From Microservices to Micro frontends: The monolith breaking story continues

Virtual Technical Experts Plenary Session – Yves MATHIEU – 2018-11-22

sopra banking SOFTWARE

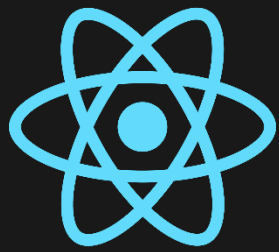
01.

The past is good to learn from but not to live in

Problems with existing approaches

We start developping apps and …

- **Build functionalities over FWK**
- **Deviate from Market Standards**
- **Add abstraction layers**
- **Onboarding**
- **People are learning the company Standards**
- **With the time FWK die or whatever**



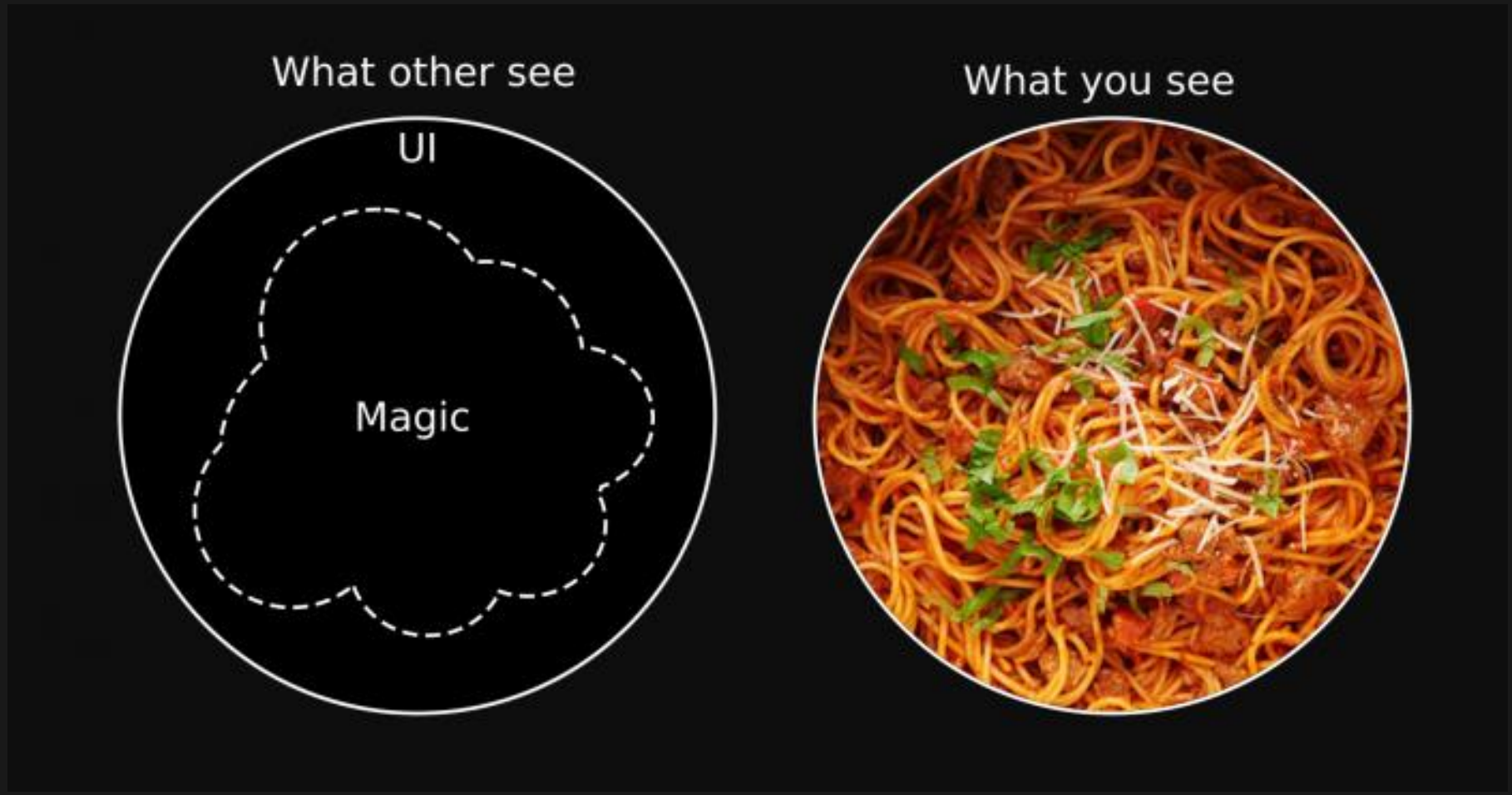
Problems with existing approaches

We create contracts

Problems with existing approaches
Team size increase

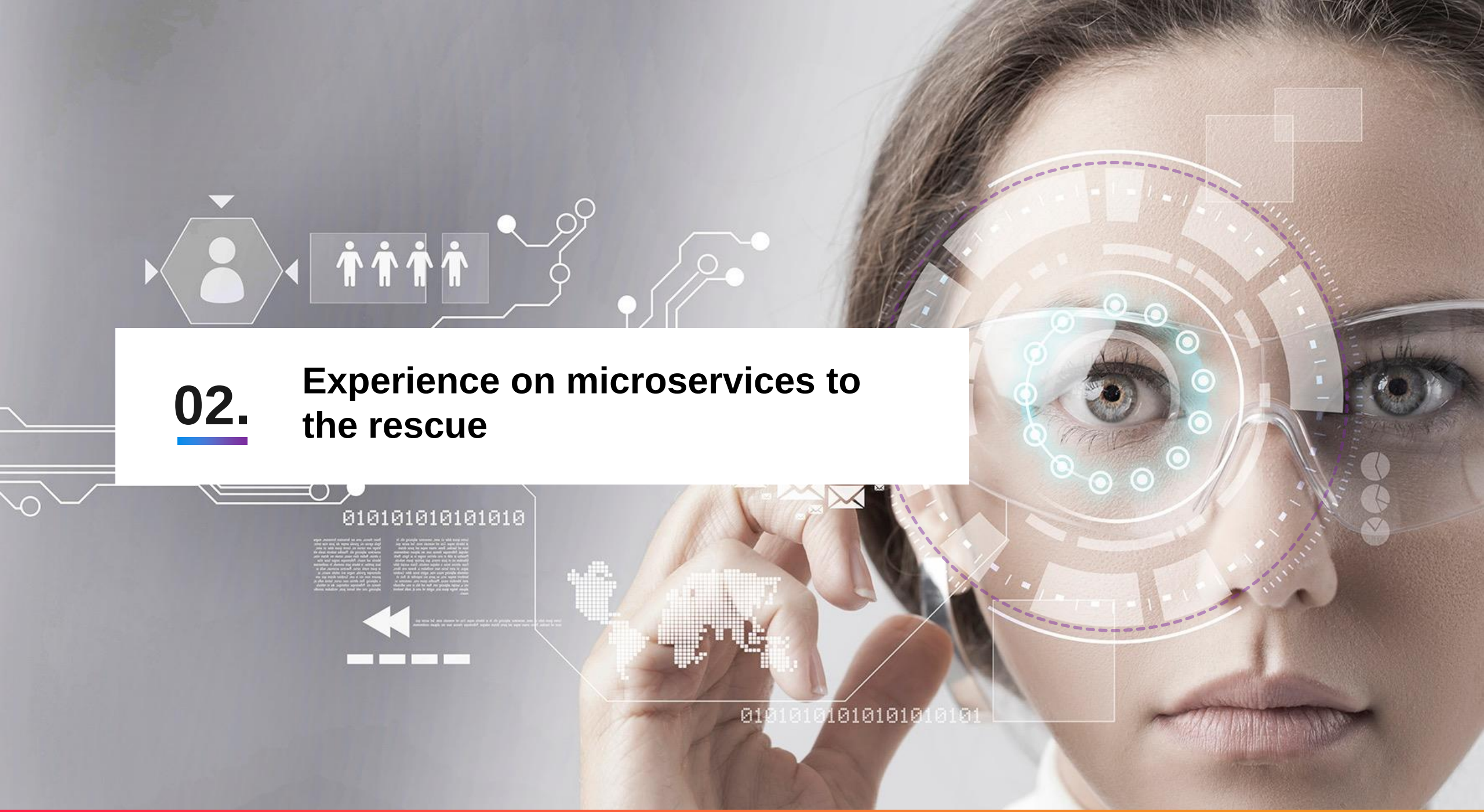
- More energy to coordinate
- Longer processes
- Side effect if everything is not automated

