

Client Situation: A Large retailer in the US leveraging the Mainframe based OMS for Improved Operational Efficiency

Problem Statement:

- Incomplete integrations which was a challenge for Omni-channel interfaces
- Escalation of Supply Chain and IT costs around Order Fulfillment
- Inaccurate Order volume forecasting Due to Data Silos
- Incorrect Order Promising (on-time / accurate deliveries) and inefficient fulfillment process

Our Solution:

- API Enablement on Mainframe
- Data Integration using APIs
- Leverage knowledge-based engineering approach to discover and analyze existing mainframe footprint
- Analyze and Deconstruct the mainframe portfolio so workloads can be integrated with minimal impact to other dependent applications

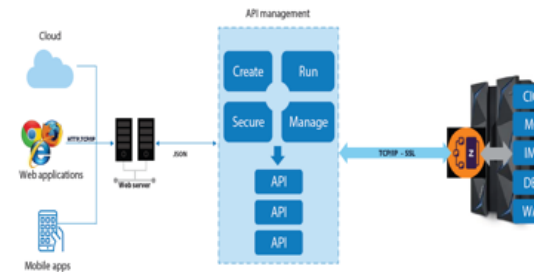
Business Outcomes

- ~15% reduction in Customer Complaints due to delayed or inaccurate fulfillments ↓
- 7% reduction in **Support Costs** due to correct identification of customer record and access to interaction trail across channels ↓
- 5%+ improvement in **On-time Order Delivery** ↑

Our Approach:

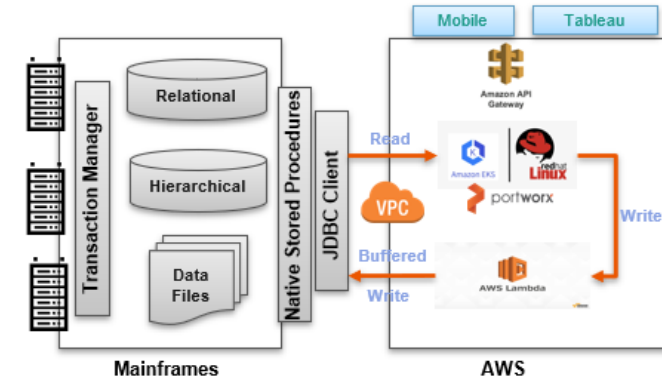


As-Is-Architecture Using Proprietary Products



Phase -0

Augmenting Mainframes – New System Around Edges



Pros

- Mostly Configuration Driven and Less Code
- Faster Time to Market
- Immediate Proof Of Value of Additional Functionality
- Leveraging Existing Cobol/DB2 Workforce

Cons

- Vendor Lock In
- Increasing Infrastructure Cost
- Less Agility
- No Voice Over Supports for new Additional Cutting-Edge Digital Transformations
- Scalability/Distributed Computing / Data Write Loss / Performance

- Relatively Cheaper
- Scalability with Autoscaling Components
- Data Loss Prevention with Buffered Writes
- No Data Persistence On Cloud with full in-transit encryption and On-Prem managed Data Access via Rest APIs
- More Use of Cloud Native Functionalities

- No Distributed Data Access
- Compromises on Data Models (Use as is)
- Cobol PS1 SPs to Native Stored Procedure Conversions
- JDBC Client Limitations