

# Migration process from monolithic to micro frontend architecture in mobile applications

International Workshop on Smalltalk Technologies 2023

Lyon 31/08/2023

**Quentin Capdepon**, Nicolas Hlad, Abdelhak-djamel Seriai and Moustapha Derras

# Introduction

- This presentation consists of three main sections:
  - **Contextual Background and Problem Statement**
    - This section provides the context surrounding our work and the issues we are addressing.
  - **Previous Experiments at Berger-Levrault**
    - In this section, we explain the implementations and experiments conducted at Berger-Levrault, along with their corresponding results
  - **Proposed Problem Solution**
    - In this section, we present our proposed migration process to resolve the identified problem.

**01**

# Contextual Background and Problem Statement

Introduction

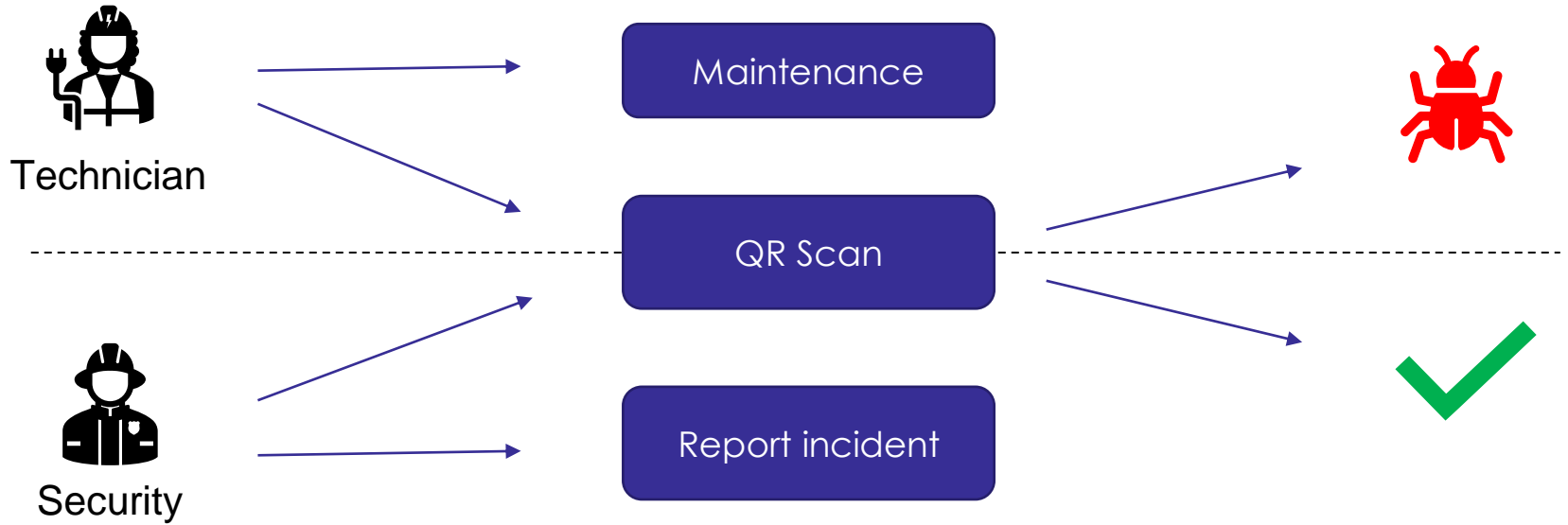
# CARL Touch



Cross-Platform Computerized Maintenance Management System (CMMS) Application for iOS and Android

