Exploring Micro Frontends

A Case Study Application in E-Commerce

https://renatocf.xyz/amp25-slides

Paper

Renato Cordeiro Ferreira

Institute of Mathematics and Statistics (IME)
University of São Paulo (USP) – Brazil

Jheronimus Academy of Data Science (JADS)
Technical University of Eindhoven (TUe) / Tilburg University (TiU) – The Netherlands

Slides

2025

Renato Cordeiro Ferreira

https://renatocf.xyz/contacts

B.Sc. and M.Sc. at University of São Paulo (BR)

Theoretical and practical experience with Machine Learning and Software Engineering

Former Principal ML Engineer at Elo7 (BR)

4 years of industry experience designing, building, and operating ML products with multidisciplinary teams

Scientific Programmer at JADS (NL)

Currently participating in the MARIT-D European project, using ML techniques for more secure seas

Ph.D. candidate at USP + JADS (BR + NL)

Research about SE4AI, in particular about MLOps and the software architecture of ML-Enabled Systems

Ricardo Kojo
ricardo.kojo@alumni.usp.br

Luiz Fernando Corte Real
sr.saude@alumni.usp.br

Renato Cordeiro Ferreira
renatocf@ime.usp.br

Thatiane de Oliveira Rosa
thatiane@ifto.edu.br

Alfredo Goldman
gold@ime.usp.br

Our paper provides **insights** into when the adoption of micro frontends **may be worthwhile**, *particularly in an industry context,* considering that research in this area is still evolving

