

Micro Frontends the nitty gritty details

OR

frontend, backend, 🌈 happyend





Michael Geers

Frontend Engineer

naltatis

on Twitter & GitHub

micro-frontends.org

e-commerce



neuland:::
Büro für Informatik

Bremen, Germany

Frontend Monolith

Black Diamond
Royal Library, Copenhagen

**faster feature
development**



Micro Frontends

1. what?

2. why?

3. how?

4. nitty gritty details 

but before ...

Where do we come from?

Frontend

Backend

The Monolith

era of the
full-stack developer



Frontend

Frontend

bye bye
full stack 🙄



Backend

Backend

Microservice Architecture

Frontend

Backend

specialists

Frontend

API Gateway (Rest, BFF, GraphQL)

Product
Service

Availability
Service

Basket
Service

Teaser
Service

Price
Service

Payment
Service

Customer
Service



Frontend Monoliths

growing complexity
no scaling

Frontend

Backend

Frontend

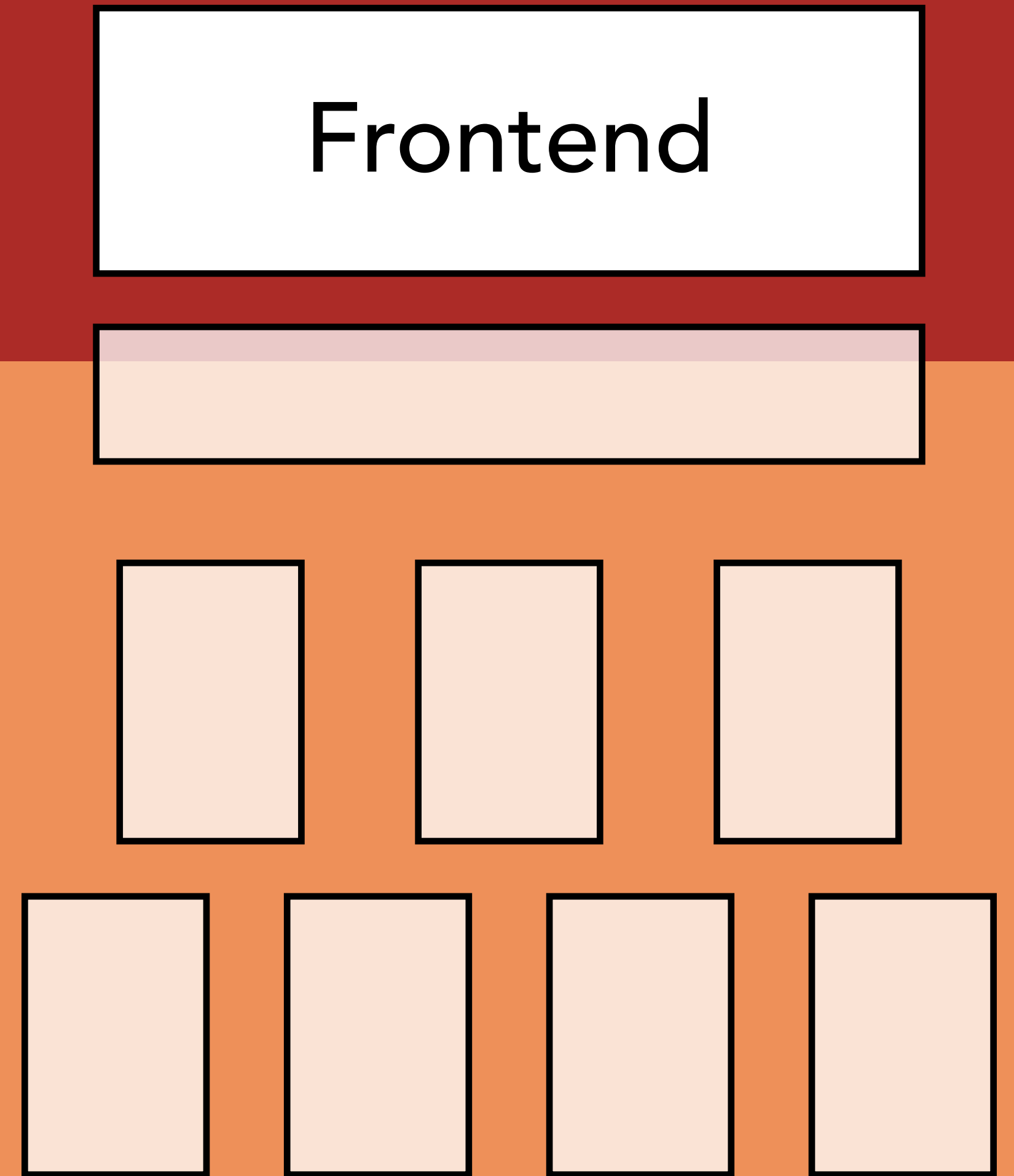
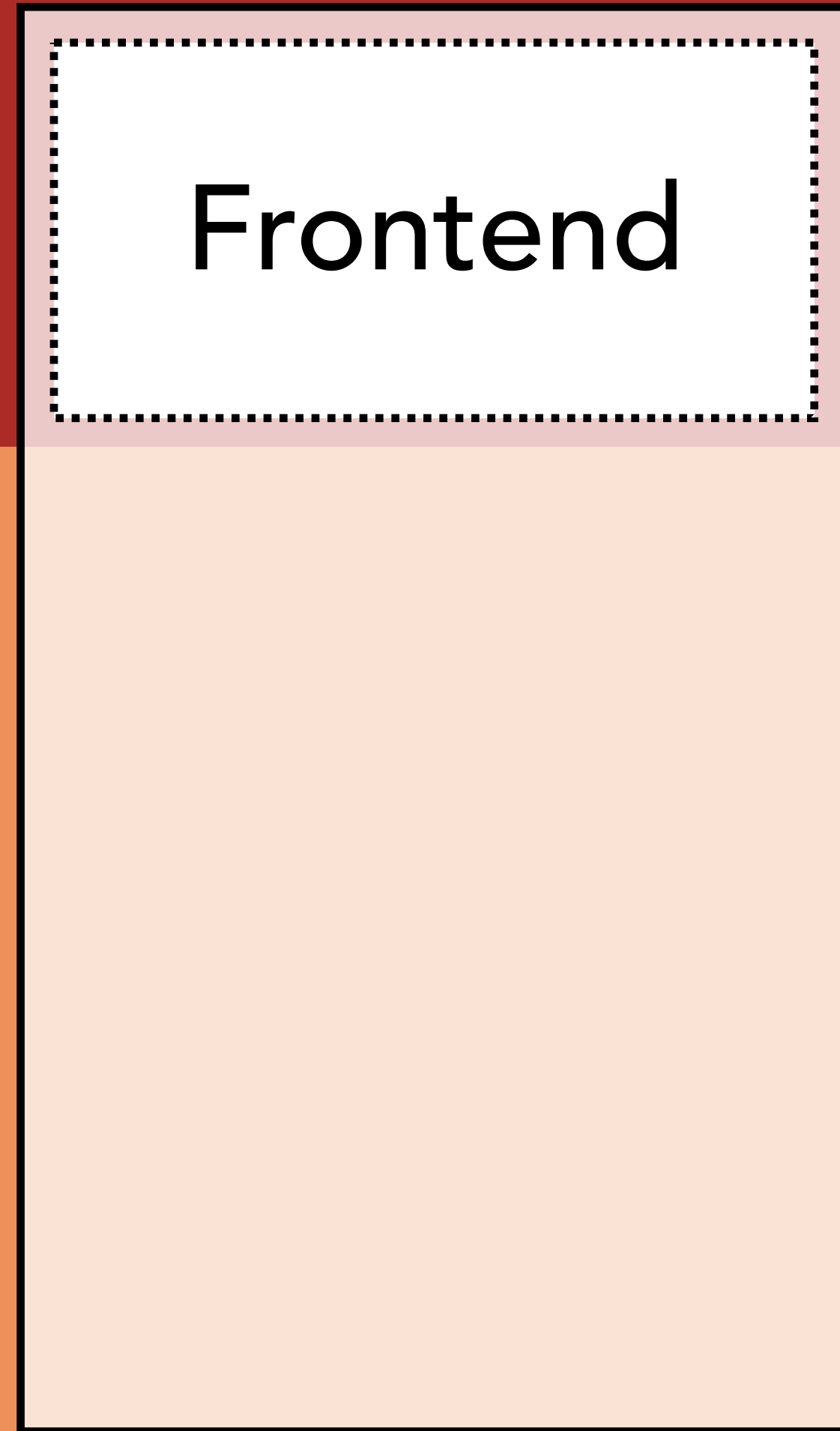
Frontend

Frontend

The Monolith

Front- & Backend

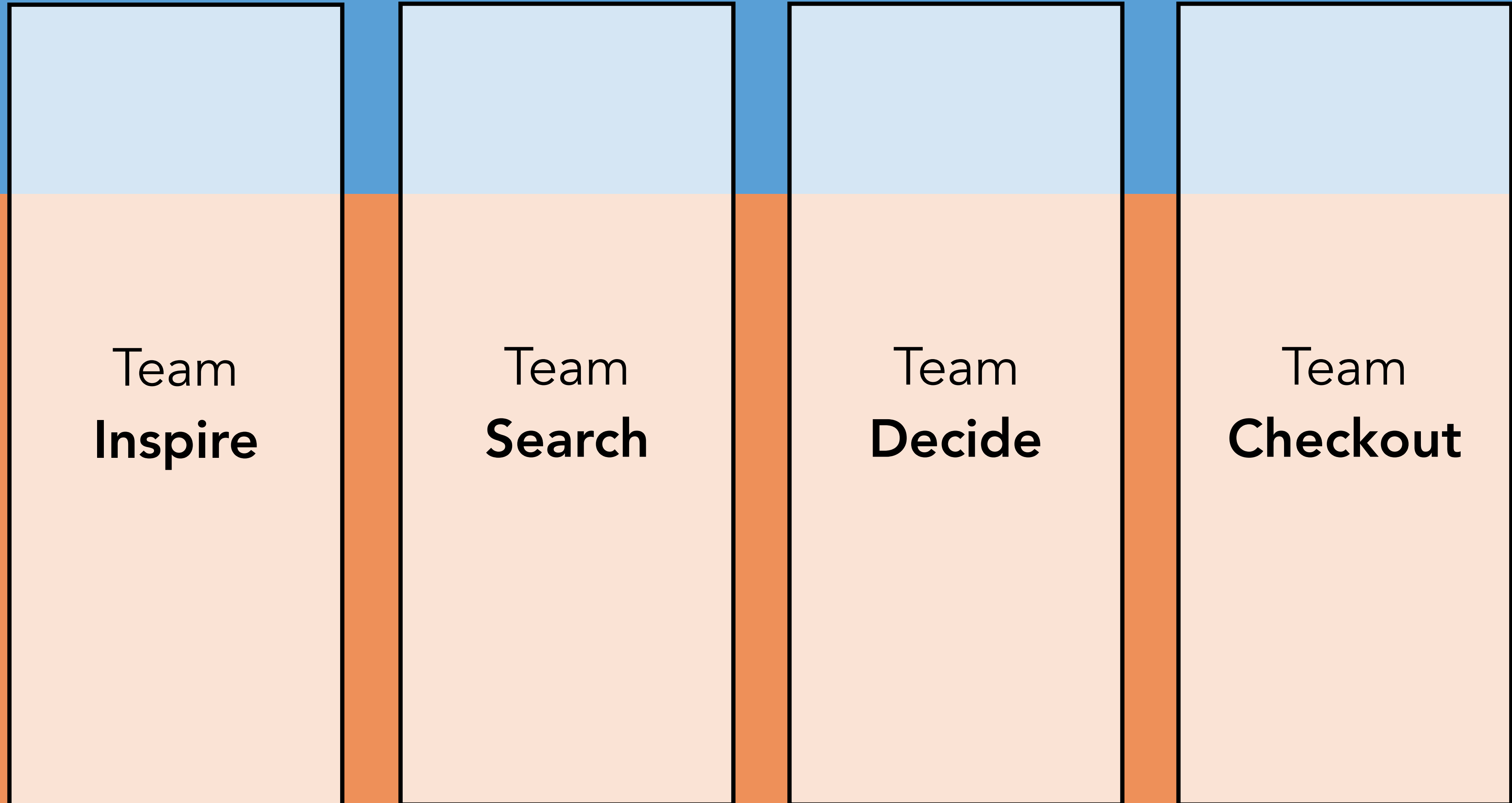
Microservices



Micro Frontends

Frontend

Backend



1.

What are Micro Frontends?

Micro Frontends Definition

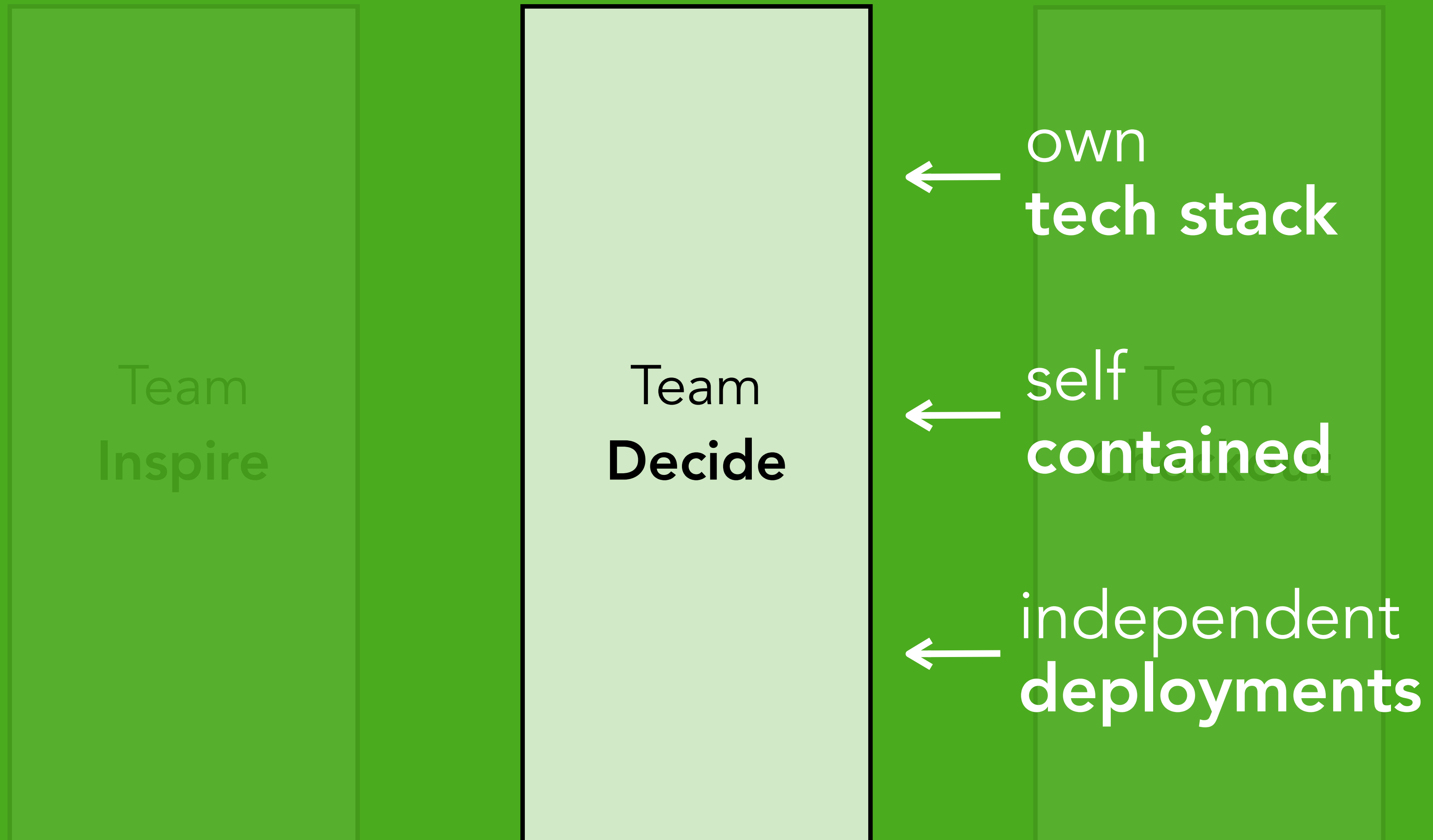
... Our preferred (and proven) approach is to split the browser-based code into micro frontends. In this approach, the web application is broken down into its features, and each feature is owned, frontend to backend, by a different team. This ensures that every feature is developed, tested and deployed independently from other features. ...

ThoughtWorks Technology Radar

<https://www.thoughtworks.com/de/radar/techniques/micro-frontends>

Stages: Access (2016) > Trial (2017) > Adopt ★ (2019)

Autonomy



Team Missions

„Make discovering and finding products easy.“

„Support the customer's buying decision.“

„Make the actual buying process happen.“



Team
Inspire

Team
Decide

Team
Checkout

technical
& organizational

Implications

2.

Why Micro Frontends?

