

# Micro Frontends

Große Web Apps modular entwickeln

Robin Müller | Jörg Neumann

ACLUE GmbH | [info@aclue.de](mailto:info@aclue.de)



# DIE SAMPLES



<https://tinyurl.com/5y5ym7bv>



# DIE DEMOS

## GITHUB REPOS

- [HTTPS://GITHUB.COM/ACLUE-DE/FRONTENDWORKSHOP](https://github.com/aclue-de/frontendworkshop)
- [HTTPS://GITHUB.COM/ACLUE-DE/FRONTENDWORKSHOP\\_PATTERNLIBRARY](https://github.com/aclue-de/frontendworkshop-patternlibrary)
- [HTTPS://GITHUB.COM/ACLUE-DE/FRONTENDWORKSHOP\\_STORYBOOK](https://github.com/aclue-de/frontendworkshop-storybook)

## CODEPEN COLLECTION

- [HTTPS://CODEPEN.IO/COLLECTION/BNRPLE](https://codepen.io/collection/BNRPLE)



# MICRO FRONTENDS

## HERAUSFORDERUNGEN

- \* Große Web Projekte
- \* Kommunikation zwischen den Teams
- \* Abhängigkeiten beim Deployment

## ZIELE

- \* Team Aufteilung
- \* Autonome Deployments
- \* Flexible Komposition in unterschiedlichen Designs

## VORAUSSETZUNGEN

- \* Cross Funktionale Teams
- \* Vertikaler Team Schnitt
- \* Unabhängige Deployment Pipelines
- \* Team-übergreifende Organisation des Designs

BACKEND

FRONTEND



# MICROSERVICES



FRONTEND

# MICROSERVICES



# MICRO FRONTENDS



# SELF-CONTAINED SYSTEM

MICROSERVICES

SERVICE  
1

SERVICE  
2

SERVICE  
3

SERVICE  
4

FRONTEND  
1

FRONTEND  
2

FRONTEND  
3

FRONTEND  
4

MICRO FRONTENDS



# MICRO FRONTENDS

## Technology Agnostic

- \* Jedes Team entscheidet selbst über die eingesetzten Technologien
- \* Implementation Hiding (z.B. Custom Elements)

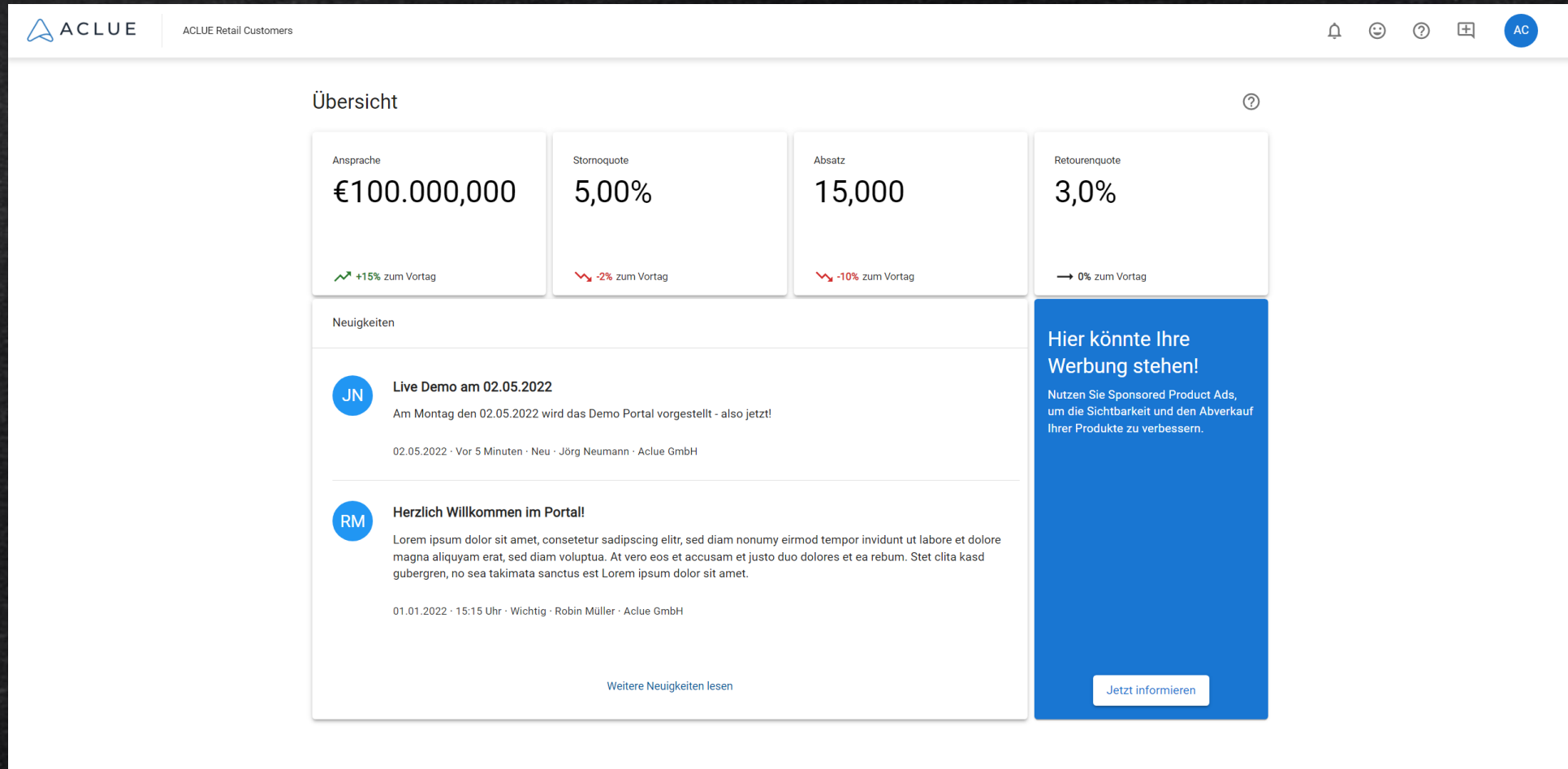
## Isolate Team Code

- \* Keine gemeinsame Runtime oder Framework
- \* Self-contained Systems
- \* Kein Shared State oder globale Variablen

