



Micro Frontends

QCon
SAN FRANCISCO by **InfoQ**

SF, 2022-10-28
Dr. Florian Rapp

Hello, I'm Florian!



Solution Architect at smapiot

IoT / Embedded Computing

Digital Transformation

Distributed Web Applications

Open-Source Enthusiast

Microsoft MVP Development Tools

Projects for .NET and JS

Articles for various blogs and magazines

Hello, you are?

Background?

Current position?

Experiences?

Expectations?

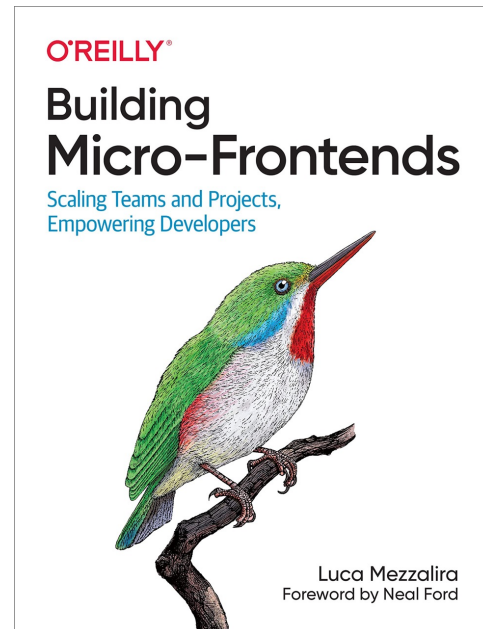
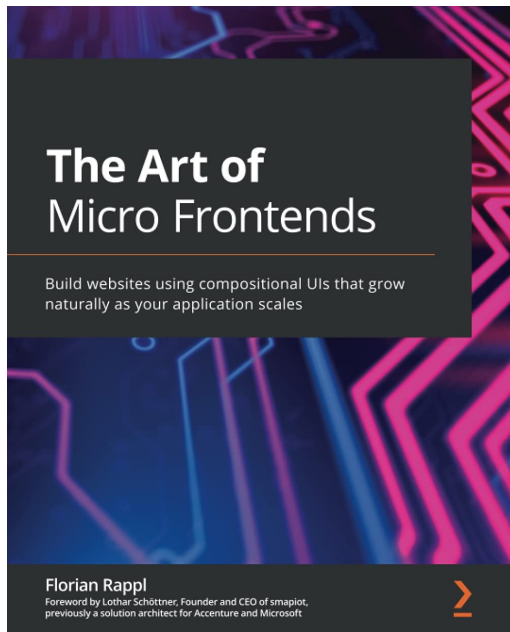
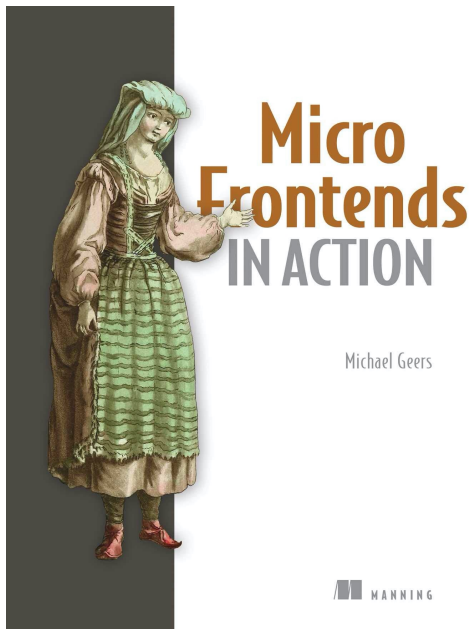
Access to the Examples

Send me your GitHub account to get access to:

<https://github.com/FlorianRappl/qconsf-mf-workshop>

(public for now, will be made private at lunch break)

Available Books



Who uses Micro Frontends?



zalando



breuninger



HELLO
FRESH



FINN

OTTO



PayPal



Alibaba Group



Thalia

KLINGEL

Gruppe

REWE

SAP



StepStone



Spotify



Klarna



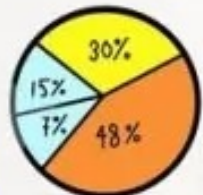
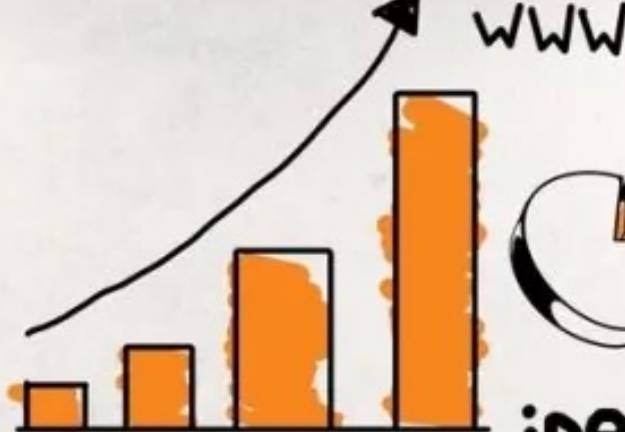
namecheap

axinom!

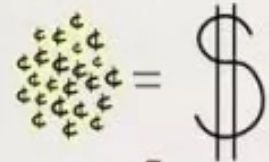
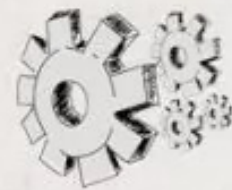


smoplot

rbi



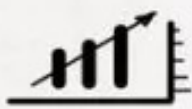
idea 45% is SALE!



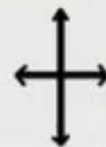
25% time



TEAM



internet



MARKETING



SUCCESS

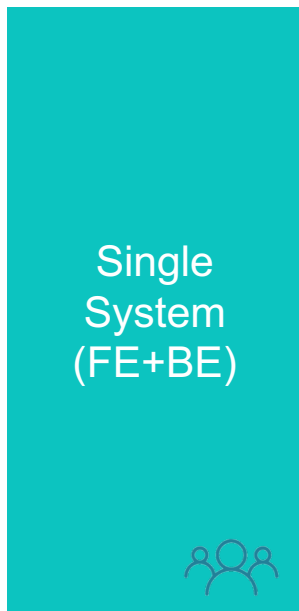




Scalability is the ability to
handle changing demands
on a business.

Web Evolution

Monolith



Sep. BE/FE



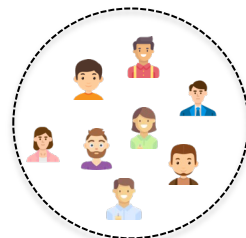
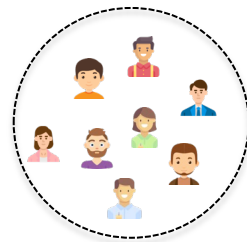
Microservices



