

React Native Introduction

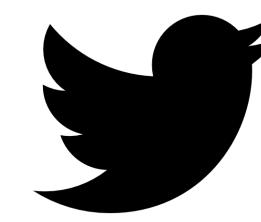
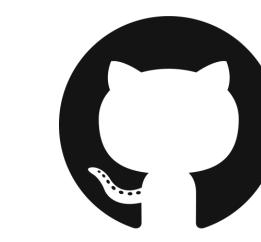
React & React Native Meetup Bonn

Juli, 9th 2019

Christoph Jerolimov



Christoph Jerolimov

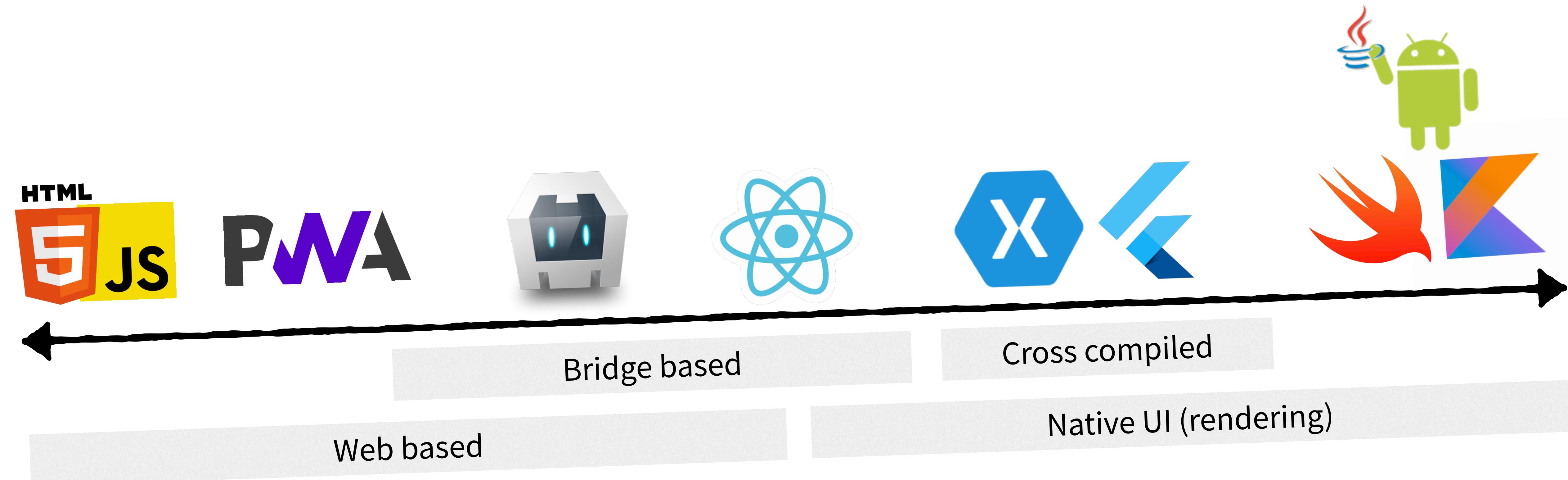


@jerolimov

Years ago: Java Backends, Web Stuff,
Cordova and Native Apps for Android and iOS

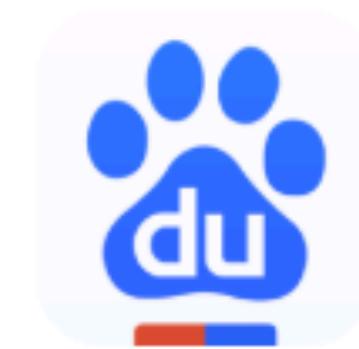
**Since several years React Native
and still some backend stuff**

WHY





B



“Write once, run anywhere.”

– Origin by Java, and later HTML 5

“Learn once, use anywhere.”

