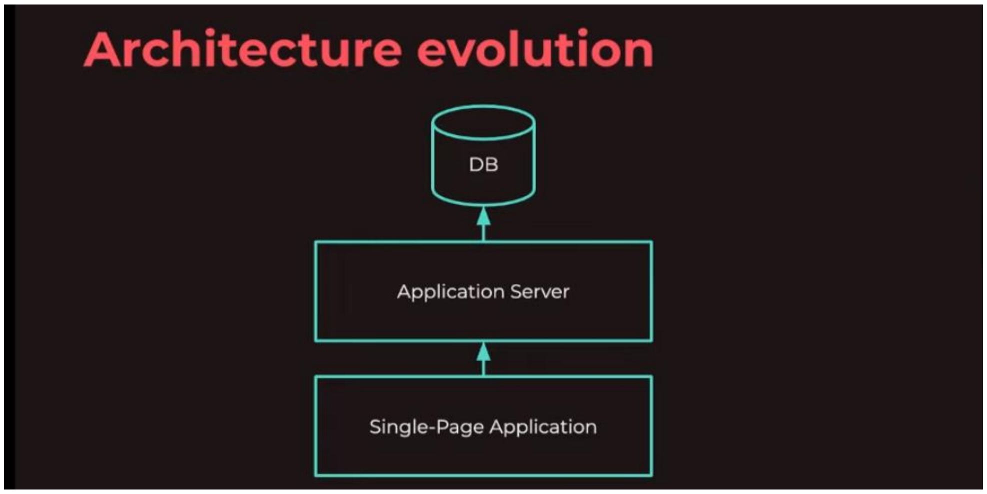
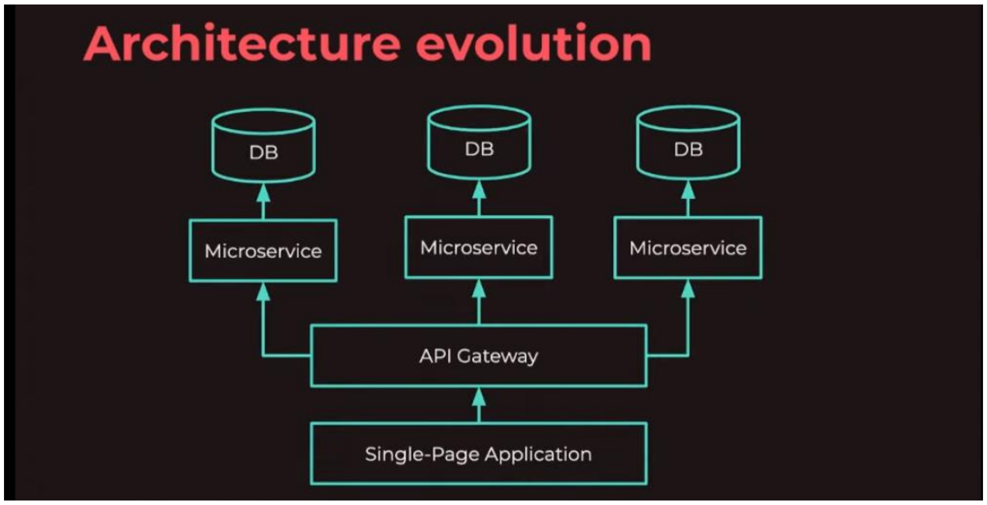
MICRO-FRONTENDS

Luca Mezzalira explains how to implement micro-frontends, enabling to scale up a project with tens of developers without reducing the throughput. **Micro-frontends** are a new architectural trend in the development of front-end applications. This style can provide several benefits to our projects, offering a level of decoupling never seen before in single-page applications or universal architectures.

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



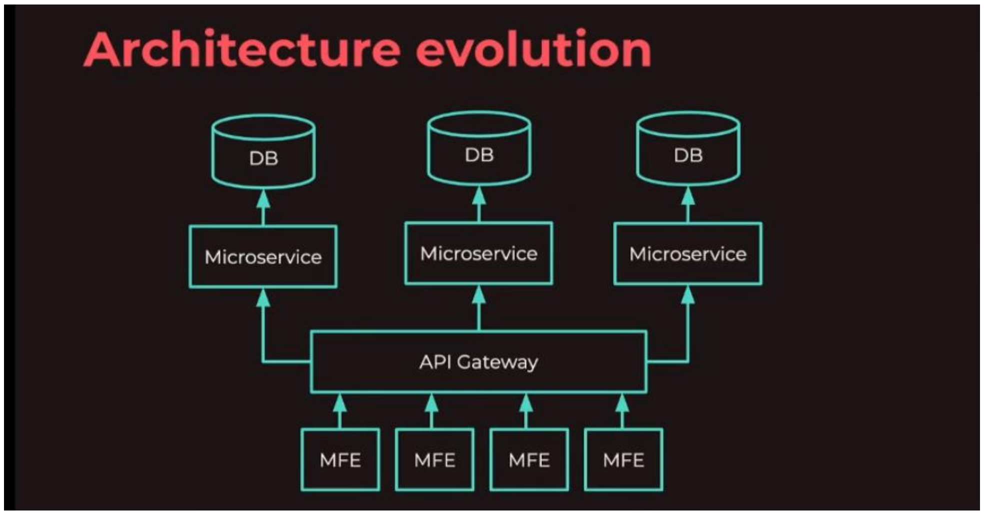
Architecture evolution



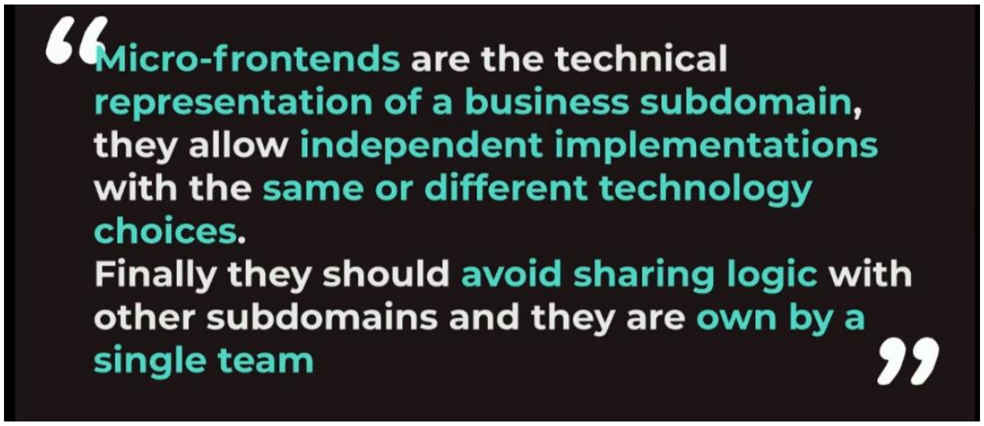
Architecture evolution

We need a way to make development scalable on the frontend

Architecture evolution

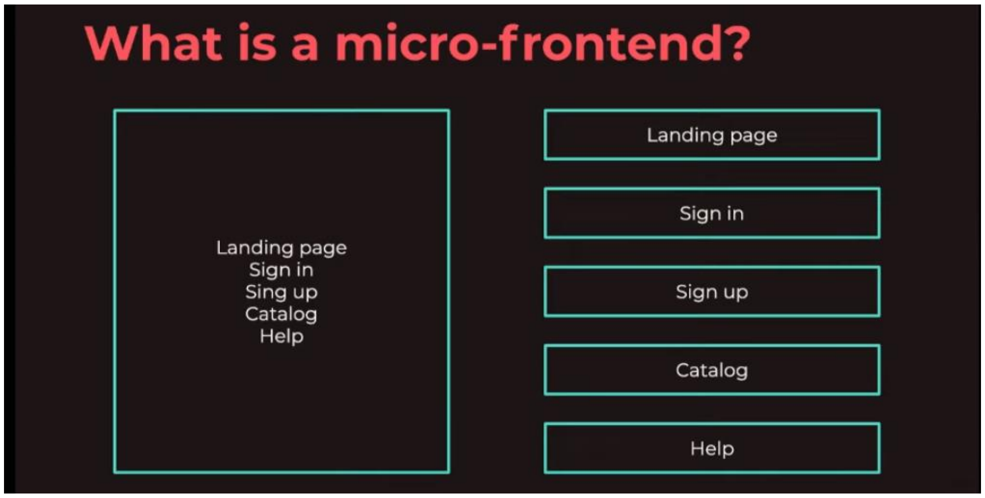


MICRO-FRONTENDS DEFINITION



“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”

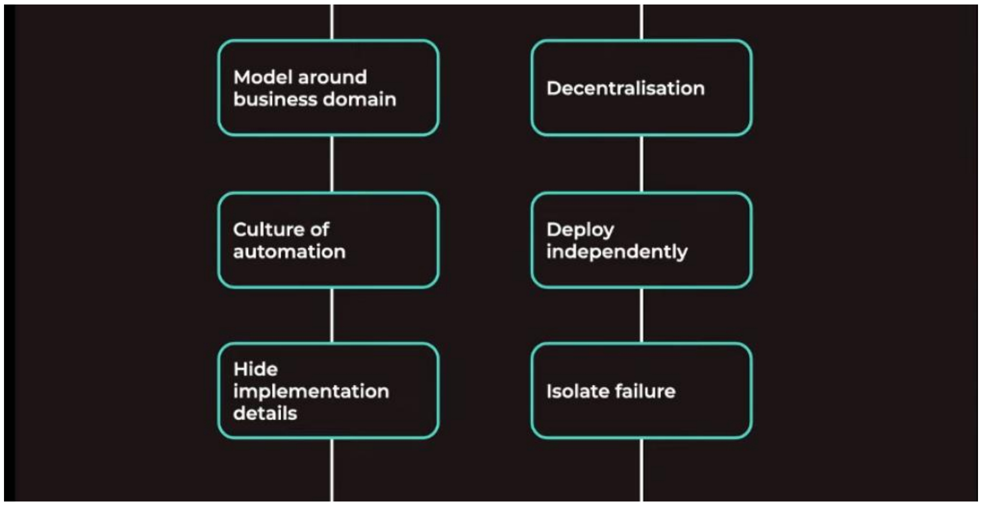
What is a micro-frontend?