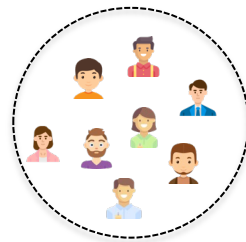
MFs



Distribution of Work



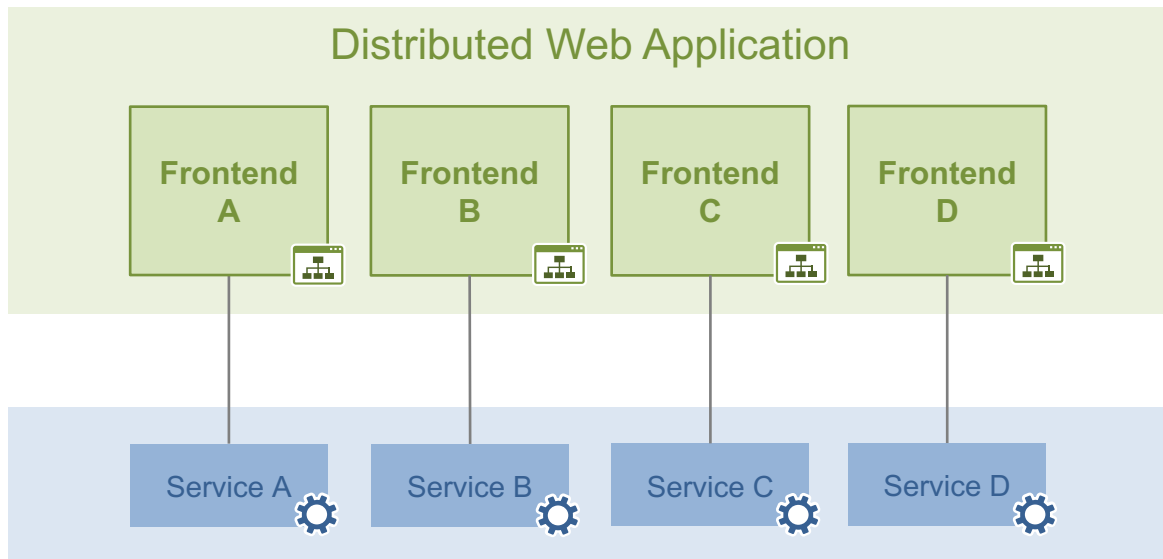
three 8-person teams



are more effective than

one 24-person team

Micro Frontend Architecture



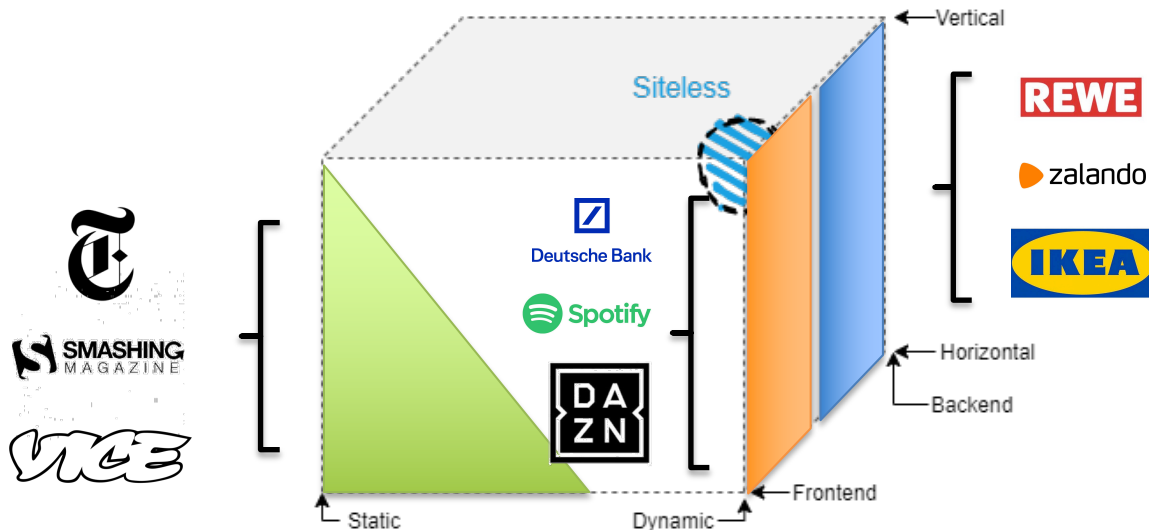
Frontend

Micro Frontend

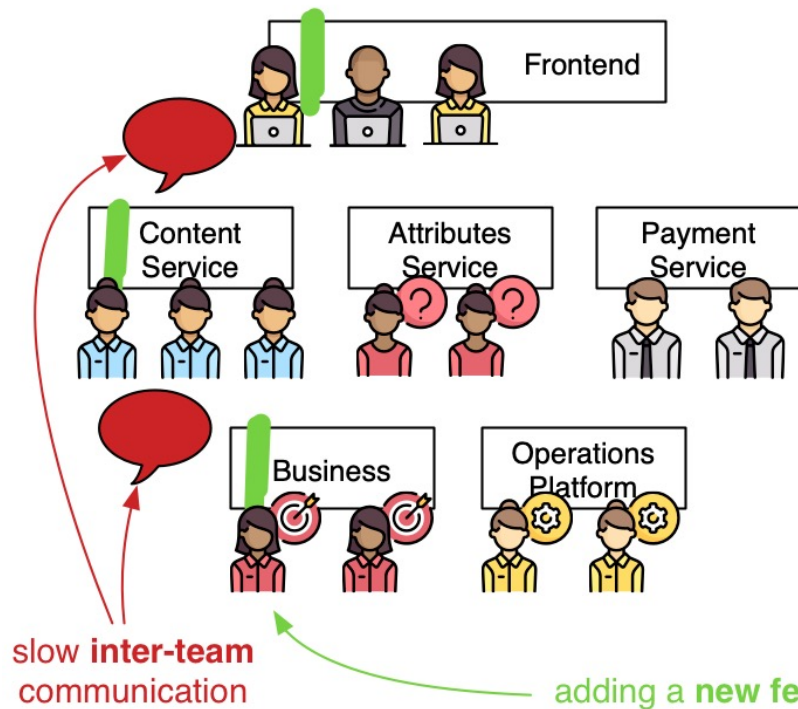
Backend

Microservice

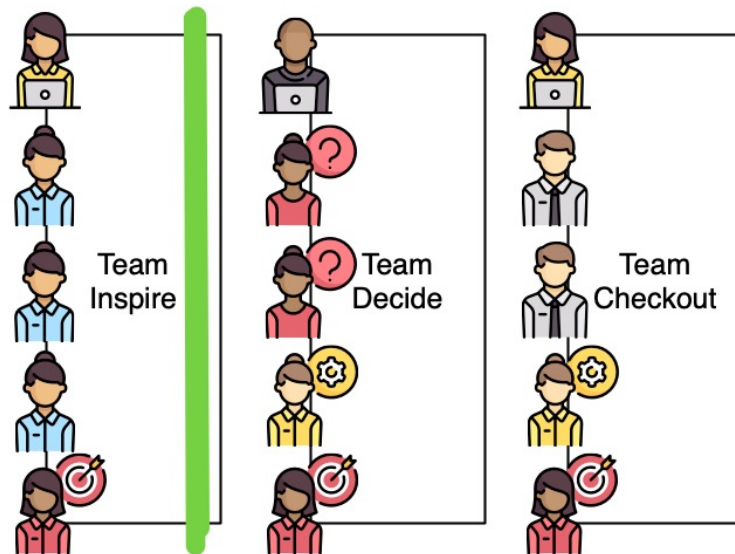
MF Architecture Solution Space



Microservices



Micro Frontends



Graphic by M. Geers

The Model Store

basket: 0 item(s)