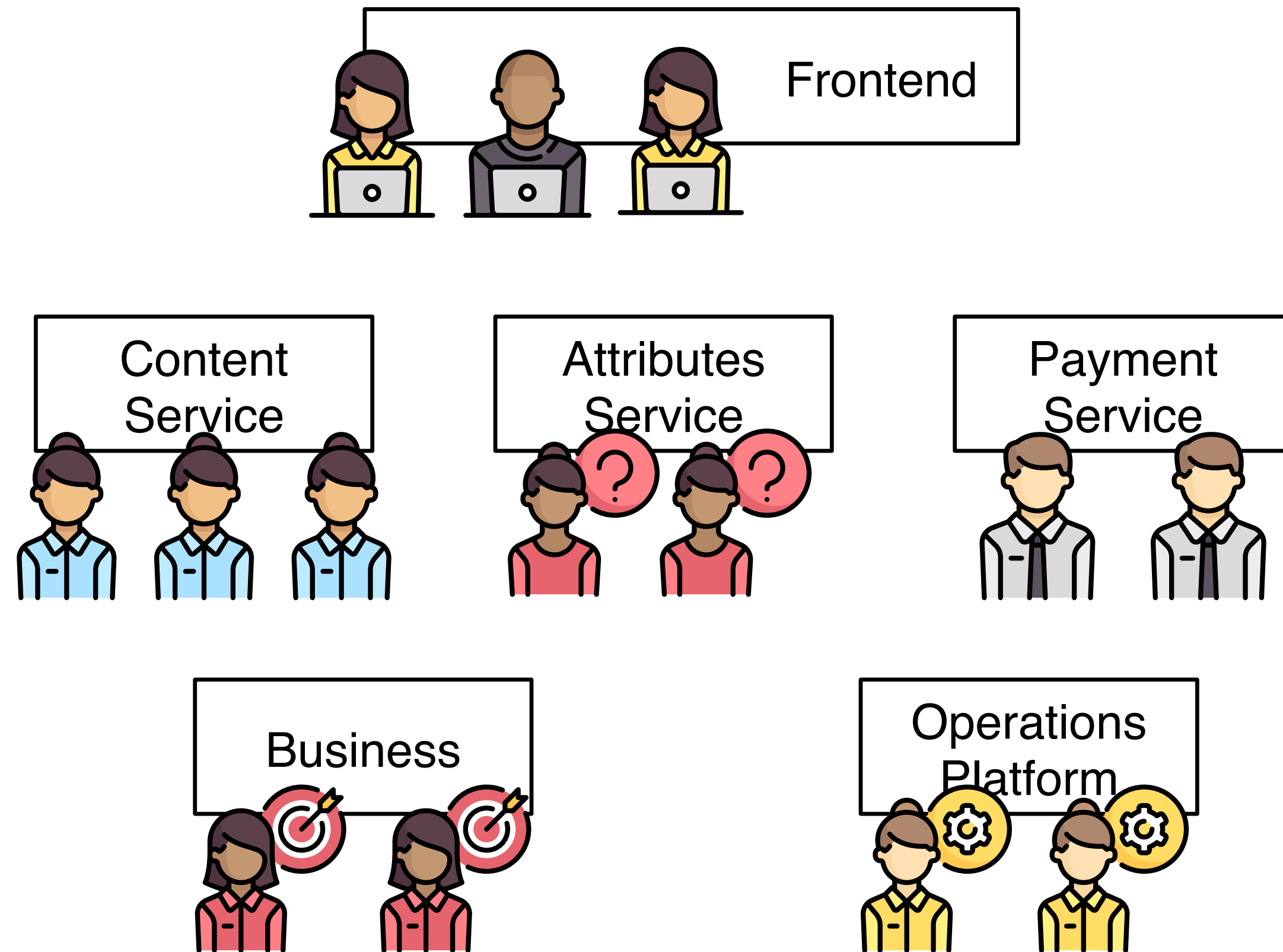


specialists team



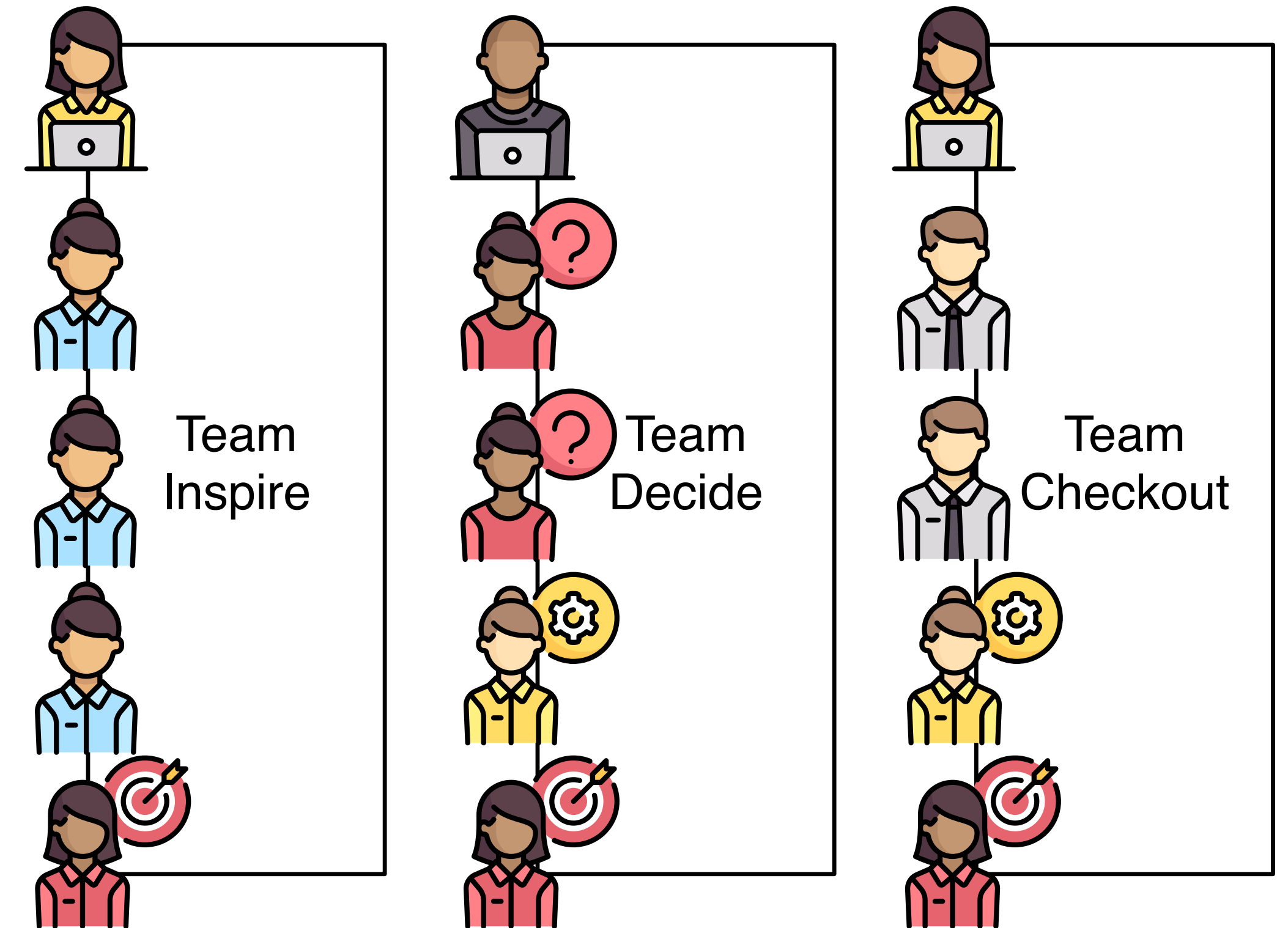
cross functional team

Specialist Teams



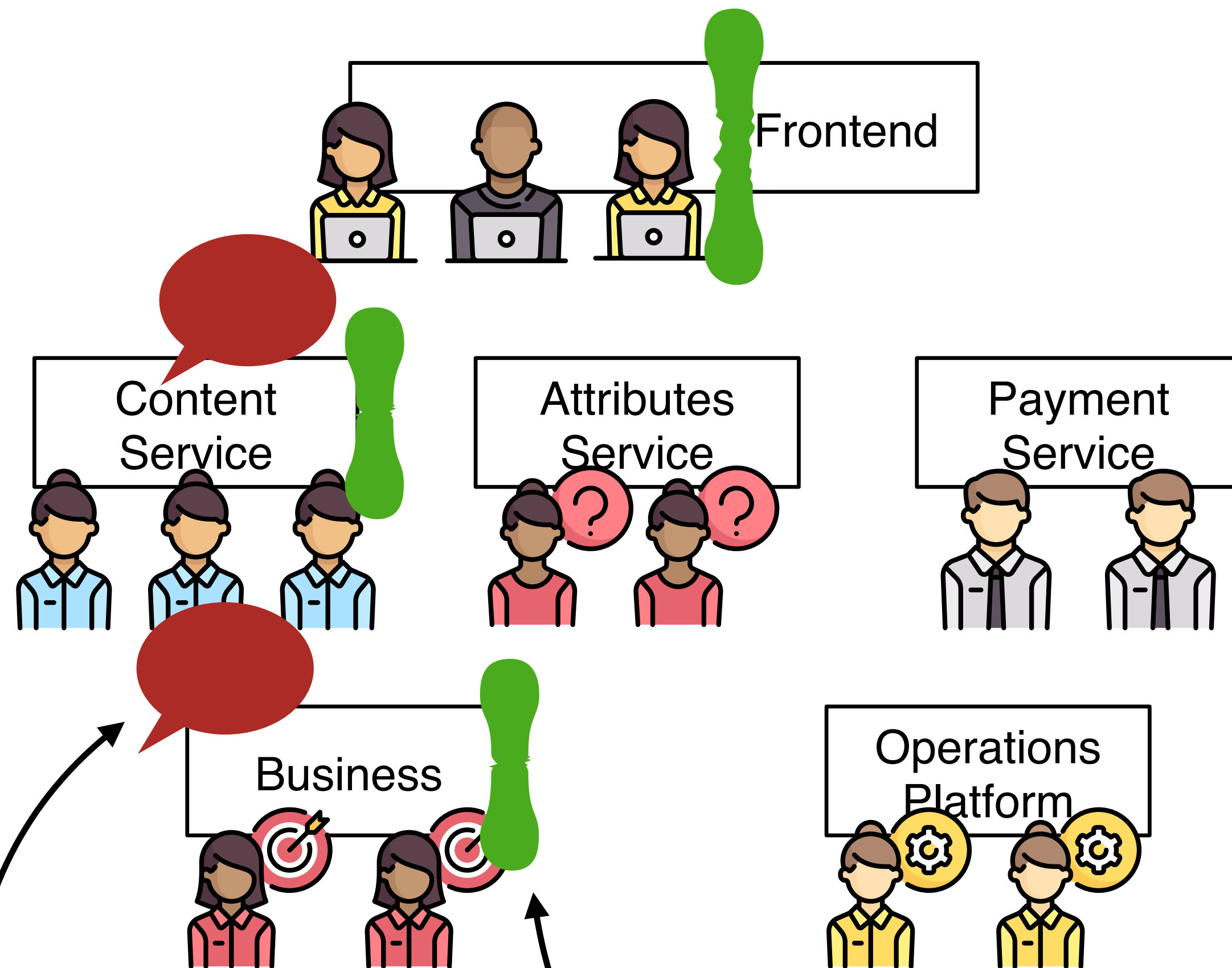
grouped around a **skill** or **technology**

