Secure Cloud-Based RAG System

Internal AI Knowledge Management – Architecture, Security, Scaling, Cost, and Rollout

Company Objective: Provide employees with natural-language Q&A; over 10+ years of project data, with reliable citations and secure document handling.

1. Project Overview & Requirements

- Goal: Enable natural-language Q&A; on internal documents with reliable citations.
- Documents: PDFs, Word, PowerPoints, and emails stored across systems.
- Features: Real-time indexing, source citations, intuitive web + mobile chat UI.
- Security: Company-only access, document-level permissions, audit trails, US-only data residency.
- Constraints: ≤ $8,000/month cloud spend, 500 concurrent users, 99.5% uptime target.

2. Proposed Technology Stack (AWS-first)

Storage & Indexing

- Amazon S3 for raw and normalized document corpora (versioned).
- Amazon OpenSearch Serverless for hybrid keyword + vector search.
- Aurora PostgreSQL with pgvector extension for metadata registry.

Ingestion & Processing

- EventBridge + S3 notifications for document triggers.
- AWS Lambda and AWS Glue for parsing and normalization.
- Amazon Textract for OCR; AWS Comprehend for PII redaction.

Embeddings & Models

- Embeddings via Amazon Bedrock (Titan) or self-hosted E5-large on ECS/SageMaker.
- LLM generation via Bedrock (Claude or Llama 3) with citation guardrails.

API & Orchestration

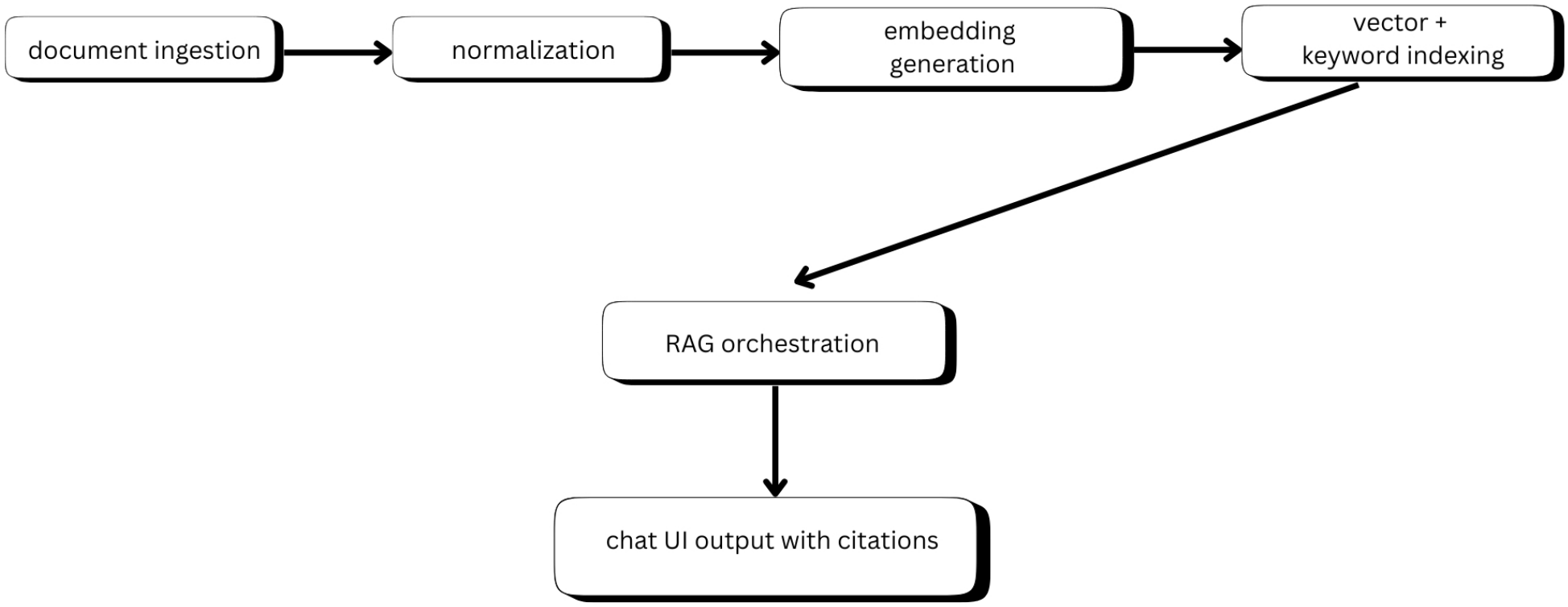
- ECS Fargate containerized RAG pipeline (LangChain or LlamaIndex).
- ElastiCache (Redis) for caching and throttling.

