

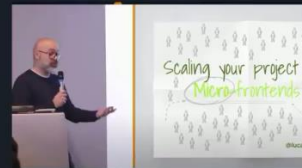
What are Micro frontends?

Contents

1. What is a Micro-frontend
2. Why they exist
3. How to build a micro-frontend
4. In Conclusion
5. Questions?

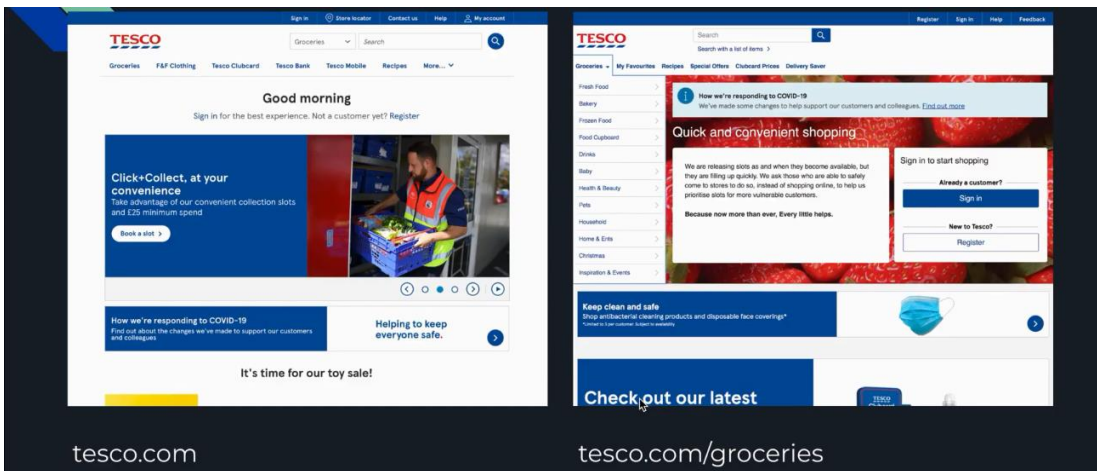
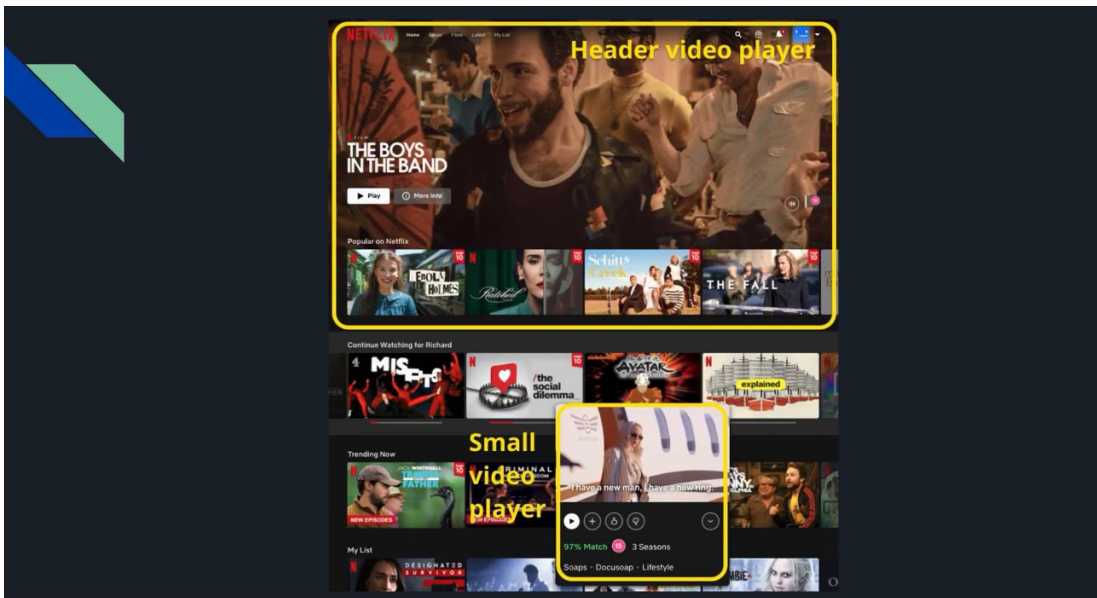


Luca Mezzalana



1. What is a Micro-frontend

“It’s trying to think of a website as a project or page of composed features that are owned by independent teams”



Benefits of micro-frontends

- Independant deployments
- Independant updates and bug fixes
- Tech stack freedom
- Independant decoupled units
- Application split up by business domain

Who is using micro-frontends



2. Why they exist

Micro Frontends
extending the microservice idea to frontend development

Techniques, strategies and recipes for building a modern web app with multiple teams that can ship features independently.