Cross Functional Teams



grouped around a **use case** or **customer need**

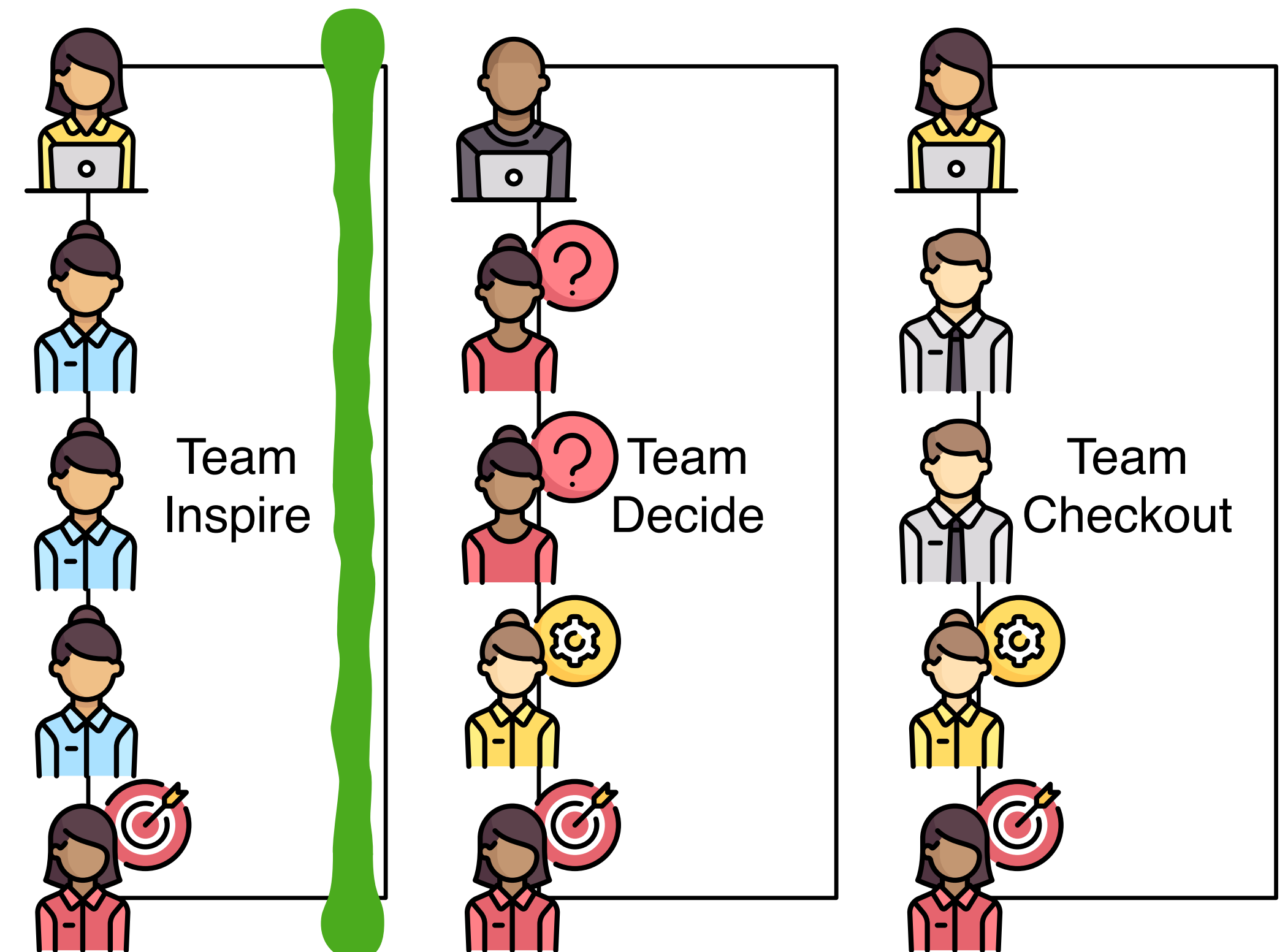
Specialist Teams



 **inter-team communication**

developing a new feature

Cross Functional Teams





Development Speed



Customer Focus

direct feedback loop

no pure api teams

Reduced Scope

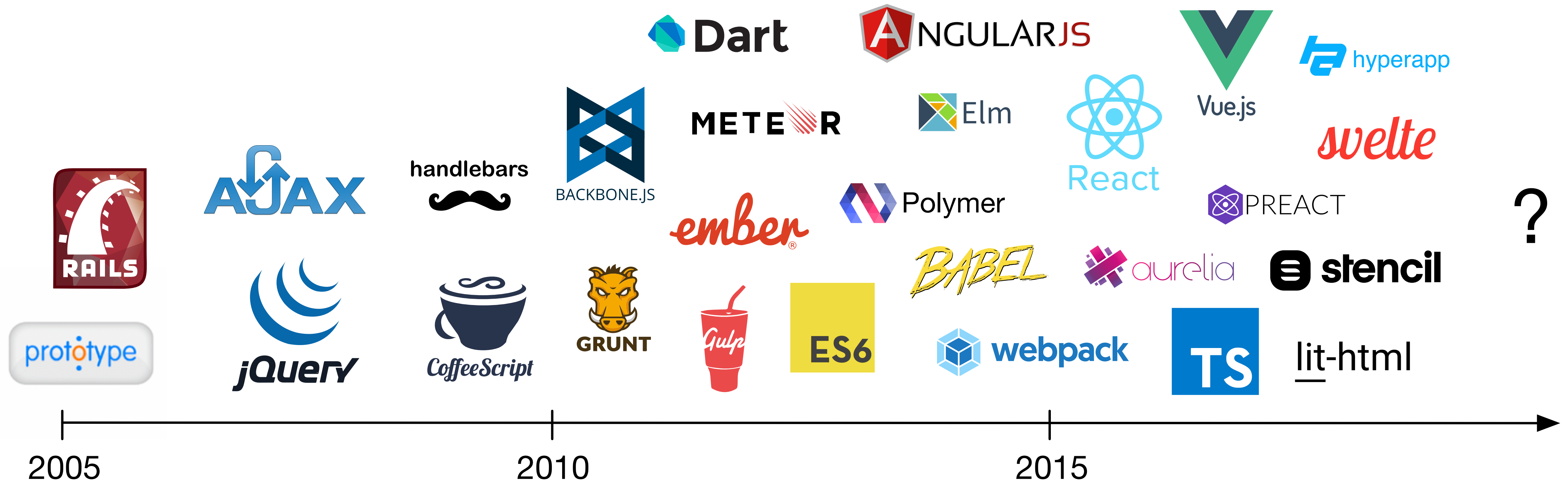
everything fits
into memory again
↑
my



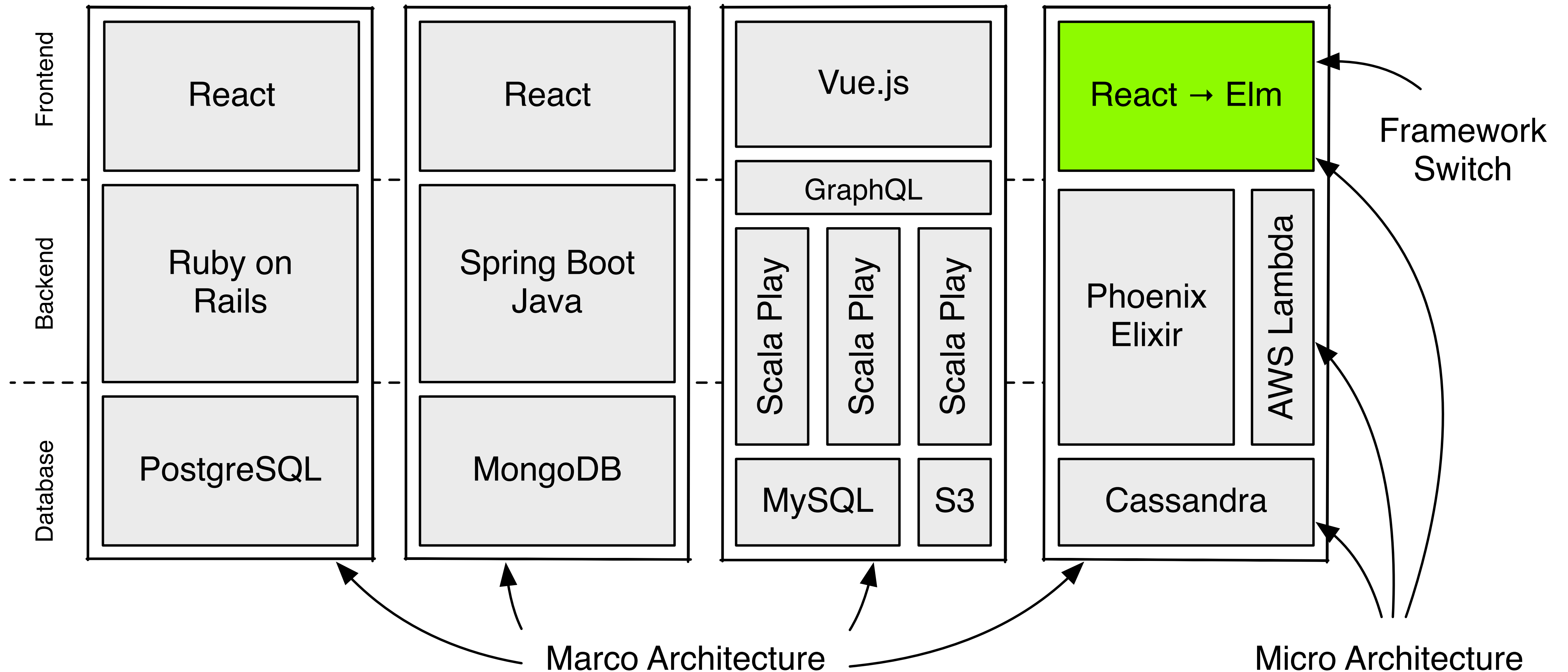


Frontend Renovation
without throwing everything away

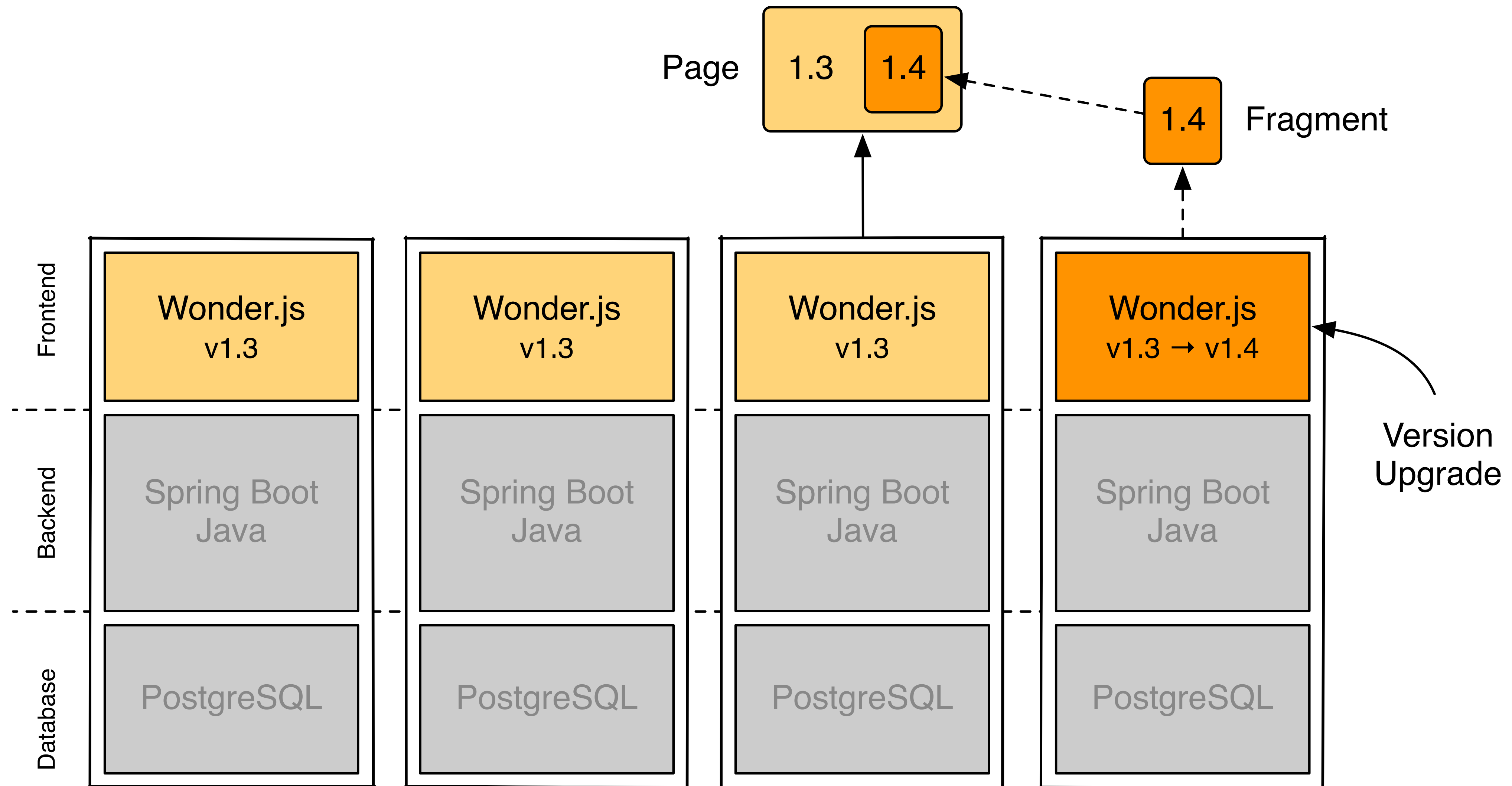
frontend tools change fast



be able to switch frameworks ...



... or upgrade at your own pace.



Who works like this?

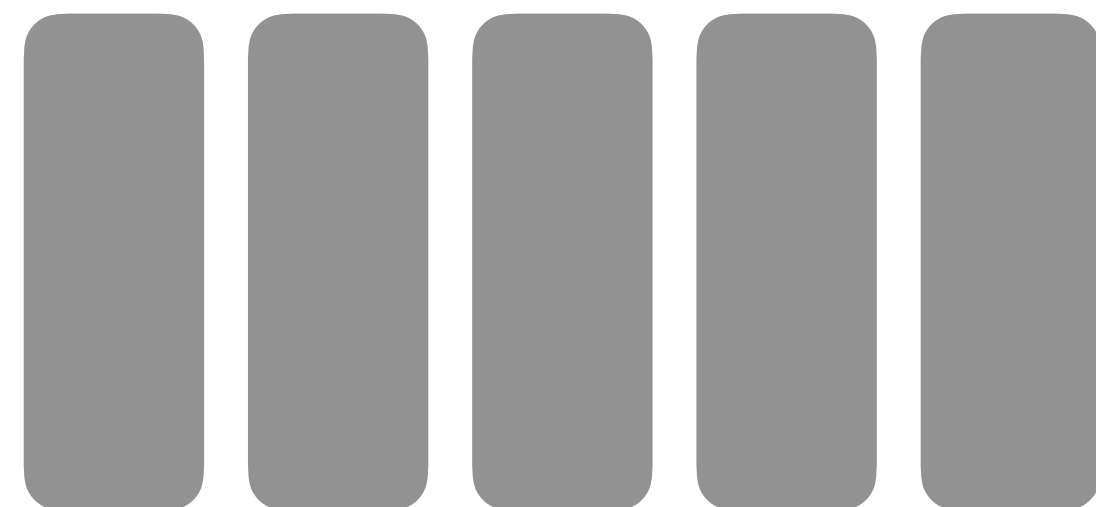


What's our experience?

New e-commerce platforms

GALERIA KAUFHOF

116 department stores
in Germany & Belgium



2014

5 teams

6-8 people/team

MANŮFACTUM.

premium material goods
print catalogs & 9 stores



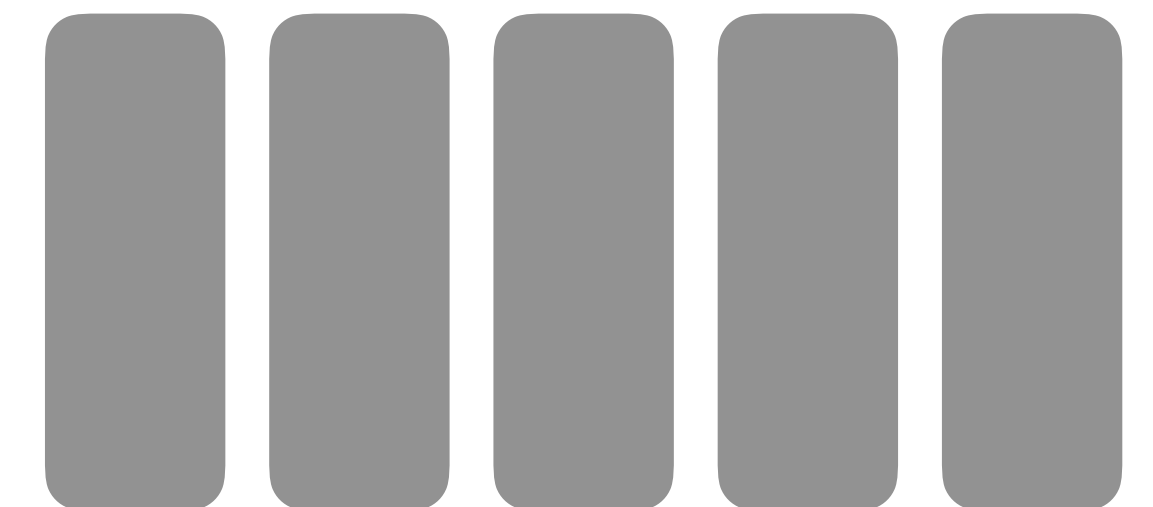
2016

2 teams

5-7 people/team

B breuninger

premium fashion
11 department stores



2016

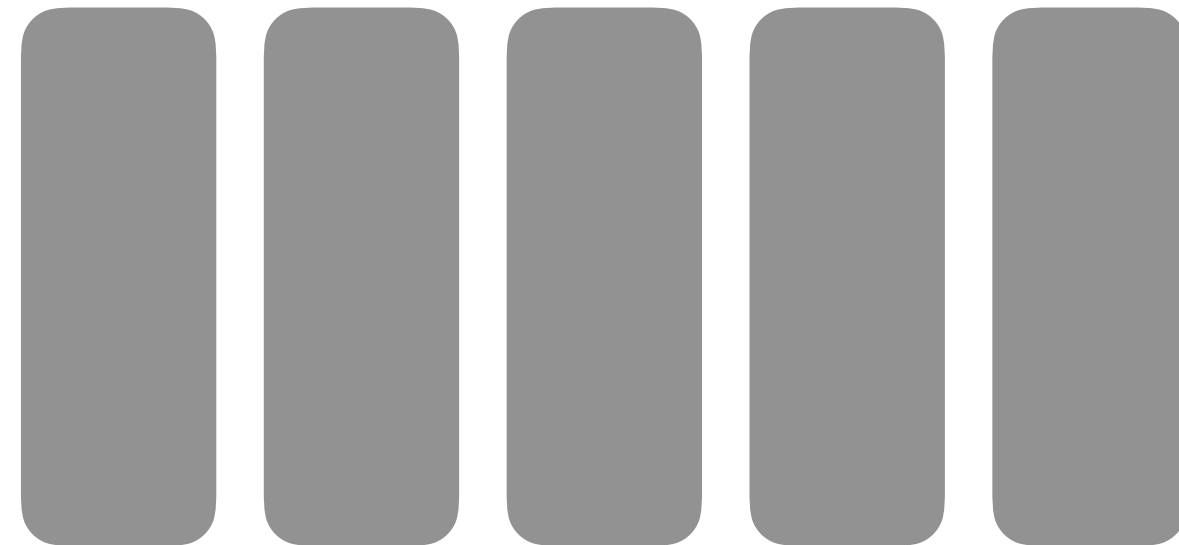
5 teams

6-10 people/team

Project: Next Level Commerce

KLINGEL
Gruppe

fashion for best ager
15 brands in up to 12 countries
~ 65 online stores



2018

5 teams
7-10 people/team
from 3 dev-companies



green field project

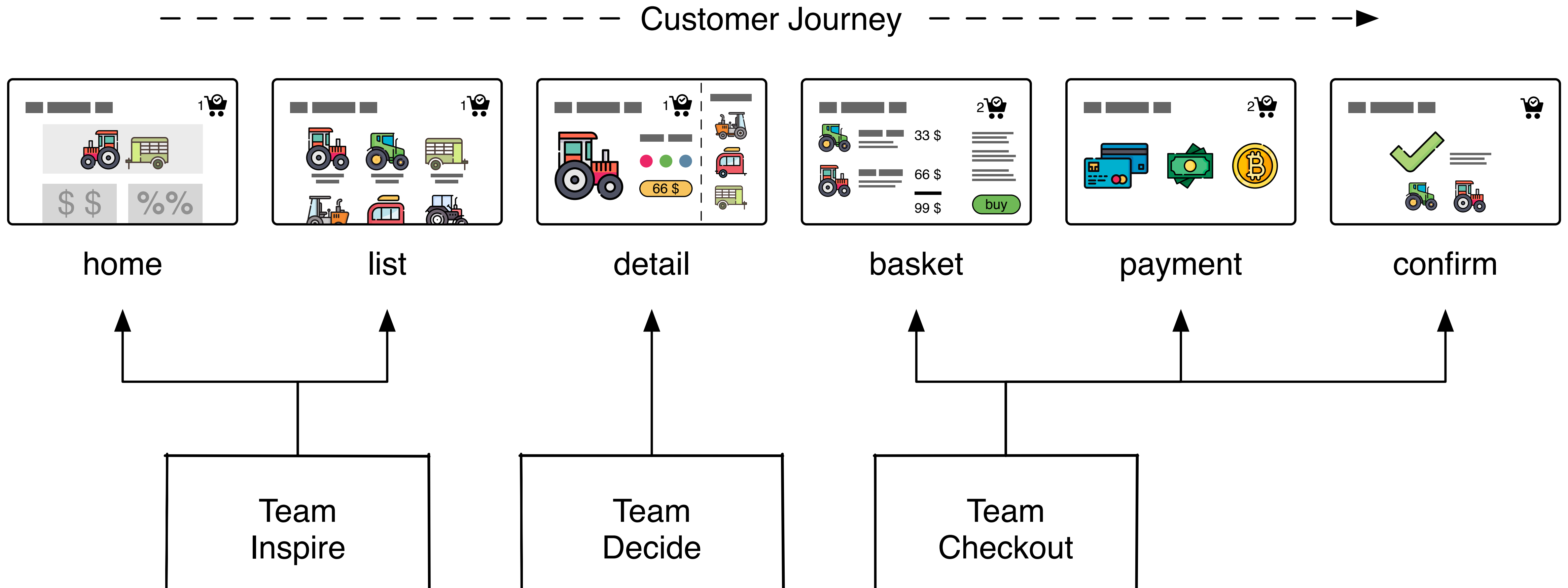
after 2 month
go-live with a
newly created brand
taschenherz.com

6 month later
first migration of a
real brand
mona.de

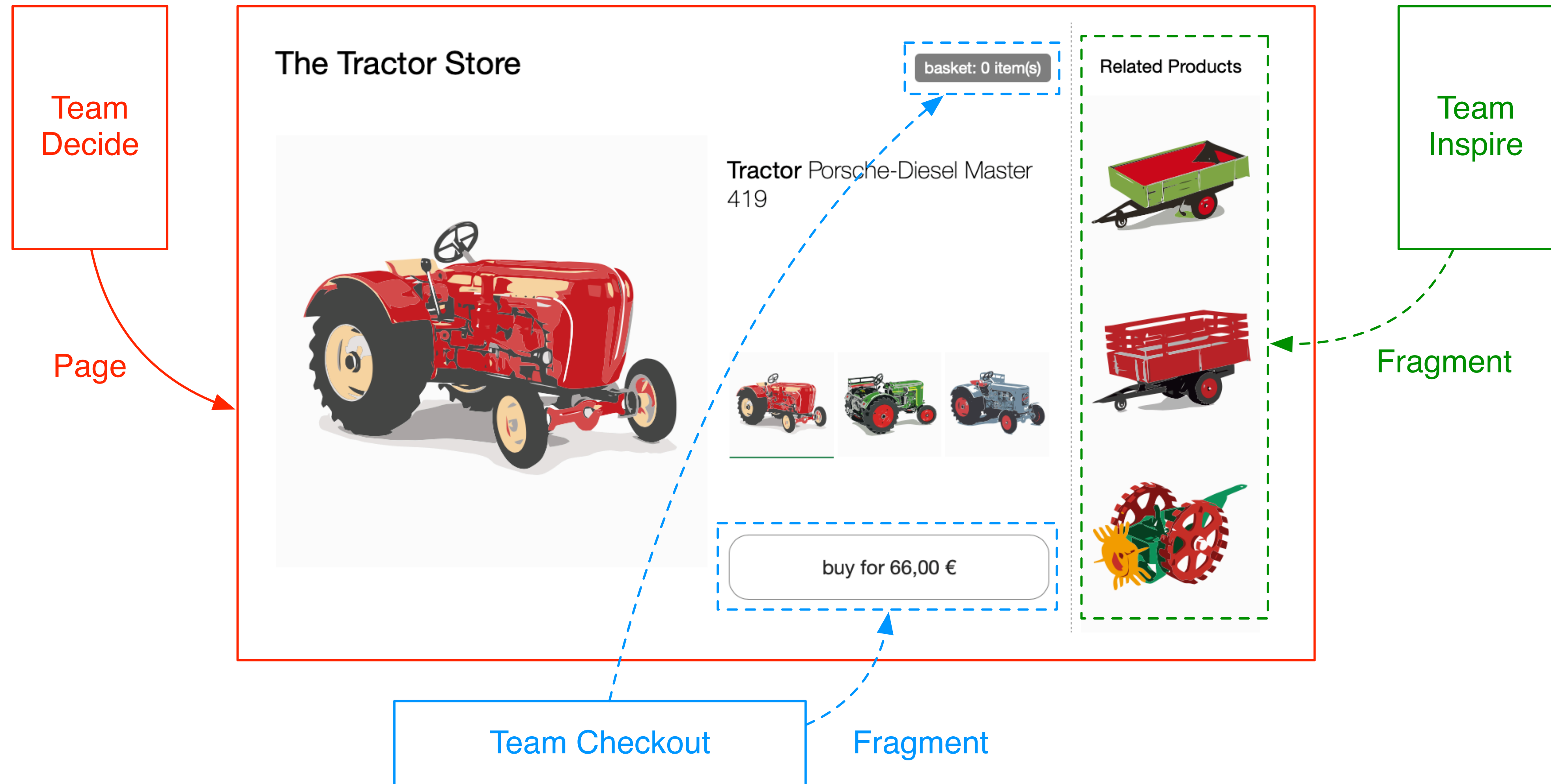
3.