Imagine a SPA that contains different **domains** like the authentication domain for sign in and sign out functionalities, the catalog domain, etc.

MICRO-FRONTENDS
PRINCIPLES

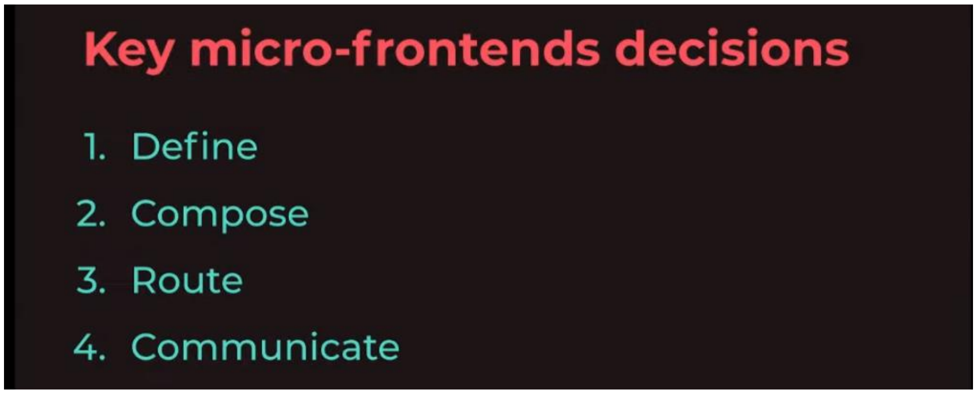
How can we scale a SPA development process for multiple developers?



Above are the principles that we base our work on. Decentralization is all about installing guardrails and let developers have the freedom to develop code. We also need a strong automation pipeline that allows our developers to get feedback quickly enough. We also want to deploy each smaller MFEs when code changes in them only. We also need to have APIs for MFE communications. We now need to isolate, handle and mitigate failures within each MFE on the larger page.

MICRO-FRONTENDS
DECISIONS FRAMEWORK

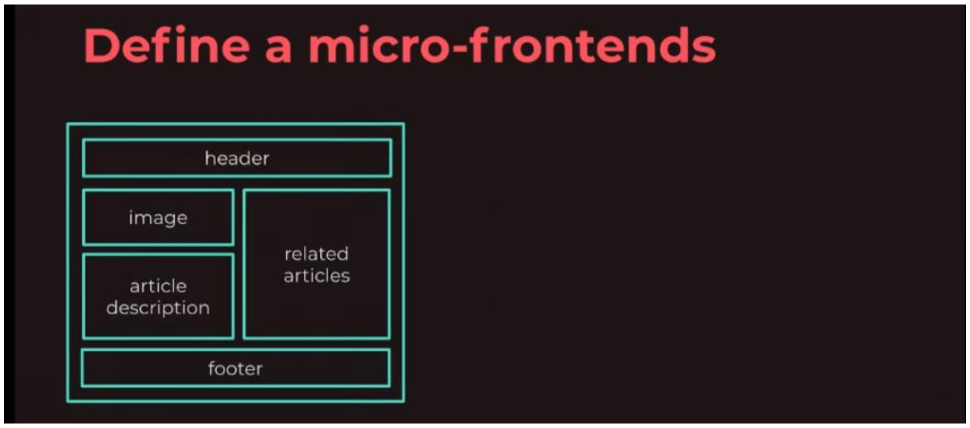
There are 4 key decisions that you need to take care of to make your MFE architecture work.



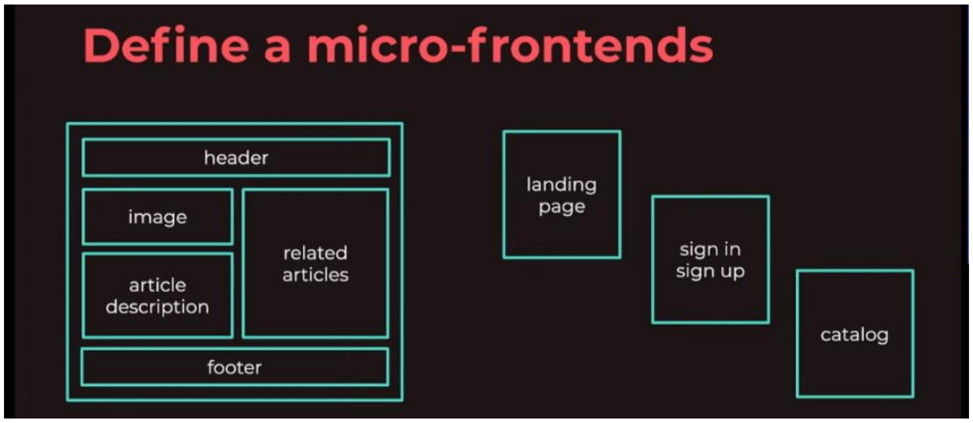
Key micro-frontends decisions

1. Define
2. Compose
3. Route
4. Communicate

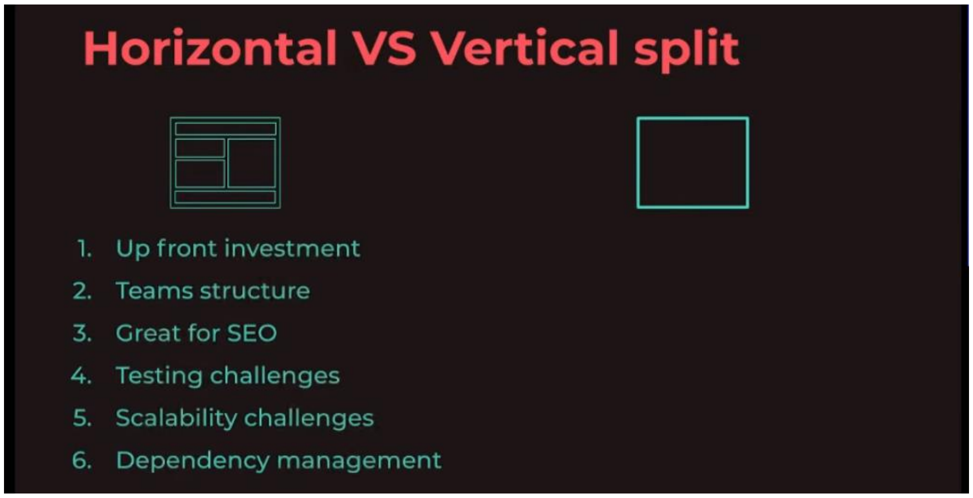
These **4 key decisions framework** will allow you to make all other decisions easily, like how you create and manage your **design system**, how you share code withing the MFE architecture, etc.


Define a micro-frontends

This **horizontal slicing approach** is representing each MFE as part of a larger composed view, e.g SAP's Luigi framework. This requires a **templating system** to arrange the MFEs on the page. This needs a determination of how the MFEs on a page will communicate together, how dependencies are managed, etc. this is good for SEO because you can compose the page on the server and serve to crawlers when required.

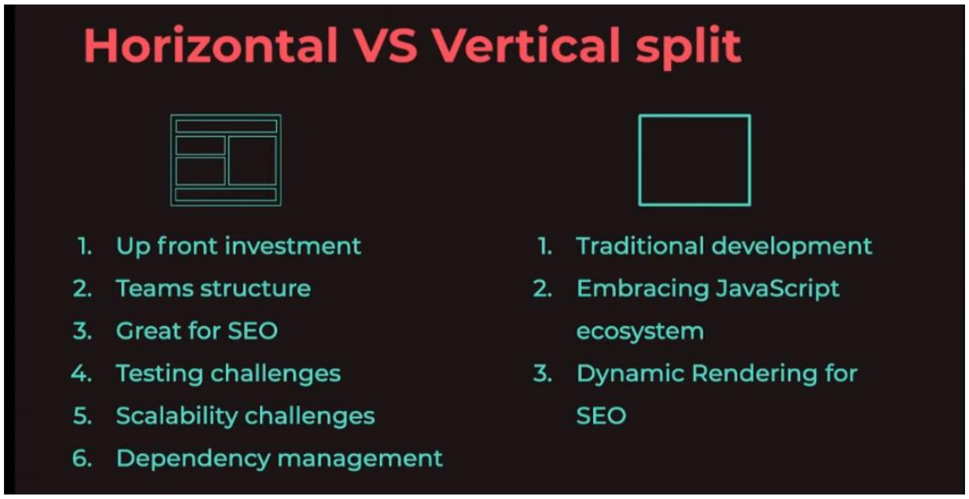

Define a micro-frontends

Another approach is to slice the application vertically with each MFE representing a separate HTML page or SPA.


