

NATIVE APP-ENTWICKLUNG "THE FACEBOOK WAY"

- EINE REACT-NATIVE EINFÜHRUNG

03.12.2016



AppWithUs

WER HIER VOR EUCH STEHT



- AppWithUs
- Sebastian Germesin
- Christoph Byza
- Wir entwickeln professionelle mobile Applikationen mit React Native
- > 10 Apps entwickelt (z.B. Avipeo - soziales Netzwerk für die Luftfahrt)
- sitzen in Hamburg (Altona)



WAS PASSIERT HIER HEUTE

- (kurze) Einführung in Javascript (ECMA Script 6)
- Theoretische Einführung in React Native
- Hands-on Programmierung einer eigenen ~~Maps~~ Chat-App
- Debriefing

JAVASCRIPT - ES6 BINDINGS

```
1  var x = 5;  
2  x = 3;  
3  console.log(x); // => 3  
4  
5  let y = 19;  
6  y = 18;  
7  console.log(y); // => 18  
8  let y = 17; // => SyntaxError: Identifier 'y' has already been declared  
9  
10 const z = 5;  
11 z = 2; // => TypeError: Assignment to constant variable.  
12
```

JAVASCRIPT - ES6 DESTRUCTURING

```
1 let cat = {  
2   name: 'Kitty',  
3   age: 3,  
4   height: 31  
5 };  
6  
7 console.log(cat.name); // => 'Kitty'  
8 console.log(cat['age']); // => 3  
9  
10 delete cat.height;  
11 console.log(cat.height) // => undefined  
12
```

```
let { name } = cat;  
console.log(name); // => 'Kitty'  
  
let othercat = {  
  ...cat,  
  name: 'Garfield',  
};  
console.log(othercat.name); // => 'Garfield';  
console.log(othercat.height); // => 31;
```

JAVASCRIPT - ES6 ARROW FUNCTIONS

```
1 function mult (a, b) {  
2   return a * b;  
3 }  
4 console.log(mult(4,5)); // => 20  
5  
6 //ES6  
7 const multES6 = (a, b) => a * b;  
8 console.log(multES6(4,5)); // => 20  
9
```

```
1 let x = ['ReactJS', 'ReactNative', 'Cordova'];  
2 let y = x.filter(n => n.indexOf('React') !== 0);  
3 let z = x.filter(n => !n.indexOf('React')); // => nice, but not understandable for others  
4 console.log(y); // => [ 'ReactJS', 'ReactNative' ]  
5 console.log(z); // => [ 'ReactJS', 'ReactNative' ]  
6
```

EINFÜHRUNG REACTJS

- ReactJS wurde 2011 von Facebook entwickelt und 2013 "open-sourced"
- Hauptbestandteile:
 - Flexible (kleine) JavaScript library
 - Große Community (> 10.000 npm repositories)
 - JSX
- Seit 2015 auch erweitert für Mobile Anwendungen (React Native)

EINFÜHRUNG REACT NATIVE



React Native hat die gleichen Ansätze wie ReactJS, aber ist von der darunterliegenden Technik sehr unterschiedlich - mit Absicht 🧐

REACT NATIVE VS. IONIC/XAMARIN/...

- React Native benutzt weitestgehend native Elemente und steuert diese über JavaScript an. Daher ergibt sich eine **natürliche** Reaktionszeit von UI-Elementen
- Hybride Apps auf Basis von Ionic o.ä. basieren auf einer WebView, welche native Elemente nachbildet. Daraus erfolgt ein leicht-verzögertes UI.

REACT NATIVE VS. TRULY-NATIVE

- React Native setzt auf einen Ansatz, der es ermöglicht, Apps für Android und iOS zu bauen und dabei den Großteil des geschriebenen Codes wiederzuverwenden (> 80%).
- "Truly Native" bedeutet, eine App für iOS Geräte in z.B. Swift zu entwickeln und anschließend, die gleiche App in Java für das Android System zu bauen. Dadurch entsteht ein Overhead von 100%, welcher die Entwicklung sehr teuer macht und Releasezyklen vergrößert.

WAS IST JSX

- Abkürzung von JavaScript extension syntaX
- <https://facebook.github.io/jsx/>

"THE PURPOSE OF THIS SPECIFICATION IS TO DEFINE A CONCISE AND FAMILIAR SYNTAX FOR DEFINING TREE STRUCTURES WITH ATTRIBUTES. A GENERIC BUT WELL DEFINED SYNTAX ENABLES A COMMUNITY OF INDEPENDENT PARSERS AND SYNTAX HIGHLIGHTERS TO CONFORM TO A SINGLE SPECIFICATION."

CONFORM TO A SINGLE SPECIFICATION."
INDEPENDENT PARSERS AND SYNTAX HIGHLIGHTERS TO

JSX - BEISPIEL

```
1 var dropdown =  
2   <Dropdown>  
3     A dropdown list  
4     <Menu>  
5       <MenuItem>Do Something</MenuItem>  
6       <MenuItem>Do Something Fun!</MenuItem>  
7       <MenuItem>Do Something Else</MenuItem>  
8     </Menu>  
9   </Dropdown>;  
10  
11 render(dropdown);
```


COMPONENT - INIT

```
1 import React, { Component } from 'react';
2 import { View, Platform } from 'react-native';
3
4 export default class Moin extends Component {
  ...
24 });
```

COMPONENT - RENDERING

```
1 import ...
3
4 export default class Moin extends Component {
...
10   render() {
11     return (
12       <View style={{ backgroundColor: 'red' }}>
13       </View>
14     );
15   }
16 };
```