— react



React Native

Build native mobile apps using JavaScript and React

[Get Started](#)[Learn the Basics](#)

Build native mobile apps using JavaScript and React

React Native lets you build mobile apps using only JavaScript. It uses the same design as React, letting you compose a rich mobile UI using declarative components.

```
import React, {Component} from 'react';
import {Text, View} from 'react-native';

class HelloReactNative extends Component {
  render() {
    return (
      <View>
        <Text>
          If you like React, you'll also like React Native.
        </Text>
        <Text>
```

CODE

React

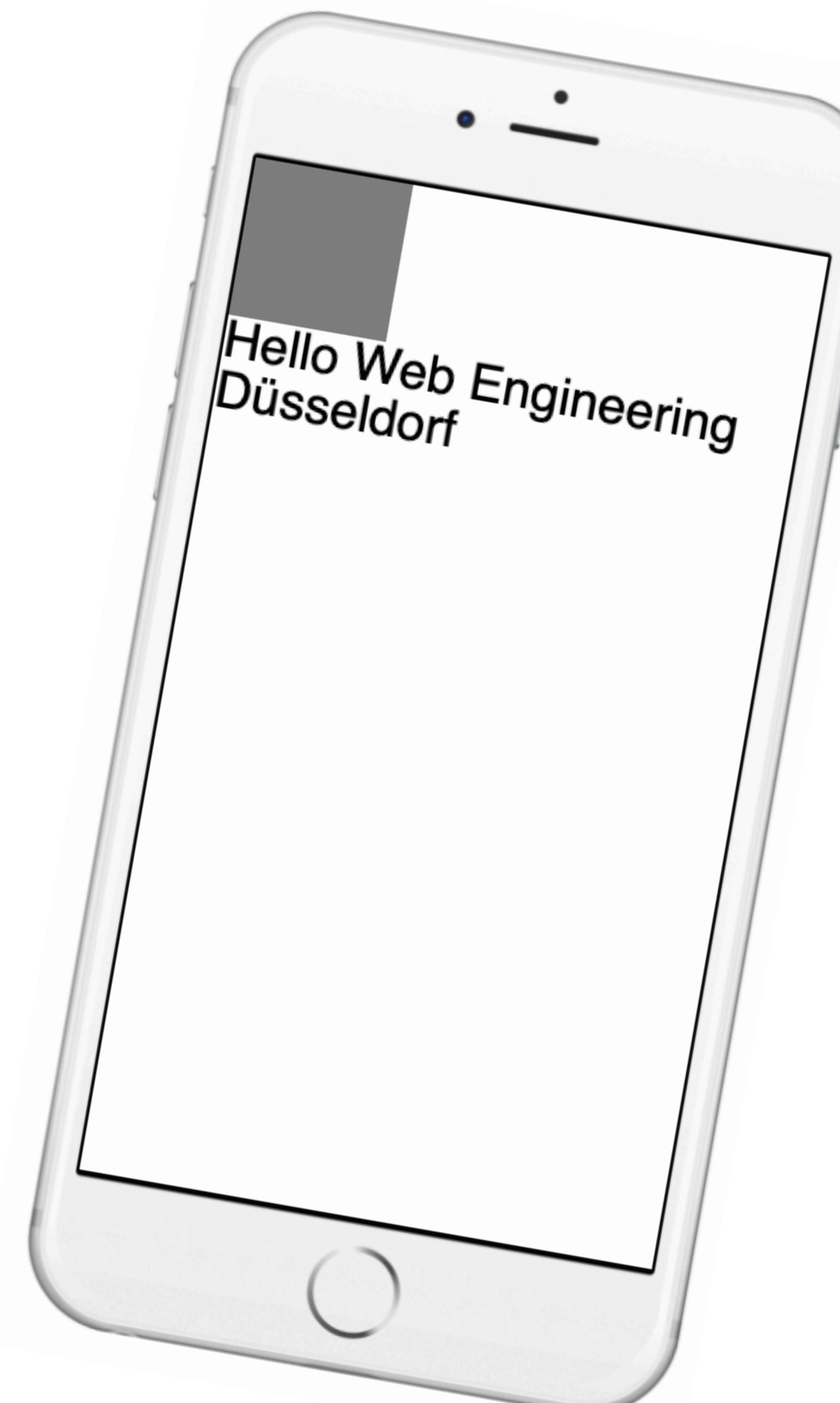


- **Reuse React.js to render a view hierarchy**
- But render ***native views***
- **Increase developer experience**
- Allows you to integrate React Native into your App (brownfield)
- Allows you to integrate native components into your React Native app (future proof)

HelloWorld.js

```
import React, { Component } from 'react';
import { Image, Text, View } from 'react-native';

export default class HelloWorld extends Component {
  render() {
    return (
      <View style={{ alignItems: 'center' }}>
        <Image
          source={{ uri: 'https://...' }}
          style={{ width: 200, height: 200 }}
        />
        <Text>Hello World</Text>
      </View>
    );
  }
}
```