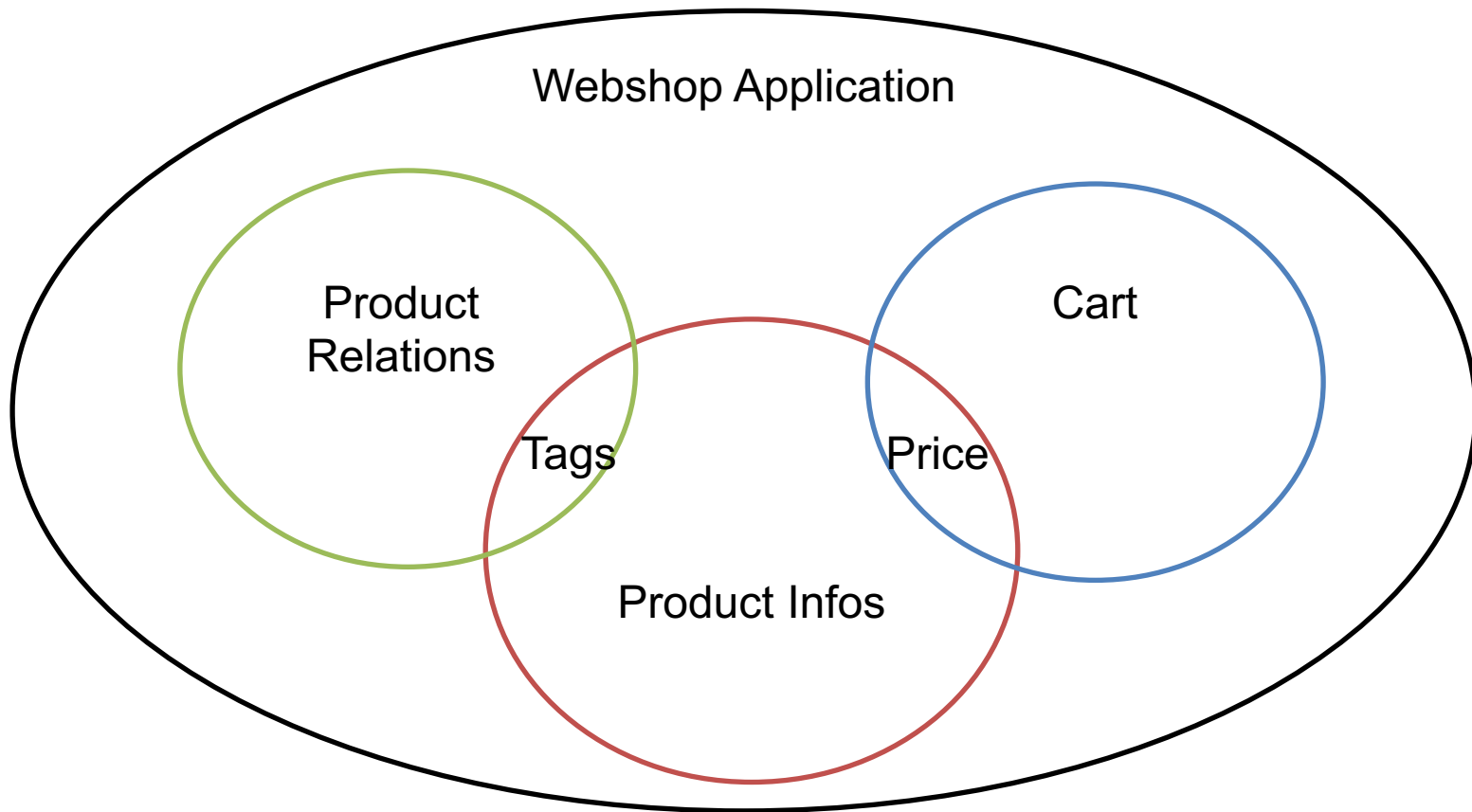
Tractor Porsche-Diesel Master
419



buy for 66,00 €

Related Products





The Model Store

basket: 0 item(s)

Related Products



Tractor Porsche-Diesel Master
419



buy for 66,00 €



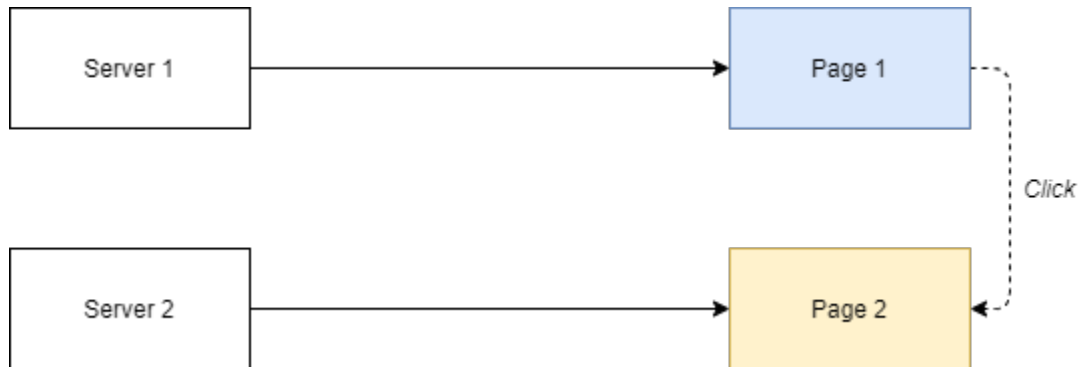
Team Product

Team Checkout

Team Inspire

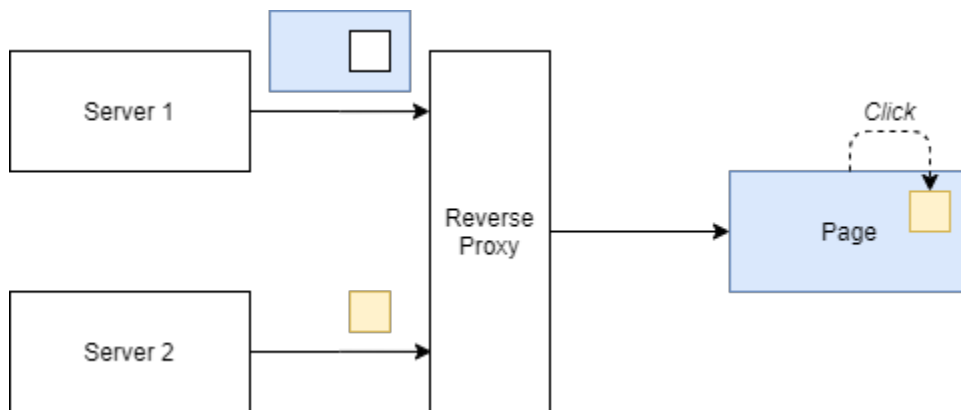
Architecture Pattern

Web Approach



Architecture Pattern

Server-Side Composition



Podium

- Server-side composition of smaller frontend parts called “podlets”
- Aggregated in a server using a layout

More information: <https://podium-lib.io/>

Architecture Pattern

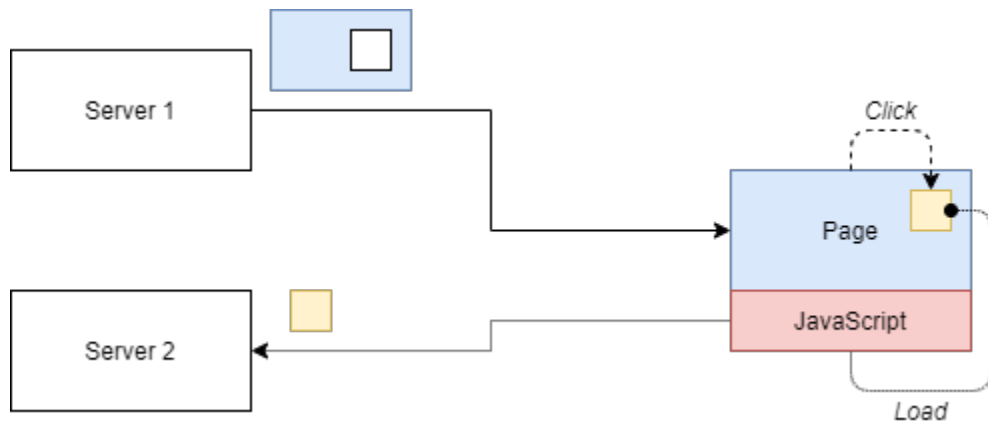
Server-Side Composition



GitHub repository
demos/demo-01

Architecture Pattern

Client-Side Composition



Web Components

- Combination of different technologies
 - Shadow DOM (style / node isolation)
 - Custom elements (register a custom tag name)
 - HTML templates (pre-parse trees into reusable fragments)
- They can be used to transport and inject a full micro document (“component”) seamlessly, i.e., without using an iframe

