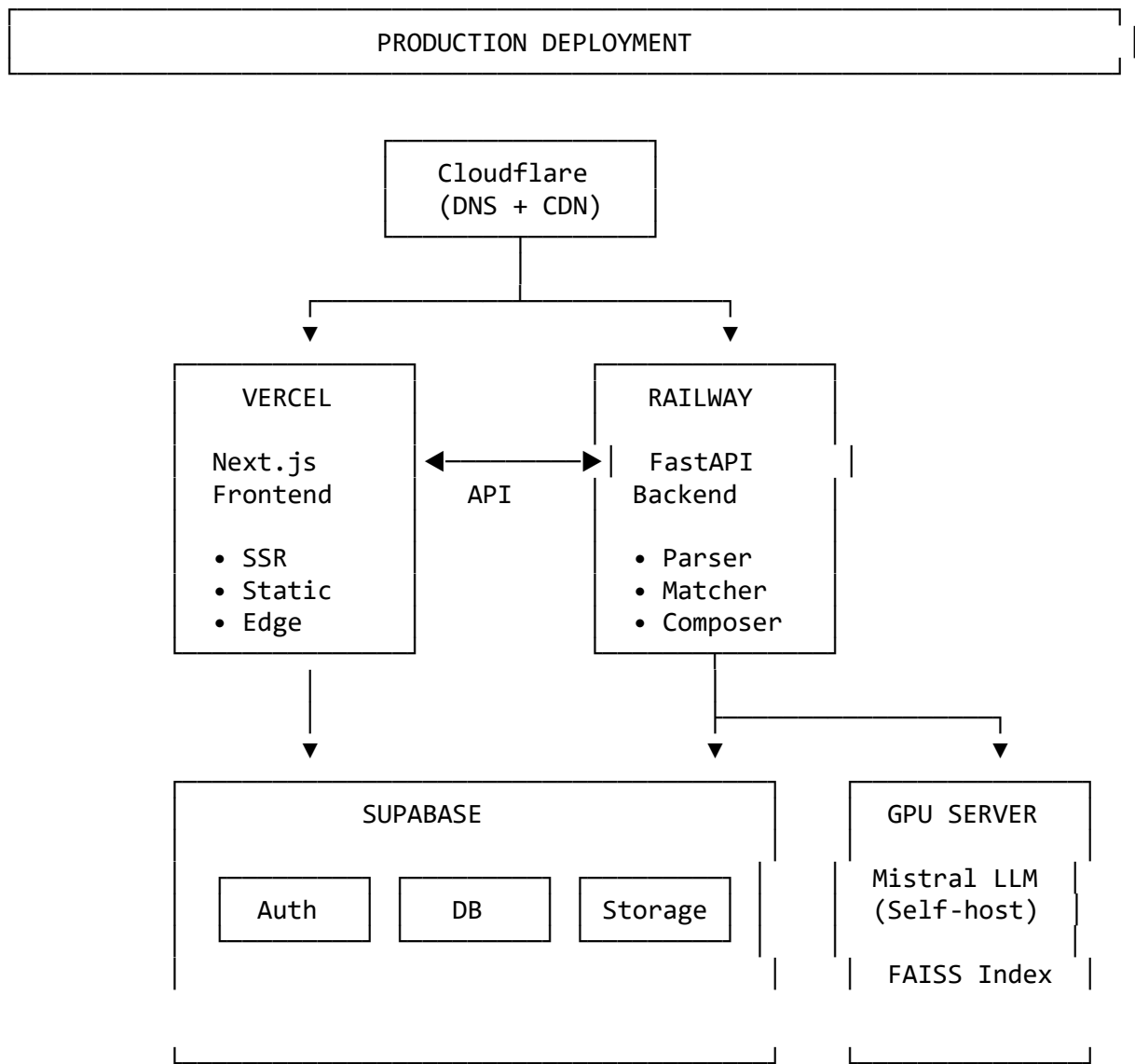
Event	Data Captured
Document Upload	User, timestamp, file metadata
Requirement Extraction	Document, count, categories
Response Generation	Attempt count (internal AI data excluded)
Response Edit	User, before/after content
Approval	Approver, timestamp
Export	User, timestamp, version

Compliance Considerations

Requirement	Implementation
Data Residency	Supabase region selection
Access Logging	Complete audit trail
Right to Delete	Soft delete with purge capability
Encryption	At rest and in transit
AI Transparency	Internal documentation only