The Author
Michael Geers is a software engineer at neuland Büro für Informis. He worked on multiple large e-commerce projects over the last ten years. He loves the frontend and is passionate about design systems and web-performance. Follow on Twitter or GitHub.

// What are Micro Frontends?

The term Micro Frontends first came up in ThoughtWorks Technology Radar at the end of 2018. It extends the concepts of micro services to the frontend world. The current trend is to build a feature-rich and powerful browser application, aka single page app, 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. That's what we call a **Frontend Monolith**.

The idea behind Micro Frontends is to think about a website or web app as a composition of features which are owned by **independent teams**. Each team has a **distinct area of business or mission** it cares about and specialises in. A team is **cross functional** and develops its features **end-to-end**, from database to user interface.

However, this idea is not new. It has a lot in common with the **Self-contained Systems** concept. In the past approaches like this went by the name of **Frontend Integration for Verticalised Systems**. But Micro Frontends is clearly a more friendly and less bulky term.

Monolithic Frontends

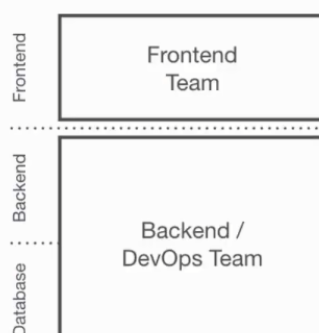
The Monolith Front t Back Microservices

The Book
Micro Frontends IN ACTION

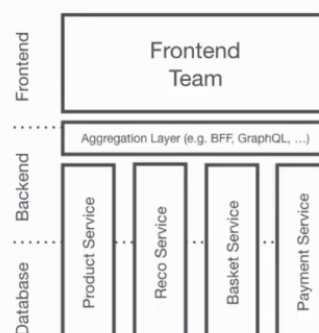
The Monolith



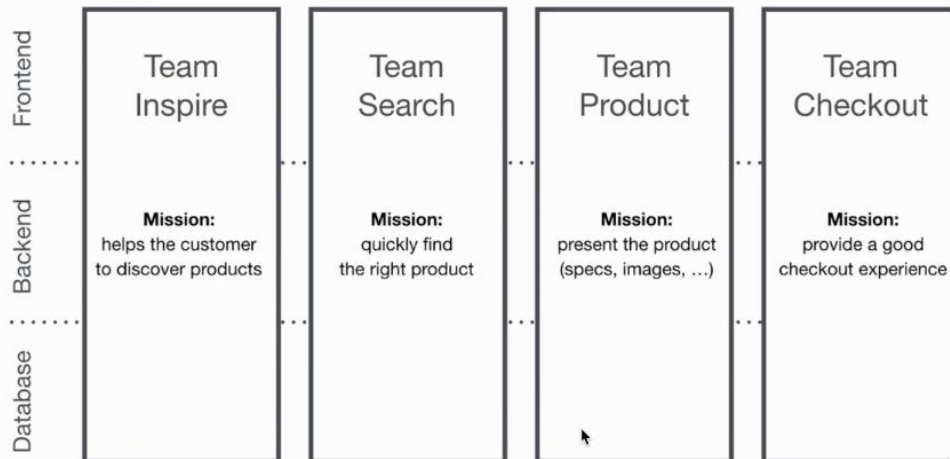
Front & Back



Microservices



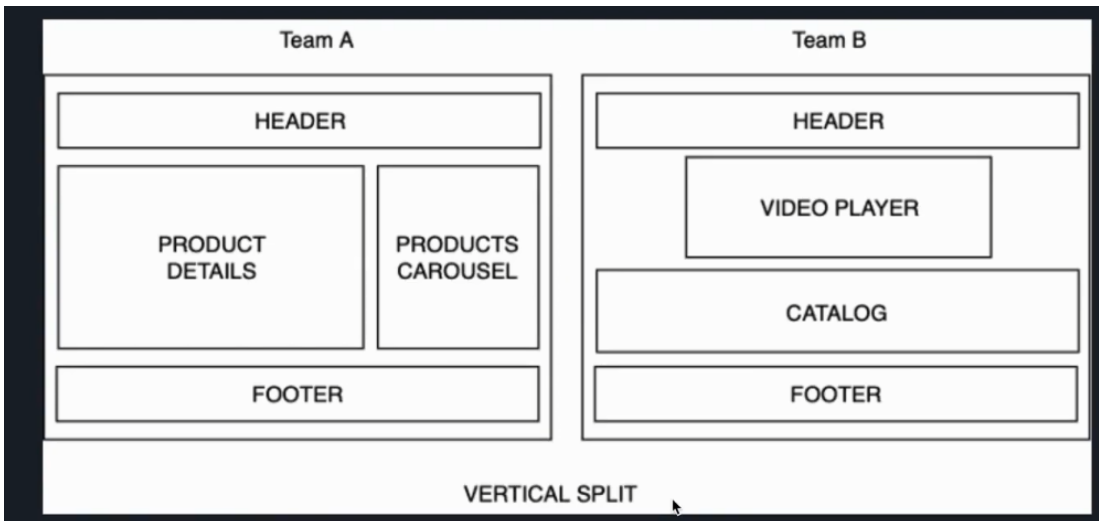
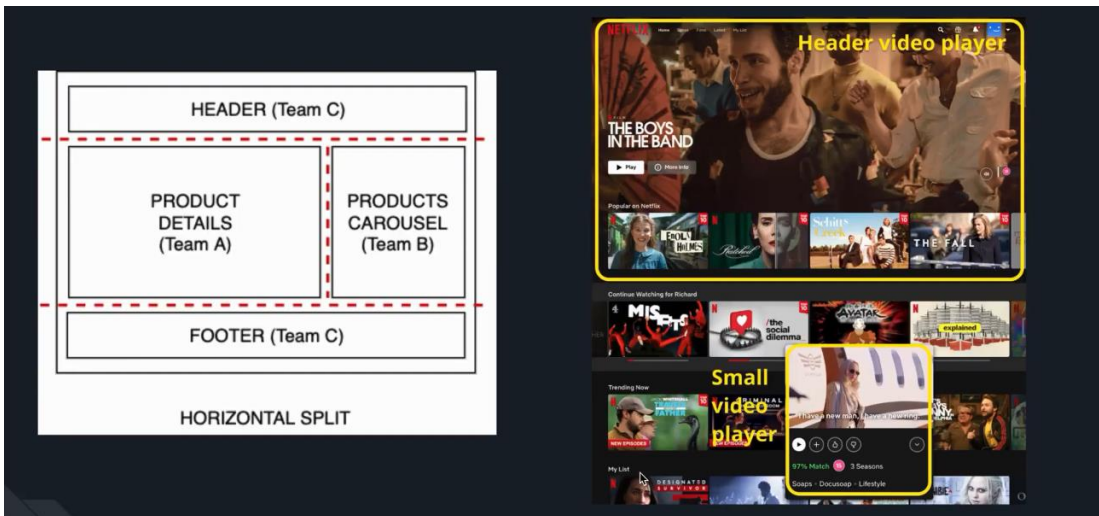
End-to-End Teams with Micro Frontends



