MIT REACT

UI-KOMPONENTEN ENTWICKELN

NILS HARTMANN | W-JAX 2015

$$\underbrace{\text{NILS}\underbrace{\text{@}\underbrace{\text{NILSHARTMANN}}_{\text{GitHub}}}_{\text{Twitter}}\text{.NET}}_{\text{E-Mail}}$$

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

React

SINGLE PAGE APPLICATIONS

React

OPEN SOURCE VON FACEBOOK

HTTPS://FACEBOOK.GITHUB.IO/REACT/

React

0.3

05 | 2013 Open Source

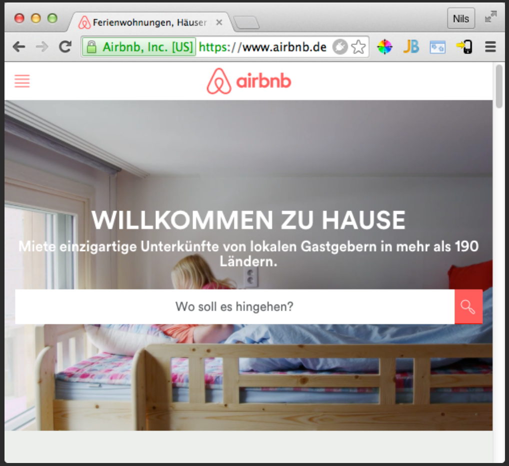
0.14.1

rc
beta
alpha

10 | 2015 Aktuelles Release

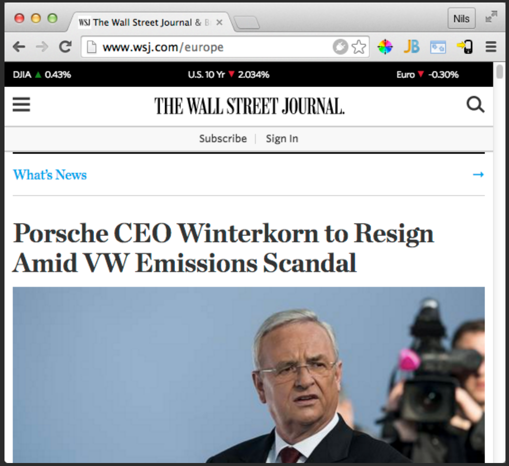
„Major“-Releases

BUILT WITH REACT

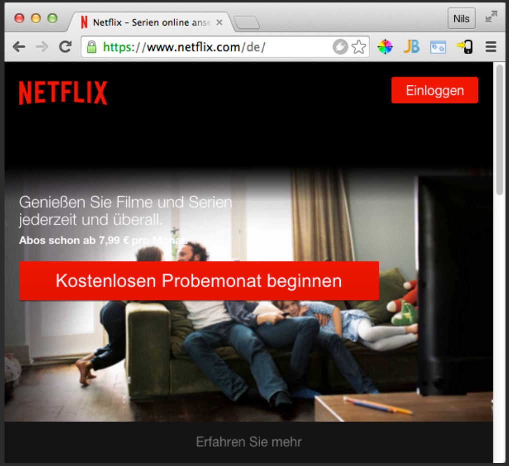


WILLKOMMEN ZU HAUSE

Miete einzigartige Unterkünfte von lokalen Gastgebern in mehr als 190 Ländern.

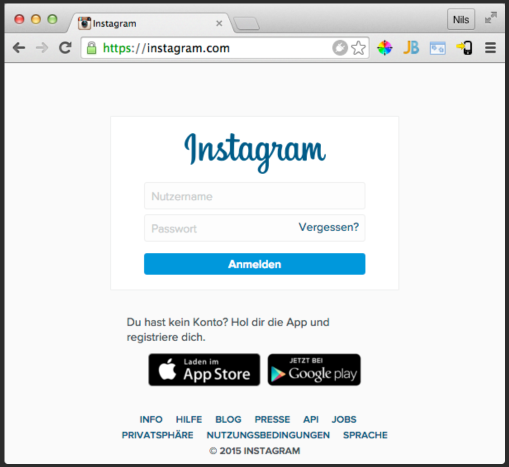


Porsche CEO Winterkorn to Resign Amid VW Emissions Scandal


Netflix – Serien online ans
https://www.netflix.com/de/
NETFLIX
Einloggen
Kostenlosen Probemonat beginnen
Erfahren Sie mehr

Genießen Sie Filme und Serien jederzeit und überall.

Abos schon ab 7,99 € pro Monat



Du hast kein Konto? Hol dir die App und registriere dich.

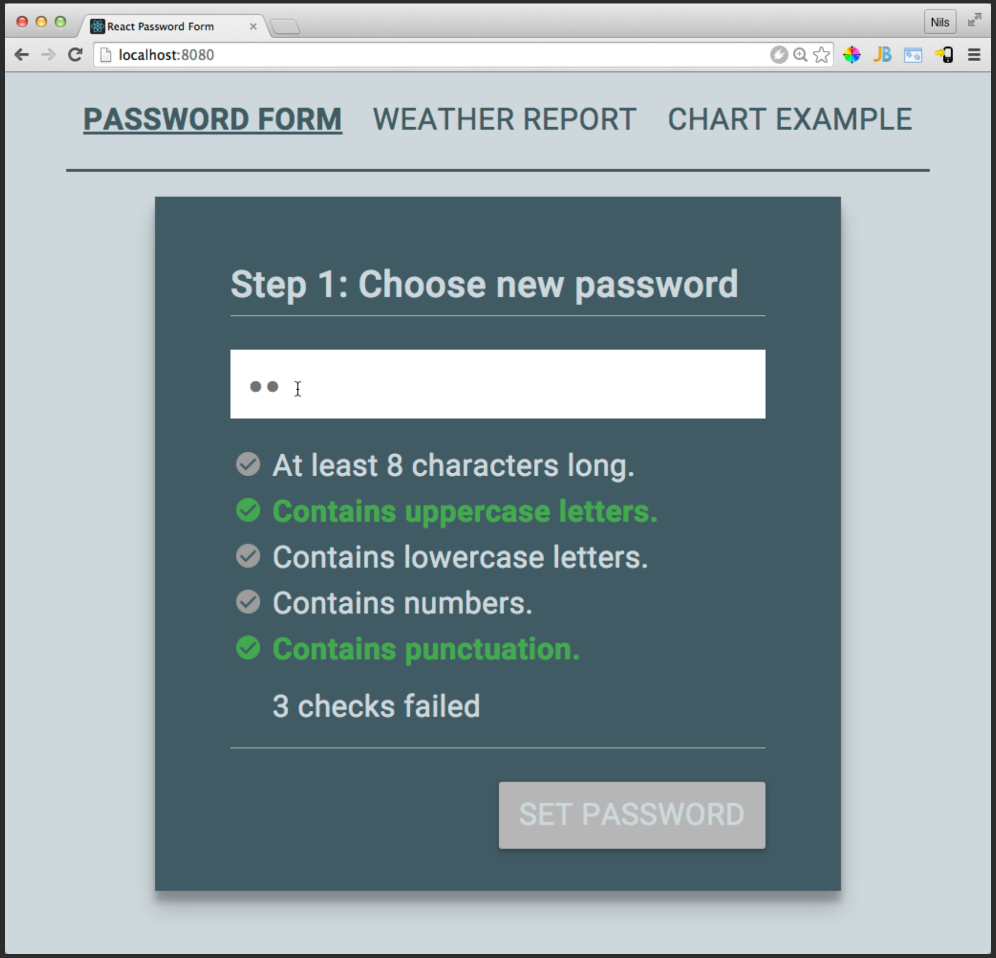
Facebook ermöglicht es dir, mit den Menschen in deinem Leben in Verbindung zu treten und Inhalte mit diesen zu teilen.

Stunning Photo Websites

For You, Your Family, Or Your Business.

V in MVC

W-JAX DEMO ANWENDUNG



Step 1: Choose new password

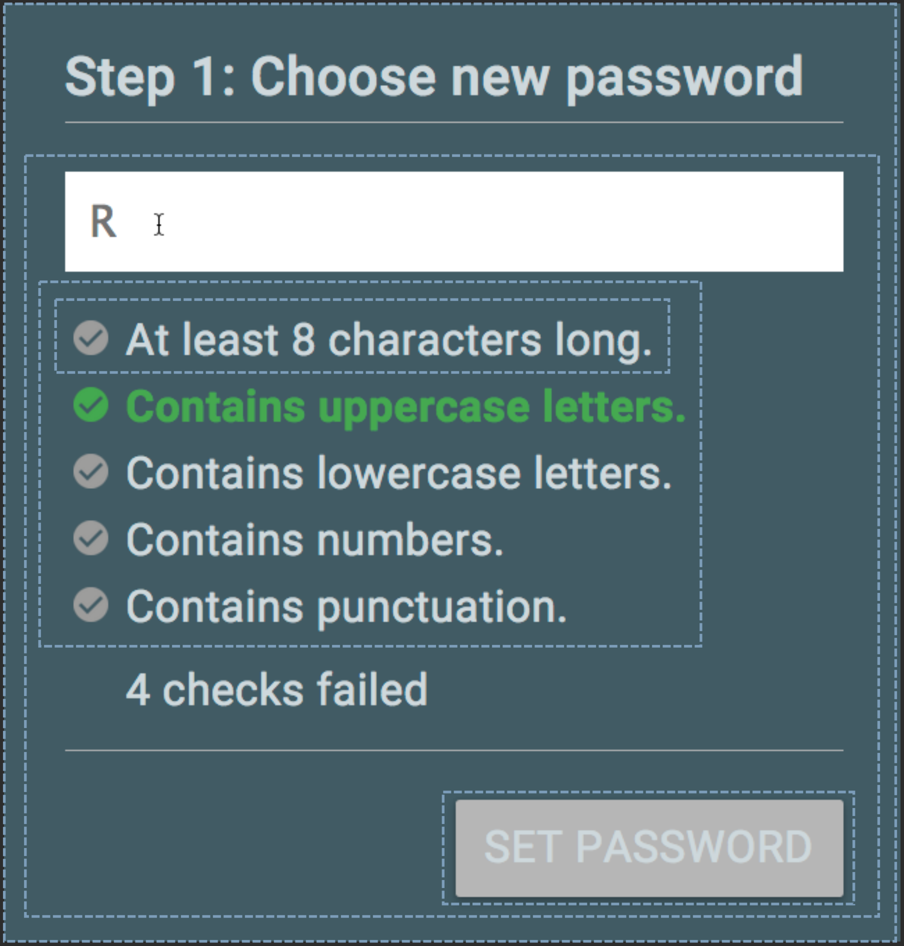
●●

- ✓ At least 8 characters long.
- ✓ Contains uppercase letters.
- ✓ Contains lowercase letters.
- ✓ Contains numbers.
- ✓ Contains punctuation.

3 checks failed

Code: https://github.com/nilshartmann/react-example-app

Demo: https://nilshartmann.github.io/react-example-app/


Step 1: Choose new password

R

At least 8 characters long.

Contains uppercase letters.

Contains lowercase letters.

Contains numbers.

