

# ANGULAR VS REACT

- David Derichs
- Mohamed Haridy
- Catherine Mamani Valles
- Oskar Schachtschneider

# INHALTSVERZEICHNIS

- Single Page Application
- Angular
- React
- Unterschiede zwischen Angular & React
- Live Coding
- Fazit

# **SINGLE PAGE APPLICATION**

# **SINGLE PAGE APPLICATION**

Server liefert eine HTML aus

# SINGLE PAGE APPLICATION

Server liefert eine HTML aus

DOM wird jeweils aktualisiert

# SINGLE PAGE APPLICATION

Server liefert eine HTML aus

DOM wird jeweils aktualisiert

keine Verbindungsprobleme

# **SINGLE PAGE APPLICATION**

Offline-friendly

Bessere Userexperience

# VIELE UNTERNEHMEN BENUTZEN DIESE

- Google
- Amazon
- Facebook
- Microsoft

# ANGULAR





# KEY FACTS



# KEY FACTS

Web-Frontend Framework



# KEY FACTS

Web-Frontend Framework

MVC



# KEY FACTS

Web-Frontend Framework

MVC

Open Source - MIT Licence



# KEY FACTS

Web-Frontend Framework

MVC

Open Source - MIT Licence

Angeführt von Google



# KEY FACTS

Web-Frontend Framework

MVC

Open Source - MIT Licence

Angeführt von Google

TypeScript (Klassen, Typen, Schleifen ...)



# RELEVANZ

● **Vue.js**  
Suchbegriff

● **React**  
Suchbegriff

● **Angular**  
Suchbegriff

+ Vergleich hinzufügen

Weltweit ▾

Letzte 12 Monate ▾

Programmieren ▾

Websuche ▾

Interesse im zeitlichen Verlauf ?





**USP**



# USP

Plattformen (Web, Mobile, Desktop)



# USP

Plattformen (Web, Mobile, Desktop)

Performance durch RxJS



# USP

Plattformen (Web, Mobile, Desktop)

Performance durch RxJS

Templating



# USP

Plattformen (Web, Mobile, Desktop)

Performance durch RxJS

Templating

Dependency & Package Management



# EVENT BINDING

```
<button (click)="onClickMe()">Click me!</button>
```

```
@Component({
  selector: 'app-click-me',
  template: `
    <button (click)="onClickMe()">Click me!</button>
    {{clickMessage}}`
})
export class ClickMeComponent {
  clickMessage = '';

  onClickMe() {
    this.clickMessage = 'You are my hero!';
  }
}
```



# DATA BINDINGS

```
<element [ngclass]="{'aktiv': this.data.aktiv }">...</element>  
  
<button (click)="toggle()"></button>  
  
<element *ngif="data.aktiv"> {{ beispieltext }}  
                                </element">
```

```
export class AppComponent {  
  data = {  
    aktiv: true;  
  }  
  text = 'beispieltext';  
  toggle(){  
    this.data.aktiv = false;  
  }  
}
```

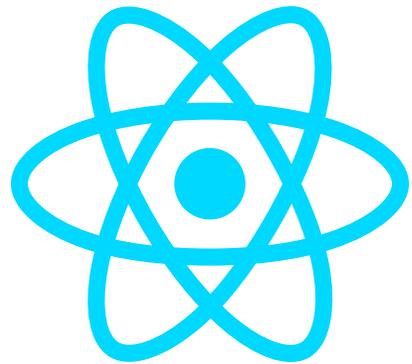


# SCHLEIFEN

```
<li *ngfor="let movie of movies">
  {{ movie }}
</li>
```

```
export class AppComponent {
  movies = ['Captain America', 'Iron Man', 'Hulk'];
  singleMovie = this.movies[0];
}
```

# REACT



React

**WAS IST REACT**

# **WAS IST REACT**

UI Javascript Framework

# **WAS IST REACT**

UI Javascript Framework

V in MVC

# **WAS IST REACT**

UI Javascript Framework

V in MVC

SINGLE PAGE APPLICATION (SPA)

# WAS IST REACT

UI Javascript Framework

V in MVC

SINGLE PAGE APPLICATION (SPA)

Entwickelt von facebook 2011

# WAS IST REACT

UI Javascript Framework

V in MVC

SINGLE PAGE APPLICATION (SPA)