Web Components

```
class MyElement extends HTMLElement {  
  constructor() { super(); }  
  connectedCallback() {}  
  disconnectedCallback() {}  
}
```

```
customElements.define('my-element', MyElement);
```


Architecture Pattern

Client-Side Composition



GitHub repository
demos/demo-02

Exercise (1)

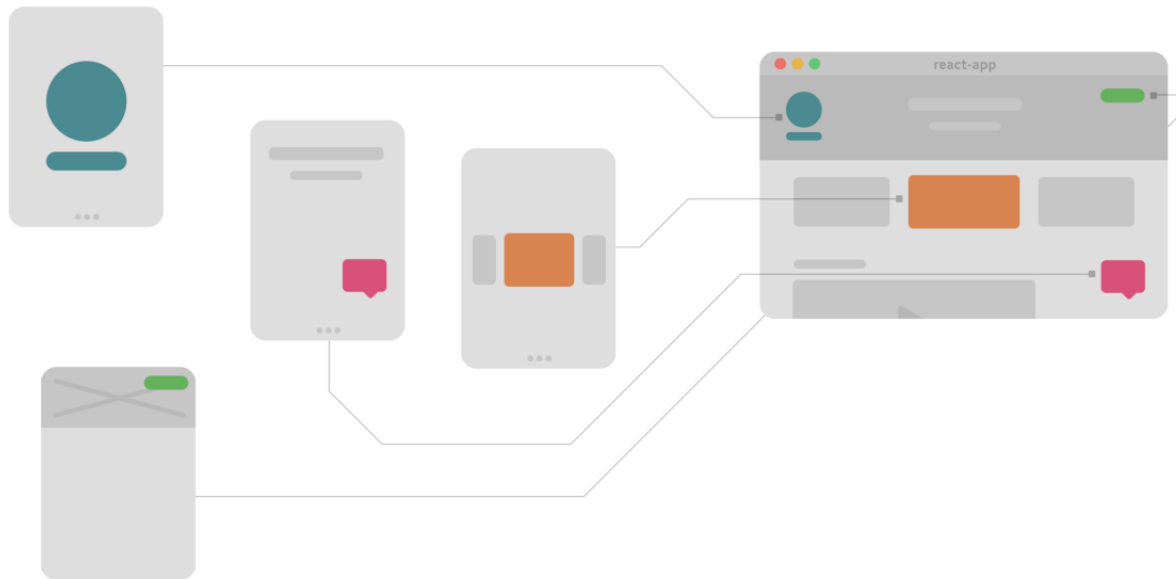
Write another micro frontend (MF) for the tractor shop using web components. The MF should show reviews related to the currently selected product.

Questions:

- How can the new MF be integrated?
- How to debug the new MF in case of issues?

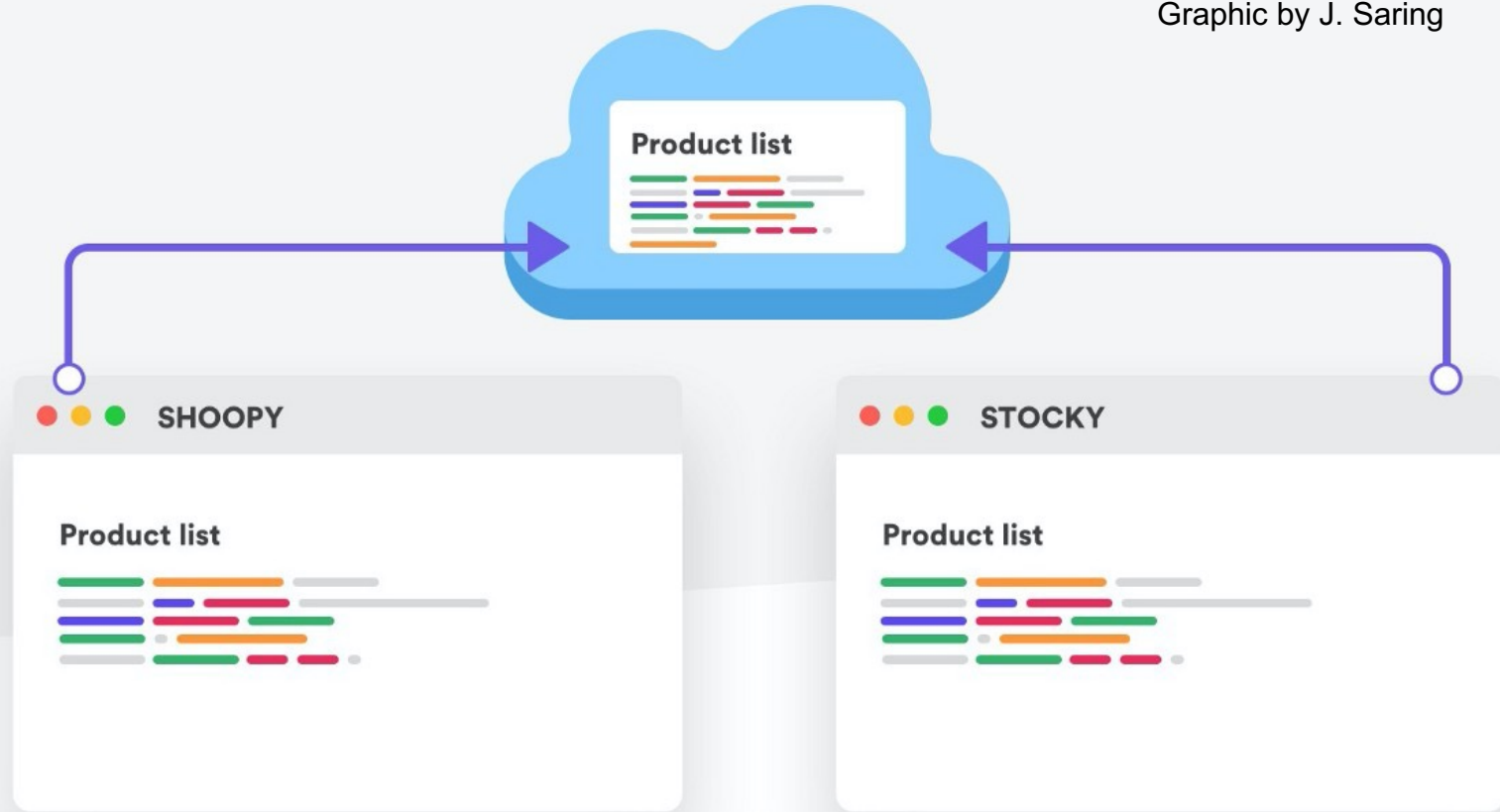
→ Use the „exercises/exercise-01“ directory as a starting point.