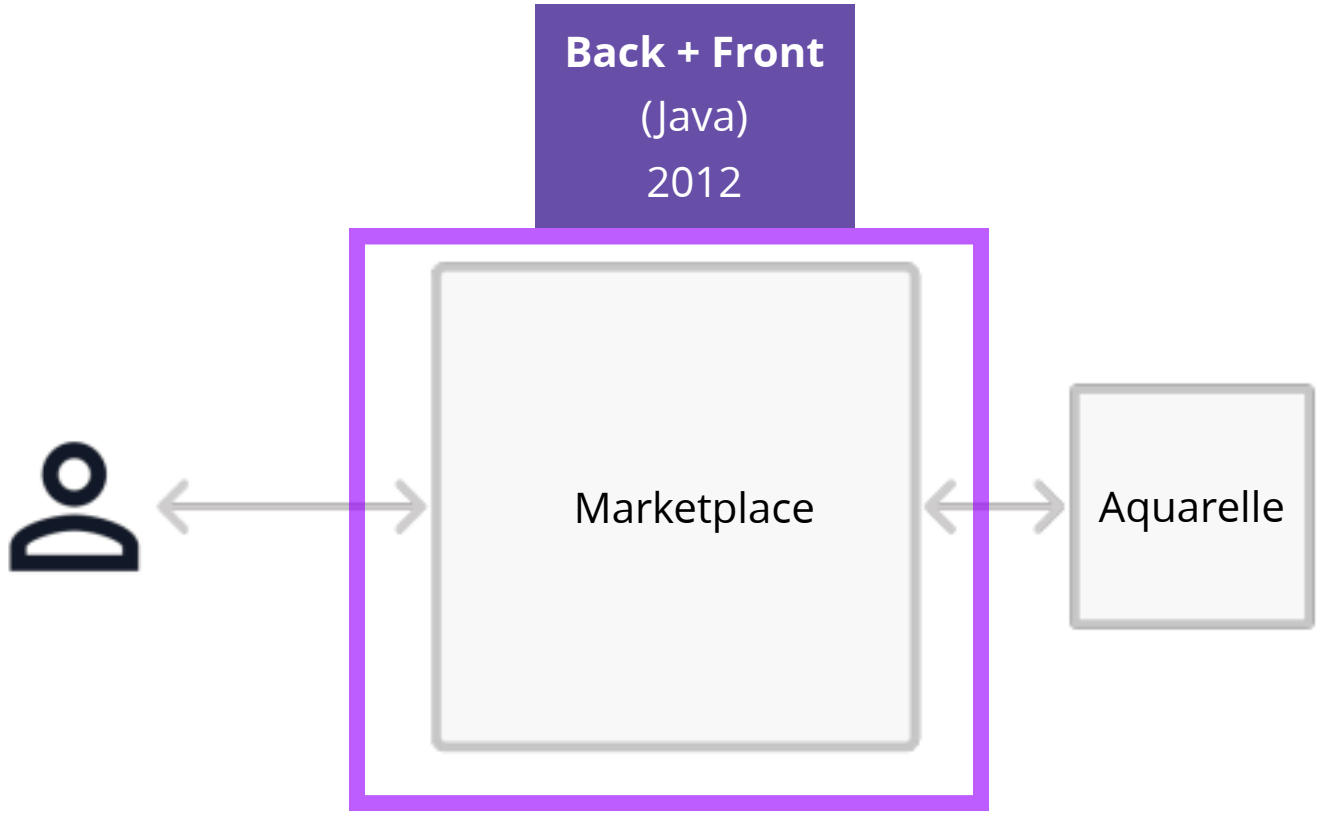
Research Questions

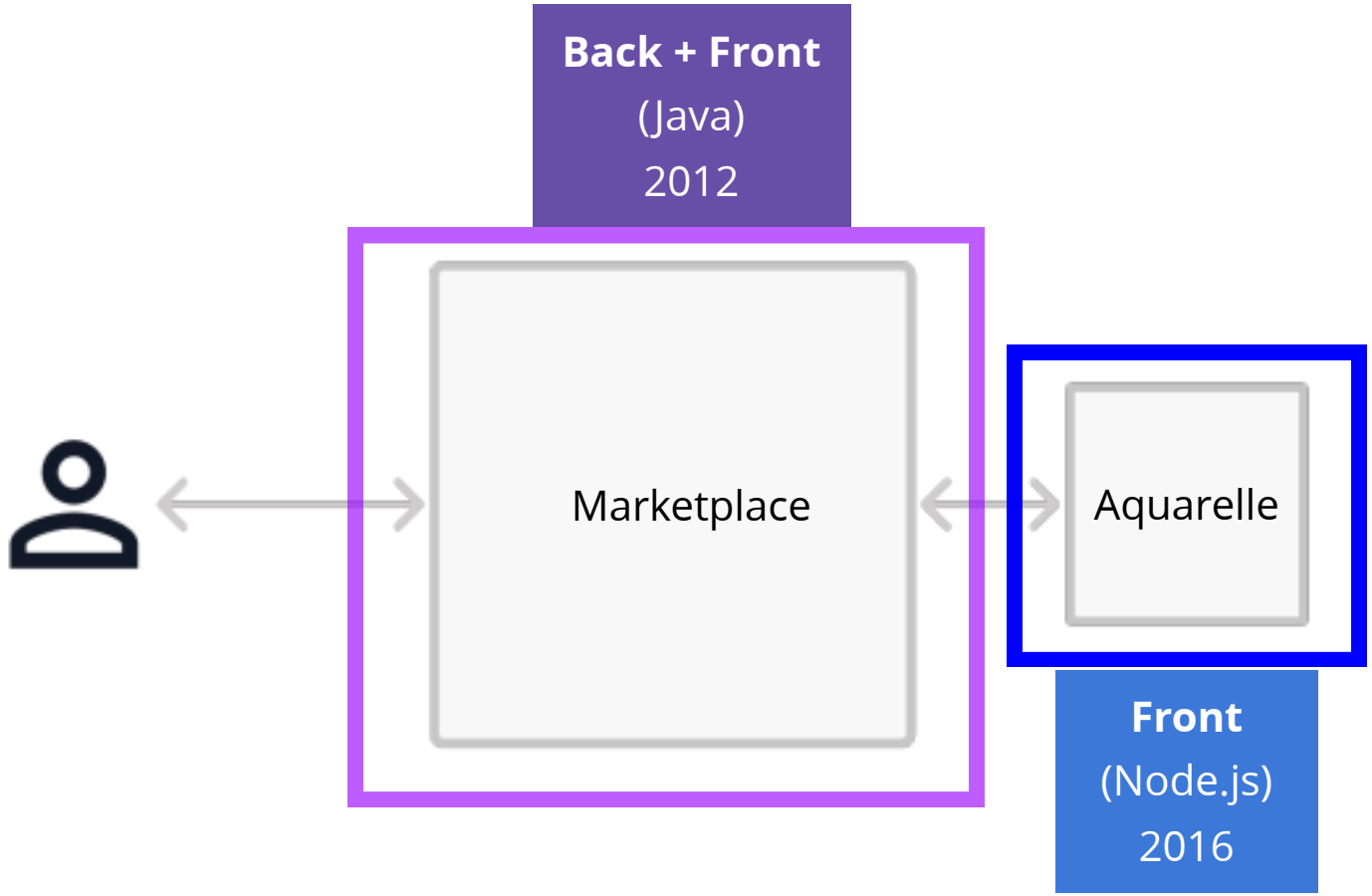
RQ1 What are the **motivations** and **challenges** involved in adopting a micro frontend architecture in the studied company, *which already uses microservices*?

RQ2 What are the perceived **benefits** and **drawbacks** reported by *developers involved in the migration* from a monolithic architecture to micro frontends?

