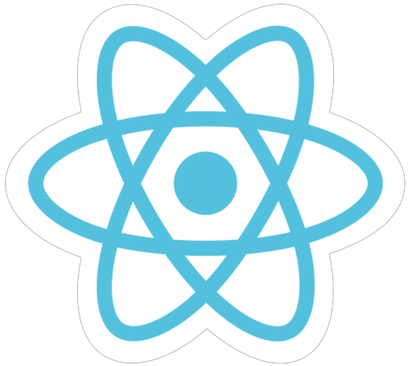


React Native Introduction

Web Engineering Düsseldorf

February, 4th 2019

Christoph Jerolimov

Hi! I’m Christoph Jerolimov 👍

GitHub / Twitter @jerolimov

OpenSource addicted

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

Last years: React Native and some backend stuff

WHY

Motivation and Requirements

- **Share code and know how** (with the bigger react team)
- Improve the **Developer Experience** (long build times, no instant feedback)
- A better experience than with WebViews

Idea React 

- Reuse React.js to **render a view hierarchy**
- Render ***native* view components**
- Allows you to integrate React Native **into your native App**
- Allows you to integrate native components **into your RN app**

React Native

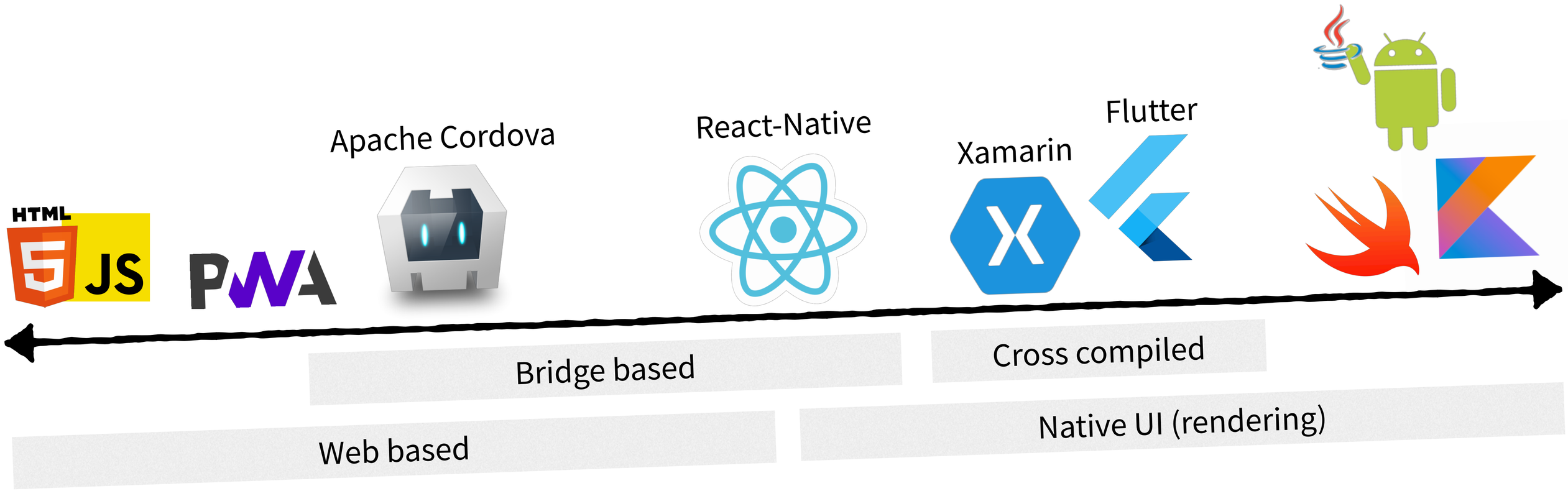
Build native mobile apps using JavaScript and React