Frontend & AuthN/Z

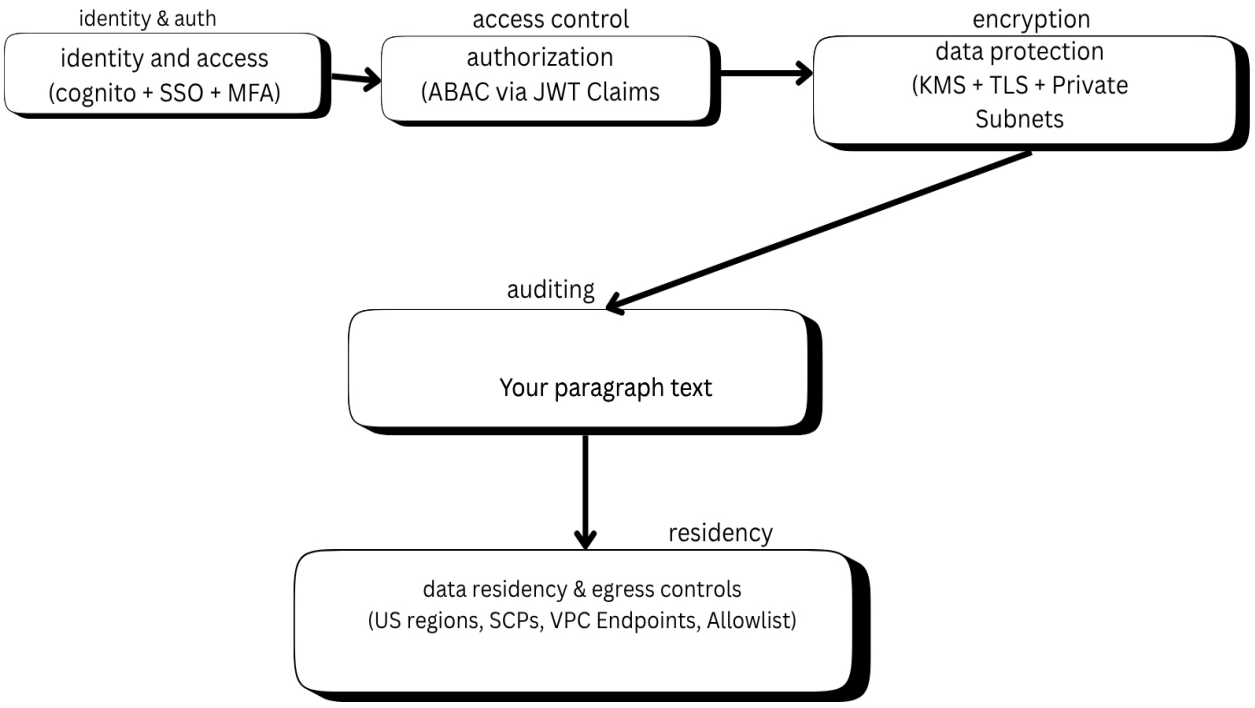
- Static web via S3 + CloudFront; mobile app via React Native/Flutter.
- Amazon Cognito (SAML/OIDC integration to Okta/Azure AD).

3. RAG Architecture (Data Flow)

Data Pipeline steps



4. Security Architecture



5. Scaling Strategy (500 Concurrency & Growth)

- ECS Fargate with autoscaling (CPU and request rate).
- Shard and replicate OpenSearch collections; cache frequently accessed results in Redis.
- Periodic re-embedding for growing corpus; reranking for precision.
- Use CloudFront and SSE/WebSockets for scalable frontends.
- Enable multi-AZ deployment with backup and disaster recovery setup.

6. Cost Strategy (≤ $8,000/month Target)

Component	Estimated Monthly Cost (USD)
OpenSearch Serverless (vector + search)	$2,500 – $3,000
ECS Fargate RAG API	$600 – $1,000
S3 Storage (5–10 TB)	$125 – $300
Textract OCR + Parsing	$500 (initial months higher)
Embeddings (Bedrock/Self-hosted)	$800 – $1,500
LLM Inference (Claude/Llama3)	$1,000 – $1,800
ElastiCache + Aurora	$400 – $700
CloudFront + Cognito + Misc.	$300 – $600
Total Range	$6,200 – $9,400

7. Implementation Phases

Phase 0: Foundations (Week 1–2)

- Set up VPC, IAM, SCPs, CI/CD, Cognito + SSO

Phase 1: MVP (Week 3–6)

- Ingest PDFs/Word/PPT; chunking + embeddings; OpenSearch index; basic chat UI.

Phase 2: Hardening & Scale (Week 7–10)

- Add ABAC filters; OCR; reranking; Redis cache; autoscaling.

Phase 3: UX & Mobile (Week 11–14)

- Enhance UI/UX; mobile app; saved queries; feedback dashboard.

Phase 4: Optimization (Ongoing)

- Distillation; evaluation suite; cost optimization; DR testing.

Disclaimer: the part 6 - 7 are just my rough estimates. They can actually change on actual execution of this proposal.