## Team Prefixes

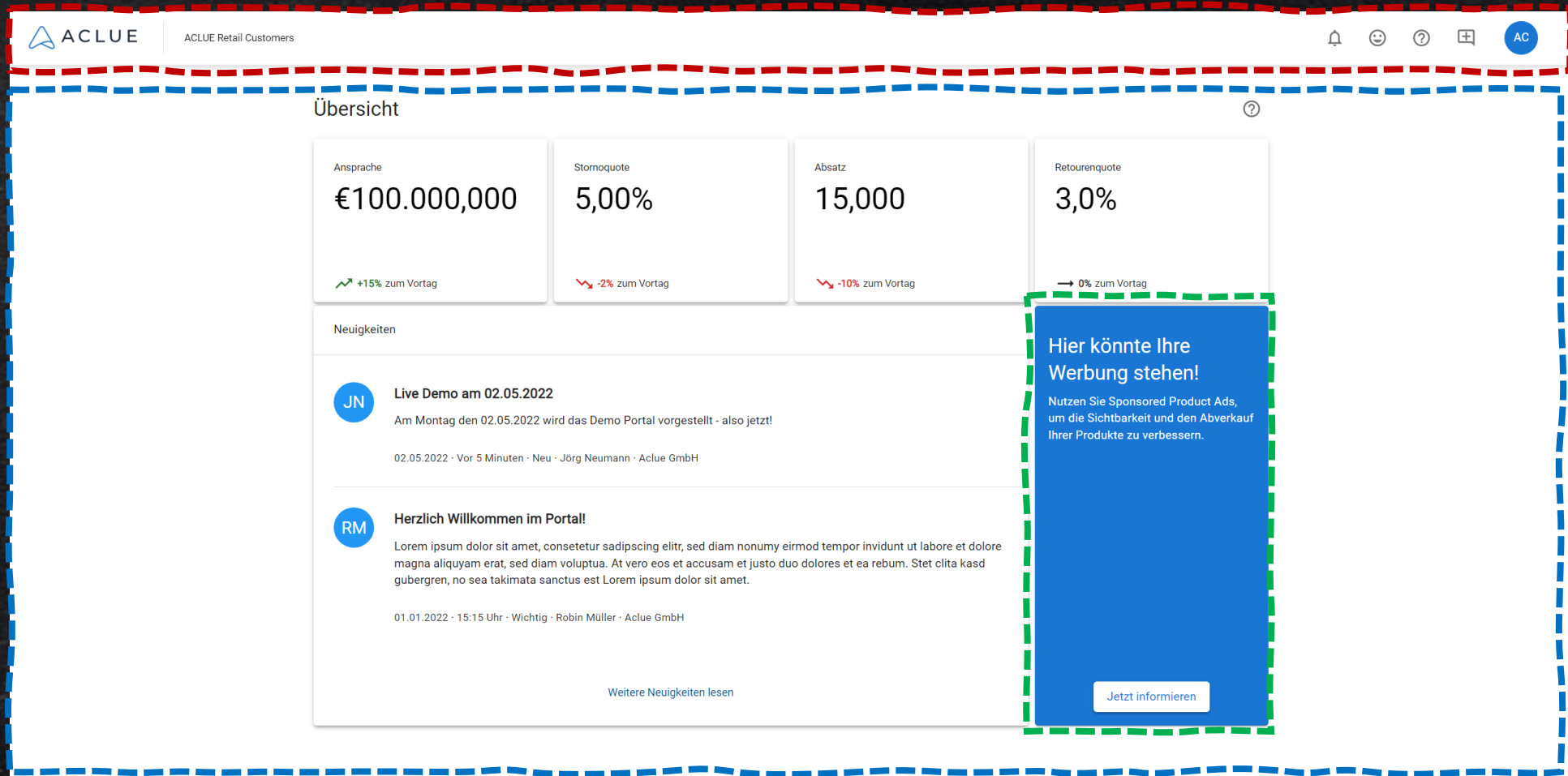
- \* Namenskonventionen wo keine absolute Isolation möglich
- \* Namespaces für CSS, Events, Local Storage, Cookies, ...


# PAGE COMPOSITION





# PAGE COMPOSITION



 TEAM HEADER

 TEAM DASHBOARD

 TEAM ADVERTISING

# PAGE COMPOSITION

```
<body>
```

```
  <red-header>
```

```
    <blue-dashboard>
```

```
      <green-adbox></green-adbox>
```

```
    </blue-dashboard>
```

```
  </red-header>
```

```
</body>
```

Team Header

Team Dashboard

Team Advertising



# PAGE COMPOSITION

## Server-side

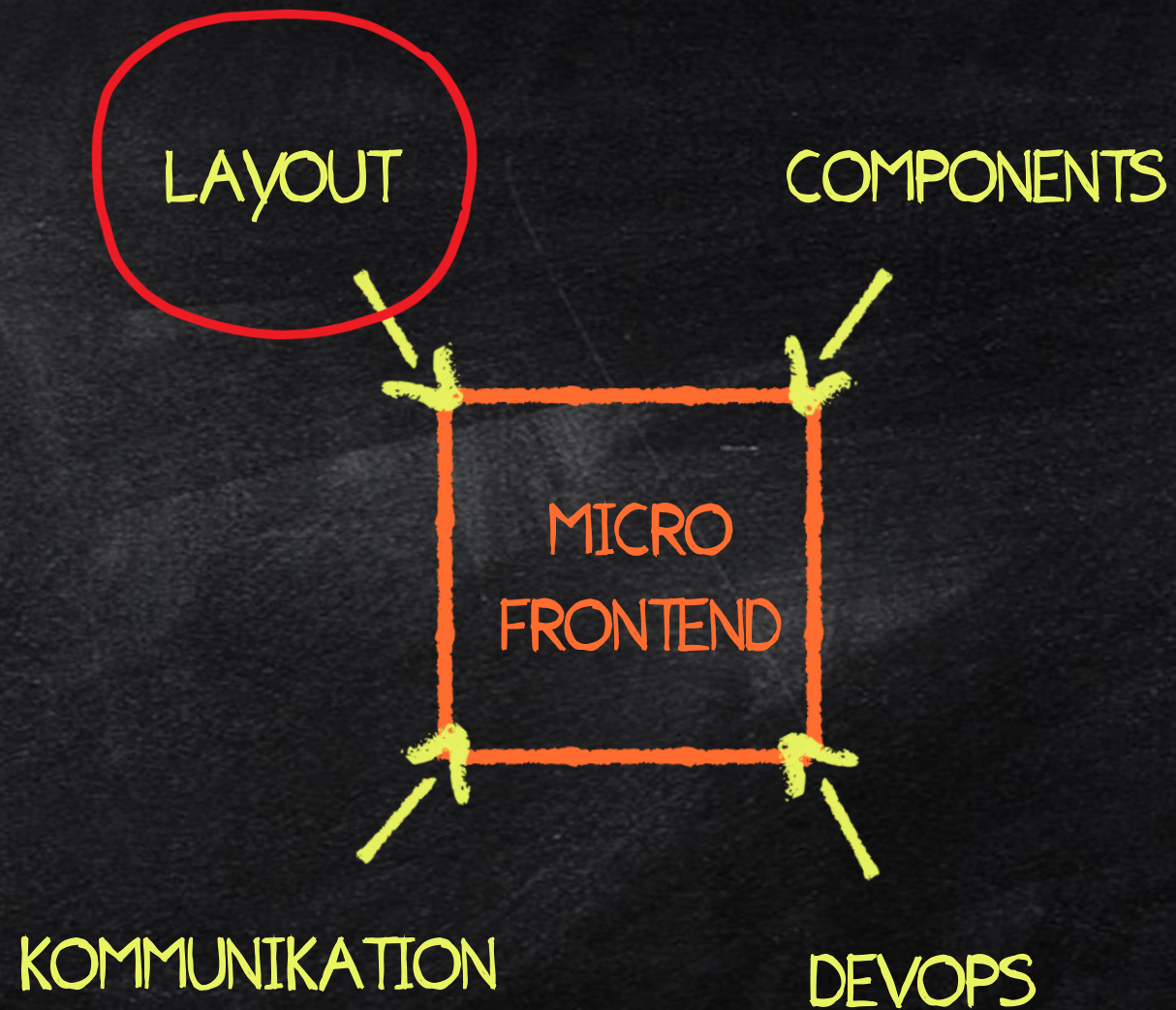
- \* Die App wird serverseitig zusammengebaut (z.B. mit Server-side Includes)
- \* Sinnvoll für die Integration von Legacy Apps

## Client-side

- \* Die App wird im Browser zusammengebaut (z.B. mit ECMA Script Modules)
- \* Sinnvoll für die Integration von SPAs

DEMO







# LAYOUT

**www.ARNGREN.net**

**el-Kjøretøy**    Teknologi & Gadgets

**el-biler til barn**  
(Fra 07.11.2004)  
< Frithjof i sin Fly-Bil

Search    [el-retur](#)  
[Slik Betaler du](#)

**el-Kjøretøy**

**el-ATV**

Akvarium  
Alarm  
Alkotester  
**ATV (el.)**  
Bildelenser  
**Bil, el-bil**  
Bilbane  
Biljardbord M/for  
Digital-Kikkert  
Disko-Lys  
DVD-Spiller  
**el-Bil-barn**  
**el-Bil-3-hjul**  
Elektronikk & DAB  
**el-Scooter & Bil**  
**el-Sykel-1, 2**  
**Fatbike-el**  
Figurer  
GoKart-Pedal, el  
**Golf-biler (m/skilt)**  
Hobby & RC  
Hoverpod  
HP-Måler (Bil)  
**Kikkert**  
**Kjøretøy-el**  
Kompass (Bil/Båt)  
Lader-230Vac  
**Luft-Jekk**  
Lykt  
**Moped-el**  
Motorsykel-Mini  
**Omformer**  
Oversetter  
**Gressklipper**  
Rakett-Fly  
RC Produkter