Horizontal VS Vertical split

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

Horizontally, how we test the page end to end? Which team owns the page? How do we scale the horizontally when we have a spike in traffic to a page? Dependency management also involves how the final bundle size is made and optimized.

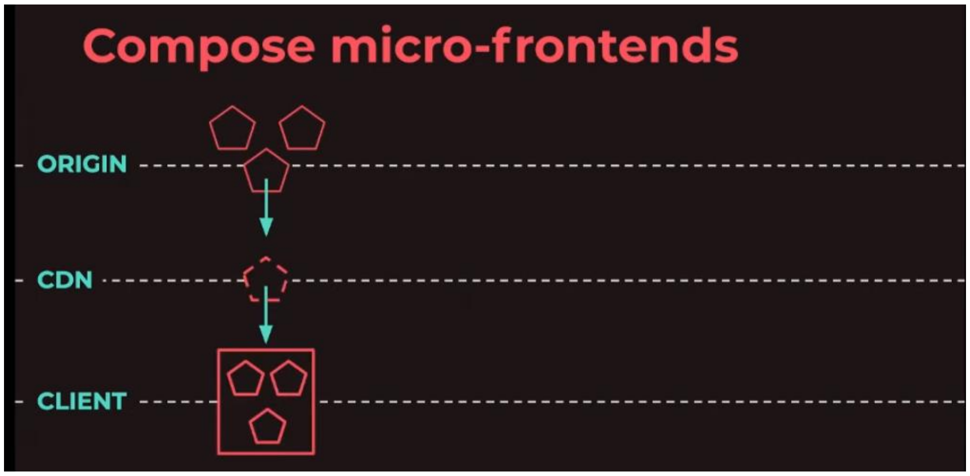


Horizontal VS Vertical split

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

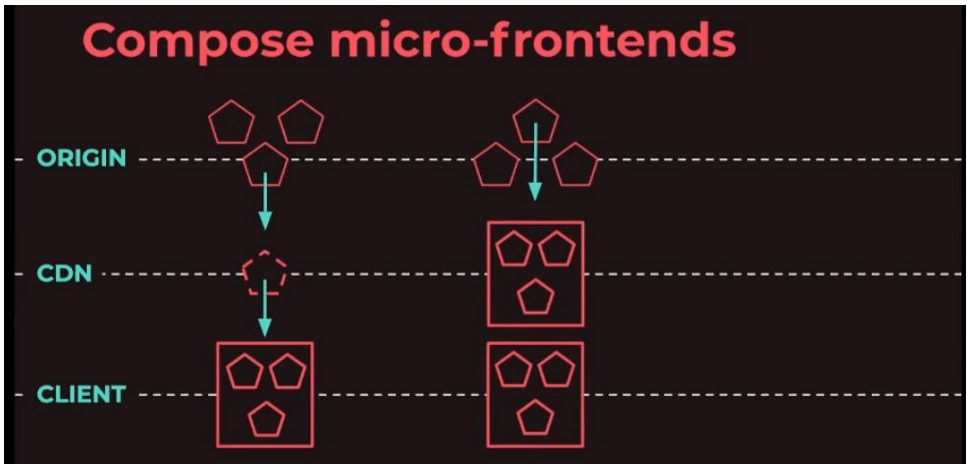
1. Traditional development
2. Embracing JavaScript ecosystem
3. Dynamic Rendering for SEO

Vertically, we have a lot of tools out there to use in normal development. You can use dynamic rendering for SEO to serve a specific version of the page that is optimized.



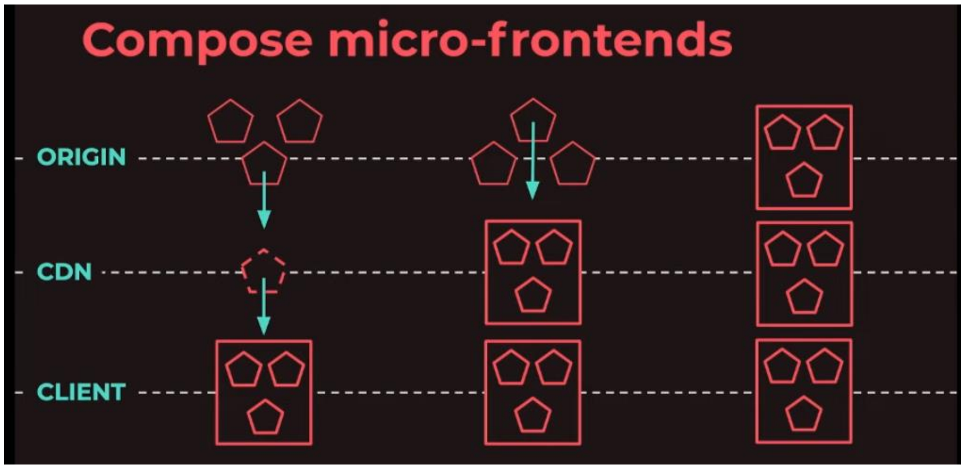
Compose micro-frontends

Here, we **compose on the client side using an app shell** that is always available and reside in the app. You can then use the app shell to load the different MFEs needed on a page or to load a SPA if vertically. Here, we will have all the different MFEs at a URL origin (an app server or a simple S3 bucket) to load from. We can also have a CDN that serves the content instead of going all the way to the origin server each time.

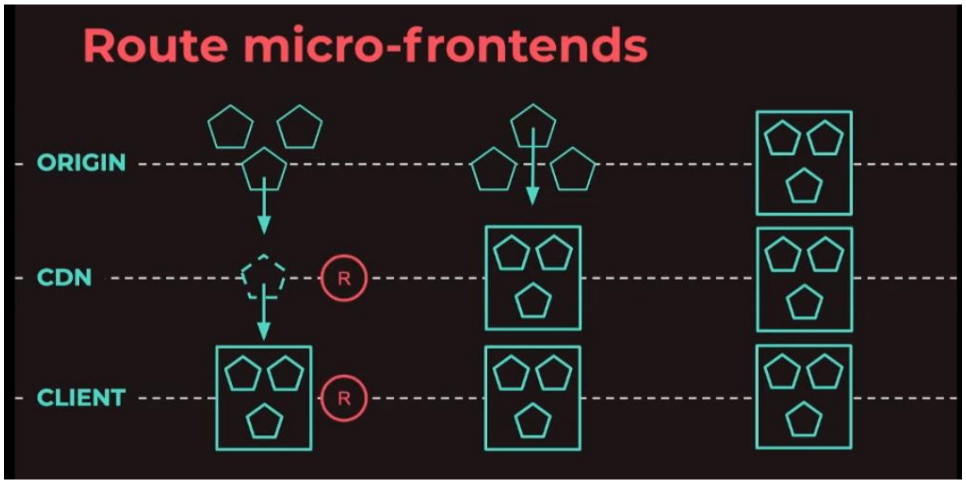


Compose micro-frontends

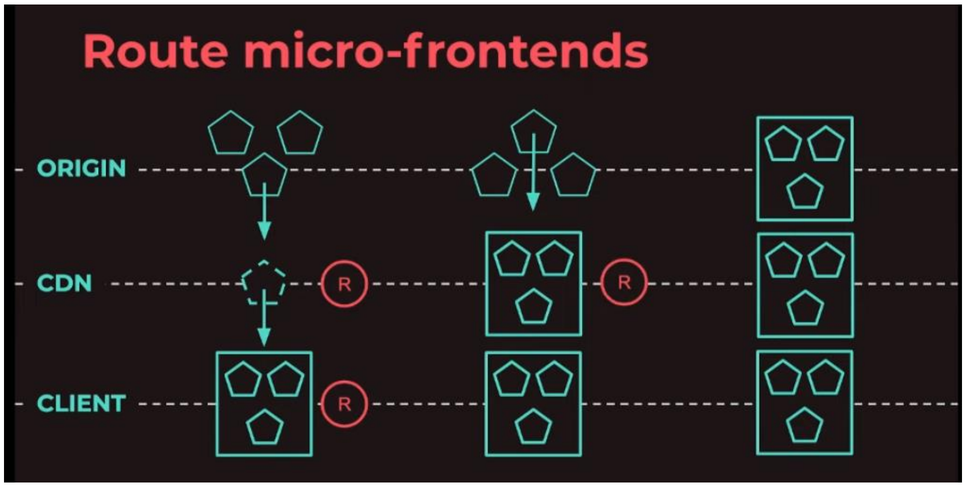
We can also compose MFEs by using **Edge Side Includes** (ESI) tags from Akamai CDN, this is a markup language that allows the CDN layer to fetch the MFEs and arrange onto the page before serving to the page. Testing on the CDN can be difficult



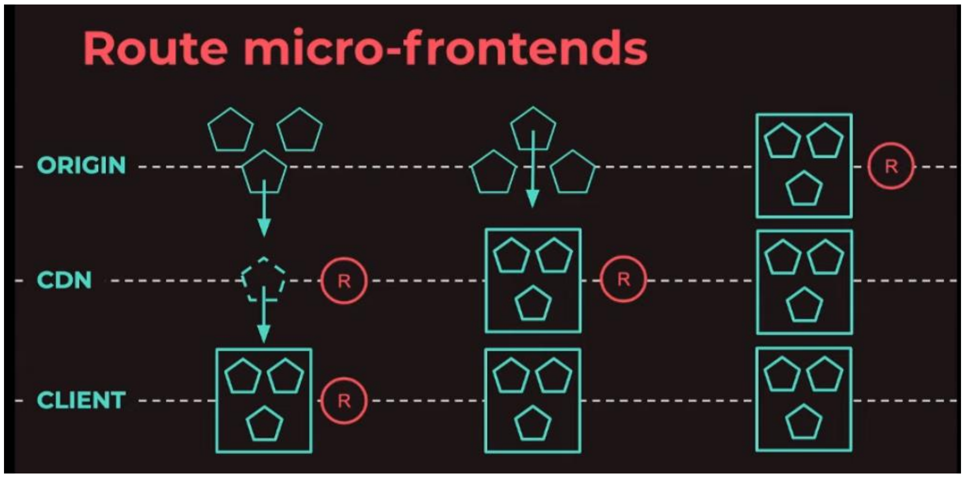
You can also compose at the origin like Zalando (using Mosaic9 and Taylor.JS that takes the fragments and stitch them together). Zolando has replaced that approach with React and GraphQL but still compose at the server-side and serve through a CDN. Other options are from OpenTable that compose at the origin and serve from a CDN.