Problems with existing approaches
With the time …



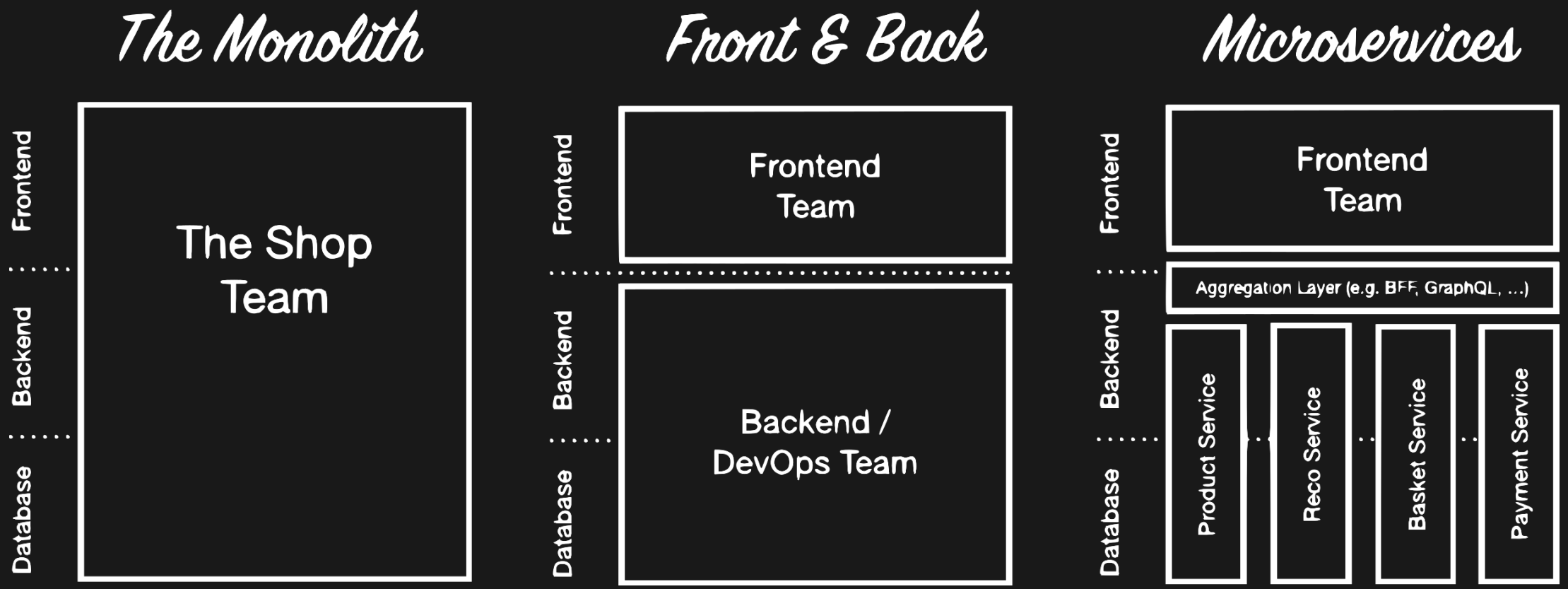
Problems with existing approaches
Migration / changes

