

Microservice Adaptation for Fintech – A necessity

A Blueprint to Transform Monolith to Microservices

Practice Head:

Aju Mathew

Enterprise Application Development

Author:

Chandramouli Parasuraman

Architect



INTRODUCTION

The demands on modern Fintech companies have been increasing dramatically with the advent of the digital revolution. Every company has to have a strategy in place to scale its business with the help of better designed software. That implies the architecture of such a software product cannot be left behind, failing which the potential **impact on cost, time and profitability** will be huge.

Microservices is one such architectural strategy the Fintech companies could adopt to reduce the risk of being left behind. Customers of modern fintech products have great expectations. Extensive personalization across the digital platforms have completely changed customers' behavior, who now expect everything to be digital.

Fintech companies are supposed to be modern in its outlook and ahead of curve in its product architecture. But here too, pockets of inefficiencies are floating around which doesn't augur well for a sunrise industry.

One of the main reasons they might find it difficult to move ahead is their old monolithic architecture. In this article, we'll look at a **blueprint** for transforming a monolithic application to microservices for fintech as part of their long term strategy.

NEW REALITIES IN FINTECH ECOSYSTEMS

Legacy applications are of different shapes and sizes. But one thing that most of them have in common is their **monolithic** nature. Though different sub-systems are neatly layered, they still represent an old way of thinking that doesn't scale well to contemporary demands.

Microservice Adaptation for Fintech – A necessity

Given that Fintech is an offshoot of the broader financial services industry, it also seems to have carried some of the baggage from the legacy systems. Most often, there are central databases with tightly coupled sub-systems and security implementations. To make them work for a modern endeavor like Fintech requires systematic and dramatic changes to them.

Such change need not be worrisome. This article gives enterprises a blueprint to go about such a significant undertaking which could have large scale implications in the organization. But the end result would be a nimble ecosystem that adapts well to the changed realities.

The blueprint covers various aspects of an application in their own sections. This way, a CXO will be in a position to have a bird's eye view of each aspect and could then work with a specialist company who could take it further.

AGILE DESIGN AND CULTURE

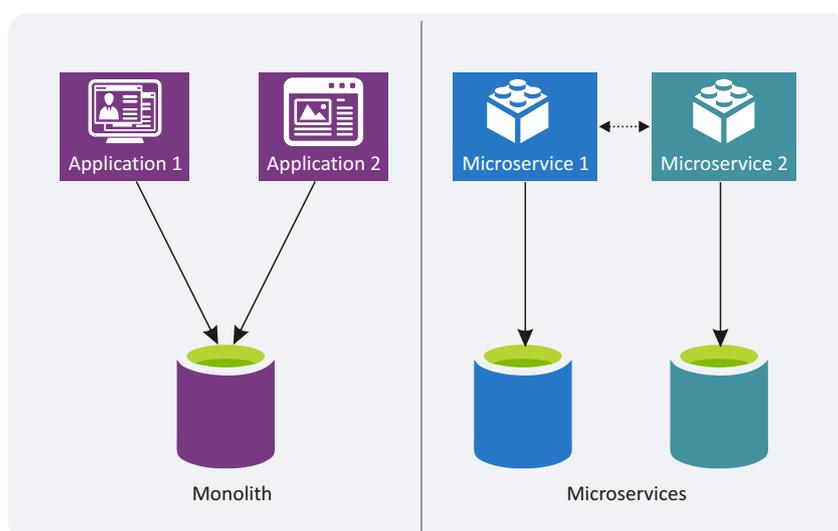
First and foremost, the design of a microservice should be around the business functionality rather than technology layers like UI, middle-tier and database. Since a microservice is a self-contained unit, it is deployed and managed independently. Hence, it is imperative to make it serve a specific, well-defined business functionality.

That reorientation of design also brings with it certain cultural challenges. Earlier where all middle-tier developers and DB developers who were in their own world have to now belong to a team that is cross-functional and highly interdependent on each other within the team. On the bright side, this enables quicker product development, motivated developers and widespread business knowledge across the team.

UNIQUE DATABASE STRUCTURE

Database is central to the system architecture of many software products. Typically, it is also one of the bottlenecks in a monolithic architecture based implementation. Most systems have a single large database that is used by multiple applications. When complexity increases in the system, this becomes a huge bottleneck.

Microservices approach recommends having a separate database for each service. When a new service is deployed, it also has its own database setup. Whenever changes are made to some other service which is part of the whole system, this database doesn't get affected. If the data from the other service has to be consumed, the service itself exposes it through an endpoint or through a simpler middleware platform.