**el-fatbike 750w**  
12.998,-

**6.998,-**  
4WD  
12V  
24V  
48V  
fra kr. 1.798,-

**el-ATV til Barn & Voksne.**  
fra kr. 3998,-

**Nyhet!**  
**el-Sykel 1.599,-**  
**el-ATV**

**el-Bil ; Cross-Rider**  
fra kr. 89.998,-

**Fatbike-500w**  
**el-sykel med skivebrems**  
fra 9.008,-  
**Luft-iek (4,2 tonn)**

**3-hjul el-Cargo-Bike**  
**Biler til barn**

**4WD**  
**4WD Land Rover**  
3.999,-

**3-hjuls moped-bil CabEasy.**  
Godkjent for : 3-personer.  
Fart : 45km/t  
Range 70 km  
kr. 59.998,-

**el-bil (Sertifikat-fri)**  
3-hjul kr. 39.998,-  
**Torro**

**4WD**  
**3-hjuls el-cargobike Dansk**  
**RC Tank.**  
Metall 55 cm med Luftkanon

**el-fatbike Sammenleggbar**  
12.998,-  
**750w/48v**  
**el-sykel**  
7.998,-  
**Fatbike-1500w**  
16.998,-

**el-bil** kr. 79.998,-

**el-ATV**

**12.998,-**

**- Elektriske-Kjøretøy**  
**- Elektrisk-ATV - Roboter**  
**- el-biler til barn/ungdom**  
**- Solcelle-produkter**  
**- Forbruker Elektronikk**  
**- el-Sykel & el-Moped**  
**- Batterier & Ladere, etc**  
**- RC-produkt - Rakett**

