



# MICRO FRONTEND APPROACHES IN DIGITAL BANKING

## Abstract

With the pandemic there a rise in need for faster feature-based releases in digital banking. Rapid feature releases require a flexible deployment of user experience across channels. as Omni-channel Digital Banking Experience becomes more and more relevant during these trying times, banks need rapid delivery of experiences across these channels to ensure maximum customer engagements, the mobile applications, online frontends should be altered and upgraded all the time. Banks have adopted distributed architecture through micro-services, they are subsequently extending the same pattern to the End-User Experience channels through Micro Frontends. Micro Frontends in digital banking can be proven as a crucial one when introducing a new feature in the e-market, with least effort and interference to the existing features.

This paper explains about the Micro Frontend approach in digital banking for front-end orchestration and routing with an Agile developing model.

# 1. Introduction

The need of the hour is to accelerate speed to market which demands companies to innovate in a much more agile and flexible way than before. A product based on micro frontends is considerably more agile as it is extremely modular, and each module is focused on specific business functions rather than specific technologies. It allows each

Team to define their respective objectives and deliver features through micro frontends. This independence means that organizations can define a release cycle or continuous delivery system depending on the individual product / features requirements. When it comes to banking systems, due to the competitive nature the banks are always updating their

digital services and products with the latest technologies to satisfy their customer's need. Since Most banks today deliver products / features through monolithic user experience across channels in a waterfall delivery model, it's becoming increasingly difficult for them to do hyper-productive engineering in an Agile Delivery Model.

## 2. Micro Frontends

### 2.1 What is micro front end

Micro frontend is basically an extension to microservices distributed architecture pattern, where concept of modularity and nature of distributed modular components is extended to the user-experience engineering across channels. The current trend is to build a feature-rich and powerful browser application, also known as single page application (SPA), which sits on top of a micro service architecture. Over time the frontend layer, often developed by a separate Team, grows, and gets more difficult to maintain. As

a result, micro frontends bring a wide range of advantages, break it down to specific advantages as bullet points.

### 2.2 Why micro frontend approaches in Digital banking

The rasping situations during the epidemic have created an absolute trend where multiple technical forecasts quickly converted into actuality. Traditional banking services annihilation is one among them. The adoption of digital banking came when lesser people are visiting the banks. The

conclusion is to allure and hang on to the customers, bank should be delicately worked to alter the existing digital services and add new services, which are amenable in this current world. However, the process of improving digital services make it tough to provide sustained user experience, as building new functionality is process need lot of time. The applications used in the banking industry have been monolithic for a long time. Additionally, in matter of monolithic user experience applications across channels, it would a time taking process.



Fig 1.0 Micro frontends

### 3. Front end design with Micro-front-end approach in Digital Banking

#### 3.1 How to design micro frontend and burst the existing monolithic design?

There should be an essential architectural design to break the monolithic design to micro frontend. The proper design ensures easy roll out of new features and application modernization. While upgrading the monolithic design to micro frontend there are basically two approaches to break down the modular user experience design such as horizontal -Used across product features and Vertical Experience -specific to product feature. The choice of approach will decide and determine the front-end team's responsibility. Both the approaches have its own pros and cons. The vertical approach is most used approach when it comes to converting monolithic to micro frontend, usually the breaking down of the modules is based on the product specific features of the monolithic web application. The horizontal split allows you to decompose an interface into various parts that can be assigned to different teams. For example, in the banking applications account related information is usually available in all the pages, however, the horizontal split is the preferred choice for applications that user similar components beyond multiple views. So vertical split will not make any sense here. There's decreased dependency between product feature owners, who can now deliver their individual features without having to depend on other feature releases.

The vertical split allows you to organize business domains and assign individual domain to separate teams. Here each and every team would be responsible for the whole interface, it'll moderately obtain more expertise in that business domain. The best example for vertical split in banking domain is credit card management interface. Since there is a no dependency between back end and front-end teams, the possession of each banking feature will be entrenched end to end and make sure that the feature has better quality and

lesser development time. In this approach the frontend of monolithic banking page can be divided into Team Loan, Team Payments, Team Accounts, Team all type of Cards, Team Insurance, Team Mutual funds etc. (Fig 1.1) so that each Team will own the feature of application and they will be responsible for developing, testing, and rolling out their respective feature. The Team should follow an Agile model such that it will be easy to deliver features in a shorter interval and any changes required at that time can be easily rectified. In the current trend we could see that the banks are developing different applications for different purposes such as for seeing passbooks, mobile banking, net banking, UPI banking etc. With the micro-front-end architecture, modular independent user experience features can be easily consumed across applications or channels