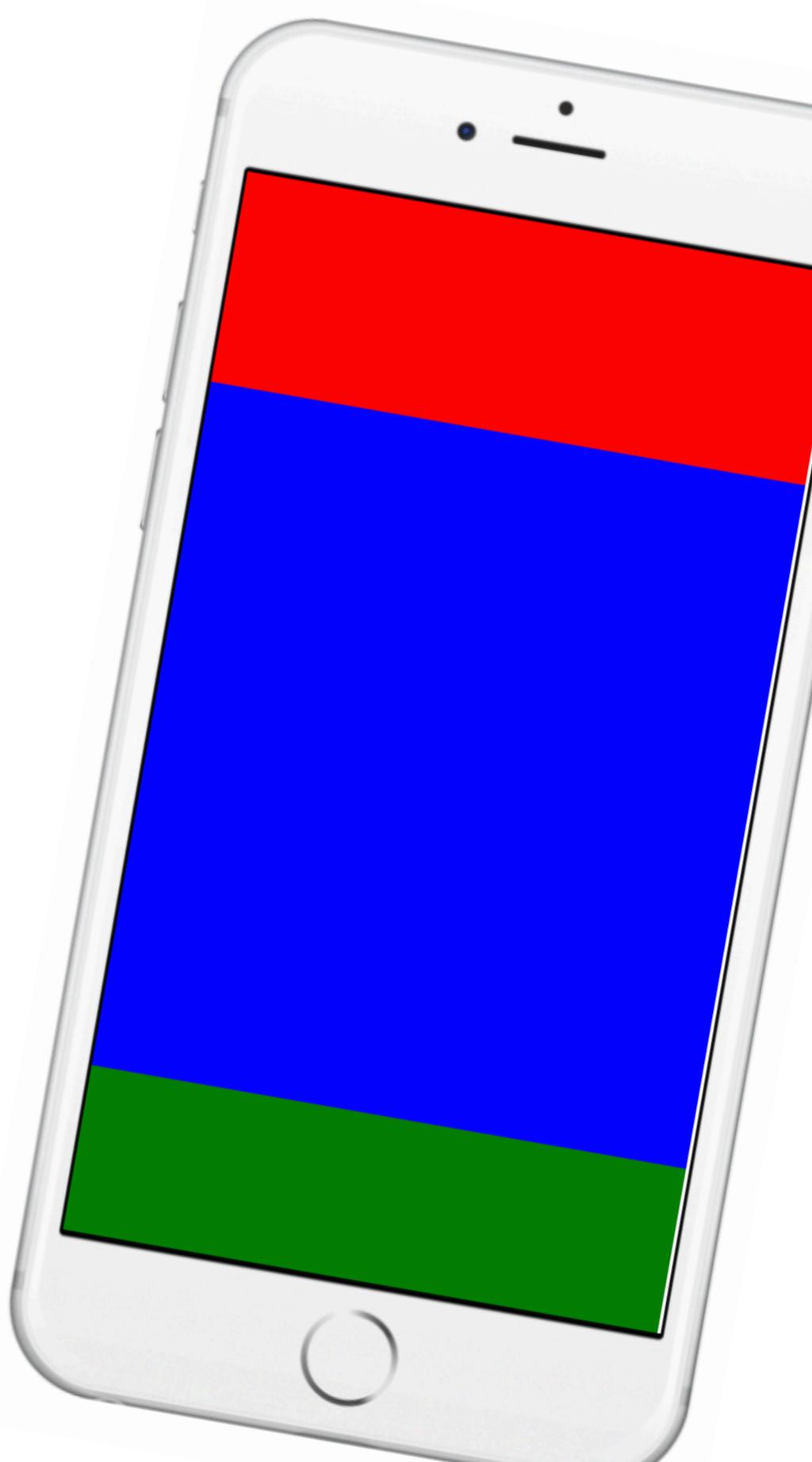


Flexbox

```
// Grow 100% with childs 50%, 30% and 20%
<View style={{ flex: 1, flexDirection: 'row' }}>
  <View style={{ flex: 0.5, backgroundColor: 'red' }} />
  <View style={{ flex: 0.3, backgroundColor: 'blue' }} />
  <View style={{ backgroundColor: 'green' }} />
</View>;
```