**Stor 12V/24V med Gummihjul**  
2-seters  
4WD  
**4.998,-**

**Biljardbord, Spisebord i ett**  
kr. 5.998,-

**100km/t**  
**Elektrisk-ATV**  
6000 watt  
kr. 59.998,-  
**EL-Scooter**  
5000 watt  
kr. 29.998,-

**el-Jeep Willy-type,**  
med 60V/1500w motor.  
6 eller 40km/t kr. 39.998,-

**4WD, 3-seters Willy el-Jeep, med Gummihjul.**  
kr. 5998,-

**Styreenhet & Fordelere til Bil**  
fra kr. 2998,-

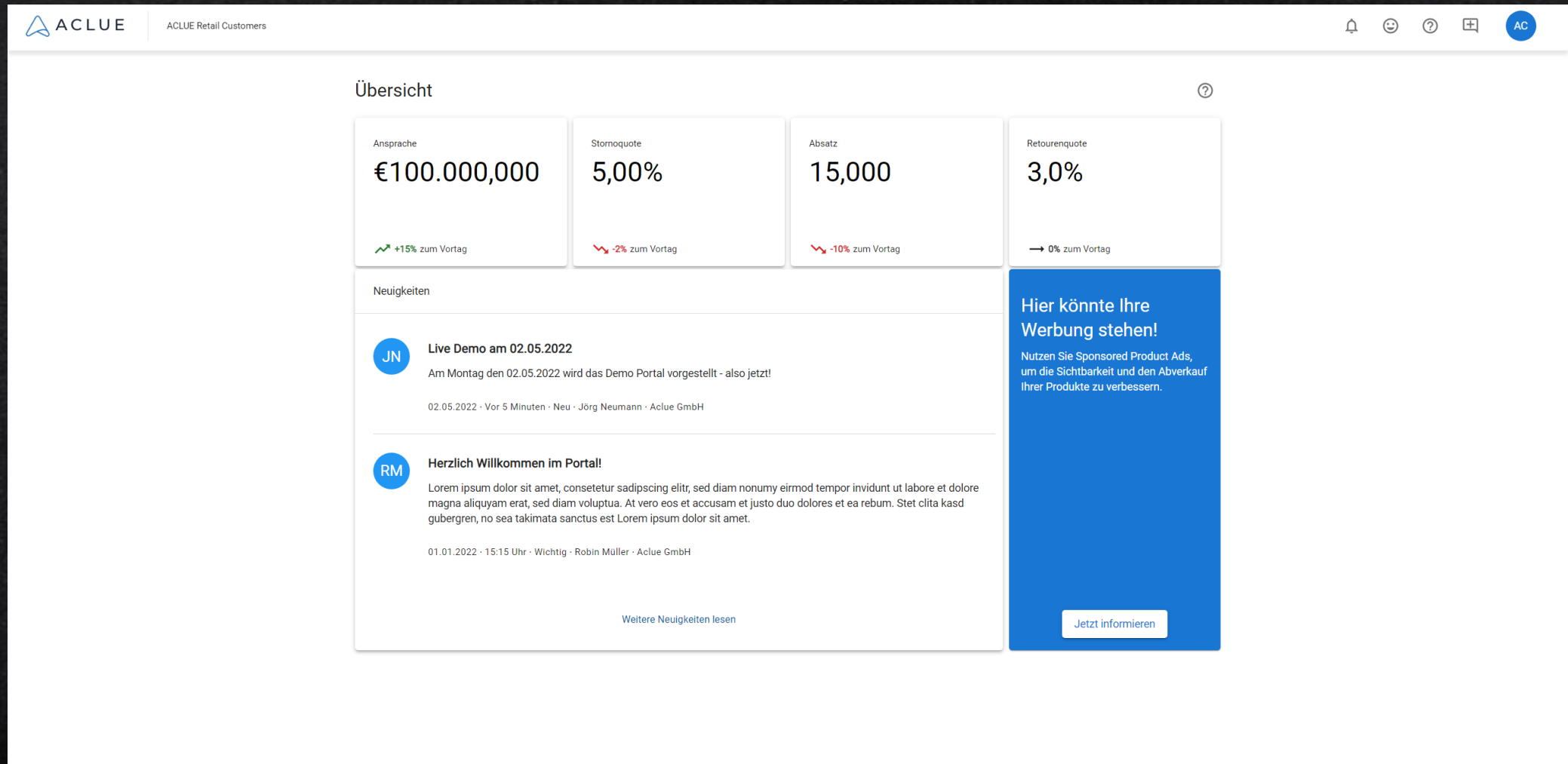
**el-Moped.** kr. 12.998,-

**el-Bil; Cor**  
2WD/4WD  
fra kr. 89.998,-

**Fotball-Trener**  
fra 2.598,-



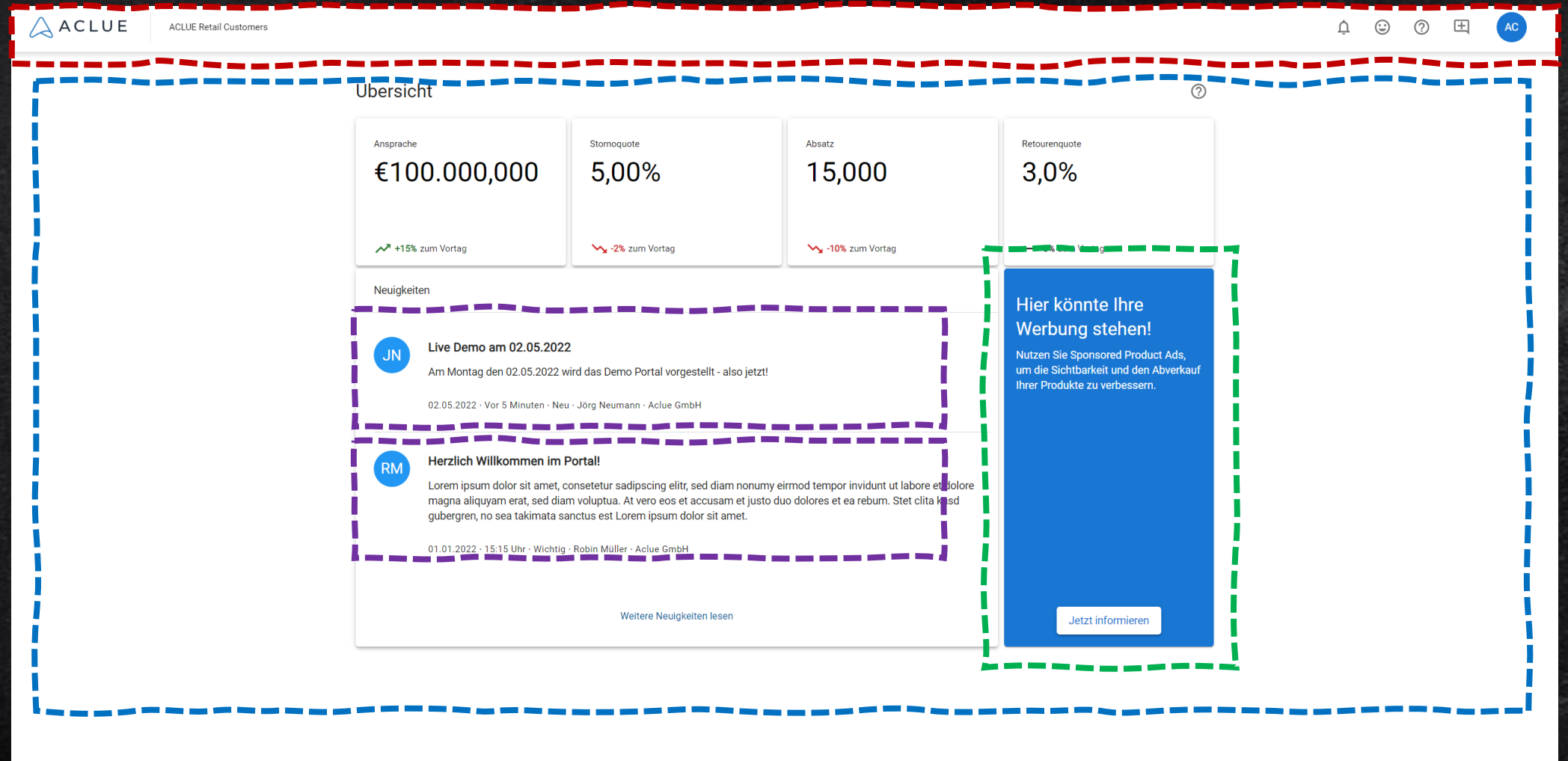
# LAYOUT



DEMO



# LAYOUT



# WAS WANN BENUTZEN?

## GRID LAYOUT

- Komposition von Containern
- Gesamtlayout einer Webseite
- Explizite Anordnung von Bereichen

## FLEX LAYOUT

- Komposition von Inhalten
- Positionierung von Container-Elementen
- Implizite Anordnung für variable Inhalte