Killing Innovation

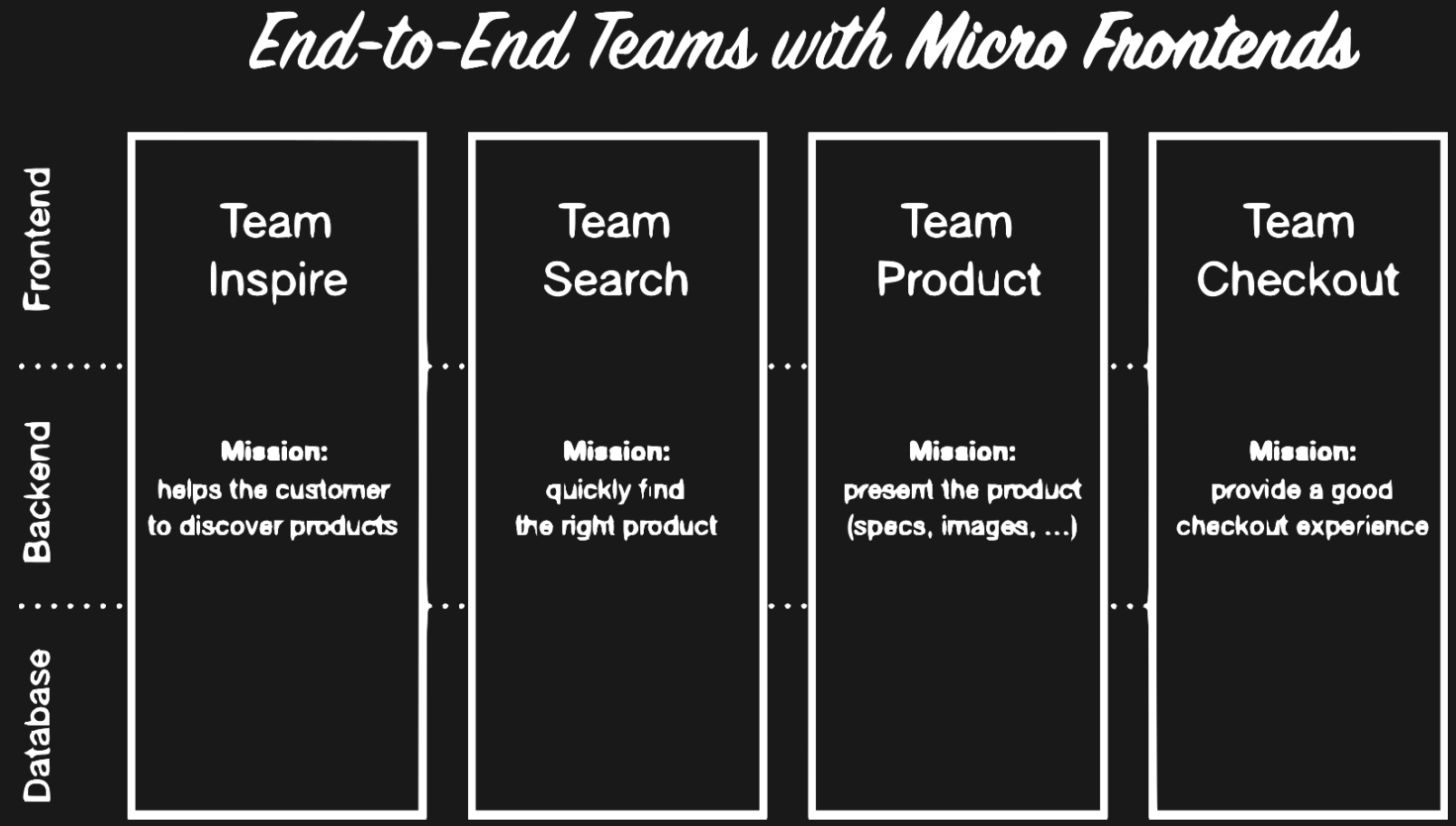


02. Experience on microservices to the rescue

Microservice to the rescue
What did we do?



Microservice to the rescue
Apply it to the frontend



Micro frontend
Advantage of the approach in the organization

- **Each team:**
  - Composed of interdisciplinary people
  - Assigned with a specific mission
  - Store its own data
  - Ships its own UI
  - Decide of its own tech stack *

Micro-frontend
Why?

- **Small autonomous teams ...**
  with a clear mission
- **Customer Focus ...**
  Every team ships directly - no pure API team (power tool)
- **Reduce scope**
  Come back on the point and can focus on the (Scope)
- **Frontend renovation**
  Without throwing everything away

Micro-frontend

Already widely used in famous companies

Ikea

Zalando

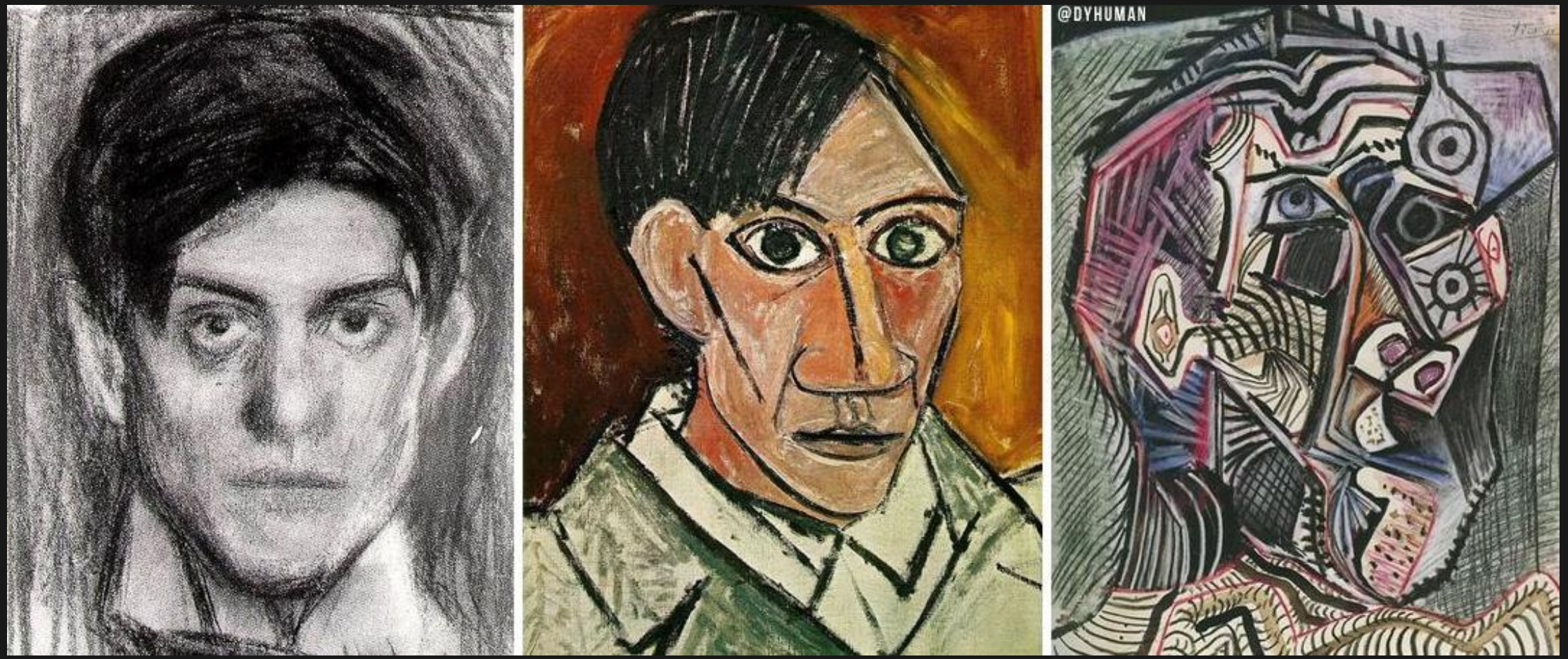
Facebook

Spotify

Amazon

We need Micro frontend

For UX we need a minimum of coherence between UIs



Picasso's self-portrait over the ages

03. Concept 1: Web Components

Web Components
What is it?