UX Scaling



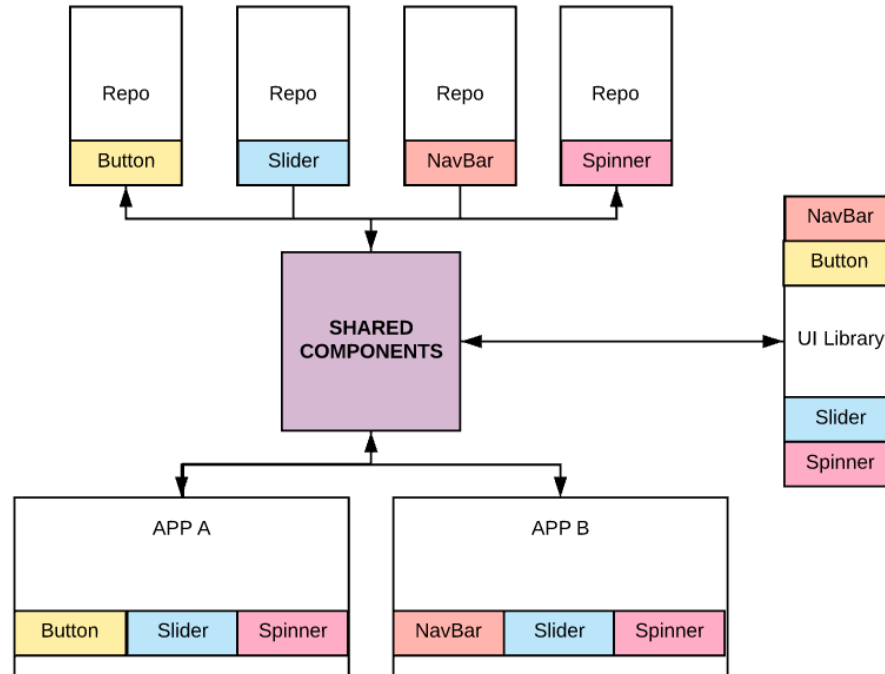
Graphic by J. Saring

Graphic by J. Saring

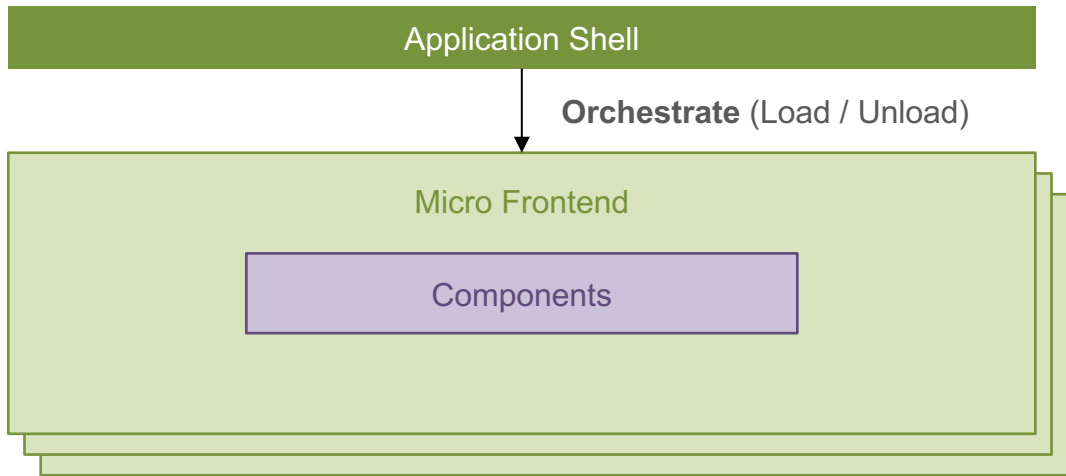




Graphic by J. Saring

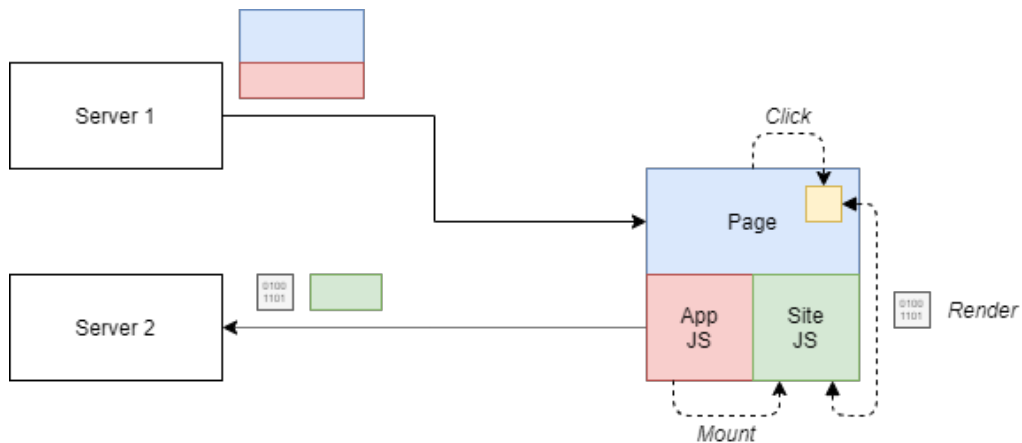


		Application Shell	
		NO	YES
Shared Libraries	YES	Framework shared libs only	„MODULITH“ Architecture Foundation, Design & Shared Libraries
	NO	Empty No basis	Layout Design only



Architecture Pattern

SPA Composition



single-spa

- First framework to wrap components in a generic lifecycle
- Can render any framework when their lifecycle is wrapped properly
- Already has lifecycle wrappers for Angular, React, ...
- Enforces strong coupling through a “root config”, which declares what components exist and when they should be rendered
- Can be also considered a meta router, i.e., router of routers
- See <https://single-spa.js.org/>

single-spa

- Two important concepts:
 - Application
 - Usable at „root“ level, i.e., directly in the app shell
 - Directly bootstrapped in the app shell or from a „root config“ library / micro frontend (defines when to activate / deactivate)
 - Parcel
 - Single component used in an application or another Parcel
 - Same lifecycle as an application, but without root config to define activation / deactivation

Architecture Patterns

SPA Composition with single-spa



GitHub repository
demos/demo-03

Exercise (2)

Write another MF for the webshop using single-spa. The MF should show reviews related to the currently selected product.

Questions:

- How are the lifecycles handled?
- How is the integration done?

→ Use the „exercises/exercise-02“ directory as a starting point.

Webpack Module Federation

- Webpack plugin
- First released with Webpack 5, part of Webpack core (no install necessary)
- Allows declaring
 - “remotes” (other chunks) to import
 - exports from the “host” for others to import
- Chunks can be arbitrary complex, versioned, and decorated with lifecycle rules (e.g., singleton)
- See <https://webpack.js.org/concepts/module-federation/>

Architecture Patterns

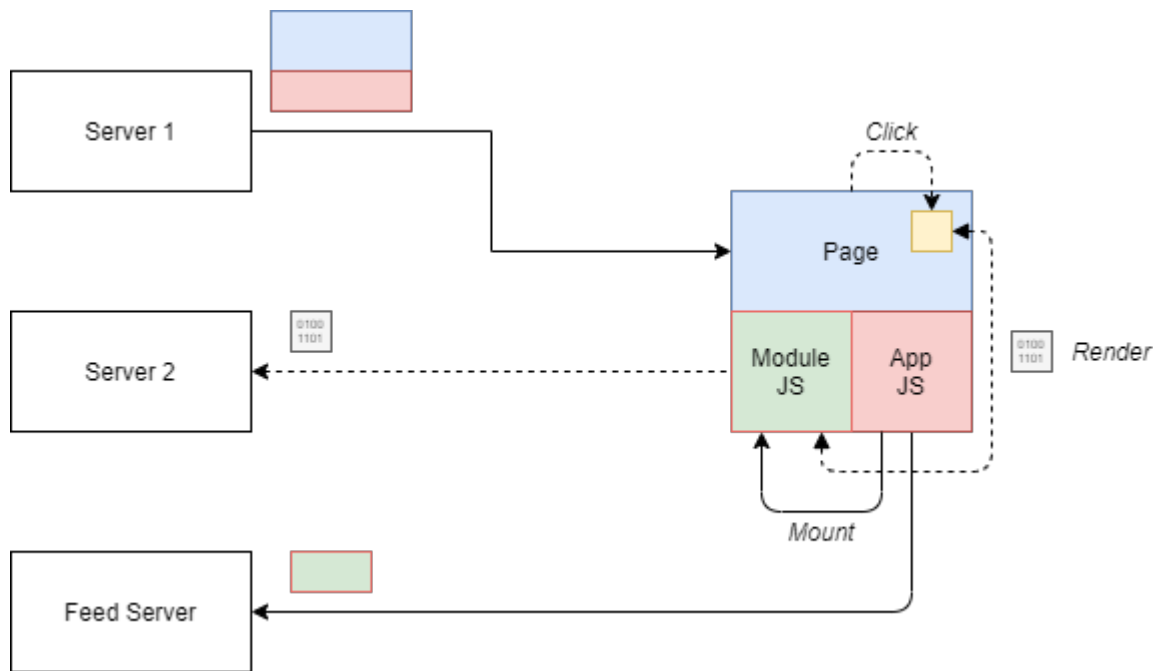
SPA Composition using Webpack



GitHub repository
demos/demo-04

Architecture Pattern

Siteless UIs



```
fetch('http://myfeed.com/modules')
  .then(res => res.json())
  .then(urls => urls
    .forEach(url => {
      const s = document.createElement('script');
      s.type = 'application/javascript';
      s.src = url;
      s.onload = () => { /* ... */ };
      s.onerror = () => { /* ... */ };
      document.body.appendChild(s);
    }));
```