# EIN LAYOUT FÜR ALLE DEVICES

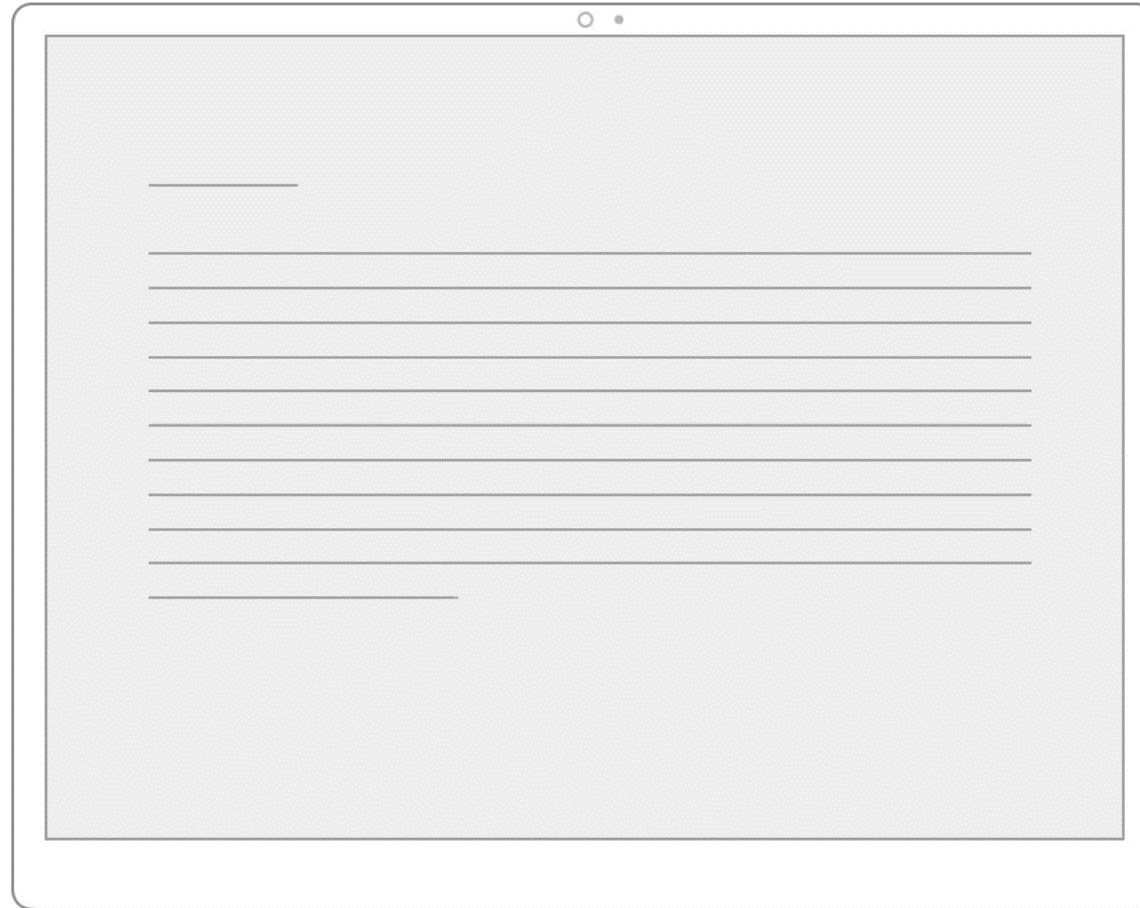


# RESPONSIVE DESIGN PATTERNS

- RESIZE
- REFLOW
- REPOSITION
- REDESIGN



# RESIZE



DEMO



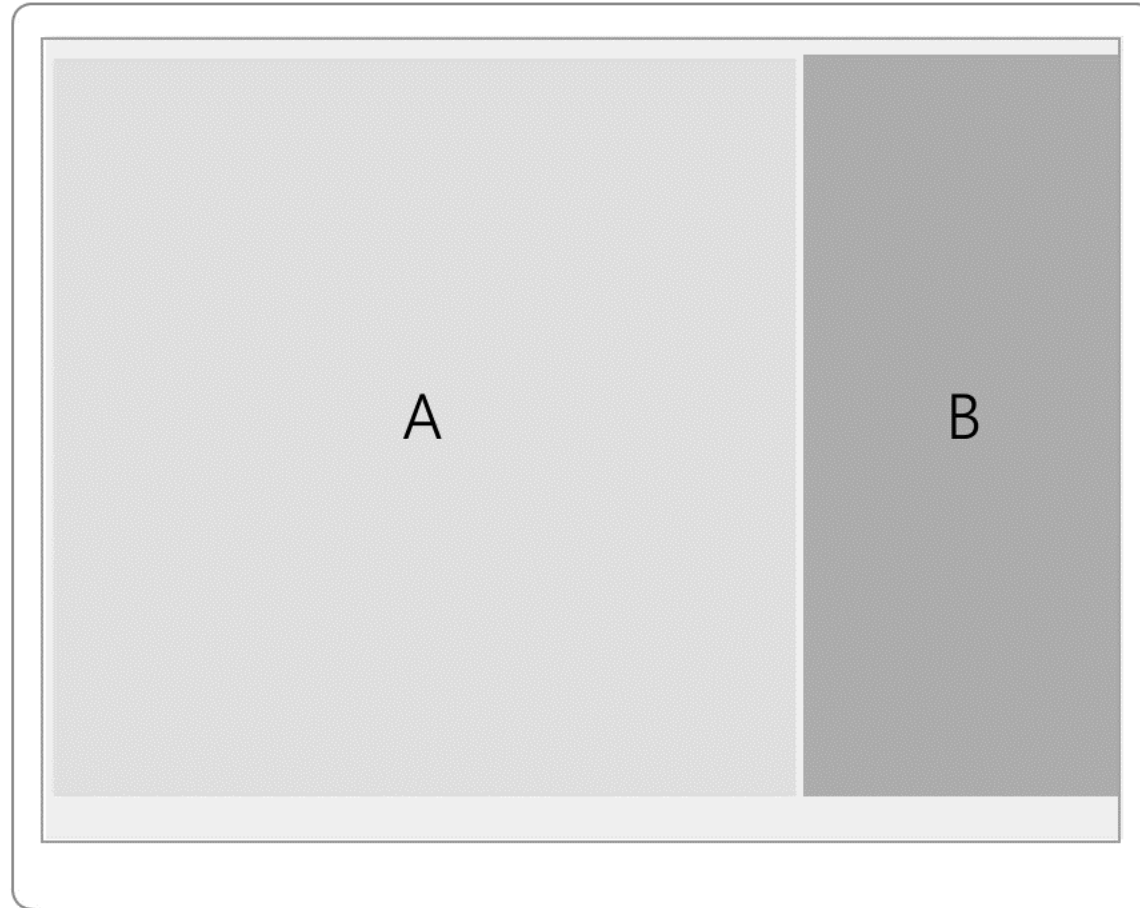
# REFLOW



DEMO



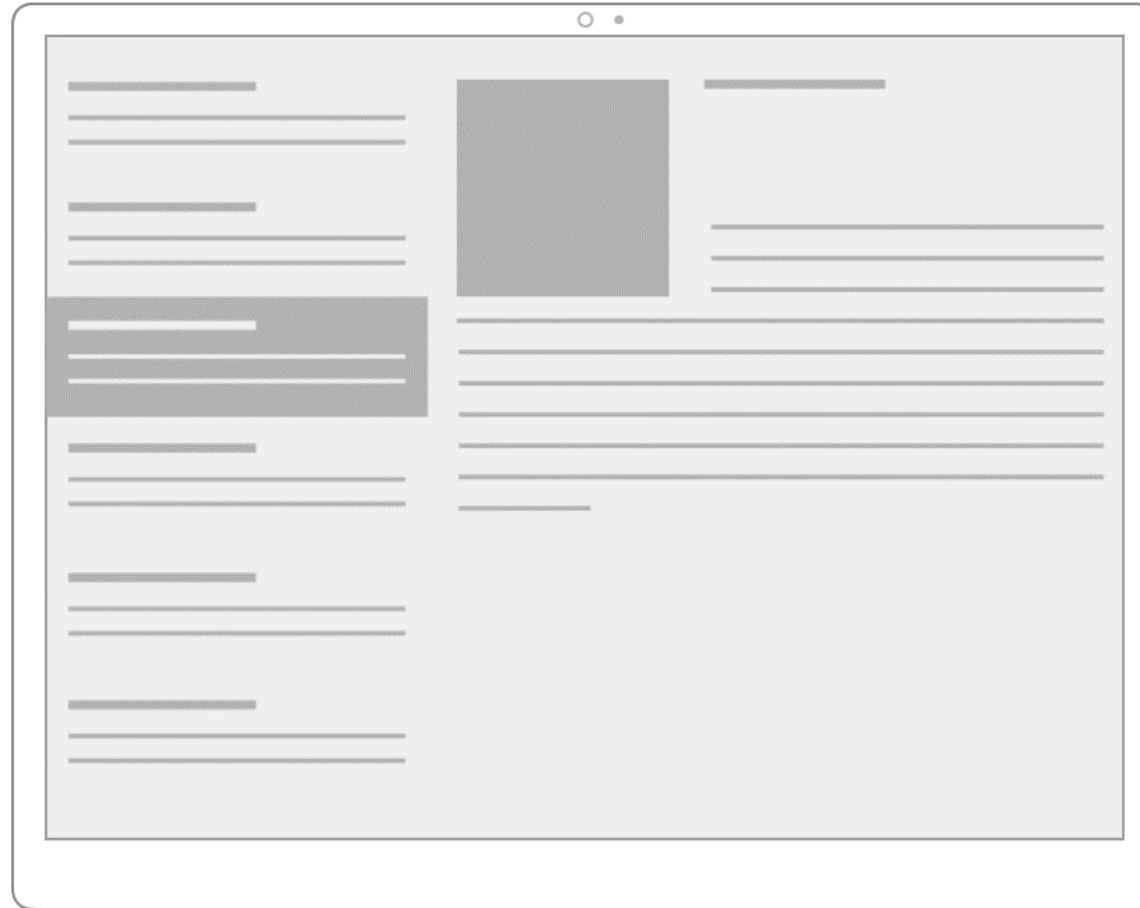
# REPOSITION



DEMO



# REDESIGN



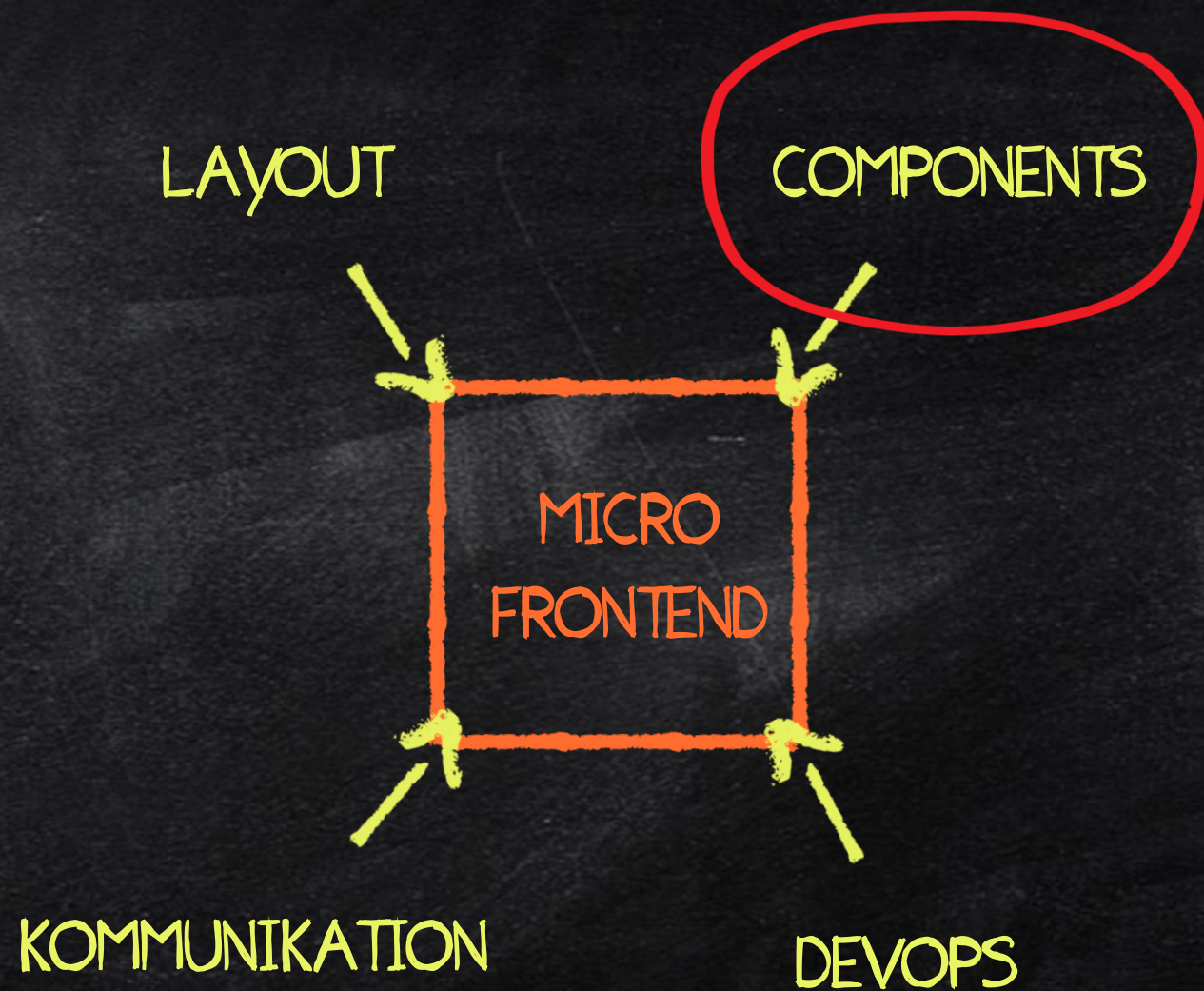
# CONTAINER QUERIES

- Tech Preview Feature in Chrome, Opera und Safari
- Für Chrome: "Enable CSS Container Queries" Feature Flag unter chrome://flags

Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
					12.5				
					14.4				
					14.8				
		1 101			15.3				
102	101	1 102	15.4		15.4				
103	102	1 103	15.5	1 86	15.5	all	103	12.12	17.0
	103	1 104	16.0	1 87	16.0				
	104	1 105	TP						
		1 106							



DEMO





# COMPONENTS

```
<html>
  <head>
    <script src="fancyFramework.js"></script>
  </head>
  <body>
    <div id="fancyWidget"></div>
  </body>
</html>
```

- Keine Semantik
- Intransparenz
- Anpassungen schwierig
- Framework-Abhängigkeit

```
<html>
  <head>
    <script type="module" src="NumberPad.js"></script>
  </head>
  <body>
    <number-pad value="1" minimum="1" maximum="1">
  </number-pad>
</body>
</html>
```

- Semantik
- Local Scoping
- Konfiguration
- Bundle Import



## CUSTOM ELEMENTS

Eine API zur Registrierung von eigenen HTML Elementen

## HTML TEMPLATES

UI-Vorlage für eine Komponente

# WEB COMPONENTS

## SHADOW DOM

Kapselung der inneren DOM-Bestandteile eines Elements

## ES MODULES

Einbinden von Komponenten

DEMO



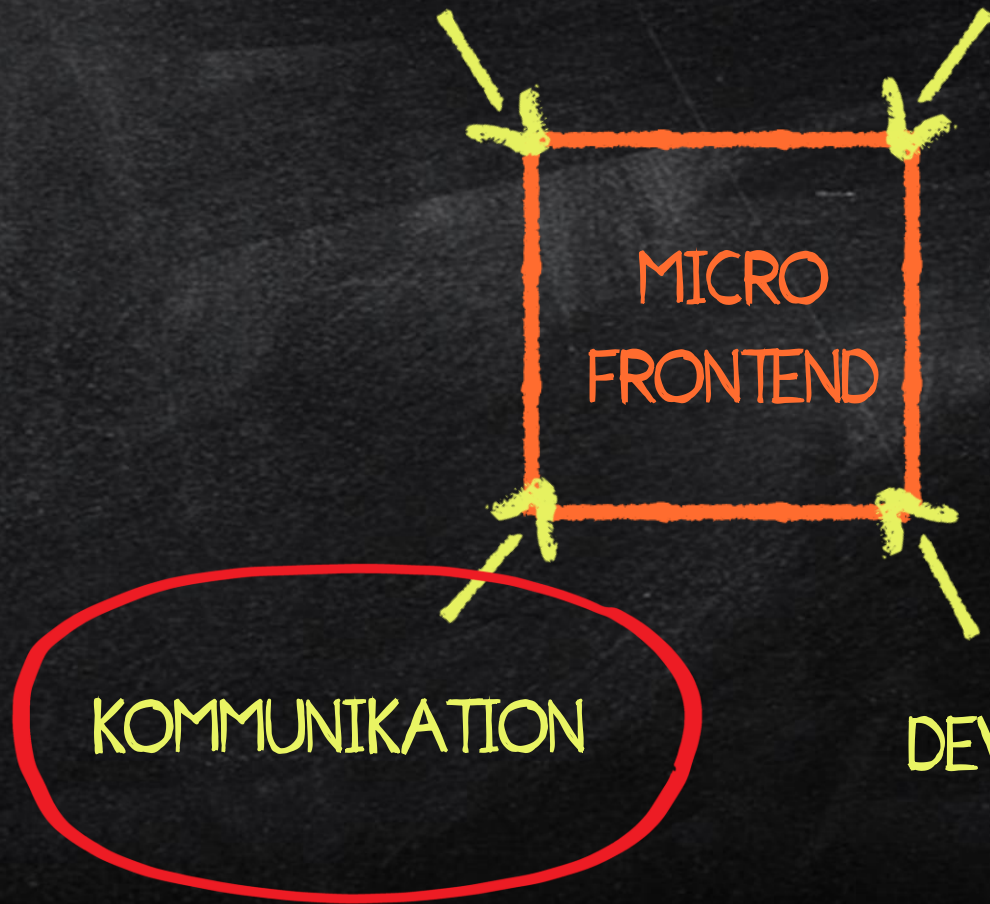