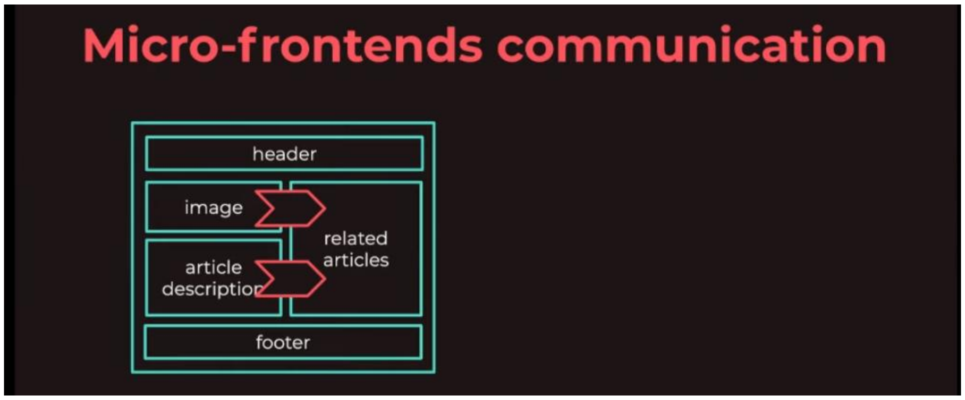
We can route on the CDN or route on the client side using micro-routing.



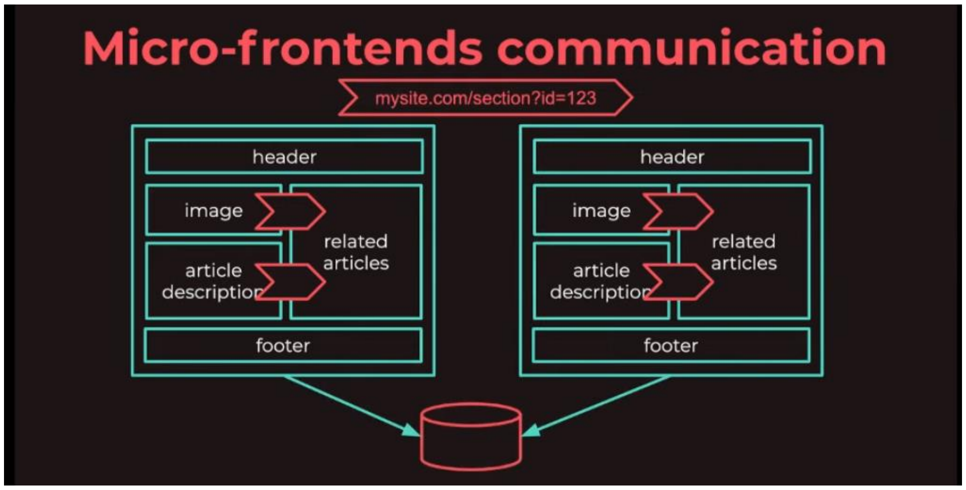
We can route on the CDN level when using ESI tags



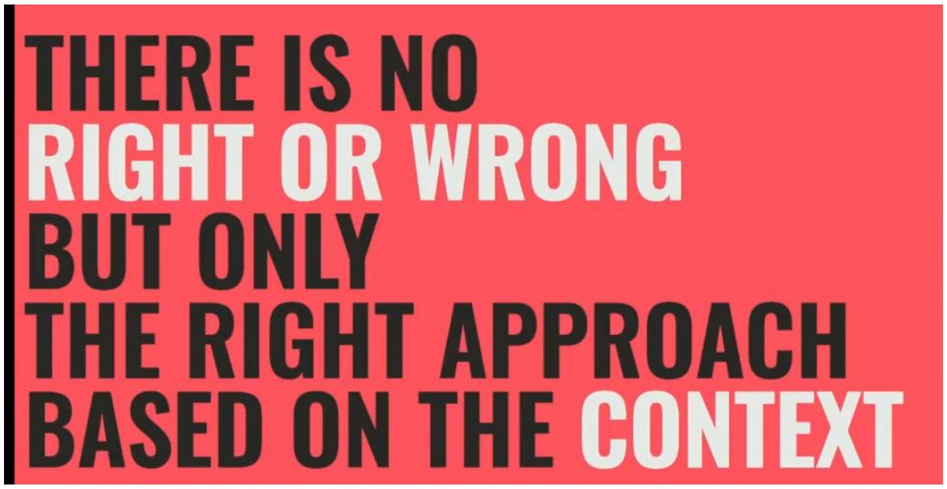
At the server side, we can do routing at the origin where we assemble the pages



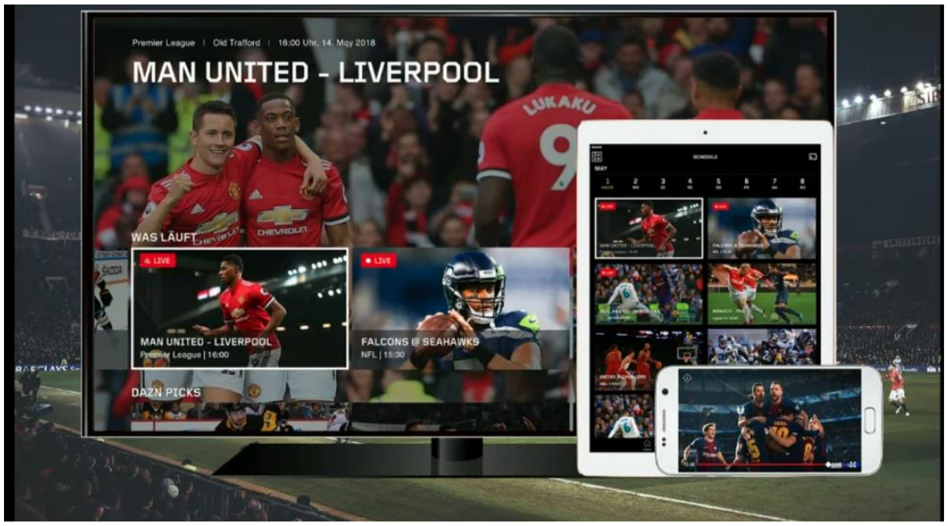
Horizontally, we can have components communicate with events on the DOM. The challenge is to find a way not to couple the MFEs, we can use **custom events by each MFE that bubble up and every other MFE is listening to and acting on if interested** or use an **event emitter that is owned by the app shell** to communicate to the residing MFEs.



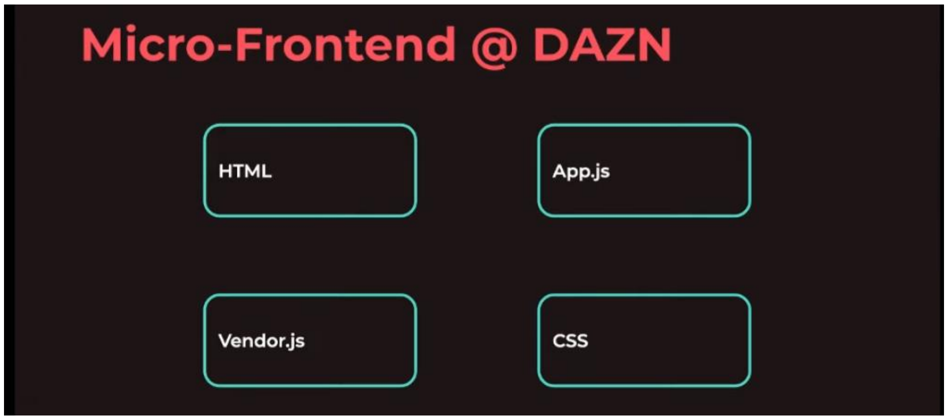
View or page-to-page communication can be done in 2 ways, we can use **normal URL parameters** (passing non-sensitive data via URL strings and having another MFE on the next page use that to consume an API it needs to render) or **web storage** (like putting in the auth token gotten after authentication in the web storage and having MFEs pick it from there to use when making requests).