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>
```


Apache Cordova
React-Native
Xamarin
Flutter
HTML
JS
PWA
Bridge based
Cross compiled
Web based
Native UI (rendering)






F8

B






WiX

adidas
GLITCH

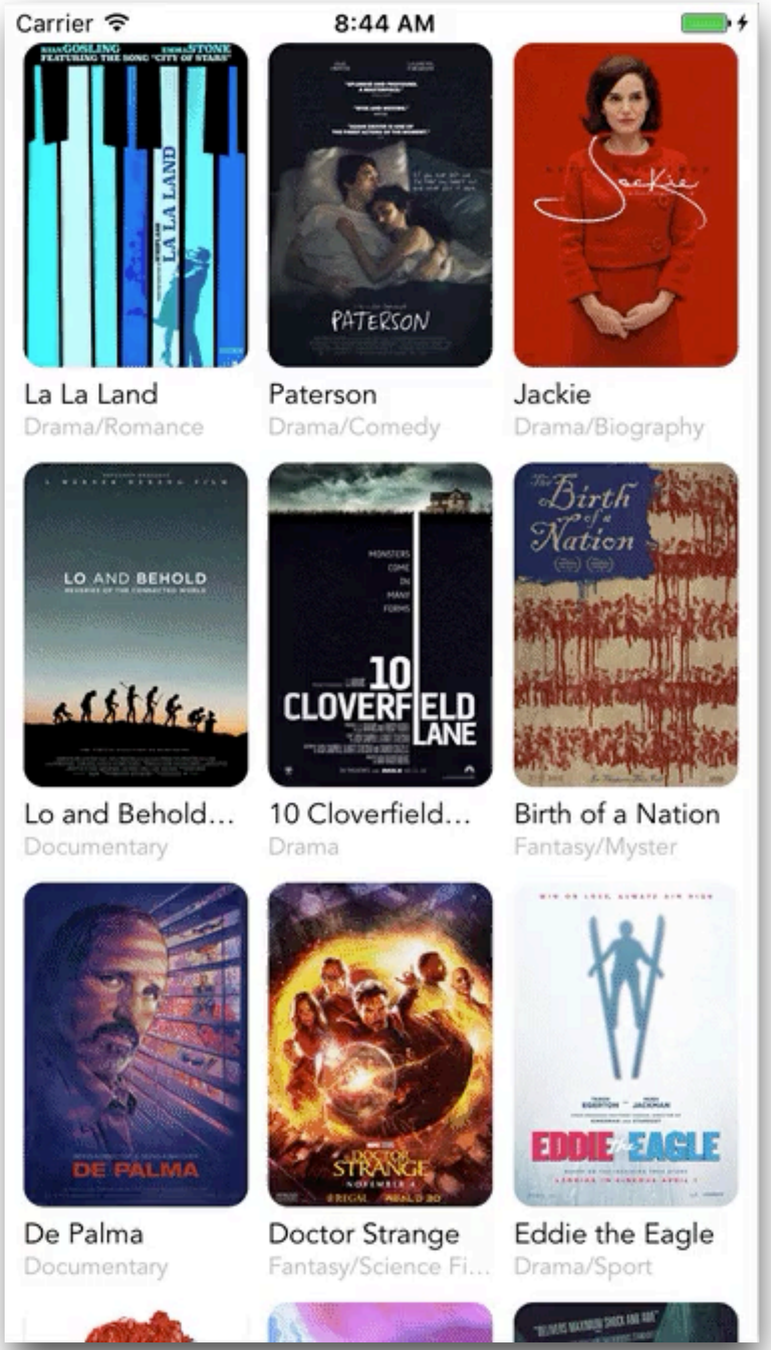


CBS SPORTS
FRANCHISE

CHOP

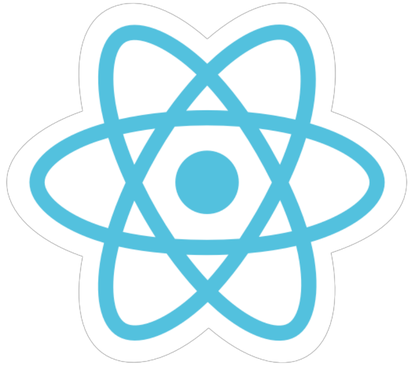


G



From https://rationalappdev.com/movie-tickets-booking-app-with-react-native/

CODE

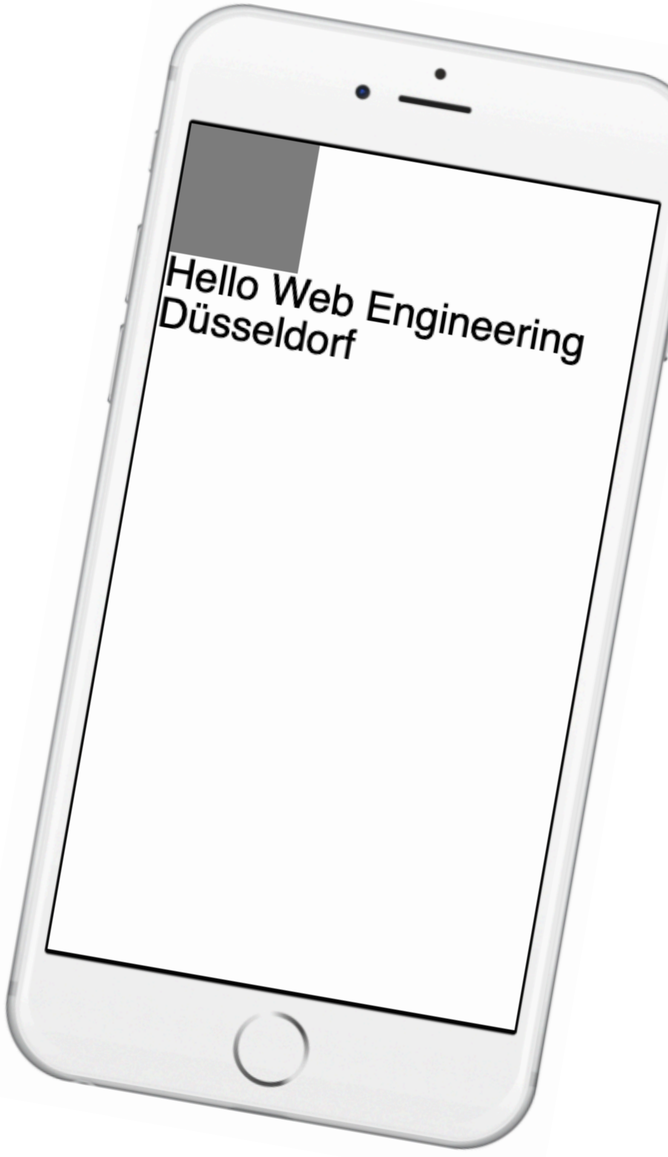


React & React-native

Flexbox Layout

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 Web Engineering Düsseldorf</Text>
      </View>
    );
  }
}
```