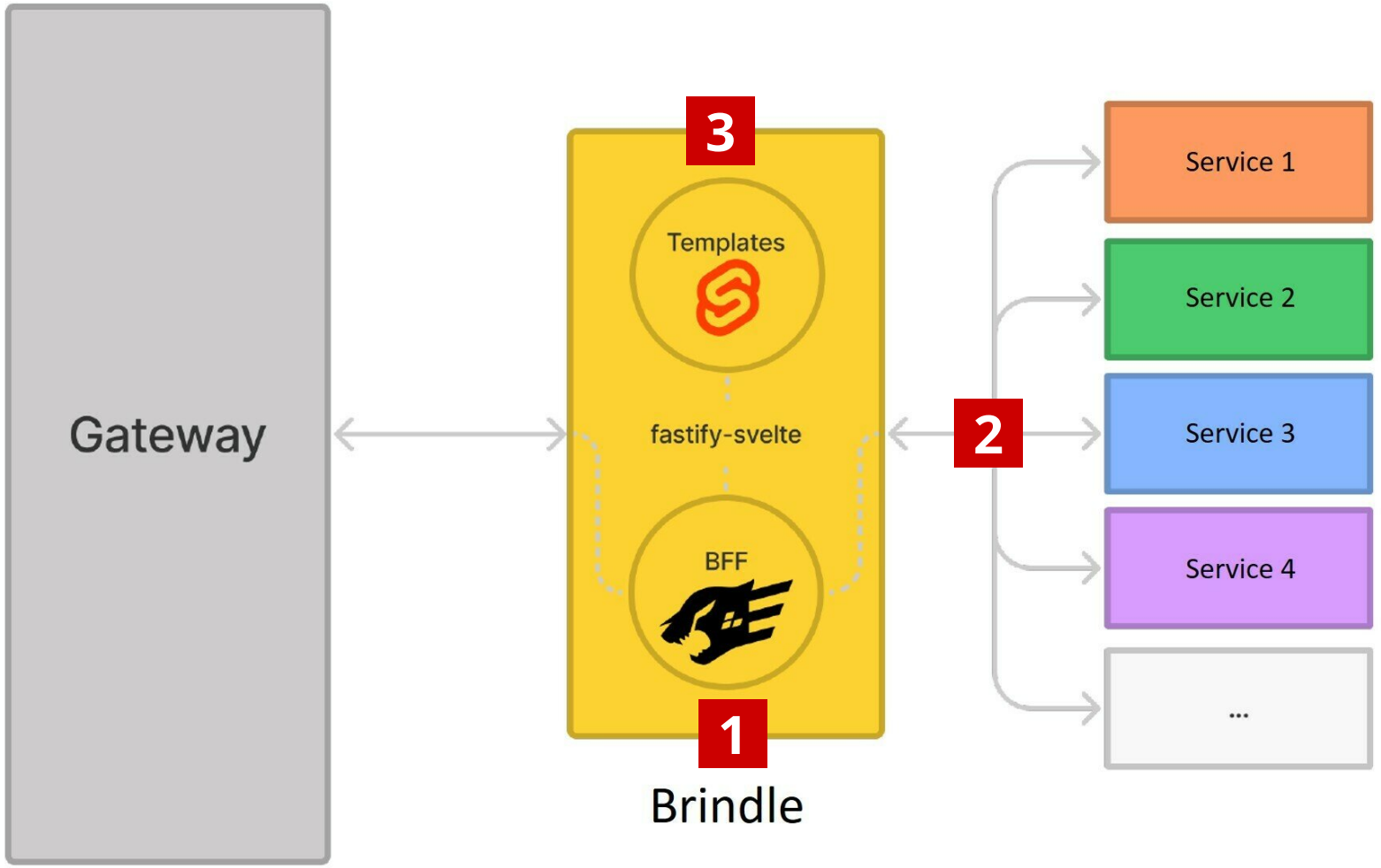
Journey to Micro Frontends


Marketplace
Aquarelle

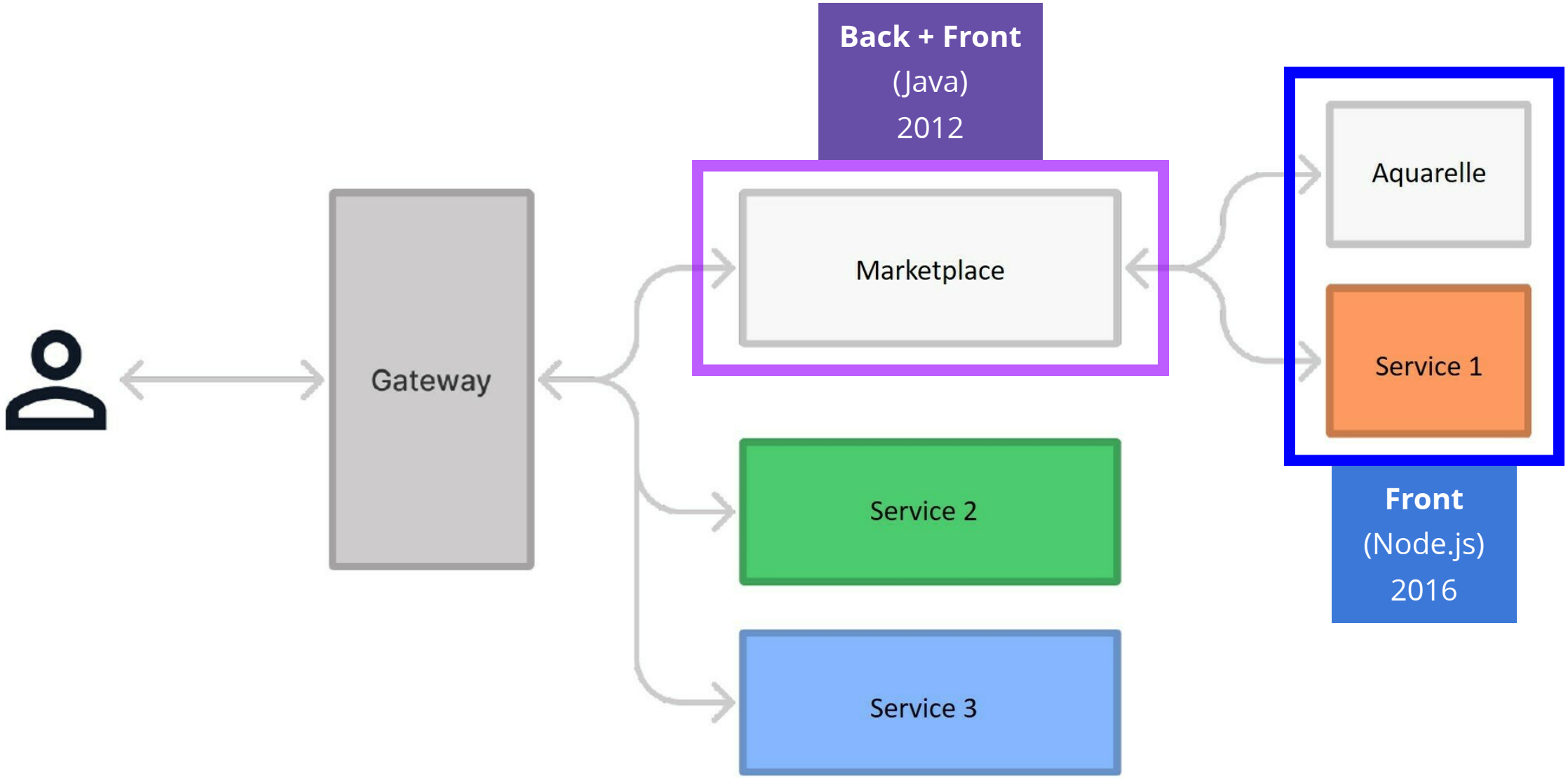

Back + Front
(Java)
2012
Marketplace
Aquarelle

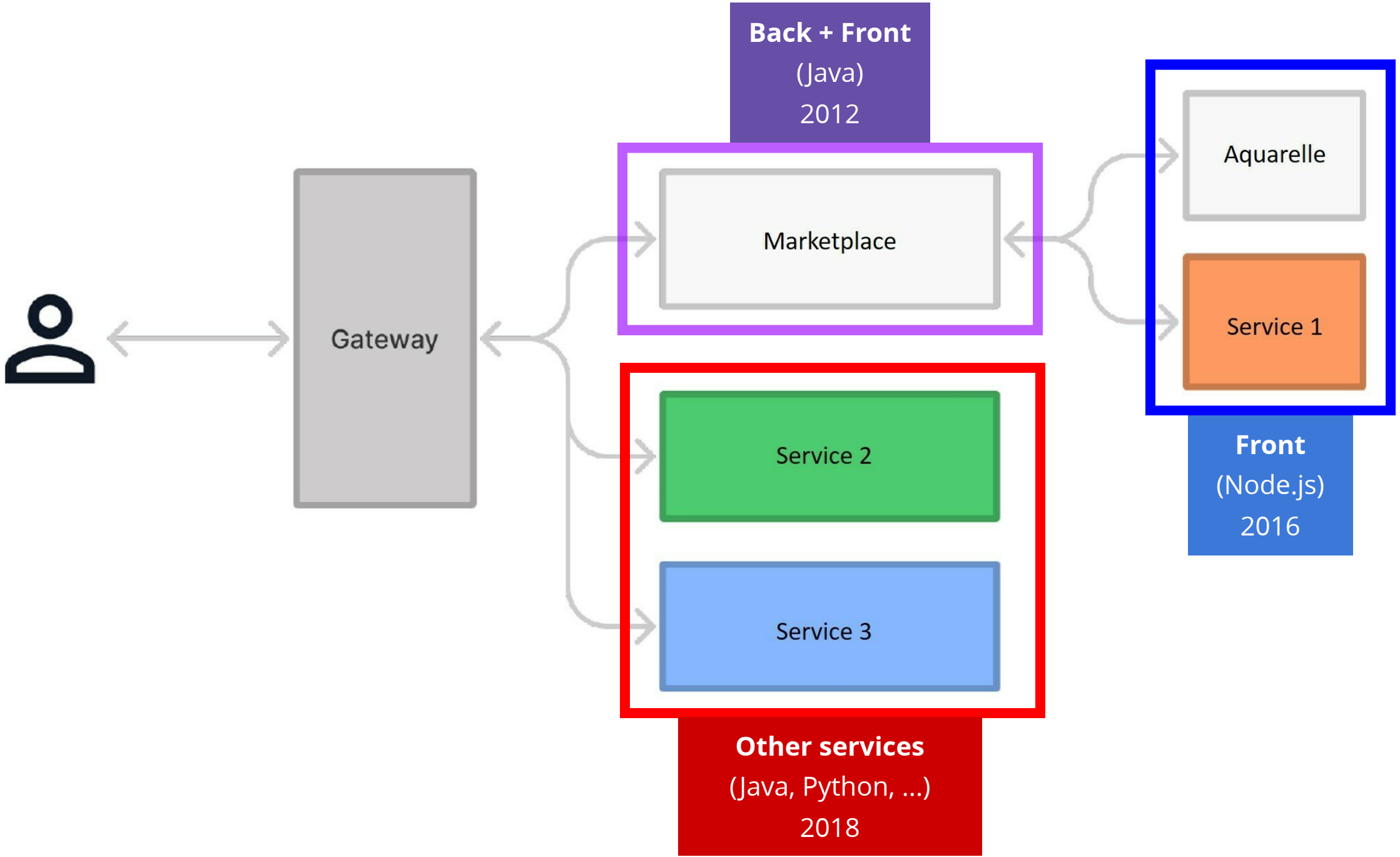


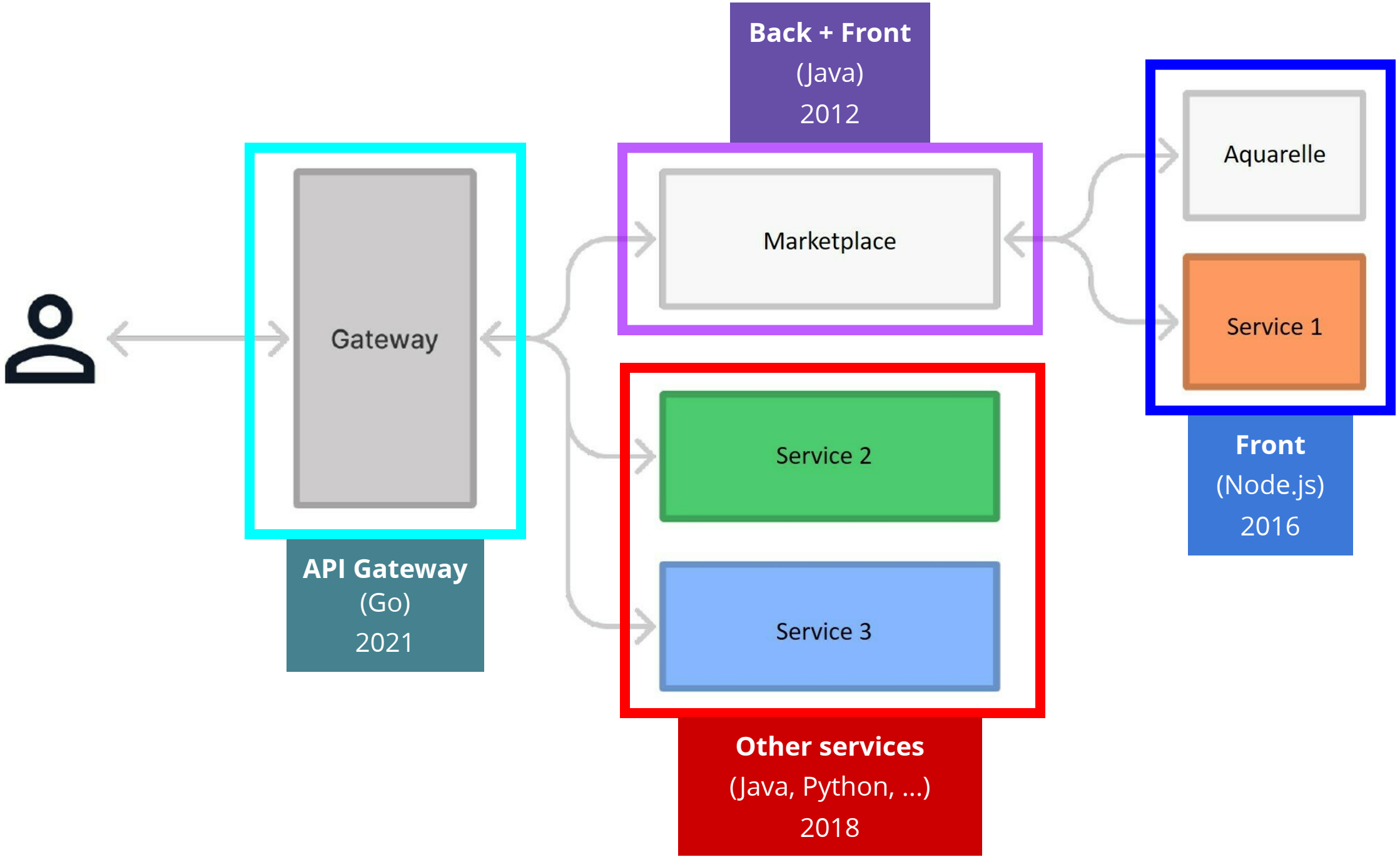
The **Aquarelle** project, built on Node.js, was introduced to implement a reactive chat feature capable of displaying dynamic backend data such as order status and user actions



Backend for Frontend (BFF) pattern handles (1) internal routing, (2) orchestrates data from microservices, and (3) forwards it to a template rendered by an open-source library developed by the Company


Back + Front
(Java)
2012
Gateway
Marketplace
Aquarelle
Service 1
Service 2
Service 3
Front
(Node.js)
2016


Back + Front
(Java)
2012
Gateway
Marketplace
Aquarelle
Service 1
Front
(Node.js)
2016
Service 2
Service 3
Other services
(Java, Python, ...)
2018


Back + Front
(Java)
2012
Gateway
Marketplace
Aquarelle
Service 1
Service 2
Service 3
API Gateway
(Go)
2021
Front
(Node.js)
2016
Other services
(Java, Python, ...)
2018