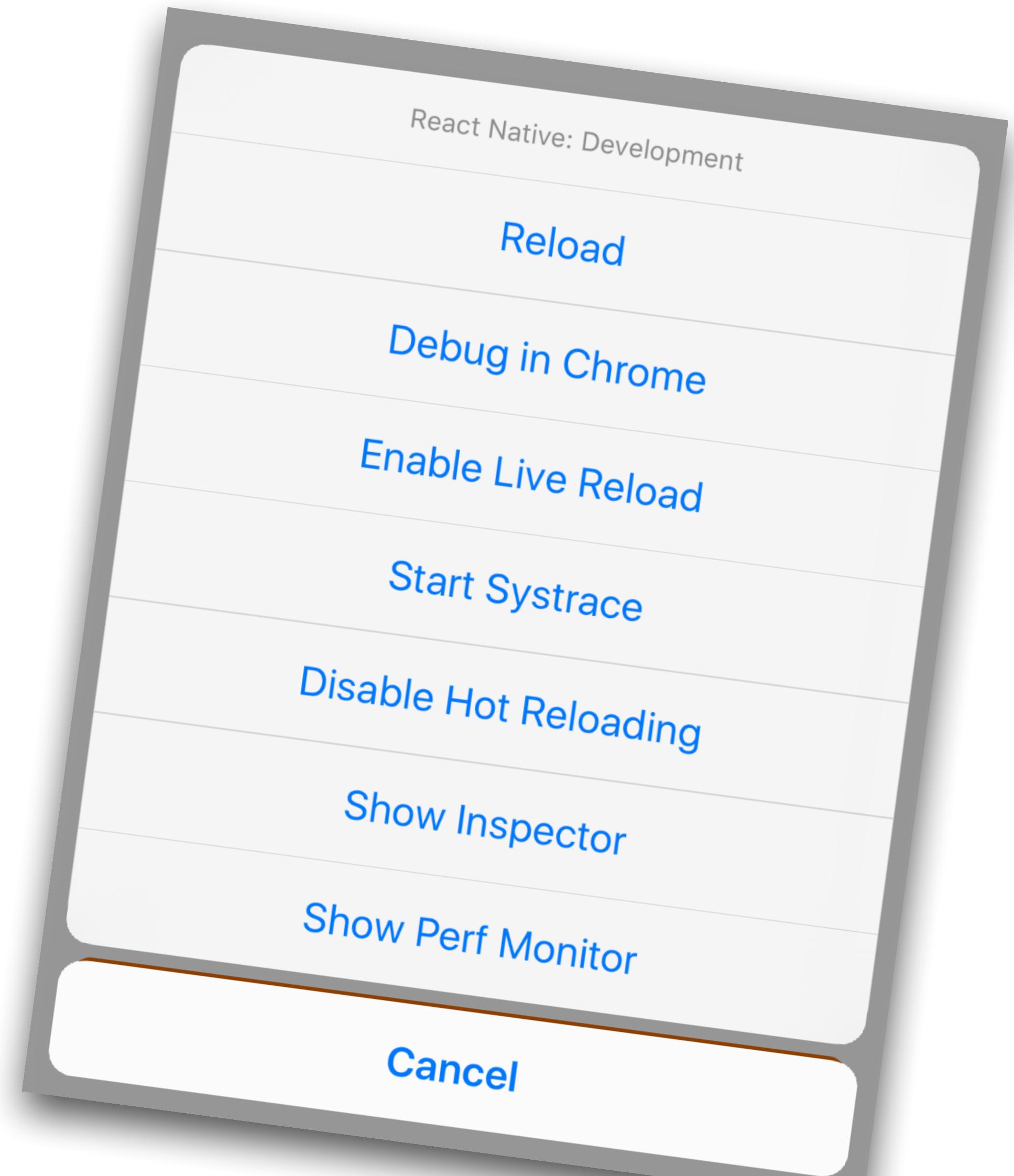
// Grow 100% where first and last child is fix

```
<View style={{ flex: 1 }}>
  <View style={{ height: 64, backgroundColor: 'red' }} />
  <View style={{ backgroundColor: 'blue' }} />
  <View style={{ height: 50, backgroundColor: 'green' }} />
</View>;
```