COMPONENTS

LAYOUT

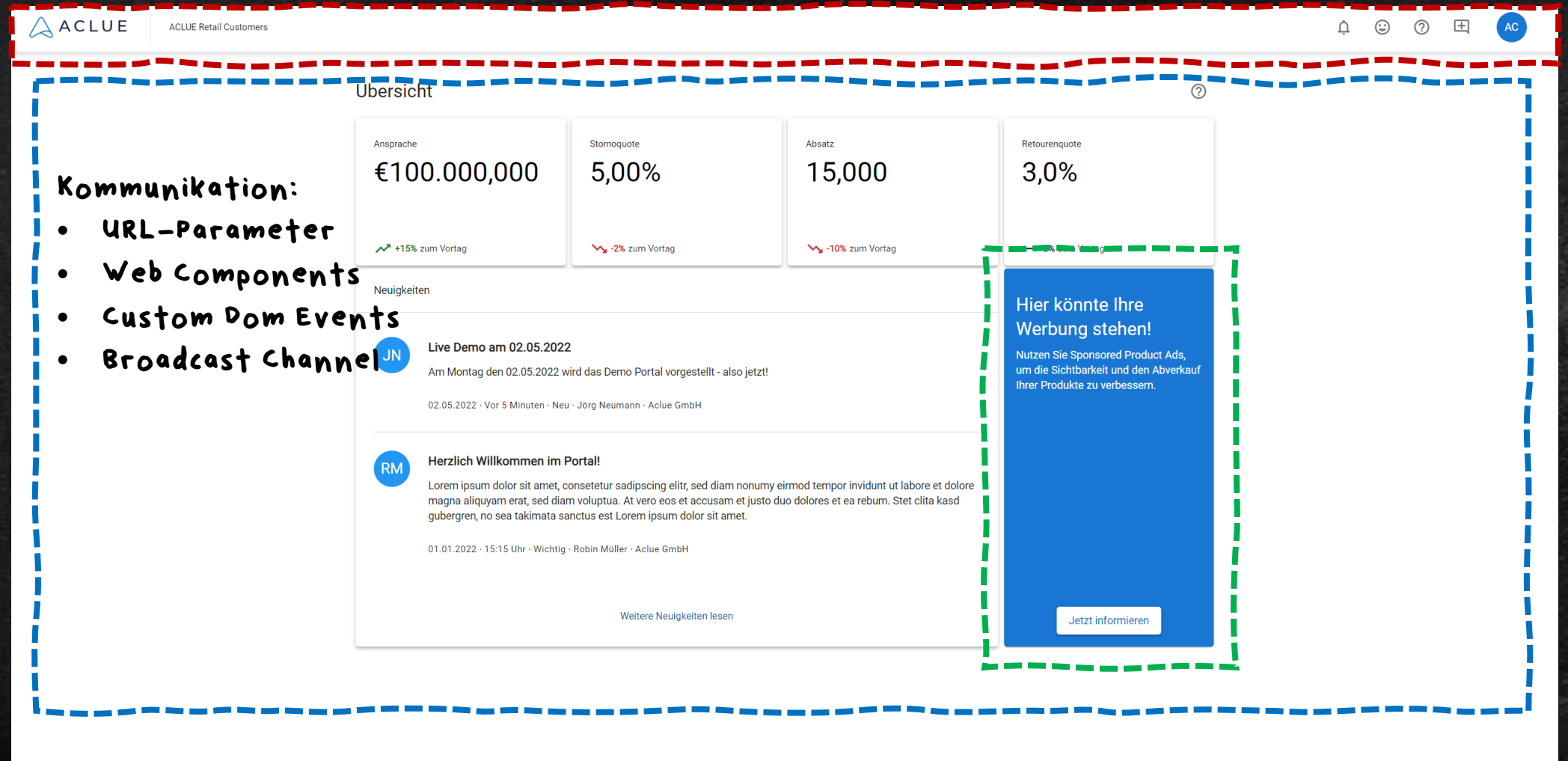
MICRO  
FRONTEND

KOMMUNIKATION

DEVOPS



# KOMMUNIKATION



TEAM HEADER



TEAM DASHBOARD



TEAM ADVERTISING



DEMO

COMPONENTS

LAYOUT



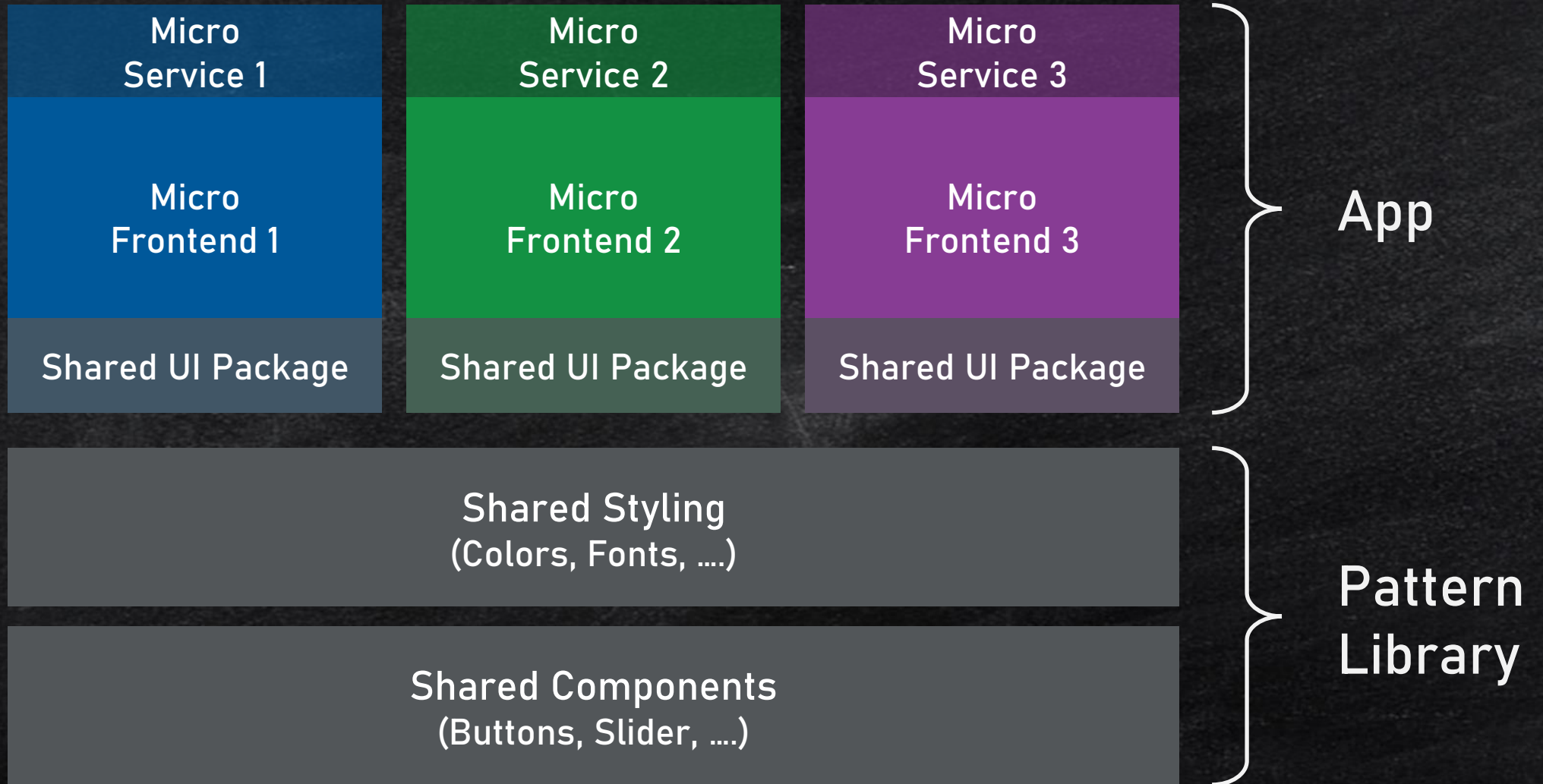
MICRO  
FRONTEND

KOMMUNIKATION

DEVOPS



# DEVOPS



# WAS WIRD BENÖTIGT?

## ZENTRALES REPOSITORY

- Wiederverwendbare Elemente (Komponenten, Styles, Assets, ...)
- Wer entwickelt neue Elemente?
- Wie ist der Prozess?

## VERFÜGBARKEIT

- Bundling als Package
- Versionierung
- Releases

## DOKUMENTATION

- Testing
- Abnahme durch UI/UX
- Guidelines für Entwickler



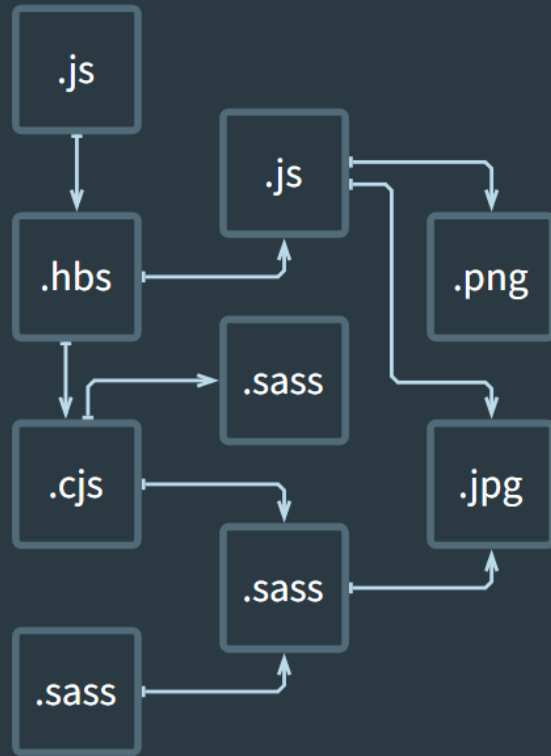
DEMO

# BUNDLING

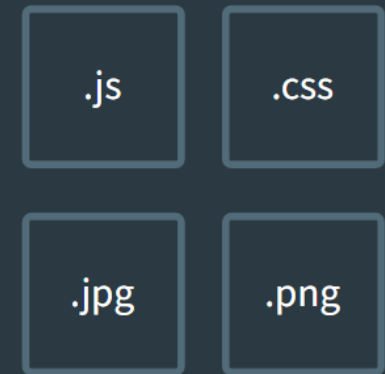
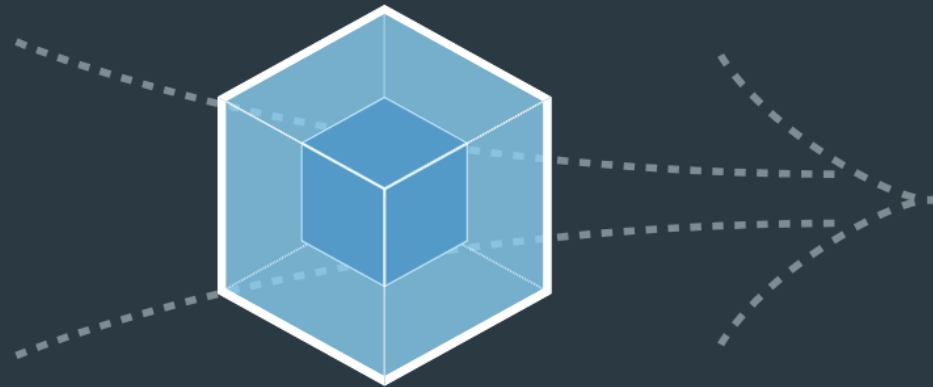
- In ein einzelnes JS-File bundeln
  - Als ECMAScript Module
  - Einbinden per Script-Tag
  - Wird zur Laufzeit der App hinzugefügt
- In mehreren Files bundeln
  - Als NPM Package publishen
  - Einbinden als versioniertes Package
  - Wird beim Kompilieren der App hinzugefügt
  - Erlaubt Tree-shaking