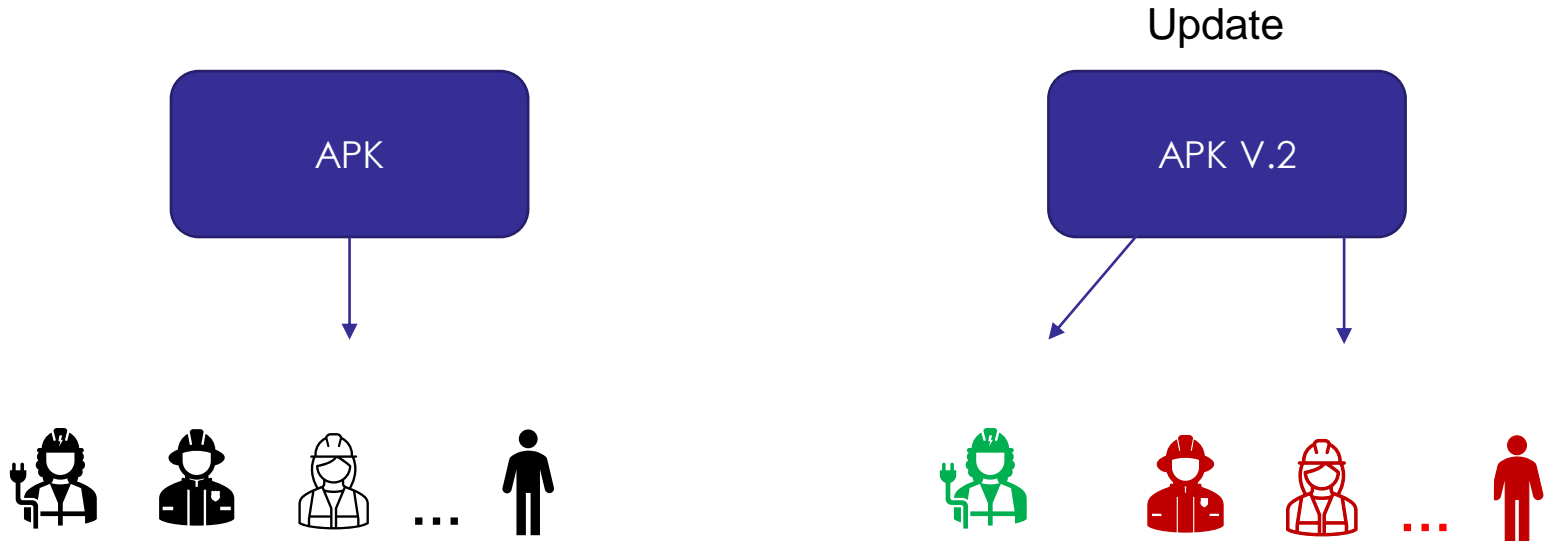
# Industrial issues encountered

- A maintenance and issue



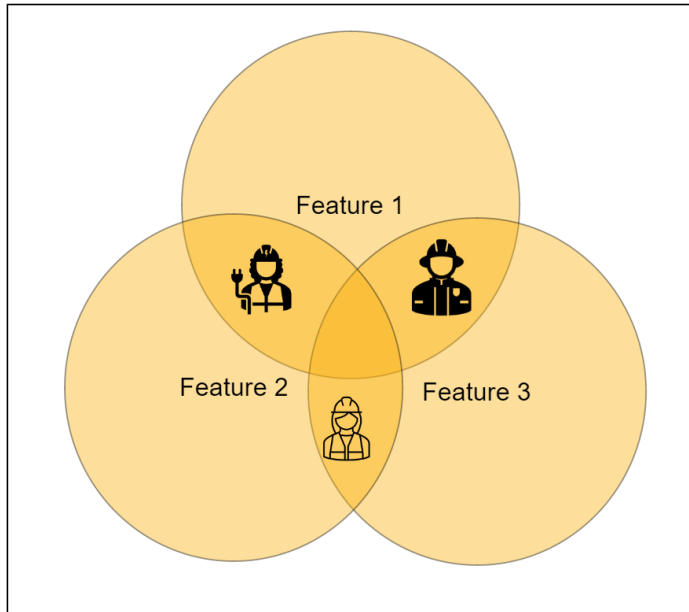
# Industrial issues encountered

- A customisation of the app issue



# What architecture to solve the prior issues ?

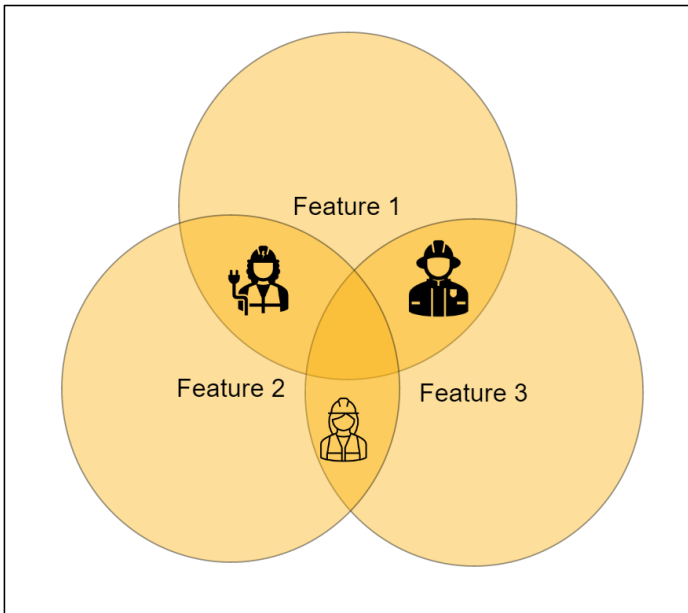
- Our current architecture



One same monolith for everyone

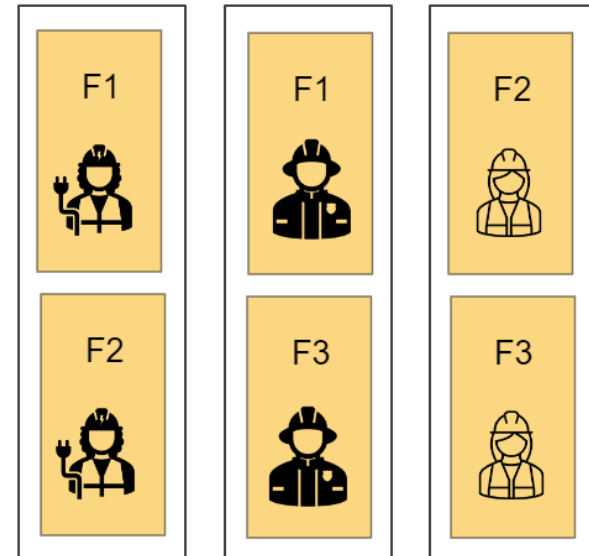
# What architecture to solve the prior issues ?

- Our current architecture



One same monolith for everyone

- Our target architecture



A moduable architecture to obtain tailored apps



State of the art

# What is a micro frontend architecture ?

- The MFE architecture

## End-to-End Teams with Micro Frontends

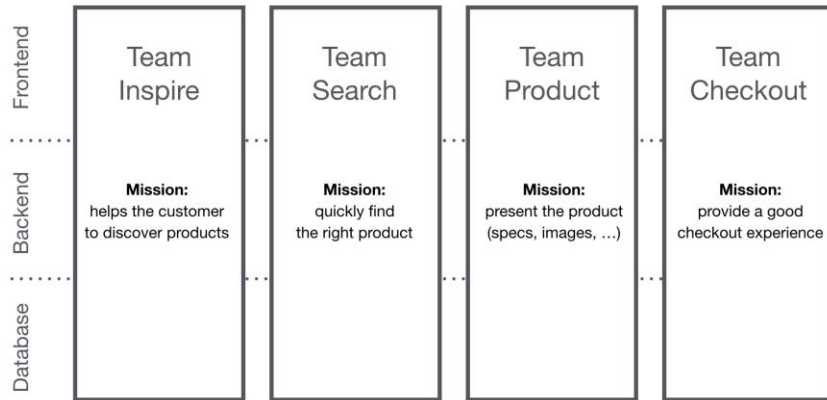


Figure 1 : Organisation in Verticals [1]

# What is a micro frontend architecture ?

- The MFE architecture

## End-to-End Teams with Micro Frontends

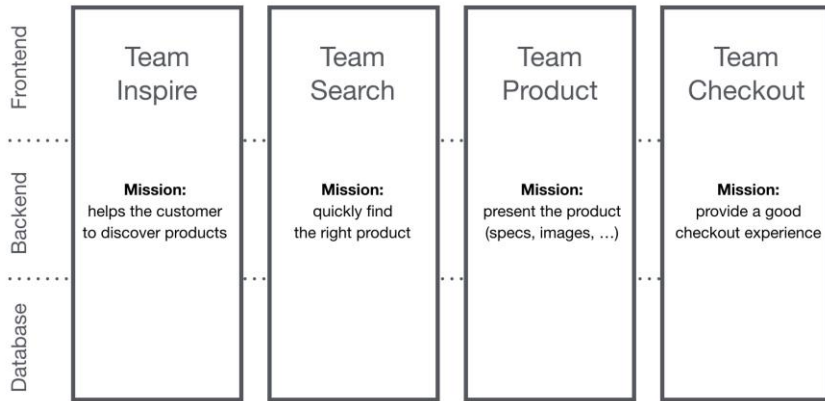


Figure 1 : Organisation in Verticals [1]

## ✓ Benefits :

- Incremental upgrades,
- Simple and decoupled codebases,
- Independent deployment,
- Autonomous teams,

# What is a micro frontend architecture ?

- The MFE architecture

## End-to-End Teams with Micro Frontends

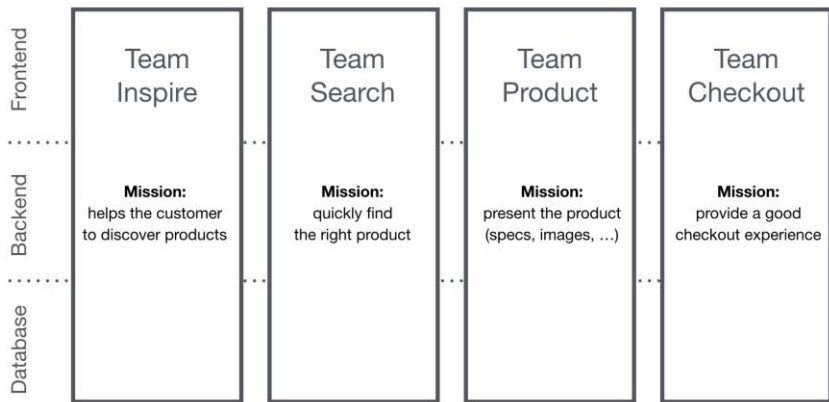


Figure 1 : Organisation in Verticals [1]

## ✓ Benefits :

- Incremental upgrades,
- Simple and decoupled codebases,
- Independent deployment,
- Autonomous teams,

## ✗ Downsides :

- Payload size,
- Environment differences,
- Operational and governance complexity.