Survey with Developers

Methodology

Methodology

- Semi-open questionnaire

Methodology

- Semi-open questionnaire
- 1 month duration

Methodology

- Semi-open questionnaire
- 1 month duration
- Employees involved in frontend projects
  - Frontend developers
  - Technical Leads
  - Engineering managers
  - Software architects

Methodology

- Semi-open questionnaire
- 1 month duration
- Employees involved in frontend projects
  - Frontend developers
  - Technical Leads
  - Engineering managers
  - Software architects
- 19 questions (15 open-ended + 4 multiple-choice)

Methodology

- Semi-open questionnaire
- 1 month duration
- Employees involved in frontend projects
  - Frontend developers
  - Technical Leads
  - Engineering managers
  - Software architects
- 19 questions (15 open-ended + 4 multiple-choice)
- Full questionnaire available at Zenodo

Methodology

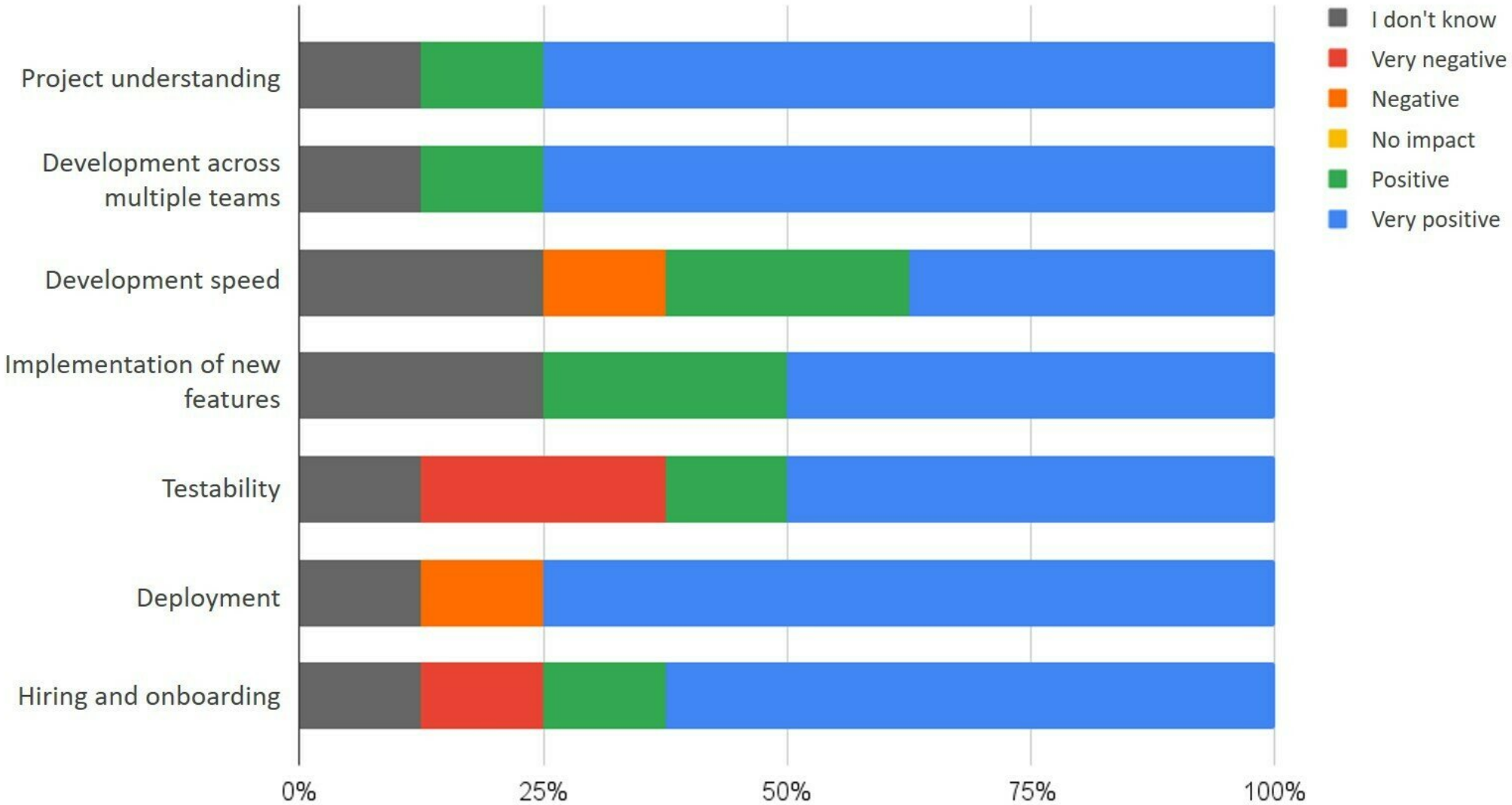
- Semi-open questionnaire
- 1 month duration
- Employees involved in frontend projects
  - Frontend developers
  - Technical Leads
  - Engineering managers
  - Software architects
- 19 questions (15 open-ended + 4 multiple-choice)
- Full questionnaire available at Zenodo