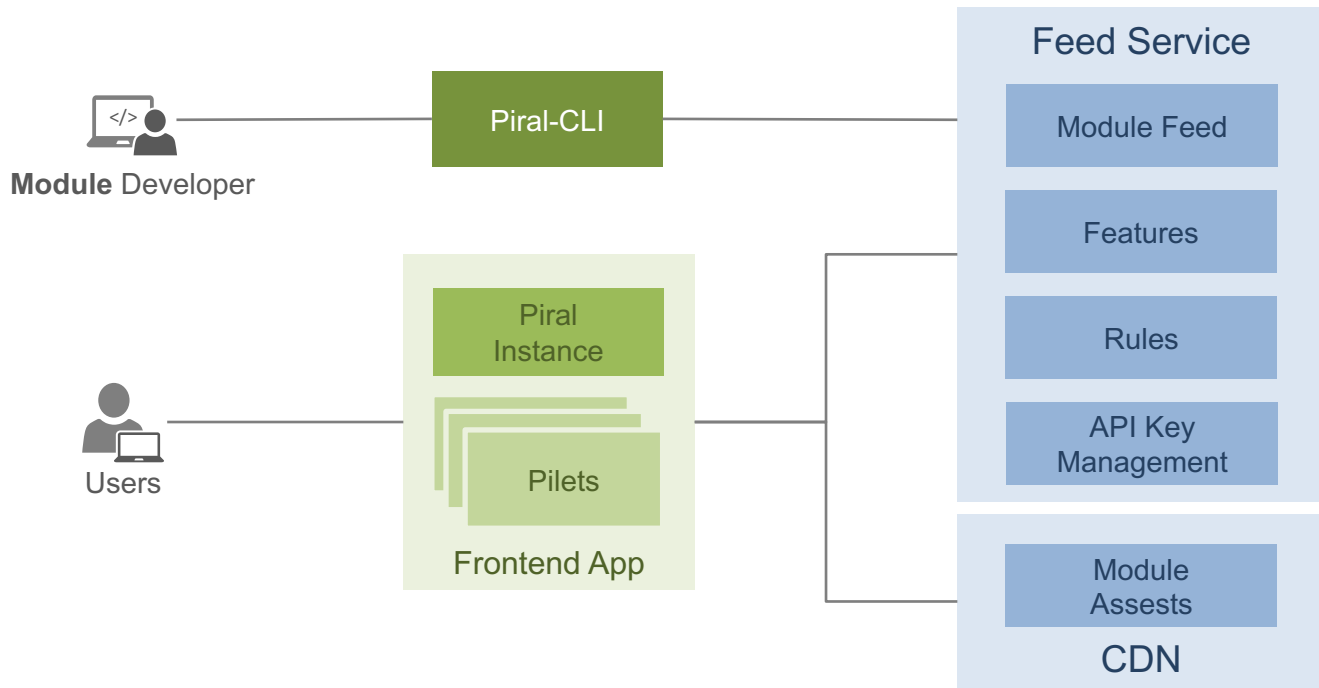
Architecture Pattern

Siteless UIs



GitHub repository
demos/demo-05

High-Level Architecture



Exercise (3)

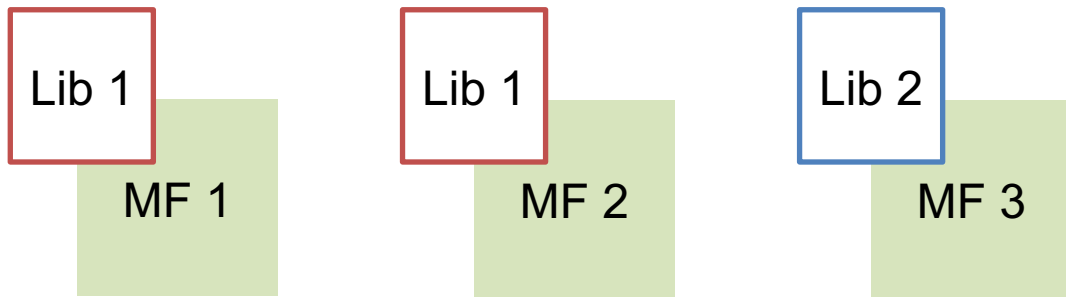
Write another MF for the webshop using Webpack Module Federation. The MF should show reviews related to the currently selected product.

Questions:

- How are the lifecycles handled?
- How is the integration done?

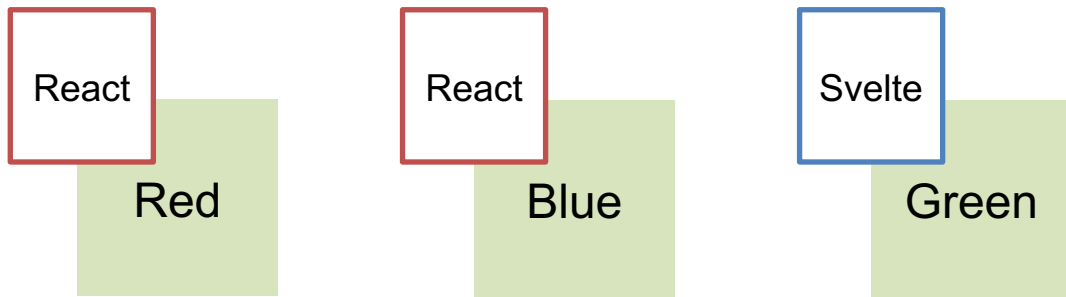
→ Use the „exercises/exercise-03“ directory as a starting point.

What About Dependencies?



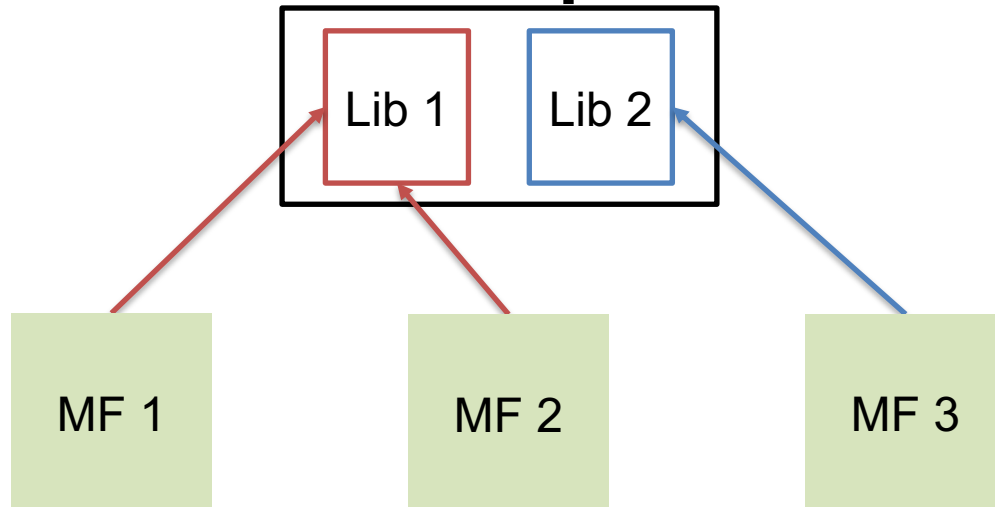
Share Nothing
(e.g., by bundling)

What About Dependencies?