COMPONENT - STATE

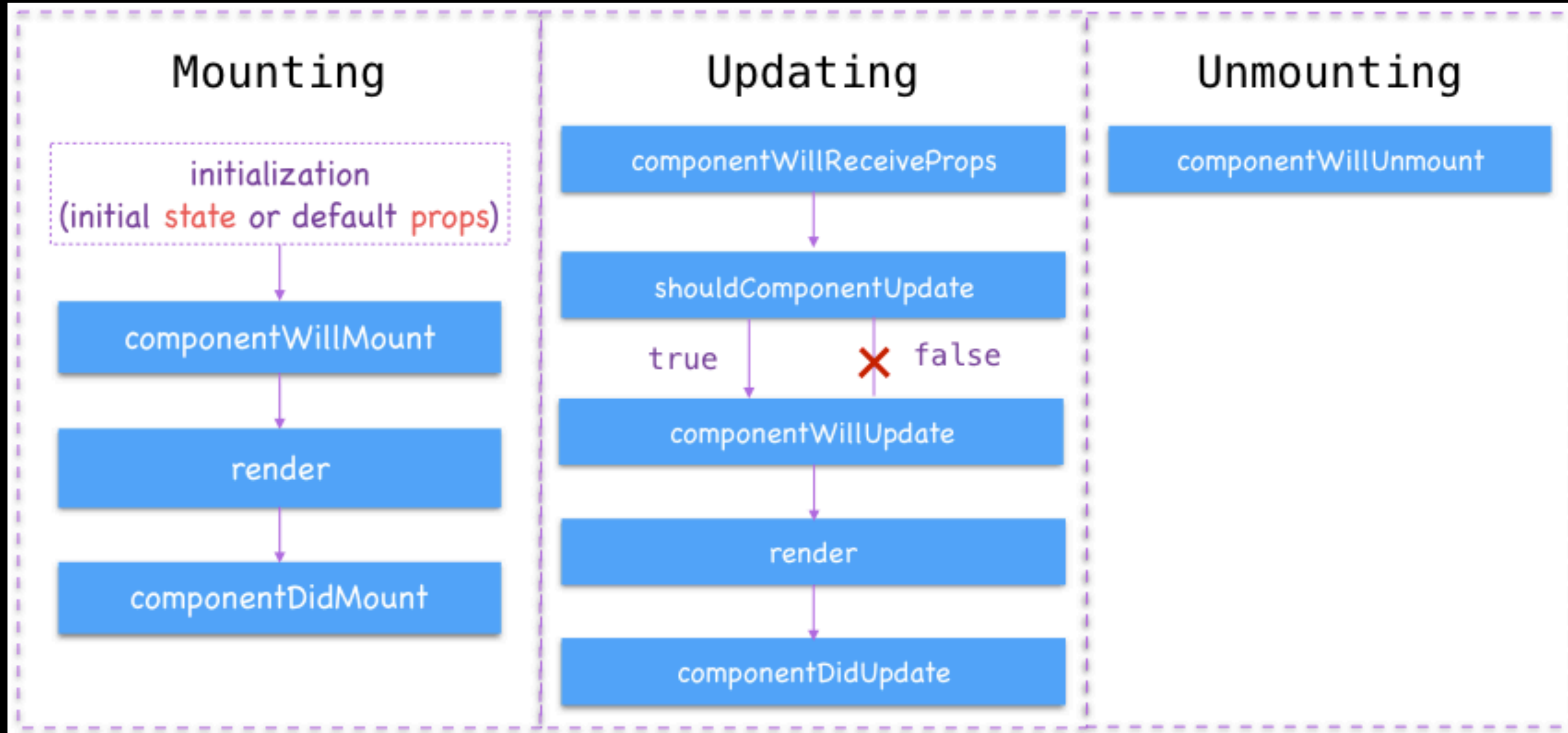
```
1 import ...
3
4 export default class Moin extends Component {
5
6     state = {
7         text: 'Hello',
8     }
9
10    render() {
11        return (
12            <View style={...}>
13                <Text>{this.state.text}</Text>
14            </View>
15        );
16    }
17 }
```

COMPONENT - LIFECYCLE

```
1 import .... Component {
5
10   componentWillMount() {
11     ...
12   }
13
14   componentDidMount() {
15     ...
16   }
17
18   componentWillReceiveProps(nextProps) {
19     ...
20   }
...
41 }
```

```
1 import .... Component {
5
22   componentWillUpdate(nextProps, nextState) {
23     ...
24   }
25
26   shouldComponentUpdate(nextProps, nextState) {
27     ...
28   }
29
30   componentWillUnmount() {
31     ...
32   }
33
34   render() {...
40   }
41 }
```


COMPONENT - LIFECYCLE



COMPONENT - UPDATE STATE

```
1 import ...
3
4 export default class Moin extends Component {
5
6     state = {
7         text: 'Hello',
8     }
9
10    componentDidMount() {
11        this.setState({
12            text: 'Test',
13        });
14    }
```

COMPONENT - PROPERTIES

```
1 import ...;
2
3 export default class Moin extends Component {
4
5     render() {
6         return (
7             <View style={styles.container}>
8                 <Text>{this.state.text}</Text>
9             </View>
10        );
11    }
12 }
```


COMPONENT - PROPERTIES

```
1 import ...;
2
3 export default class Moin extends Component {
4
5     render() {
6         return (
7             <View style={styles.container}>
8                 <SomeComponent
9                     value={this.state.text}>
10                 </SomeComponent>
11             </View>
12         );
13     }
14 }
```

COMPONENT - UPDATE STATE

```
1 import ...;
2
3 export default class SomeComponent extends
4 Component {
5
6   render() {
7     return (
8       <Text>
9         {this.props.value}
10      </Text>
11    );
12  }
13 }
```

PROPTYPES

COMPONENT - TOUCHABLE ELEMENTS

```
1 import ...;
2
3 export default class SomeComponent extends
4 Component {
5
6   render() {
7     return (
8       <TouchableOpacity onPress={this.props.onClick}>
9         ...
10      </TouchableOpacity>
11    );
12  }
13 }
```

