Technical Implementation Deep-Dive Report

GlobalTech Financial Services

Industry:	Financial Technology
Assessment Type:	Technical Implementation
Assessment Date:	2025-10-08
Current State:	Pilot Phase
Tech Stack:	

Prepared for:

GlobalTech Financial Services

Prepared by:

Cloud202 Technical Architecture Team

CONFIDENTIAL - GlobalTech Financial Services Technical Assessment

Report Generated: October 17, 2025

Current State Assessment

GlobalTech Financial Services operates a hybrid infrastructure with on-premises portfolio management systems (BlackRock Aladdin) and cloud-based CRM (Salesforce), creating a fragmented architecture that impedes real-time analysis capabilities. The current topology consists of on-premises data centers hosting 150TB of historical portfolio data with limited horizontal scaling, connected via Direct Connect to AWS for pilot GenAI workloads. The pilot deployment runs on EC2 instances (m5.2xlarge) in us-east-1, utilizing OpenAI GPT-4 and Claude 3 Sonnet APIs with basic request/response patterns. Baseline performance metrics reveal significant bottlenecks: average client analysis requires 5.2 hours with 85% accuracy, system response times during market volatility exceed 30 seconds, and the infrastructure supports only 50 concurrent users before degradation. The pilot environment lacks formal SLAs, operates with 95% uptime (target 99.9%), and experiences frequent timeout errors during peak market hours (6:30-10:00 AM ET) when Bloomberg Terminal data ingestion spikes to 50,000 messages per second.

Performance bottlenecks manifest across multiple layers. Database queries against the monolithic PostgreSQL instance (on-premises) exhibit P95 latencies exceeding 8 seconds during concurrent advisor access, with connection pool exhaustion occurring at 200 simultaneous queries. Network I/O constraints emerge from the 1Gbps Direct Connect link, which saturates during real-time market data synchronization, causing 15-20 second delays in data availability. The pilot LLM integration lacks caching mechanisms, resulting in redundant API calls costing $12,000 monthly for just 10 advisors—a cost structure that extrapolates to $600,000 monthly at full scale, far exceeding the $200,000 target. Memory pressure on application servers peaks at 92% utilization during document processing tasks (100+ page PDFs), triggering garbage collection pauses that compound latency issues. The absence of asynchronous processing patterns forces synchronous workflows, blocking advisor interfaces during long-running portfolio optimization calculations.

Security and compliance posture reveals critical gaps for production deployment. The pilot environment implements basic TLS 1.2 encryption and API key authentication but lacks comprehensive audit logging, with only 30-day CloudWatch retention versus the required 7-year regulatory retention. PII handling remains manual, without automated detection or tokenization, creating GDPR and SEC compliance risks. Identity management relies on disparate systems—Active Directory for internal users, Okta for external advisors, and API keys for system integration—without unified IAM policies or least-privilege enforcement. Secrets management uses hardcoded credentials in configuration files rather than AWS Secrets Manager or Parameter Store. The current logging architecture captures application logs but misses critical security events: API access patterns, data exfiltration attempts, and privilege escalation indicators. SOC 2 Type II certification requirements mandate tamper-evident logging with blockchain verification, which is absent. The pilot lacks Web Application Firewall (WAF) protection, DDoS mitigation, or input validation against prompt injection attacks, exposing the system to adversarial exploitation.

Data architecture constraints severely limit scalability and analytical capabilities. The 200TB structured data and 50TB unstructured documents reside in siloed systems: portfolio data in on-premises Oracle databases, client interactions in Salesforce, and research documents in SharePoint, with no unified data lake or catalog. Data quality assessment reveals 87% accuracy with known gaps in emerging markets coverage (15% of portfolios), incomplete alternative investment data, and demographic bias toward high-net-worth clients. The absence of a feature store or ML-ready data pipelines forces manual data preparation, consuming 40% of data science team capacity. Real-time data ingestion from Bloomberg and Reuters lacks stream processing capabilities, relying on batch ETL jobs with 15-minute latency—insufficient for volatility-driven recommendations. Vector database evaluation remains incomplete, with the Pinecone pilot storing only 2 million embeddings versus the required 50 million for full research corpus coverage.