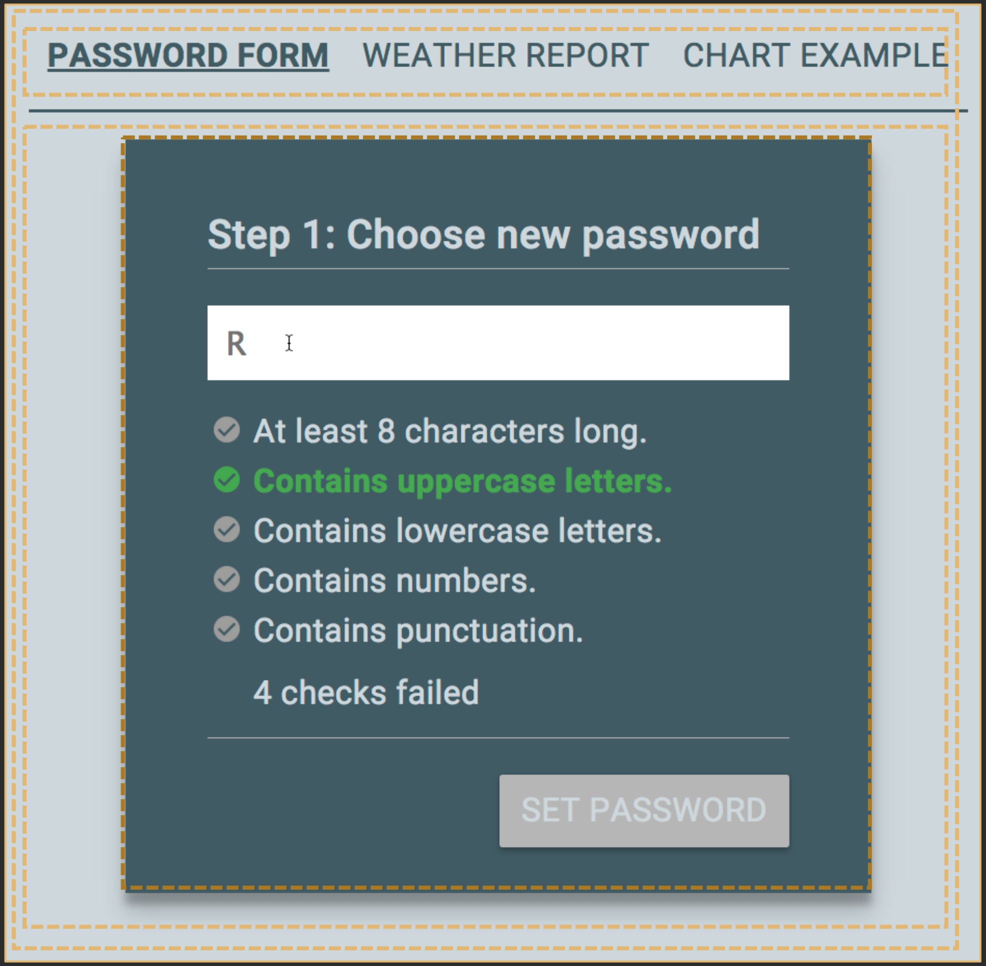
Contains punctuation.

4 checks failed

```
<PasswordView>
  <PasswordForm>
    <input>
    <CheckLabelList>
      <CheckLabel />
      <CheckLabel />
    </CheckLabelList>
    <Label />
    <Button />
  </PasswordForm>
</PasswordView>
```

Wiederverwendbar
Hierarchisch
Logik und UI

KOMPONENTEN



Step 1: Choose new password

R

- ✓ At least 8 characters long.
- ✓ Contains uppercase letters.
- ✓ Contains lowercase letters.
- ✓ Contains numbers.
- ✓ Contains punctuation.

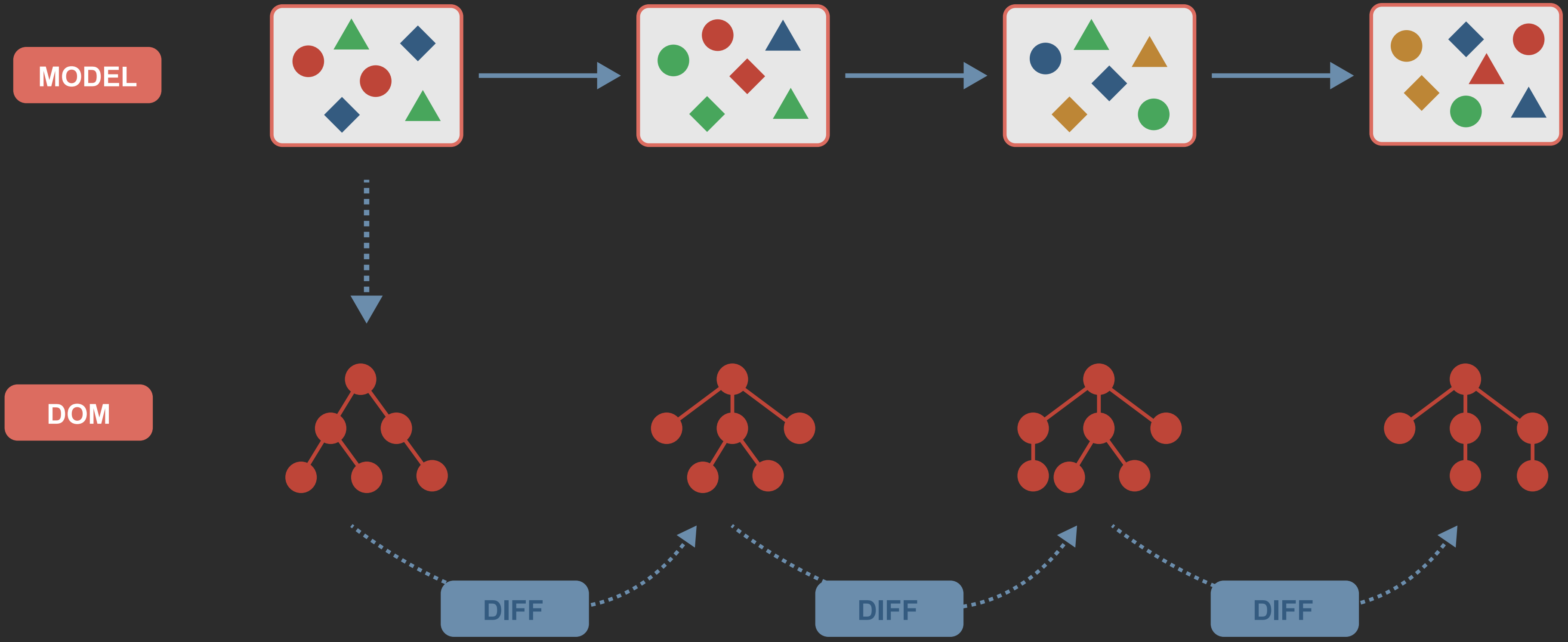
4 checks failed

```
<Application>
  <Navigation />
  <ViewContainer>
    <PasswordView>
      . . .
    </PasswordView>
  </ViewContainer>
</Application>
```

Aus Komponenten aggregiert

ANWENDUNGEN

Hintergrund



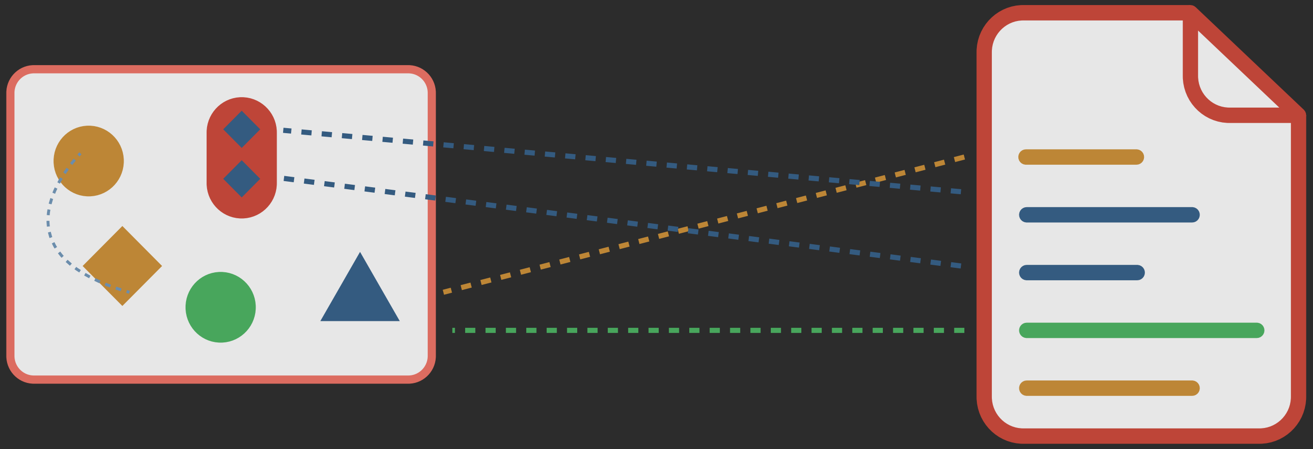
Manuelle DOM-Manipulationen

Umständliche API

Fehleranfällig

Performance-kritisch

DOM OPERATIONEN



Verbinden von Model und View

Wann wird was gebunden?

Wie funktioniert das Binding?

Reihenfolge von Events?

DATA BINDING

respond to events & render UI

Einfachheit

REACT

$$\begin{tikzcd}
\text{Event} \arrow[r, bend left, dashed] & \text{Zustand} \arrow[r, bend left, dashed] & \text{Rendern}
\end{tikzcd}$$

- Mausklick
- Texteingabe
- Timer
- Serverantwort
- . . .

Einfachheit

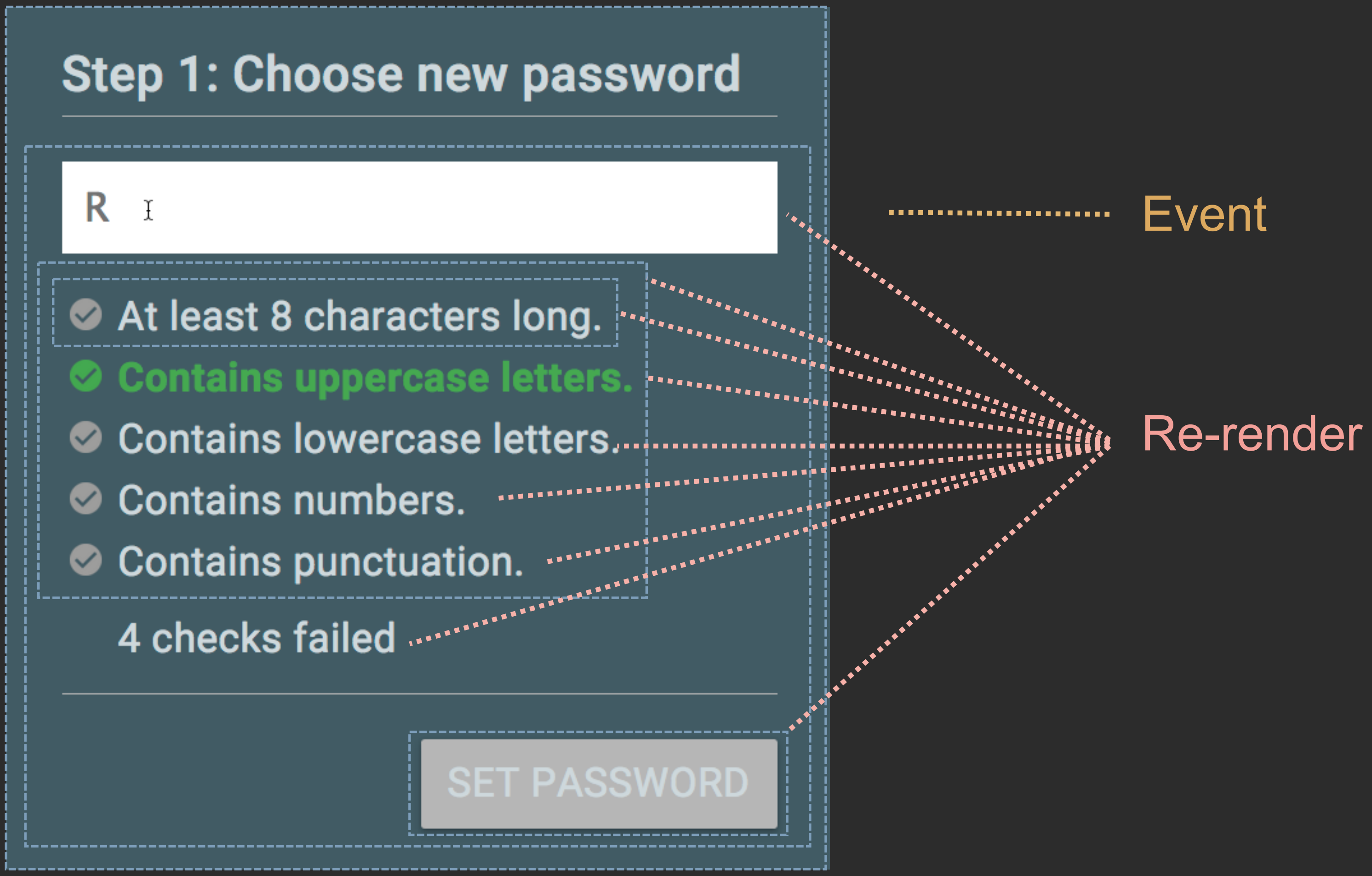
respond to events & and render UI

$$\begin{tikzcd}\text{Event} \arrow[r, bend left, dashed] & \text{Zustand} \arrow[r, bend left, dashed] & \text{Rendern} \arrow[d, dash, dashed] \\ & & \text{Immer \underline{ganze} Komponente rendern}\end{tikzcd}$$

Kein 2-Way-Databinding
Kein dirty checking

Einfachheit

respond to events & and render UI


Step 1: Choose new password

R

✓ At least 8 characters long.

