technovate.one

Cloud-Native Analytics Platform For A Managed Connectivity & Communications Industry

This case study describes the design and implementation of a cloud-native analytics platform on AWS for a managed connectivity and communications services provider operating across SD-WAN, Voice, SIP, and UCaaS offerings. The solution unified data from multiple transactional systems into a high-performance OLAP data warehouse using StarRocks, with a scalable API-driven presentation layer for analytics and reporting.

Key Questions Answered

01

How can data from multiple telecom and network transactional systems be unified into a single analytics platform?

02

How can analytics performance be improved for complex, cross-product reporting?

03

What cloud-native architecture best supports scalability, reliability, and future growth?

Evaluation Criteria

- Query Performance And Concurrency
- Accuracy And Reconciliation Across Systems
- Scalability And Reliability On Cloud Infrastructure
- Adoption By Operational, Finance, And Leadership Teams

Sources Used

- Eight Independent Transactional Data Sources
- Internal Operational And Financial Reporting Definitions
- Stakeholder Interviews And Workflow Analysis
- Cloud-Native And OLAP Architecture Best Practices

Context And Background

The Situation

Before this work began, the organization delivered multiple managed services—including SD-WAN, Voice, SIP, and Unified Communications—supported by eight different transactional systems. Each system handled a specific function such as billing, provisioning, usage, customer management, or network operations.

Baseline conditions included:

- Data siloed across multiple systems
- Manual effort required for cross-product reporting
- Slow turnaround for usage, revenue, and performance analytics
- Operational systems being queried directly for reporting

The Problem

The lack of a centralized analytics platform made it difficult to gain a unified view of customers, services, usage, and financial performance. Reporting across products and systems was slow, inconsistent, and not scalable. Analytical workloads placed additional strain on transactional systems not designed for high-concurrency analytics.

What Had Been Tried

1. Direct reporting on individual transactional systems
2. Periodic data exports and spreadsheet-based consolidation
3. Static reports with limited drill-down and filtering

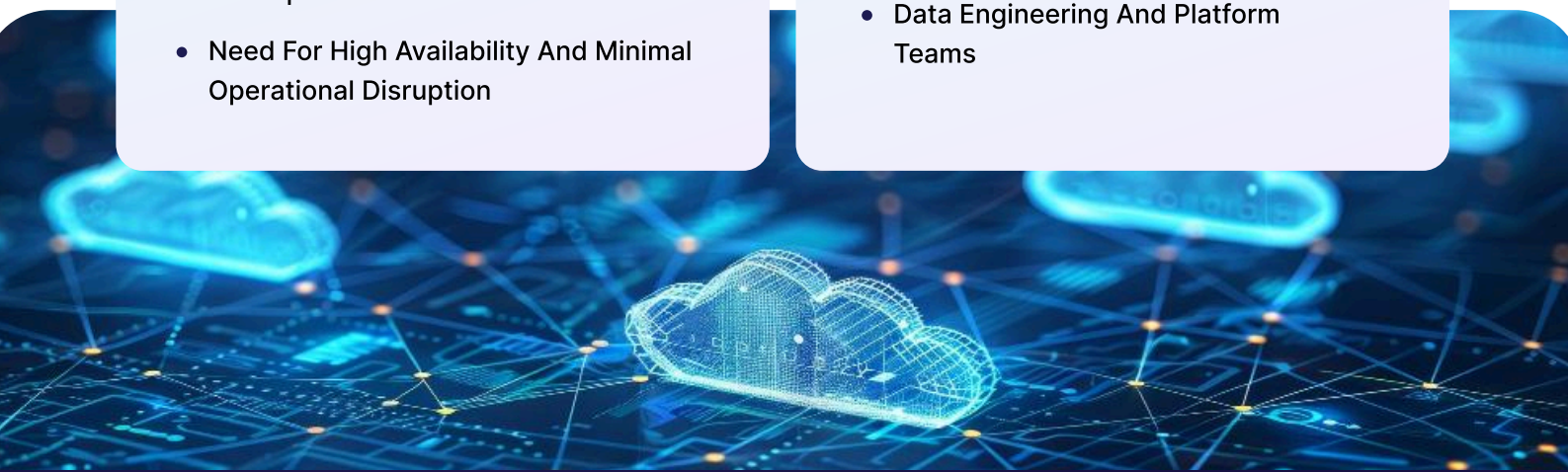
These approaches did not scale across eight systems and could not support timely, reliable analytics.