Entwickelt von facebook 2011

Apache License 2.0 2013

# WAS IST REACT

UI Javascript Framework

V in MVC

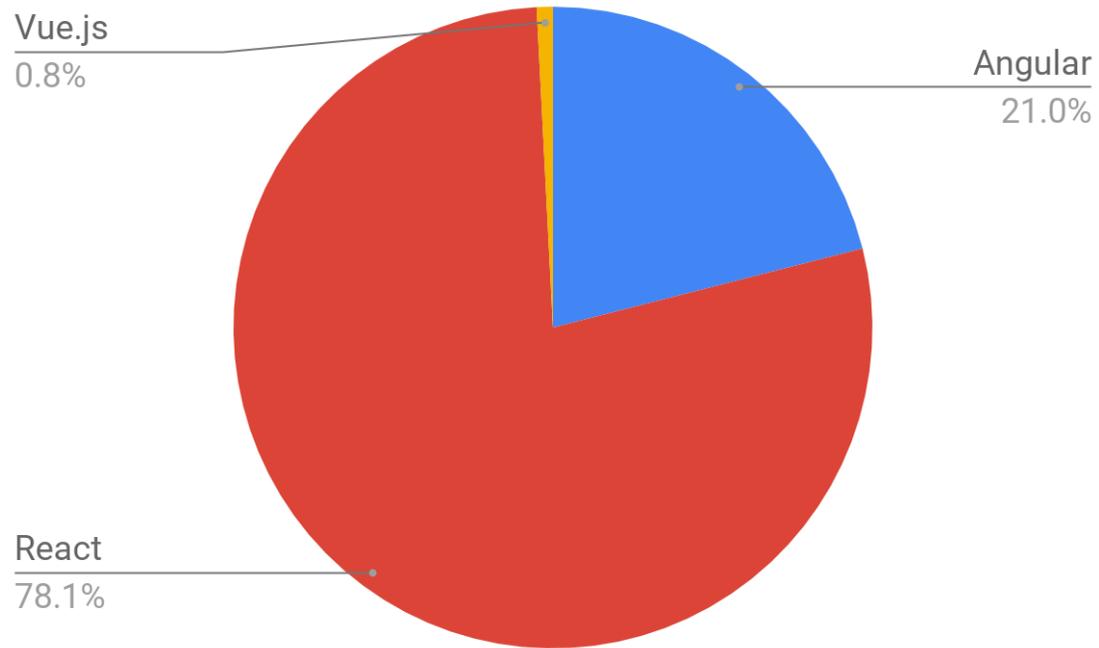
SINGLE PAGE APPLICATION (SPA)

