techolution

Taking your business to the cloud and beyond!

Largest US Bank & 7th Largest Bank in the world

CASE STUDY

Overview

As the largest bank in the US and seventh largest in the world, this bank must constantly improve its customer experience via development of internal and customer facing applications. With the need to remain competitive with agile Fintech startups, the bank needed a way to speed time to market for application development. The bank wanted to modernize their legacy-based applications, reduce the impact of technical debt with legacy mainframe infrastructure, and ensure integration across a modern cloud architecture for CI/CD. **The overall goal of this was to speed up the application development release cycles to enhance mobile customer user experience and increase operational agility.** This would be necessary to retain their position as a banking market leader capable of competing with agile Fintech startups.

Challenges

With over 30,000 developers managing thousands of tightly coupled applications, this bank was dealing with a slow and cumbersome release cycle schedule. This was primarily due to complex application dependencies and data flow challenges between legacy mainframe infrastructure and the network edge . With quarterly release schedules to bring order to highly coupled applications across thousands of developers, this bank was facing bottleneck challenges with application updates, complex dependency matrixes, and weekend based releases.

While this bank had implemented a robust framework meant to decouple applications for faster and more frequent applications update cycles, it was not delivering the desired results. The Techolution team reviewed and then helped create a spring-based application framework to ensure that application teams could be autonomous in their SDLC allowing for decoupled applications with independent release cycles.

This framework baked in a lot of the complex repeatable this bank specific operations into it so that every application development team conformed to the same set of standards and they didn't have to reinvent the wheel every time. The framework also enabled auditing of transaction events and local data stores that would increase developer autonomy through complete data access and complex transaction event fulfilment. This included data encryption and decryption methods for data flow from the legacy mainframes to the edge. The overall solution was based on taking bank's application data flows from synchronous-asynchronous.

The goal of the modified framework applied to the bank framework was to make every incoming and outgoing data transaction request between the cloud edge and the legacy compute architecture an event. By creating an asynchronous service based on an event queue approach, legacy compute architectures that had finite simultaneous event fulfilment capabilities would not crash due to synchronous request bottlenecks.

A major part of making this asynchronous service a reality by the Techolution team was that the framework enabled creation of local data stores. This removed data dependencies and searches of application master records that would support the asynchronous services process between the legacy architecture and the many cloud application edge points.

Results

Techolution worked closely with the bank to deploy a modern, cloud-native digital banking platform that significantly improved the user experience for mobile banking customers. By implementing full automation and compliance within a modified CI/CD pipeline, Techolution was able to deliver a modified development framework that resulted in:

- Decoupling of applications allowing for independent release cycles, reduction of integration testing time, and autonomous application development teams.
- Shrinking dev cycle from previous quarterly release cycle update windows to a far more flexible and timely schedule of every other week.
- Eliminating 80 percent of the prospective cloud-targeted applications for migration and moving only the 20 percent that would provide the biggest benefit to the business, which saved countless weeks and hours for migration along with associated massive migration costs
- Creation of an audit system within the framework to ensure asynchronous event data flows completed successfully or alert stakeholders when they did not
- Enabling connections to potentially dozens or even hundreds of different systems for event data access that enable new application access and transaction event opportunities for new customer services. This would set the bank up for agile and unlimited services growth in real time based on identified opportunities.
- Major PII/PCI and proprietary data security improvements for data in transit via Implementation of asynchronous data flow encryption and decryption. This provides simplified regulatory auditing and increased customer loyalty through data safety.
- Training and placement of developers to run and maintain the modified framework that could be placed in-house within this bank locations where there was a low availability of access to developers processing this expertise and experience. The result was avoiding time loss for massive recruitment searches, unworkable salary requirements based on needed developer experience, and developer team compatibility integration risk with the broader developer, IT, and g this bank teams.
- Shifted data to the edge, allowing for reduced dependency on legacy system, quick retrieval time of data since data is close to the request, and built in data replication.

> *This bank is now rated as the top digital customer experience bank in the US. In addition to this UI enhancement, this bank is also experiencing faster time to market and has secured new IT cost efficiencies as a result of modernizing their entire mobile banking infrastructure.*