How to implement?

A Page is Owned by one Team



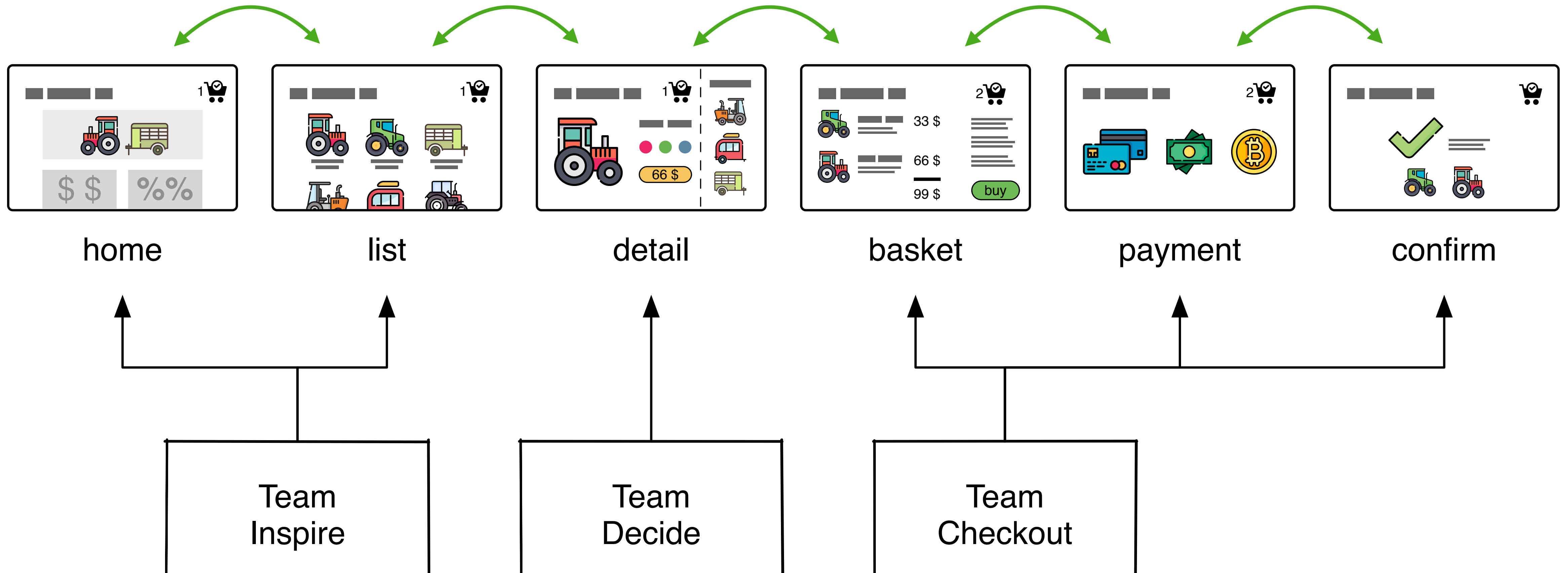
Including Fragments from other Teams





Server Side Integration

via Links



The Model Store



Tractor Porsche-Diesel Master
419



basket: 0 item(s)

buy for 66,00 €

Related Products



Team Decide
owns the page

Team Inspire
provides a fragment

Team Checkout
provides two fragments

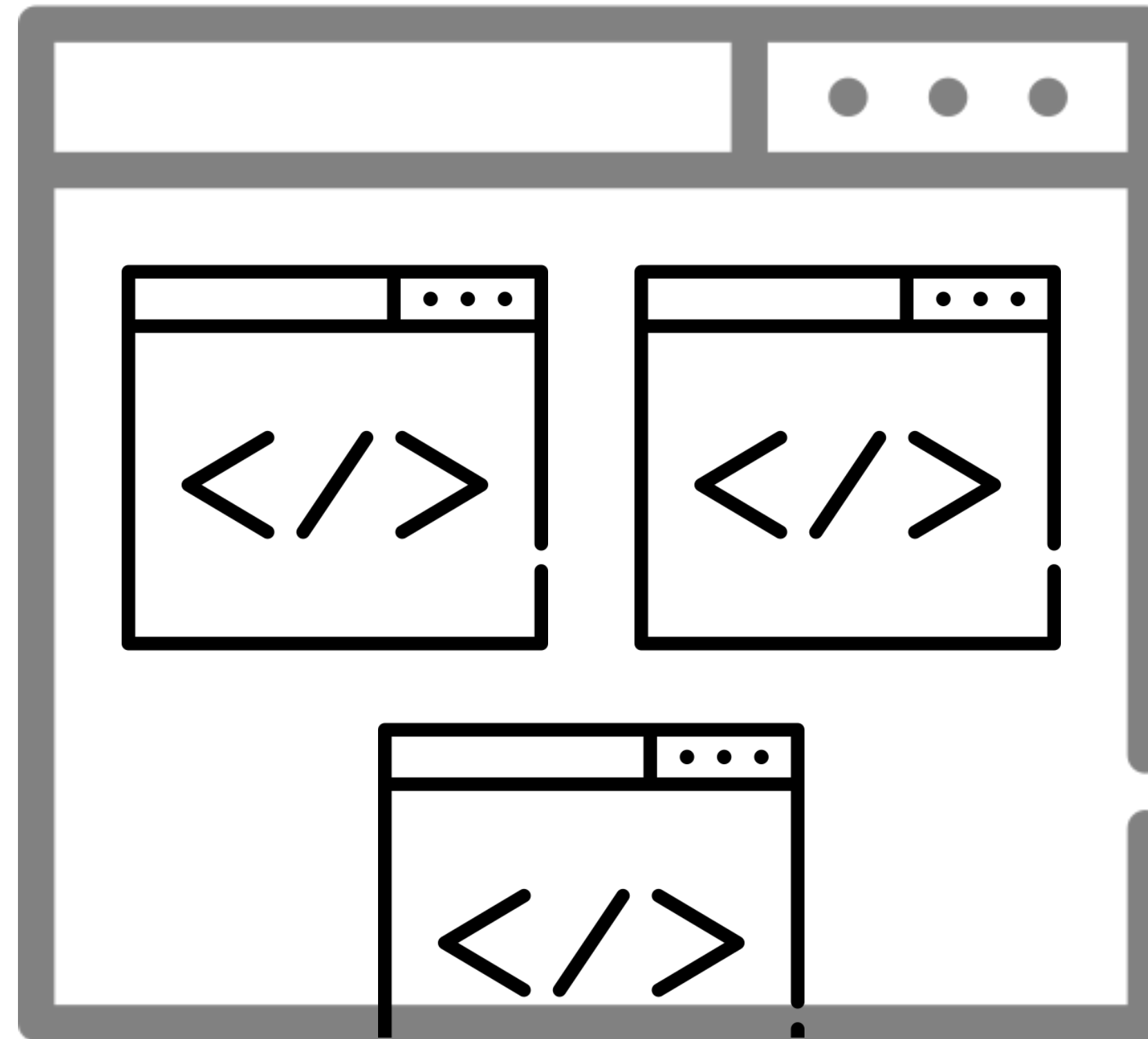
via iframes

Advantages

- strong isolation
- work everywhere

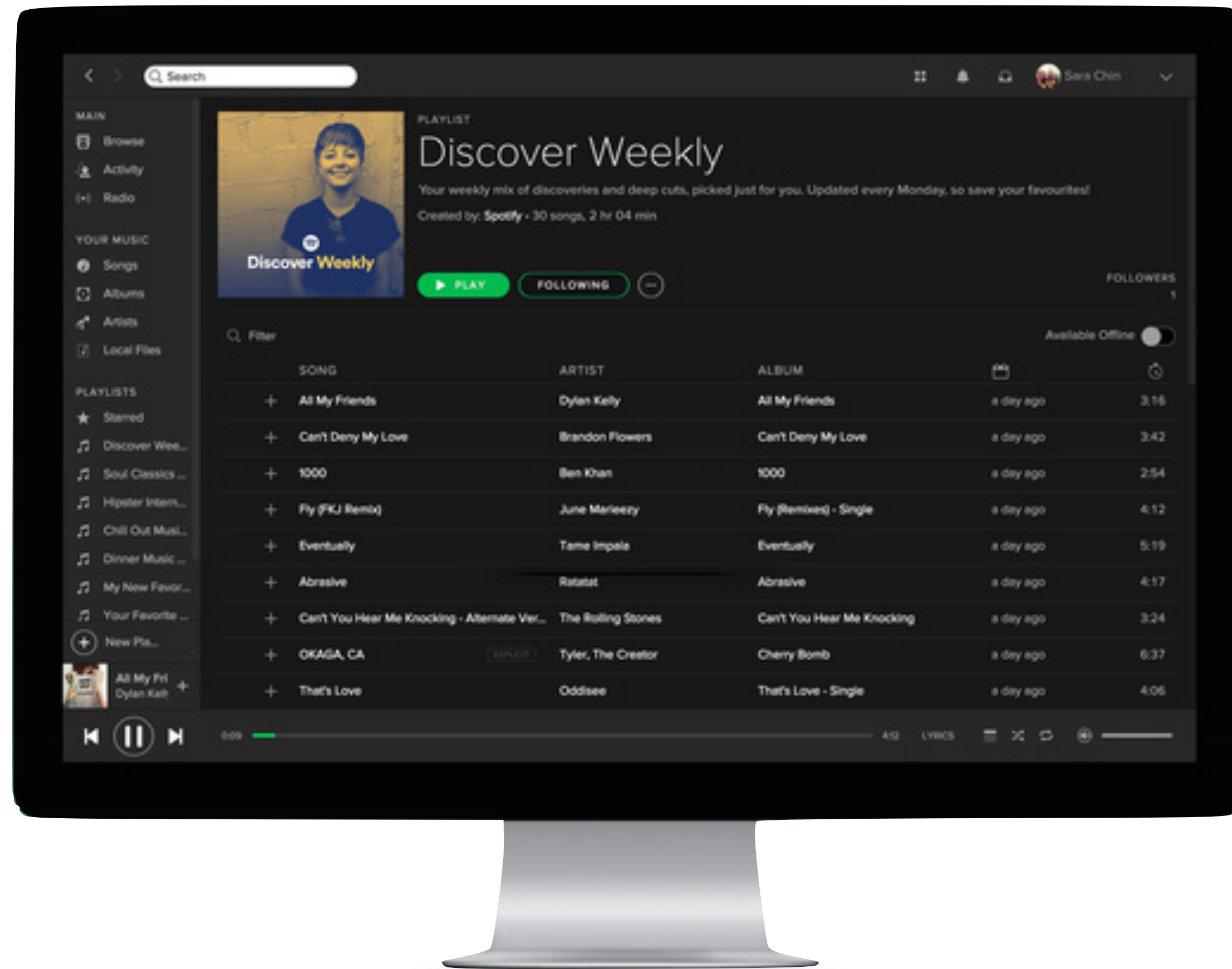
Disadvantages

- bad for SEO
- load performance
- layout constrains
- poor accessibility



O RLY?

via iframes



Spotify web player (until early 2019) & desktop app
<https://labs.spotify.com/2019/03/25/building-spotifys-new-web-player/>

*SSI Server Side Includes**

```
<!--#include virtual="/some-url" -->
```

*or ESI Edge Side Includes - they have similar taste

Revised for
Internet Explorer 5
and Netscape 6



VISUAL
QUICKSTART
GUIDE

JASON CRANFORD TEAGUE

DHTML AND CSS FOR THE WORLD WIDE WEB

Second Edition

Teach yourself DHTML and CSS
the quick and easy way! This
Visual QuickStart Guide uses

pictures rather than lengthy

The easiest way to import external content
into a Web page is to use a server-side
include.

no time!

Urheberrechtlich geschütztes Material

Chapter 23

Using Server-Side Includes

The easiest way to import external content into a Web page is to use a *server-side include*. This is not an HTML tag, but a tag that tells the computer that serves your Web pages to the Internet (called a *server*) to add the referenced file to the page. Although this method's success depends on whether your server understands this tag, almost all servers do these days.

To add a server-side include:

1. `<!-- #include virtual="external.html" -->`
Add the include tag anywhere in the `<body>` of your HTML document, and set it to import `external.html` (**Code 23.3**).
2. `<!-- #echo var="DATE_LOCAL" -->`
`<!-- #echo var="DATE_GMT" -->`
Another useful server side tag is the `#echo` tag in which the server includes either its local time and date or Greenwich Mean Time (GMT) on the page.
3. `external.html`
Create a new HTML file, and save it as `external.html`. This file does not contain the regular `<html>` open and close tags, which are supplied by the main document—only the `<body>` tag and any HTML that could be included in a regular HTML document (**Code 23.4**). The results are shown in **Figure 23.3**.

✓ Tips

- The disadvantage of this method is that you cannot see the external content unless it is coming off a server. If you try to view this file on your local hard disk, you will see a whole lot of nothing.
- You can also add a clock that shows visitors their local time (see "Creating a Clock" in Chapter 26).

Code 23.3 The server-side include tags allow you to import external content, but only if the page is being delivered from a Web server.

```
code
<html>
<body>
  <!-- #include virtual="external.html" -->
  <br><br>
  <!-- #echo var="DATE_LOCAL" -->
  <br><br>
  <!-- #echo var="DATE_GMT" -->
  <br><br>
</body>
</html>
```

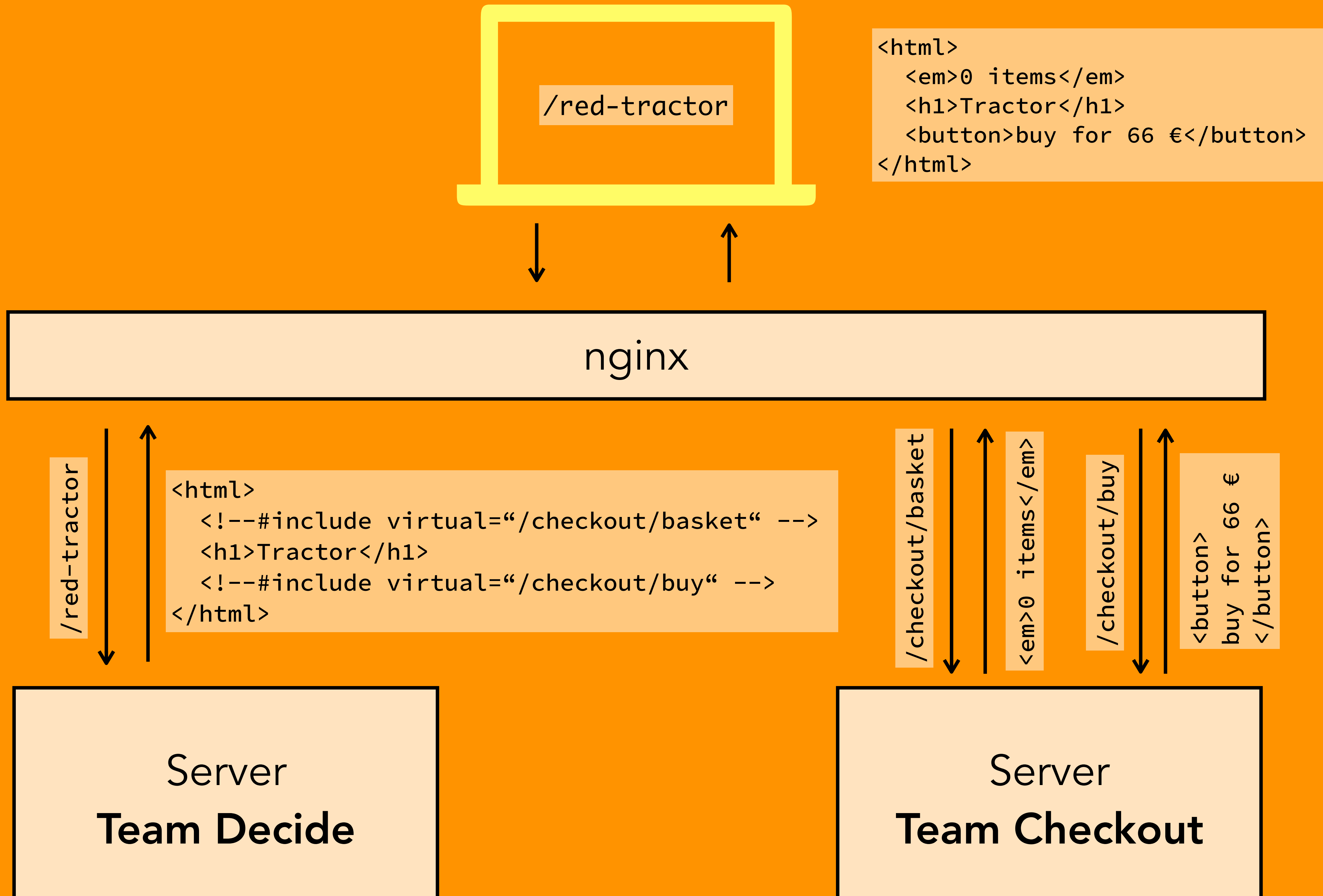
Code 23.4 The external content being imported into `index.html`. This content can be any standard HTML code; just don't use `<html>` or `<body>` tags.

```
code
<div style="text-align:center">
  <h1>Alice In Wonderland</h1>
  <h3>Chapter 1</h3>
  
```



Figure 23.3 The external content has been imported successfully.