COMPONENT - CHILD -> PARENT

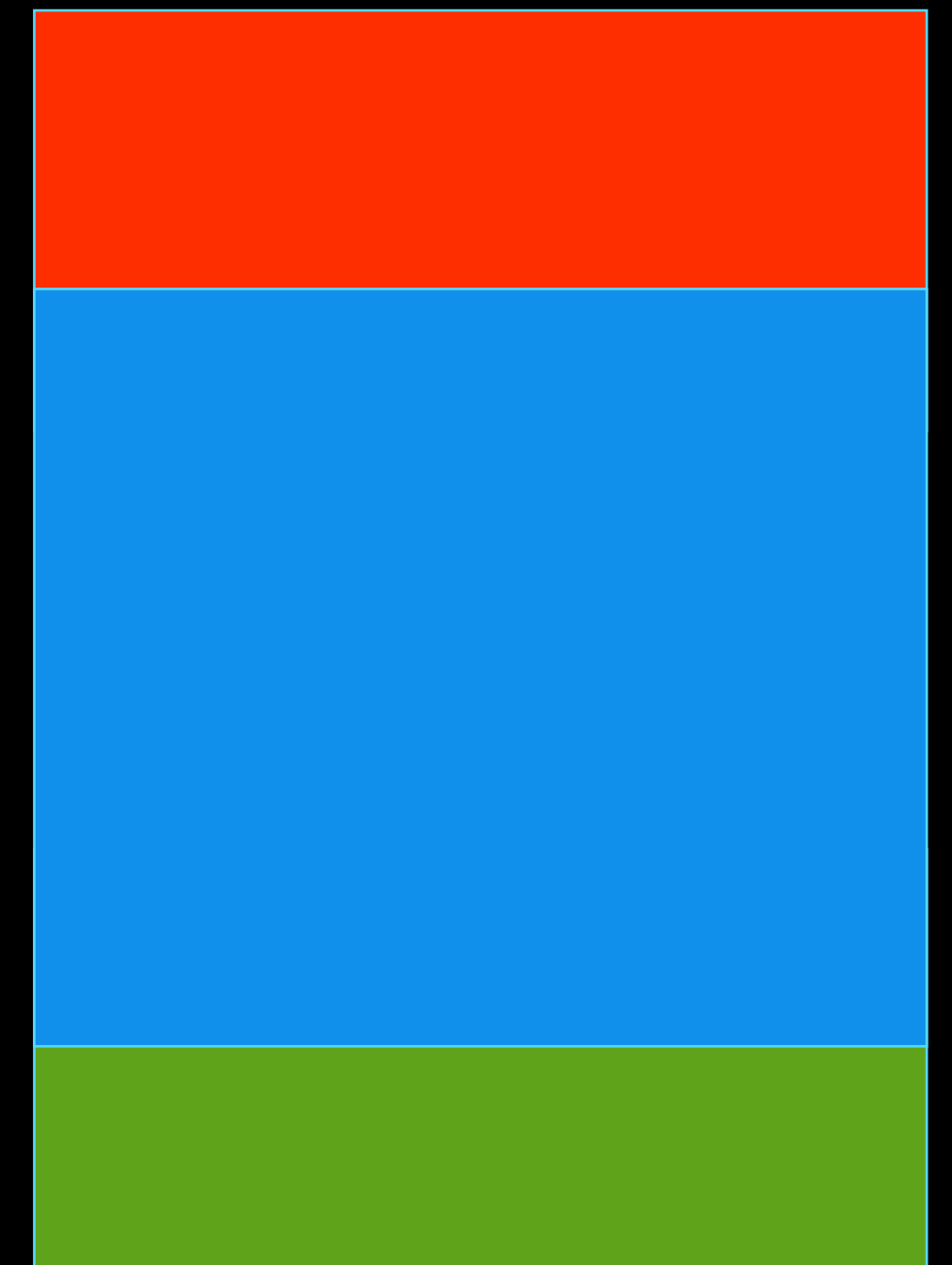
```
1 import ...;
2
3 export default class Moin extends
4 Component {
5
6   render() {
7     return (
8       <SomeComponent
9         onClick={() => this.setState({text: 'test'})}>
10       ...
11     </SomeComponent>
12   );
13 }
```


COMPONENT - DIMENSIONS, PLATFORM

```
1 import { Dimensions, Platform } from 'react-native'
2
3
4 export default class Moin extends Component {
5
6   render() {
7     return (
8       <Text>
9         {Dimensions.get('window').height}
10      </Text>
11      <Text>
12        {Dimensions.get('window').width}
13      </Text>
14      <Text>
15        {Platform.OS}
16      </Text>
17    );
18  }
19 }
```

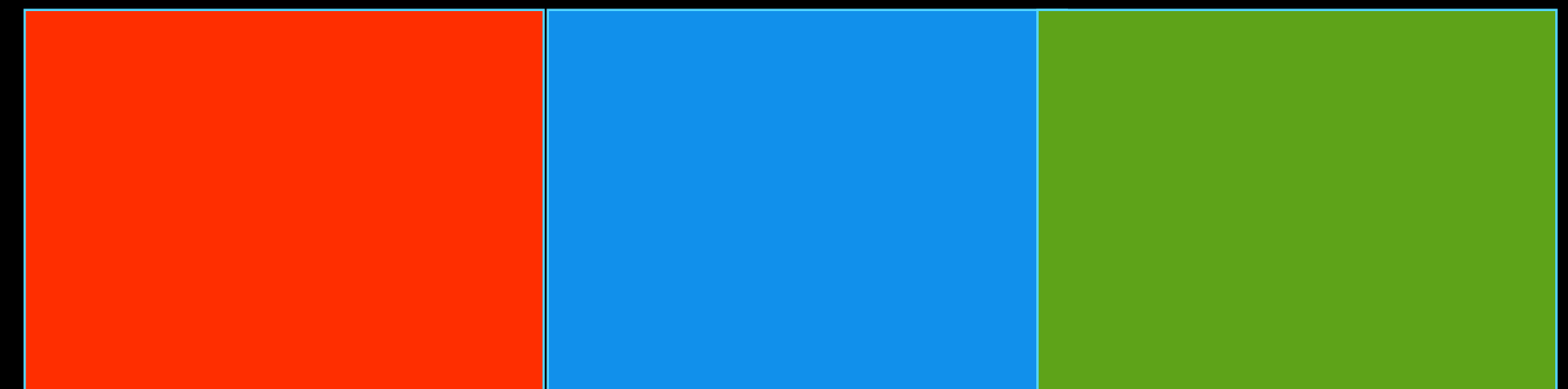
COMPONENT - FLEXBOX

```
1 <View style={{flex: 1}}>
2   <View style={{flex: 1, backgroundColor: 'red'}} />
4   <View style={{flex: 2, backgroundColor: 'blue'}} />
6   <View style={{flex: 1, backgroundColor: 'green'}} />
8 </View>
```