✓ Contains uppercase letters.

✓ Contains lowercase letters.

✓ Contains numbers.

✓ Contains punctuation.

4 checks failed

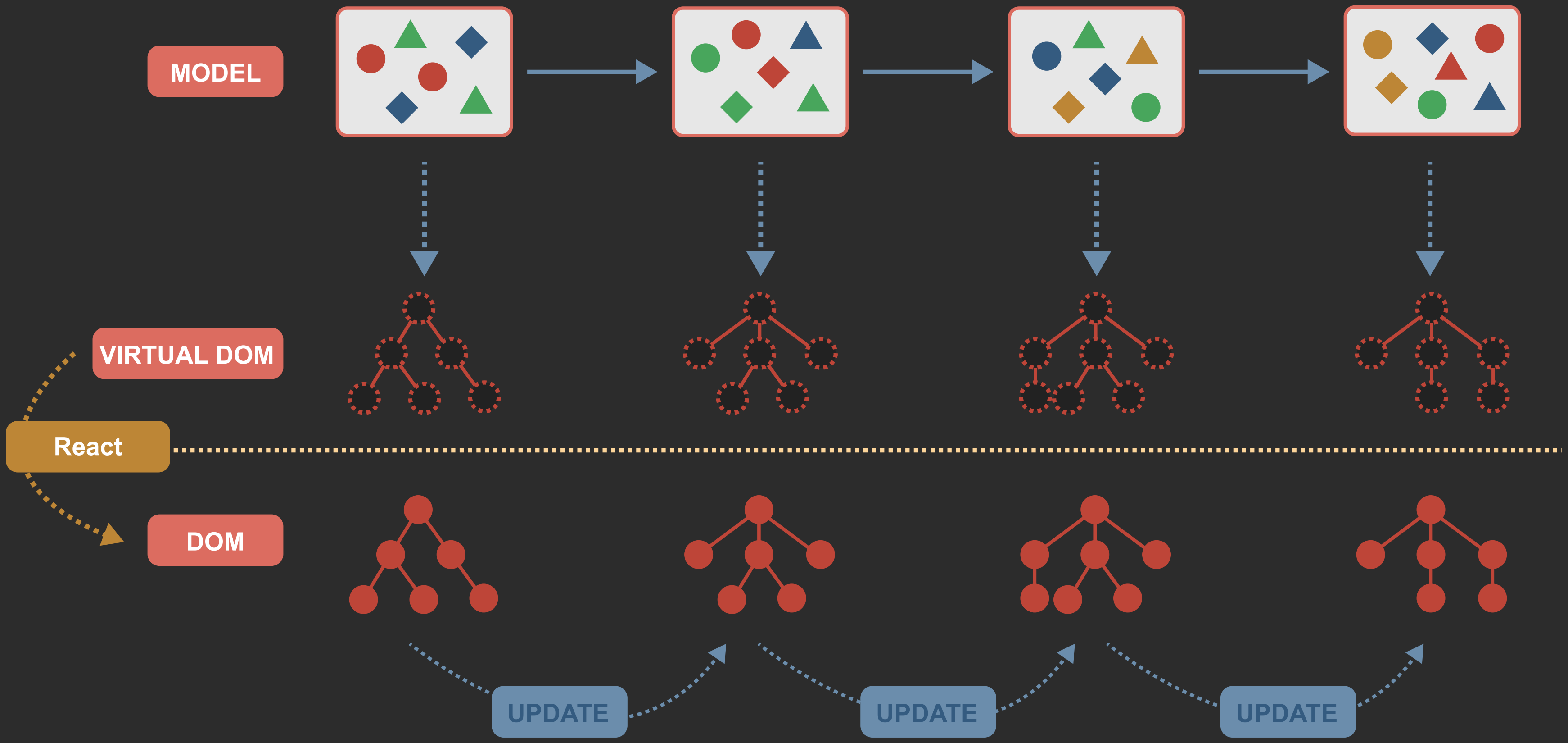
RENDER UI

$$\begin{tikzcd}\text{Event} \arrow[r, bend left, dashed] & \text{Zustand} \arrow[r, bend left, dashed] & \text{Rendern} \arrow[d, dash, dashed] \\ & & \text{Immer ganze Komponente rendern ?}\end{tikzcd}$$

Performance?

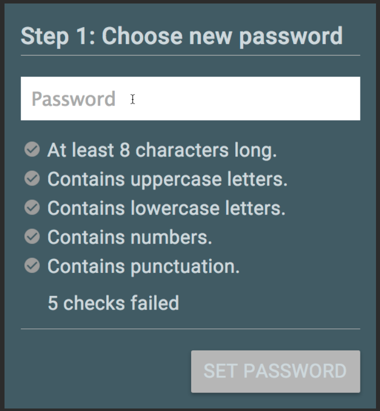
Einfachheit

respond to events & and render UI



VIRTUELLER DOM

`render()`

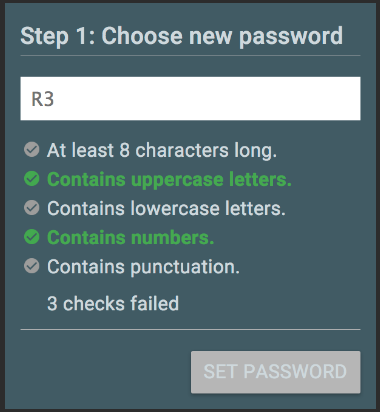

Step 1: Choose new password

Password

- At least 8 characters long.
- Contains uppercase letters.
- Contains lowercase letters.
- Contains numbers.
- Contains punctuation.

5 checks failed

`render(R3)`

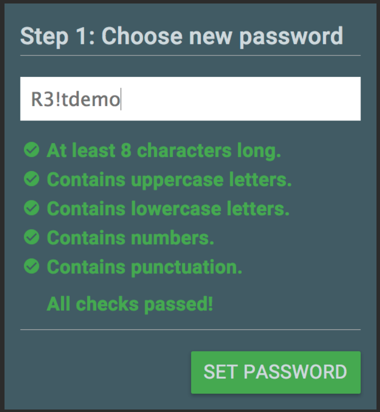

Step 1: Choose new password

R3

- At least 8 characters long.
- ✓ Contains uppercase letters.
- Contains lowercase letters.
- ✓ Contains numbers.
- Contains punctuation.

3 checks failed

`render(R3!tdemo)`

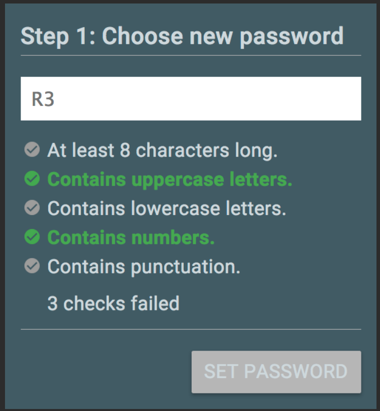

Step 1: Choose new password

R3!tdemo

- ✓ At least 8 characters long.
- ✓ Contains uppercase letters.
- ✓ Contains lowercase letters.
- ✓ Contains numbers.
- ✓ Contains punctuation.

All checks passed!

`render(R3)`


Step 1: Choose new password

R3

- At least 8 characters long.
- ✓ Contains uppercase letters.
- Contains lowercase letters.
- ✓ Contains numbers.
- Contains punctuation.

3 checks failed

$\int$ (model) → UI

Model mit allen Zuständen (Textfelder, Auswahllisten etc)

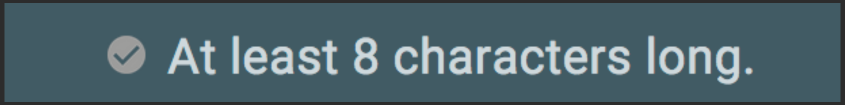
Immer ein Zeitpunkt

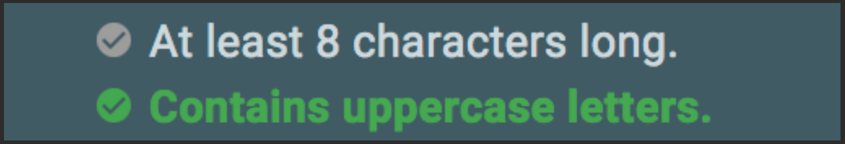
Keine Dynamik

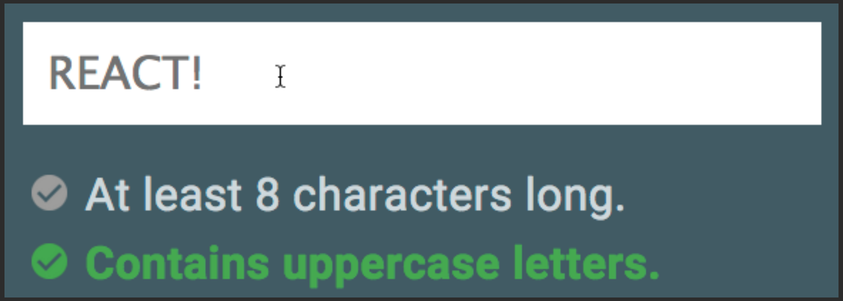
UI AS A FUNCTION

React

PRAXIS


✓ At least 8 characters long.


✓ At least 8 characters long.
✓ Contains uppercase letters.

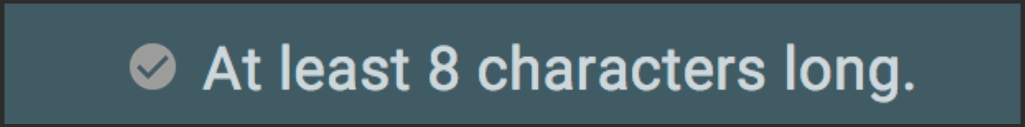

REACT!

✓ At least 8 characters long.
✓ Contains uppercase letters.

React

SCHRITT FÜR SCHRITT

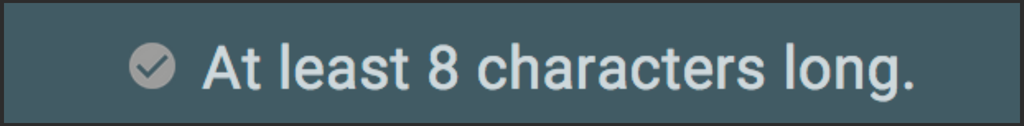
EINE REACT KOMPONENTE 1


At least 8 characters long.

```
<div
  class="CheckLabel-unchecked">
  At least 8 characters long.
</div>
```

HTML

EINE REACT-KOMPONENTE 2



Komponente CheckLabel

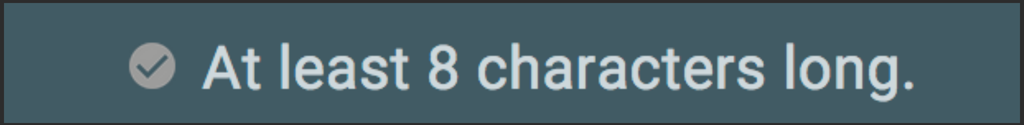
Komponentenfunktion (seit 0.14)

```
function CheckLabel() {
  return <div
    className="CheckLabel-unchecked">
    At least 8 characters long.
  </div>;
}
```

JSX: Statt Template-Sprache

CheckLabel.js

EINE REACT-KOMPONENTE 3

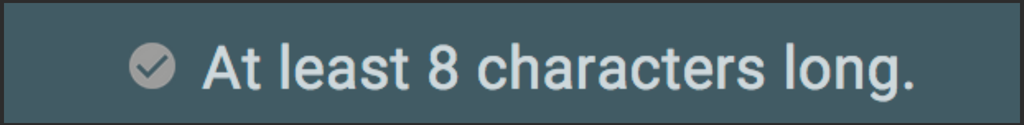


Erzeugt „virtuelles“ DOM-Element

```
React.createElement(
  "div",
  { className: "CheckLabel-unchecked" },
  "At least 8 characters long."
);
```

