CASE STUDY

Large American Payments Organization: Debit Card Management Solution using Microservices

Client Overview

The client is a leading global ATM solution provider, enabling its customers to grow revenue, reduce cost, manage risk, and enhance the customer experience. It also enables financial institutions and independent ATM deployers to meet the self-service banking needs of consumers.

The Challenge

The clients' existing Card Management System was a heavy and monolithic application which was built using legacy technologies like EJBs, Java 6 for back-end and HTML, CSS, JavaScript and jQuery for front-end. The core problem with monolithic applications is that they are designed and operated as a single entity — one big hunk of code. If any part of the application fails, the whole application stops working.

This legacy architecture and technical platform was difficult to update, slow to meet the rapidly developing needs of the marketplace and costly to update and maintain. To meet consumers' evolving market demands, the client decided to modernize its platform from a monolithic design to micro services-based architecture, which would be more flexible to maintain and integrate with new solutions.

The client also wanted a resolution which easily extends any financial institution's needs.

info@opusconsulting.com

www.opusconsulting.com

The Solution

The client decided to partner with Opus to leverage its expertise in card management systems and Micro Services. Opus deployed their team of cards experts and technical architects to study the existing architecture and come up with a plan and new architectural approach to solve the challenges of micro-front-end, and focus on decomposition of applications into cohesive, loosely coupled suite of services, where each service completes one task independently that represents a small business capability.

The development was managed by full-stack developers, thereby delivering and maintaining complex software systems and improving velocity with quality.

Opus' workforce is committed to high quality implementations using the latest technologies.

- *Spring Boot, Java 8 and JPA were used for back-end development*
- *React-strap for front-end programing with Thyme leaf template engine*
- *Jenkins tools were used for automation*
- *Maven was incorporated as a build tool*

With React-strap, each service was built on micro-fronted style. This enabled each service to come up with its own front-end which is exactly opposite with monolithic architecture. This caused the applications to run smoothly. For Continuous Integration (CI) and Continuous Deployment (CD) methodologies of Agile, both Manual and Automation testing methods were executed.

Benefits Delivered

- Reduced maintenance cost for the providers
- Reduced latency by utilizing newer optimal technologies in the developed solution
- **Independent upgrades:** Each service has their own UI so can be deployed independent of other services
- Demonstrated 200-300 transactions per second
- **Independent scaling:** Each micro service now scaled independently based on load parameters
- User-friendly web/app experience for end users by intercommunicating with their internal systems