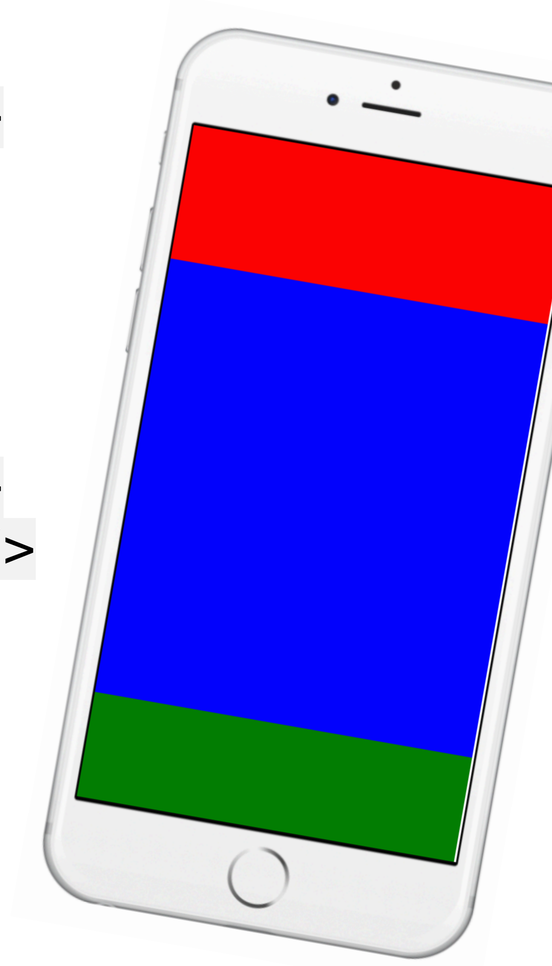
Stylesheet

```
const onlyBold = {
    fontWeight: 'bold' // A string!
};

const styles = StyleSheet.create({
    boldText: {
        ...globalStyle.text,
        fontWeight: 'bold'
    }
});
```

```
<View style={{ borderWidth: 1, borderColor: 'red' }}>
    <Text style={ onlyBold }>Hello World</Text>
    <Text style={ styles.boldText }>Hello World</Text>
</View>
```