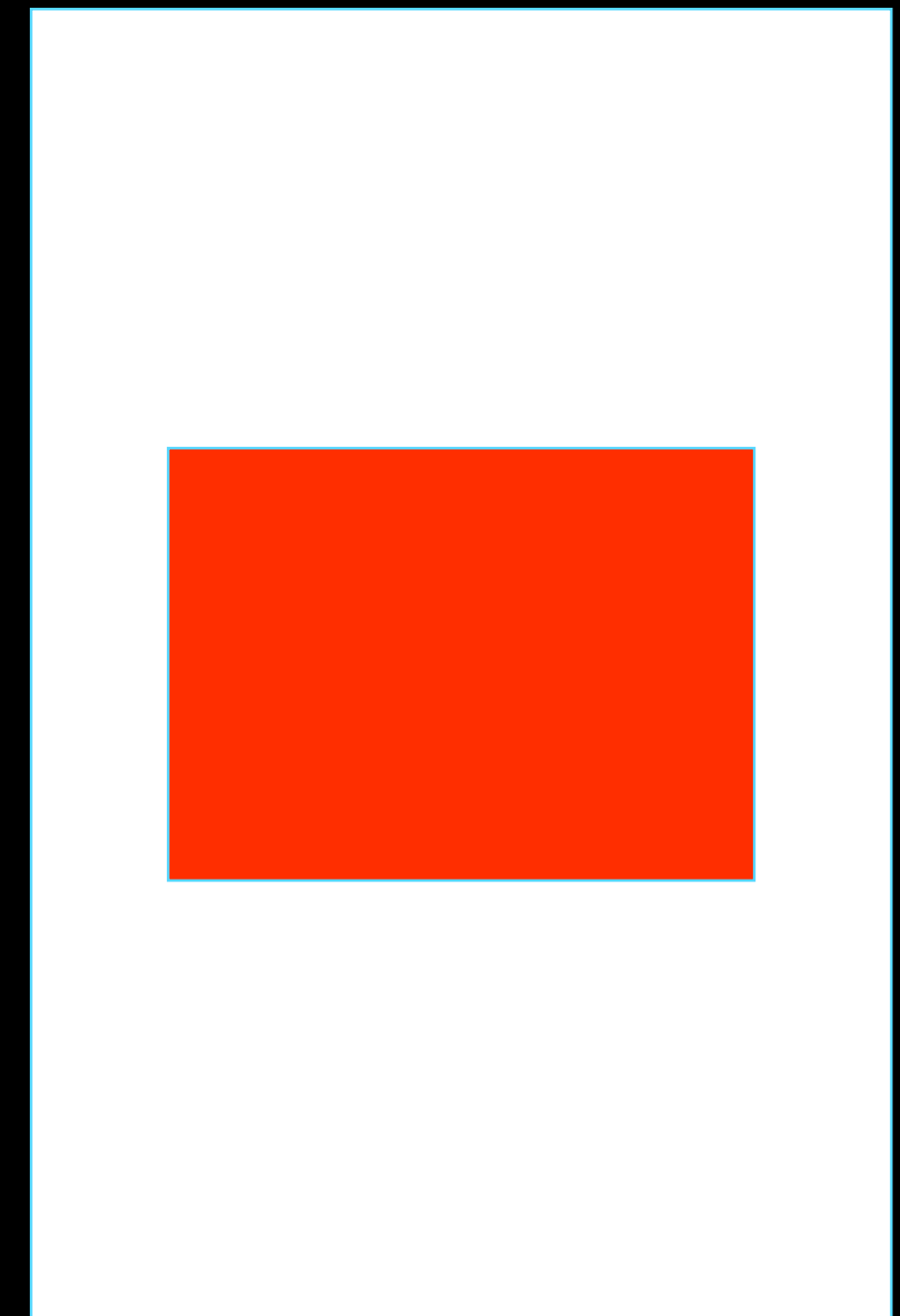
COMPONENT - FLEXBOX

```
1 <View style={{flex: 1, flexDirection: 'row'}}>
2   <View style={{flex: 1, backgroundColor: 'red'}} />
4   <View style={{flex: 1, backgroundColor: 'blue'}} />
6   <View style={{flex: 1, backgroundColor: 'green'}} />
8 </View>
```



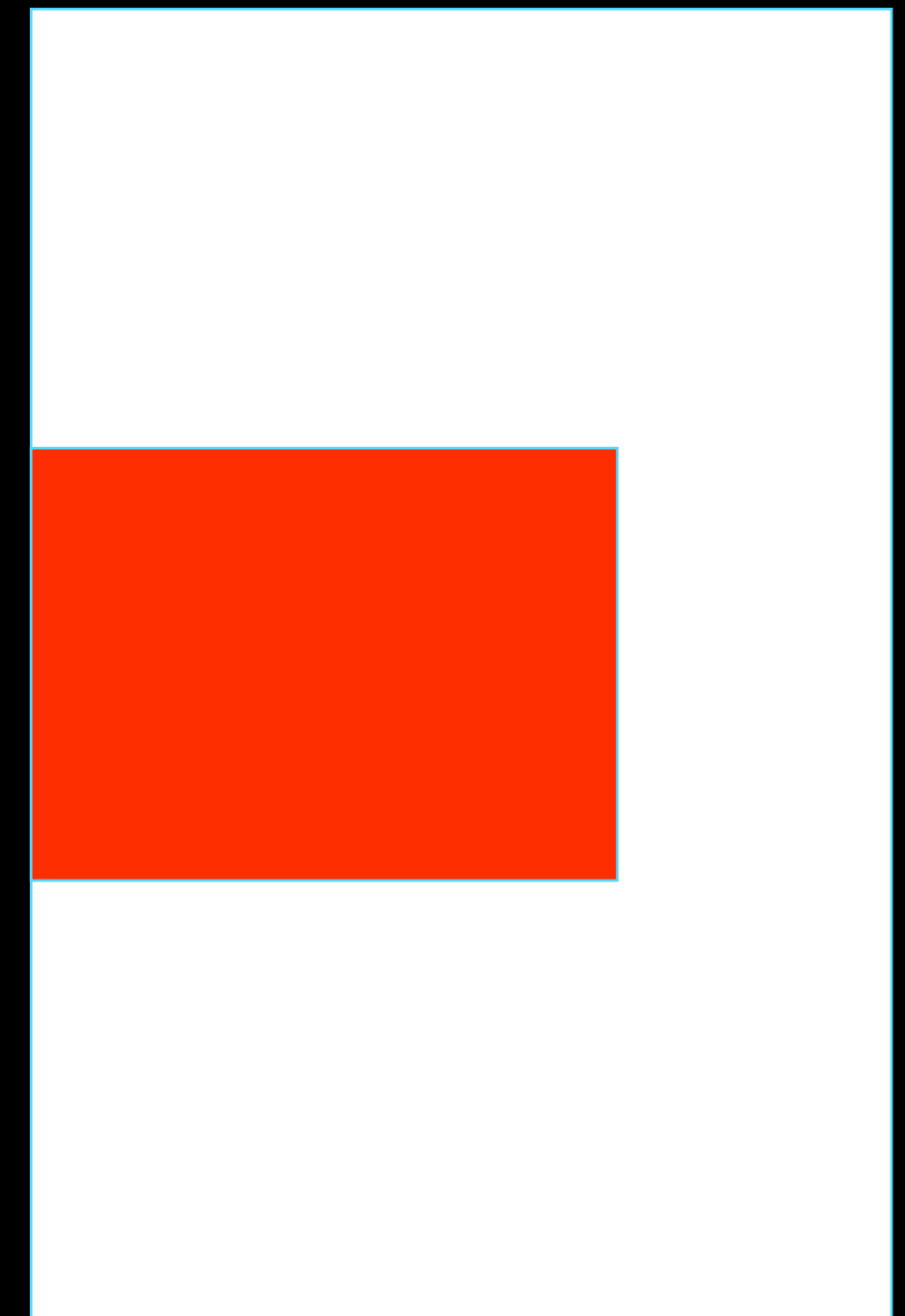
COMPONENT - FLEXBOX

```
1 <View style={{flex: 1, alignItems: 'center', justifyContent: 'center'}}>  
3   <View style={{width: 40, height: 40, backgroundColor: 'red'}} />  
5 </View>
```