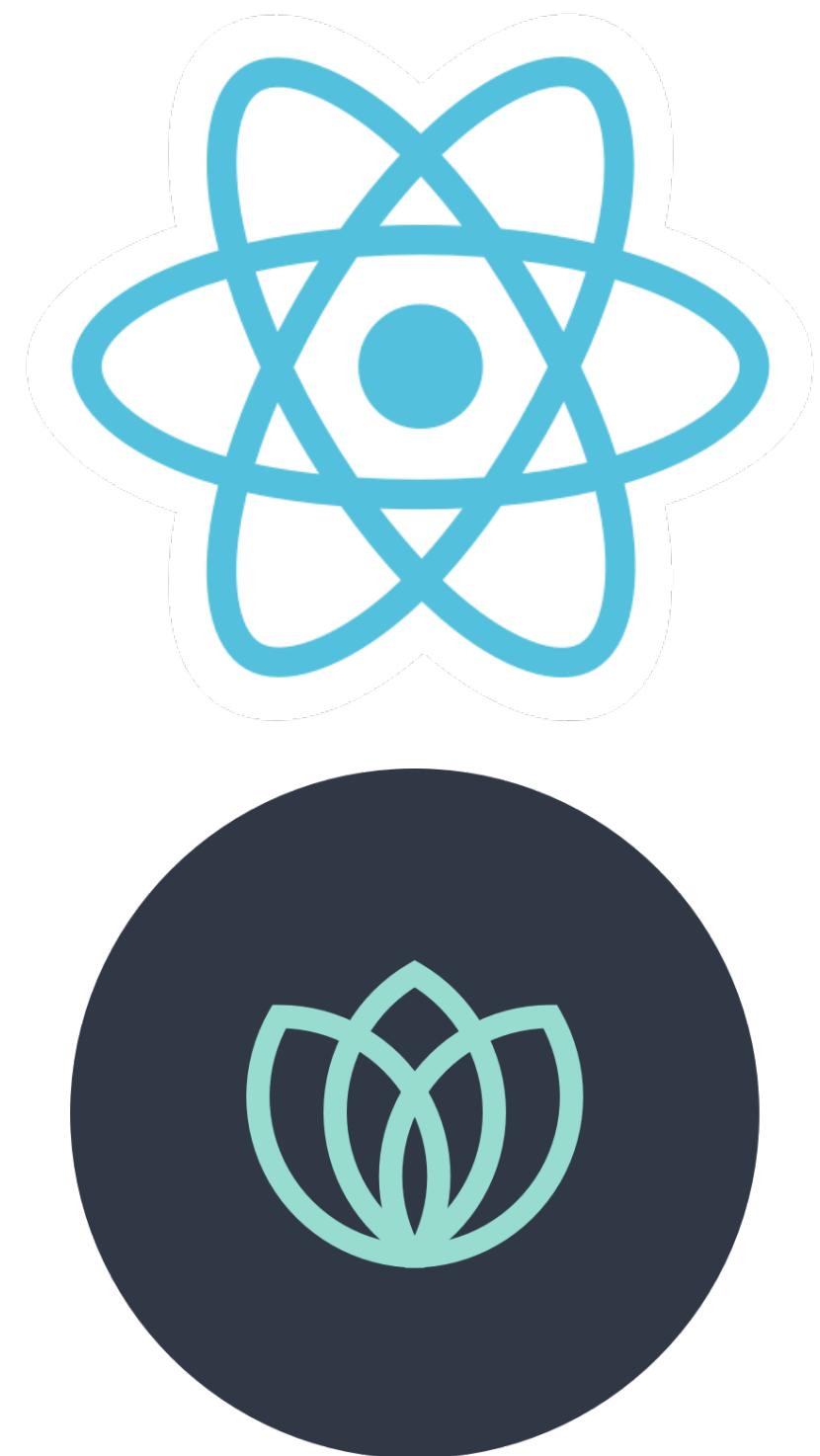


Developer Experience? ☀️ D

- “HTML- & CSS-like” => JSX + Flexbox
- (Fast) Reload your app (⌘R)
- (Auto) Live Reload & Hot-Reloading
- Debugger, UI Inspector, Profiling
-



