HALORADIUS
Customer Case Study

Enterprise-grade Data Platform with Global Data Democratization and Declarative, Intent-based Data Orchestration

OVERVIEW

Our client, a leading technology company, with $29.1 billion in annual revenue, a 62,000-strong workforce with 70+ Million devices sold and deployed globally tackled a critical challenge: managing and deploying data services for internal development teams. Their existing approach, riddled with complexity and inconsistency, hampered developer productivity and inflated costs.

"$6 Million Savings

Halo Radius built and led a team of 50+ who successfully achieved a $4 million reduction in spending on data services along with a $2M annual reduction in our AWS spend."

GM and Senior Vice President of Cloud Products

THE CHALLENGE

The company's business units (BUs) lacked a common data processing platform, leading to several issues:

- **Slow Time to Market**: Sharing data across BUs was cumbersome, requiring custom integrations and lengthy approvals, often taking multiple quarters
- **Reinvented Wheel**: Each BU independently addressed similar data challenges using disparate technologies, limiting knowledge sharing and causing significant inefficiencies.
- **Resource Drain**: Building and maintaining data pipelines in-house required dedicated DevOps and cloud expertise, diverting valuable resources from data engineering and machine learning (ML) tasks.

THE APPROACH

Halo Radius addressed the client's challenges by providing a unified data processing platform that empowers data engineers and ML engineers. Key features include:

- **Declarative, Intent-Based Orchestration**: This eliminates the need for DevOps expertise in each team by allowing engineers and data scientists to safely define their data pipelines.
- **Automated Deployments**: Halo Radius automated the deployment of data pipelines, including ingestion and serving components, accelerating time to market.
- **Data Democratization**: The platform fosters collaboration by enabling BUs to share data and code seamlessly, breaking down data silos and accelerating new product development.

HALORADIUS

Delivering Future-Proof Data Services at Cloud Scale

∿ THE SOLUTION

Halo Radius delivered a new platform that overhauled data management, unlocking secure data sharing and efficient handling of diverse formats. Its innovative, adaptive architecture guarantees seamless scaling, readying the development environment for machine learning breakthroughs.

The project ultimately delivered significant benefits, including:

- **Improved Data Management**: Secure data sharing and support for various data formats (structured, unstructured, and rich media) enabled efficient data utilization.
- **Enhanced Scalability**: The platform's architecture allows for seamless scaling to accommodate future growth.
- **ML Workload Readiness**: Integration with MLflow facilitated the deployment of ML models within the data processing pipelines.
- **Microservices Architecture**: This approach promotes modularity, maintainability, and performance optimization.

These advancements, paired with a faster time to market, propelled the client's data teams to remarkable victories.

⌕ INSIDE THE PLATFORM

The Halo Radius solution created numerous developer-focused benefits that have fundamentally changed how the client builds and delivers data services company-wide.

- **Schema Management**: Data schemas are first-class citizens, can be evolved, and play an essential role at maintaining consistency across each step of the pipeline.
- **Error-free Pipeline Definition**: The Scala-based declarative DSL allows to check the robustness of the pipeline definition by detecting problems at compile-time, before anything gets deployed.
- **Automated Table Creation**: By using the declarative DSL, users will be able to create and update data tables (Delta Lake and SQL).
- **Data Catalog**: For audit and debug, users are able to inspect every data file ingested into the system.
- **Automated Ingest Servers**: Structured files are automatically ingested and promoted into a Delta Lake table. Semi-structured files and binary files only require an additional custom ETL service, behaving the same otherwise.
- **Automated Data Serving**: A Prisma-based, fully automated service exposes a GraphQL API for users, so that they can fetch data from time-series and relational databases.

code frameworks	Scala, Python Java, Go Spark
data storage	Delta Lake, TimescaleDB, RDS PostgreSQL, DynamoDB, S3
data streaming	Spark, Kafka, Pulsar, MLflow, Jupyter
deployment orchestration	Prometheus, Grafana, Jenkins, Argo, Terraform, Amazon Managed Workflows, Rancher, EKS
cloud services	Kubernetes, EKS, IAM, S3, DynamoDB, RDS, Compute, Data, Cost Optimization