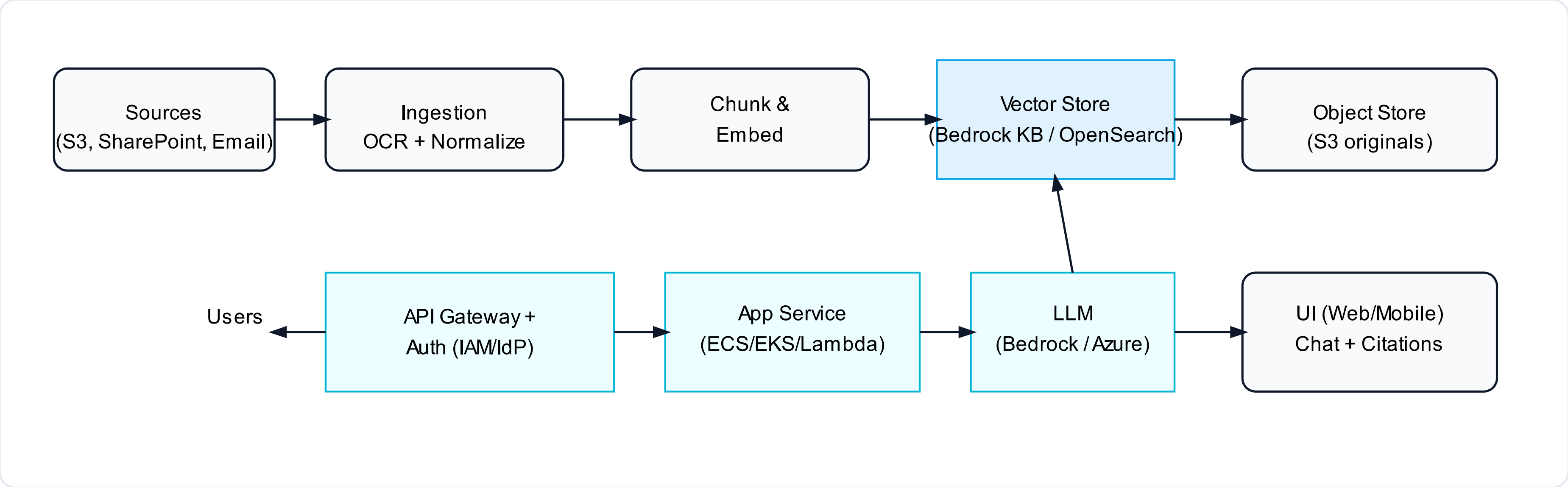
Cloud Test – RAG Architecture

Secure, scalable Retrieval-Augmented Generation platform for a 500-person consultancy with 10+ years of documents. All data and services run in US regions, deliver 99.5% uptime, and respect document-level permissions.

1) Assumptions

- Corpus: ~12 TB across PDFs/Office/email; +100 GB/month growth; 30% OCR needed.
- Queries: 500 peak concurrent, median 6 queries/min/user during peak launches.
- Updates: Daily batches; critical content can be pushed within 15 minutes.
- Stack familiarity: AWS managed services; Terraform for IaC; GitHub Actions CI/CD.

2) High-Level Architecture



3) Ingestion & Indexing Pipeline

- Collectors: S3 event triggers, SharePoint/Exchange connectors (Graph API), secure SCP drop for legacy shares.
- Conversion: OCR via Textract; normalize to PDF/text; extract metadata (owner, ACLs, system of record).
- Chunking: 500–1,000 tokens with overlap; store chunk-to-doc map.
- Embedding: Managed model (Bedrock Titan v2 or Azure Text Embedding 3 Large) via batch jobs.
- Storage: Embeddings to vector DB (OpenSearch w/ kNN or Bedrock Knowledge Base on top of Aurora/S3); raw+normalized to S3 with bucket keys matching ACLs.
- Index refresh: Daily bulk + event-driven micro-batches; DLQ + retries; lineage recorded in DynamoDB/Aurora.

