

## Case Study

# Migration from Enosix to MuleSoft

### Client Overview

The customer is a global supply chain partner to thousands of companies in manufacturing, infrastructure, broadcast, professional audio-visual, and telecommunications. The customer provides complete solutions for cable and wire management needs.



### Requirement

The customer uses SAP for their Inventory, pricing, quotes data, while the sales team uses Salesforce for its activity tracking. Daily pipelines are triggered for data synchronization within SAP and Salesforce systems. These pipelines ensure the customer accounts, inventory details, price, quotes, etc. are in sync between the two systems. The customer has implemented Enosix, an out-of-box services layer on top of SAP that helps with access to SAP objects for read and write operations.

Enosix Surface comes with functionality to connect to SAP from within Salesforce. Enosix provides Salesforce (Lightning) screens through which a connection to Enosix Surface may be established and SAP data may be retrieved for viewing and/or generating reports.

The customer was looking to implement an enterprise integration iPaaS solution that helps standardize the flow of data between systems and provides reusable APIs that may be invoked from other channels, as needed. The customer sought to transform enterprise technology landscape through reusable APIs and microservices.

### Challenges

- The usage of Enosix helped connecting SAP with Salesforce, and that worked for the immediate needs.
- It helped with real-time and batch updates to SAP objects. However, it introduced tightly coupled, point-to-point connections between the systems.
- For any other user or channel that needed the same data, outside of salesforce, the service was not reusable, it was also not horizontally scalable.
- The usage of Enosix surface helped with quick setup of screens within Salesforce that would connect to SAP to view the data. However, these screens were only accessible through Salesforce, and the underlying functions were not exposed as APIs for other users.

## Prowess Provided Solutions & Services

As part of the assessment of existing system, it was discovered that there are tightly coupled connections in the existing system.

The functionality within this system was not reusable by any other external channel, neither was this functionality individually scalable. The components or services were not separated based on the business logic and data operations.

MuleSoft provides an API-first methodology that helps with building reusable APIs. It also has a diverse set of connectors that help with connecting to various systems like databases, ERPs and CRMs. It provides out-of-box connectors for SAP and Salesforce that can help with quick connection to these systems for integration.

The Anypoint Platform is an iPaaS solution that combines design-time and run-time platforms into an integrated dashboard. It helps with the complete lifecycle of an API that covers design, implementation, deployment, and monitoring.

## Methodology and Process followed

As part of refactoring the project, the following methodologies were adopted

### 1) Separated APIs across Experience, Process and System layers.

As part of the refactoring the existing system, the following system interactions were studied

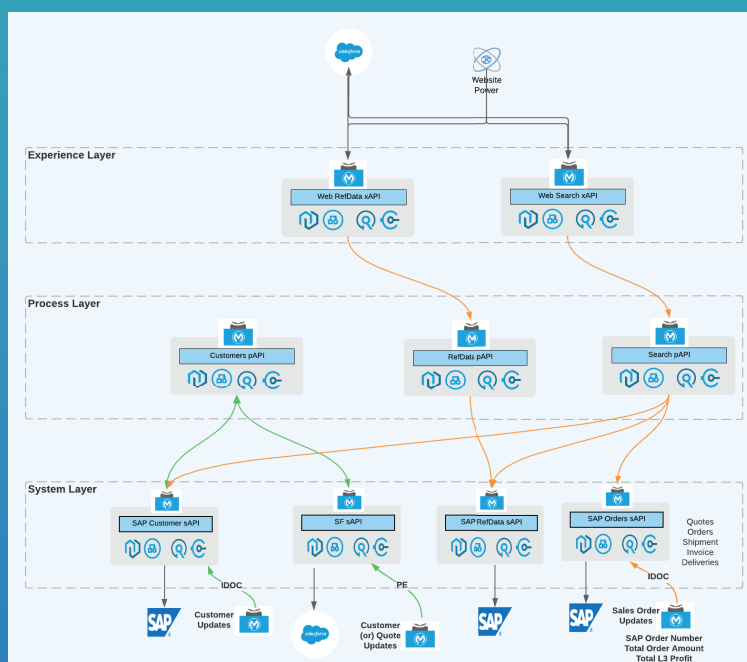
- External systems that interact with the current system (Experience APIs)



- Business logic that can be implemented and reused as needed (Process APIs)
- System interactions in the data layer (System APIs)

Based on the above touchpoints, a design was laid out to build APIs across Experience layer. This design ensured that individual APIs are scalable and reusable. Also, they would be deployed as microservices on the Anypoint CloudHub runtime environment provided by MuleSoft.

The picture below shows the various APIs designed by the team as part of reengineering the system.



## 2) API Design-first methodology

As part of design-first methodology, the API specification was built and shared with external system consumers. Appropriate adjustments were done based on feedback and a mocking service was provided for consumers. This ensured that the interface was well-understood by the consumers and the implementation was understood by the Mule application developers.

## Value Proposition from Prowess

- The introduction of an integration layer helped customer with removing the clutter of point-to-point connections that had become unmanageable.
- The implementation of MuleSoft iPaaS platform helped with quick setup of the platform without much infrastructure delays.

- The usage of Prowess out-of-box frameworks for observability helped bootstrap the platform with needed common functionality across the system.
- The reengineering of existing system through loosely coupled components helped with adapting quickly for dynamic scaling needs.
- The layered approach in the design helped with effective reusability through API connectivity.

For further information, please contact:

Prowess Software Services

Visit us at

[www.prowesssoft.com](http://www.prowesssoft.com)

Copyright © 2023 ProwessSoft

All rights reserved

All trademarks mentioned herein belong to their respective owners.

