Albocensa

Modernizing Legacy Systems through Data Integration

Data Warehouse Integration with Legacy Systems for a Fuel & Gas Company

Customer Case Study
Logistics Industry

From Months to Days
Reduced time for implementing any 3rd party

Cutting Edge Capabilities
10+ years old systems replaced with modern solutions

Linux at it's Core
Replaced Windows XP to modern Linux solutions

True CI/CD
Every change, tested, tagged, and deployed

Client Background

A long-standing player in the fuel and gas sector was grappling with outdated internal applications, tooling, and systems that had become slow and cumbersome over the years. These inefficiencies were particularly problematic for employees dealing with real-time client interactions, either in person or over the phone, leading to delays and a decline in customer service quality.

The Challenge

The core challenge was the lagging speed and efficiency of the company's legacy systems. With an extensive market presence, the client needed to modernize these systems to better support their operational needs. The aim was to enhance data accessibility and performance to ensure that employees could retrieve and display data swiftly during direct client interactions. This was critical not only for improving customer service but also for maintaining operational efficiency and compliance with regulatory changes, such as product restrictions in various countries.

PROBLEM STATEMENT

How can we modernize outdated internal systems to support fast, compliant, real-time customer interactions?

To address these challenges, the Albocensa team and the client embarked on a project to better leverage their existing Data Warehouse infrastructure.

Modernizing Data Access: The Albocensa Integration Blueprint

By leveraging existing Data Warehouse infrastructure, Albocensa created a **unified integration layer** that transforms raw data into actionable business intelligence. The solution bridges the gap between complex historical datasets and modern application needs, ensuring that critical information - from inventory to compliance policies **is always a single API call away.**

The Solution Framework

The Data Engine: Python-Powered ETL

The backbone of this solution is a modern **ETL (Extract, Transform, Load)** pipeline. Python was chosen for its versatility and efficiency in handling scalable, high-volume data transformations.

- **Scalable Extraction:** Efficiently pulls both real-time and historical data from the core Data Warehouse.
- **Complex Logic:** Handles intricate transformations that legacy systems might struggle with.
- **Performance:** Optimized for large datasets, ensuring data is cleaned and standardized before entering the API layer.

API Integration (MuleSoft)

MuleSoft serves as the orchestration layer, connecting the Data Warehouse to the client's diverse application ecosystem. It follows an **API-led connectivity approach**, promoting reusability and governance.

- **Seamless Integration:** Connects legacy applications and modern cloud tools through a unified platform.
- **Reusability:** Individual components (e.g., Inventory, Customer info) are built once and used across multiple projects.
- **Governance:** Provides a secure, monitored environment for data flowing between systems.

Core Reusable API Components

Customer Intelligence API

Provides a 360-degree view of the customer by merging various data streams into a single endpoint.

Data Scope
Latest & Historical

Primary Use
CRM Integration

Key Benefit
Personalized customer experiences

Real-Time Inventory API

Tracks stock levels and movements across different locations, updated via the Python ETL processes.

Data Scope
Live & Historical Transactions

Primary Use
Supply Chain Management

Key Benefit
Reduced stockouts & Optimized storages

Policy & Compliance API

A specialized component designed to enforce specific product rules, such as regional bans or restrictions.

Data Scope
Global & Local policies

Primary Use
Legal & Regulatory compliance

Key Benefit
Automated prevention of banned sales

Strategic Project Outcomes

Data Reusability
APIs designed once can be integrated across all existing and future client systems.

System Agnosticism
The Mulesoft layer ensures the Data Warehouse remains accessible regardless of the end-user application.

Operational Efficiency
Automated ETL and API flows eliminate manual data retrieval and transformation tasks.

Regulatory Safety
Embedded product policies prevent the distribution of restricted items in specific markets automatically.

Technologies

The technology stack was carefully chosen to support the high demands of the fuel and gas industry.

See our full potential >

APIs
APIs built on robust platforms ensured that data interactions were secure and efficient.

Mulesoft
Mulesoft provided the necessary middleware to ensure that existing and new integrations worked flawlessly together, enhancing the overall system architecture.

Python
Python was utilized for its powerful data processing capabilities, allowing for quick manipulation and retrieval of data.

Team

The project was executed by a small yet highly efficient Albocensa team. This streamlined structure enabled close collaboration between development and quality assurance, with developers focused on building and integrating system features while QA specialists continuously validated functionality, identified issues early, and ensured reliability. This approach supported agile development practices, allowing for rapid testing, quick feedback cycles, and efficient implementation of the new systems.

See our story >

Client Benefits

A significant innovation in this project was the ability to reuse many of the integrations previously developed for the Data Warehouse. This approach not only sped up the project delivery timeline but also reduced the costs associated with developing new software components from scratch. The team implemented advanced data caching strategies to decrease load times and improve the responsiveness of applications during peak usage periods.

Accelerated Delivery & Efficiency

- **Accelerated Timeline:** Repurposing existing Data Warehouse integrations and APIs allowed for a significantly faster go-live compared to building from scratch.
- **Cost Reduction:** This approach minimized the need for custom middleware and interface development, leading to substantial savings in software development costs.

System Performance & Satisfaction

- **Enhanced Stability:** The new architecture increased throughput and slashed transaction processing times, resulting in far fewer system timeouts and errors.
- **Improved Morale:** Reliability and speed directly increased employee satisfaction, leading to a marked decline in IT support tickets related to system slowness.

Optimized Customer Service

- **Rapid Information Retrieval:** Employees now have near-instant access to complete customer and product data during live interactions.
- **Service Excellence:** Faster access has shortened average interaction times and improved first-contact resolution rates, enabling more effective and professional service.

Performance Optimization

- **Superior Responsiveness:** Advanced caching strategies dramatically reduced data load times and ensured the application remained responsive even during peak usage.
- **Infrastructure Efficiency:** API optimizations significantly lowered backend query loads, improving overall system architecture and longevity.

Global Regulatory Compliance

- **Risk Mitigation:** Automated management of product-specific policies reduced compliance exceptions and policy violation incidents across different regions.
- **Audit Readiness:** By automating compliance checks, the solution strengthened audit preparation and lowered international regulatory risk.