LTIMindtree

Transforming the Core Banking Platform for a Global Banking and Financial Services Giant

Client

One of the largest banking and financial services organizations aimed to boost revenue by adopting industry-standard protocols for inter-bank operability. To realize this vision, they wanted to rationalize and enable an Application Programming Interface (API)-led economy, which would positively impact the organization's profitability, growth, and innovation. The bank wanted to move away from licensed products to open-source technologies, which could help it drive down operational costs even further.
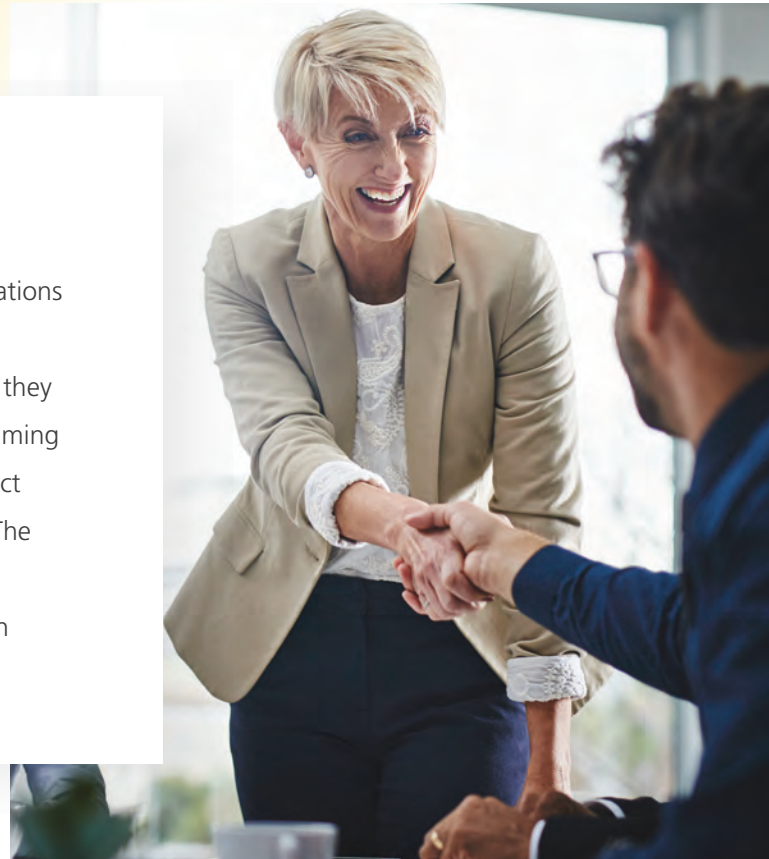
Challenges

Need to enhance core banking platform communication

The bank aimed to enhance communication between its core banking systems and modern API-enabled apps to unlock new user features. This aligned with their long-term strategy for growth, ensuring security and reusability.

Longer Time to Market (TTM) and increased cost of ownership

Prolonged application development resulted in significant feature rollout delays. These delays were worsened by licensing costs of proprietary API management products, leading to increased long-term ownership expenses.

Redundant bloated functionalities on the platform

The existing platform required rationalization to identify candidates for removal and baselining existing useful features.

Limited interoperability

Partners and customers had stringent requirements for interfacing via APIs compatible with the Banking Industry Architecture Network (BIAN).

LTIMindtree Solution

LTIMindtree conducted an API rationalization to enable the client's API-led business model and enhance operational efficiency. Introducing a developer portal significantly increased developer productivity and established a sustainable API-driven development model.

01 **Rationalized APIs**

The solution scanned, analyzed, consolidated, and rationalized approximately 500 APIs by removing the redundant ones (around 40% of unused/duplicate APIs). Additionally, it implemented reusable and scalable interface components, resolving performance bottlenecks. Shared services, customer data, and operational reporting were some of the diverse functional areas covered in this exercise.

02 **API standardization to increase interoperability**

BIAN-based APIs were designed and implemented, subscribed by third-party consumers.

03 **Reusable features and interfaces**

Around 90% of the system APIs were developed to be reusable with minor customizations specific to the use case. This reduced overall cost as well as TTM.

04 **Migration to open source**

More than 600 APIs were migrated from MuleSoft to open-source Java-based microservices.

Business Benefits

BIAN API standardization enabled seamless operations

More than **30** highly reusable **APIs** implemented

More than **1200 APIs** published to the the portal

30% increase in developer productivity

A **70% boost** in API performance through rationalization and modern microservices architecture