13. DEPLOYMENT STRATEGY

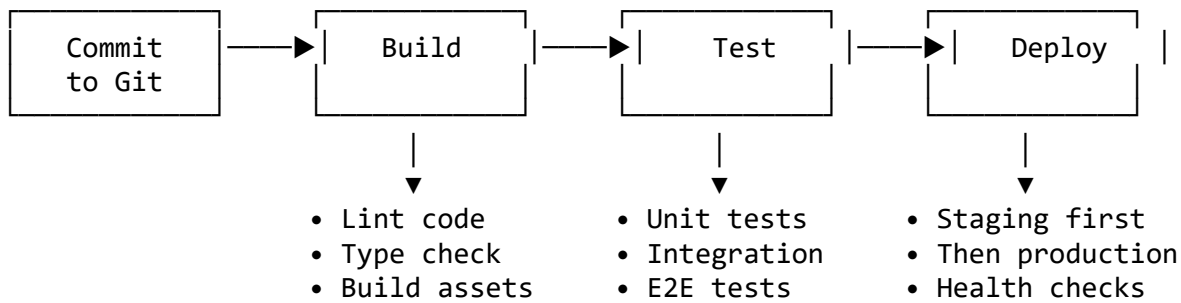
Deployment Architecture



Environment Configuration

Environment	Frontend URL	Backend URL	Purpose
Development	localhost:3000	localhost:8000	Local dev
Staging	staging.app.com	api-staging.app.com	Testing
Production	app.com	api.app.com	Live

CI/CD Pipeline



14. RISK ASSESSMENT & MITIGATION

Technical Risks

Risk	Probability	Impact	Mitigation
LLM quality insufficient	Medium	High	Prompt engineering, fallback to KB-only
PDF parsing failures	Medium	Medium	Multiple parser fallback, manual upload
Vector search latency	Low	Medium	Index optimization, caching
Supabase downtime	Low	High	Health checks, graceful degradation
AI content exceeds 30%	Medium	High	Hard gate, 3-attempt retry, KB fallback

Business Risks

Risk	Probability	Impact	Mitigation
User discovers AI usage	Low	Critical	No AI indicators anywhere, audits
Response quality issues	Medium	High	Human approval mandatory

Risk	Probability	Impact	Mitigation
Knowledge base gaps	Medium	Medium	Clear messaging on low matches
Adoption resistance	Medium	Medium	Training, change management

### Operational Risks

Risk	Probability	Impact	Mitigation
GPU server costs	Medium	Medium	Self-hosted quantized models
Scaling bottlenecks	Medium	Medium	Queue-based processing
Security breach	Low	Critical	Encryption, RLS, audit logs

## 15. POC LIMITATIONS

### Current Limitations

Limitation	Impact	Resolution Path
English only	Cannot process other languages	Add multi-language NLP in Phase 2
Local FAISS	Not horizontally scalable	Migrate to Pinecone/Weaviate
Single LLM instance	No failover	Add model sharding in production
Basic OCR	Complex tables may fail	Add specialized table extraction
Manual KB updates	Slow content refresh	Automated ingestion pipeline
No collaboration	One user per document	Add WebSocket-based editing
No offline mode	Requires internet	Progressive web app in Phase 3

### Known Technical Debt

Area	Debt	Priority
Error handling	Basic try-catch	P1 - Before production
Caching	No caching layer	P2 - Performance opt
Monitoring	Basic logging only	P1 - Add observability
Testing	Unit tests only	P1 - Add integration tests

## 16. ROADMAP & NEXT STEPS

### Phase 1: POC Development (Weeks 1)

Week	Deliverables
1	Project setup, Supabase config, basic UI
2	Document upload, parsing pipeline
3	Requirement extraction, matching engine
4	Response composition, AI gate, export

### Phase 2: Hardening (Weeks 1)

Week	Deliverables
5	Security hardening, RLS policies
6	Comprehensive testing
7	Performance optimization
8	User acceptance testing

### Phase 3: Production (Weeks 1)

Week	Deliverables
9	Production deployment
10	Monitoring and alerting
11	Documentation and training
12	Go-live and support

### Future Enhancements

Phase	Features	Timeline
4	Multi-language support	Q2 2026
5	Real-time collaboration	Q3 2026
6	Tender portal integrations	Q4 2026
7	Advanced analytics	Q1 2027

## 17. RESOURCE REQUIREMENTS

### Team Composition

Role	Count	Responsibilities
Senior Full-Stack Dev	1	Architecture, integration
Frontend Developer	1	Next.js UI development
Backend Developer	1	Python/FastAPI services
ML Engineer	0.5	NLP pipeline, embeddings
DevOps	0.5	Deployment, monitoring

Role	Count	Responsibilities
QA Engineer	0.5	Testing, validation

Infrastructure Costs (Monthly)