THERE IS NO
RIGHT OR WRONG
BUT ONLY
THE RIGHT APPROACH
BASED ON THE CONTEXT

MICRO-FRONTENDS @DAZN

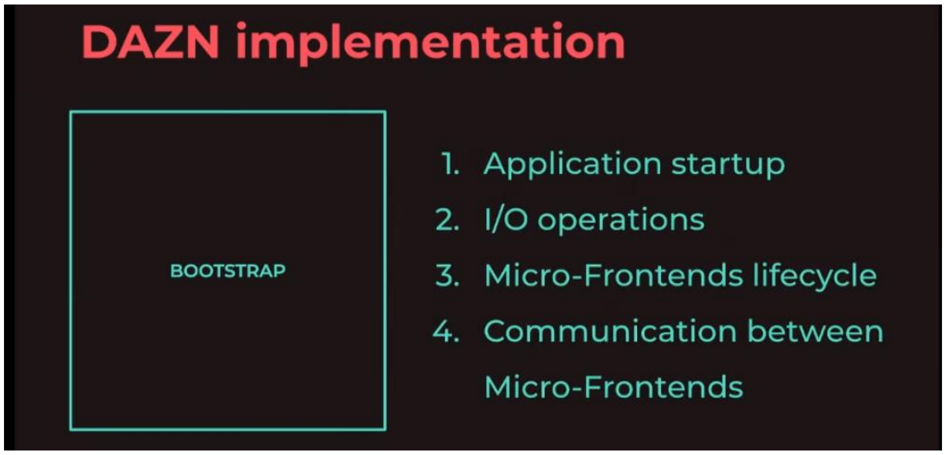


We have a lot of different clients (like web. Mobile, TV clients) that we have to serve


Micro-Frontend @ DAZN

- HTML
- App.js
- Vendor.js
- CSS

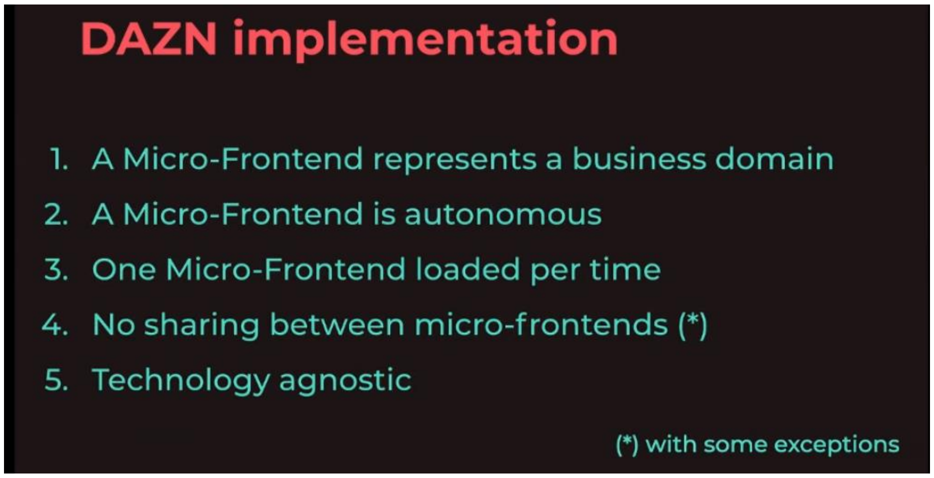
These are the artifacts that are in each MFE



DAZN implementation

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

Then we have something that orchestrates everything. We choose to do this on the client side with a bootstrap that is loaded first when the user goes to dazn.com. the bootstrap is a simple HTML page that contains some logic for loading the MFEs. The Application startup code is loaded that will detect which device you are using, which country you are, what feature flags to use, etc. The bootstrap also abstracts the input/output operations (needed for TVs). The bootstrap also triggers some lifecycle things like callbacks when it is mounted completely, or when it is about to be unmounted so that it can remove all the listeners that it had created in-memory. The bootstrap also helps with communication and storing things in the local storage.

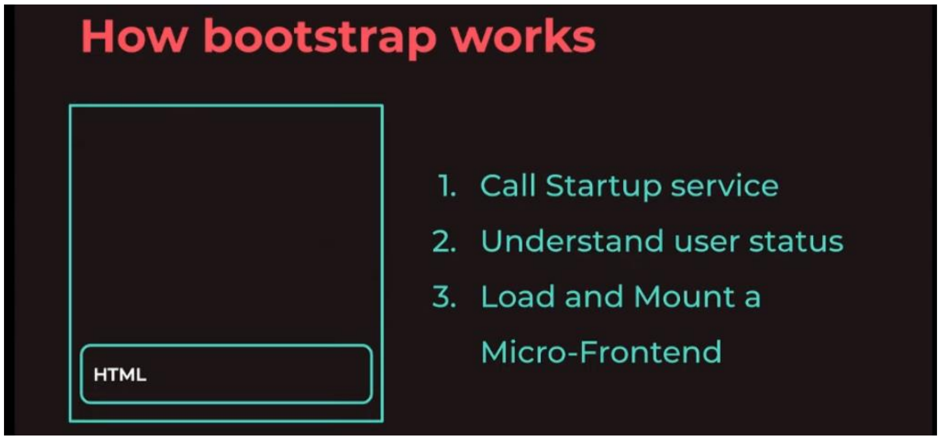


DAZN implementation

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

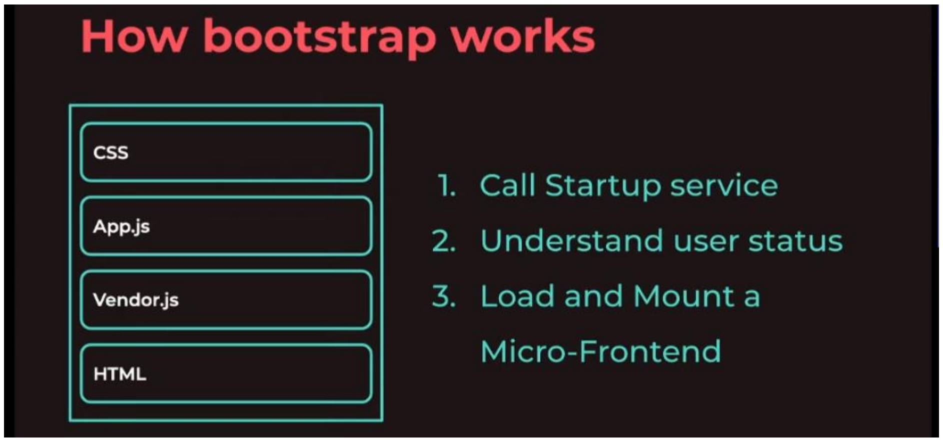
We have a payment shared library that is being shared by 2 MFEs to have the same version, the MFEs have duplicated parts and are updated and deployed together.



How bootstrap works

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

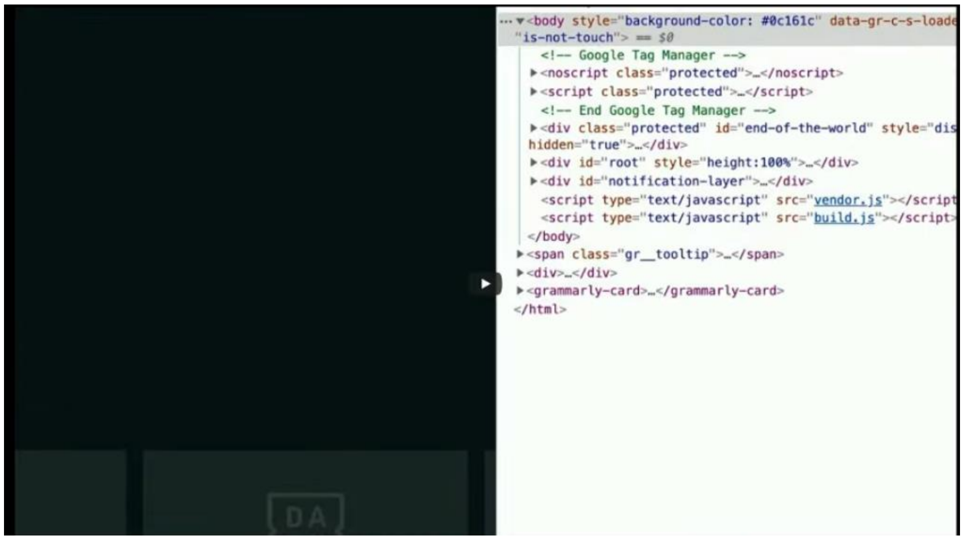
When we load the web app, it calls the Startup service API and retrieves a JSON file with a lot of information, the bootstrap then understands the user status by validating and reading the JWT token stored in local storage.