Organizational and operational constraints compound technical challenges. The 8-person engineering team lacks deep GenAI expertise, with only 2 engineers experienced in LLM orchestration and prompt engineering. Change management processes require 4-week lead times for production deployments, incompatible with the 3-6 month delivery timeline. Vendor dependencies create lock-in risks: BlackRock Aladdin integration requires proprietary APIs with 500ms baseline latency, and Bloomberg Terminal licensing costs $24,000 per user annually. The absence of Infrastructure as Code (IaC) practices results in manual provisioning taking 2-3 weeks for new environments. Disaster recovery capabilities are limited to daily backups with 24-hour RPO and 48-hour RTO, failing to meet the 99.9% availability requirement. Cost visibility remains opaque, with no tagging strategy or FinOps practices to track GenAI inference costs, data transfer fees, or compute utilization by business unit.

Target Architecture Design

The target architecture implements a multi-region, highly available AWS infrastructure leveraging a hub-and-spoke VPC topology with dedicated VPCs for production (us-east-1 primary, us-west-2 DR), non-production (us-east-1), and shared services (Transit Gateway for centralized networking). The production VPC spans three Availability Zones with public subnets (ALB, NAT Gateway), private subnets (ECS Fargate tasks, Lambda functions), and isolated subnets (Aurora PostgreSQL, ElastiCache Redis), enforcing strict network segmentation via Security Groups and NACLs. Internet-facing traffic flows through Application Load Balancer (ALB) with AWS WAF rules for OWASP Top 10 protection, rate limiting (10,000 requests per minute per IP), and geo-blocking for non-US/EU regions. AWS Shield Advanced provides DDoS mitigation with 24/7 response team engagement. CloudFront CDN caches static assets and API responses with 300-second TTL, reducing origin load by 60% and improving P95 latency to sub-500ms for global advisor access.

The compute layer adopts serverless-first architecture with Amazon ECS Fargate for stateless API services (portfolio analysis, recommendation engine) and AWS Lambda for event-driven workflows (document processing, notification dispatch). The core API service runs on Fargate tasks (4 vCPU, 8GB RAM) with Application Auto Scaling policies targeting 70% CPU utilization, scaling from 20 to 200 tasks based on CloudWatch metrics. Lambda functions handle asynchronous processing: S3-triggered document ingestion (15-minute timeout, 10GB memory), EventBridge-scheduled batch jobs (nightly portfolio valuation), and SQS-driven recommendation generation (visibility timeout 5 minutes, DLQ after 3 retries). Amazon Bedrock provides managed LLM access with Claude 3.5 Sonnet for financial analysis (on-demand pricing) and Claude 3 Haiku for high-volume summarization tasks (provisioned throughput for cost optimization). Bedrock Guardrails enforce content filtering policies: PII redaction, prompt injection detection, and regulatory keyword blocking with custom deny lists. Amazon Kendra powers enterprise search across the 25TB research database, with custom document enrichment pipelines extracting entities, sentiment, and financial metrics.

Data plane architecture separates hot, warm, and cold storage tiers optimized for access patterns and cost. Amazon S3 serves as the unified data lake with Intelligent-Tiering for automatic lifecycle management: frequently accessed market data (20TB) in S3 Standard, weekly-accessed client documents (100TB) transitioning to S3 Standard-IA after 30 days, and archival compliance data (150TB) in S3 Glacier Flexible Retrieval with 7-year retention policies. Amazon Aurora PostgreSQL Serverless v2 (ACU range 8-128) hosts transactional data with Multi-AZ deployment, automated backups (35-day retention), and read replicas for reporting workloads. Amazon DynamoDB stores user sessions, feature flags, and real-time portfolio positions with on-demand capacity mode, point-in-time recovery enabled, and global tables for cross-region replication (RPO <1 second). Amazon ElastiCache for Redis (r6g.2xlarge cluster mode enabled, 6 shards, 2 replicas per shard) caches LLM responses (TTL 3600 seconds), embedding vectors for semantic search, and frequently accessed portfolio data, achieving 95%

cache hit ratio and reducing database load by 70%.