- **W3C**
- **4 main features which can be used separately or all together:**
  - Custom Elements – APIs to define new HTML elements

```
<wks-header></wks-header>
```

  - Shadow DOM – Encapsulated DOM and styling, with composition
  - (HTML Imports) – Declarative methods of importing HTML documents into other documents
  - HTML Templates, an HTML fragment is not rendered, but stored until it is instantiated via JavaScript

Web Components

Is it supported by every browser?

Use web components today and have them work in all major browsers.

CHROME OPERA FIREFOX SAFARI 7+ EDGE / IE11+

Micro-frontend

How – Web Components

Browser support	CHROME	OPERA	SAFARI	FIREFOX	EDGE
HTML TEMPLATES	STABLE	STABLE	STABLE	STABLE	STABLE
CUSTOM ELEMENTS	STABLE	STABLE	STABLE	STABLE	POLYFILL CONSIDERING
SHADOW DOM	STABLE	STABLE	STABLE	STABLE	POLYFILL CONSIDERING
ES MODULES	STABLE	STABLE	STABLE	STABLE	STABLE

Micro-frontend

How – Web Components

Intermediary frameworks providing Web component features / facilities:

Angular **Elements**

STENCIL

Others….

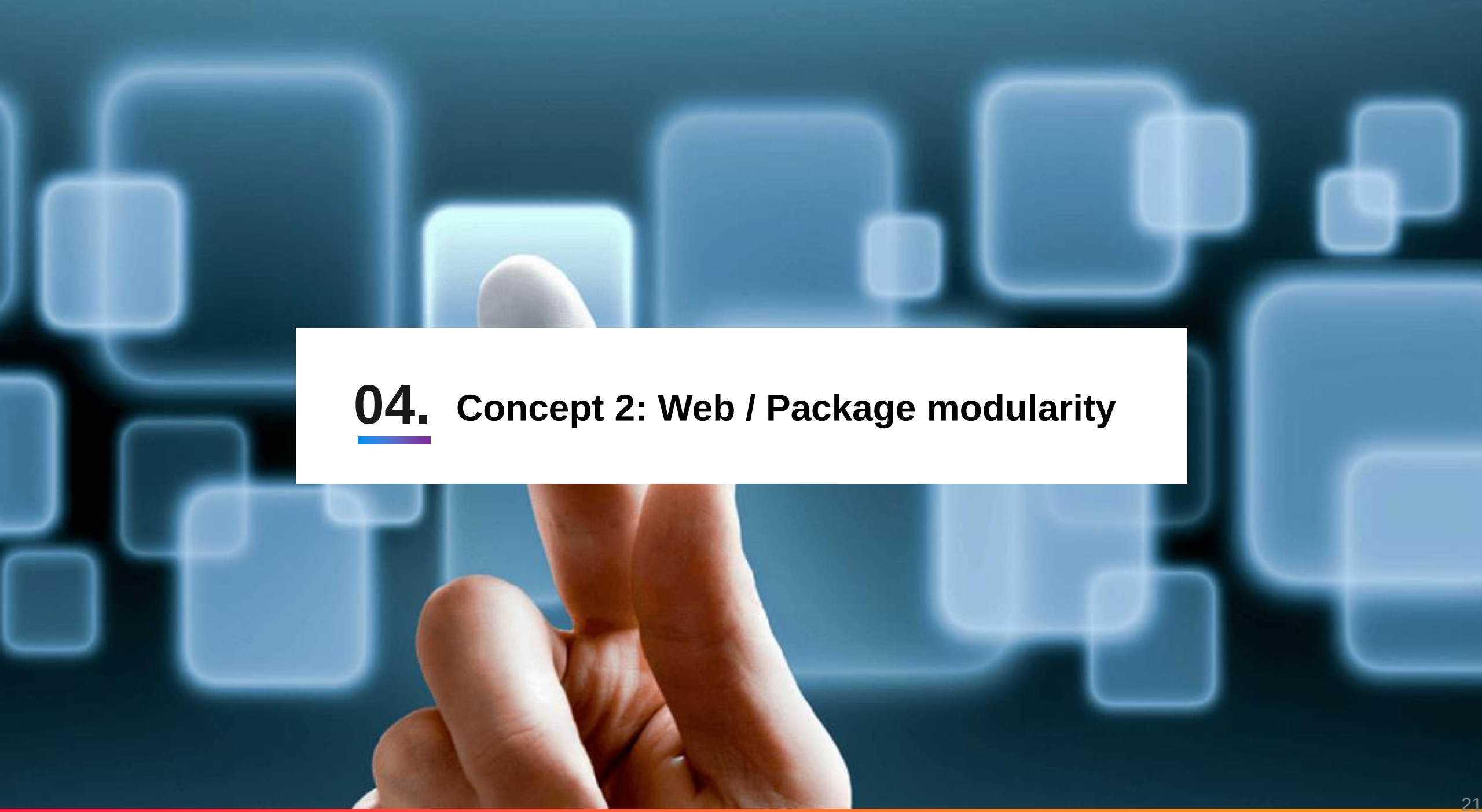
Micro-frontend

How – Communication

The DOM is the API

Reduce framework use and only define contracts

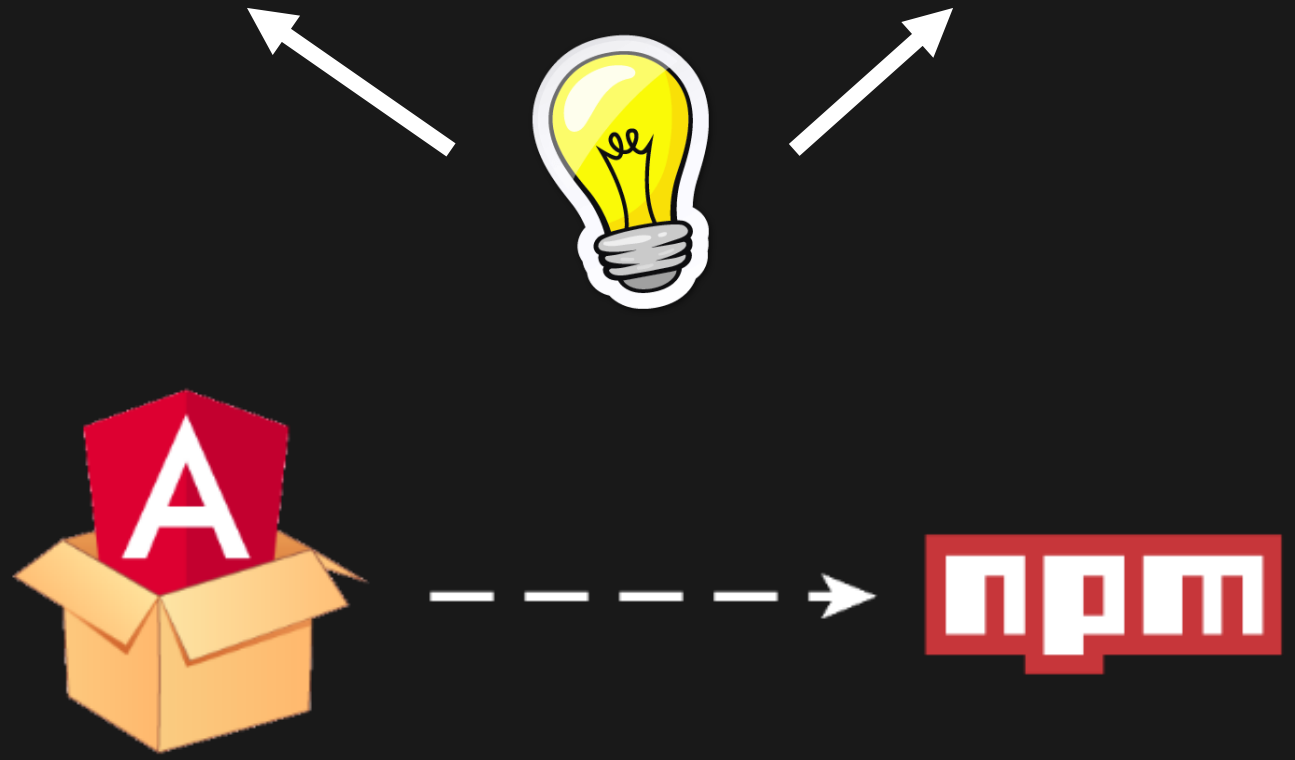
Element names – attributes and event



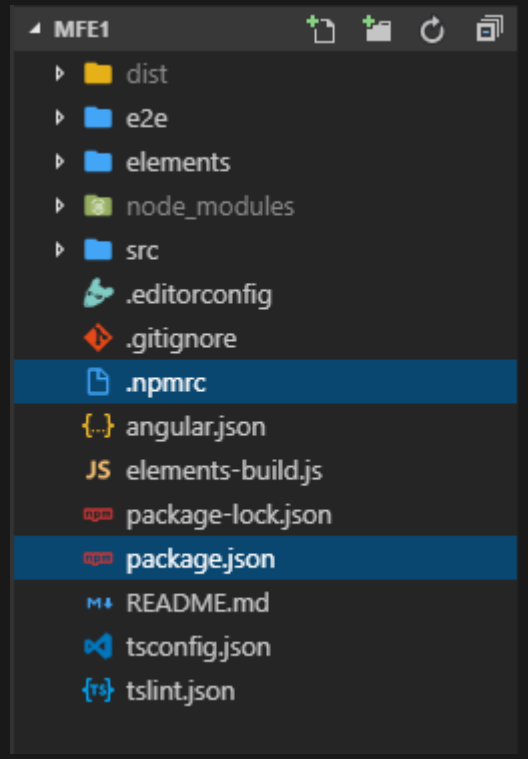
04. Concept 2: Web / Package modularity

Web / Package modularity
What do you mean?

Create packages to isolate modules from each others



Web / Package modularity
Node module



- **Create the npm project -> Link**
- **Share it by publishing**

```
ymathieu@ITEM-S64580 MINGW64 ~/git/mfe1 (master)
$ npm publish --registry http://pdtinteg.ptx.fr.sopra/artifactory/api/npm/npm.workstation/
```

- Use .npmrc file

Web / Package modularity
Advantages / Disavantages

Advantages

- ✔ Distribution
- ✔ Versioning

Disadvantages

- ✘ Distribution
- ✘ Versioning

Web / Package modularity
Mono-repo Vs Many-repo

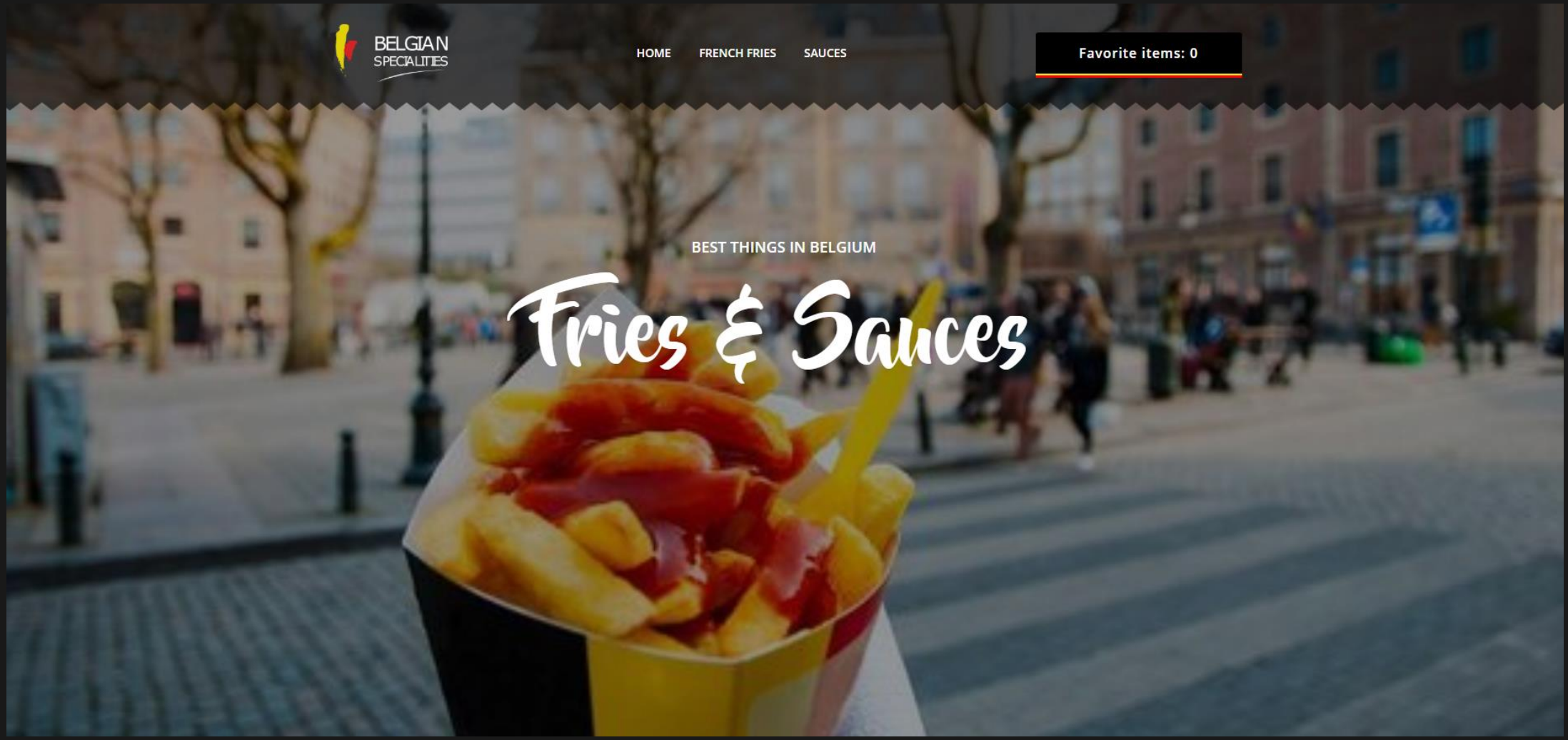
When managing different projects that are loosely coupled, you can get the advantages of the Mono-repo approach.