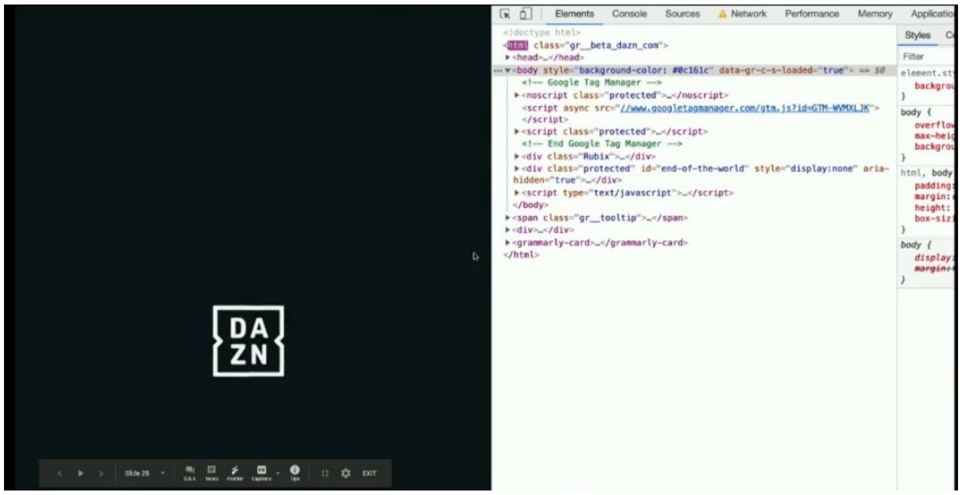
How bootstrap works

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

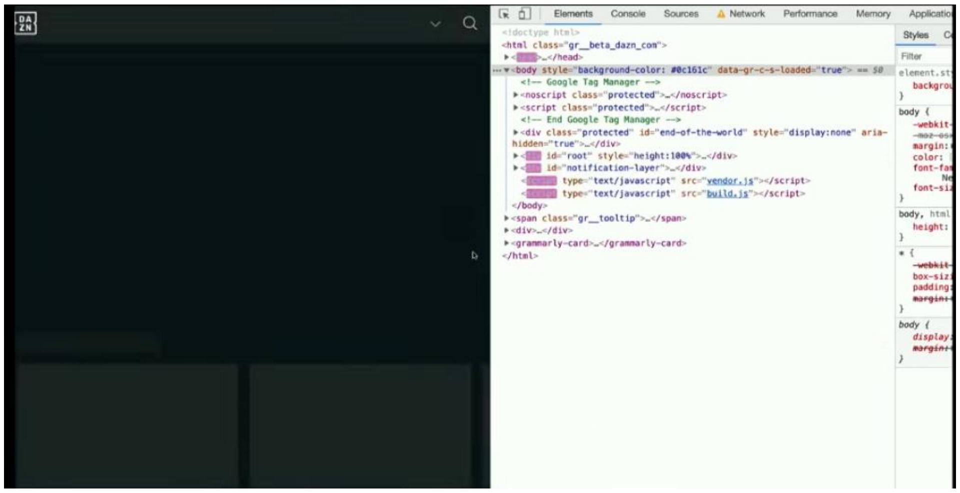
If user is valid, load the page skeleton and the HTML DOM elements for the page progressively (using server-side rendering).



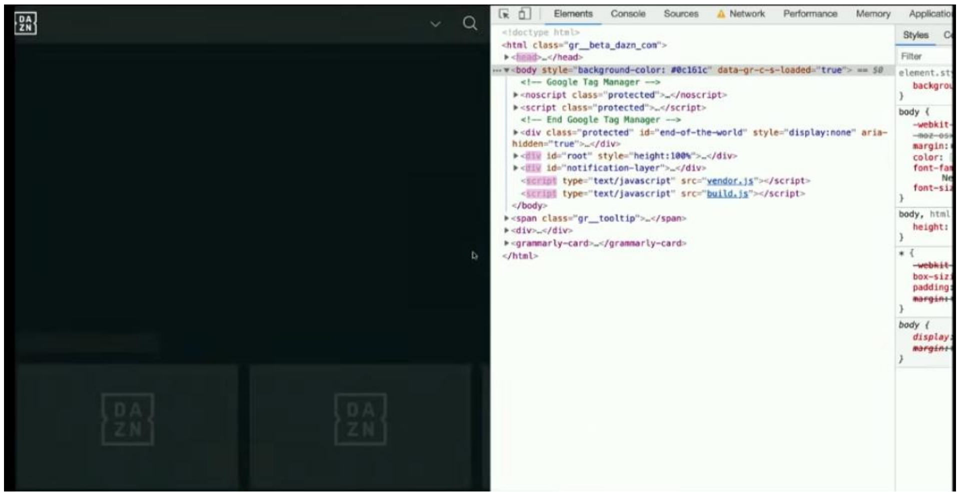
```
··· ▼<body style="background-color: #0c161c" data-gr-c-s-loade
  "is-not-touch"> == $0
      <!-- Google Tag Manager -->
    ▶<noscript class="protected">…</noscript>
    ▶<script class="protected">…</script>
      <!-- End Google Tag Manager -->
    ▶<div class="protected" id="end-of-the-world" style="dis
    hidden="true">…</div>
    ▶<div id="root" style="height:100%">…</div>
    ▶<div id="notification-layer">…</div>
      <script type="text/javascript" src="vendor.js"></script
      <script type="text/javascript" src="build.js"></script>
    </body>
  ▶<span class="gr__tooltip">…</span>
  ▶<div>…</div>
  ▶<grammarly-card>…</grammarly-card>
  </html>
```



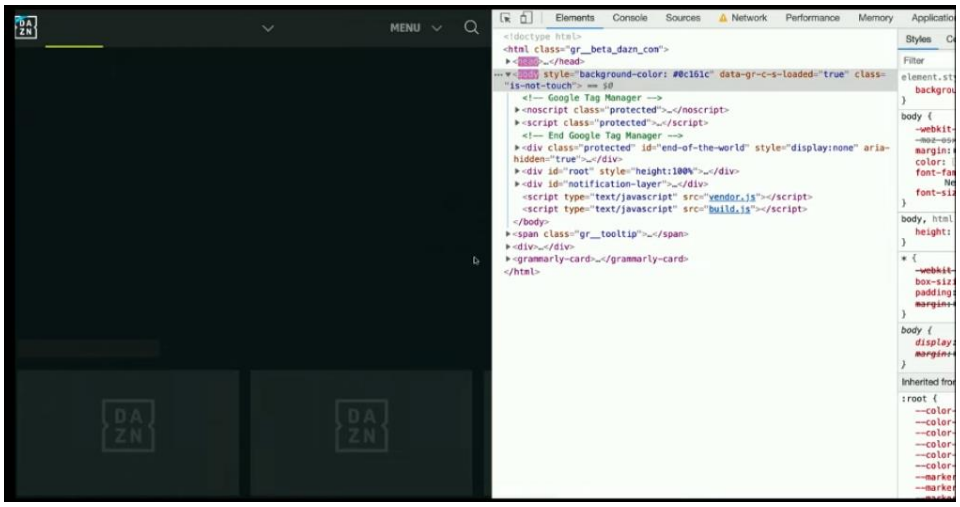
```
<!doctype html>
<html class="gr__beta_dazn_com">
 ▶<head>…</head>
··· ▼<body style="background-color: #0c161c" data-gr-c-s-loaded="true"> == $0
      <!-- Google Tag Manager -->
    ▶<noscript class="protected">…</noscript>
      <script async src="//www.googletagmanager.com/gtm.js?id=GTM-WVMXLJK">
      </script>
    ▶<script class="protected">…</script>
      <!-- End Google Tag Manager -->
    ▶<div class="Rubix">…</div>
    ▶<div class="protected" id="end-of-the-world" style="display:none" aria-
    hidden="true">…</div>
    ▶<script type="text/javascript">…</script>
    </body>
 ▶<span class="gr__tooltip">…</span>
 ▶<div>…</div>
 ▶<grammarly-card>…</grammarly-card>
</html>
```



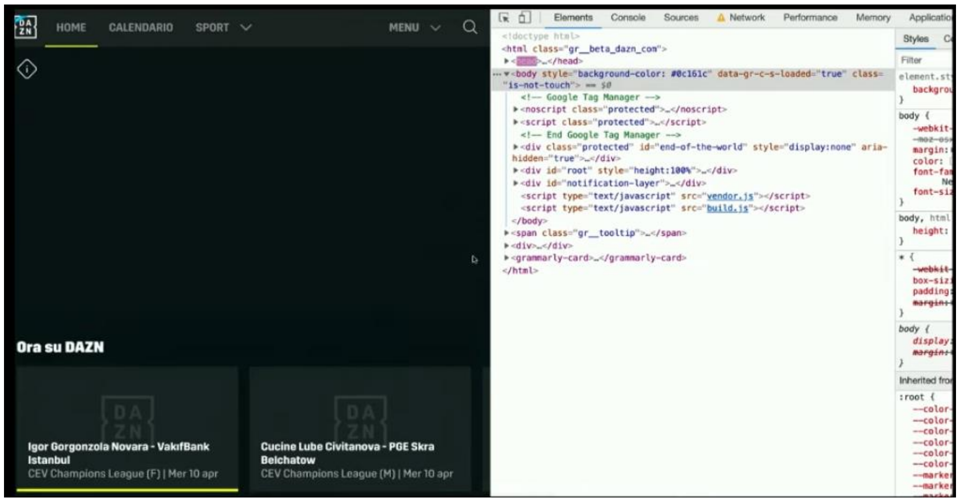
```
<!doctype html>
<html class="gr__beta_dazn_com">
 ▶<head>…</head>
…▼<body style="background-color: #0c161c" data-gr-c-s-loaded="true"> == $0
    <!-- Google Tag Manager -->
   ▶<noscript class="protected">…</noscript>
   ▶<script class="protected">…</script>
    <!-- End Google Tag Manager -->
   ▶<div class="protected" id="end-of-the-world" style="display:none" aria-
   hidden="true">…</div>
   ▶<div id="root" style="height:100%">…</div>
   ▶<div id="notification-layer">…</div>
    <script type="text/javascript" src="vendor.js"></script>
    <script type="text/javascript" src="build.js"></script>
  </body>
 ▶<span class="gr__tooltip">…</span>
 ▶<div>…</div>
 ▶<grammarly-card>…</grammarly-card>
</html>
```



```
<!doctype html>
<html class="gr__beta_dazn_com">
 ▶<head>…</head>
…▼<body style="background-color: #0c161c" data-gr-c-s-loaded="true"> == $0
    <!-- Google Tag Manager -->
   ▶<noscript class="protected">…</noscript>
   ▶<script class="protected">…</script>
    <!-- End Google Tag Manager -->
   ▶<div class="protected" id="end-of-the-world" style="display:none" aria-
   hidden="true">…</div>
   ▶<div id="root" style="height:100%">…</div>
   ▶<div id="notification-layer">…</div>
    <script type="text/javascript" src="vendor.js"></script>
    <script type="text/javascript" src="build.js"></script>
  </body>
 ▶<span class="gr__tooltip">…</span>
 ▶<div>…</div>
 ▶<grammarly-card>…</grammarly-card>
</html>
```