HOW



React & React-native

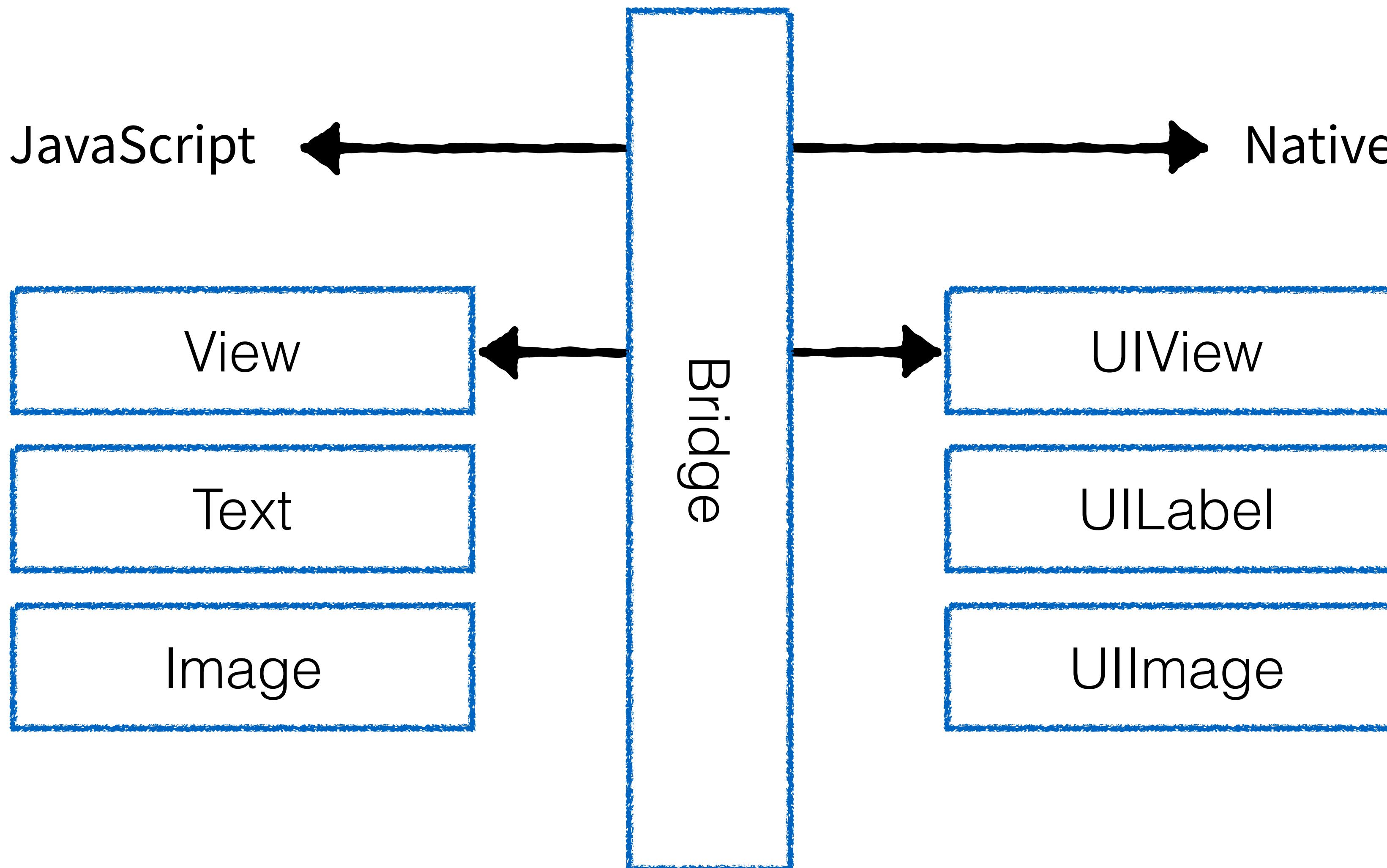
Flexbox Layout



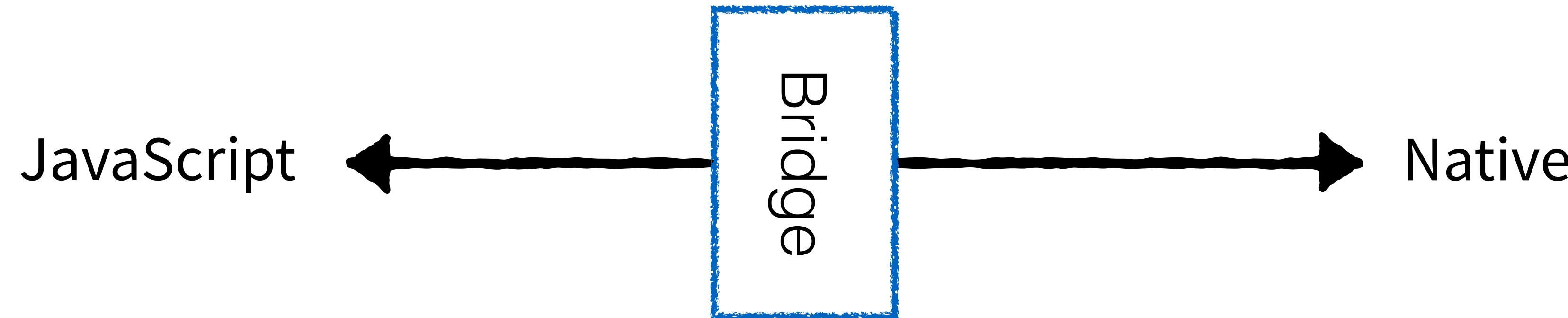
How it works?