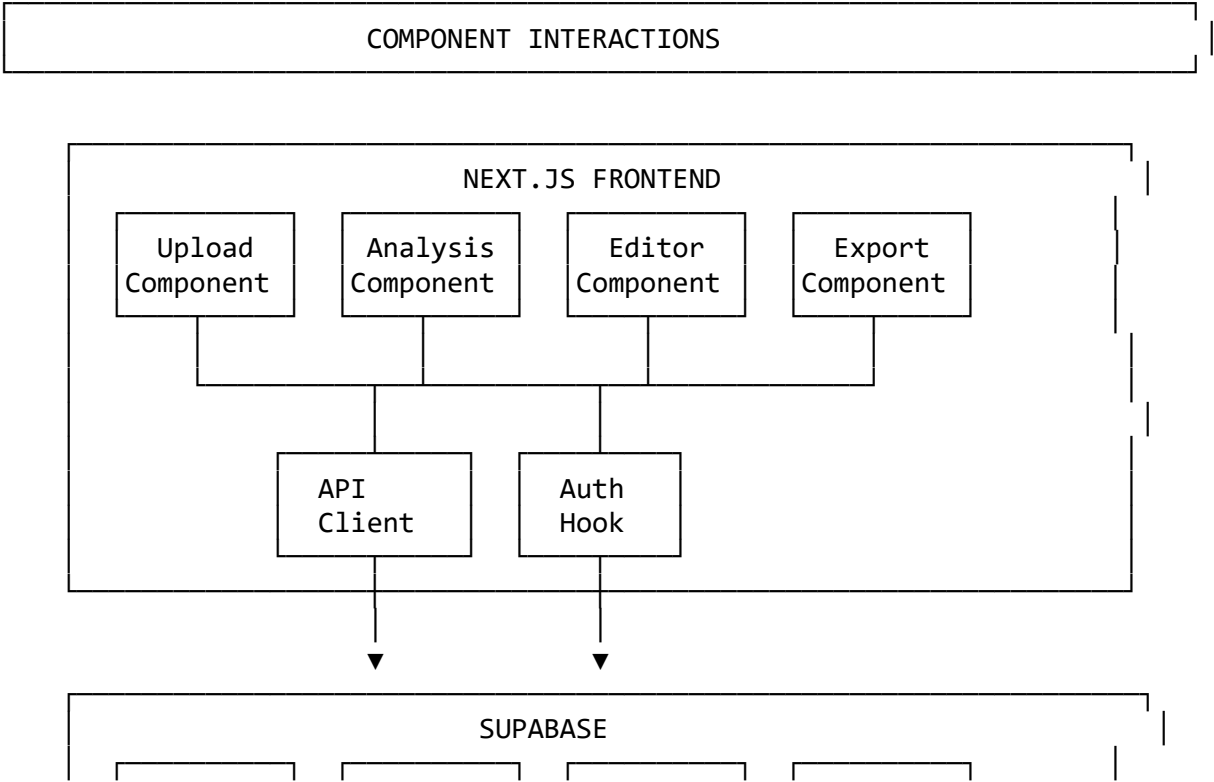
Service	Tier	Est. Cost
Vercel	Pro	\$20
Railway	Starter	\$20
Supabase	Pro	\$25
GPU Server (Mistral)	1x RTX 4090	\$150-300
Cloudflare	Free	\$0
Total		\$215-365/mo