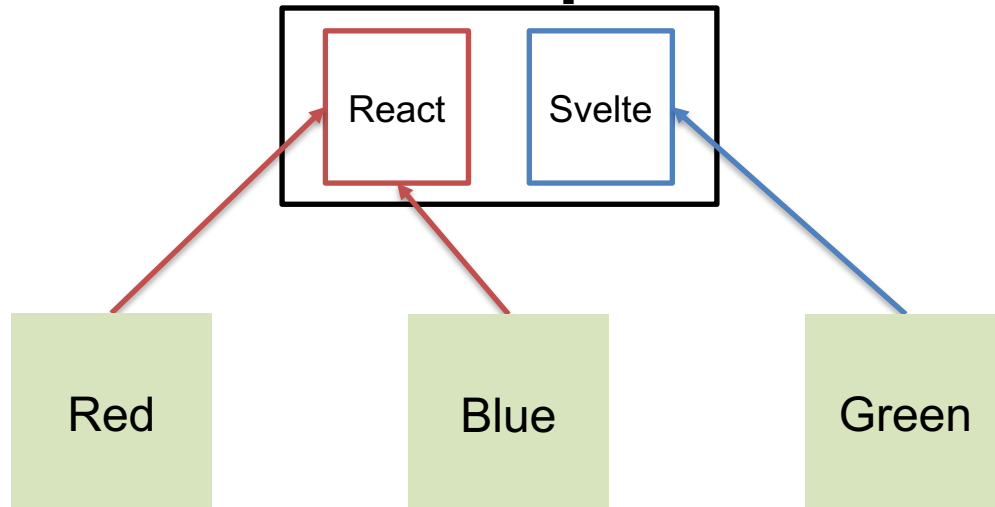
Share Nothing
(e.g., by bundling)

What About Dependencies?



Central
(e.g., via import maps)

What About Dependencies?



Central
(e.g., via import maps)

Importmaps

- Proposed standard for aliasing ESMs (e.g., using “react” instead of an URL)
- Supports full resolutions and partial resolutions (“path prefix”)
- Supports context-aware resolutions (“scopes”)
- Currently only (inlined) working in Chrome
- Alternative: use SystemJS, which supports importmaps
- More information: <https://github.com/WICG/import-maps>

Importmaps

```
<script type="importmap">
{
  "imports": {
    "lodash": "https://cdn.jsdelivr.net/npm/lodash-es@4.17.21/lodash.min.js",
    "lodash/": "https://cdn.jsdelivr.net/npm/lodash-es@4.17.21/"
  }
}
</script>
```

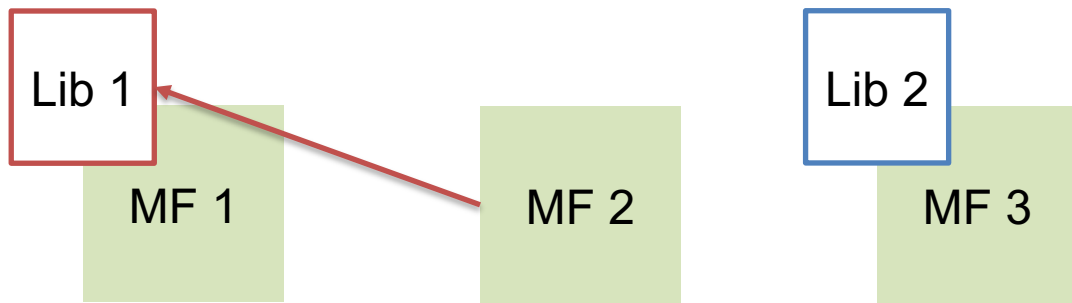
Dependencies

Central Sharing



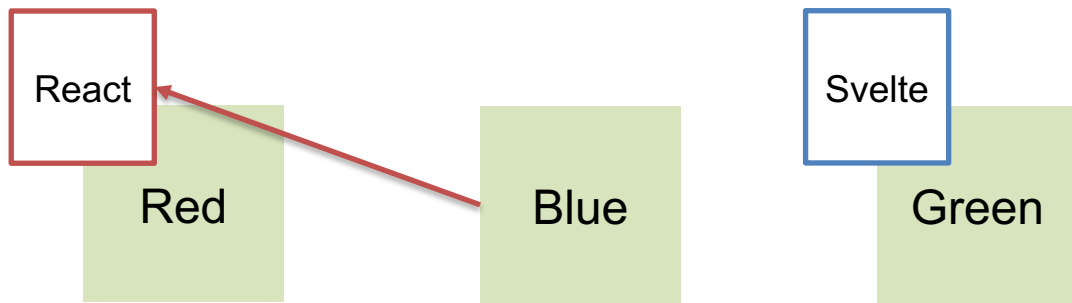
GitHub repository
demos/demo-06

What About Dependencies?

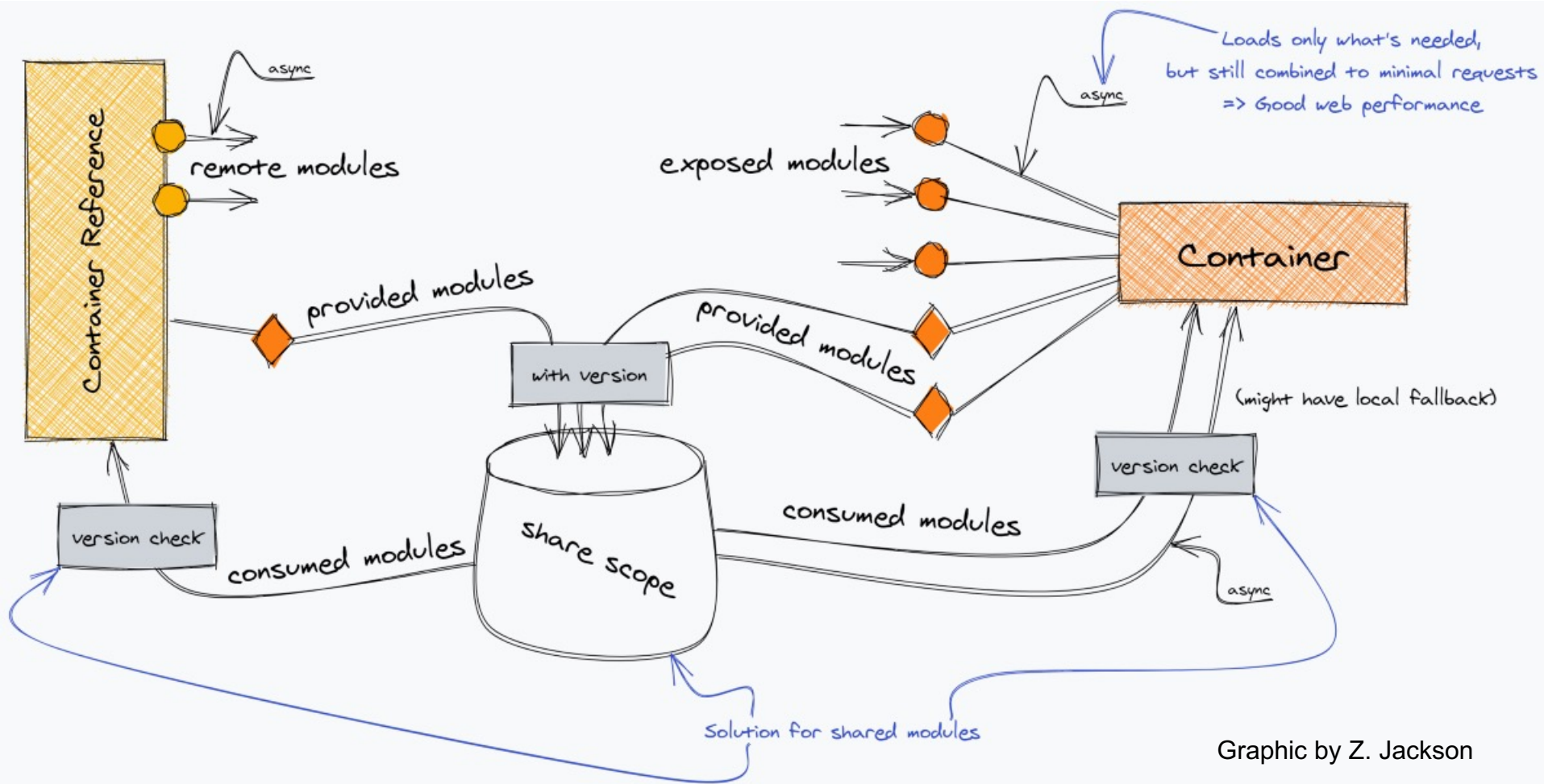


Distributed
(e.g., via Webpack Module Federation)

What About Dependencies?