# How does the MFE architecture work?

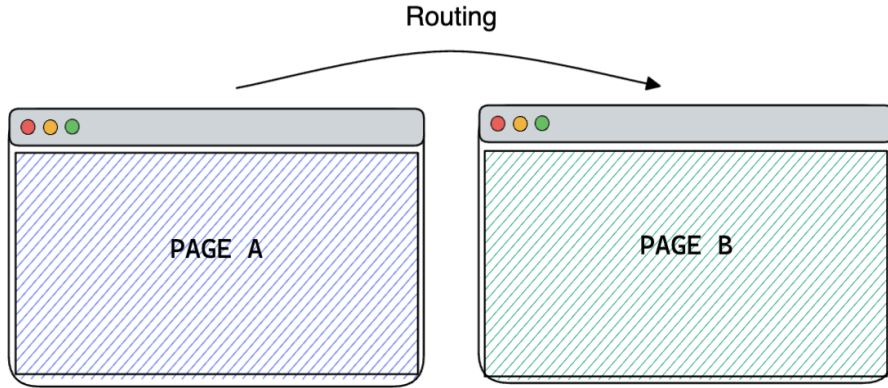


Figure 2 : The three fundamental concepts of Micro Frontend [1]

# How does the MFE architecture work?

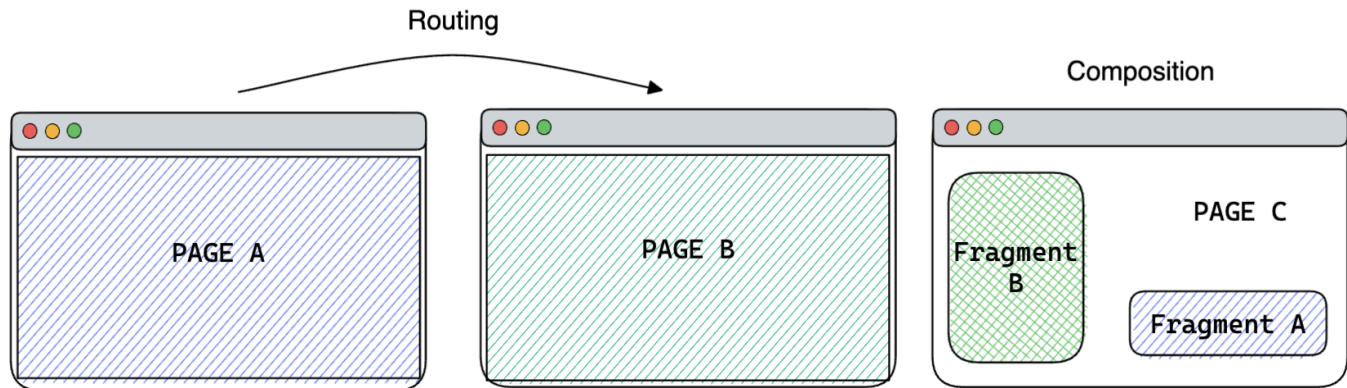


Figure 2 : The three fundamental concepts of Micro Frontend [1]

# How does the MFE architecture work?

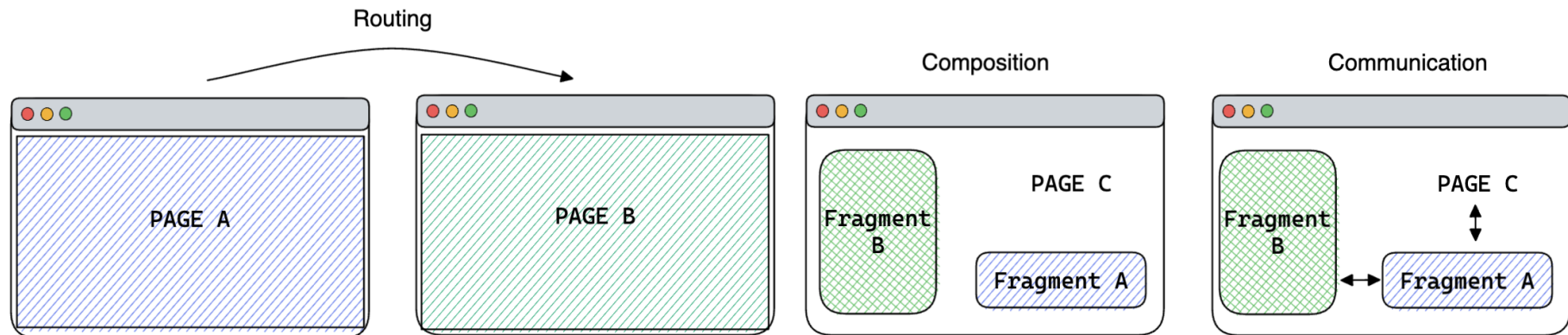


Figure 2 : The three fundamental concepts of Micro Frontend [1]

# Engineering & Research questions

- **Topic 1 : Engineering questions**

- How to achieve MFE principle on mobile ?

- **Topic 2 : Research questions**

- How to migrate monolith architecture to micro frontend for mobile applications ?