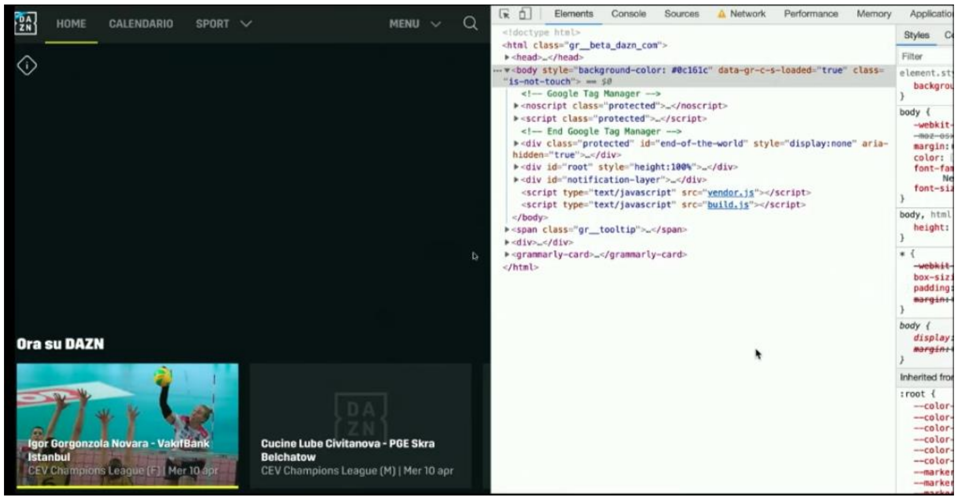
```
<!doctype html>
<html class="gr__beta_dazn_com">
 ▶<head>…</head>
…▼<body style="background-color: #0c161c" data-gr-c-s-loaded="true" class=
 "is-not-touch"> == $0
    <!-- Google Tag Manager -->
   ▶<noscript class="protected">…</noscript>
   ▶<script class="protected">…</script>
    <!-- End Google Tag Manager -->
   ▶<div class="protected" id="end-of-the-world" style="display:none" aria-
   hidden="true">…</div>
   ▶<div id="root" style="height:100%">…</div>
   ▶<div id="notification-layer">…</div>
    <script type="text/javascript" src="vendor.js"></script>
    <script type="text/javascript" src="build.js"></script>
  </body>
 ▶<span class="gr__tooltip">…</span>
 ▶<div>…</div>
 ▶<grammarly-card>…</grammarly-card>
</html>
```



Igor Gorgonzola Novara - VakifBank Istanbul
CEV Champions League (F) | Mer 10 apr

Cucine Lube Civitanova - PGE Skra Belchatow
CEV Champions League (M) | Mer 10 apr

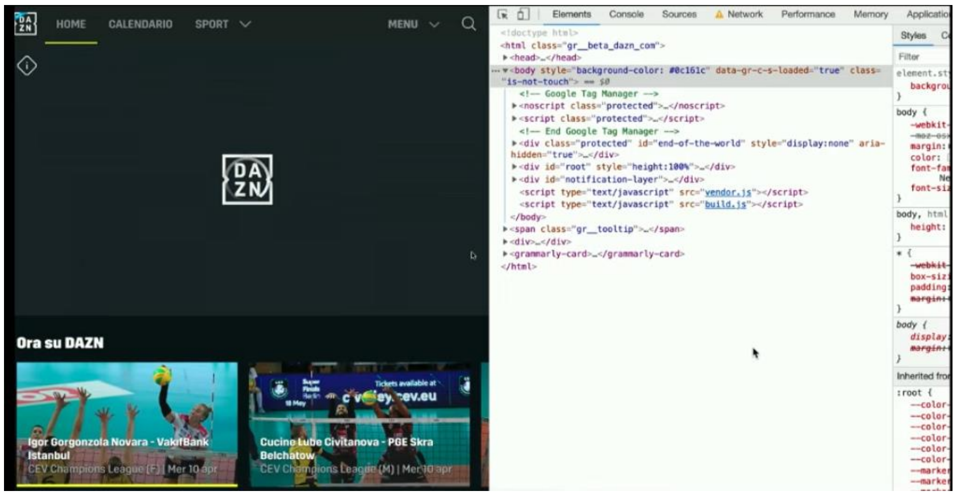
```
<!doctype html>
<html class="gr__beta_dazn_com">
 ▶<head>…</head>
… ▼<body style="background-color: #0c161c" data-gr-c-s-loaded="true" class=
"is-not-touch"> == $0
    <!-- Google Tag Manager -->
   ▶<noscript class="protected">…</noscript>
   ▶<script class="protected">…</script>
    <!-- End Google Tag Manager -->
   ▶<div class="protected" id="end-of-the-world" style="display:none" aria-
  hidden="true">…</div>
   ▶<div id="root" style="height:100%">…</div>
   ▶<div id="notification-layer">…</div>
    <script type="text/javascript" src="vendor.js"></script>
    <script type="text/javascript" src="build.js"></script>
  </body>
 ▶<span class="gr__tooltip">…</span>
 ▶<div>…</div>
 ▶<grammarly-card>…</grammarly-card>
</html>
```



Igor Gorgonzola Novara - VakifBank Istanbul
CEV Champions League (F) | Mer 10 apr

Cucine Lube Civitanova - PGE Skra Belchatow
CEV Champions League (M) | Mer 10 apr

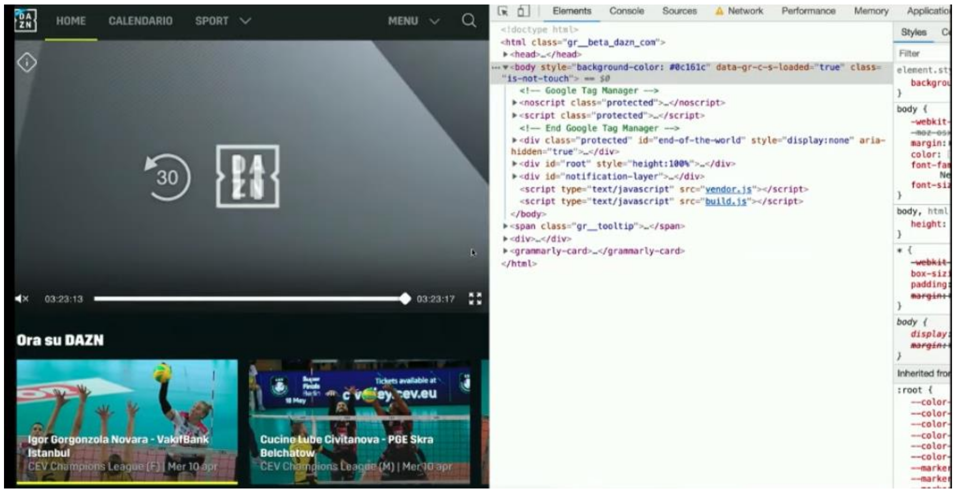
```
<!doctype html>
<html class="gr__beta_dazn_com">
 ▶<head>…</head>
… ▼<body style="background-color: #0c161c" data-gr-c-s-loaded="true" class=
"is-not-touch"> == $0
    <!-- Google Tag Manager -->
   ▶<noscript class="protected">…</noscript>
   ▶<script class="protected">…</script>
    <!-- End Google Tag Manager -->
   ▶<div class="protected" id="end-of-the-world" style="display:none" aria-
  hidden="true">…</div>
   ▶<div id="root" style="height:100%">…</div>
   ▶<div id="notification-layer">…</div>
    <script type="text/javascript" src="vendor.js"></script>
    <script type="text/javascript" src="build.js"></script>
  </body>
 ▶<span class="gr__tooltip">…</span>
 ▶<div>…</div>
 ▶<grammarly-card>…</grammarly-card>
</html>
```