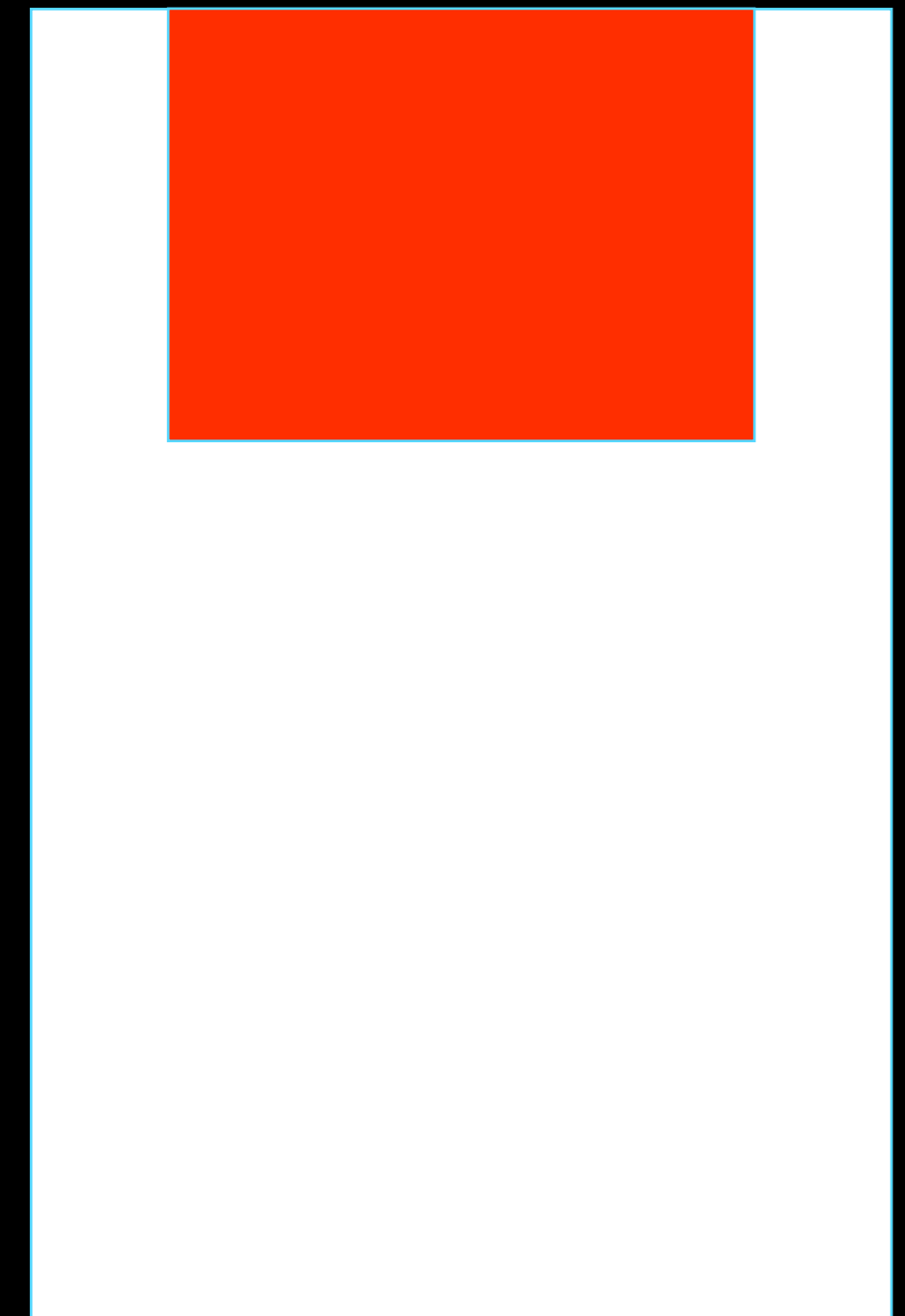
COMPONENT - FLEXBOX

```
1 <View style={{flex: 1, alignItems: 'flex-start', justifyContent: 'center'}}>  
3   <View style={{width: 40, height: 40, backgroundColor: 'red'}} />  
5 </View>
```



COMPONENT - FLEXBOX

```
1 <View style={{flex: 1, alignItems: 'center', justifyContent: 'flex-start'}}>  
3   <View style={{width: 40, height: 40, backgroundColor: 'red'}} />  
5 </View>
```



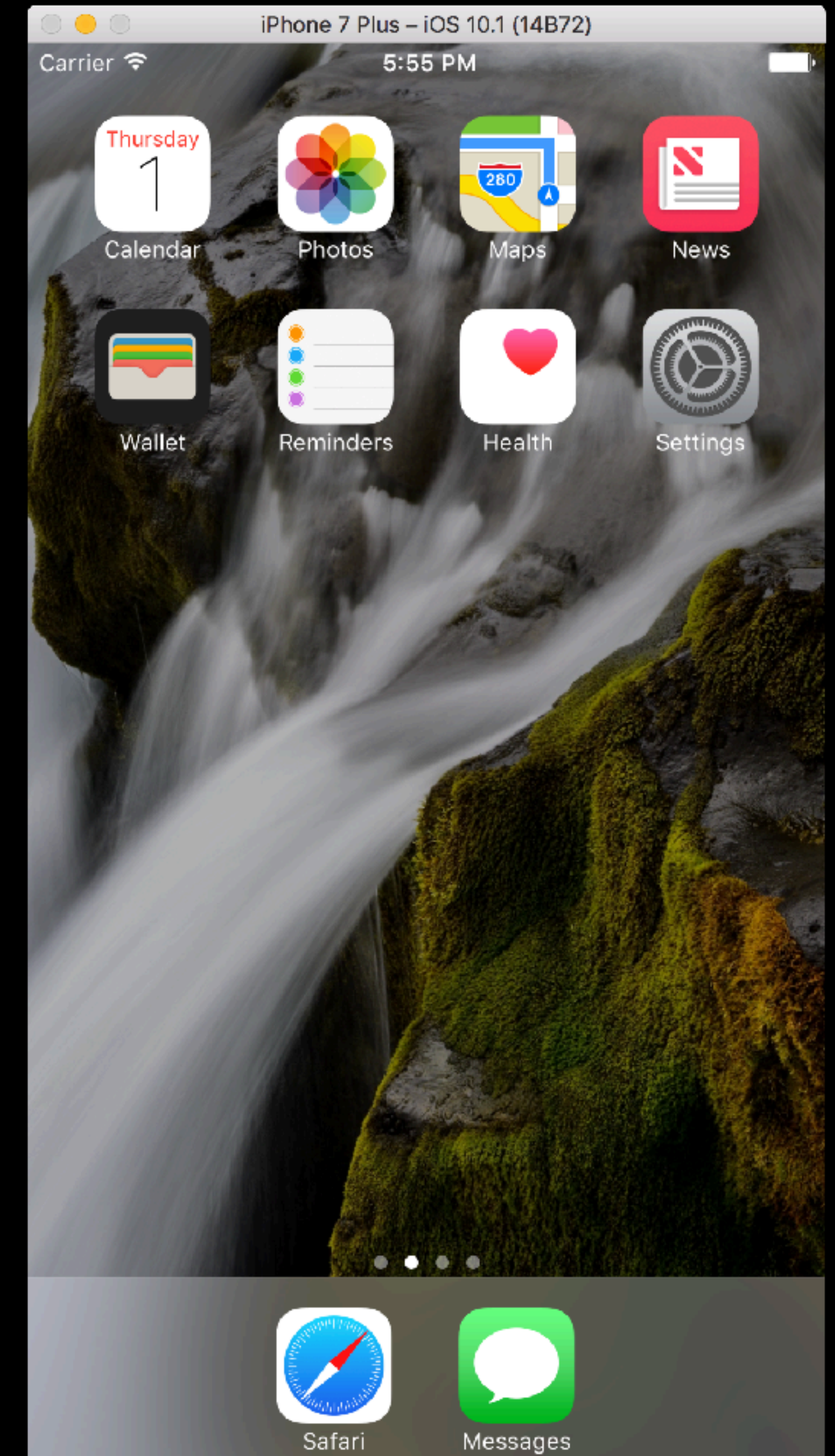
COMPONENT - LISTVIEW

```
1 import { ListView } from 'react-native';
2
3 export default class ChatMessages extends Component {
4
5   state = {
6     dataSource: new ListView.DataSource({rowHasChanged: (r1, r2) => r1 !== r2}),
7   };
8
9   renderRow = (data) => {
10     return <View />
11   };
12
13   render() {
14     let { dataSource } = this.state;
15     return (
16       <ListView
17         dataSource={dataSource.cloneWithRows(this.props.list)}
18         renderRow={this.renderRow}
19       />
20     );
21   }
22 }
23 }
```