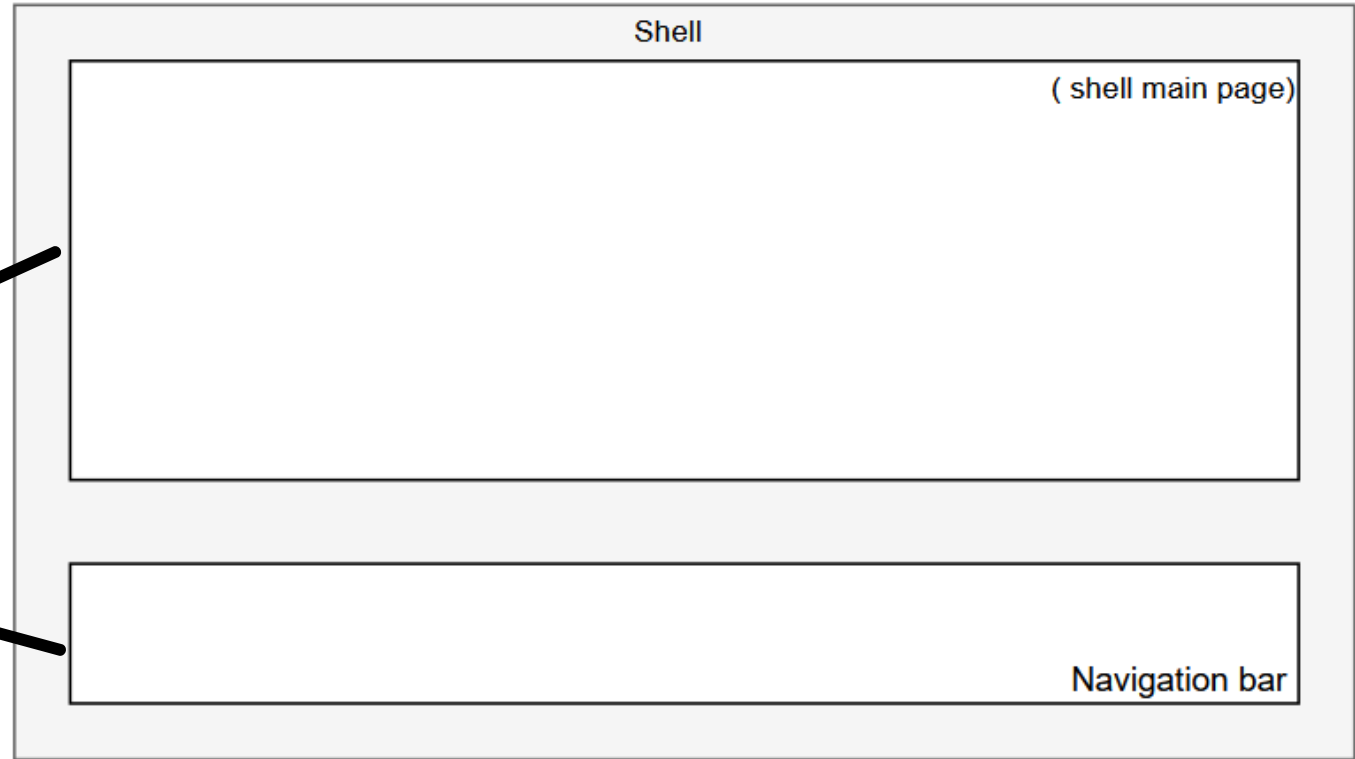
**02**

# Micro frontend on mobile



Our mobile micro frontend shell

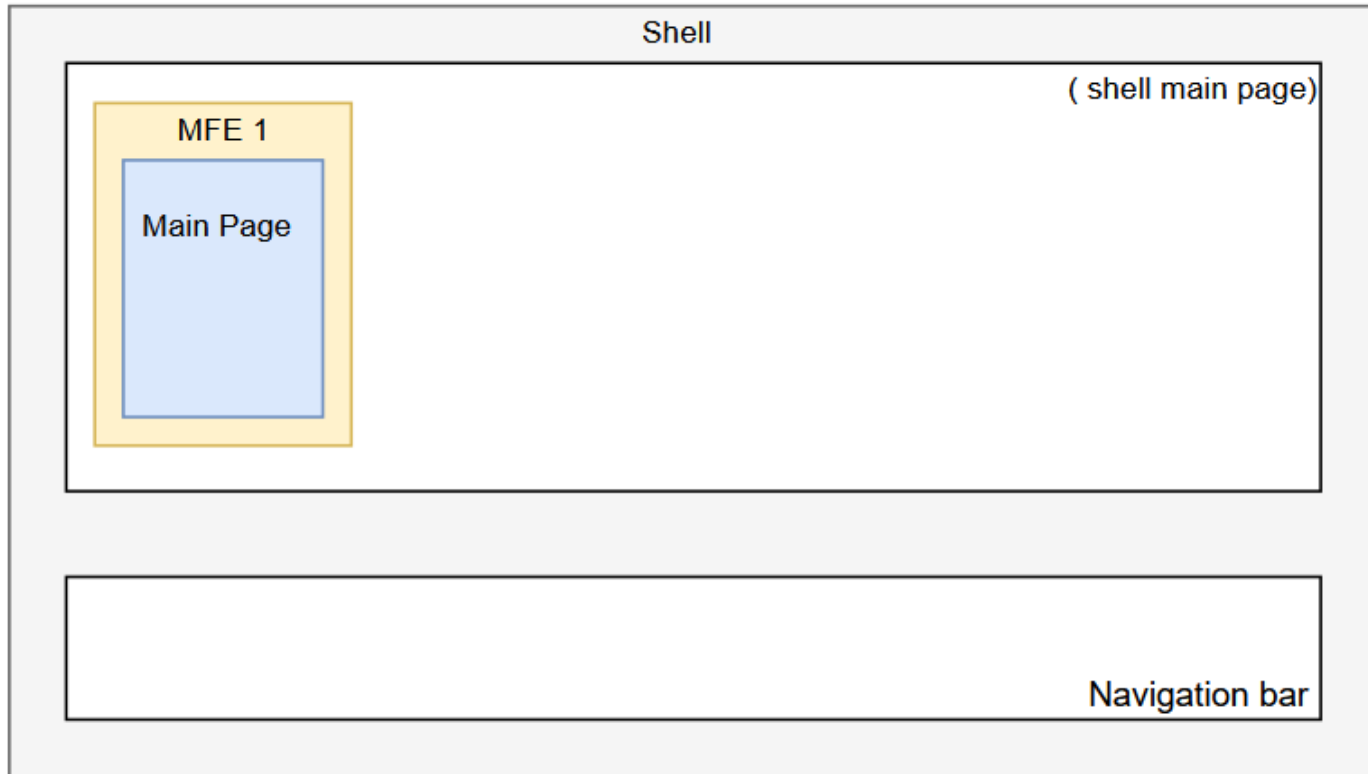
# BL MicroFrontendShell



CarlTouch.app

Our mobile micro frontend shell

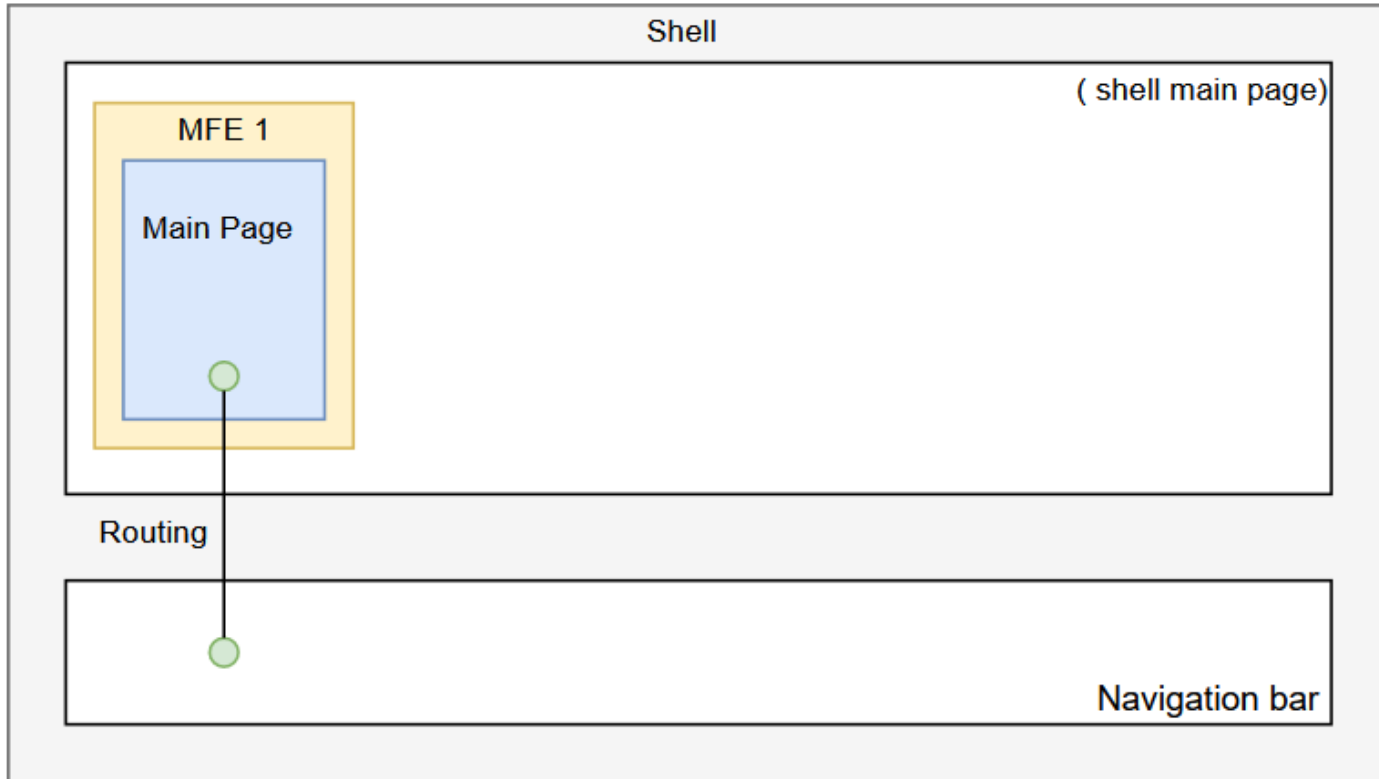
## BL MicroFrontendShell



CarlTouch.app

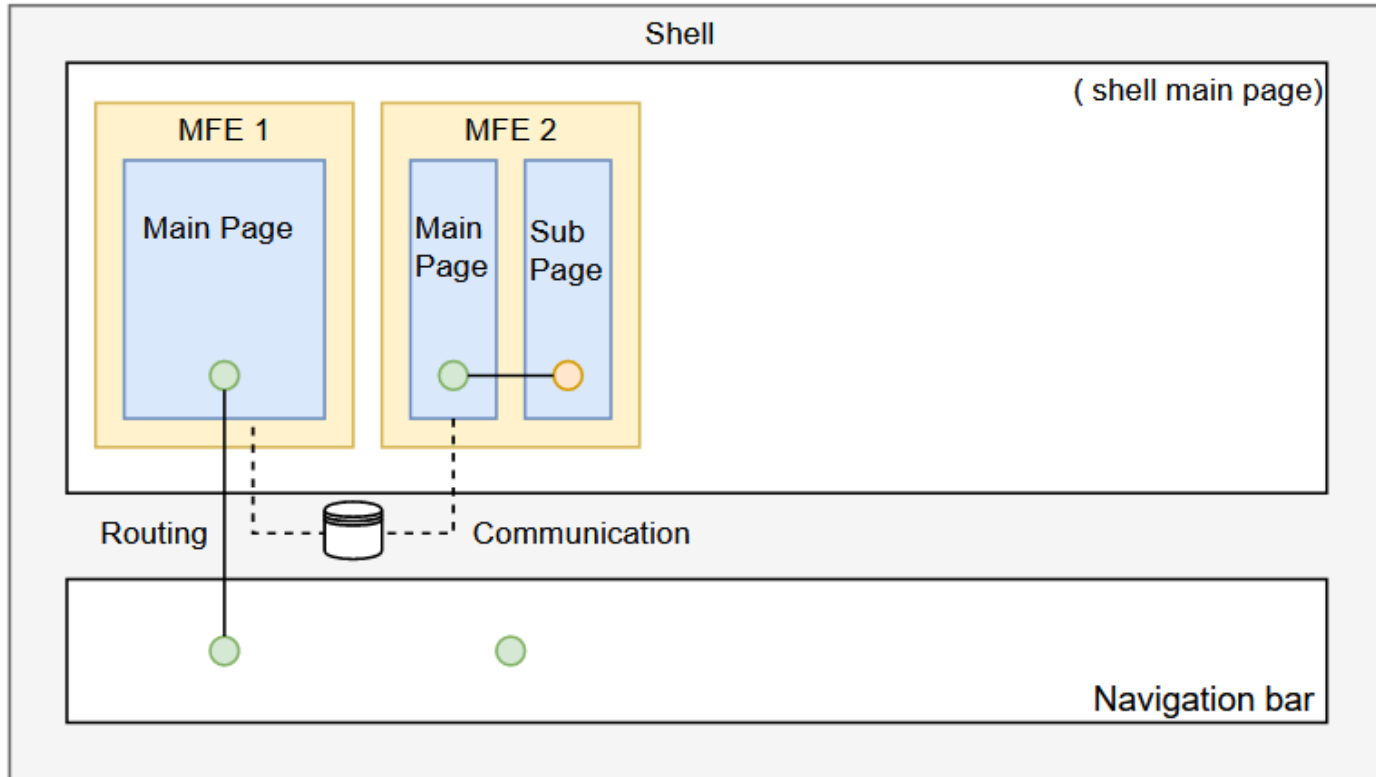
Our mobile micro frontend shell

## BL MicroFrontendShell



Our mobile micro frontend shell

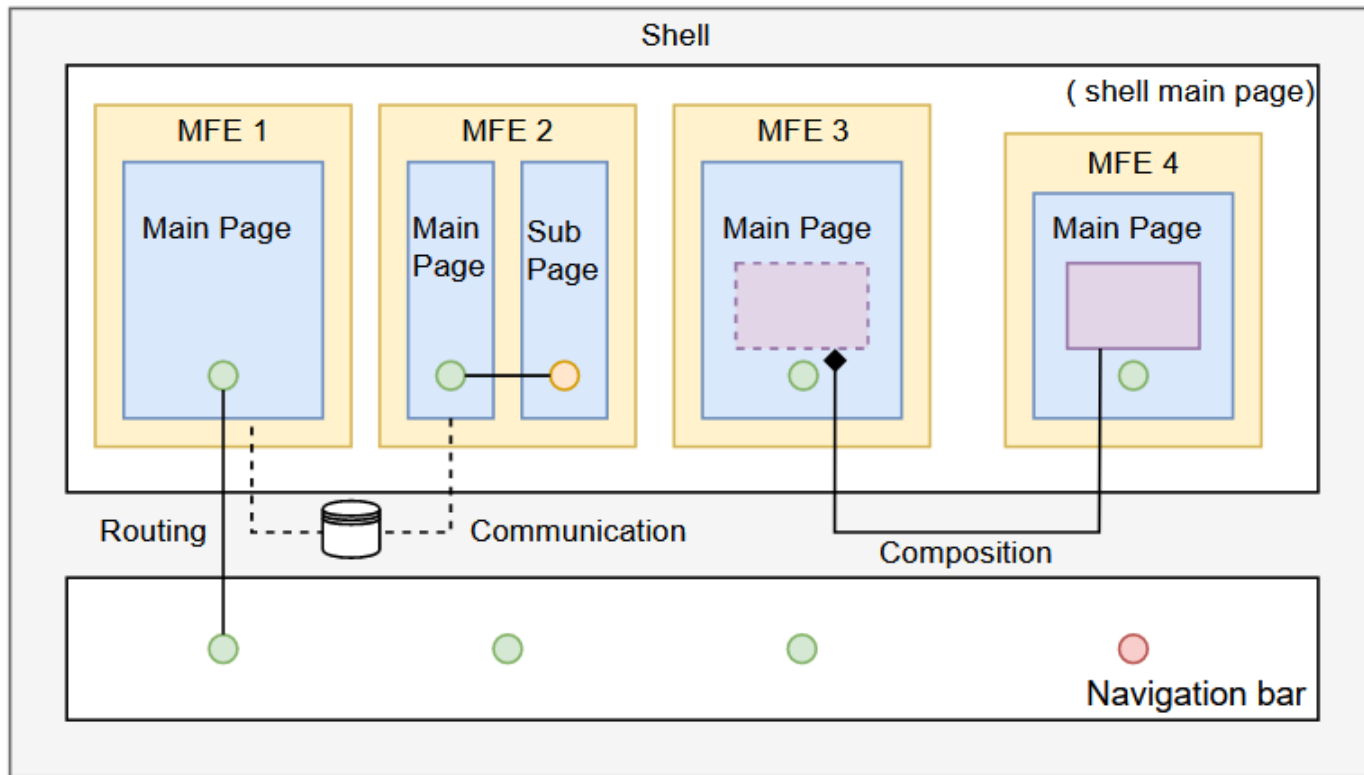
## BL MicroFrontendShell



CarlTouch.app

Our mobile micro frontend shell

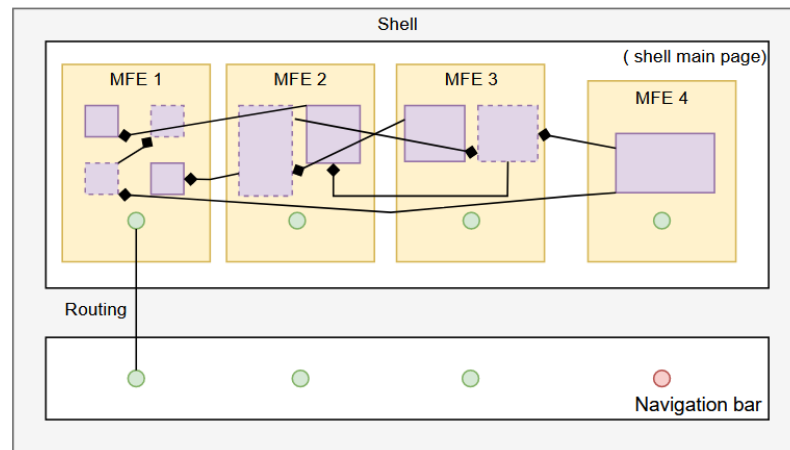
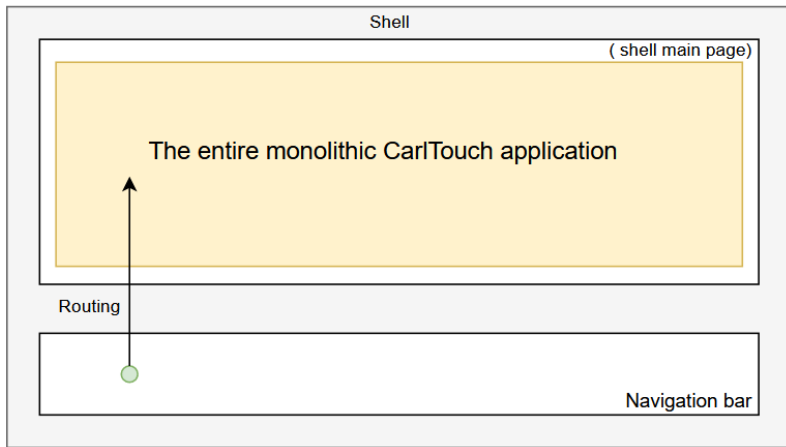
## BL MicroFrontendShell



CarlTouch.app

## Experimentations

- Two teams are responsible for migrating the Carl Touch monolith to MFE using our Shell (in Flutter).