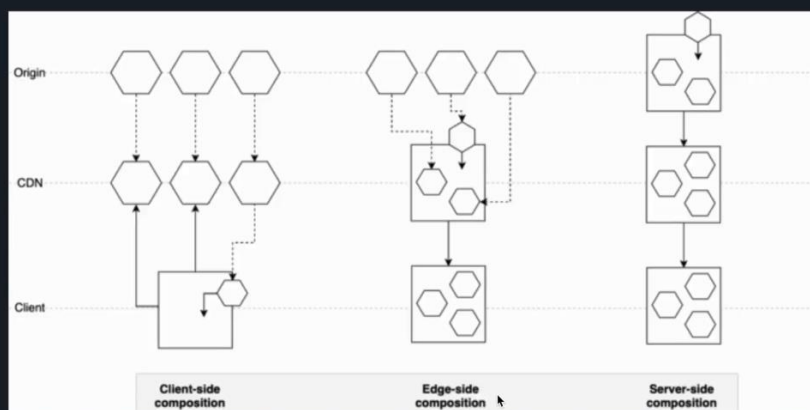
3. How to build a micro-frontend



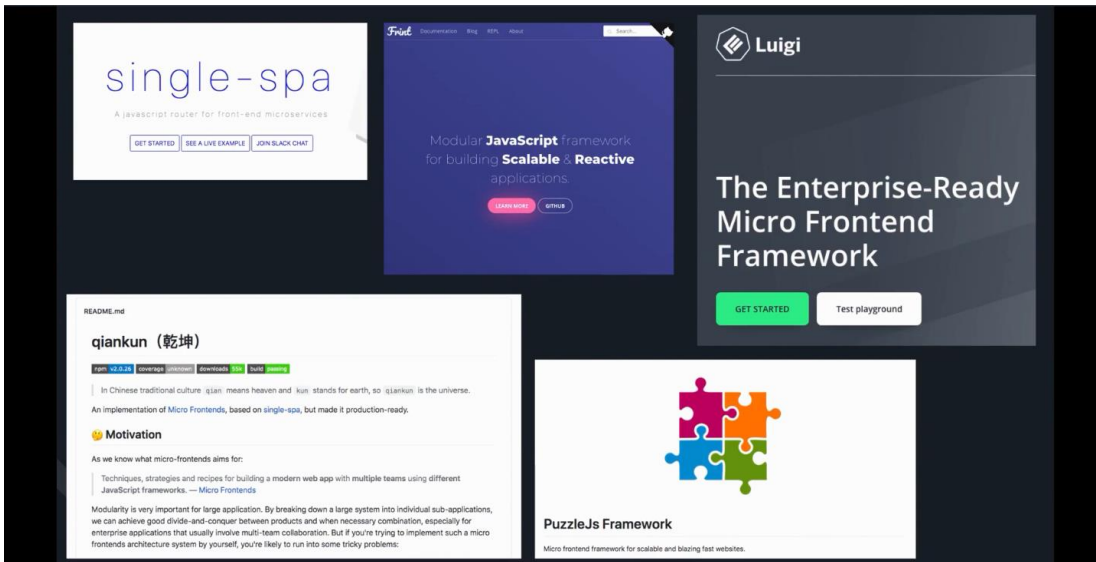
- Define what how you want the micro-frontend split for your project



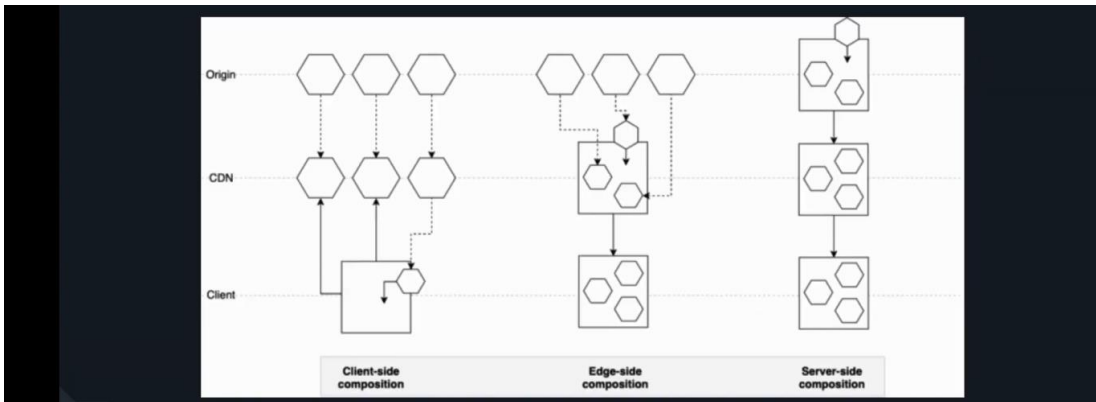
- Figure out how you want to compose the micro-frontend



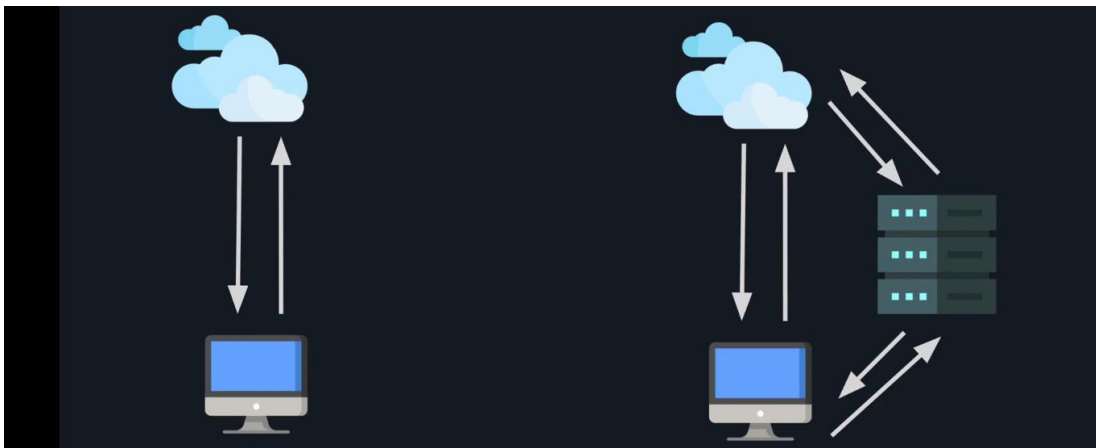
Client side



These are all tools that support client-side MFE development



For Edge-Side Composition approach, you can use AWS Lambda@Edge or Azure Stack Edge to provide ESI from your CDNs



