RAG Pipeline PoC

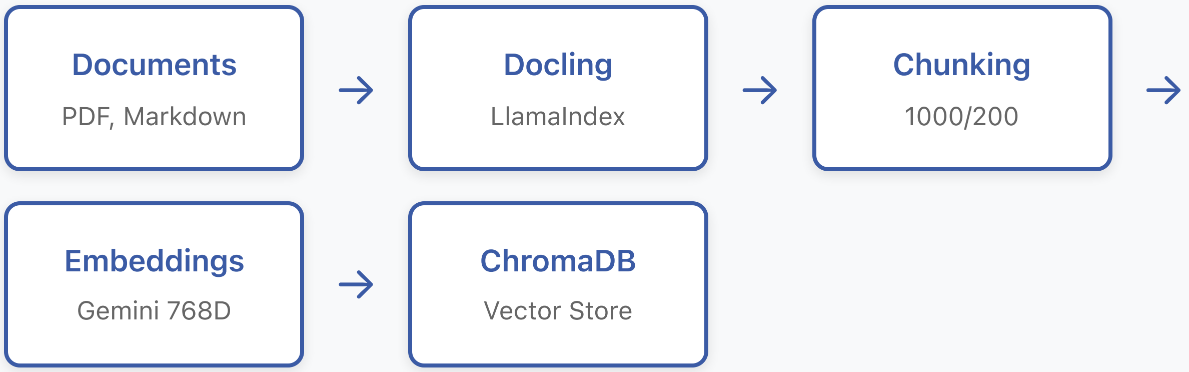
Retrieval-Augmented Generation System

AI Engineer Take-Home Assignment

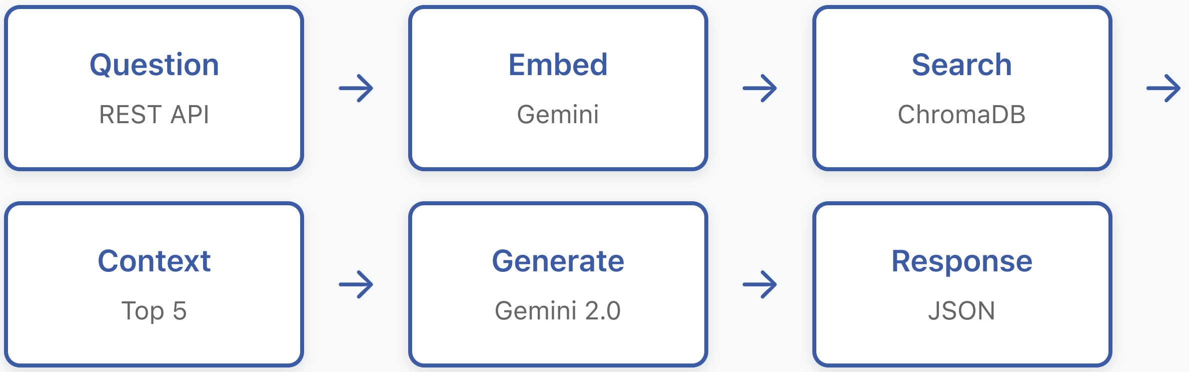
Daniel Boadzie | November 1, 2025

Architecture Overview

Data Ingestion Pipeline (Python)



Query API (Kotlin/Spring Boot)



Stack: LlamaIndex, Gemini text-embedding-004, Gemini 2.0 Flash, Spring Boot 3.2.1, ChromaDB, Docker Compose

Assumptions & Trade-offs

Key Assumptions

- Single-tenant deployment
- Small to medium corpus (under 10K docs)
- Gemini API availability with acceptable rate limits
- Trusted internal document sources
- English-language documentation

Critical for Production

Authentication, rate limiting, error handling, and monitoring must be added before production.

PoC Trade-offs for Speed

- **No authentication:** Open endpoints
- **Minimal error handling:** Basic try-catch only
- **No rate limiting:** Could overwhelm under load
- **Synchronous processing:** Blocking I/O
- **No test coverage:** Manual testing only
- **No response caching:** Every query hits LLM
- **Single database:** No replication

From PoC to Production

Performance Bottlenecks

1. LLM API Latency: 1-2s per query

2. Synchronous Processing: Blocking I/O

3. No Caching: Repeated LLM calls

4. Single Database: Limited throughput

Target: 1,000 req/min, 99.9% uptime

Production Roadmap

PostgreSQL + pgvector

Replace ChromaDB with pgvector for production-grade vector storage with ACID guarantees

Real-time RAG with Pathway

Implement Pathway framework for real-time document processing and live data synchronization

Caching + Async Processing

Redis caching (60-70% reduction) + Kotlin Coroutines for non-blocking I/O

Horizontal Scaling

Multiple API instances with load balancer + reranker integration

Est. Capacity: 1,200-1,500 req/min

Monitoring & Metrics

Service Health

Uptime: 99.9% SLA target

Error Rate: 5xx errors/min (alert if >1%)

Resources: Memory, CPU, GC pauses

Query Quality

Retrieval Accuracy: User feedback tracking

Relevance: Session duration, re-queries

Cache Hit Rate: Target >60%

User Satisfaction: Thumbs up/down

API Performance

Latency: P50, P95, P99 (P95 <3s)

Throughput: Requests per minute

Retrieval: ChromaDB latency (<200ms)

LLM Time: Gemini API (<2s)

Cost Management

API Costs: Token usage ($/1K queries)

Infrastructure: Storage, compute

Per Query: Target <$0.01/query

Stack: Prometheus, Grafana, Loki, AlertManager

Thank You

Questions?

Email: boadziedaniel43@gmail.com

Documentation: See README.md for setup instructions