Übersetzter JavaScript Code

KOMPONENTE RENDERN


✓ At least 8 characters long.

```
import React from 'react';
import ReactDOM from 'react-dom';

import CheckLabel from './CheckLabel';


ReactDOM.render(<CheckLabel />,
  document.getElementById('mount'));
```

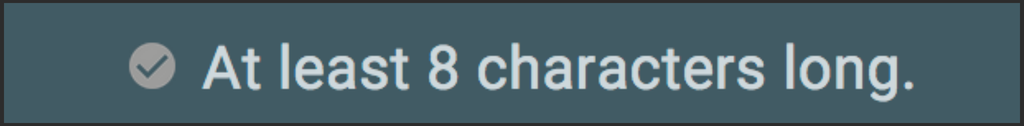
app.js

```
<html>
  <body>
    <div id="mount"></div>
  </body>
  <script src="dist/dist.js"></script>
</html>
```

Webpack

index.html

PROPERTIES

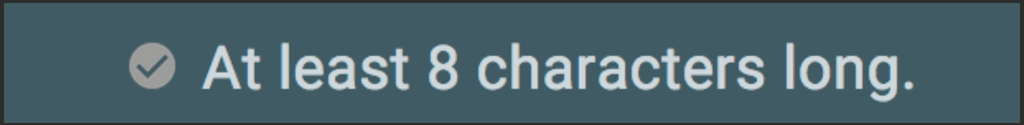

At least 8 characters long.

```
{
  checked: false,
  label: ‘At least 8 characters long.’
}
```

```
function CheckLabel({checked, label}) {
  return <div
    className=
      {checked?'CheckLabel-checked':'CheckLabel-unchecked'}>
    {label}
  </div>;
}
```

Properties (destrukturiert)

PROPERTIES BESCHREIBEN

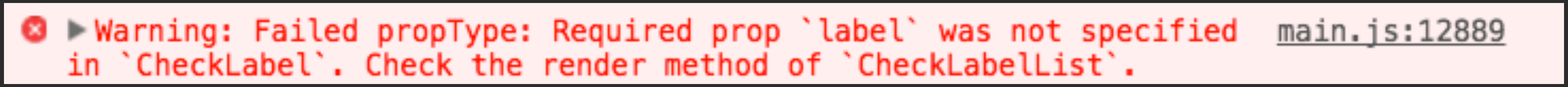

✓ At least 8 characters long.

```
function CheckLabel({checked, label}) {
  // . . .
}
```

Beschreibung der Properties

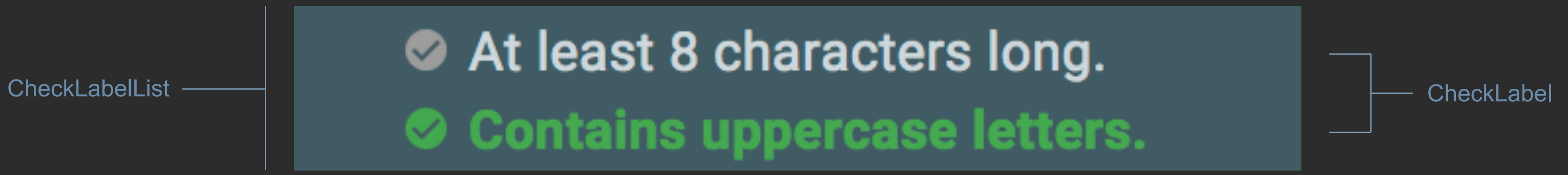
```
CheckLabel.propTypes = {
  label:   React.PropTypes.string.isRequired,
  checked: React.PropTypes.bool
};
```

Überprüfung zur Laufzeit



```
▶ Warning: Failed propType: Required prop `label` was not specified   main.js:12889
  in `CheckLabel`. Check the render method of `CheckLabelList`.
```

KOMPONENTEN VERWENDEN

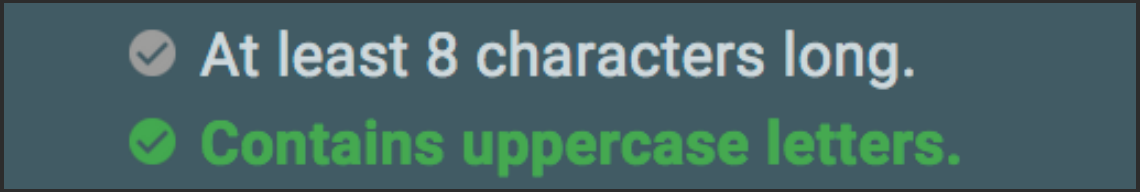


At least 8 characters long.

Contains uppercase letters.

```
function CheckLabelList() {
  return <div>
    <CheckLabel checked={false}
       label='At least 8 characters long' />
    <CheckLabel checked={true}
       label='Contains uppercase letters.' />
  </div>;
}

function CheckLabel({checked, label}) {
  // . . .
}
```

LISTEN



✓ At least 8 characters long.
✓ Contains uppercase letters.

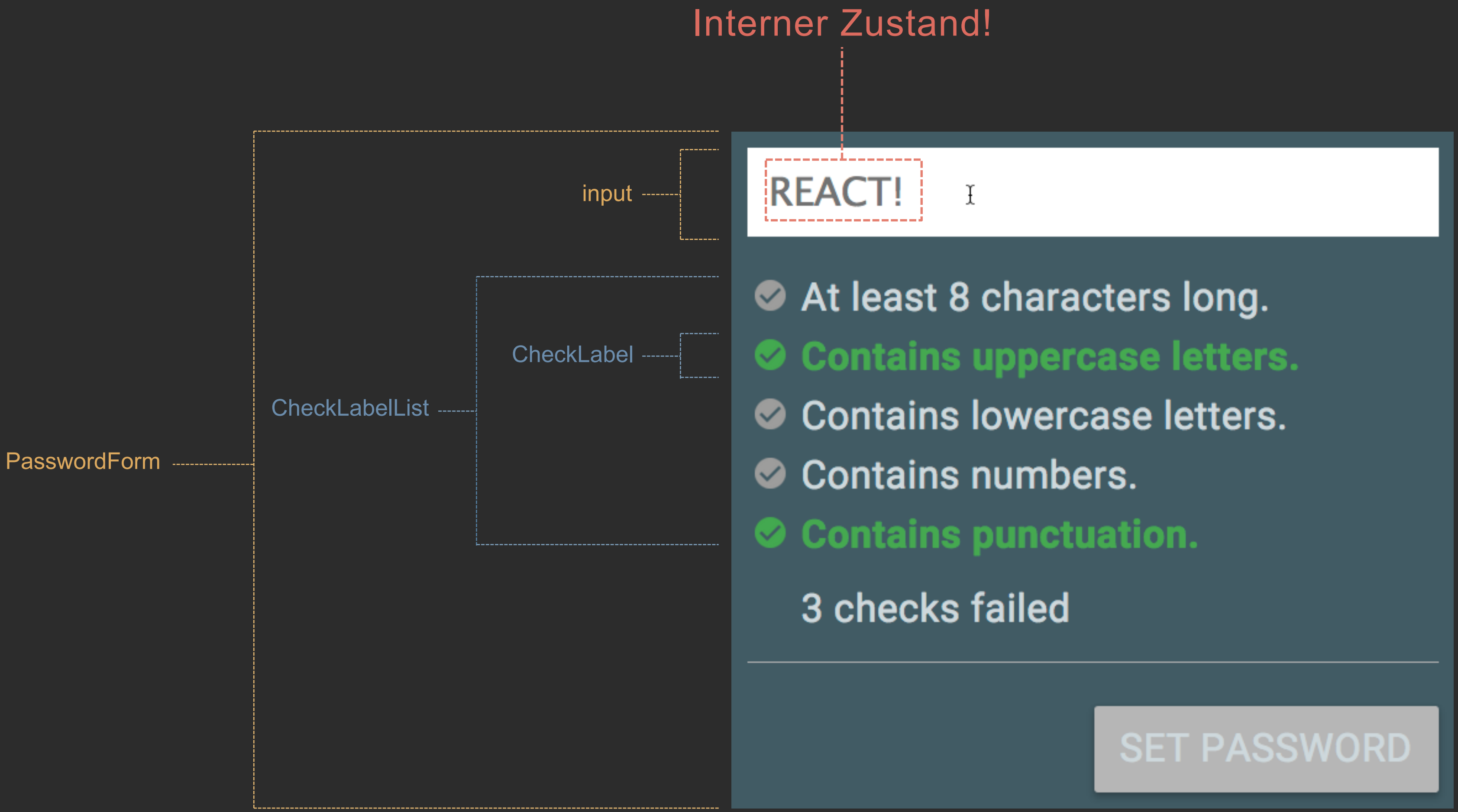
```
[
  { checked: false, label: ‘At least 8 characters long.’ },
  { checked: true,  label: ‘Contains uppercase letters’ }
];
```

```
function CheckLabelList({checks}) {
  return <div>
    {checks.map((c) => <CheckLabel label={c.label}
                                   checked={c.checked}
                                   key={c.label} />
    )}
  </div>;
}
```

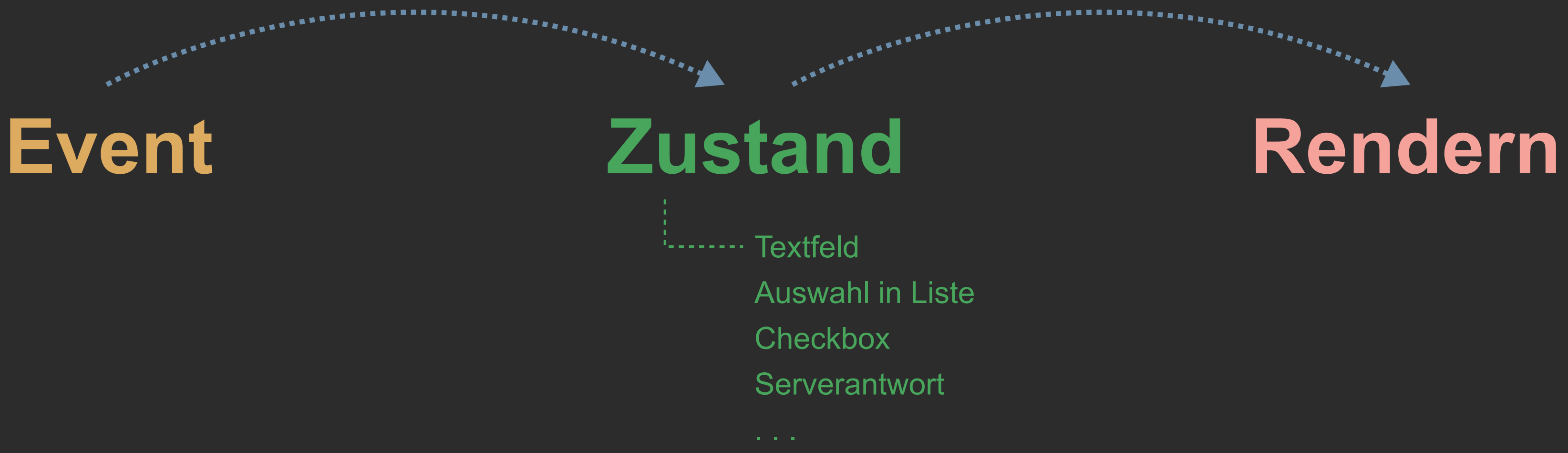
ES5 Array.prototype.map()

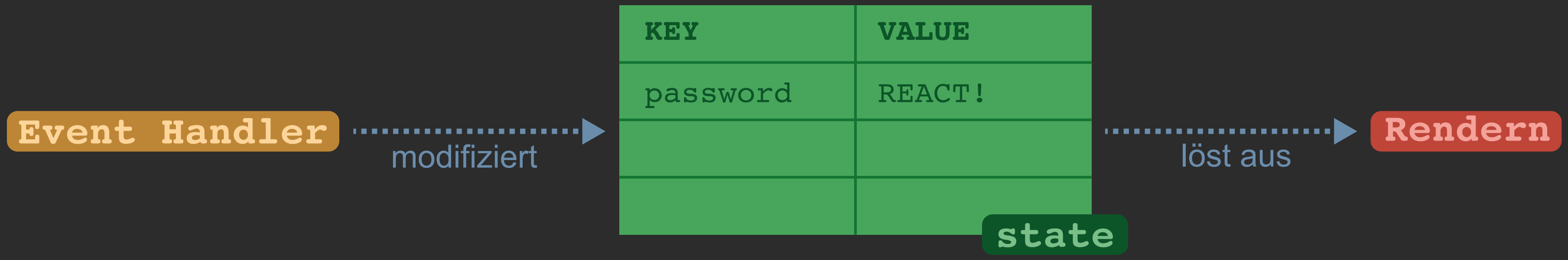
Eindeutiger Schlüssel

ZUSTANDSBEHAFTETE KOMPONENTEN



ZUSTAND





KEY	VALUE
password	REACT!