Used in many companies like …

- **Complete discussion on the topic:**
  - Monorepos in the Wild
  - Mono-repo or multi-repo? Why choose one, when you can have both?

05. Micro frontends: Different approaches

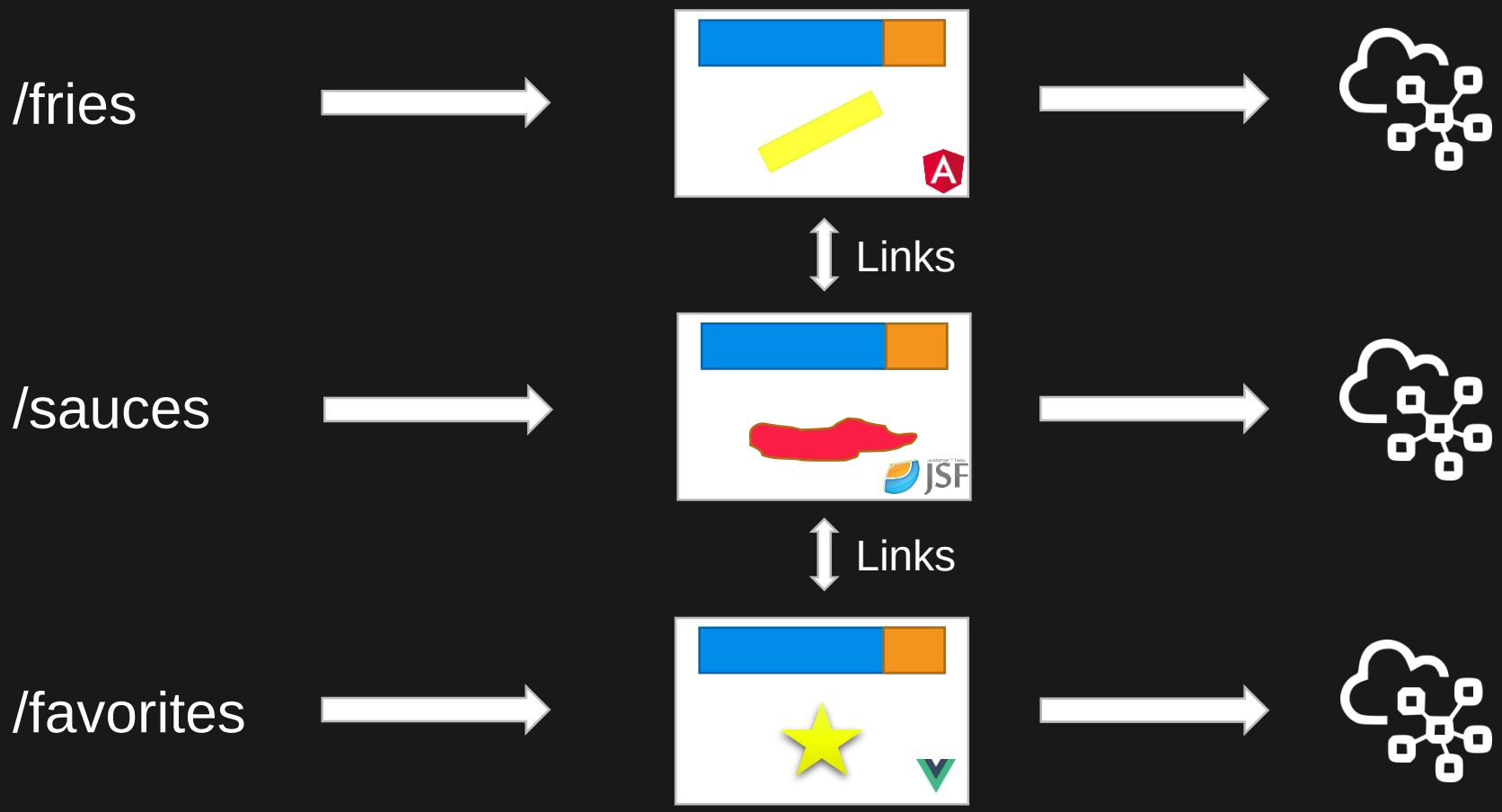
Micro frontends
Situation



Micro frontends
Check points

- Independently developed
- Independently ran
- Independently manage their tech stack
- Fast loading / Optimized
- Native support of the Browser
- Sharing Basics (Header / Footer)
- Sharing identity / UI Coherence
- Smooth UX