Server Side Includes



Client Side Integration



Interactivity

The Model Store



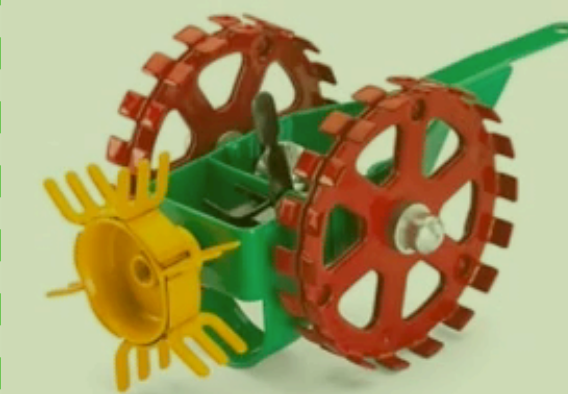
Tractor Porsche-Diesel Master
419



basket: 0 item(s)

buy for 66,00 €

Related Products



Team Decide
owns the **page**

Team Inspire
provides a **fragment**

Team Checkout
provides two **fragments**

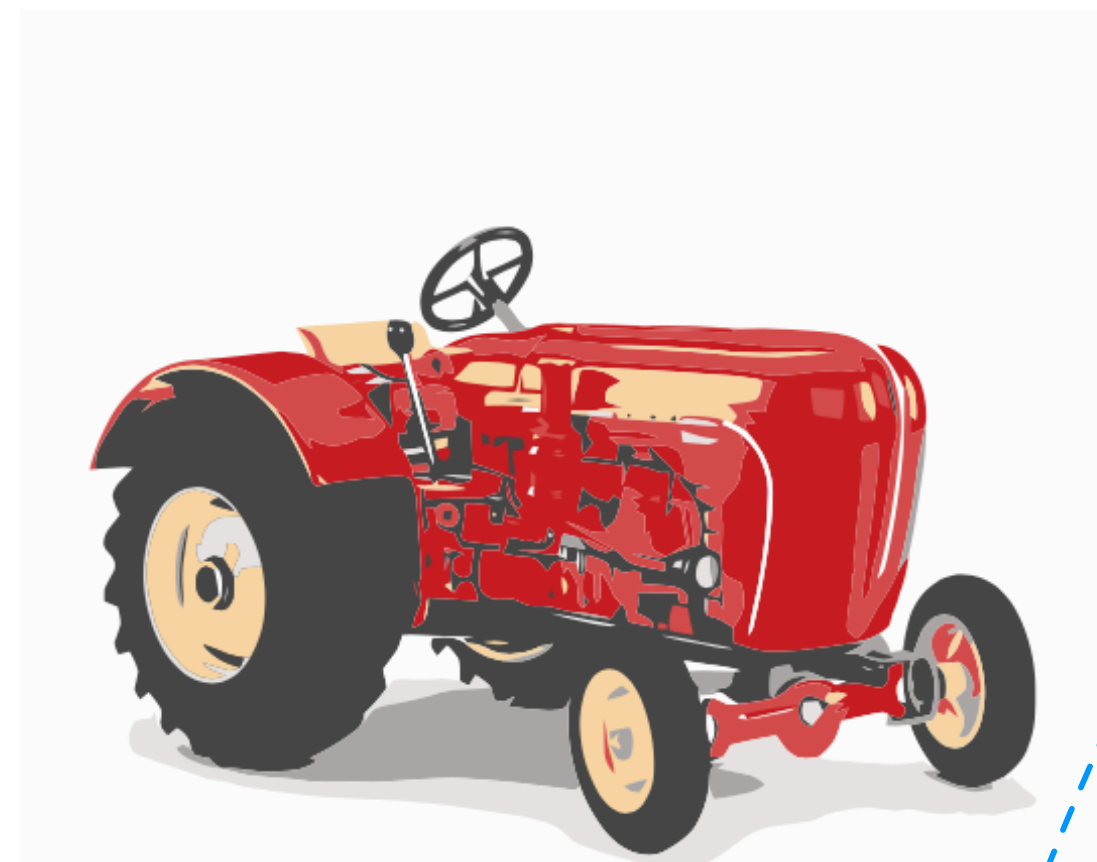
Each Fragment as a Mini Application



Team
Decide

Page

The Tractor Store



Tractor Porsche-Diesel Master
419

basket: 0 item(s)



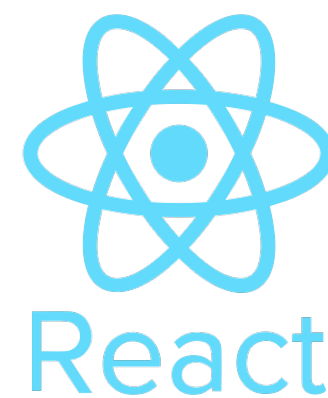
buy for 66,00 €

Related Products



Team
Inspire

Fragment



Team Checkout

Fragment

Web Components

Custom
Elements

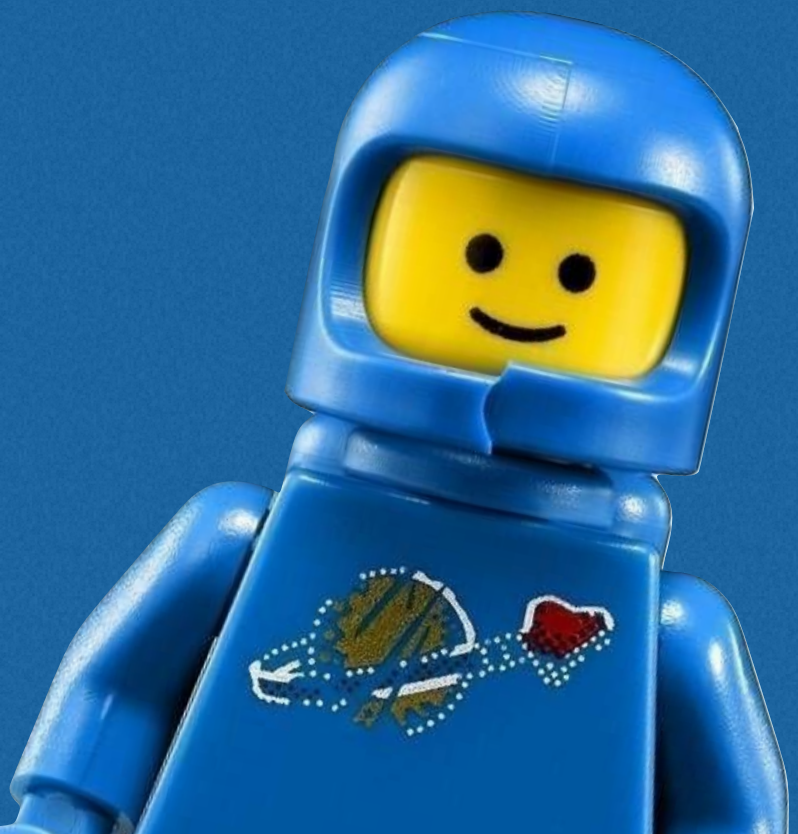
Shadow
DOM

HTML
Template



Custom Elements

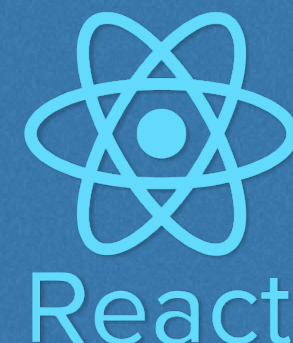
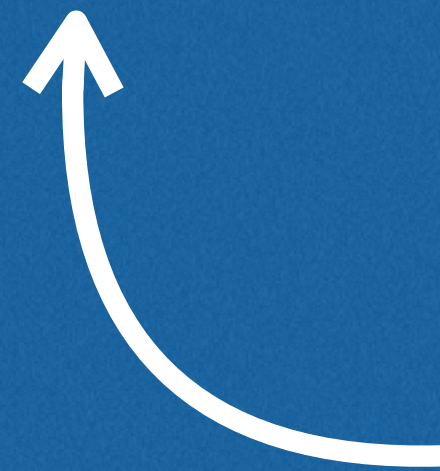
`<checkout-basket></checkout-basket>`





Custom Elements

```
class CheckoutBasket extends HTMLElement {  
  connectedCallback() {  
    this.innerHTML = 'mini basket ...';  
  }  
}
```



Choo

```
customElements.define('checkout-basket', CheckoutBasket);
```


Element Lifecycle

```
class CheckoutBasket extends HTMLElement {
```

```
  constructor() {...}    is created
```

```
  connectedCallback() {...}    attached to DOM
```

```
  attributeChangedCallback(attr, oldVal, newVal) {...}
```

```
  disconnectedCallback() {...}    someone changed an attribute
```

```
    removed from DOM, cleanup!
```

```
}
```


Browser Support



Custom Elements v1

API stabilized end of 2016

Browser Support



<https://github.com/webcomponents/custom-elements>

<checkout-basket>
</checkout-basket>

basket: 0 item(s)

Tractor Porsche-Diesel Master
419



buy for 66,00 €

Related Products







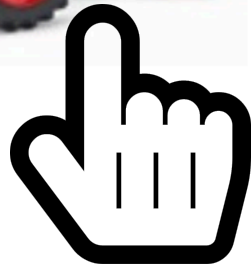
<inspire-reco sku="t_red">
</inspire-reco>

<checkout-buy sku="t_red">
</checkout-buy>

<checkout-basket>
</checkout-basket>

basket: 0 item(s)

Tractor Porsche-Diesel Master
419



buy for 66,00 €

Related Products

<inspire-reco sku="t_green">
</inspire-reco>

<checkout-buy sku="t_green">
</checkout-buy>

the DOM is the API

Teams publish their Custom Elements documentation
Element-Name, Attributes, Events

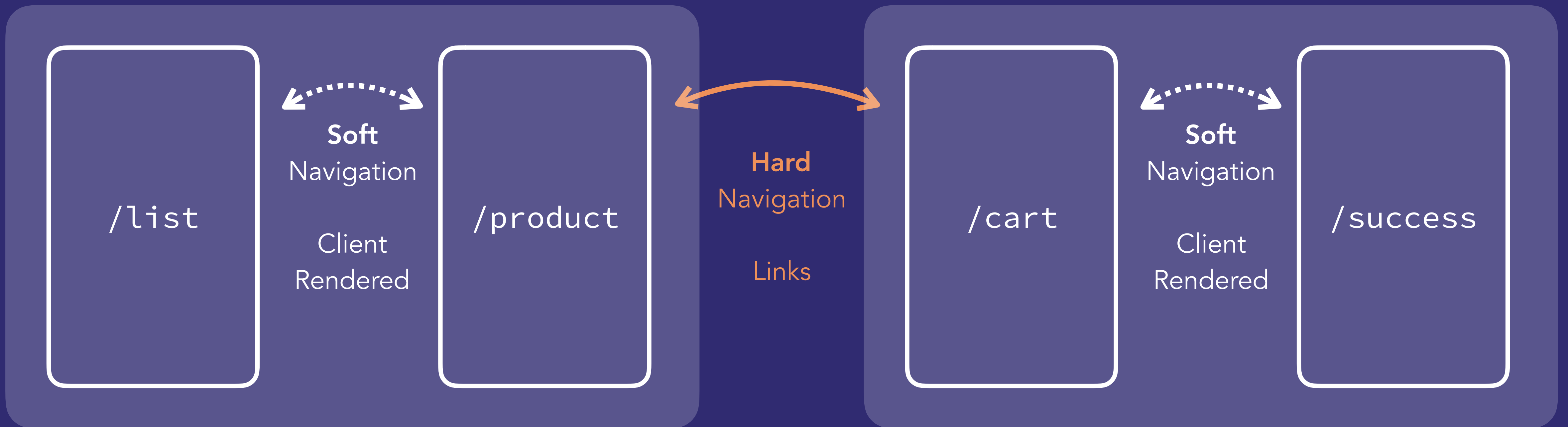
Page Transitions

Only Inside a Team

one router per team

Team A

Team B

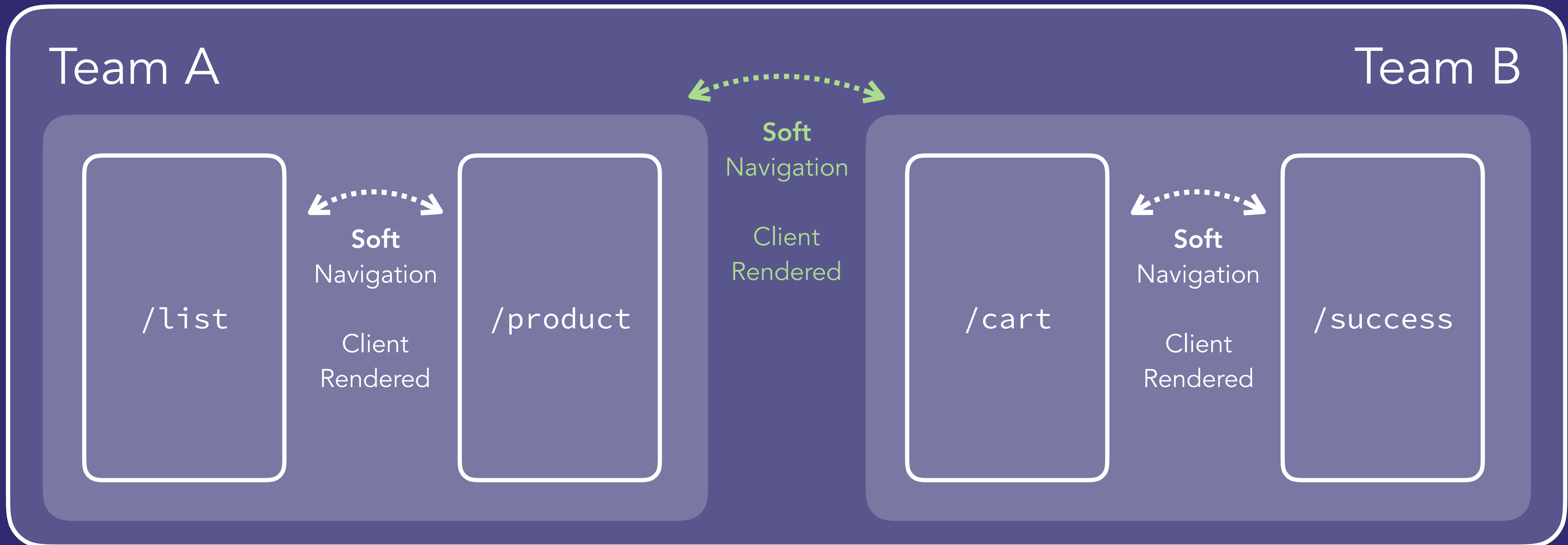


Across Teams

increased complexity!
shared runtime code
harder debugging

App Shell

meta router & router per team



Short Integration Recap

	Pages	Fragments
Server	Links	SSI/ESI
Client	App-Shell Meta-Router	Web Components

other Libraries & Meta Frameworks



single-spa

<https://github.com/CanopyTax/single-spa>



Podium

<https://podium-lib.io>



OpenComponents

<https://github.com/opencomponents/oc>



<https://github.com/onerzafer/microfe-client>



<https://github.com/zalando/tailor>

meta-spa-router

<https://github.com/manfredsteyer/meta-router>

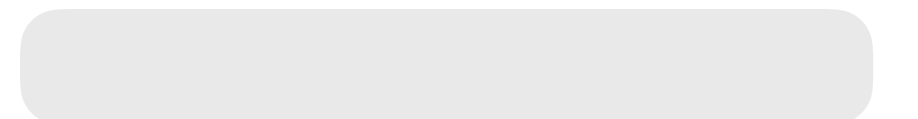
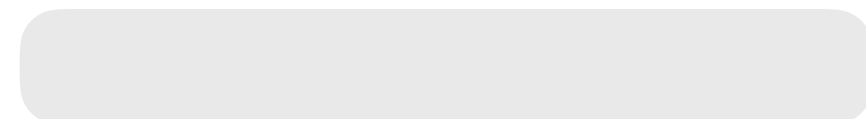
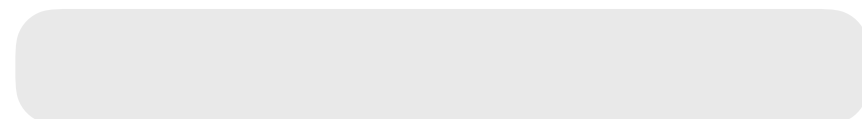
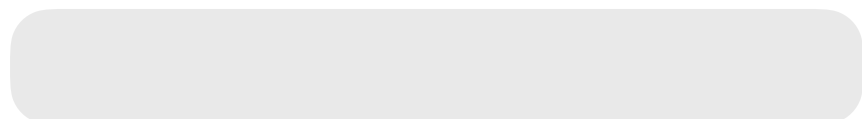