- We obtain 2 different architectures



**Observation:** Developers encounter difficulties in identifying 'Optimal' MFEs and struggle implementing them from the monolith.

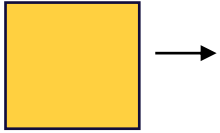
**Goal :** Developing a semi-automated migration process from monolith to MFE

**03**

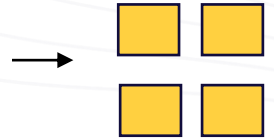
# Migration process

# Our migration process

monolithic



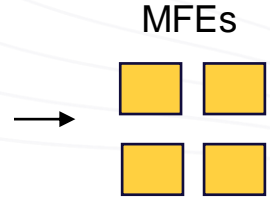
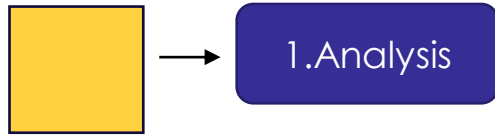
MFEs





# Our migration process

monolithic



## 1. Analysis

1. Dart Parser Development
2. Creation of a Famix Dart Importer
3. Creation and generation of a Dart Metamodel



# Our migration process

monolithic

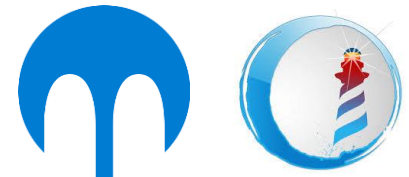


## 1. Analysis

1. Dart Parser Development
2. Creation of a Famix Dart Importer
3. Creation and generation of a Dart Metamodel

## 2. Identification *[9][10][11][12][13]*

1. Creation of Roassal visualization for the model
2. Proposed Clustering Approach for the Model
3. Interactive Roassal Visualization to help Expert Decision Making



# Our migration process

monolithic



## 1. Analysis

1. Dart Parser Development
2. Creation of a Famix Dart Importer
3. Creation and generation of a Dart Metamodel

## 2. Identification *[9][10][11][12][13]*

1. Creation of Roassal visualization for the model
2. Proposed Clustering Approach for the Model
3. Interactive Roassal Visualization to help Expert Decision Making