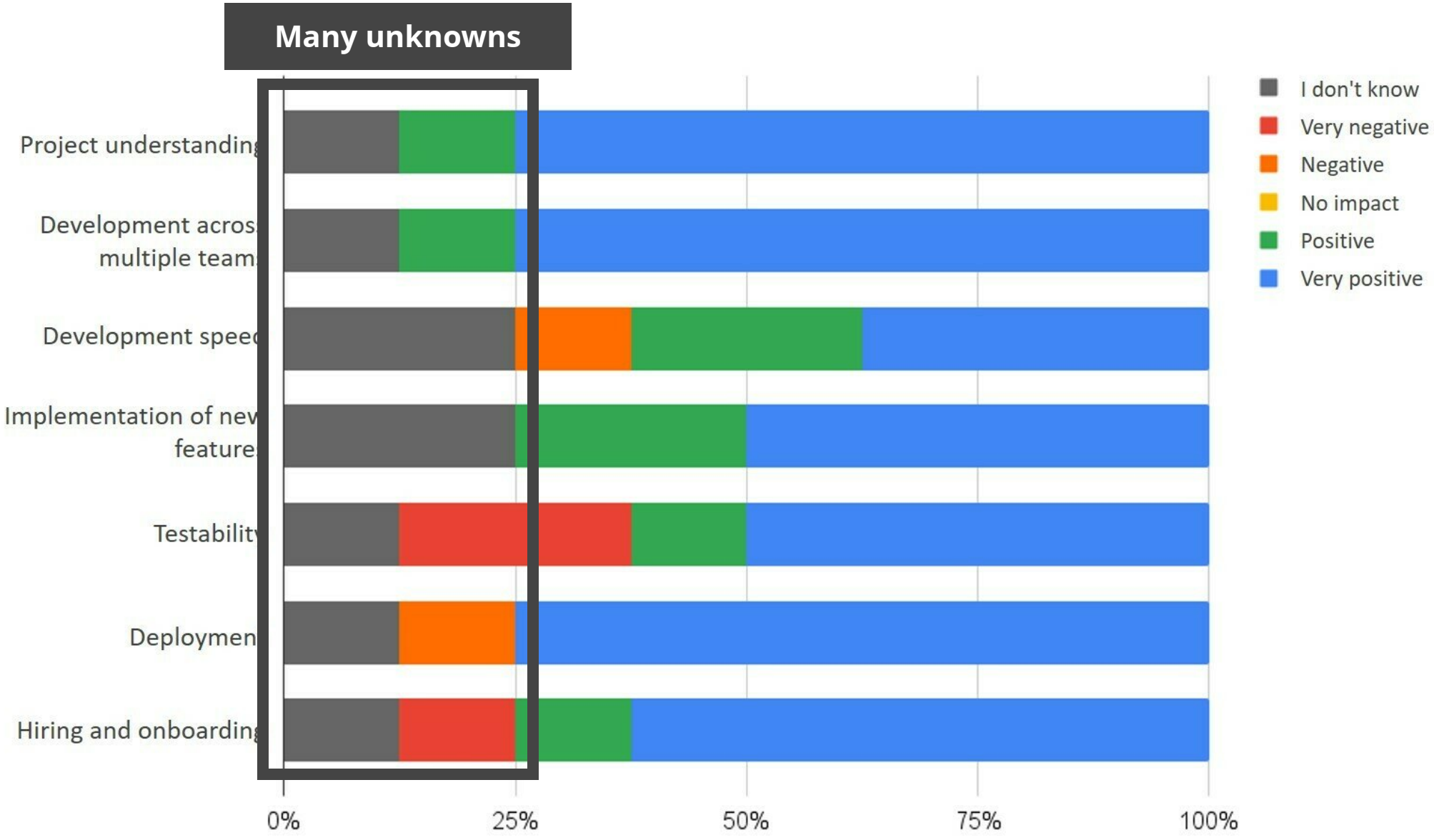
8 participants

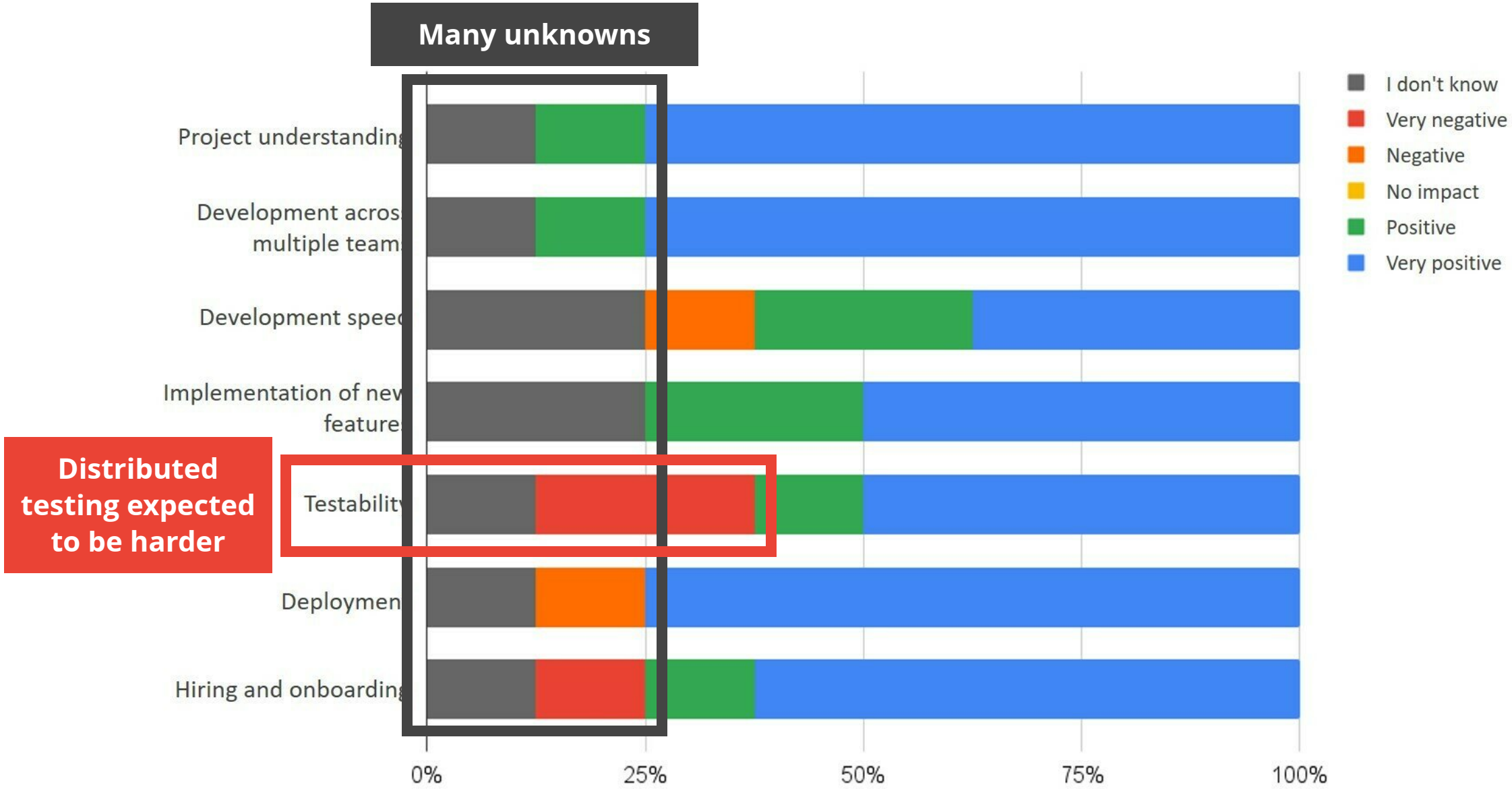
7 men / 1 women

5 participants

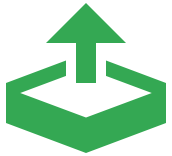
>10 years of experience


Project understanding
Development across multiple teams
Development speed
Implementation of new features
Testability
Deployment
Hiring and onboarding
0%
25%
50%
75%
100%
I don't know
Very negative
Negative
No impact
Positive
Very positive