Nun aber los..... 

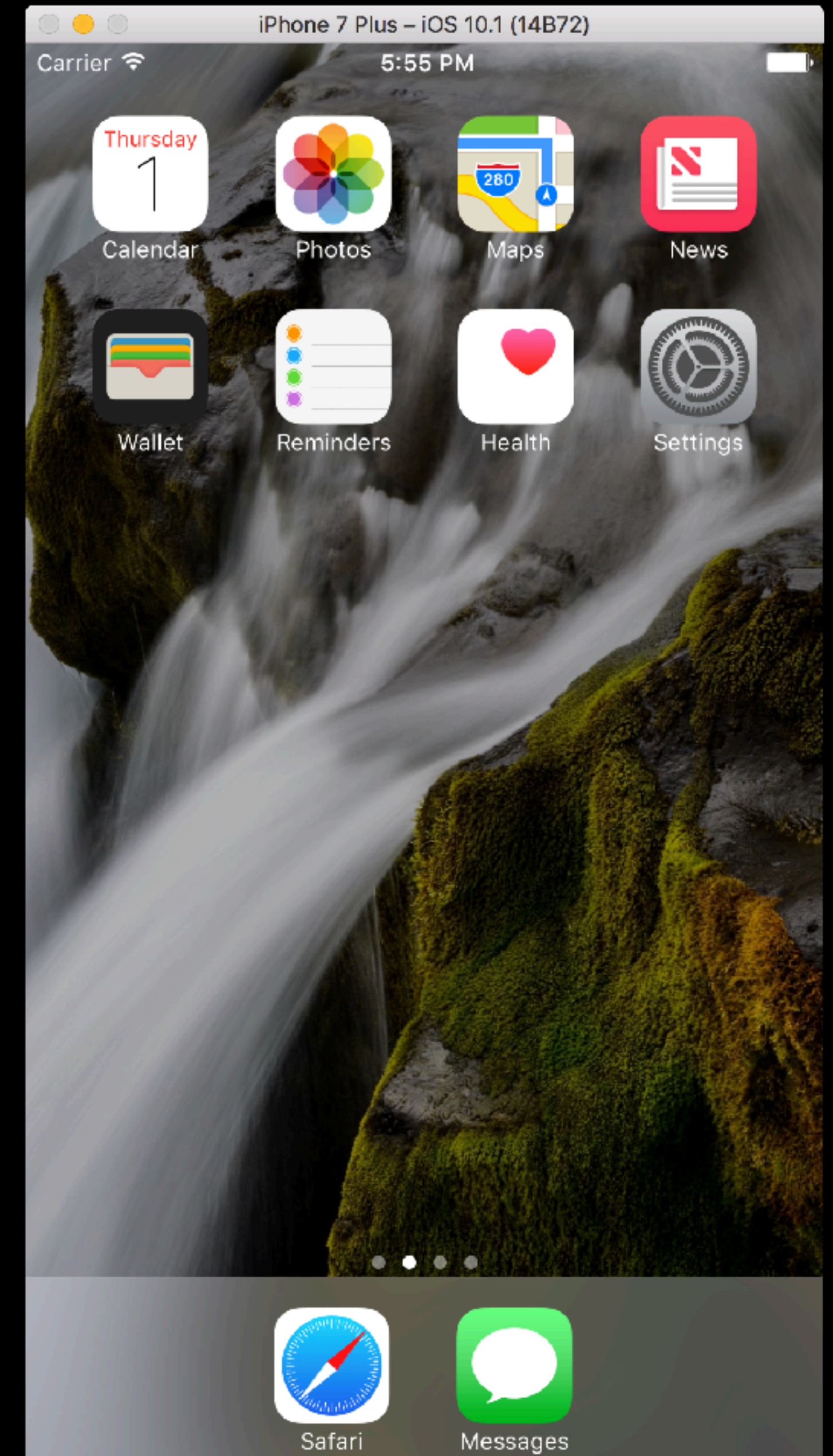
HANDS-ON BRIEFING

- Ihr sollt eine Chat-App entwickeln
- Auswählen eines Benutzernamens
- Ansehen aller Nachrichten
- Verschicken einer Nachricht im eigenen Namen