#### 3.2 How to implement the micro frontend approach in banking applications through Agile Model?

Micro frontend enables developers to integrate various modern frontend technologies together. Usually in banks there are multiple applications to serve the different services. One approach can be combining these services together to form a new application which will follow a vertical structure or can divide the application based on the functionality dependency like insurance, card, payment, loan etc. as a horizontal structure. In most of the banking software's the back end has already been developed with micro services and frontend

is monolithic, so upgrading the frontend also with micro teams and each Team to be collaborate with their corresponding micro services makes more sense. The key feature in this approach is that each Team can use their own technology to implement the product which will not disturb the other teams. This architectural technique ensures to follow agile model with smaller software release cycles, with easy roll out of feature, automated testing and continuous delivery allows innovating and allows your ideas to reach the market faster.

#### 3.3 How micro frontends can assist banks to accelerate their app modernizations and maximize customer engagements

When the development teams get a chance to blend various significant technologies smoothly and those might be of different business requirements, banks obtain their upgrades and scalable applications quickly and each Team can build and test their services, independent of another teams' progression in their work. The absence of interconnection between the development teams significantly reduces both building time and testing time, which allows banks to provide users with better services rapidly. Micro frontend allows individual widget or piece of frontend to be added removed or update without rebuilding others. If a huge number of independent people work in parallel to make the existing application better, the modernization process will be finished very quickly.

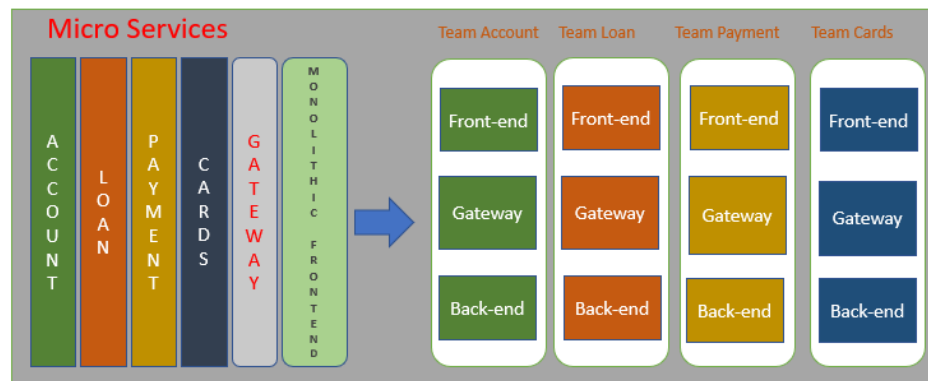


Fig 1.1 Monolithic frontend to micro front end

## Pros of Micro frontend

- Faster development, as teams can work independently. One of the main features is that small teams carry part of application and are responsible for building, testing, and deploying applications. Each Team will have their own development cycles in agile model without disturbing the other teams. There might be continuous delivery from each micro team for their feature updates.

- Product agnostic development
- Micro frontends can make sure that the teams are truly agile and cross-functional with full-stack skilled developers.
- Codebases are compact and can manage easily.
- Easier hiring of experts. With micro frontends, you look for professionals to work on a specific part of an app where a particular tech stack is used, so you do not need them to know technologies

that other teams use; The micro frontend can be developed by traditional front-end developers, there is no need of full stack developers.

## Cons of Micro frontend

- Building standards around multi-technology usage, upgrades can sometimes be difficult. So, choices are to be made depending on the complexity of the feature and how the same gets consumed.

## Conclusion

In this digital banking era, to compete with their rivals, banks should be able to preserve their competitive nature and retain the digital users engaged with new arrival of technologies and enhanced features. They should update their digital services constantly to hold the users with the bank. Because of these reasons top employees taking heeds on the modernization of the products and they prefer easy roll out of products instead of long implementation and testing cycles. Micro frontends are designed for solving these problems.

The perfect architectural design improves security level at frontend as well as eradicates obstacles in front-end improvisation by turning the bigger applications into small pieces more controllable ones.

## References

- <https://micro-frontends.org/>
- <https://www.qulix.com/about/micro-frontends-in-digital-banking/#:~:text=The%20micro%2Dfrontend%20approach%20implies,managed%20by%20different%20teams%20independently.&text=Such%20an%20absence%20of%20interdependence,users%20with%20improved%20services%20quicker>

## About the Author



Vidhya Radhakrishnan Chandrika  
Technology Architect



For more information, contact [askus@infosys.com](mailto:askus@infosys.com)



© 2022 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.