Vector database implementation uses Amazon OpenSearch Service with k-NN plugin for semantic search across 50 million document embeddings (1536 dimensions from text-embedding-3-large). The OpenSearch cluster (r6g.xlarge.search data nodes, 6-node configuration) implements index sharding by document type (research reports, client portfolios, regulatory filings) with replica count 2 for high availability. Embedding generation pipeline uses AWS Batch for bulk processing (10,000 documents per job) and Lambda for real-time ingestion, with embeddings stored in OpenSearch and metadata in Aurora for hybrid search capabilities. RAG implementation follows a two-stage retrieval pattern: initial semantic search returns top 50 candidates (P95 latency 200ms), followed by reranking via cross-encoder model on SageMaker endpoint (ml.g5.xlarge) to select top 5 context documents (P95 latency 150ms). Amazon Bedrock Agents orchestrate multi-step reasoning workflows, invoking Lambda functions for portfolio calculations, market data retrieval from Amazon Timestream, and compliance checks via Step Functions state machines.

CI/CD pipeline implements GitOps principles with AWS CodePipeline, CodeBuild, and CodeDeploy, orchestrating infrastructure provisioning via Terraform (v1.6+) and application deployment via ECS task definitions. The pipeline enforces security gates: Checkov for IaC scanning, Trivy for container vulnerability assessment, and custom Lambda validators for compliance policy verification (encryption at rest, VPC endpoints, IAM least privilege). Environment strategy follows immutable infrastructure patterns with dedicated AWS accounts per environment (dev, staging, prod) managed via AWS Organizations and AWS Control Tower guardrails. Feature flags via AWS AppConfig enable progressive rollout with targeting rules by advisor cohort, supporting canary deployments (5% traffic for 2 hours) and blue/green cutover with automated rollback on CloudWatch alarm breach (error rate >1%, P95 latency >3 seconds). Disaster recovery architecture achieves RTO 1 hour and RPO 5 minutes through Aurora Global Database (cross-region replication lag <1 second), S3 Cross-Region Replication (CRR) for data lake, and Route 53 health checks with automatic failover to us-west-2 region. Monthly DR drills validate runbooks and recovery procedures, with automated testing via AWS Fault Injection Simulator (FIS) for chaos engineering experiments.

Data Strategy

Data governance framework implements AWS Lake Formation for centralized access control, with data classification tags (Public, Internal, Confidential, Restricted) applied at S3 object and Glue Catalog table levels. Lake Formation blueprints automate ingestion from source systems: incremental CDC from Aurora via DMS, full snapshots from Salesforce via AppFlow (daily sync), and streaming market data from Bloomberg via Kinesis Data Streams (500,000 records per second throughput). AWS Glue Data Catalog serves as the central metadata repository with 1,200+ registered tables, enforcing schema evolution policies (backward compatibility required, breaking changes trigger approval workflow). Data lineage tracking via AWS Glue DataBrew and custom Lambda functions captures transformation logic, enabling impact analysis for regulatory audits and root cause investigation. Data quality framework uses AWS Glue Data Quality rules (completeness >95%, uniqueness for client IDs 100%, timeliness <15 minutes for market data) with automated alerts to SNS topics and quarantine workflows for failed records.

Storage architecture implements multi-tier strategy optimized for access patterns and compliance requirements. Hot tier (S3 Standard, 20TB) stores real-time market data, active client portfolios, and current research with 99.99% availability SLA and sub-100ms GET latency. Warm tier (S3 Standard-IA, 100TB) houses historical analysis, archived client communications, and regulatory reports with lifecycle transition after 30 days and retrieval latency <3 seconds. Cold tier (S3 Glacier Flexible Retrieval, 150TB) archives compliance data with 7-year retention, supporting bulk retrieval (5-12 hours) for audit requests. Encryption strategy mandates AES-256 server-side encryption with AWS KMS customer-managed keys (CMK), separate keys per data classification level with automatic rotation every 90 days. Field-level encryption via client-side encryption libraries protects PII (SSN, account numbers, addresses) before S3 upload, with key material stored in AWS Secrets Manager and access logged to CloudTrail. Tokenization service on Lambda replaces sensitive identifiers with surrogate keys stored in DynamoDB, enabling analytics on pseudonymized data while maintaining referential integrity.