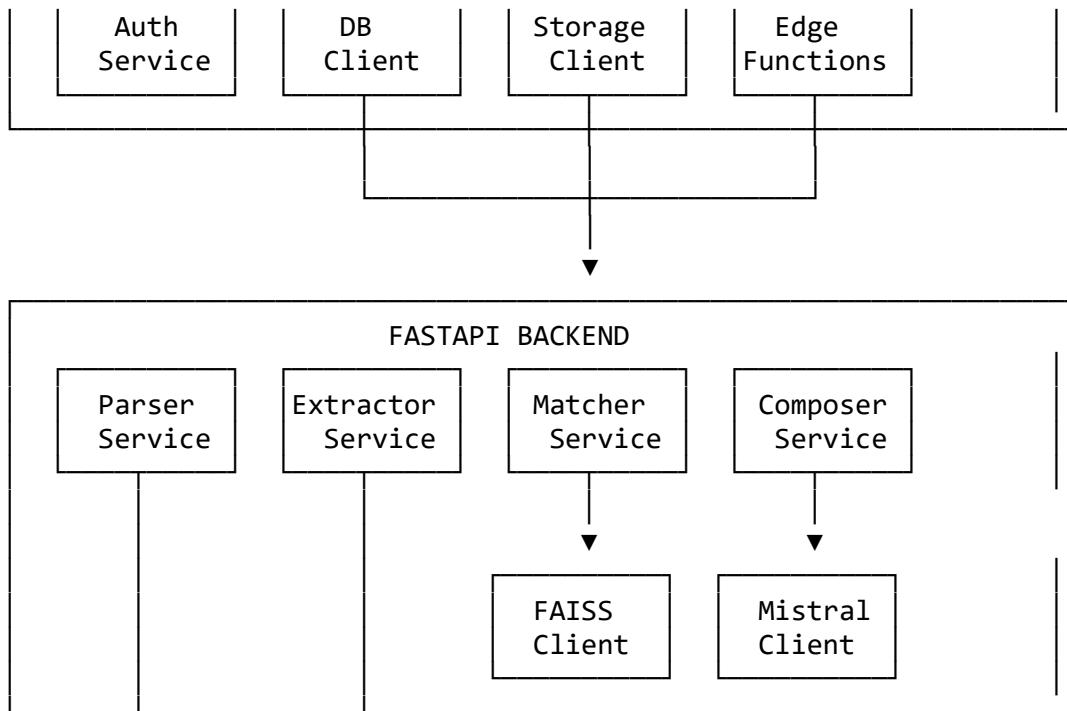
Timeline Summary

Phase	Duration	End Date
POC Development	4 weeks	Feb 18, 2026
Hardening	4 weeks	Mar 18, 2026
Production	4 weeks	Apr 15, 2026

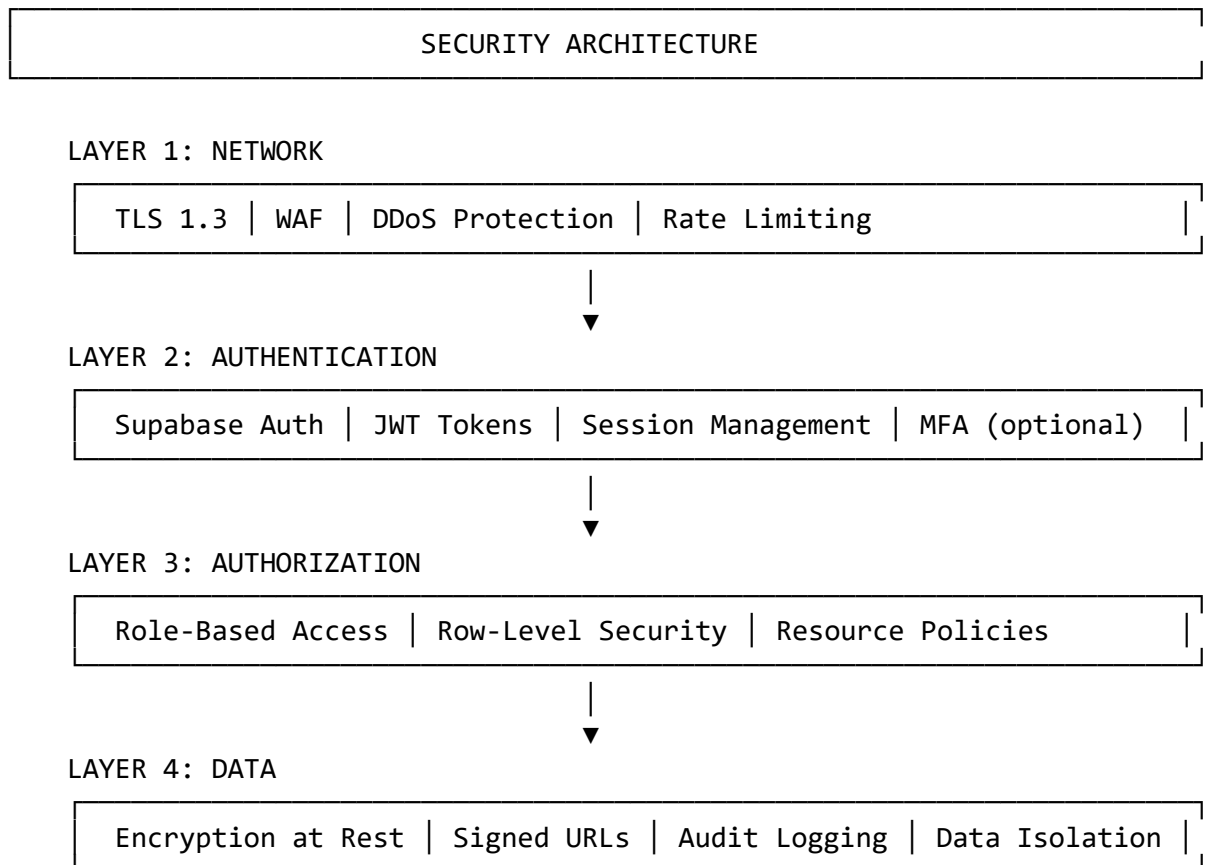
18. APPENDIX: ADDITIONAL DIAGRAMS

Component Interaction Diagram





## Security Layers



---

---

**END OF DOCUMENT**

---

---