nitty gritty details



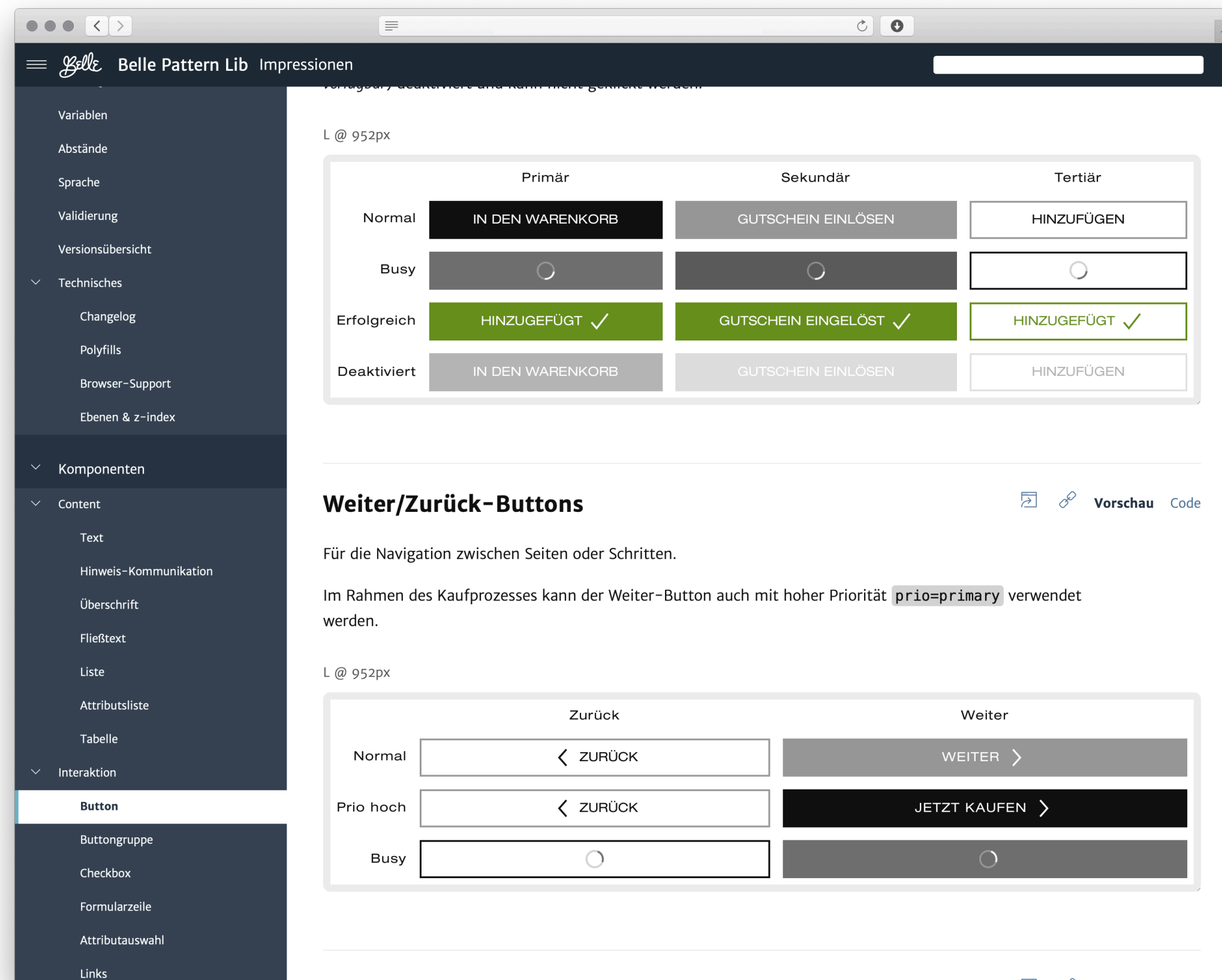


coherent
ux & design



Have a Design System

Design System / Pattern Library



Contents

Design Tokens (fonts, colors, ...)

Usable Components (buttons, ...)

Aspects

Design

Users Experience

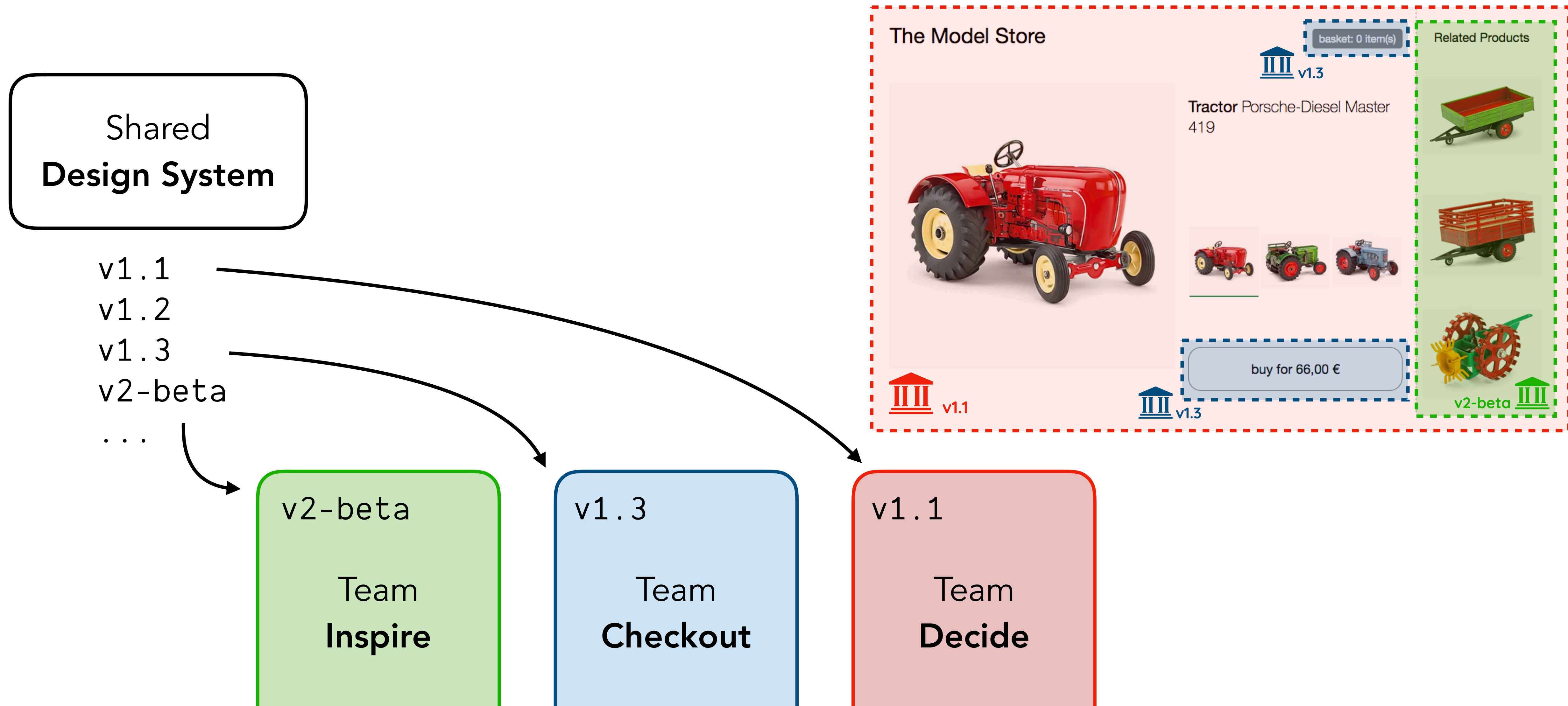
Performance

Accessibility

Dev Experience

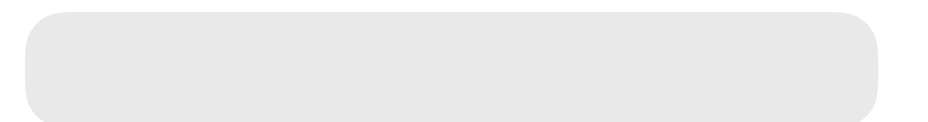
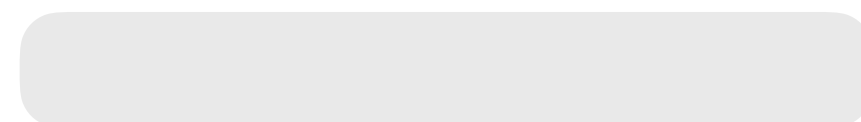
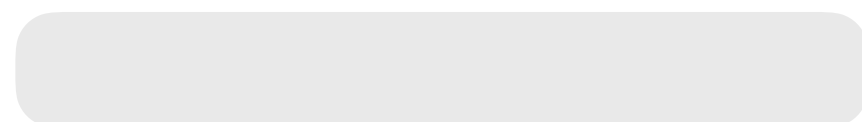
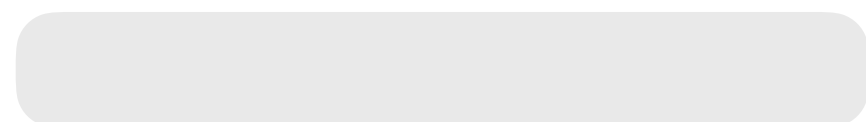
Tools: Ulengine, Storybook, ...

Distributed as **Versioned Package**





web performance



Performance **First**

measure regularly
pick your **metrics**

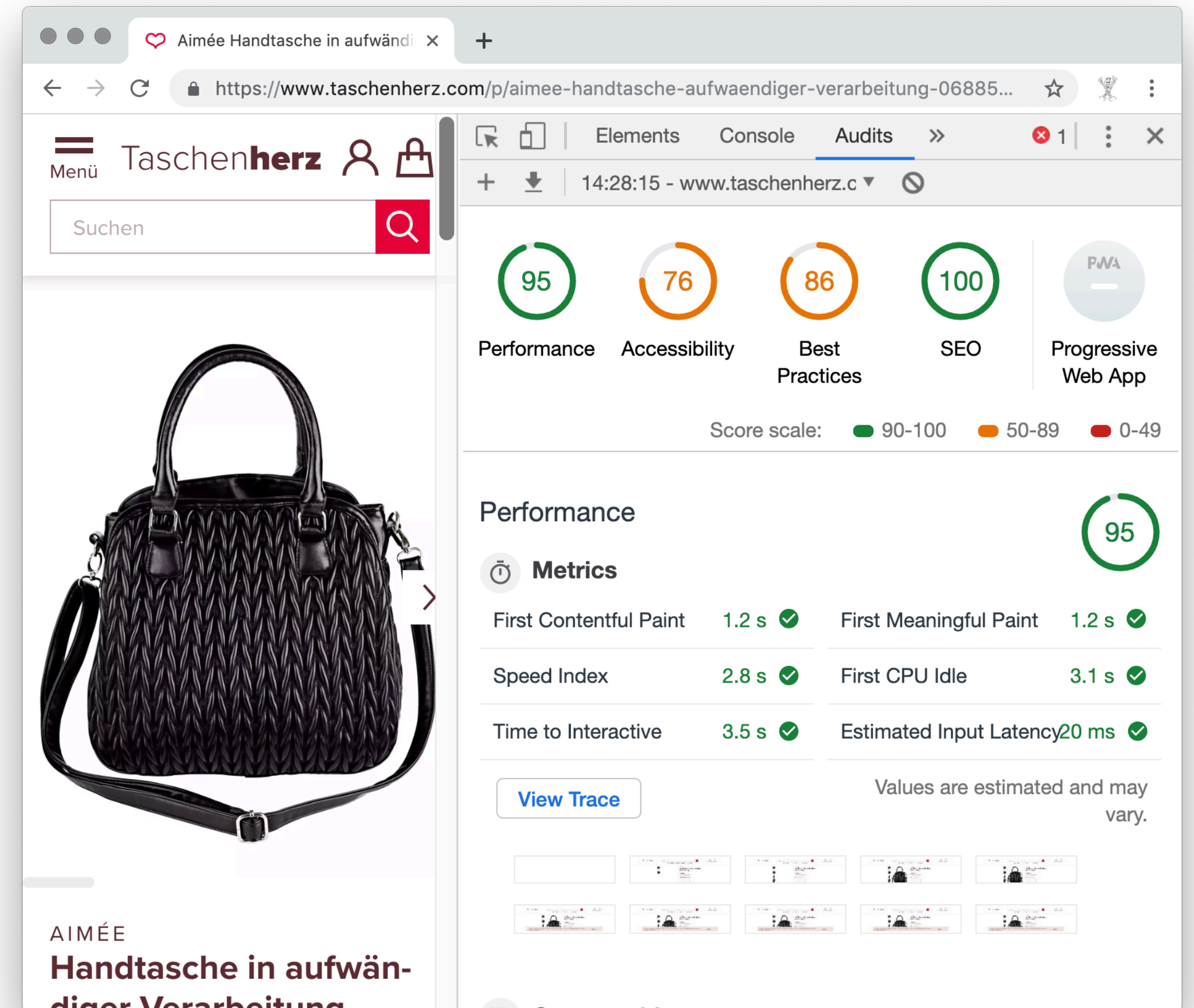


Tools

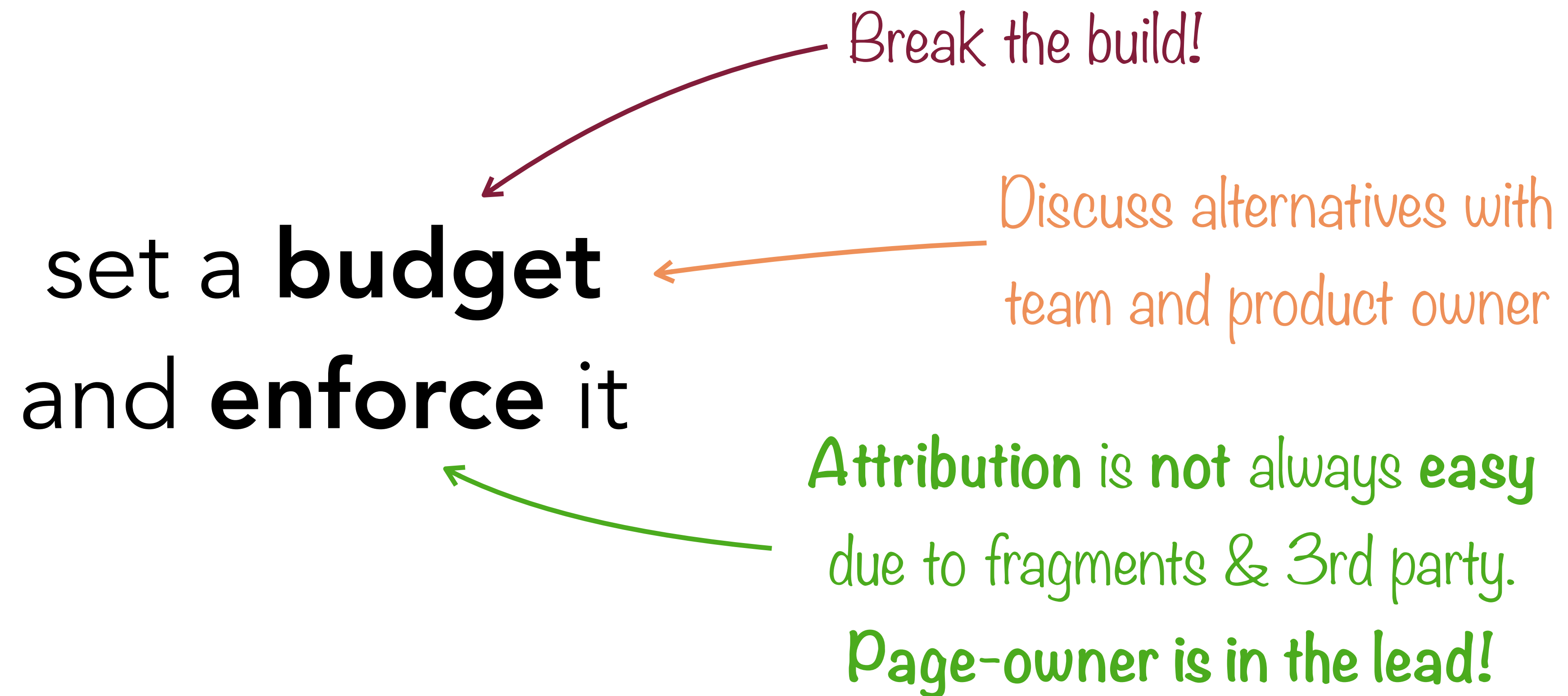
Lighthouse
Sitespeed.io
WebpageTest
Speedcurve
...