## 3. Transformation

1. Violation resolver to solve generic violation
2. Transformation Rules for MFE Dart code generation



# Our migration process

monolithic



## 1. Analysis

1. Dart Parser Development
2. Creation of a Famix Dart Importer
3. Creation and generation of a Dart Metamodel

## 2. Identification [9][10][11][12][13]

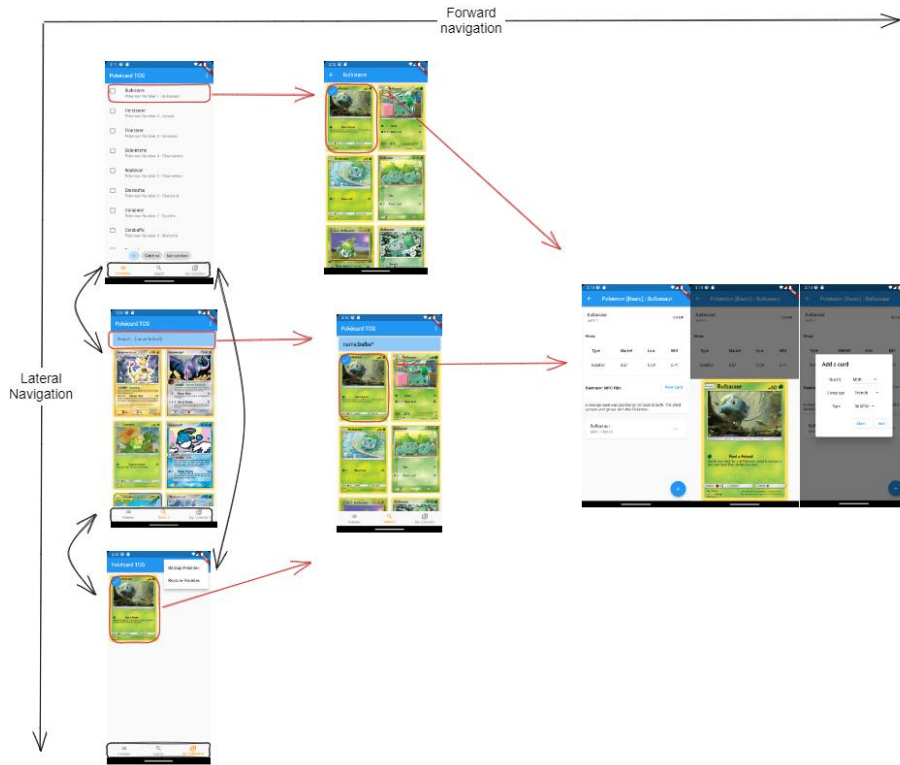
1. **Creation of Roassal visualization for the model**
2. Proposed Clustering Approach for the Model
3. Interactive Roassal Visualization to help Expert Decision Making

## 3. Transformation

1. Violation resolver to solve generic violation
2. Transformation Rules for MFE Dart code generation



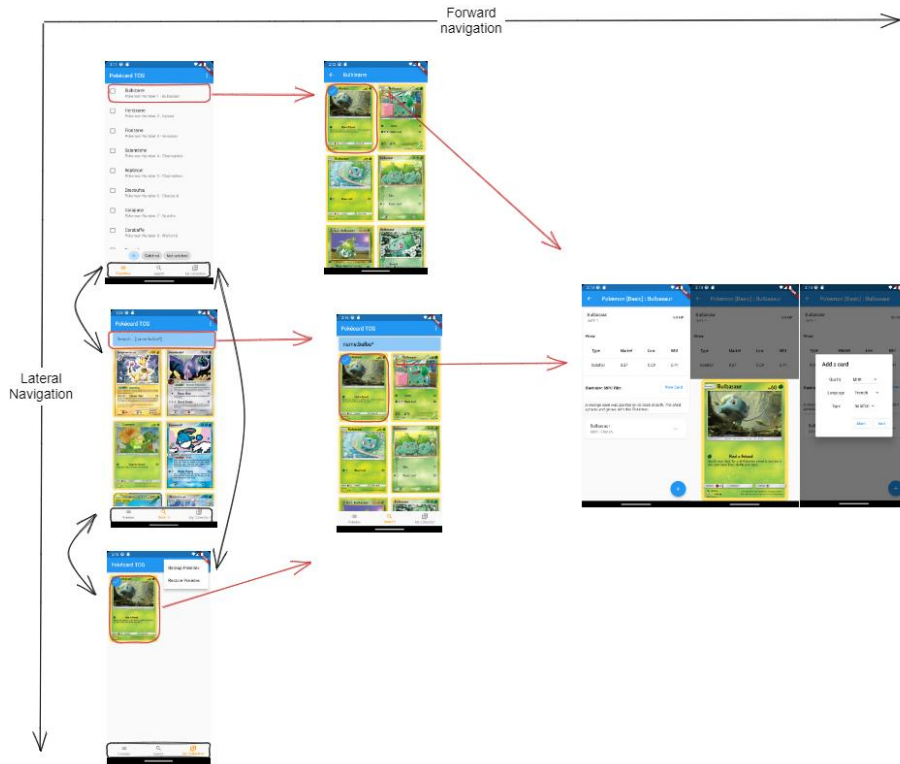
# Visualization : Navigation graph



## ✓ Benefits :

- Visual help on the application Navigation flow,
- User-Centric information for a transformation,
- Lateral navigation identify good MFE candidat,
- Forward navigation highlight inusable candidat.

# Visualization : Navigation graph



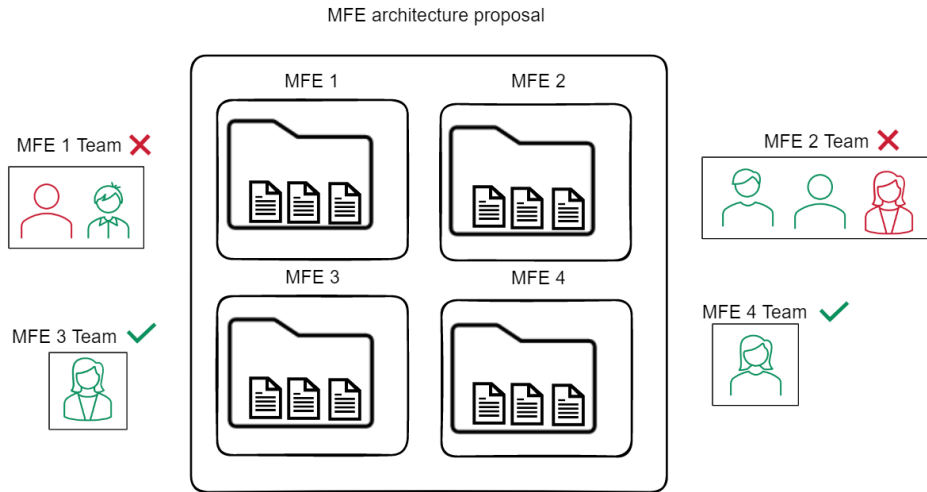
## ✓ Benefits :

- Visual help on the application Navigation flow,
- User-Centric information for a transformation,
- Lateral navigation identify good MFE candidat,
- Forward navigation highlight inusable candidat.

## ✗ Downsides :

- Not a clustering
- Require a dynamic and a static code analysis,
- Missing information on the communication.

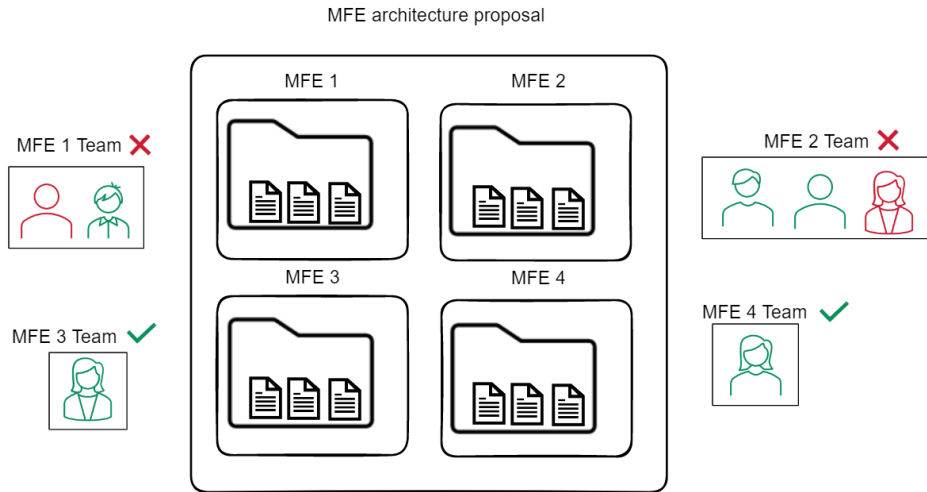
# Visualization : Git Contribution Analysis



## ✓ Benefits :

- Proposition of a team clustering,
- Provide insight on the best MFE suited team,
- Allocation of Dedicated Teams for Each MFE,
- Improved assessment of team profiles.