Micro frontends
1: Split web sites

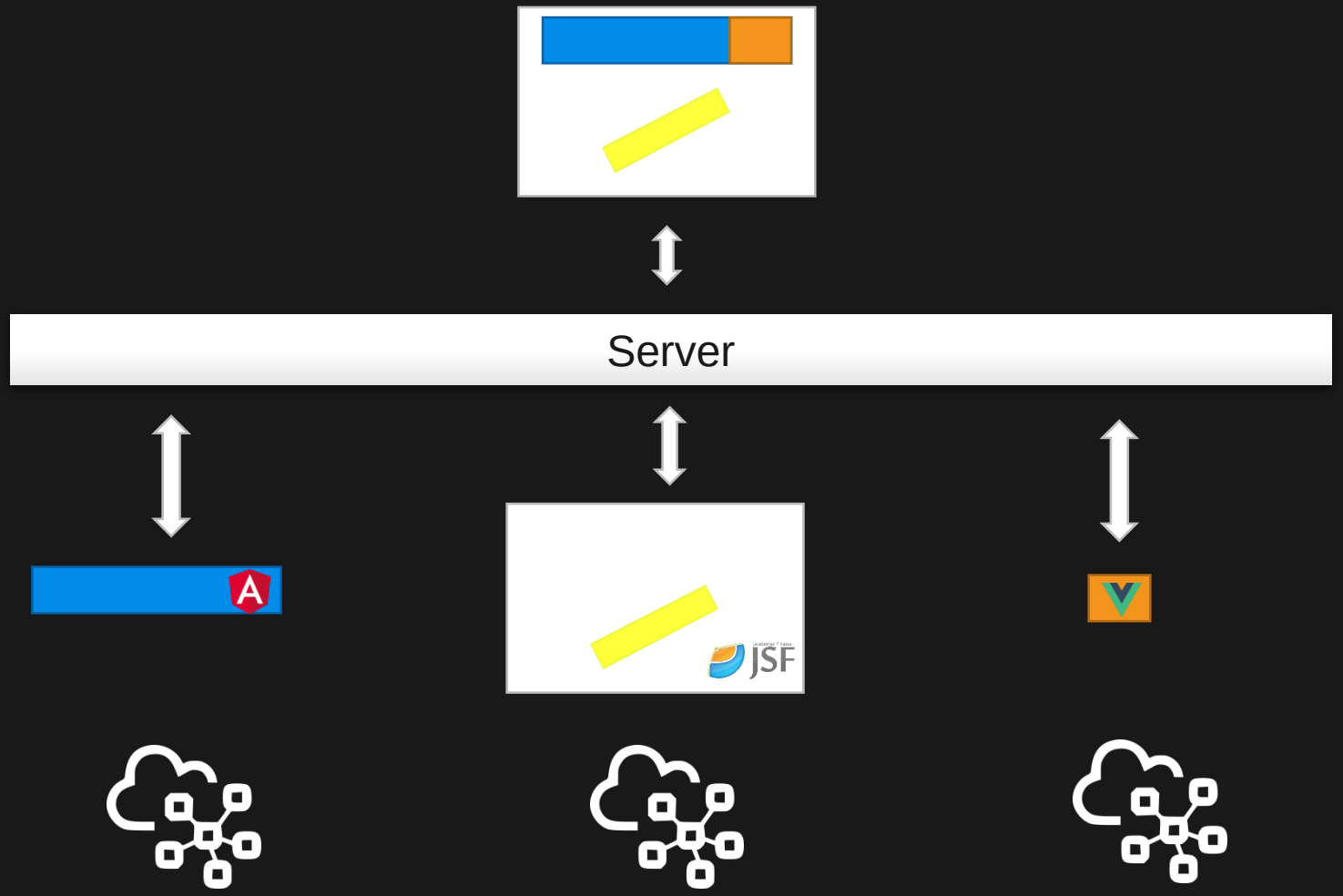


Micro frontends
1: Split web sites

- ✔ Independently developed
- ✔ Independently ran
- ✔ Independently manage their tech stack
- ✔ Fast loading / Optimized
- ✔ Native support of the Browser
- ✘ Sharing Basics (Header / Footer)
- ✘ Sharing identity / UI Coherence
- ✘ Smooth UX

Micro frontends

Approach 2: Server Side include (Something like WKS 1.6)



Micro frontends
Approach 2: Server Side include (Something like WKS 1.6)

- **Server side include protocol**

```
<html>
  <body>
    <!-- #include virtual="/header" -->

    <!-- #include virtual="/fries" -->

    <!-- #include virtual="/faviorites" -->
  </body>
</html>
```

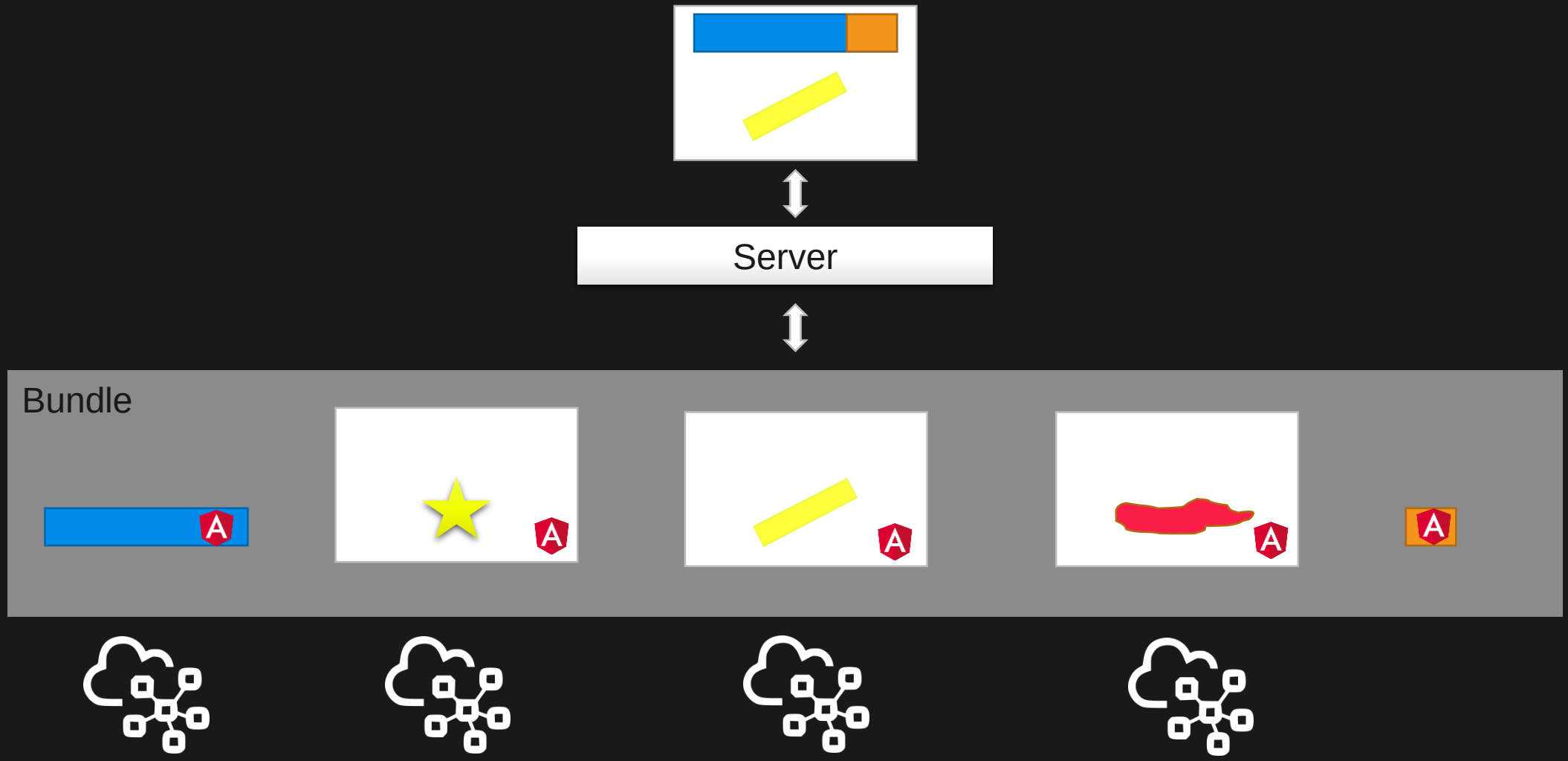
- **Portal approach**

Micro frontends

Approach 2: Server Side include (Something like WKS 1.6)