# BUNDLING



MODULES WITH DEPENDENCIES



STATIC ASSETS



# WEBPACK

- Meistgenutzter Bundler im Einsatz bei Create React App, Angular CLI und weiteren Frameworks
- Wenig out-of-the-box - dafür weitreichenden Support durch Plugins (macht die Config sehr komplex)
- Sehr langer Buildprozess

```
// webpack.config.js
module.exports = {
  entry: {
    "bundle.js": __dirname + "/src/index.tsx",
  },
  output: {
    filename: "minified.js",
  },
  module: {
    rules: [
      {
        test: /\.tsx?$/,
        loader: 'ts-loader',
        options: {
          configFile: "tsconfig.build.json",
        },
        exclude: [/node_modules/, /local/],
      },
      {
        test: /\.s?css$/,
        use: ["style-loader", "css-loader", "sass-
loader"],
      },
      {
        test: /\.svg/,
        loader: "svg-url-loader",
      },
      {
        test: /\.(png|jpg|gif)$/i,
        loader: 'url-loader',
      }
    ],
  },
  resolve: {
    extensions: [
      ".ts", ".tsx", ".s?css", ".js", ".jsx"
    ],
  }
}
```



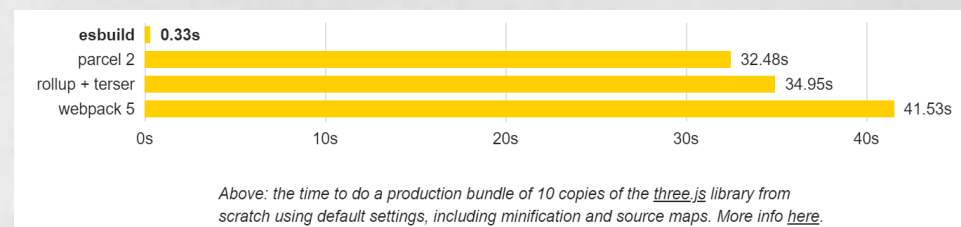
DEMO

# ESBUILD

- 10x-100x schneller als andere Buildtools wie Webpack, Parcel oder Rollup\*

\* <https://esbuild.github.io/faq/#why-is-esbuild-fast>

- Nativer Support für ES6, TypeScript, JSX und mehr
- Noch sehr neu (Stand April 2022 in v0.14), gewinnt aber immer mehr Aufmerksamkeit



```
// build.js
require('esbuild').build({
  entryPoints: ['/src/index.tsx'],
  bundle: true,
  loader: {
    /\.html/: "text",
    /\.svg/: "dataurl",
    /\..(png|jpg|gif)$/i: "dataurl",
  },
  outfile: 'minified.js',
}).catch(() => process.exit(1))
```



DEMO

# PATTERN LIBRARY VERFÜGBAR MACHEN



## NPM

- Öffentliche Packages sind kostenlos, private Packages aber kostenpflichtig
- Über [npmjs.com](https://npmjs.com) gehosted



## NEXUS REPOSITORY

- Kostenpflichtige Lösung von [Sonatype](https://sonatype.com)
- Self-Hosted, nur für organisationsinterne Packages geeignet



## GITHUB PACKAGES

- Öffentliche und private\* Packages sind kostenlos (\*mit Storage-Grenze)
- Über [github.com](https://github.com) gehosted

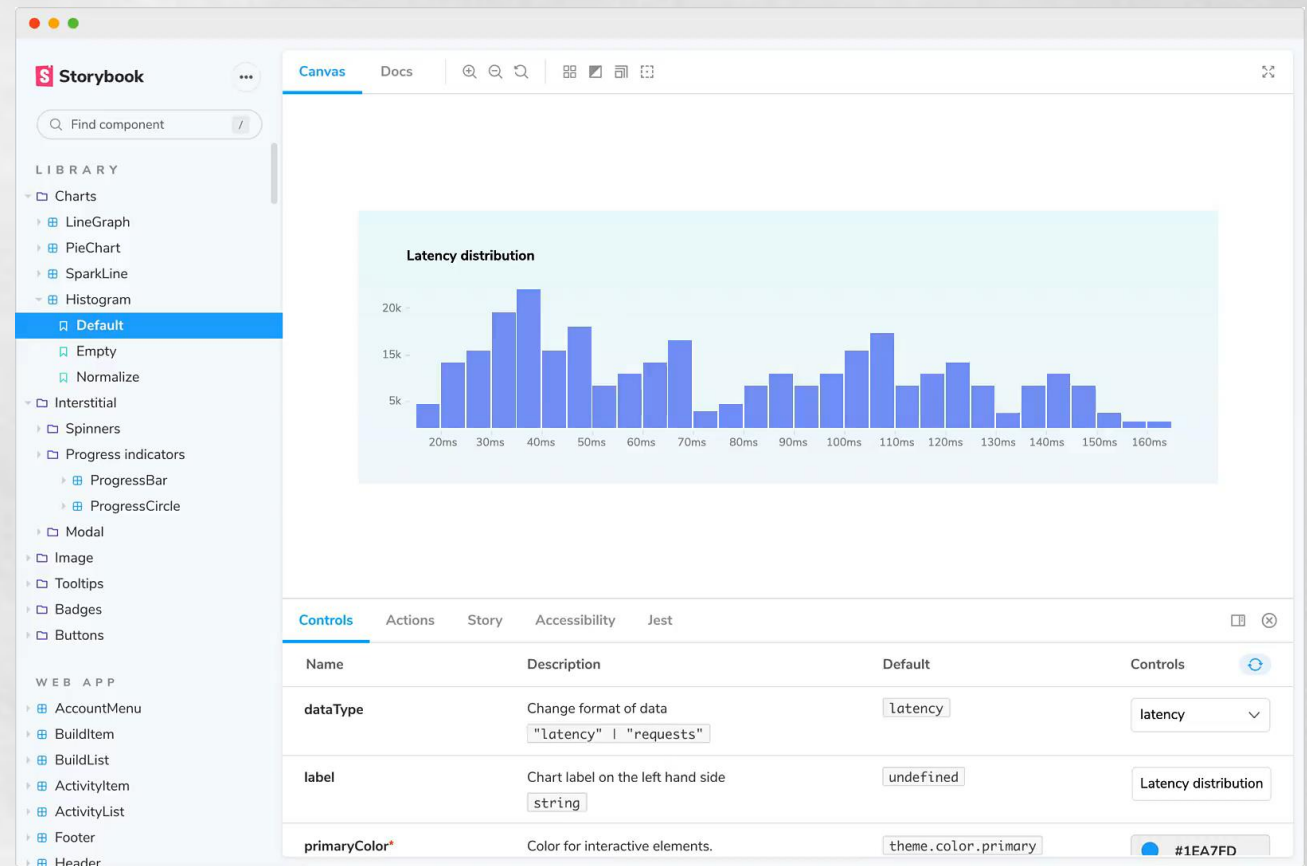


DEMO

# DOKUMENTATION



- Open Source Tool für alle gängigen Frameworks
- Interaktive Stories für isolierter UI Komponenten
- Hilfreich für
  - Entwickeln
  - Testen
  - Dokumentieren



<https://storybook.js.org/>

DEMO



# FAZIT

## PATTERN LIBRARIES SPAREN WIEDERHOLTEN IMPLEMENTIERUNGS-AUFWAND

- Einheitliche Implementierung eines Style Guides
- Erleichtert die Zusammenarbeit zwischen Designern und Entwicklern, sowie zwischen Entwicklungsteams
- Zentraler Ablageort für gemeinsam UI-Fragmente (Komponenten, Styles, Assets, ...)
- Erleichtere die Nutzung von Elementen durch ein Dokumentationstool (z.B. Storybook)



# FAZIT

- Micro Frontends fördern die Skalierung von Projekten durch vertikale Aufteilung
- NPM Module helfen beim Paketieren und Verteilen der Assets
- Flex und Grid helfen bei der Umsetzung eines Responsive Layouts
- Mit Web Components können die Bereiche der App gekapselt und wiederverwendet werden
- Mit Pattern Libraries können wiederverwendbare Elemente teamübergreifend verfügbar gemacht werden
- Storybook hilft zur Dokumentation einer Pattern Library



Q

&

A