# Visualization : Git Contribution Analysis



## ✓ Benefits :

- Proposition of a team clustering,
- Provide insight on the best MFE suited team,
- Allocation of Dedicated Teams for Each MFE,
- Improved assessment of team profiles.

## ✗ Downsides :

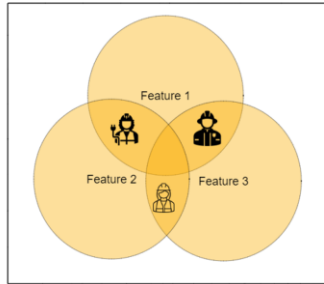
- Not a codebase clustering,
- Requirement for Prior MFE Clustering.



# Conclusion

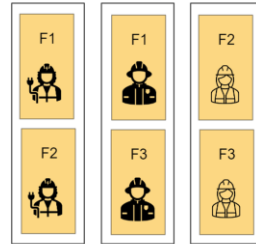
## What architecture to solve the prior issues ?

- Our current architecture



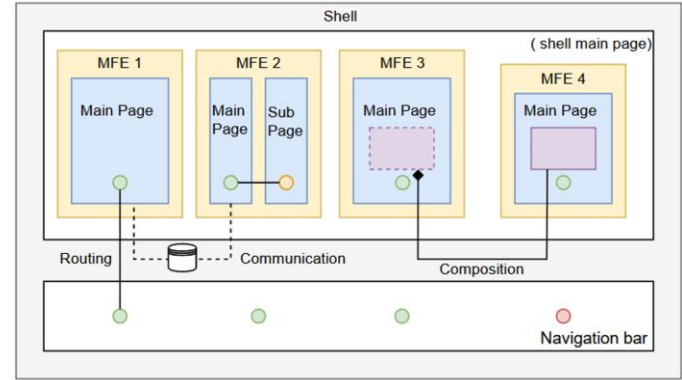
One application for everyone

- Our target architecture



A modulable architecture to obtain tailored apps

## BL MicroFrontendShell



## Our migration process



### 1. Analysis

- Dart Parser Development
- Creation of a Famix Dart Importer
- Creation and generation of a Dart Metamodel

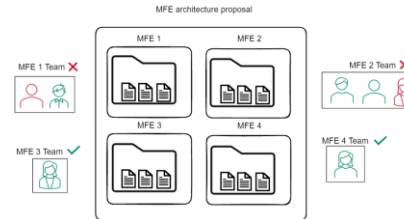
### 2. Identification [9] [10] [11] [12] [13]

- Creation of Roassal visualization for the model
- Proposed Clustering Approach for the Model
- Interactive Roassal Visualization to help Expert Decision Making

### 3. Transformation

- Violation resolver to solve generic violation
- Transformation Rules for MFE Dart code generation

## Visualization : Git Contribution Analysis



### ✓ Benefits :

- Proposition of a team clustering,
- Provide insight on the best MFE suited team,
- Allocation of Dedicated Teams for Each MFE,
- Improved assessment of team profiles.

### ✗ Downsides :

- Not a codebase clustering,
- Requirement for Prior MFE Clustering.

# Bibliography

- [1] M. Geers, Micro Frontends in Action, manning publications ed., 2020. URL: <https://livebook.manning.com/book/micro-frontends-in-action/>.
- [2] <https://micro-frontends.org/> by M. Geers
- [3] S. Peltonen, La. Mezzalira, D. Taibi, Motivations, benefits, and issues for adopting microfrontends: A multivocal literature review, Information and Software Technology 136 (2021) 106571. doi:10.1016/j.infsof.2021.106571.
- [4] <https://martinfowler.com/articles/micro-frontends.html> - Cam Jackson in 19 June 2019
- [5] E. Braz, Flutter micro app - a package to speed up the creation of micro frontend(or independent features) structure in flutter applications, 2022. URL: [https://web.archive.org/web/20220804142023/https://flutterrepos.com/lib/emanuel-braz-flutter\\_micro\\_app](https://web.archive.org/web/20220804142023/https://flutterrepos.com/lib/emanuel-braz-flutter_micro_app)
- [6] Ionic, Micro frontend architecture for mobile web apps - ionic portals, 2022. URL: <https://ionic.io/portals>.
- [7] D. Taibi, L. Mezzalira, Micro-frontends: Principles, implementations, and pitfalls, ACM SIGSOFT Software Engineering Notes 47 (2022) 25–29. doi:10.1145/3561846.3561853.
- [8] L. Mezzalira, Microfrontends anti-patterns: Seven years in the trenches, 2022. URL: <https://www.infoq.com/presentations/microfrontend-antipattern/>.

# Bibliography

- [9] M. Brito, J. Cunha, J. a. Saraiva, Identification of microservices from monolithic applications through topic modelling, in: Proceedings of the 36th Annual ACM Symposium on Applied Computing, SAC '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 1409–1418. URL: <https://doi.org/10.1145/3412841.3442016>. doi:10.1145/3412841.3442016.
- [10] F. Auer, V. Lenarduzzi, M. Felderer, D. Taibi, From monolithic systems to microservices: An assessment framework, Information and Software Technology 137 (2021) 106600. doi:10.1016/j.infsof.2021.106600.
- [11] A. Bucchiarone, N. Dragoni, S. Dustdar, S. T. Larsen, M. Mazzara, From monolithic to microservices: An experience report from the banking domain, IEEE Softw. 35 (2018) 50–55. doi:10.1109/MS.2018.2141026.
- [12] R. Capuano, H. Muccini, A systematic literature review on migration to microservices: a quality attributes perspective, in: IEEE 19th International Conference on Software Architecture Companion, ICSA Companion 2022, Honolulu, HI, USA, March 12-15, 2022, IEEE, 2022, p. 120–123. URL: <https://doi.org/10.1109/ICSA-C54293.2022.00030>. doi:10.1109/ICSA-C54293.2022.00030.
- [13] F. Freitas, A. Ferreira, J. Cunha, Refactoring java monoliths into executable microservicebased applications, in: C. D. Vasconcellos, K. G. Roggia, P. Bousfield, V. Collereii, J. P. Fernandes, M. Pereira (Eds.), SBLP'21: 25th Brazilian Symposium on Programming Languages, Joinville, Brazil, 27 September 2021 - 1 October 2021, ACM, 2021, p. 100–107. URL: <https://doi.org/10.1145/3475061.3475086>. doi:10.1145/3475061.3475086.