Metrics

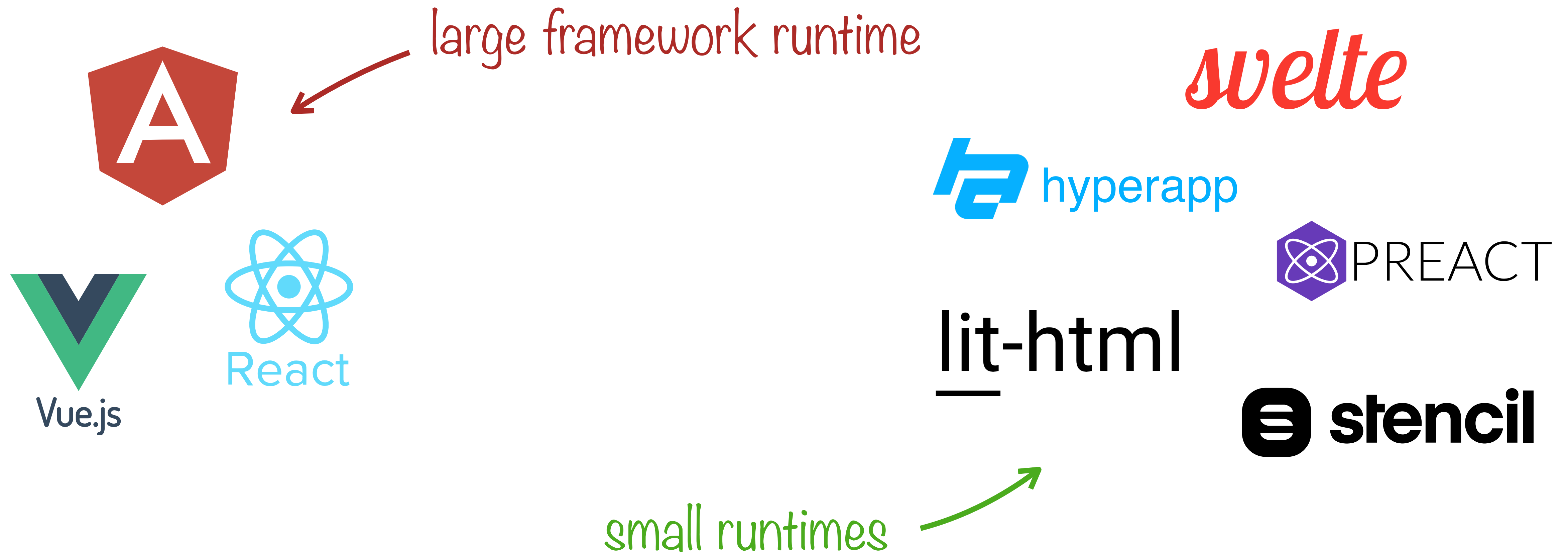
Lighthouse Scores
Time to Interactive
Visually Complete
Page Weight
...

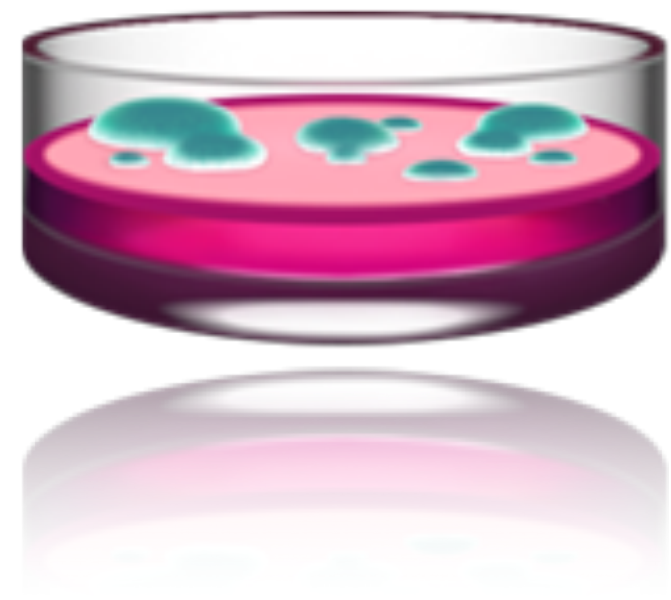


integrate perf into you culture

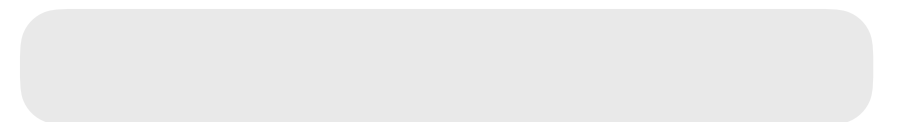
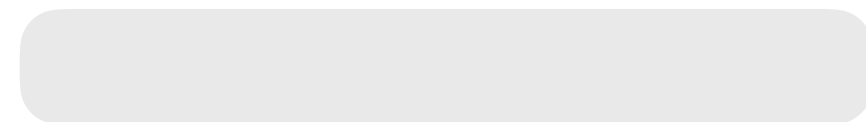
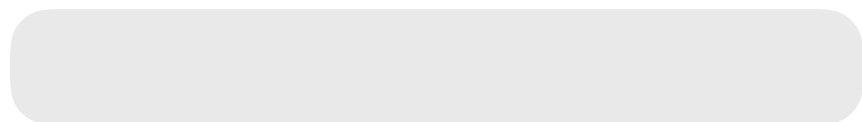
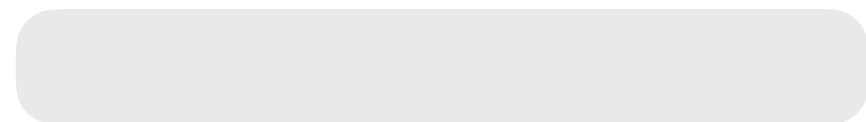