KOMPONENTEN KLASSEN

ECMAScript 2015 Klasse

Properties über Konstruktor

React Lifecycle Methoden

Render Methode

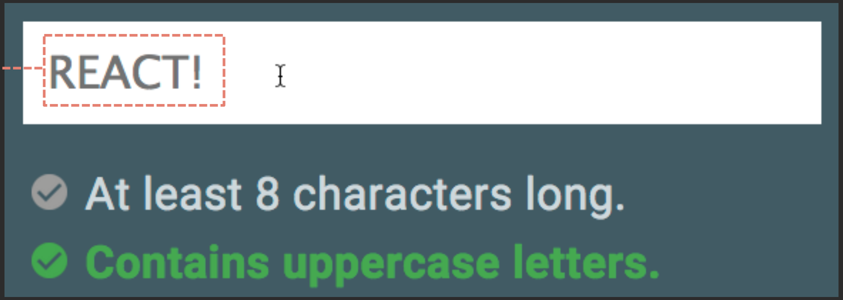
Properties über props-Objekt

Property-Beschreibungen

```
class PasswordForm extends React.Component {
  PasswordForm(props) {
    super(props);
  }

  componentDidMount() { . . . }
  componentWillReceiveProps() { . . . }
  shouldComponentUpdate() { . . . }
  . . .

  render() {
    return <div>{this.props.label}</div>;
  }
}

PasswordForm.propTypes = {
  . . .
};
```

ZUSTAND UND RENDERING

Zustand!



At least 8 characters long.

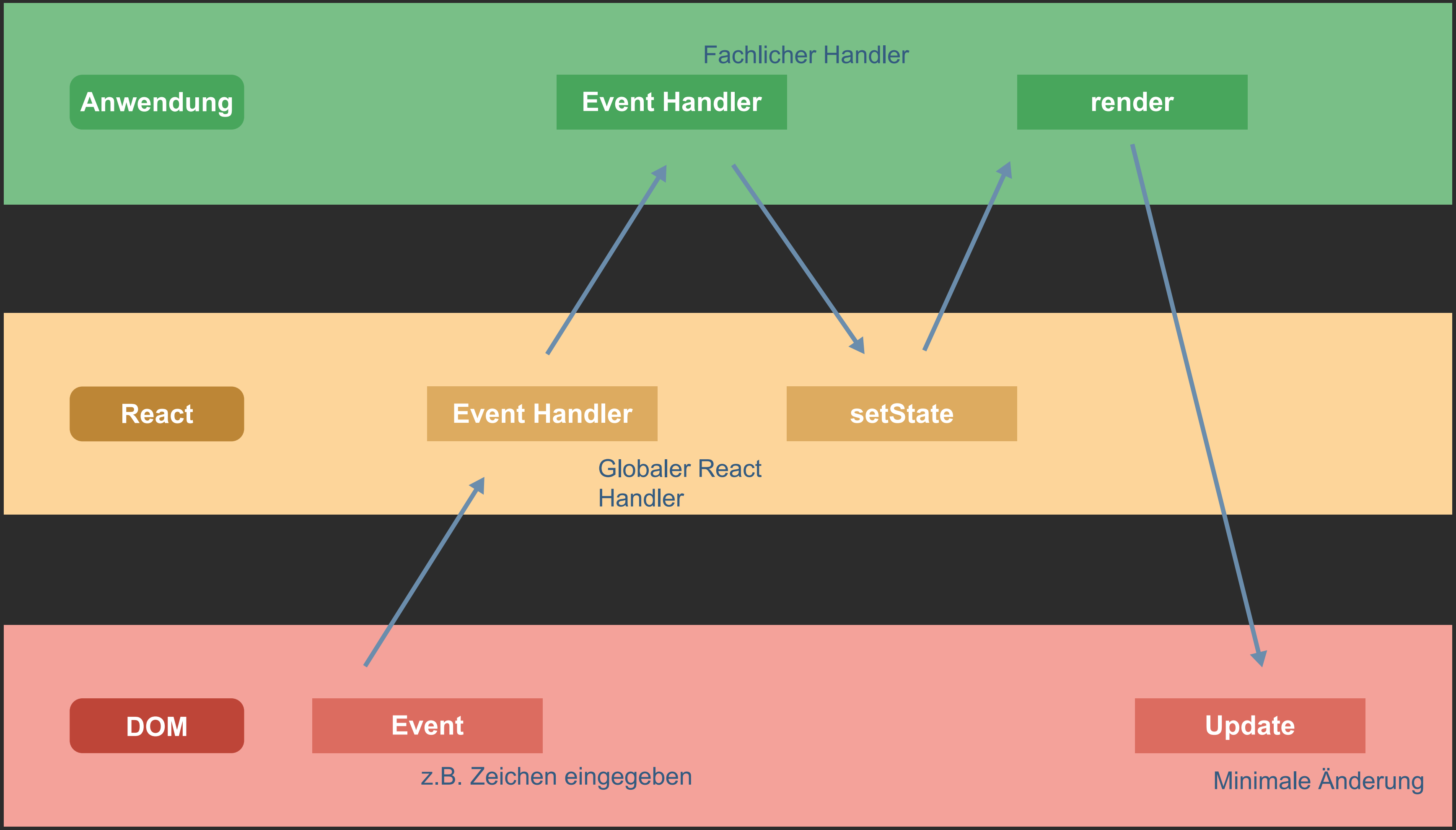
Contains uppercase letters.

```
class PasswordForm extends React.Component {
 checkPassword(password) { return [ . . . ]; }

 onPasswordChange(input) {
  this.setState({password: input});
 }

 render() {
  const checks = this.checkPassword(this.state.password);

  return . . .
   <input value={this.state.password}
        onChange={e=>this.onPasswordChange(e.target.value)}
   />
   <CheckLabelList checks={checks} />
   <Button enabled={passwordValid} />
 }
}
```

2. Zustand neu setzen

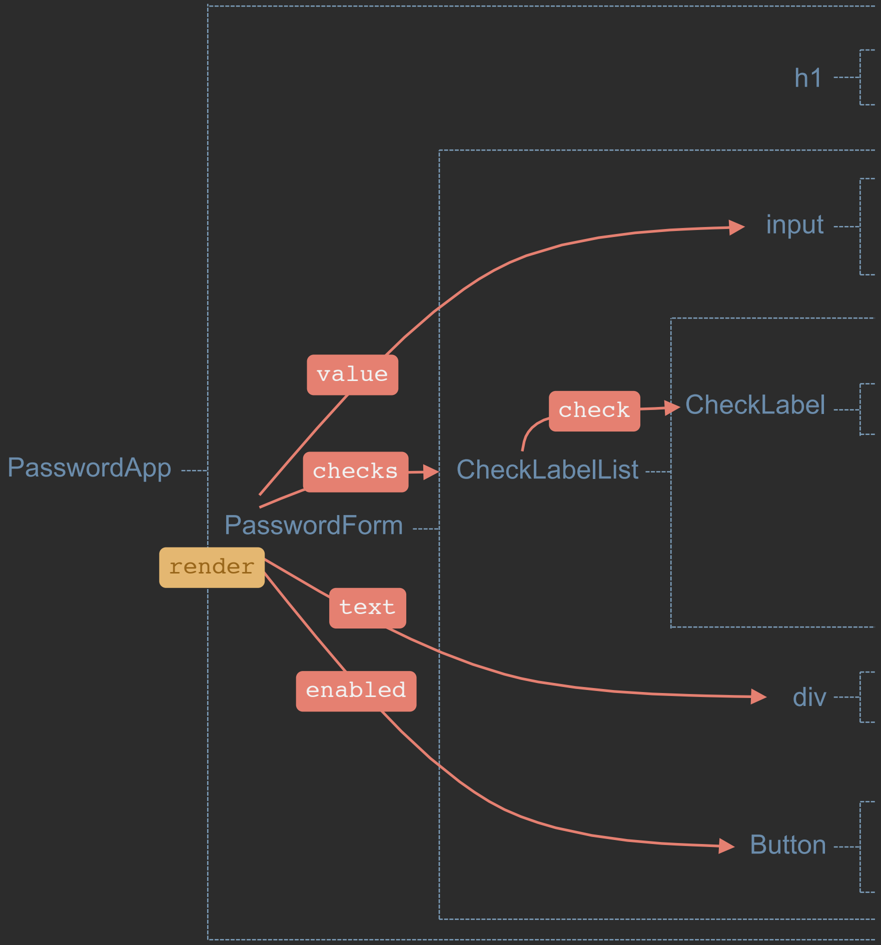
3. löst rendern der gesamten Komponente aus