Data ingestion pipelines support three patterns: batch, micro-batch, and streaming. Batch ingestion via AWS Glue ETL jobs (Python Shell, 10 DPU allocation) processes daily portfolio valuations, client document uploads, and research database updates with incremental load strategies (high-water mark tracking in DynamoDB). Micro-batch processing via Glue Streaming jobs (5-minute tumbling windows) aggregates market events, advisor activity logs, and system metrics for near-real-time dashboards. Streaming ingestion via Kinesis Data Streams captures Bloomberg Terminal feeds, client portal interactions, and API requests with Kinesis Data Firehose delivering to S3 (Parquet format, Snappy compression) and OpenSearch for real-time search. Data partitioning strategy uses Hive-style partitions (year/month/day/hour) for time-series data and hash partitioning by client_id for portfolio data, enabling partition pruning that reduces query scan volume by 90%. Compaction jobs via AWS Glue consolidate small files (target 128MB per file) to optimize S3 LIST operations and Athena query performance.

Retrieval-Augmented Generation (RAG) implementation leverages a three-layer architecture: embedding generation, vector indexing, and context retrieval. Embedding pipeline uses Amazon Bedrock Titan Embeddings (1536 dimensions) for general text and custom fine-tuned FinBERT embeddings (768 dimensions) deployed on SageMaker for financial domain specificity. Hybrid embedding strategy combines both models via weighted fusion (0.7 Titan, 0.3 FinBERT) to balance semantic understanding and domain accuracy. Vector indexing in OpenSearch uses HNSW algorithm (M=16, ef_construction=128) with index refresh interval 30 seconds, supporting 10,000 queries per second with P95 latency <200ms. Metadata filtering enables faceted search by document type, publication date, asset class, and regulatory category without full vector scan. Context retrieval implements maximal marginal relevance (MMR) algorithm to balance relevance and diversity in top-K results, reducing redundancy in LLM context windows. Caching layer in ElastiCache stores embedding vectors for frequently accessed documents (TTL 86400 seconds) and precomputed similarity scores for common queries, achieving 80% cache hit ratio and reducing embedding API costs by $15,000 monthly. Data access policies enforce row-level security via Lake Formation with IAM principal tags matching advisor specialization (equities, fixed income, alternatives) to client portfolio asset classes, ensuring advisors access only relevant data subsets.

Model Evaluation Recommendations

Model evaluation framework implements continuous validation across offline, online, and human-in-the-loop dimensions with quantitative acceptance criteria. Offline evaluation uses golden test sets (5,000 labeled examples per use case: portfolio analysis, risk assessment, market commentary) with monthly refreshes incorporating recent market conditions and advisor feedback. Evaluation metrics include exact match accuracy (target >90%), semantic similarity via BERTScore (target >0.85), and financial domain metrics (return calculation accuracy >99.5%, risk score deviation <5%). Regression testing suite runs on every model version change, comparing outputs against baseline (current production model) across 10,000 test cases with automated pass/fail gates: accuracy degradation <2%, latency increase <10%, cost increase <15%. AWS SageMaker Model Monitor tracks data drift via KL divergence (alert threshold 0.3) and concept drift through prediction distribution shifts, triggering retraining workflows when drift persists for 7 consecutive days.

Hallucination detection implements multi-layered verification. Factual consistency checks compare LLM outputs against source documents via entailment models (DeBERTa-v3-large on SageMaker), flagging contradictions with confidence scores <0.8 for human review. Numerical validation extracts financial figures from responses and cross-references against portfolio management system APIs, rejecting outputs with >1% variance. Citation verification ensures all investment recommendations reference specific research documents, with Lambda functions validating document IDs exist in Glue Catalog and access timestamps are within 90 days. Confidence calibration via temperature scaling adjusts model output probabilities to match empirical accuracy, enabling reliable uncertainty quantification. Responses with calibrated confidence <80% route to senior advisor review queue in Salesforce with context highlighting uncertain claims. Guardrail policies in Amazon Bedrock enforce content filtering: PII redaction (SSN, account numbers), profanity blocking, competitor mention restrictions, and regulatory keyword detection (insider trading, market manipulation) with custom deny lists updated weekly.