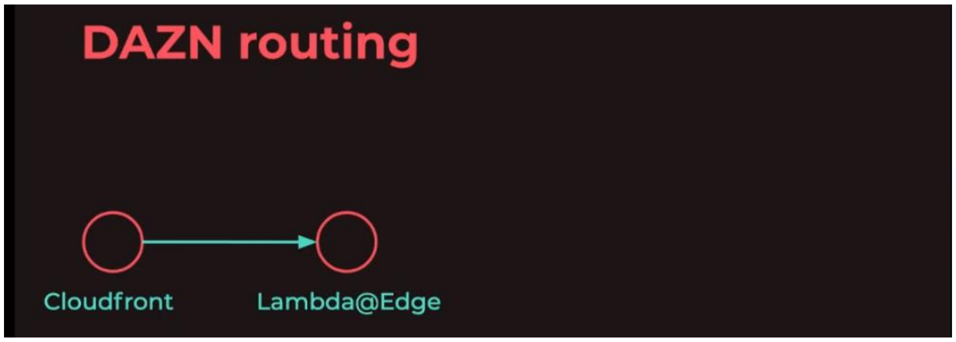
Igor Gorgonzola Novara - VakifBank Istanbul
CEV Champions League (F) | Mer 10 apr

Cucine Lube Civitanova - PGE Skra Belchatow
CEV Champions League (M) | Mer 10 apr

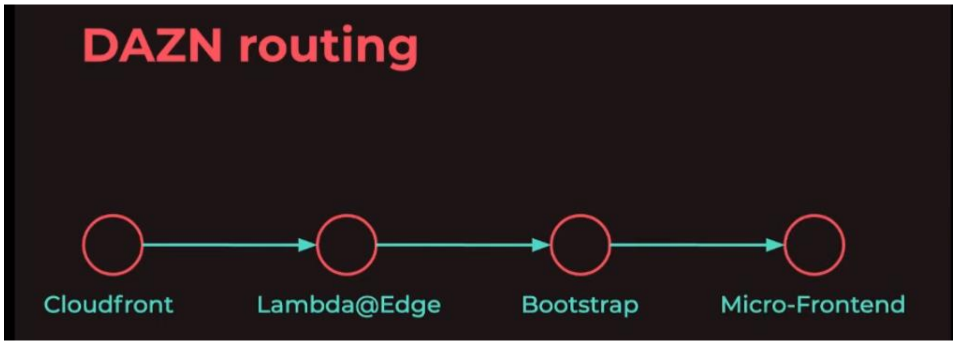
```
<!doctype html>
<html class="gr__beta_dazn_com">
 ▶<head>…</head>
… ▼<body style="background-color: #0c161c" data-gr-c-s-loaded="true" class=
"is-not-touch"> == $0
    <!-- Google Tag Manager -->
   ▶<noscript class="protected">…</noscript>
   ▶<script class="protected">…</script>
    <!-- End Google Tag Manager -->
   ▶<div class="protected" id="end-of-the-world" style="display:none" aria-
  hidden="true">…</div>
   ▶<div id="root" style="height:100%">…</div>
   ▶<div id="notification-layer">…</div>
    <script type="text/javascript" src="vendor.js"></script>
    <script type="text/javascript" src="build.js"></script>
  </body>
 ▶<span class="gr__tooltip">…</span>
 ▶<div>…</div>
 ▶<grammarly-card>…</grammarly-card>
</html>
```


HOME
CALENDARIO
SPORT
MENU
03:23:13
03:23:17
Ora su DAZN
Igor Gorgonzola Novara - VakifBank Istanbul
CEV Champions League (F) | Mer 10 apr
Cucine Lube Civitanova - PGE Skra Belchatow
CEV Champions League (M) | Mer 10 apr
Elements
Console
Sources
Network
Performance
Memory
Application
Styles

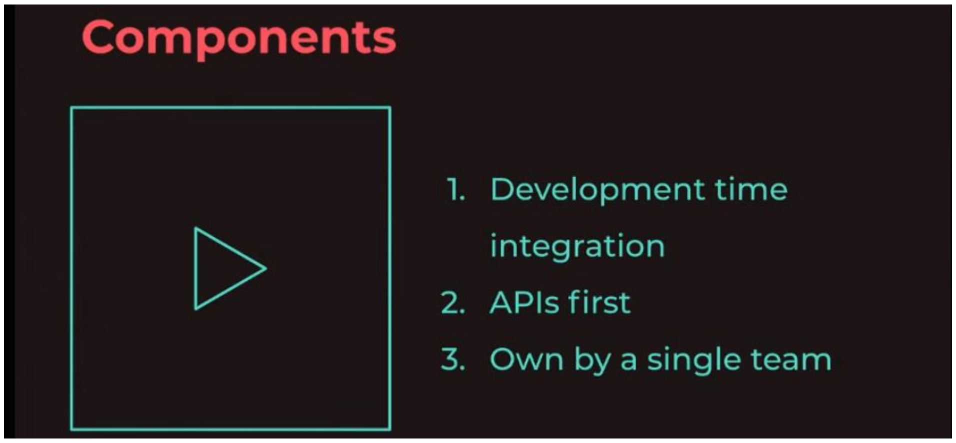
```
<!doctype html>
<html class="gr__beta_dazn_com">
 ▶<head>…</head>
…▼<body style="background-color: #0c161c" data-gr-c-s-loaded="true" class=
 "is-not-touch"> == $0
    <!-- Google Tag Manager -->
   ▶<noscript class="protected">…</noscript>
   ▶<script class="protected">…</script>
    <!-- End Google Tag Manager -->
   ▶<div class="protected" id="end-of-the-world" style="display:none" aria-
   hidden="true">…</div>
   ▶<div id="root" style="height:100%">…</div>
   ▶<div id="notification-layer">…</div>
    <script type="text/javascript" src="vendor.js"></script>
    <script type="text/javascript" src="build.js"></script>
  </body>
 ▶<span class="gr__tooltip">…</span>
 ▶<div>…</div>
 ▶<grammarly-card>…</grammarly-card>
</html>
```



Our infra is simple, we use Cloudfront (caching and with Lambda@Edge) and S3. We generate and do everything at compile time and then store our artifacts in S3 and use Cloudfront to serve them. Cloudfront runs some lambdas at the closest edge nodes to our users to implement some logic like doing canary routing between app versions based on the browser type.



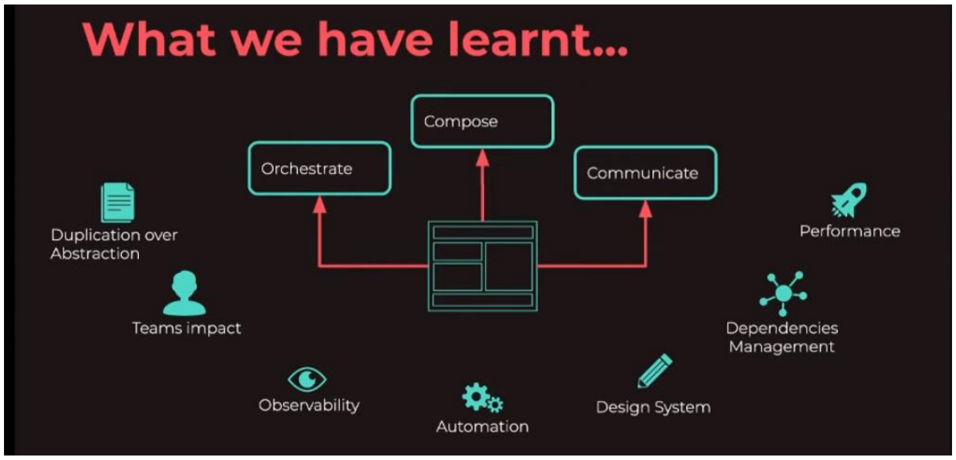
The Lambdas at the edge are responsible for loading the bootstrap and the bootstrap is responsible for loading the page (either the authenticated or unauthenticated page versions).



Components

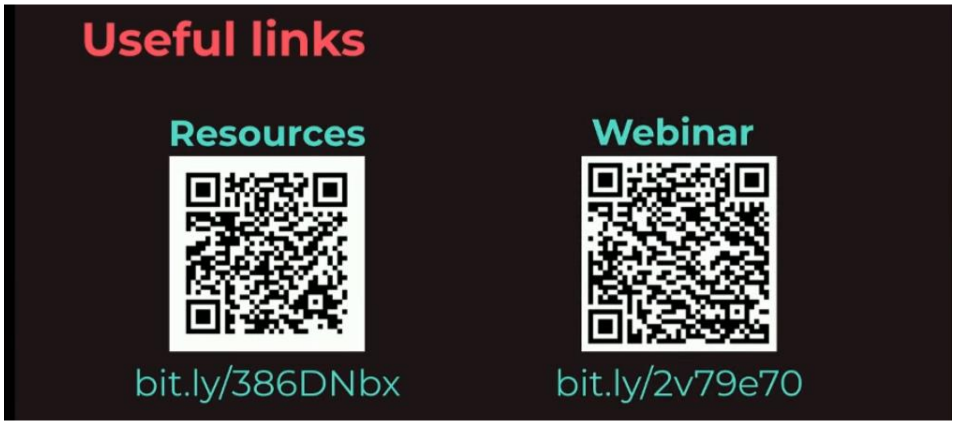
1. Development time integration
2. APIs first
3. Own by a single team

Complex parts like the video player are developed as separate component library and published on our private NPM registry, any MFE that needs that component then imports that library and use it.



What we have learnt...

How to startup, compose, orchestrate, communicate between our MFEs. We also need to invest in automation, observability (using tools like LogRocket, Sentry etc to know what is happening), we also need a design system for our components that each MFE can implement. We can optimize for less than 4secs to load a page.



Useful links

Resources

bit.ly/386DNbx

Webinar

bit.ly/2v79e70

buildingmicrofrontends.com