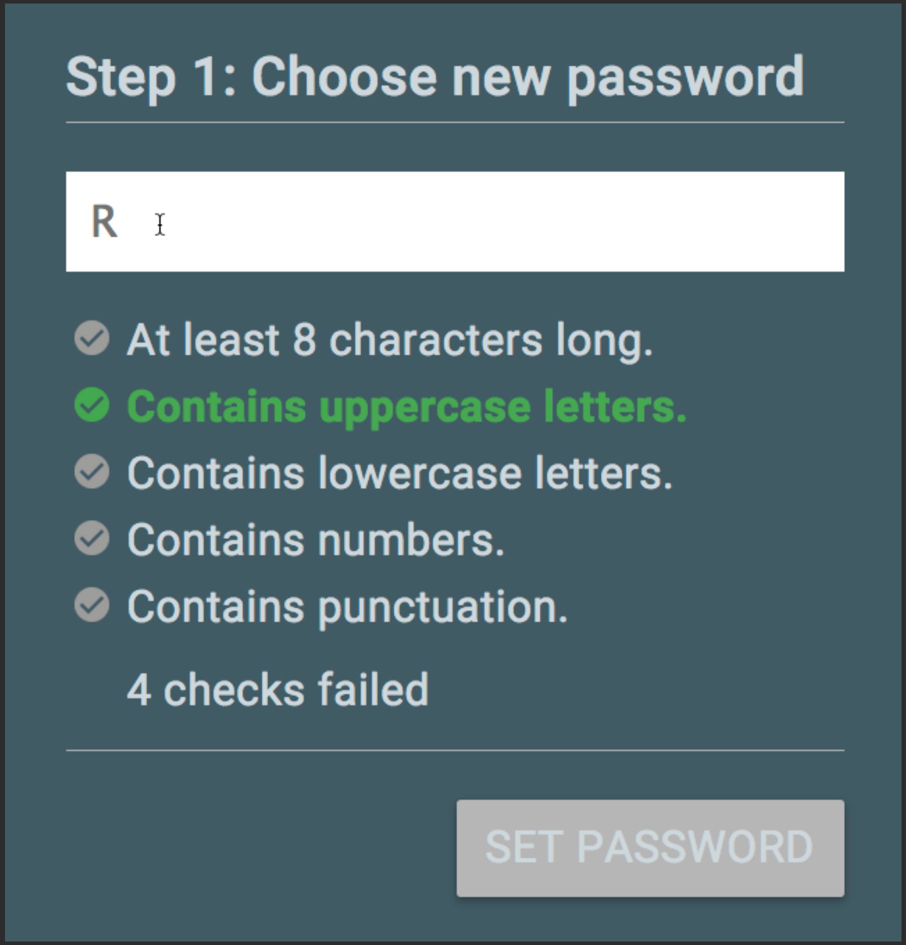
1. Event tritt ein



DOM UPDATES - BIG PICTURE

KOMMUNIKATION: PROPERTIES





Step 1: Choose new password

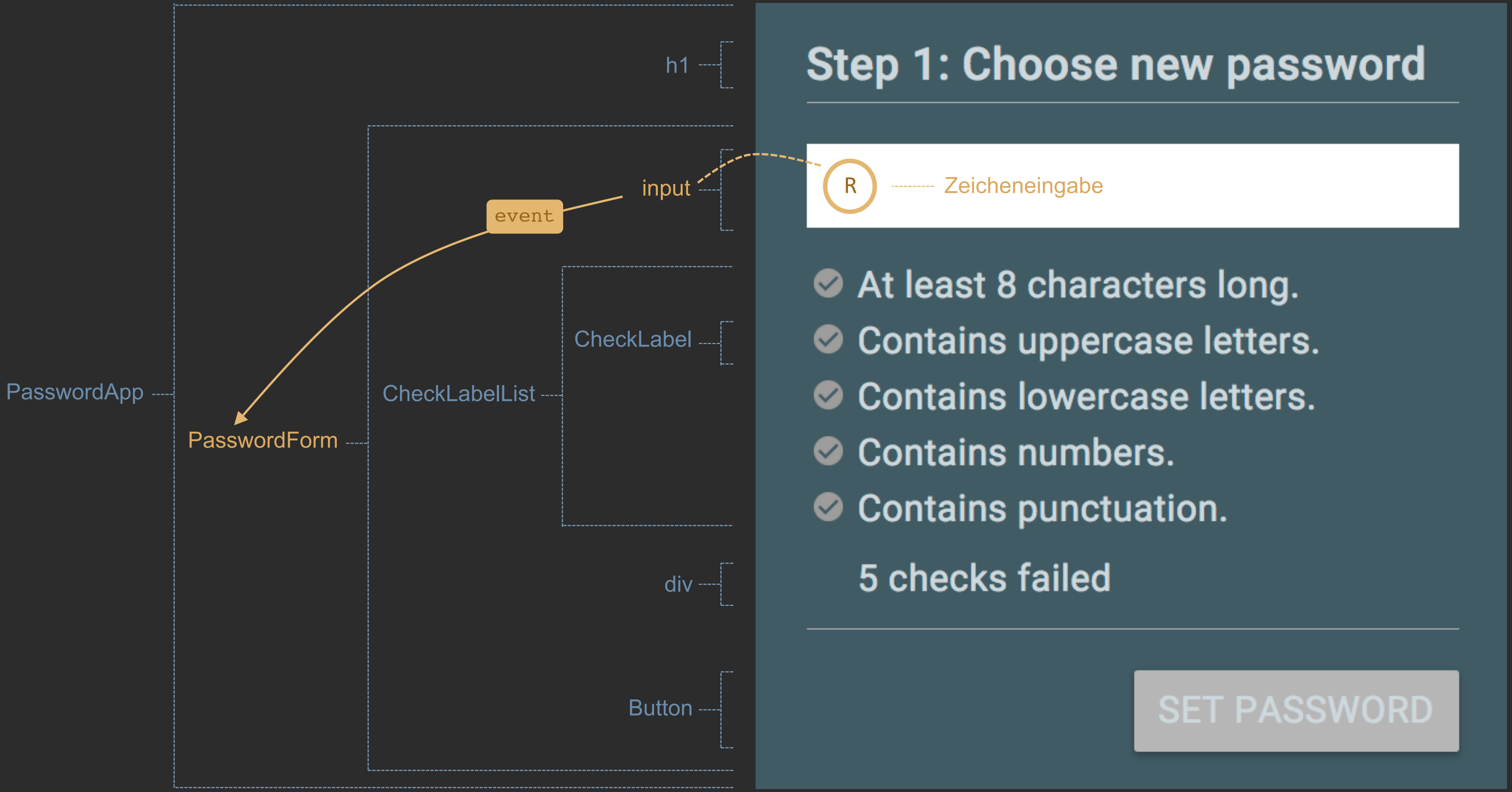
R

✓ At least 8 characters long.
✓ Contains uppercase letters.
✓ Contains lowercase letters.
✓ Contains numbers.
✓ Contains punctuation.

4 checks failed

Von oben nach unten: **props**

KOMMUNIKATION: EVENTS 1



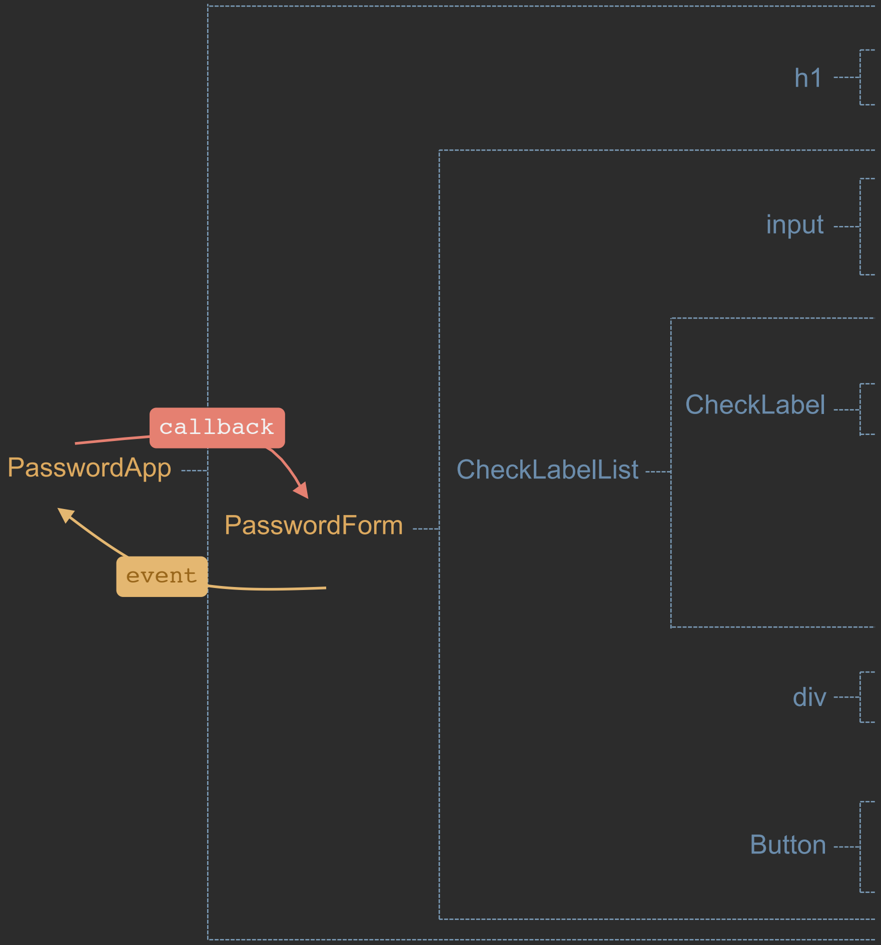
Step 1: Choose new password

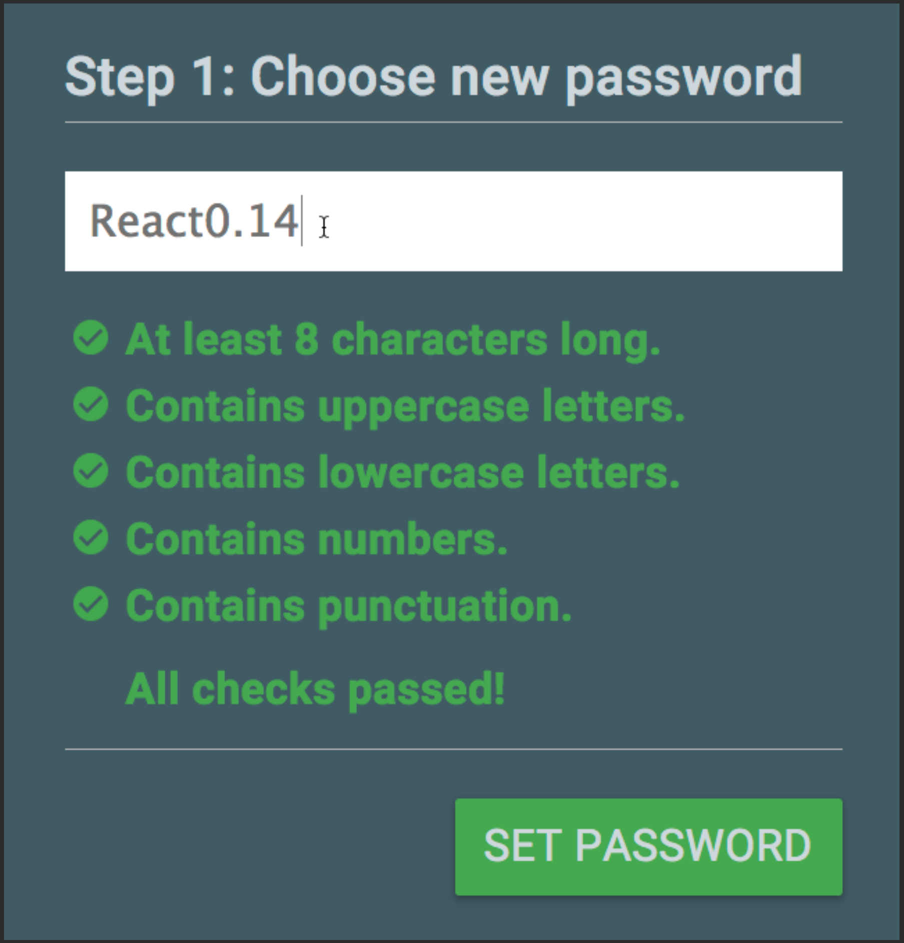
- At least 8 characters long.
- Contains uppercase letters.
- Contains lowercase letters.
- Contains numbers.
- Contains punctuation.

5 checks failed

Von unten nach oben: **events**

KOMMUNIKATION: EVENTS 2




Step 1: Choose new password

React0.14

✓ At least 8 characters long.
✓ Contains uppercase letters.
✓ Contains lowercase letters.
✓ Contains numbers.
✓ Contains punctuation.

All checks passed!

Von unten nach oben: **events** und **callbacks**

KOMMUNIKATION: CALLBACK

```
class PasswordApp extends React.Component {
 onSetPassword(password) { . . . }

 render() {
  return . . .
   <PasswordForm . .  .
         onSetPasswordHandler={p=>this.onSetPassword(p)}
   />;
 }
}
```

3. Event verarbeiten (ggf. Zustand setzen)

1. Callback übergeben

event

```
class PasswordForm extends React.Component {
 render() {
  return . . .
   <input value=“. . .” onChange=“. . .” />
   <Button label=“Set new Password”
     onClickHandler=
       {()=>this.props.onSetPasswordHandler(this.state.password)}
   />
 }
}

PasswordForm.propTypes = {
 onSetPasswordHandler: React.PropTypes.func.isRequired
}
```

2. Callback aufrufen

UNIT-TESTS (OHNE DOM)

```
import TestUtils from 'react-addons-test-utils';

describe('CheckLabel', () => {
 it('should render a "checked" label', () => {

  const renderer = TestUtils.createRenderer();
  renderer.render(
   <CheckLabel label='My Label' checked={true}/>
  );

  const tree = renderer.getRenderOutput();
  expect(tree.type).to.equal('div');
  expect(tree.props.className).to.equal('CheckLabel-checked');
  expect(tree.props.children).to.equal('My Label');
 });
});
```

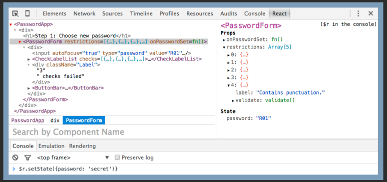
„Shallow rendering“

EINE „ANWENDUNG“ (BEISPIEL)

```
class App extends React.Component {
 handleItemSelected(item) {
  this.setState({component: item.component});
 }

 render() {
  return <div className='App'>
   <NavigationBar
     onItemSelected={item=>this.handleItemSelected(item)}
     items={[
      { label: 'Change password', component: <PasswordApp />},
      { label: 'Show weather',    component: <WeatherApp /> }
     ]}
   />

   <MainView>
    {this.state.component}
   </MainView>
  </div>;
 }
}
```

Navigation (Komponentenauswahl)

Ausgewählte Komponente einfügen



```
▾ <PasswordApp>
  ▾ <div>
      <h1>Step 1: Choose new password</h1>
    ▾ <PasswordForm restrictions=[{…},{…},{…},…] onPasswordSet=fn()>
      ▾ <div>
          <input autoFocus="true" type="password" value="R01"…/>
        ▸ <CheckLabelList checks=[{…},{…},{…},…]>…</CheckLabelList>
        ▾ <div className="Label">
            "3"
            " checks failed"
          </div>
        ▸ <ButtonBar>…</ButtonBar>
        </div>
      </PasswordForm>
    </div>
  </PasswordApp>
```

```
<PasswordForm>                    ($r in the console)
Props
 ▸ onPasswordSet: fn()
 ▾ restrictions: Array[5]
   ▸ 0: {…}
   ▸ 1: {…}
   ▸ 2: {…}
   ▸ 3: {…}
   ▾ 4: {…}
       label: "Contains punctuation."
     ▸ validate: validate()
State
  password: "R01"
```

```
> $r.setState({password: 'secret'})
```

dev tools

material-design

flux

bootstrap

graphql & relay

router

native

fertige Komponenten

Ökosystem

AUSBLICK

Mittagspause (wohlverdient!)