Constraints

- Heterogeneous Schemas And Data Semantics Across Systems
- High Data Volumes Driven By Usage And Event Records
- Fixed Reporting Cycles For Business And Operational Reviews
- Need For High Availability And Minimal Operational Disruption

Stakeholders

- Network Operations And Service Delivery Teams
- Finance, Billing, And Revenue Assurance Users
- Executive Leadership
- Data Engineering And Platform Teams



Scope And Approach

What was in scope

- Cloud-native ingestion of data from eight transactional systems
- Centralized OLAP data warehouse built using StarRocks on AWS
- Standardized analytical and dimensional data models
- A .NET Core API layer to expose governed analytics endpoints
- An Angular-based web application for dashboards and reporting

What was out of scope

- Replacement or modification of transactional systems
- Real-time network monitoring or alerting
- Advanced predictive or AI-driven analytics

Inputs

- Transactional data related to customers, services, usage, billing, and operations
- Existing reports and KPI definitions
- Data profiling and quality assessments
- Stakeholder interviews and analytics requirements

Validation

- Source-to-warehouse reconciliation checks
- Cross-system KPI validation workshops
- Performance and concurrency testing
- User acceptance testing

Findings And Insights

What the data showed

- Most analytics use cases required combining data across multiple systems and products
- Queries were aggregation-heavy, time-series driven, and high in concurrency
- Performance degraded significantly when analytics were run on source systems

Root causes

- Data fragmentation across eight independent systems
- Lack of a unified analytical model spanning services and products
- Reporting infrastructure not designed for cloud-scale OLAP workloads

Critical insights

- A cloud-native OLAP platform was essential for performance and scalability
- Centralized KPI logic was necessary for consistency and trust
- API-based delivery would decouple analytics from presentation and enable reuse

Solution

What Was Recommended

A cloud-native analytics architecture on AWS, centered around a high-performance StarRocks OLAP data warehouse, with analytics delivered through a stateless API and a modern web UI. This approach provided scalability, reliability, and performance that incremental improvements to legacy reporting could not achieve.

What Was Built Or Changed

- A StarRocks-based OLAP data warehouse deployed on AWS
- Unified analytical models integrating data from eight transactional systems
- Cloud-native ingestion pipelines optimized for scale and reliability
- A .NET Core API exposing secure, reusable analytics endpoints
- An Angular-based presentation layer for interactive dashboards and insights

How It Was Delivered

Timeline and key milestones

- Cloud architecture design and environment setup
- Data source analysis and schema harmonization
- Incremental ingestion and validation of eight systems
- OLAP modeling and performance optimization
- API and UI development
- Load testing and production rollout

Who did what, by role

- Data engineering: ingestion pipelines, modeling, and performance tuning
- Backend engineering: API design, security, and scalability
- Frontend engineering: dashboard and UX development
- Business stakeholders: validation and acceptance

Dependencies, integrations, or partnerships

- AWS cloud infrastructure and services
- Secure integrations with transactional systems
- Existing authentication and access management platforms

Results

Quantitative Outcomes

- Significant reduction in analytics query response times
- Stable performance under concurrent access
- Elimination of reporting load on transactional systems

Qualitative Outcomes

- Unified view of customers, services, usage, and revenue
- Faster operational and financial decision-making
- Increased confidence in enterprise-wide KPIs

Adoption or usage

- Strong adoption across operations, finance, and leadership
- Reduced dependency on manual and ad-hoc reporting

Reflection

What worked and why

- Cloud-native architecture enabled elastic scaling
- StarRocks delivered consistent OLAP performance for large datasets
- API-first design improved flexibility and governance

What did not work and what changed

- Initial data normalization required additional iterations
- KPI definitions evolved through stakeholder alignment

What is reusable

- Cloud-native ingestion and modeling patterns
- OLAP-optimized schema designs
- API-driven analytics delivery framework

Limits and where adaptation is needed

- Real-time analytics would require streaming ingestion
- Predictive insights would need additional data science layers

Takeaway

By building a cloud-native analytics platform on AWS using StarRocks for OLAP and a modern API-driven presentation layer, the organization successfully unified data from eight transactional systems supporting managed connectivity and communications services. The key lesson was that centralized, cloud-native OLAP architectures are critical for scalable analytics in complex telecom environments.