Online evaluation leverages A/B testing framework with advisor cohort randomization (control: current process, treatment: AI-assisted workflow) measuring business KPIs: analysis time reduction, recommendation acceptance rate, client satisfaction scores, and revenue per advisor. Statistical significance testing via sequential probability ratio test (SPRT) enables early stopping when treatment superiority reaches 95% confidence, reducing experiment duration from 8 weeks to 4 weeks average. Multi-armed bandit algorithms (Thompson Sampling) dynamically allocate traffic across model variants (Claude 3.5 Sonnet, GPT-4 Turbo, fine-tuned domain model) optimizing for composite reward function: 0.5 * accuracy + 0.3 * latency + 0.2 * cost. SLI/SLO definitions establish quality gates: recommendation acceptance rate >75% (SLI: accepted recommendations / total recommendations, SLO: 7-day rolling average), analysis accuracy >95% (SLI: senior advisor validation score, SLO: weekly audit of 100 random samples), hallucination rate <2% (SLI: factual consistency failures / total responses, SLO: daily monitoring).

Cost-performance optimization implements tiered model routing based on query complexity and business value. Simple queries (portfolio balance, recent transactions) route to Claude 3 Haiku (cost $0.25 per 1M tokens) with 500ms latency target. Complex analysis (multi-asset optimization, scenario modeling) routes to Claude 3.5 Sonnet (cost $3 per 1M tokens) with 3-second latency budget. High-stakes recommendations (>$1M portfolio changes) invoke ensemble approach with multiple models and human validation, accepting 10-second latency for accuracy assurance. Prompt caching via semantic hashing stores responses for identical queries (TTL 3600 seconds) and similar queries within cosine similarity 0.95 (TTL 1800 seconds), reducing API calls by 40% and saving $60,000 monthly. Batch processing for non-urgent tasks (nightly portfolio reports, weekly market summaries) uses provisioned throughput pricing, achieving 50% cost reduction versus on-demand. Failure isolation patterns implement circuit breakers (failure threshold 10%, timeout 5 seconds, half-open retry after 60 seconds) preventing cascade failures when LLM APIs experience degradation, with graceful fallback to cached responses or simplified rule-based recommendations.

Implementation Plan

Phase 1 (Months 1-2) establishes foundational infrastructure and baseline observability. Week 1-2 activities include AWS account structure setup via AWS Organizations (6 accounts: management, shared-services, dev, staging, prod, security), AWS Control Tower guardrails deployment (encryption enforcement, VPC flow logs, CloudTrail organization trail), and AWS SSO configuration with Okta federation. Terraform workspace initialization provisions VPC architecture (CIDR 10.0.0.0/16 prod, 10.1.0.0/16 non-prod), Transit Gateway for inter-VPC routing, and VPC endpoints for S3, DynamoDB, Secrets Manager reducing NAT Gateway costs by $3,000 monthly. Week 3-4 focuses on observability stack: CloudWatch Log Groups with 7-year retention for audit logs and 90-day retention for application logs, X-Ray tracing enabled for all Lambda functions and ECS tasks, CloudWatch Dashboards for golden signals (latency, traffic, errors, saturation), and PagerDuty integration for on-call rotation. Week 5-6 implements data lake foundation: S3 bucket structure with lifecycle policies, Glue Catalog database schemas, Lake Formation permissions model, and initial data ingestion pipelines from Salesforce (AppFlow) and portfolio system (DMS). Week 7-8 delivers CI/CD pipeline: CodePipeline with GitHub integration, CodeBuild projects for Terraform validation and Docker image builds, ECR repositories with image scanning enabled, and automated deployment to dev environment. Resourcing: 2 cloud architects, 2 DevOps engineers, 1 security engineer. Risks: AWS service quota limits (mitigate via advance quota increase requests), Terraform state management (mitigate via S3 backend with DynamoDB locking), team AWS certification gaps (mitigate via 40 hours training budget per engineer).