React Router

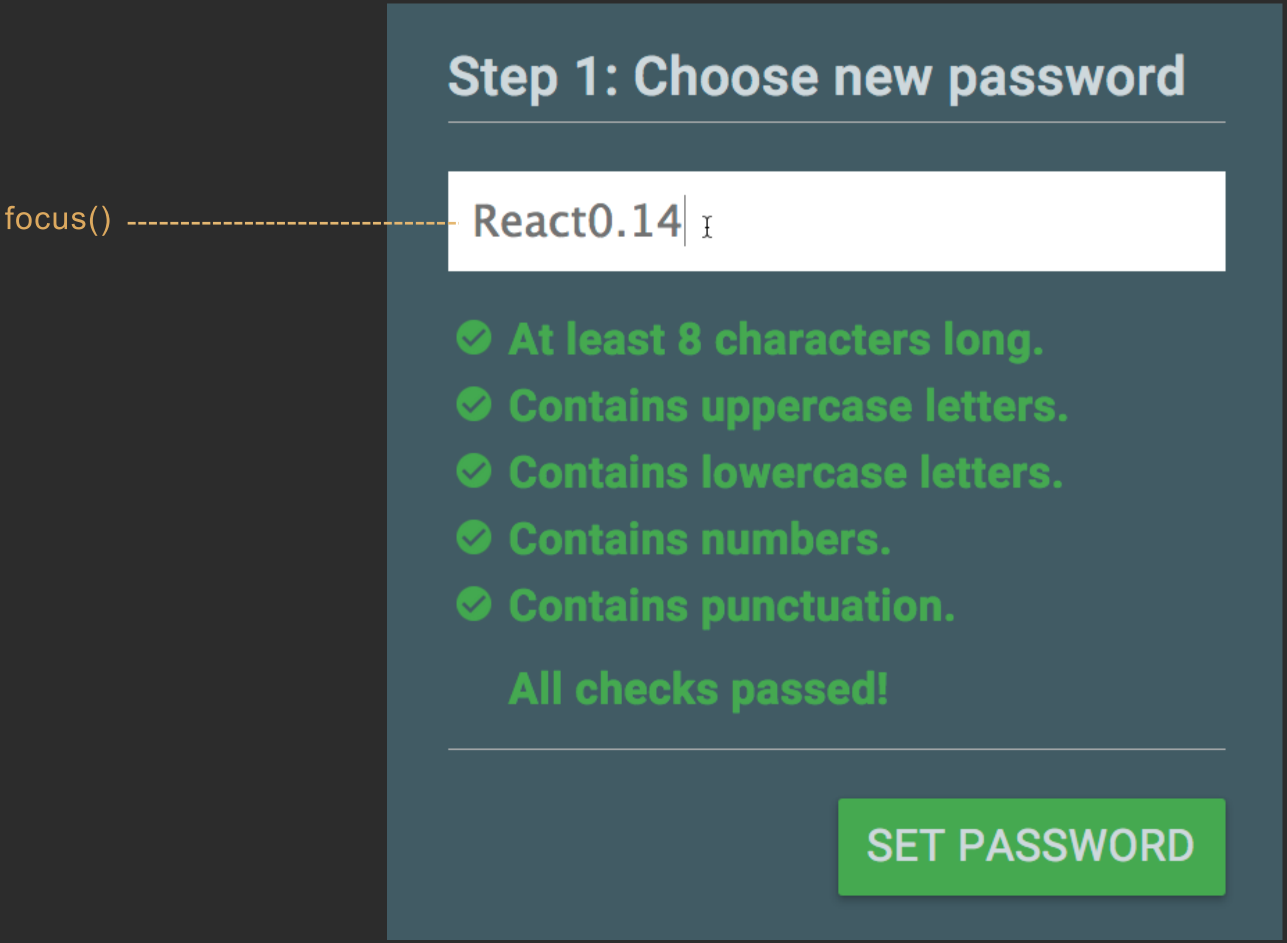
Serverzugriffe

Integration von Dritt-Bibliotheken

Build-Prozess

Zugabe

ZUGRIFF AUF NATIVEN DOM 1



Step 1: Choose new password

React0.14

✓ At least 8 characters long.
✓ Contains uppercase letters.
✓ Contains lowercase letters.
✓ Contains numbers.
✓ Contains punctuation.

All checks passed!

SET PASSWORD

Beispiel: `focus()` auf input-Feld aufrufen

ZUGRIFF AUF NATIVEN DOM 2

React-Callback:
Komponente wurde in den nativen DOM gehängt

Natives DOM-Element

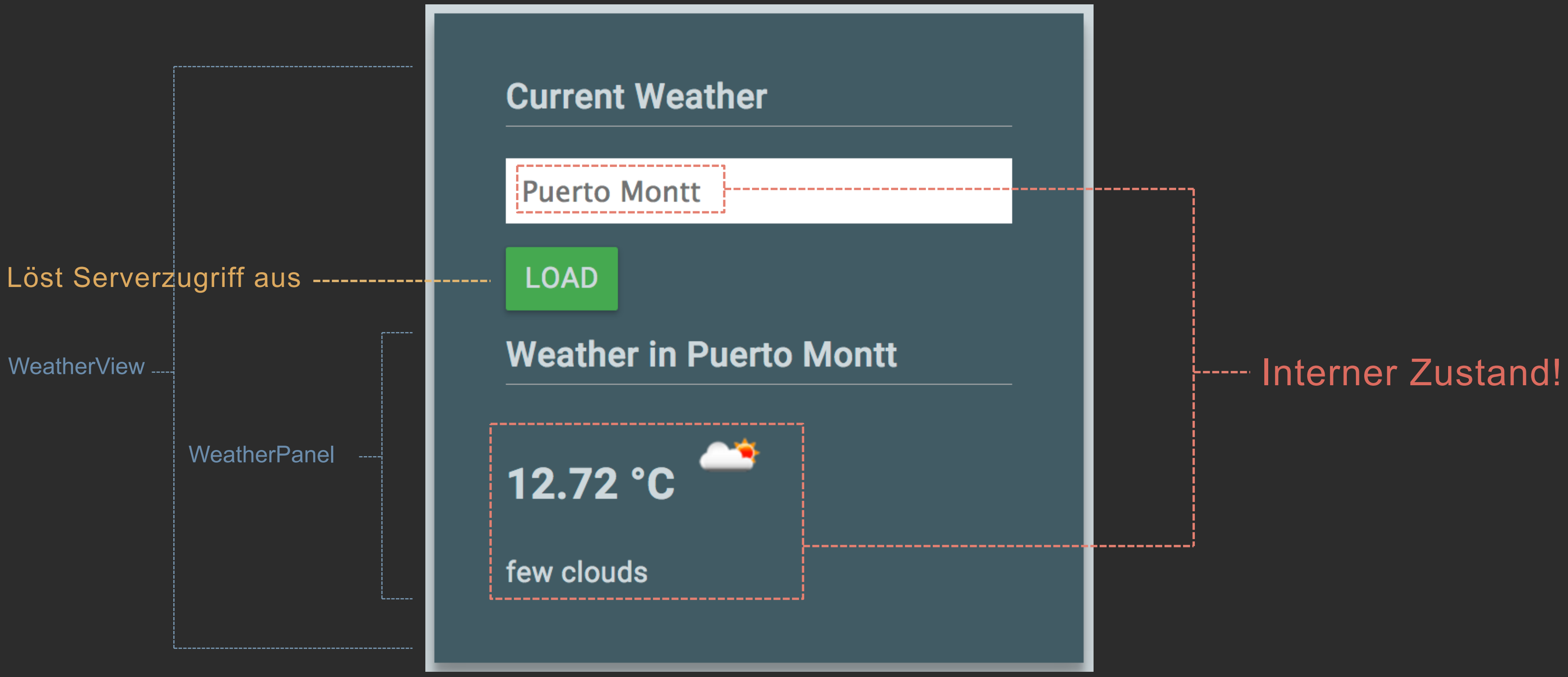
Virtuelles DOM-Element

Referenz anlegen

```
class PasswordForm extends React.Component {

  componentDidMount() {
    this.refs.passwordField.focus();
  }

  render() {
    const password = this.state.password;

    return <div>
      <input ref='passwordField' value={password} />
      . . .
    </div>;
  }
}
```

`this.refs` enthält native DOM-Elemente, die mit `ref` ausgezeichnet wurden

SERVERZUGRIFF



SERVERZUGRIFF 1

fetch-Bibliothek: https://fetch.spec.whatwg.org/

```
import WeatherPanel from './WeatherPanel';

class WeatherView extends React.Component {
 constructor() {
  super();
}

 fetchWeather() {
  const { city } = this.state;
  const fetchUrl = `http://api.w.org/${city}`;
   fetch(fetchUrl)
    .then( response => response.json())
    .then( weather => this.setState({weather}))
   ;
 }

 render() {
  const { city, weather } = this.state;
  <input type='text' value={city} onChange={. . .} />
  <Button label='Load' onClick={() => this.fetchWeather()}
  <WeatherPanel weather={weather} />
 }
}
```

Daten vom Server laden — `fetch(fetchUrl)`

Zustand neu setzen (triggert Rendering) — `.then( weather => this.setState({weather}))`