With this freedom, the Fintech product could even choose polyglot persistence (i.e. different database software for different kind of requirements). A large-scale logging service could be implemented with a NoSQL based database like MongoDB that is excellent for such write-heavy scenarios, whereas payment services could still be implemented with an RDBMS database like MySQL.

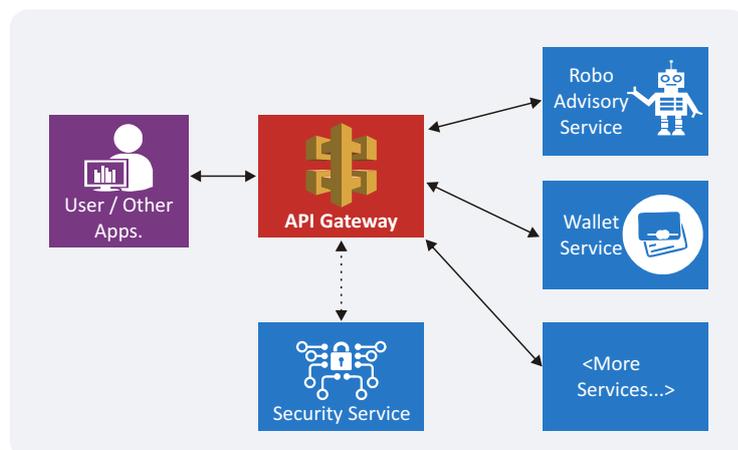
Microservice Adaptation for Fintech – A necessity

ENHANCED SECURITY

Modern Fintech requirements put forth newer challenges to the security infrastructure of the Fintech products. Features like digital wallet, robo advisory, blockchain and so on mandate that a solid security mechanism is in place.

To this end, Microservices implementations across the world follow a best practice of creating a separate identity service to handle such scenarios. This could be an OAuth2 authorization service that helps in authentication of users and authorization of access to resources in various other functional micro-services.

An API gateway sits in front of those services and coordinates request. This gateway is also the one which handles security by making use of the OAuth-based security service. This also means the individual microservices that deliver features could be completely internal without being exposed to the public, thereby bettering the security of the whole system.



SEAMLESS INTEGRATION

Integration layer is the nerve center in a Fintech product which has to communicate with many other financial services exposing their API in various formats. When applications of monolithic nature are integrated, an ESB (Enterprise Service Bus) software is typically used to manage the integration. The goal of an ESB software is to integrate disparate systems in a technology neutral manner. At some point, this software itself becomes so huge and complex to manage and the IT departments ends up wondering why they are not able to smoothly transition to something simpler, given the fact that they are in a vendor lock-in with the ESB itself.

The general practice in a Microservices ecosystem is simple REST-based or SOAP-based communication. Where required, a simple messaging platform like ZeroMQ or RabbitMQ is used which is not too heavy for an integration platform. Also, areas which were initially part of the ESB stack such as transformation, management, schema-validation etc. could themselves be written as micro-services in their own right or redistributed across an apt service, thereby reducing bloat and duplication.

FASTER DEPLOYMENT AND SCALABILITY

Being fast and nimble in bringing features to the market is one of the cornerstones of new age fintech companies. Having a solid deployment architecture gives them the required edge.

Microservices are a natural fit when it comes to having such an architecture in place. By their nature, microservices scale well to larger sizes in terms of users and transaction volumes. These services are not only easier to distribute across instances, but also due to their independent nature are technology agnostic.

“POLYGLOT” – FOR TECHNOLOGY CHOICES

A significant benefit microservices offer is the freedom in the choice of technology with which the services are developed. A fully transaction-oriented service could be developed with, say, Java and MySQL backend; whereas a high traffic chat service could be developed using Erlang and MongoDB.

It is therefore important for the Fintech companies to choose the right technology while developing a microservice. “Polyglot” need not be a term associated only with databases, but also to language or runtime choices in a microservices ecosystem!

Microservice Adaptation for Fintech – A necessity

CONCLUSION

Fintech is paving the way to newer possibilities in the digital arena. As trailblazers in the world of finance, Fintech companies should be at the forefront of the revolution. Adopting microservices as a long term strategy would only help in making this happen for them.



ABOUT ASPIRE

Aspire Systems is a global technology services firm serving as a trusted technology partner for our customers. We work with some of the world's most innovative enterprises and independent software vendors, helping them leverage technology and outsourcing in our specific areas of expertise. Our core philosophy of "Attention. Always." communicates our belief in lavishing care and attention on our customer and employees.

SINGAPORE
+65 3163 3050

NORTH AMERICA
+1 630 368 0970

EUROPE
+44 203 170 6115

INDIA
+91 44 6740 4000

MIDDLE EAST
+971 50 658 8831

For more info contact
info@aspire.sys.com or visit www.aspiresys.com