- ✔ Independently developed
- ✔ Independently ran
- ✔ Independently manage their tech stack
- ✘ Fast loading / Optimized
- ✘ Native support of the Browser
- ✔ Sharing Basics (Header / Footer)
- ✔ Sharing identity / UI Coherence
- ✘ Smooth UX

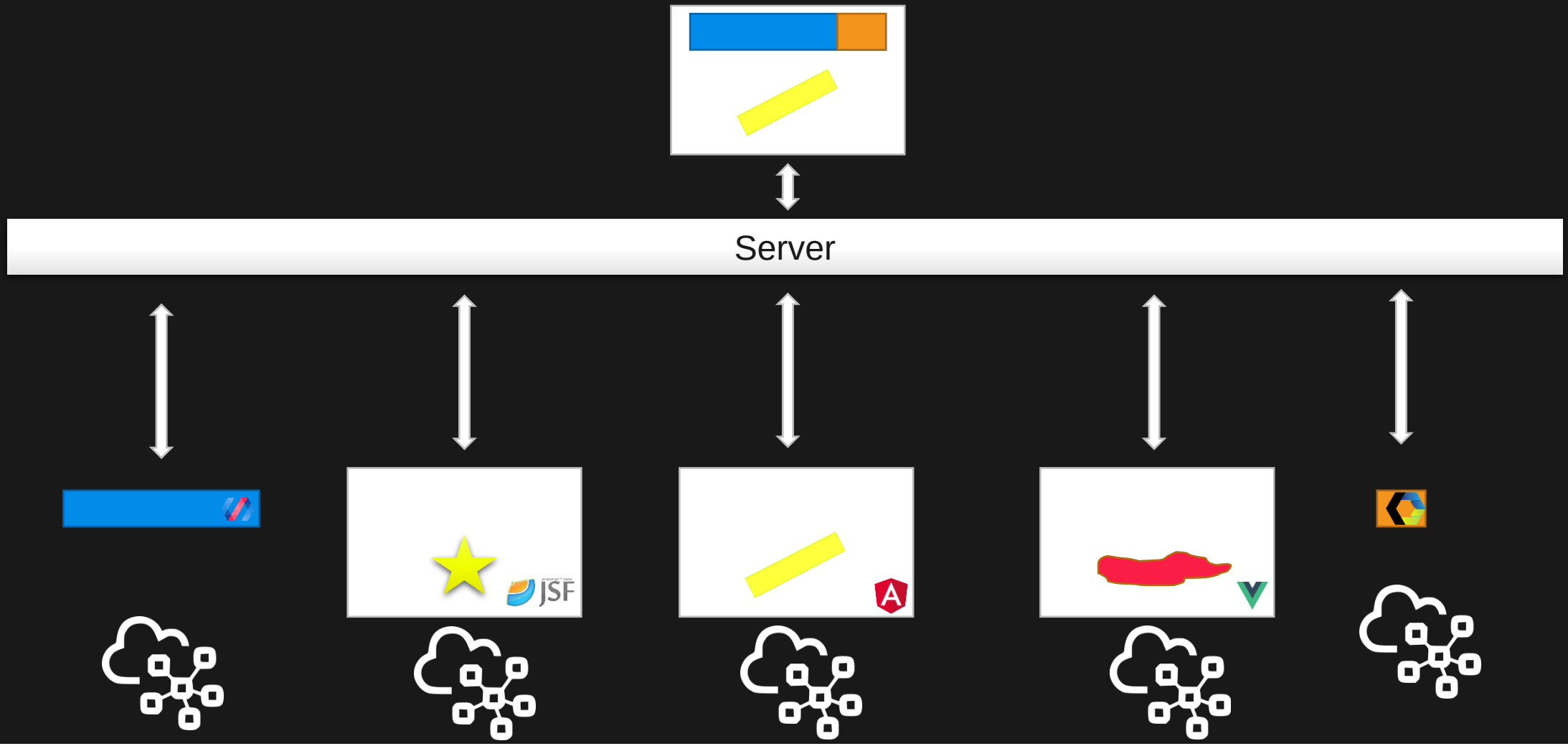
Micro frontends
Approach 3: Bundle



Micro frontends
Approach 3: Bundle

- ✔ Independently developed
- ✘ Independently ran
- ✘ Independently manage their tech stack
- ✘ Fast loading / Optimized
- ✔ Native support of the Browser
- ✔ Sharing Basics (Header / Footer)
- ✔ Sharing identity / UI Coherence
- ✔ Smooth UX

Micro frontends
Approach 4: App Shell



Micro frontends
Approach 4: App Shell

Several options:

- **Iframe inclusion …**

Micro frontends
Approach 4: App Shell

- Iframe inclusion doesn’t mean iframe inception

Micro frontends
Approach 4: App Shell

- **Iframe inclusion benefits from new browser security features**

Micro frontends
Approach 4: App Shell

Several options:

- **Iframe inclusion …**
- **Web Components packaging**

Micro frontends
Approach 4: App Shell

- ✓ Independently developed
- ✓ Independently ran
- ✓ Independently manage their tech stack
- ✓ Fast loading / Optimized
- ✓ ✗ Native support of the Browser
- ✓ Sharing Basics (Header / Footer)
- ✗ Sharing identity / UI Coherence
- ✓ Smooth UX

Micro frontends
Approaches

Every approach comes with advantages and disadvantages.

It’s up to you and based on your **constraints** and **needs** to set the focus on one approach.

You can even **mix them**.

04.

Let’s keep our feet on the ground

Micro frontends
Positive

- Fast onboarding
- Independence
- Easy testing
- Resilience
- **Futur proof**

Micro frontends
REX

- There is no silver bullet
  - To every need we have to identify the most appropriate approach
  - Some approaches could be mixed (Shell)
- Start small
- Same repo
- Naming convention
- Design system

Micro frontends
REX

- Be rational and try to base all your frontends on the same technology stack.

Micro frontends
References

- https://micro-frontends.org/
- Michael Geers
- Elisabeth Engel
- Manfred Steyer
- Me ☺ (13 years of experience in a R&D)