Distributed
(e.g., via Webpack Module Federation)



Graphic by Z. Jackson

Webpack Module Federation

```
new ModuleFederationPlugin({
  shared: {
    lodash: "^4.17.0",
    react: {
      singleton: true,
      strictVersion: true, // errors out if version mismatch
    },
    "react-dom/client": {
      singleton: true,
      version: "18.0.1",
      requiredVersion: "^18.0.0",
    },
  },
})
```

Dependencies

Distributed Sharing



GitHub repository
demos/demo-07

Exercise (4)

Create a React web application to be bundled by Webpack. Share some dependencies like react via Module Federation.

Include the consent dialog MF from the *consent* directory in the application.

Questions:

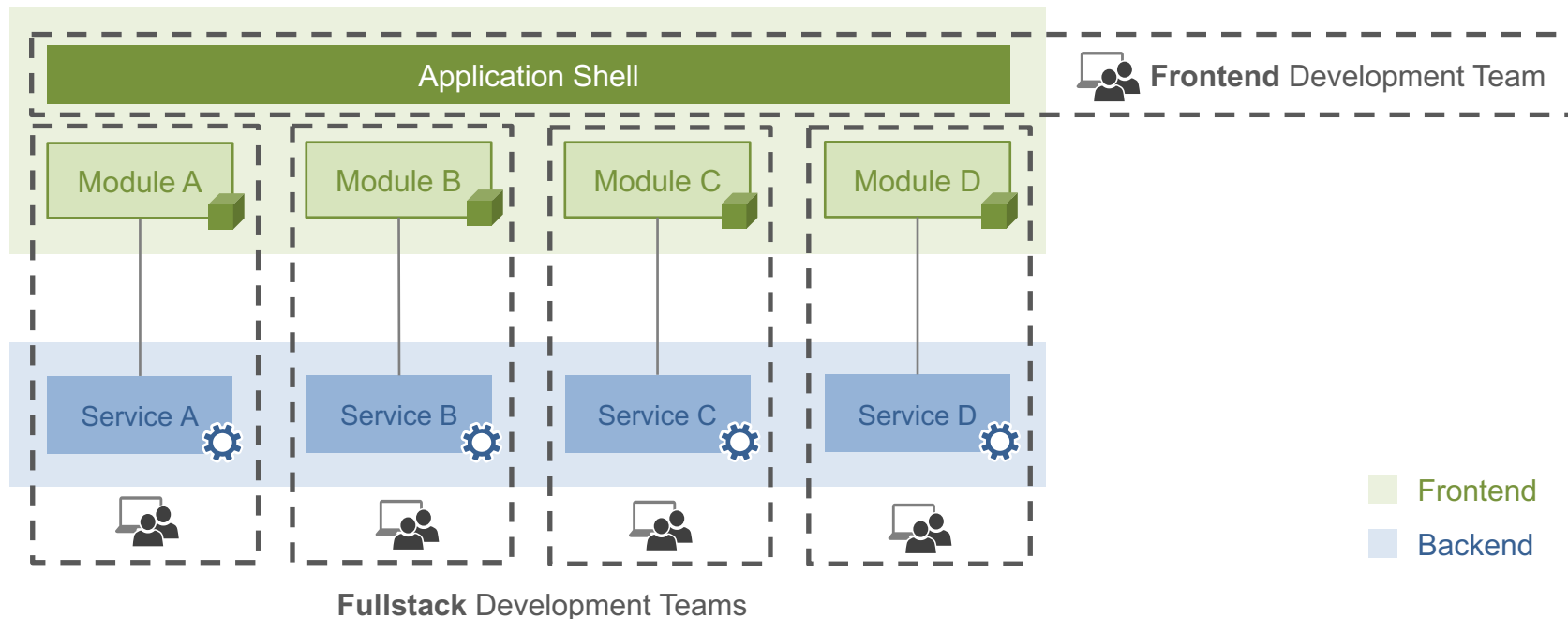
- What version of React is the consent MF using?
- What if the sharing of React is turned off?

→ Use the „**exercises/exercise-04**“ directory as a starting point.

Performance Considerations

- In general server-side composed MF are just cached and delivered as any other SSR page – higher complexity, but no/not much penalty
- Client-side composed MF seem more inefficient first, but actually can be faster than non-MF
 - Many scripts, but HTTP/2 bundles request, i.e., nearly same payload overall
 - Every script can be cached indefinitely, i.e., only smaller things require refresh
 - Only code can be delivered that will actually be useful – purpose-driven shipping
- Don't optimize too early, but think about the goal you have in mind

Development Approach

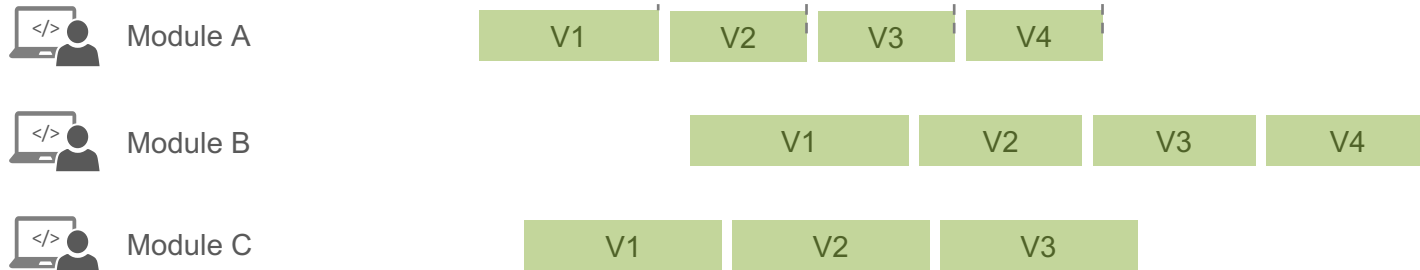


Development Lifecycles

App Shell Development



Module Development



Micro Frontend Checklist

- ✓ Team too large or more teams should be onboarded in the future
- ✓ Team too slow to respond to feature requests from backend teams
- ✓ Bundle size too large / build too slow
- ✓ Some parts of the code are touched way more often than others
- ✓ The code is filled with (inconsistent) feature flags
- ✓ The need for alignment with the backend w.r.t. authorization is high
- ✓ Existing framework need to be replaced (progressively) soon
- ✓ New features may be implemented in other frameworks easier

Feed Service

- Micro Frontend discovery is necessary for having loose coupling and true scalability
- A feed service is not only at the heart of Siteless UIs, but can actually be also added to any other pattern
- Usual responsibilities include publish, rollback, asset storage, ...

smapiot

smapiot

smapiot.com

github.com/smapiot



Florian Rappl

florian.rappl@smapiot.com

@FlorianRappl



Piral

piral.io

docs.piral.io

Thank you!