Lambda@Edge

Run your code closer to your users

Get started with Lambda@Edge

Lambda@Edge is a feature of Amazon CloudFront that lets you run code closer to users of your application, which improves performance and reduces latency. With Lambda@Edge, you don't have to provision or manage infrastructure in multiple locations around the world. You pay only for the compute time you consume - there is no charge when your code is idle.

With Lambda@Edge, you can run your code closer to your users, which improves performance and reduces latency. With Lambda@Edge, you don't have to provision or manage infrastructure in multiple locations around the world. You pay only for the compute time you consume - there is no charge when your code is idle.

Azure Stack Edge

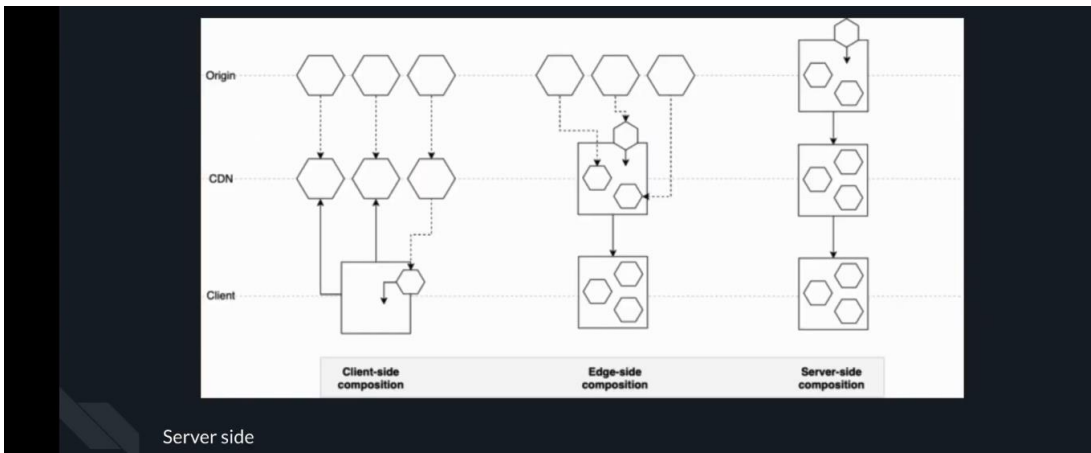
A portfolio of devices that bring the compute, storage and intelligence to the edge right where data is created

Start now

Home / Azure Stack / Azure Stack Edge

Azure Stack Edge

Features Benefits Devices Use cases Pricing Getting started



The Server-Side approach means we go all the way to the server to get the HTML to display, tools for this approach are below

Ara Framework

Build Micro-frontends easily using Airbnb Hypervnova

GET STARTED

OpenComponents

Painless micro frontends delivery

GitHub Documentation Ask the team

Piral

Build highly modular applications fully flexible for large scale portal solutions. Adjusted to your development needs, not the other way around.

Get Started Open Playground

Tailor

npm non testall node-tailor 2 dependencies version 1.1.2 updated 2 years ago

Build deploy cdn cdn cdn cdn cdn

5. In Conclusion

“Too often I see people judging other companies or colleague’s decisions without understanding the context in which that decision was made. In reality, the context is even more important than the decision itself, because without it, the decision could sound horrible or totally inappropriate. In reality, that choice could have been the best (or only) option for that specific context.”

Luca Mezzalana

Sources and credits

- <https://micro-frontends.org/>
- <https://blog.bitsrc.io/state-of-micro-frontends-9c0c604ed13a>
- <https://opencomponents.github.io/>
- <https://medium.com/@lucamezzalana/micro-frontends-decisions-framework-ebcd22256513>
- <https://dev.to/florianrappl/communication-between-micro-frontends-41fe>
- https://developer.mozilla.org/en-US/docs/Web/Guide/Events/Creating_and_triggering_events