Phase 2 (Months 3-5) develops core application services and GenAI capabilities. Month 3 activities include API service development: FastAPI application on ECS Fargate with OpenAPI specification, JWT authentication via Cognito, rate limiting middleware (100 requests per minute per user), and integration with Aurora PostgreSQL for user profiles and session management. Bedrock integration implements prompt templates for portfolio analysis, market commentary, and risk assessment use cases with version control in Git and A/B testing framework via LaunchDarkly feature flags. Month 4 focuses on RAG pipeline: embedding generation jobs via AWS Batch processing 25TB research corpus (estimated 30 days runtime), OpenSearch cluster provisioning with index templates and mapping definitions, and retrieval service API with semantic search and metadata filtering. Data pipeline development includes Glue ETL jobs for portfolio data transformation, Kinesis streams for real-time market data ingestion, and Lambda functions for document processing (PDF extraction via Textract, entity recognition via Comprehend). Month 5 delivers recommendation engine: portfolio optimization algorithms on Lambda (15-minute timeout, 10GB memory), risk scoring models on SageMaker endpoints (ml.m5.xlarge), and compliance checking workflows via Step Functions integrating with internal risk management APIs. Guardrails implementation includes Bedrock Guardrails configuration, custom content filtering Lambda layers, and human review queue in Salesforce with SLA tracking. Resourcing: 4 backend engineers, 2 ML engineers, 1 data engineer, 1 QA engineer. Dependencies: Bloomberg API credentials and sandbox access (lead time 3 weeks), BlackRock

Aladdin integration documentation (request via account manager), Salesforce API governor limits increase (submit case 4 weeks advance). Risks: Bedrock model availability in us-east-1 (mitigate via multi-region fallback), embedding generation timeline (mitigate via parallel processing and spot instance cost optimization), third-party API rate limits (mitigate via request queuing and exponential backoff).

Phase 3 (Months 6-7) executes comprehensive testing and validation. Month 6 load testing uses Artillery.io generating 500 concurrent users with realistic usage patterns (60% portfolio queries, 30% document analysis, 10% recommendation requests) targeting P95 latency <3 seconds and error rate <0.1%. Chaos engineering experiments via AWS FIS inject failures: AZ outage simulation, DynamoDB throttling, Lambda timeout scenarios, and Aurora failover testing, validating RTO <1 hour and RPO <5 minutes. Security testing includes OWASP ZAP automated scans, manual penetration testing by third-party firm (budget $50,000), prompt injection attack simulations, and data exfiltration attempt monitoring. Month 7 user acceptance testing engages 50 pilot advisors with structured test scenarios (20 hours per advisor), collecting feedback via surveys and usability sessions. Model validation involves senior advisor review of 1,000 AI-generated recommendations measuring accuracy (target >95%), relevance (target >90%), and actionability (target >85%). Performance tuning optimizes database queries (index creation reducing P95 from 8 seconds to 500ms), implements ElastiCache for hot data (cache hit ratio >90%), and tunes Bedrock inference parameters (temperature, top_p, max_tokens) balancing quality and latency. Resourcing: 2 QA engineers, 1 security consultant, 1 performance engineer, 50 pilot advisors (20 hours each). Acceptance criteria: zero critical security findings, P95 latency <3 seconds under load, >90% pilot advisor satisfaction, model accuracy >95% on golden test set.

Phase 4 (Month 8) executes production deployment with risk mitigation strategies. Week 1 implements blue/green deployment: duplicate production environment (green) in separate Auto Scaling groups, deploy new version to green environment, execute smoke tests (100 synthetic transactions), and validate metrics match blue environment. Week 2 performs canary release: Route 53 weighted routing policy directing 5% traffic to green environment for 48 hours, monitor CloudWatch alarms (error rate, latency, business KPIs), and expand to 25% traffic if metrics acceptable. Week 3 completes cutover: shift 100% traffic to green environment, maintain blue environment for 72 hours as rollback target, and execute DR failover test to us-west-2 region validating RTO/RPO targets. Week 4 focuses on operational readiness: runbook documentation for 15 common scenarios (API degradation, database failover, LLM service outage), on-call rotation setup with PagerDuty escalation policies, and knowledge transfer sessions (40 hours) to SRE team. Rollback procedures include automated CloudWatch alarm triggering CodeDeploy rollback, manual rollback via Terraform workspace switch (execution time <15 minutes), and data rollback via Aurora point-in-time recovery. Resourcing: 3 SRE engineers, 2 backend engineers, 1 database administrator. Go/no-go criteria: zero P1 incidents in canary phase, <0.5% error rate, P95 latency <3 seconds, >99.5% availability.

