

# Migration from Monolithic Architecture to Microservices Stack for Masterpass

The client is a leading global payments network that offers digital payment solutions, enabling its customers to develop and implement credit, debit, prepaid and related payment programs.



The client also serves as an economic link among financial institutions, businesses, merchants, cardholders and governments worldwide, enabling them to use electronic forms of payment instead of cash and checks. Opus was the chosen implementation partner for one of their big retail wallet programs MasterPass.

## About MasterPass

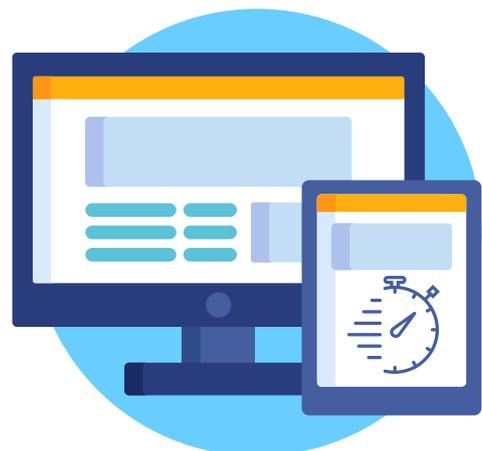
- Master Pass allows consumers to pay with any payment card, stored in any wallet, and on any device.
- The Master Pass suite of services includes:
  - In-store checkout – supports NFC, QR codes, tags and mobile devices used at points of sale.
  - Online purchases – eliminates entering detailed shipping & card information with every purchase.
  - Master Pass-connected wallets – enable banks, merchants and partners to offer their own wallets.

## Problem Statement

- Client's Digital Wallet was a heavy monolithic application; which was build using legacy technologies like EJBs, Java 6 for back-end and Ajax, JQuery for front-end. This older platform was difficult to update, slow to meet the rapidly developing needs of the marketplace and costly to update and maintain.
- To meet consumer market demands, client decided to modernize its platform from a monolithic design to a simpler, more flexible and easy to maintain solution.
- Client also wanted a solution which extend easily to merchants, providers and partners.
- In its migration plan, to remain competitive in the marketplace, the client was determined to leverage the latest technologies associated with Digital wallet modern solutions, and realize the associated benefits.

## Opus Solution

- Microservices: Opus solved the challenges with a new architectural approach emphasizing the decomposition of applications into single-purpose, loosely coupled suite of services. The development was managed by cross-functional teams, thereby delivering and maintaining complex software systems and improving velocity and quality required by today's digital business.



> Opus leveraged the latest technologies including Spring boot, Java 8 for back-end development and Angular JS for front-end programming. Ansible and Jenkins tools were used for Automation. Archaius was used for configuration externalization. Maven was incorporated as a build tool.

> For Continuous Integration (CI) and Continuous Deployment (CD) methodologies of Agile, both Manual and Automation testing were executed.

> Manual and Automated testing which works well with Continuous Integration (CI) and Continuous Deployment (CD) methodologies of Agile.



## Benefits

- > This Migration helped customer to gain dramatic improvements over its older Legacy monolithic applications, including:
- Reduced Latency by utilizing newer optimal technologies in the developed solution.
  - Independent scaling – each micro service now scales independently based on load parameters
  - Independent upgrades - each service can be deployed independent of other services.
  - Smooth user-friendly web/app experience for end users by utilizing the updated Digital wallet during checkouts.