Entwickelt von facebook 2011

Apache License 2.0 2013

MIT license 2017

# BRAUCHE ICH REACT 🤔



# WER BENUTZT REACT

# WER BENUTZT REACT



# WER BENUTZT REACT



**Dropbox**

# WER BENUTZT REACT



Dropbox

**NETFLIX**

# WER BENUTZT REACT



Dropbox

**NETFLIX**



zalando

# WER BENUTZT REACT



Dropbox

**NETFLIX**



zalando



# WER BENUTZT REACT



Dropbox

**NETFLIX**



zalando



# WER BENUTZT REACT



Dropbox

**NETFLIX**



zalando



**ATLASSIAN**

# WER BENUTZT REACT



Dropbox

**NETFLIX**



zalando



**ATLASSIAN**



Jira Software

# WER BENUTZT REACT



Dropbox

NETFLIX



zalando



ATLASSIAN



Jira Software

ATLASSIAN



Bitbucket

# WER BENUTZT REACT



Dropbox

NETFLIX



zalando



ATLASSIAN



Jira Software

ATLASSIAN  
Bitbucket

Trello

# WER BENUTZT REACT



Dropbox

NETFLIX



zalando



ATLASSIAN



Jira Software

ATLASSIAN  
Bitbucket

Trello

codecademy

**USP**

# USP

Components

# USP

Components

JSX

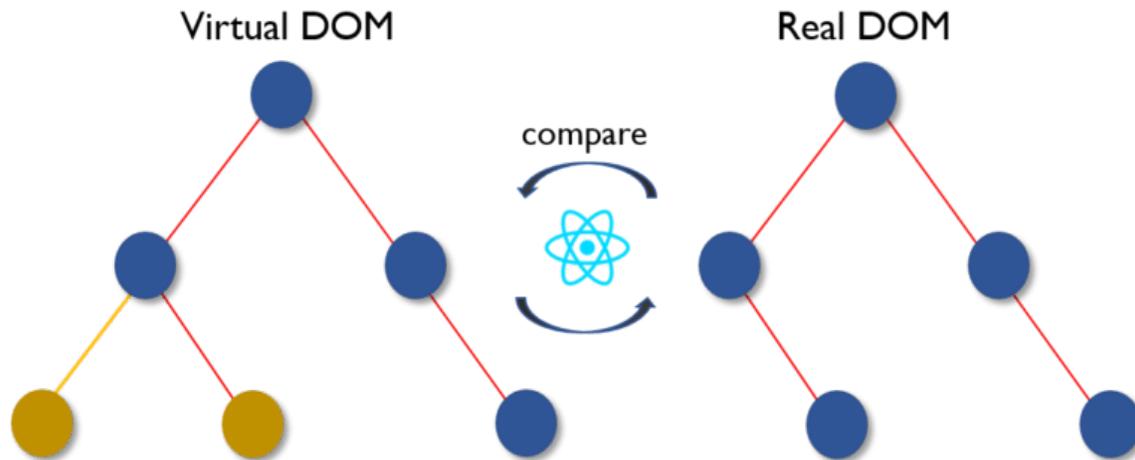
# USP

Components

JSX

virtual DOM

# VIRTUAL DOM



# EVENT BINDING

```
<button onclick="{this.handleClick}">Clik me!</button>
```

```
handleclick = (num) => {console.log(num)}
```

# DATA BINDING

```
state = {  
  defaultNumber: 1,  
  selectedCounter: 0  
}
```

```
{this.setState({ selectedNumber: 5 })}
```

# SCHLEIFEN

```
let list = [1, 2, 3, 4, 5, 6]
```

```
list.map((number) => {  
  console.log(number)  
  return (  
    <react.fragment>  
      <button onclick="{this.handleClick}">Click me!</button>  
      {this.setState({ selectedNumber: 5 })}  
      <div> {number}</div>  
    </react.fragment>  
  )  
})
```

# **UNTERSCHIEDE ANGULAR & REACT**

Philosophie

Zielgruppe

# PHILOSOPHIE

**Angular**

**React**

---

Framework

Library

---

erweitertete HTML  
version

JSX, eine XML-artige  
Sprache

# ZIELGRUPPE

Angular

React

---

Enterprise Orientiert

Web Entwickler Orientiert

## Angular

---

Skalierte Business  
Anwendungen

---

MVVM Paradigma(MVC)

## React

---

Für komplexe User  
Interaktionen

---

Mehr V als MVC

**ANGULAR**

# ANGULAR

ist ein Framework

# ANGULAR

ist ein Framework  
eingebautes Routing

# ANGULAR

ist ein Framework

eingebautes Routing

eingebautes State management

# ANGULAR

ist ein Framework

eingebautes Routing

eingebautes State management

two-way binding

**ANGULAR**

# ANGULAR

shadow DOM

# ANGULAR

shadow DOM

mit Tests im Sinn gebaut

# ANGULAR

shadow DOM

mit Tests im Sinn gebaut

Native Script

# ANGULAR

shadow DOM

mit Tests im Sinn gebaut

Native Script

Ionic

**REACT**

# REACT

ist eine Library

# REACT

ist eine Library

erweiterbar durch andere Libraries

# REACT

ist eine Library

erweiterbar durch andere Libraries

Komponenten Basiert

# REACT

ist eine Library

erweiterbar durch andere Libraries

Komponenten Basiert

kein two-way binding

**REACT**

# REACT

Virtual DOM

# REACT

Virtual DOM

Komponenten initial virtual gerendert, nur bei Änderung neu

# REACT

Virtual DOM

Komponenten initial virtual gerendert, nur bei Änderung neu

Separation zwischen immutable und mutable state

# REACT

Virtual DOM

Komponenten initial virtual gerendert, nur bei Änderung neu

Separation zwischen immutable und mutable state

React kann leicht zu existierenden Applikation hinzugefügt werden

**LIVE BEISPIEL**

**FAZIT**



beide sind gut

beide sind gut

kommt auf use case an

beide sind gut  
kommt auf use case an  
größe von Applikation

**DANKE FÜR EURE  
AUFMERKSAMKEIT**

# LINKS

- <https://reactjs.org/docs/getting-started.html>
- <https://angular.io/guide/quickstart>