Phase 5 (Months 9-10) stabilizes operations and optimizes performance. Month 9 activities include SRE playbook development for incident response (15 documented scenarios with step-by-step resolution procedures), capacity planning analysis (forecast 100% annual growth, provision 40% headroom), and cost optimization (Reserved Instance purchases for baseline capacity, Savings Plans for Fargate and Lambda, S3 Intelligent-Tiering reducing storage costs 30%). Month 10 delivers knowledge transfer: 80 hours instructor-led training for operations team covering architecture, troubleshooting, and deployment procedures, shadowing program pairing SREs with development team for 2 weeks, and certification program requiring operations team to resolve 10 simulated incidents independently. Optimization initiatives include query performance tuning (reduce Aurora CPU utilization from 70% to 40%), cache hit ratio improvement (ElastiCache from 85% to 95% via TTL tuning), and LLM cost reduction (implement prompt compression reducing token usage 25%). Resourcing: 2 SRE engineers, 1 FinOps analyst, 1 technical writer. Success criteria: mean time to resolution (MTTR) <30 minutes for P2 incidents, operations team independently managing deployments, monthly infrastructure costs within $180,000 budget.

Integration & Operations

Integration architecture implements API-first design with OpenAPI 3.0 specifications defining contracts for 12 core services: portfolio analysis, recommendation engine, document processing, market data ingestion, client profile management, compliance checking, reporting, notification, authentication, audit logging, feature flags, and health monitoring. REST APIs use JSON payloads with gzip compression, OAuth 2.0 bearer tokens (JWT with 1-hour expiration), and rate limiting via API Gateway (10,000 requests per minute organization-wide, 100 requests per minute per user). Salesforce integration leverages Platform Events for real-time notifications (advisor task creation, client alert dispatch) and Bulk API 2.0 for batch data synchronization (nightly client profile updates, weekly performance reporting). BlackRock Aladdin integration uses proprietary REST APIs with mutual TLS authentication, implementing retry logic with exponential backoff (initial delay 1 second, max delay 60 seconds, max attempts 5) and circuit breaker pattern (failure threshold 20%, timeout 10 seconds). Bloomberg Terminal integration consumes real-time market data via WebSocket connections (500,000 messages per second) with Kinesis Data Streams buffering and Lambda functions transforming FIX protocol messages to normalized JSON schema. Data exchange patterns include synchronous request/response for interactive queries (timeout 5 seconds), asynchronous job submission for long-running analysis (SQS queue with visibility timeout 15 minutes), and event-driven notifications via EventBridge (12 event types with schema registry validation).

Observability stack implements comprehensive monitoring across infrastructure, application, and business metrics. CloudWatch Metrics captures 150+ custom metrics: API latency percentiles (P50, P95, P99), error rates by endpoint and status code, LLM inference duration and token usage, cache hit ratios, database connection pool utilization, and business KPIs (recommendations generated, advisor productivity, client satisfaction). CloudWatch Alarms define 45 alert conditions with severity-based routing: P1 critical (>1% error rate, P95 latency >5 seconds, availability <99.9%) pages on-call engineer via PagerDuty, P2 high (>0.5% error rate, P95 latency >3 seconds) creates Jira ticket with 4-hour SLA, P3 medium (cost anomaly >20%, cache hit ratio <85%) sends Slack notification to engineering channel. X-Ray distributed tracing instruments all service calls with custom subsegments for LLM invocations, database queries, and external API calls, enabling end-to-end request flow visualization and bottleneck identification. CloudWatch Logs Insights queries support operational investigations with saved queries for common patterns: error rate by endpoint, slowest database queries, LLM hallucination incidents, and authentication failures. CloudWatch Dashboards provide role-based views: executive dashboard (business KPIs, cost trends, availability), engineering dashboard (latency percentiles, error rates, deployment frequency), and operations dashboard (infrastructure health, alert status, incident timeline).