- “**Components**” wrap native UI elements
`<View ... />`
- “**CSS**” re-implementation, incl. **Flexbox**
`width: 100, height: 100, backgroundColor: 'red', ...`
- “**Modules**” wrap native APIs
 - Provide some **polyfills**
(fetch for Network requests, geolocation for Location tracking, etc.)

How does it work?



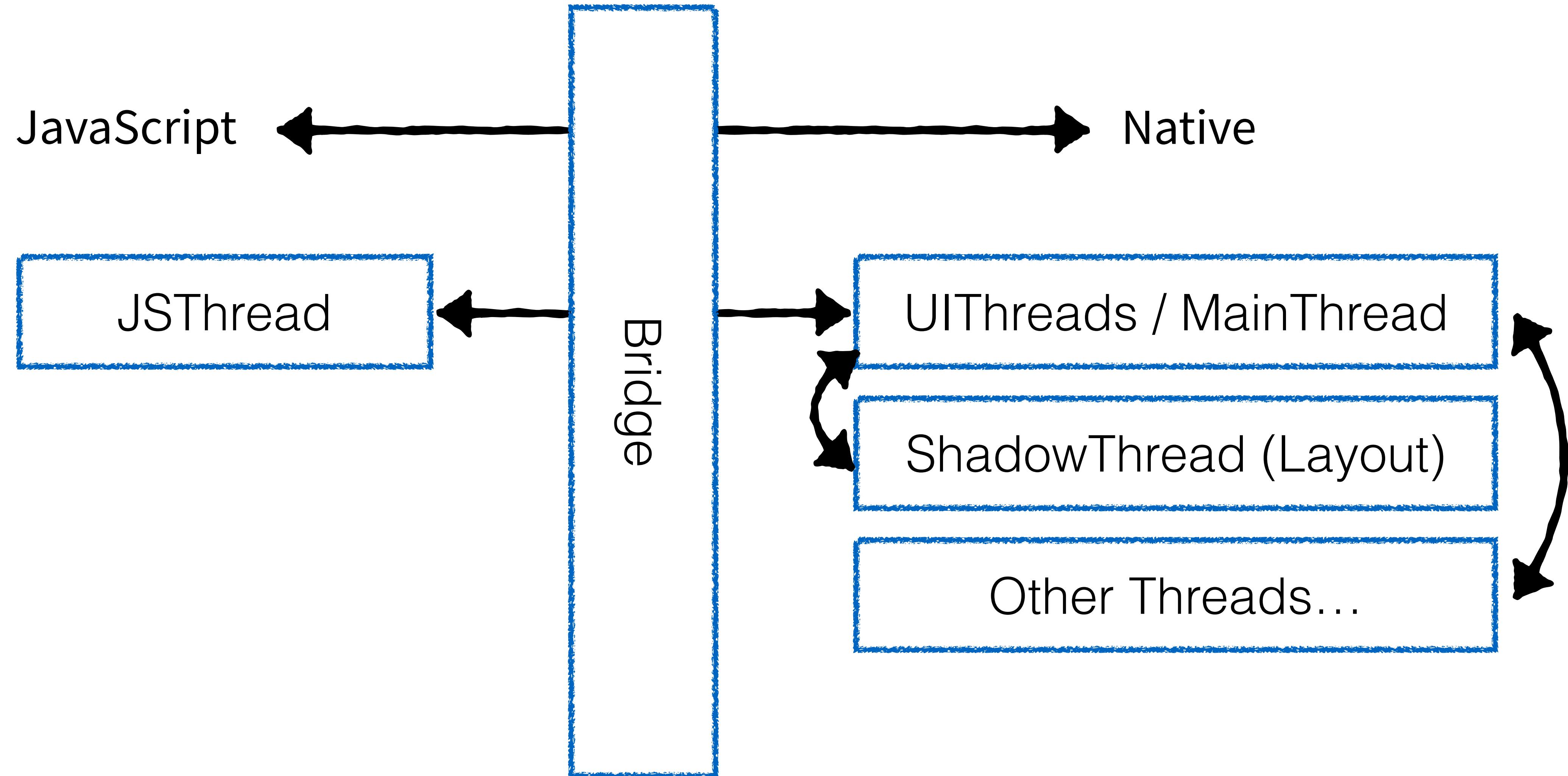
How does it work?



