Company A - Technical Integration Overview

This document outlines the current system architecture and security posture of Company A's customer data platform (CDP). The CDP is a cloud-native, microservices-based application hosted on AWS using ECS Fargate.

Key modules include:
- Data ingestion pipelines from REST APIs and batch uploads (CSV, JSON) through AWS API Gateway and Lambda
- Stream processing using Apache Kafka (MSK) and enrichment via Spark jobs
- Storage layer: Amazon S3 for object data, DynamoDB for metadata, Redshift for analytics
- Machine learning module using SageMaker models for segmentation scoring

Security and access controls:
- TLS 1.2 enforced on all endpoints
- IAM roles and resource-based policies for access control
- Data encrypted at rest using AWS KMS-managed keys
- Single Sign-On (SSO) via SAML 2.0 integration with Okta
- No customer PII is stored unencrypted

All infrastructure is deployed via Terraform and CI/CD is managed through GitHub Actions. Vulnerability scans run nightly using Snyk and container images are verified with AWS ECR signing.

Documentation last updated: July 2025

3.0 Data Infrastructure and Warehousing

Company A utilizes Amazon Redshift as its primary data warehousing solution. ETL pipelines are constructed using Apache Airflow, and data is ingested from both structured (PostgreSQL, Oracle) and unstructured (CSV, JSON from S3) sources. Data is cleaned and validated using Great Expectations before being loaded into Redshift tables. Daily and hourly snapshots are maintained to enable rollback and historical auditing.

Redshift Spectrum is used to query S3 datasets directly when batch analysis is required. Usage of AWS Glue Data Catalog allows for unified metadata management. In the event of infrastructure failure, snapshots are stored with 7-day redundancy using Amazon S3 Glacier.

4.0 Microservices Architecture

The core application is structured as a suite of microservices, with individual services responsible for user management, authentication, notifications, billing, and search functionality. Each service communicates via REST and gRPC. Deployment is handled via Kubernetes (EKS), with Helm charts used for templating. All services expose Prometheus metrics and are monitored through Grafana dashboards.

Secrets and keys are managed via AWS Secrets Manager, and internal service-to-service authentication is implemented with mutual TLS. Canary deployments are run on staging

environments for 48 hours prior to full rollout. APIs are rate-limited with Kong Gateway and logged with Fluentd.

5.0 Logging, Monitoring, and Alerting

All system logs are aggregated via Fluentd and shipped to Amazon CloudWatch and Splunk. Application logs are structured in JSON format with severity levels. Business events are pushed to Kafka for later analysis in Snowflake. Alerts are defined in Prometheus AlertManager and routed to PagerDuty with escalation policies in place.

SLA breaches trigger real-time alerts to the SRE team. Latency and throughput metrics are visualized via Grafana dashboards, which are shared across both engineering and product teams. Deployment health is evaluated based on custom error budgets defined per service.

6.0 Security and Compliance

Company A complies with SOC 2 Type II and ISO/IEC 27001 standards. Data is encrypted at rest using AES-256 and in transit via TLS 1.3. Endpoint security is enforced through CrowdStrike Falcon and internal device posture compliance. Access control is managed using Role-Based Access Control (RBAC) and AWS IAM federation.

The company runs quarterly penetration tests through a third-party vendor and maintains a public vulnerability disclosure program. Incident response policies follow NIST 800-61 and are documented within internal Confluence wikis. PII and PHI fields are automatically redacted during logging and reporting processes.