Service Level Objectives (SLOs) establish measurable reliability targets with error budgets driving operational decisions. Availability SLO: 99.9% uptime (43 minutes monthly downtime budget) measured via CloudWatch Synthetics canaries executing synthetic transactions every 5 minutes from 6 geographic locations. Latency SLO: P95 <3 seconds for interactive queries measured via CloudWatch Metrics with 7-day rolling window, error budget consumption triggers deployment freeze when <10% budget remains. Error rate SLO: <0.5% for API requests measured via ALB access logs and application error logging, budget exhaustion requires incident review and corrective action plan. Data freshness SLO: market data latency <15 minutes measured via timestamp comparison between Bloomberg ingestion and OpenSearch availability. Error budget policy defines consequences: 100-75% budget remaining allows weekly deployments, 75-25% requires change advisory board approval, <25% triggers deployment freeze and mandatory postmortem. SLO reporting dashboard tracks budget consumption trends, forecasts budget exhaustion date, and highlights services at risk.

Operational playbooks document procedures for 20 common scenarios with step-by-step resolution guides. Incident response playbook defines severity classification (P1: customer-impacting outage, P2: degraded performance, P3: isolated errors), escalation procedures (P1 immediate page, P2 within 15 minutes, P3 next business day), and communication templates (status page updates, stakeholder notifications, postmortem reports). Database failover playbook covers Aurora automatic failover (RTO 2 minutes), manual failover procedures via AWS CLI, connection string updates in Secrets Manager, and application restart procedures. LLM service degradation playbook implements fallback strategies: switch to alternative model (Claude to GPT-4), enable cached response serving, activate simplified rule-based recommendations, and communicate degraded functionality to users. Deployment playbook standardizes release procedures: pre-deployment checklist (backup verification, rollback plan, stakeholder notification), deployment execution (Terraform apply, ECS task definition update, health check validation), and post-deployment validation (smoke tests, metric comparison, error log review). Change management process requires RFC submission 5 business days advance for standard changes, emergency change approval within 2 hours for P1 incidents, and change advisory board review for high-risk changes (database schema modifications, security policy updates, multi-service deployments).

Performance management implements proactive capacity planning and auto-scaling policies. Load profile analysis identifies three daily patterns: market open surge (6:30-10:00 AM ET, 500 concurrent users), steady state (10:00 AM-3:00 PM ET, 200 concurrent users), and market close spike (3:00-5:00 PM ET, 400 concurrent users). ECS Service Auto Scaling uses target tracking policies: 70% CPU utilization target, 60-second scale-out cooldown, 300-second scale-in cooldown, minimum 20 tasks, maximum 200 tasks. Scheduled scaling pre-provisions capacity 30 minutes before market open (scale to 100 tasks) and market close (scale to 80 tasks), reducing cold start latency. Lambda concurrency reservations allocate 500 concurrent executions for critical functions (document processing, recommendation generation) preventing throttling during peak load. Aurora Auto Scaling adjusts ACU allocation (8-128 range) based on CPU and connection metrics with 5-minute evaluation period. DynamoDB on-demand capacity mode eliminates provisioning complexity, with monthly cost review identifying tables suitable for provisioned capacity conversion (>1M requests daily with predictable patterns). Cost management implements AWS Budgets with $200,000 monthly threshold, 80% alert triggering

FinOps review, and 100% alert requiring executive approval for additional spend. Cost anomaly detection via AWS Cost Anomaly Detection identifies unusual spending patterns (>20% variance from 7-day average) with root cause analysis by service and usage type. Tagging strategy enforces mandatory tags (Environment, Application, CostCenter, Owner) via AWS Config rules, enabling cost allocation reports by business unit and chargeback to advisory departments. Day-2 operations include quarterly DR drills validating us-west-2 failover procedures (RTO <1 hour, RPO <5 minutes), monthly backup restore tests verifying Aurora snapshots and S3 versioning, weekly security patching for ECS task images and Lambda runtimes, and continuous compliance evidence collection via AWS Audit Manager for SOC 2, SEC 17a-4, and GDPR requirements.