Geladene Daten anzeigen — `<WeatherPanel weather={weather} />`

SERVERZUGRIFF 2

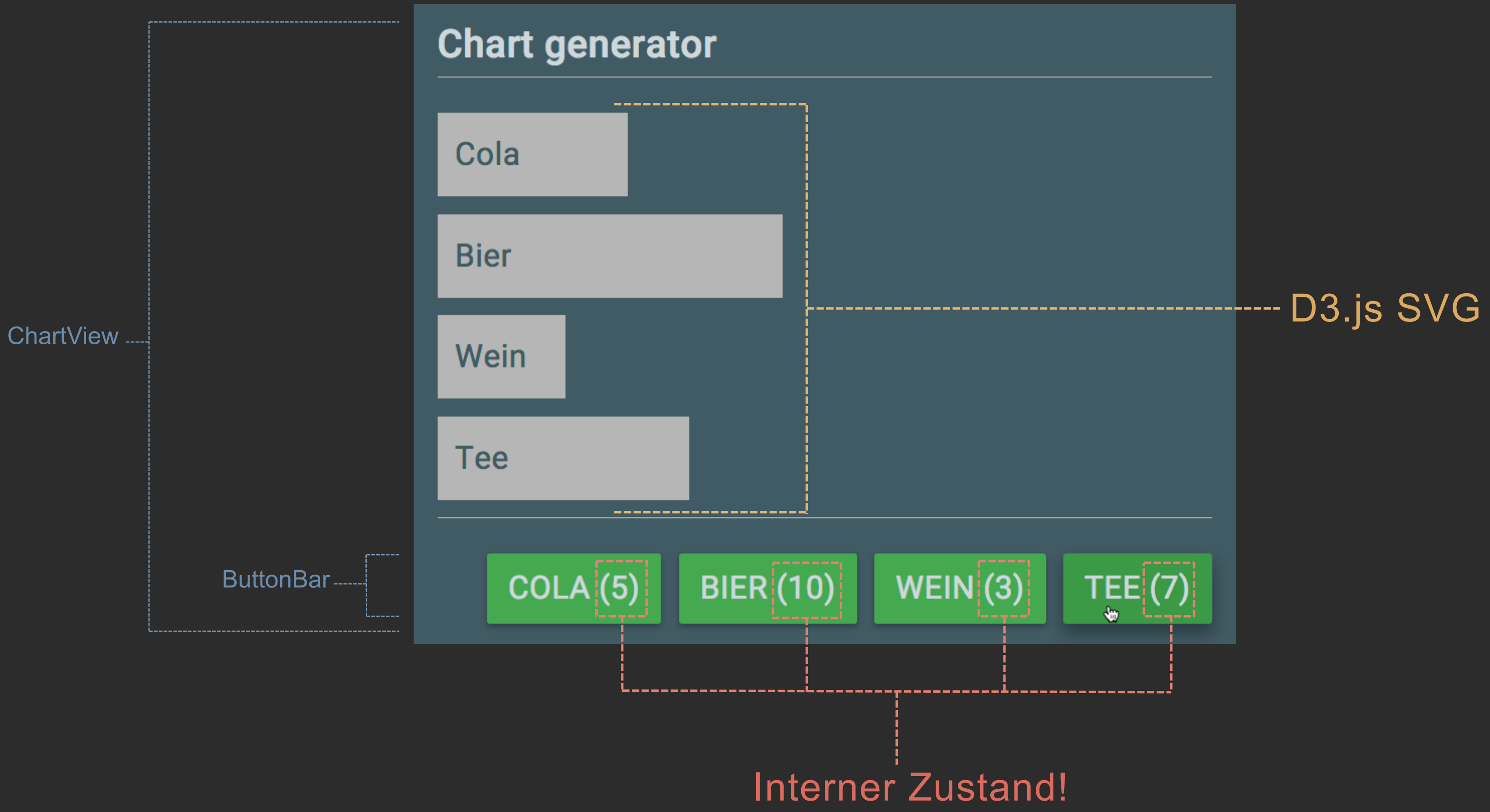
State initialisieren

Wetterdaten laden, sobald Komponente in DOM gehängt wurde

```
class WeatherView extends React.Component {
 constructor() {
  super();
  this.state = { city: 'Hamburg' };
 }

 componentDidMount() {
  this.fetchWeather();
 }

 fetchWeather() { . . . }

 render() { . . . }
}
```

INTEGRATION THIRD-PARTY-LIBS

Beispiel: D3.js



D3.JS

```
import d3 from ‘d3’;

class ChartView extends React.Component {
 constructor() { this.state = { . . .}; }

 increaseDrink(drink) { this.setState({ . . .}); }

 componentDidMount() { this.renderChart(); }
 componentDidUpdate() { this.renderChart(); }

 renderChart() {
  d3.select(this.refs.chart)
   .data(this.state.drinks)
   .enter().append(. . .);
 }
 render() {
  return . . .
   <div ref=‘chart’></div>
   <ButtonBar>
     { drinks.map(d => <Button
         label={ . . .}
         onClickHandler={()=>this.increaseDrink(d)} />)
     }
   </ButtonBar>
 }
}
```

Diagramm (neu) zeichnen, sobald Komponente in DOM gehängt bzw dort aktualisiert wurde

Natives DOM-Element

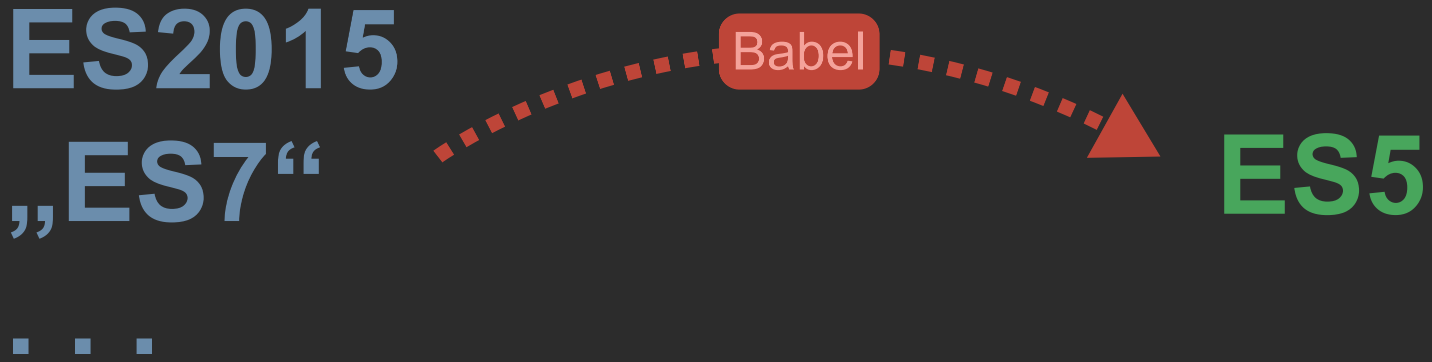
D3 Diagramm wird nicht in render() gezeichnet, weil hier kein natives DOM-Element

Viele ECMAScript 2015 Features

Compiler notwendig

Alias: ES6

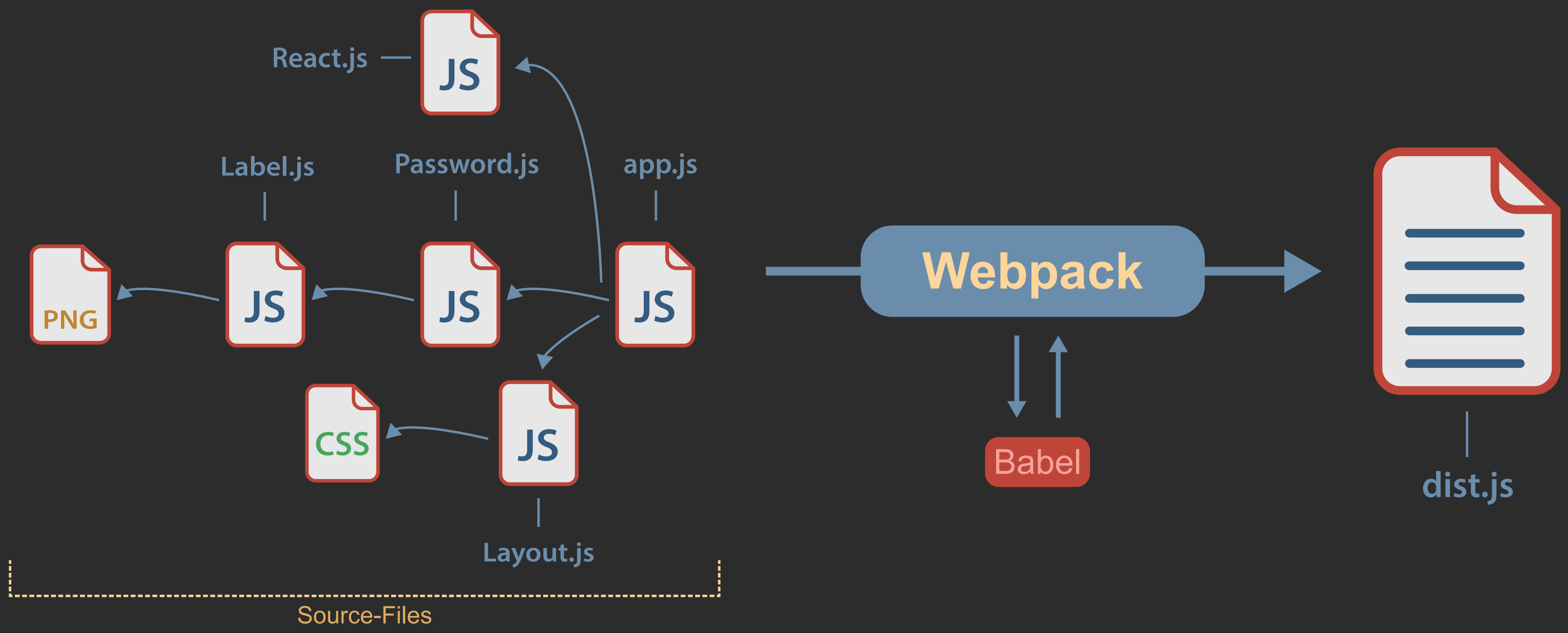
ES2015



Babel is a JavaScript compiler

http://babeljs.io

Babel

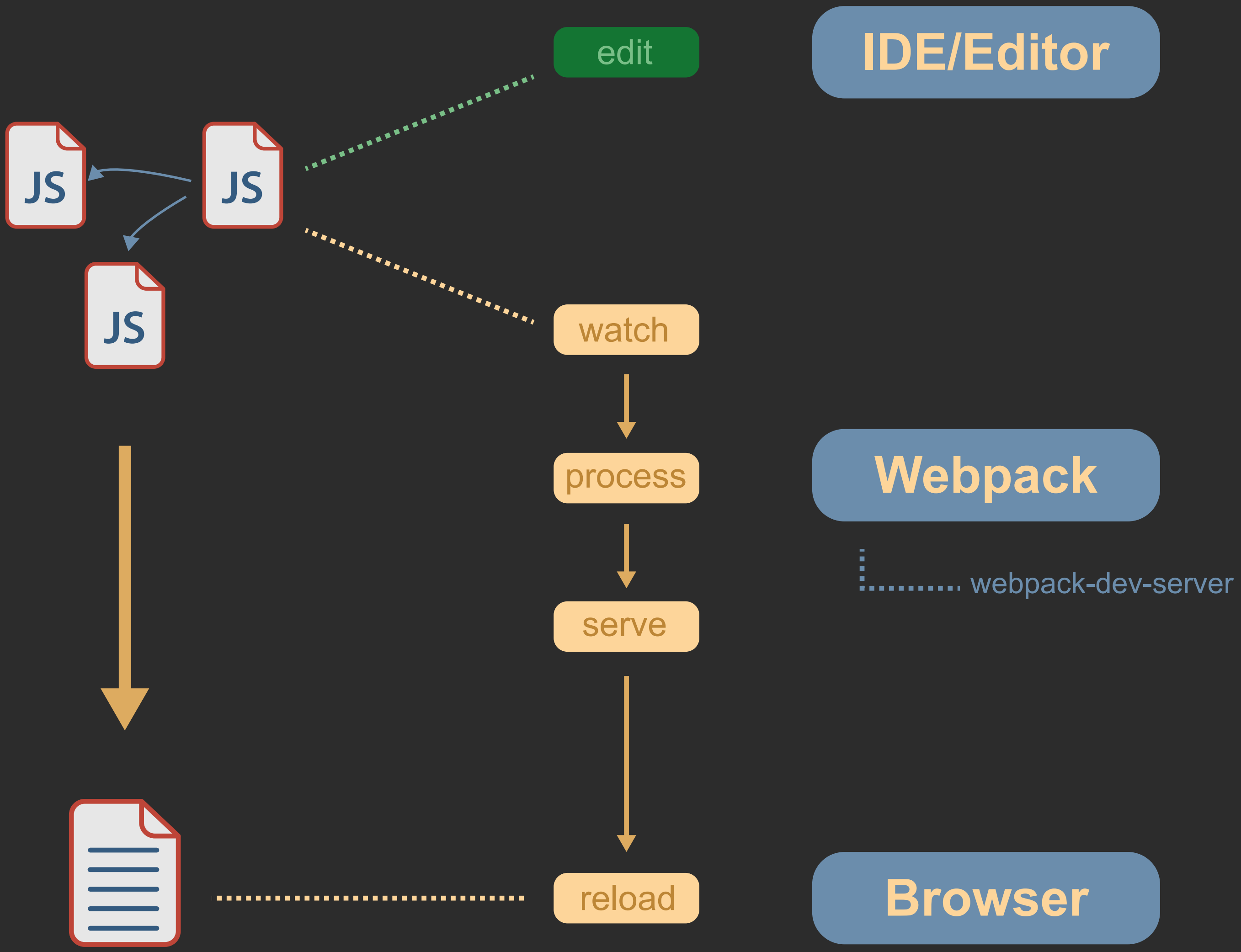


Module bundler - Erzeugt zentrales JavaScriptFile

Webpack

https://webpack.github.io

BUILDPROZESS



webpack-dev-server

BUILDPROZESS

Vielen Dank!

Fragen?

NILS@NILSHARTMANN.NET