make **responsible** tech choices





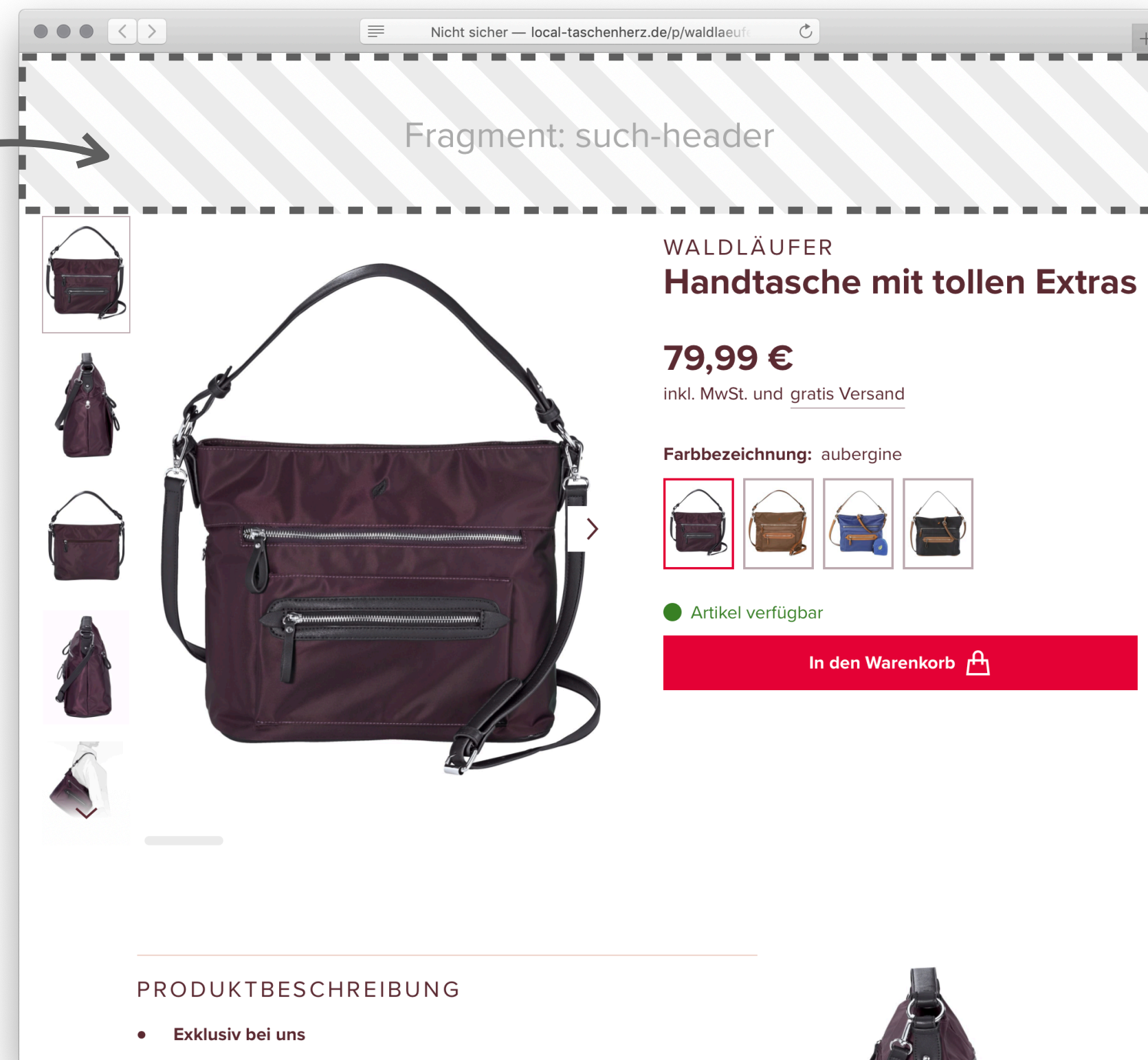
quality & testing



testing in isolation

majority of
the tests

dummy
fragment

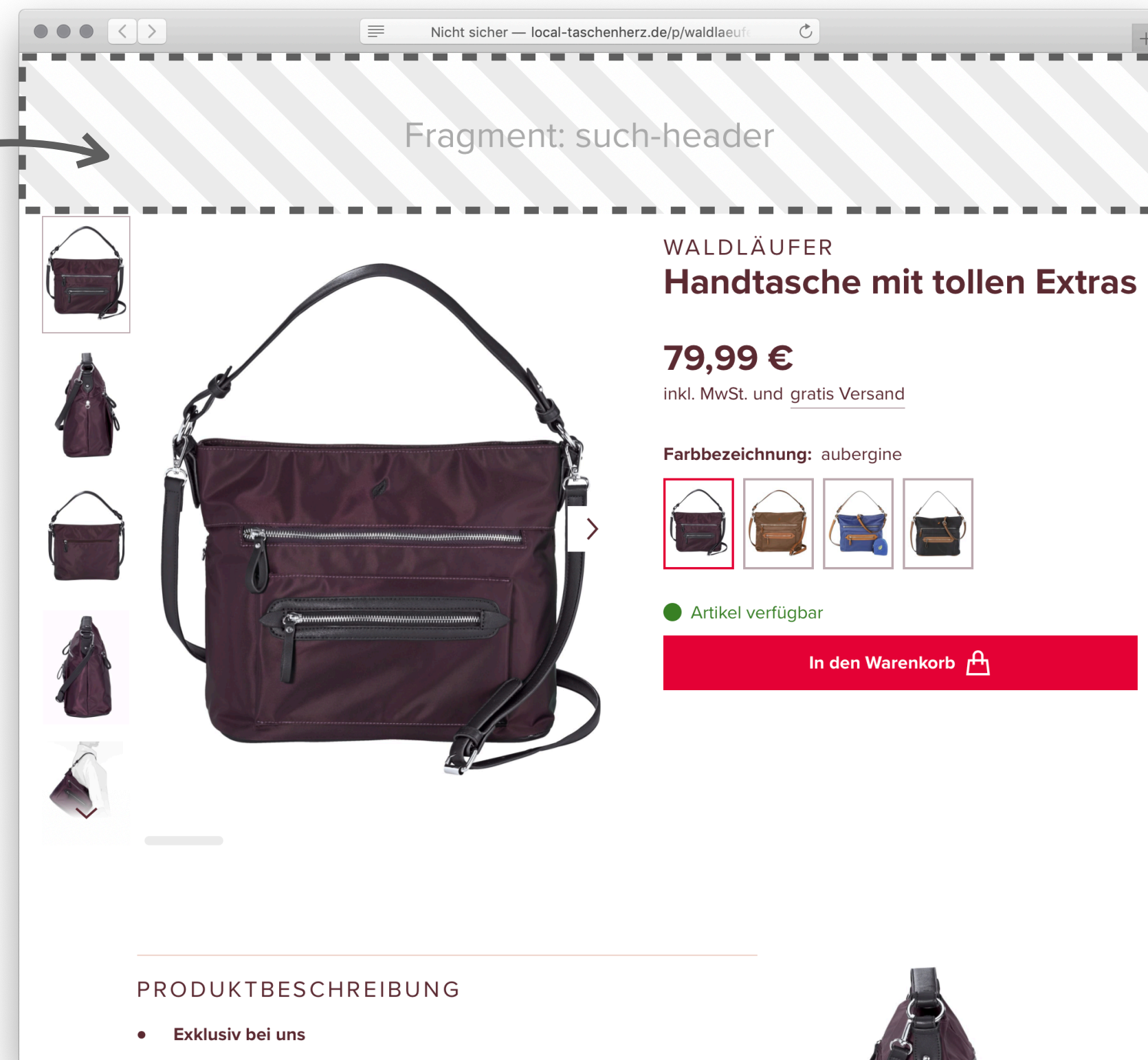


product page works
without fragments
Team Decide **owns & tests**

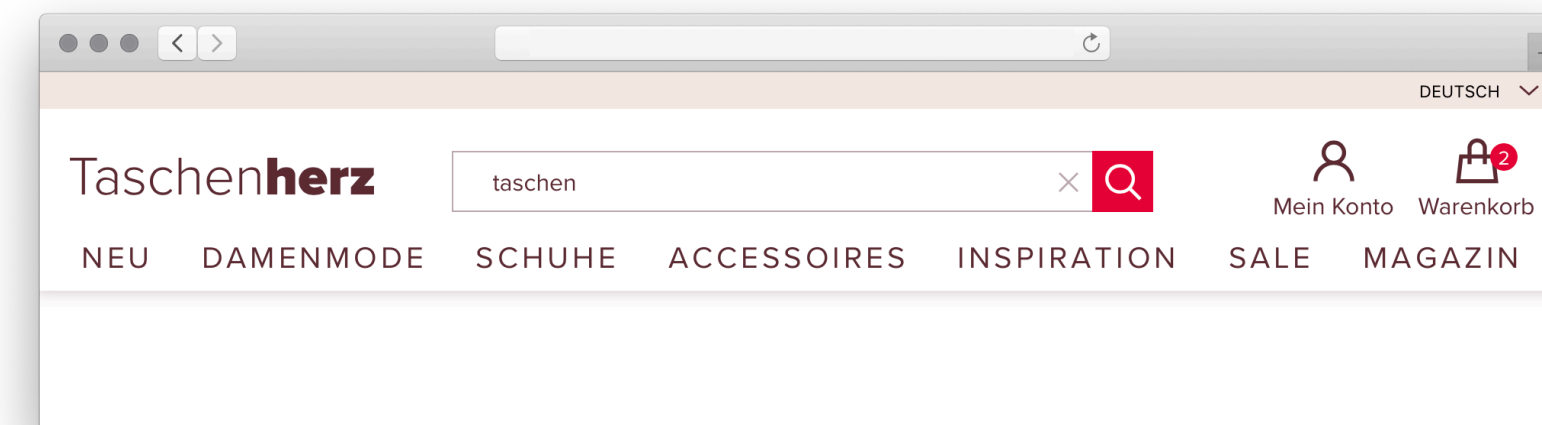
testing in isolation

majority of
the tests

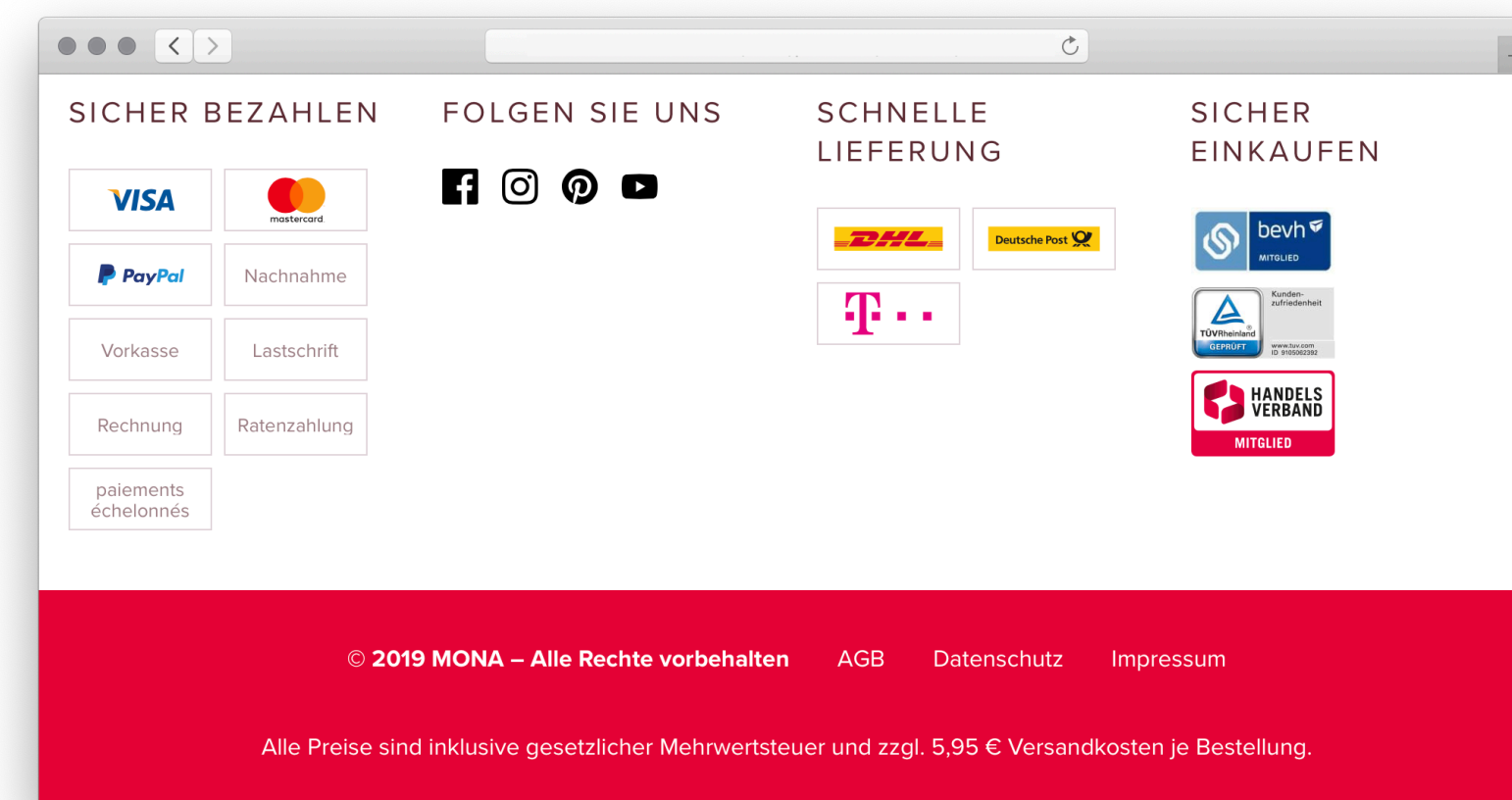
dummy
fragment



product page works
without fragments
Team Decide owns & tests



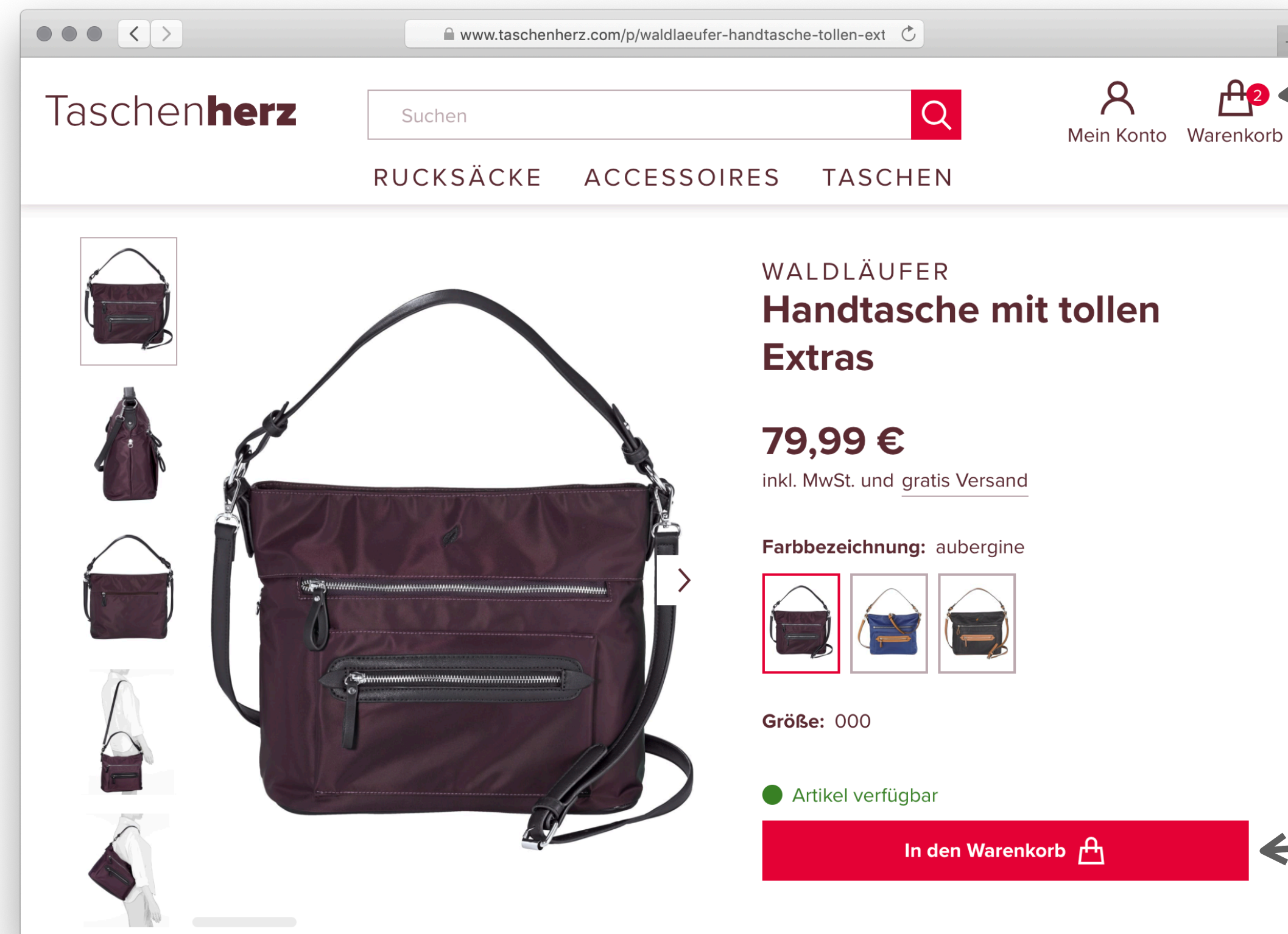
header & footer are
standalone fragments
Team Search owns & tests



testing accross boundaries

In some cases

cross team tests exist but are the exception

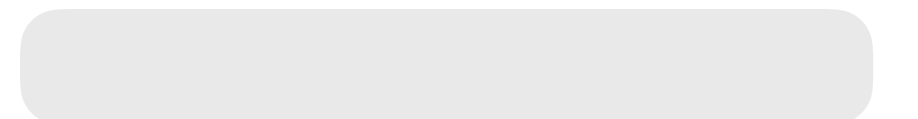
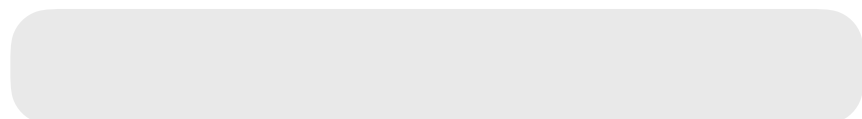
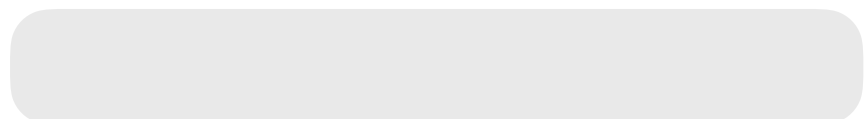
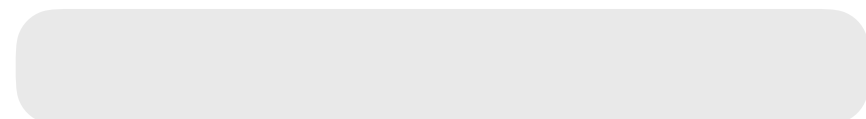


... increments this.

Clicking here ...



knowledge
sharing



regular rituals

people from all teams
talk on a specific topic

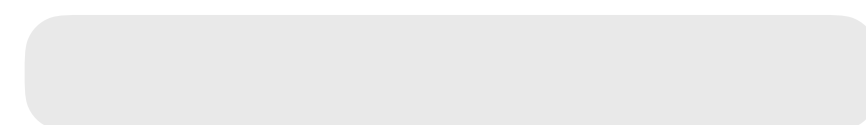
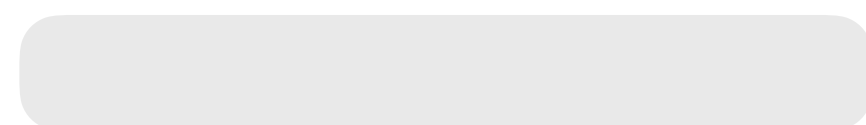
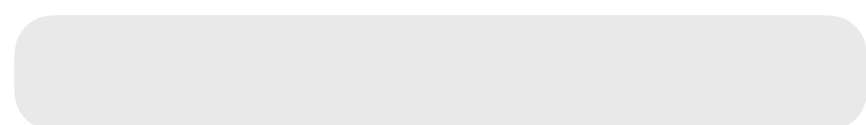
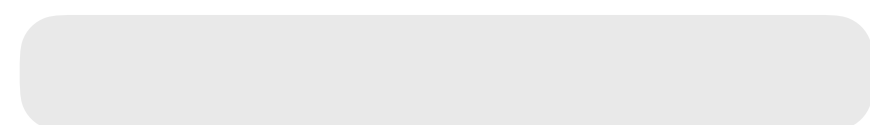


mostly weekly

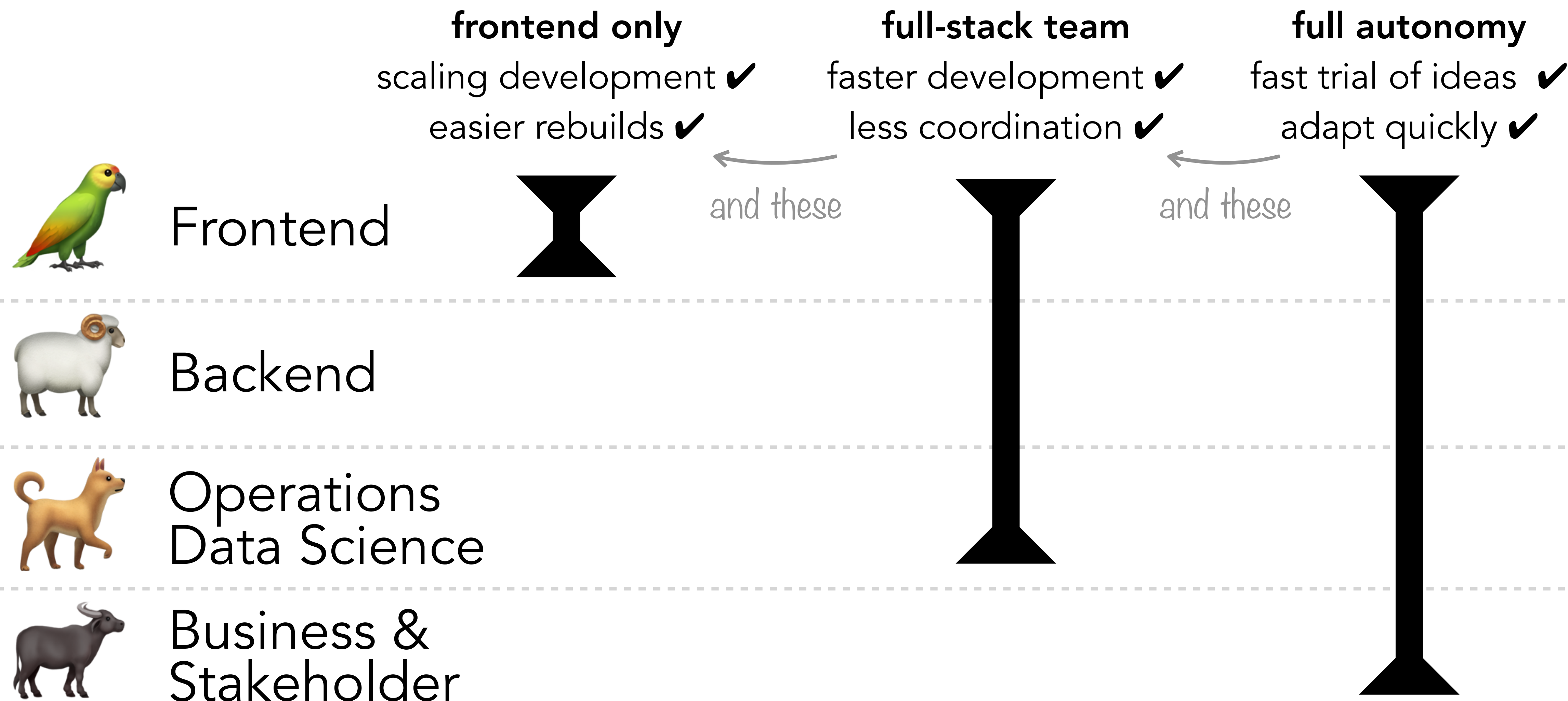
...



vertical depth



How Deep Should a Vertical Team be?



Want to learn more?

Extensive Link Collection

by Elisabeth Engel [@_lizzelo_](#)

micro-frontends.org 

by Me [@naltatis](#)

Microservice Websites

by Gustav Nilsson Kotte [@gustaf_nk](#)

Self-Contained Systems

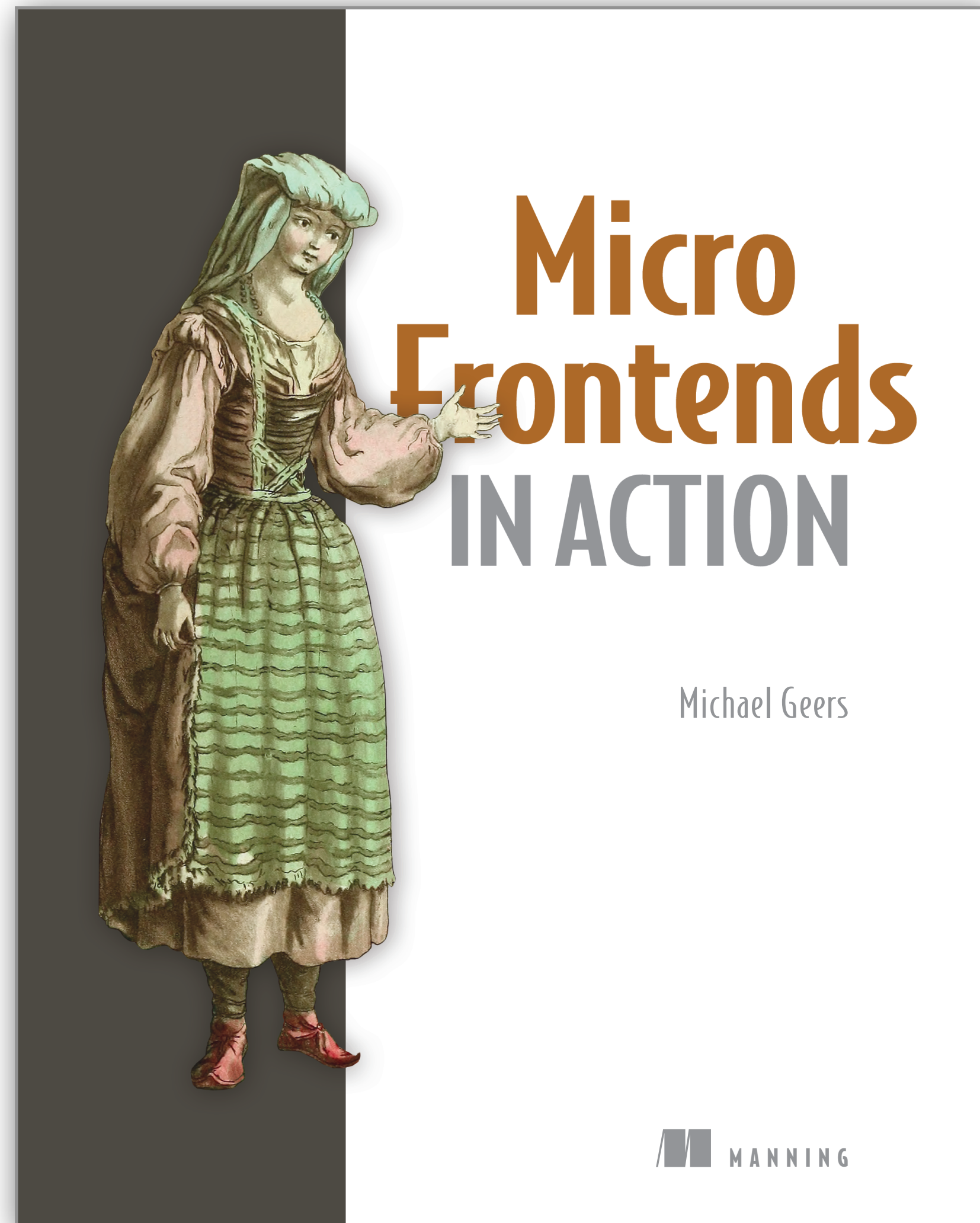
by INNOQ

Micro Frontends at DAZN

by Luca Mezzalira [@lucamezzalira](#)

Talk: Frontend Monolith

by Stefan Tilkov [@stilkov](#)



I'm writing a book

MEAP - Manning Early Access Program
this summer

follow [@naltatis](https://twitter.com/naltatis) or [@manningbooks](https://twitter.com/manningbooks) for updates

Frontend

Backend

lets' blur this line!





A diagram showing a vertical stack of three rectangular layers. The top layer is dark gray and contains the text 'Frontend'. The middle layer is a thin, light gray horizontal bar. The bottom layer is dark gray and contains the text 'Backend'. A white arrow points from the text 'let's blur this line!' to the middle light gray bar.

Frontend

Backend

let's blur this line!

Thanks for listening!



happyend

Michael Geers
@naltatis

Credits



Burano Pixaline

<https://pixabay.com/images/id-2522251/>

Black Diamond (me)

Holiday Snap 2: Tron Apionid

<https://flic.kr/p/r23SxR>

Drummers skeeze

<https://pixabay.com/images/id-642540/>

Tocotronic Stefan Müller

<https://flic.kr/p/JCTQXd>

People Icons freepik

<https://www.flaticon.com/family/special/lineal-color>

Action Bicycling Roman Pohorecki

<https://www.pexels.com/photo/action-bicycling-bike-biking-287398/>

Wooden Counter David Siglin

<https://www.pexels.com/photo/blur-blurry-bokeh-close-up-347139/>

The Tool Guy Tirachard Kumtanom

<https://unsplash.com/photos/UuW4psOb388>

Woodshop Igor Ovsyannykov

<https://unsplash.com/photos/iXV0i4Wo4yc>

HP C7000 BladeSystem pchow98

<https://flic.kr/p/7AVF23>

Close up Susanne Nilsson

<https://flic.kr/p/nppBcE>

Unicorn d97jro

<https://pixabay.com/images/id-671593/>

Browser Logos Cătălin Mariș

<https://github.com/alrra/browser-logos>

Lego Benny Cheezburger

<http://gph.is/1BCRCKh>

Tractors Manufactum

<https://www.manufactum.com/tin-toys-c193667/>

built with Micro Frontends