HANDS-ON BRIEFING

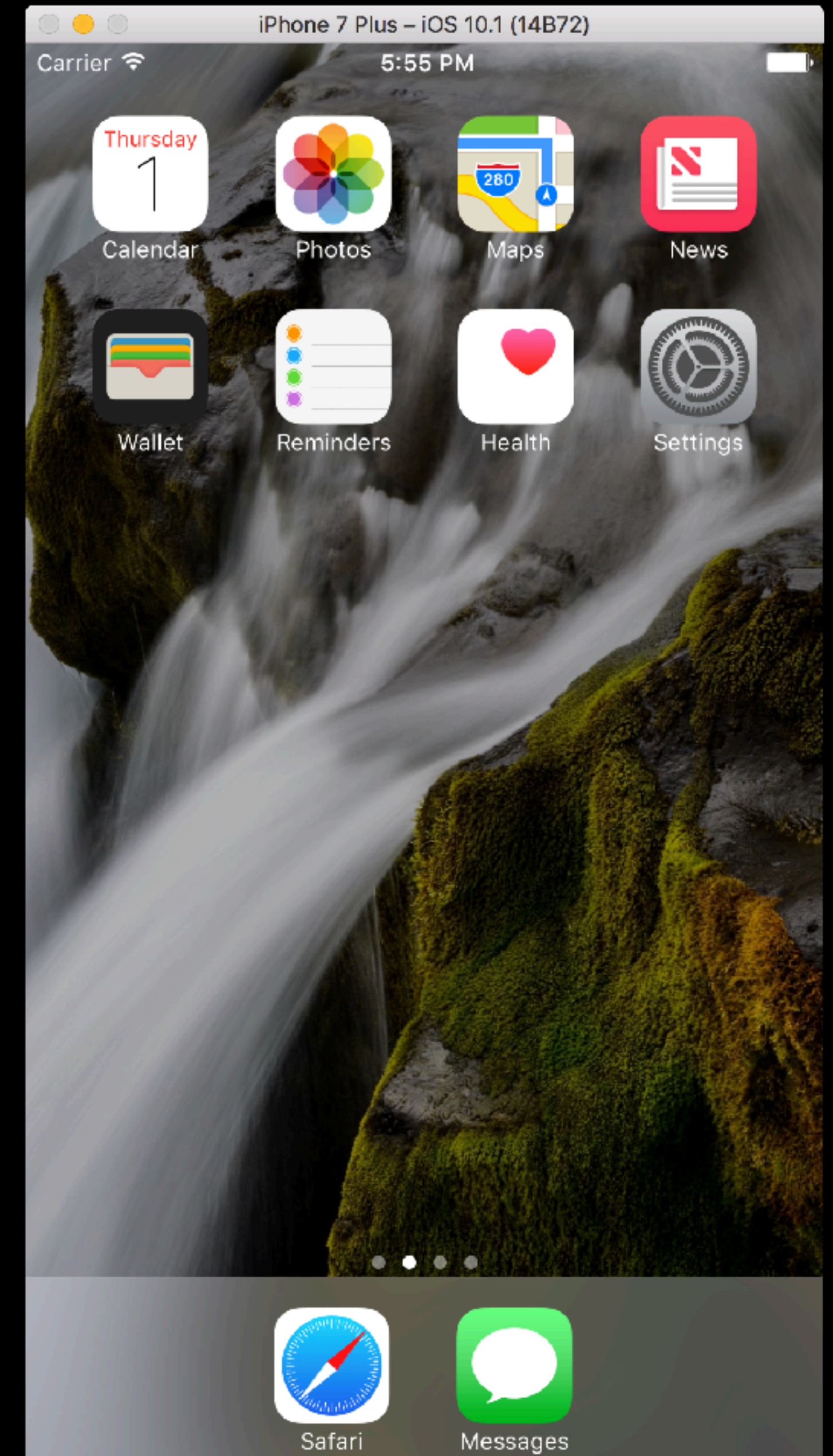
- Backend wird von uns gestellt
 - <https://moin-backend.herokuapp.com>
- möglichst Gruppenarbeit
 - 2-3 Personen pro Rechner
- Sebastian und ich "kommen rum"



HANDS-ON BRIEFING

- **Step 1:** Checkout Template Repo
- <https://github.com/AppWithUs/moin-template>

```
git clone https://github.com/AppWithUs/moin-template.git
```



HANDS-ON BRIEFING

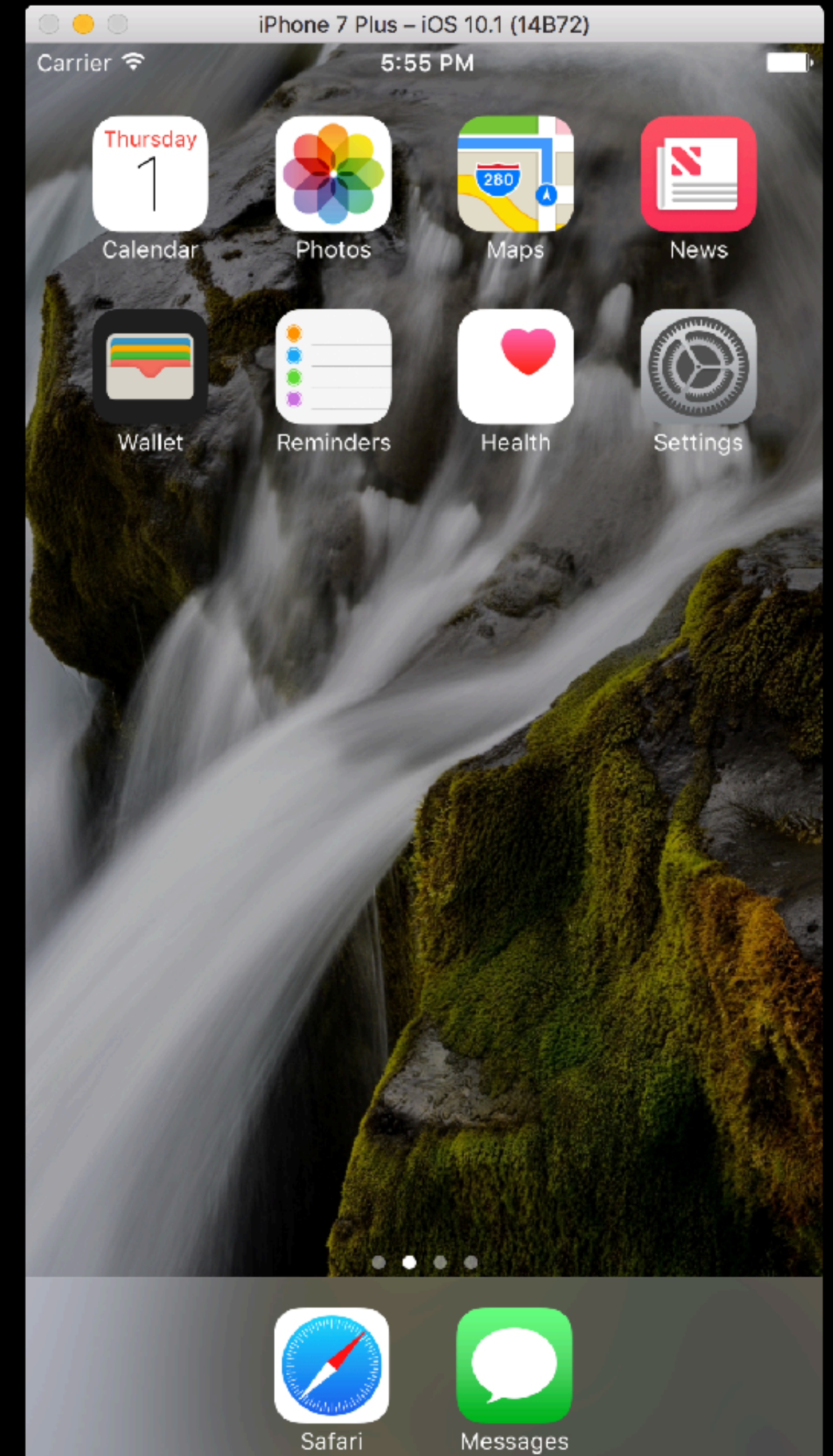
- **Step 2:** Installiert alle Dependencies
 - (stehen auch in der README.md)

```
# install homebrew
/usr/bin/ruby -e "$(curl -fsSL https://
raw.githubusercontent.com/Homebrew/install/master/
install)"
```

```
brew install node watchman
```

```
npm install -g react-native-cli
```

```
npm install
```