Many unknowns
Project understanding
Development acros
multiple team
Development speed
Implementation of nev
feature
Testability
Deploymen
Hiring and onboarding
0%
25%
50%
75%
100%
I don't know
Very negative
Negative
No impact
Positive
Very positive


Many unknowns
Project understanding
Development acros
multiple team
Development speed
Implementation of nev
feature
Testability
Deploymen
Hiring and onboarding
0%
25%
50%
75%
100%
I don't know
Very negative
Negative
No impact
Positive
Very positive
Distributed testing expected to be harder

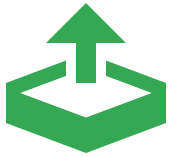
Trade-offs of the new architecture



Positives

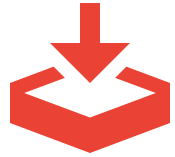
Negatives

Trade-offs of the new architecture



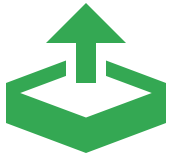
Positives

- New **developers**



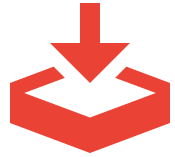
Negatives

Trade-offs of the new architecture



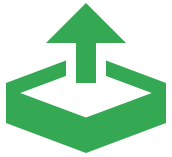
Positives

- New **developers**
- New **technologies**



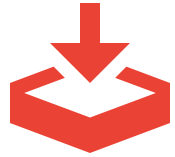
Negatives

Trade-offs of the new architecture



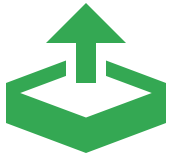
Positives

- New **developers**
- New **technologies**
- Faster **deployments**



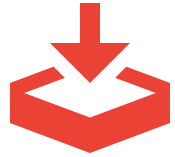
Negatives

Trade-offs of the new architecture



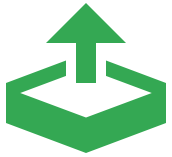
Positives

- New **developers**
- New **technologies**
- Faster **deployments**
- Simpler **implementation**



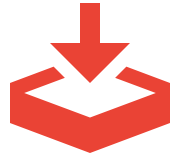
Negatives

Trade-offs of the new architecture



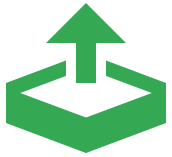
Positives

- New **developers**
- New **technologies**
- Faster **deployments**
- Simpler **implementation**
- More transparency in the **project understanding**



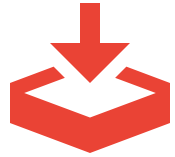
Negatives

Trade-offs of the new architecture



Positives

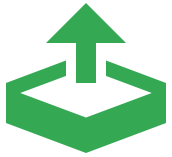
- New **developers**
- New **technologies**
- Faster **deployments**
- Simpler **implementation**
- More transparency in the **project understanding**



Negatives

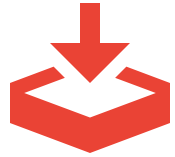
- **Evolution** depends on extracting services from the monolith

Trade-offs of the new architecture



Positives

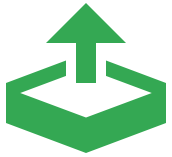
- New **developers**
- New **technologies**
- Faster **deployments**
- Simpler **implementation**
- More transparency in the **project understanding**



Negatives

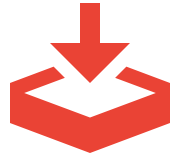
- **Evolution** depends on extracting services from the monolith
- **Onboarding** can be harder (because of complexity)

Trade-offs of the new architecture



Positives

- New **developers**
- New **technologies**
- Faster **deployments**
- Simpler **implementation**
- More transparency in the **project understanding**



Negatives

- **Evolution** depends on extracting services from the monolith
- **Onboarding** can be harder (because of complexity)
- **Data conversion** affects speed (formerly done by the monolith)

Research Questions

RQ1 What are the **motivations** and **challenges** involved in adopting a micro frontend architecture in the studied company, *which already uses microservices*?

RQ2 What are the perceived **benefits** and **drawbacks** reported by *developers involved in the migration* from a monolithic architecture to micro frontends?

While not the only possible solution,
micro frontends turned out to be
the most convenient
within that specific context