Transfer serializable data (JSON)

For each view change, touch event, etc.

Threads



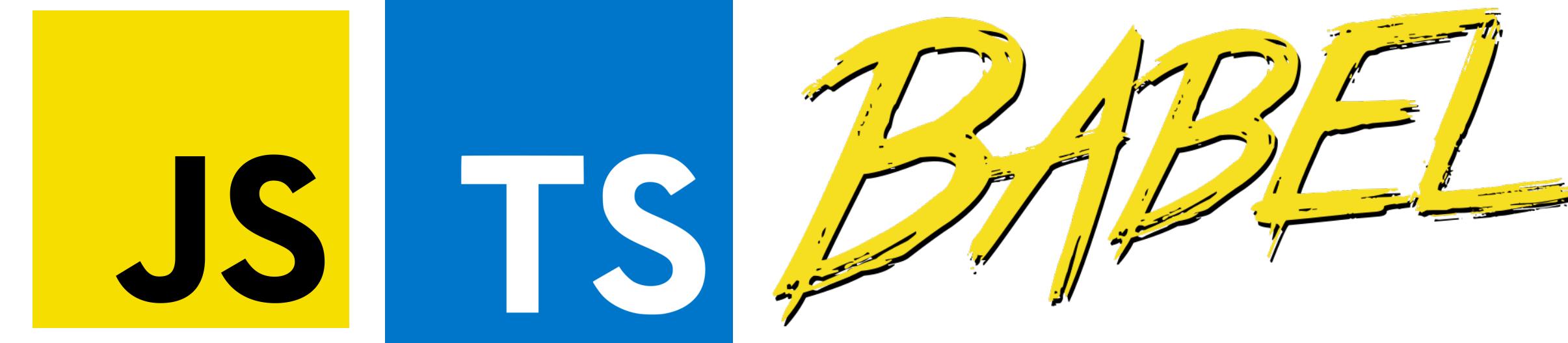
Project Structure



Dep. Structure



How it works?



- You write “modern” JavaScript (or TypeScript)
- *Babel* transform your sources (ES 6 and more ...)
- While developing the app fetches the compiled JS from **local http server**
- In a production app, the app contains a **minified JS bundle**
- JS was executed in a **minimal “JavaScript VM” with injected polyfills**
(JavaScriptCore is part of the WebKit OSS project)

**GETTING
STARTED**

How to get started?

- 🔥 react-native
- ⚡ Expo
- ⚓ react-navigation
- 👤 react-native-community

<https://facebook.github.io/react-native/blog/2018/11/01/oss-roadmap>
<https://github.com/react-native-community/discussions-and-proposals/issues/6>
<https://github.com/react-native-community/discussions-and-proposals/issues/63>

Getting started

- If you are familiar with Android or iOS development, try pure React Native
- If you want just prototype and/or have no experience with Android/iOS, try  **Expo**.io
Expo is a CLI and cloud based react-native infrastructure with a “Viewer” app
- Use react-navigation as navigation library.
- If you are a Designer, take a look at <https://designcode.io/>
(for the upcoming React Native Course!)
- If you are a Developer: <https://egghead.io/browse/frameworks/react-native>

Pros

- Share knowledge (with the web team)
(and maybe source code as well)
- Single codebase for Android and iOS
- Great layout system
- Reuse of the JavaScript ecosystem
- Fast iteration
- Native UI/UX

Cons

- More complexity in one project
- Still many complexity from the native parts
(Xcode and Android SDK)
- Less stable than pure native
- Missing some core technique like form validation, navigation

More resources

- React-Native Docs & Blog
<http://facebook.github.io/react-native/>
- React Navigation!!
<https://reactnavigation.org/>
- If your interested in Redux, try the Redux Starter Kit
<https://redux-starter-kit.js.org/>
- React-Native Newsletter
<http://reactnative.cc/>
- React Conf 2016
<https://www.youtube.com/playlist?list=PLb0IAmt7-GS0M8Q95Rlc2lOM6nc77q1Y>
- React Conf 2017
<https://www.youtube.com/playlist?list=PLb0IAmt7-GS3fZ46lGFirdqKTlxlws7e0>
- “awesome-react-native” link list
<https://github.com/jondot/awesome-react-native>

**LIVE
DEMO**

Q&A