HANDS-ON BRIEFING

- **Step 3:** Im Simulator/Emulator installieren

iOS

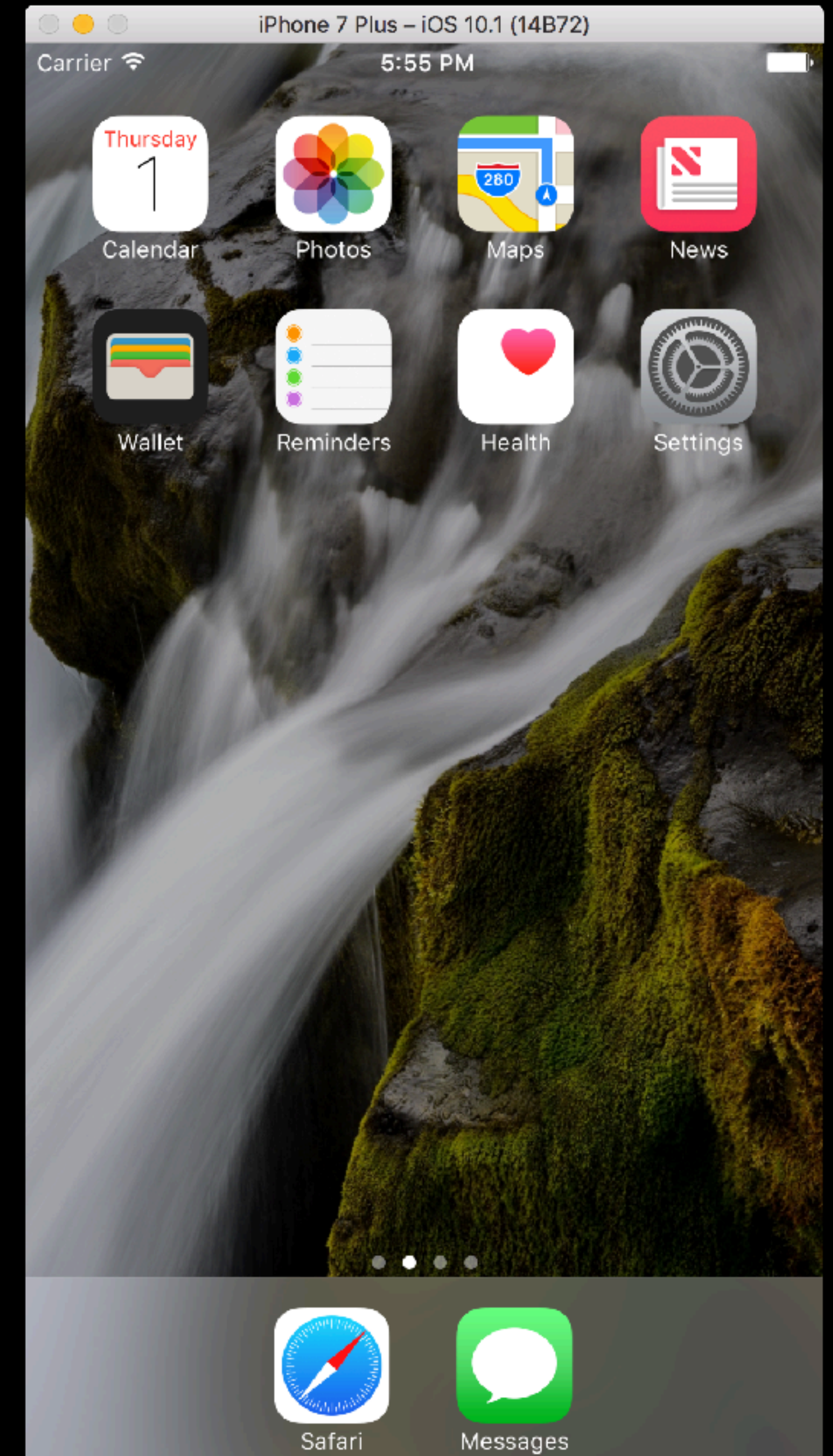
```
react-native run-ios
```

Android

```
# either connect your  
Android device via USB, or  
start an emulator
```

```
android avd
```

```
react-native run-android
```



HANDS-ON BRIEFING

- **Step 4:** SetNameView
- Stolperfallen:
 - Keyboard Avoiding View
 - TextInput -> State
- Pluspunkte:
 - "Go" Button nur klickbar wenn Name ausgefüllt



HANDS-ON BRIEFING

- **Step 5:** ChatView
- Components:
 - ChatMessages
 - ChatInput
- Stolperfallen:
 - PullIntervall
 - Properties / State
 - ScrollToBottom



IHR HABT NOCH

3 Stunden...

DEMO

- Jeder zeigt seine Lösung und berichtet kurz!
 - 5 Minuten pro Team

DEBRIEF

- Wir wollen wissen:
 - Was habt ihr gelernt?
 - Was lief gut?
 - Was lief nicht so gut?
 - Was sollte das nächste Mal anders sein?

改善

WE'RE HIRING

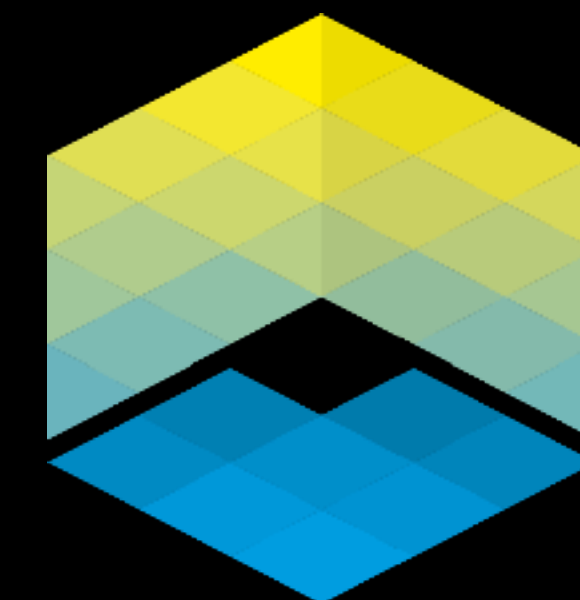


Wir suchen weiter nach schlaunen Köpfen, die gerne Hand anlegen und Lust haben, mit uns zusammen die Welt der mobilen Endgeräte zu erweitern.

Neben **Festanstellungen**, haben wir auch Platz für **Werkstudenten**, **Seminararbeiten**, **Bachelor-Arbeiten**, ...

Schaut doch einfach mal auf:

appwith.us/jobs



VIELEN DANK