Exploring Micro Frontends

A Case Study Application in E-Commerce

https://renatocf.xyz/amp25-slides

Paper

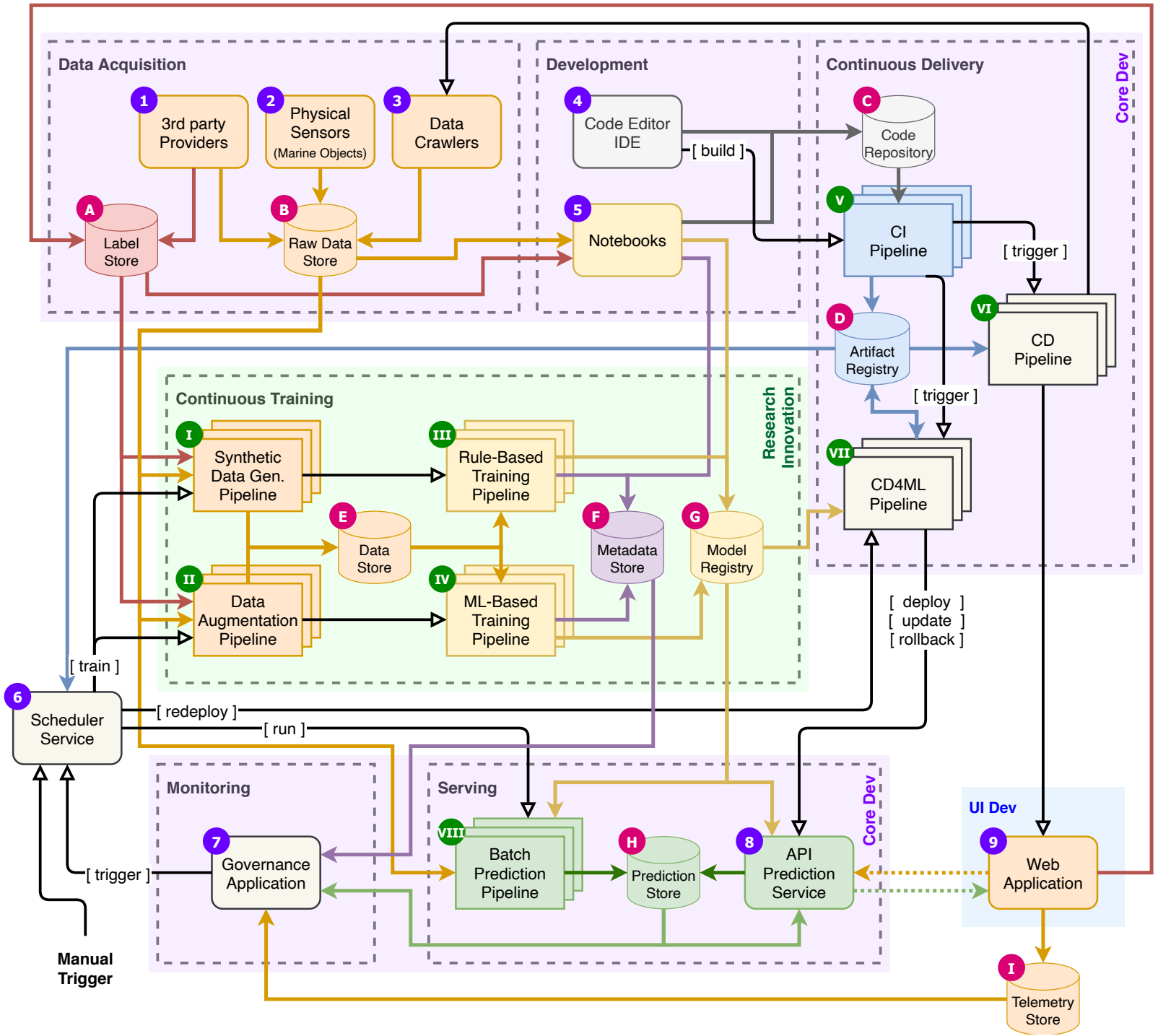
Slides

Renato Cordeiro Ferreira

Institute of Mathematics and Statistics (IME)
University of São Paulo (USP) – Brazil

Jheronimus Academy of Data Science (JADS)
Technical University of Eindhoven (TUe) / Tilburg University (TiU) – The Netherlands

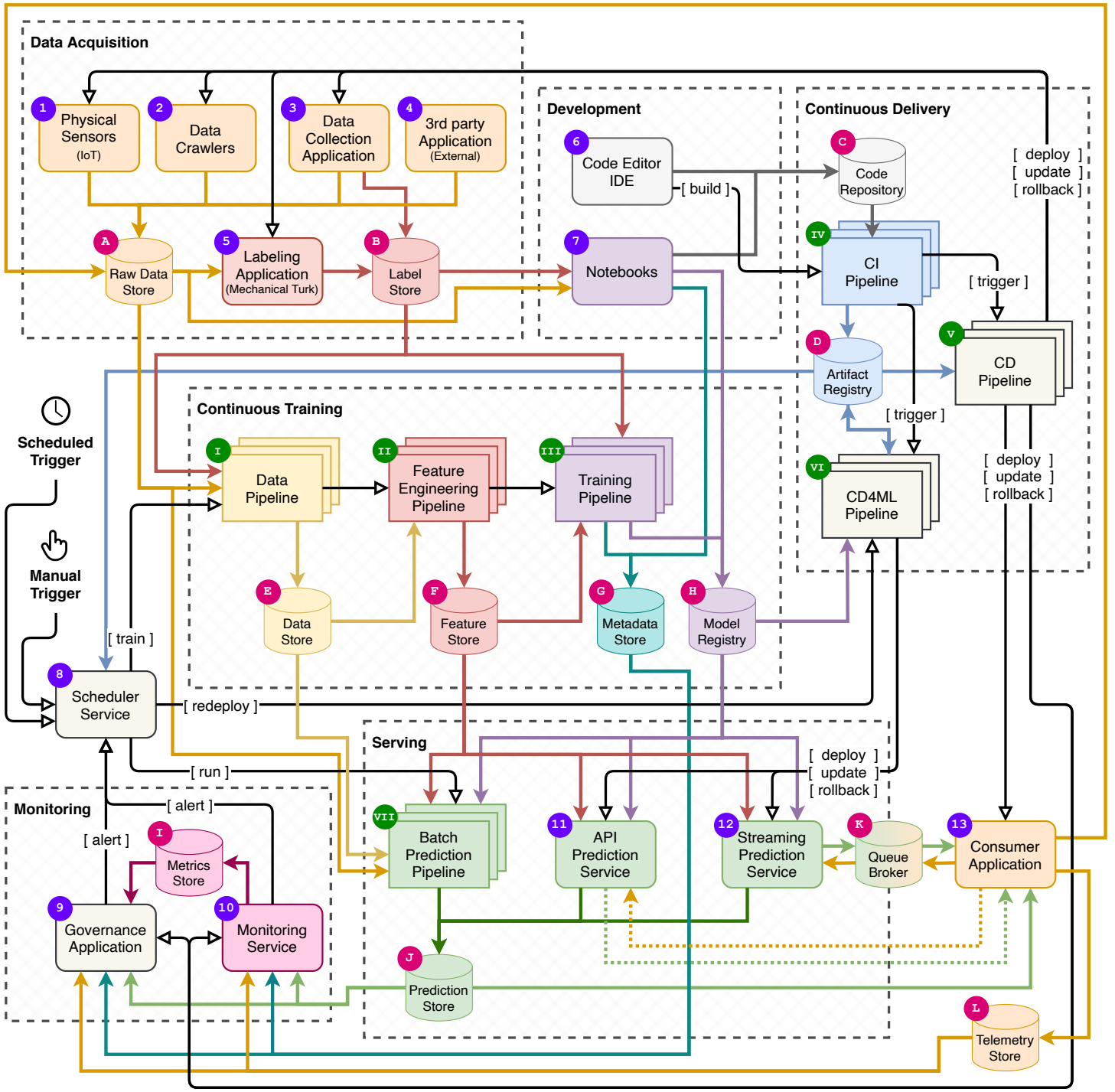
2025



Research Track - SummerSOC 2025

MLOps with Microservices:
A Case Study on the Maritime Domain

https://renatocf.xyz/ssoc25-paper



Doctoral Symposium - CAIN 2025

A Metrics-Oriented Architectural Model to Characterize Complexity on Machine Learning-Enabled Systems

https://renatocf.xyz/cain25-paper