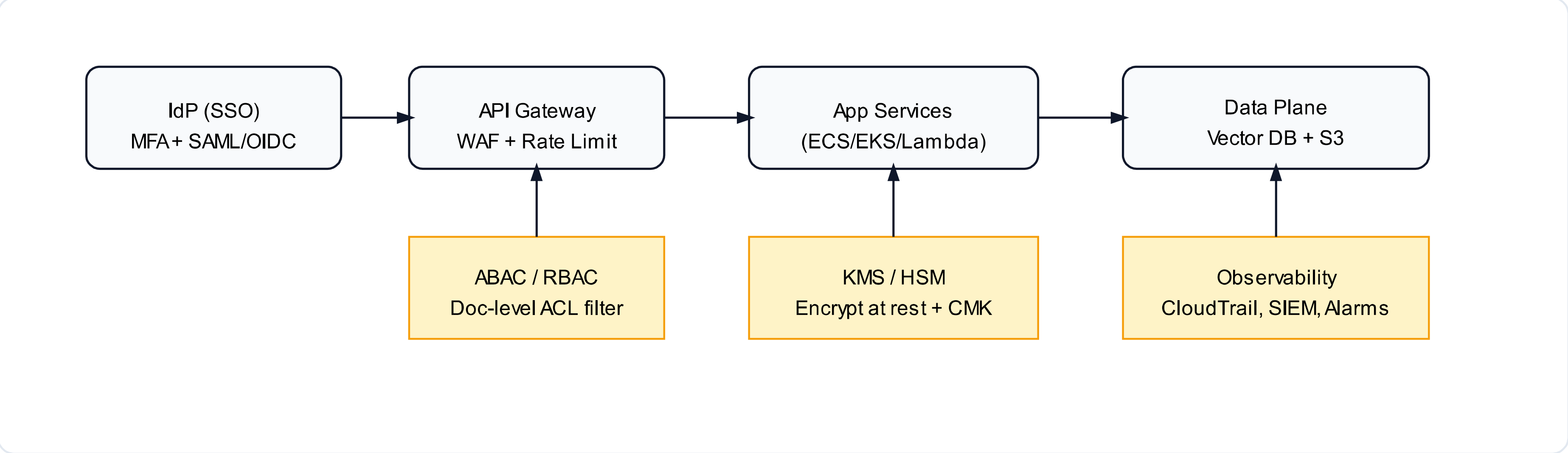
4) RAG Retrieval & Response

- AuthN at gateway (OIDC/SAML to corporate IdP); authZ filter applies user’s document ACLs before retrieval.
- Query flow: embed user query → vector search (k=20) → semantic re-rank (top 5) → policy filter (classify PII/secrets) → construct prompt with citations → LLM generation with grounding guardrails.
- Citations: each chunk carries doc ID, path, owner; surfaced as inline numbered links.
- Feedback loop: thumbs up/down stored; low-confidence answers trigger auto-retrieval tuning.

5) UI + App Layer

- Web (React) and mobile shell hitting a single API Gateway; chat UX with history pinned per user.
- Backend: ECS/Fargate or EKS for stateless APIs; async workers (SQS/SNS) for ingestion and re-embedding.
- Rate limiting + WAF at the edge; token-based session to backend; streaming responses to reduce latency.

6) Security Architecture



- All traffic through WAF+Gateway; private subnets for app + vector DB; VPC endpoints for S3.
- SSO + MFA; short-lived tokens; fine-grained ABAC using document metadata.
- Encryption: KMS CMKs for S3, EBS, OpenSearch/Aurora; TLS everywhere; secrets in Secrets Manager.
- Monitoring: CloudTrail, GuardDuty, Config, SIEM forwarding; alarms on anomalous query volume or vector scans.
- Data residency: single-region primary (us-east-1/us-west-2) with cross-region backups; no cross-border egress.

7) Scaling Strategy

- Stateless APIs on Fargate/EKS with auto-scaling by QPS/CPU; provisioned LLM endpoints with burstable tier.
- Vector DB: sharded OpenSearch domain with UltraWarm for colder vectors; adaptive k (10-50) based on latency.
- Ingestion: parallel workers; backpressure via SQS; nightly bulk for history, micro-batch for deltas.
- Caching: query+embedding cache (Redis/Memcached); CDN for static assets.

8) Cost Strategy (target <$8k/mo)

- Use managed services: Bedrock/Azure OpenAI pay-per-use; OpenSearch medium cluster with auto-tune; Fargate spot for workers.

- Reserve baseline for API/DB, auto-scale for spikes; lifecycle S3 to infrequent access for cold docs.
- Batch embeddings (night) to use discounted compute; monitor top queries to tune k and limit context size.

9) Risks, Tradeoffs, Alternatives

- Latency vs. grounding: smaller k reduces latency but risks misses; mitigated with re-rank and dynamic k.
- Vendor lock-in: Bedrock/OpenAI managed ease vs. self-hosted models; note migration plan to self-hosted on EKS if costs rise.
- OCR quality: may degrade retrieval; add confidence thresholds and human review for low-confidence batches.
- Security drift: rely on IaC, SCPs, and continuous controls testing; periodic permission recertification.

Implementation Phases

Phase 1 (0-4 wks):

- Core ingestion to S3 + embeddings to vector DB.
- Basic chat UI, API Gateway, SSO, RBAC filter.
- MVP LLM grounding with citations; monitoring baseline.

Phase 2 (5-8 wks):

- Re-rank + policy classifier, feedback loop.
- Autoscaling, caching, cost dashboards; backup/DR.
- Mobile shell; batch + delta indexing automation.

Phase 3 (9-12 wks):

- Advanced relevancy tuning; domain adapters.
- Offline eval harness; red-team prompts; periodic pentest.
- Multi-region DR ready; optimize storage tiers.