Scaling Your Project with MICRO-FRONTENDS

@lucamezzalira
Luca Mezzalira
VP of Architecture at DAZN
Google Developer Expert
London JavaScript Community Manager
Architecture evolution

DB

Application Server

Single-Page Application
Architecture evolution

- API Gateway
- Single-Page Application
- Microservice
- DB
Architecture evolution

- DB
- Microservice
- API Gateway
- MFE
- Single-Page Application
- MFE
- MFE
- MFE
MICRO-FRONTENDS
DEFINITION
Micro-frontends are the technical representation of a business subdomain, they allow independent implementations with the same or different technology choices. Finally they should avoid sharing logic with other subdomains and they are own by a single team.
What is a micro-frontend?
MICRO-FRONTENDS
PRINCIPLES
Model around business domain

Decentralisation

Culture of automation

Deploy independently

Hide implementation details

Isolate failure
MICRO-FRONTENDS
DECISIONS FRAMEWORK
Key micro-frontends decisions

1. Define
2. Compose
3. Route
4. Communicate
Define a micro-frontends

header

image

article
description

related
articles

footer

landing
page

sign in

sign up

catalog
Horizontal VS Vertical split

1. Up front investment
2. Teams structure
3. Great for SEO
4. Testing challenges
5. Scalability challenges
6. Dependency management

1. Traditional development
2. Embracing JavaScript ecosystem
3. Dynamic Rendering for SEO
Compose micro-frontends

ORIGIN

CDN

CLIENT
Route micro-frontends
Micro-frontends communication

mysite.com/section?id=123
THERE IS NO RIGHT OR WRONG BUT ONLY THE RIGHT APPROACH BASED ON THE CONTEXT
MICRO-FRONTENDS @DAZN
MAN UNITED - LIVERPOOL

DAZN PICKS

Premier League | 16:00

NFL | 15:30

FALCONS @ SEAHAWKS

MAN UNITED - LIVERPOOL

WAS LÄUFT

Premier League | 16:00

NFL | 15:30
Micro-Frontend @ DAZN

- HTML
- App.js
- Vendor.js
- CSS
DAZN implementation

1. Application startup
2. I/O operations
3. Micro-Frontends lifecycle
4. Communication between Micro-Frontends
DAZN implementation

1. A Micro-Frontend represents a business domain
2. A Micro-Frontend is autonomous
3. One Micro-Frontend loaded per time
4. No sharing between micro-frontends (*)
5. Technology agnostic

(*) with some exceptions
How bootstrap works

1. Call Startup service
2. Understand user status
3. Load and Mount a Micro-Frontend
DAZN routing

Cloudfront → Lambda@Edge → Bootstrap → Micro-Frontend
Components

1. Development time integration
2. APIs first
3. Own by a single team
What we have learnt...

- Orchestrate
- Compose
- Communicate
Building Micro-Frontends
Scaling Teams and Projects
Empowering Developers

Early Release
RAW & UNEDITED

Luca Mezzalira
THANK YOU

t: @lucamezzalira
w: lucamezzalira.com
e: mezzalab@gmail.com