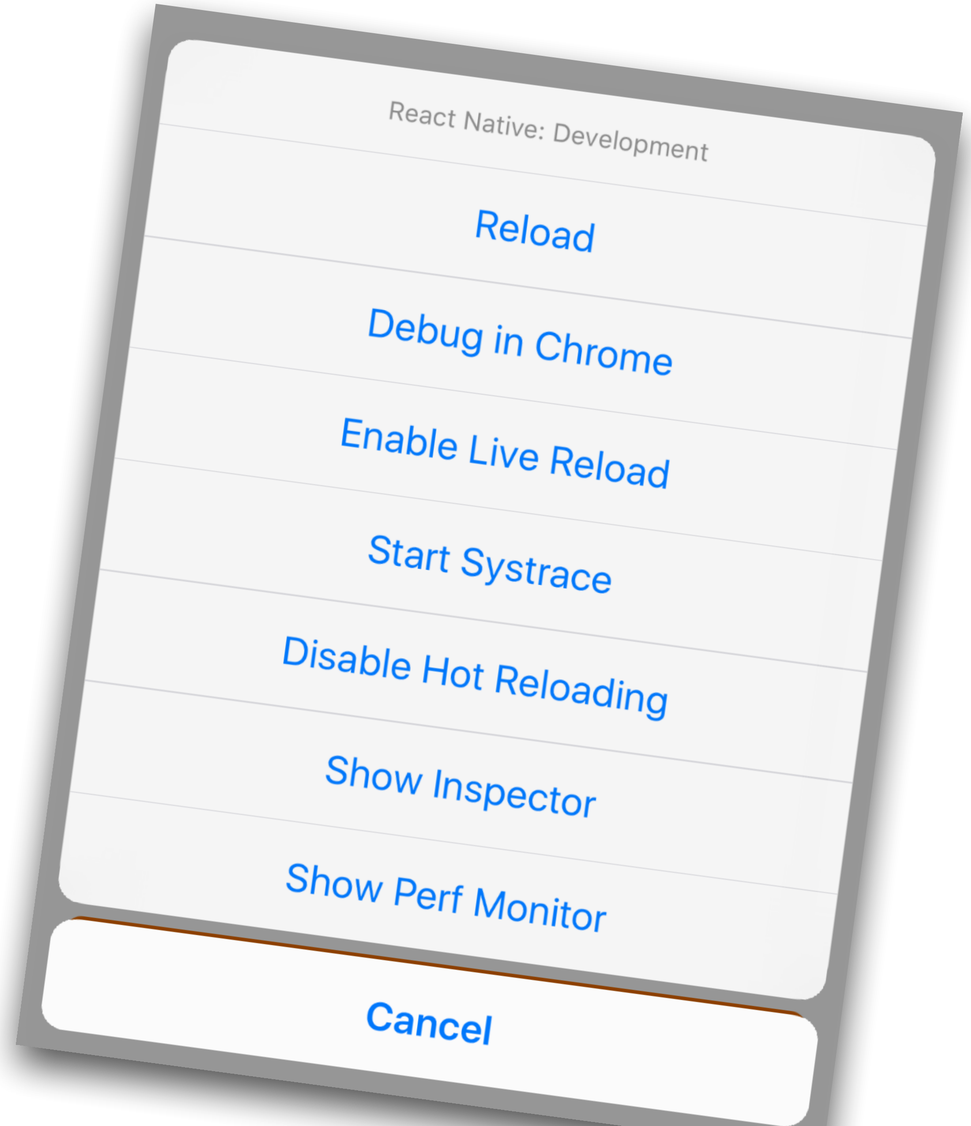
Flexbox

```
// 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
- ….



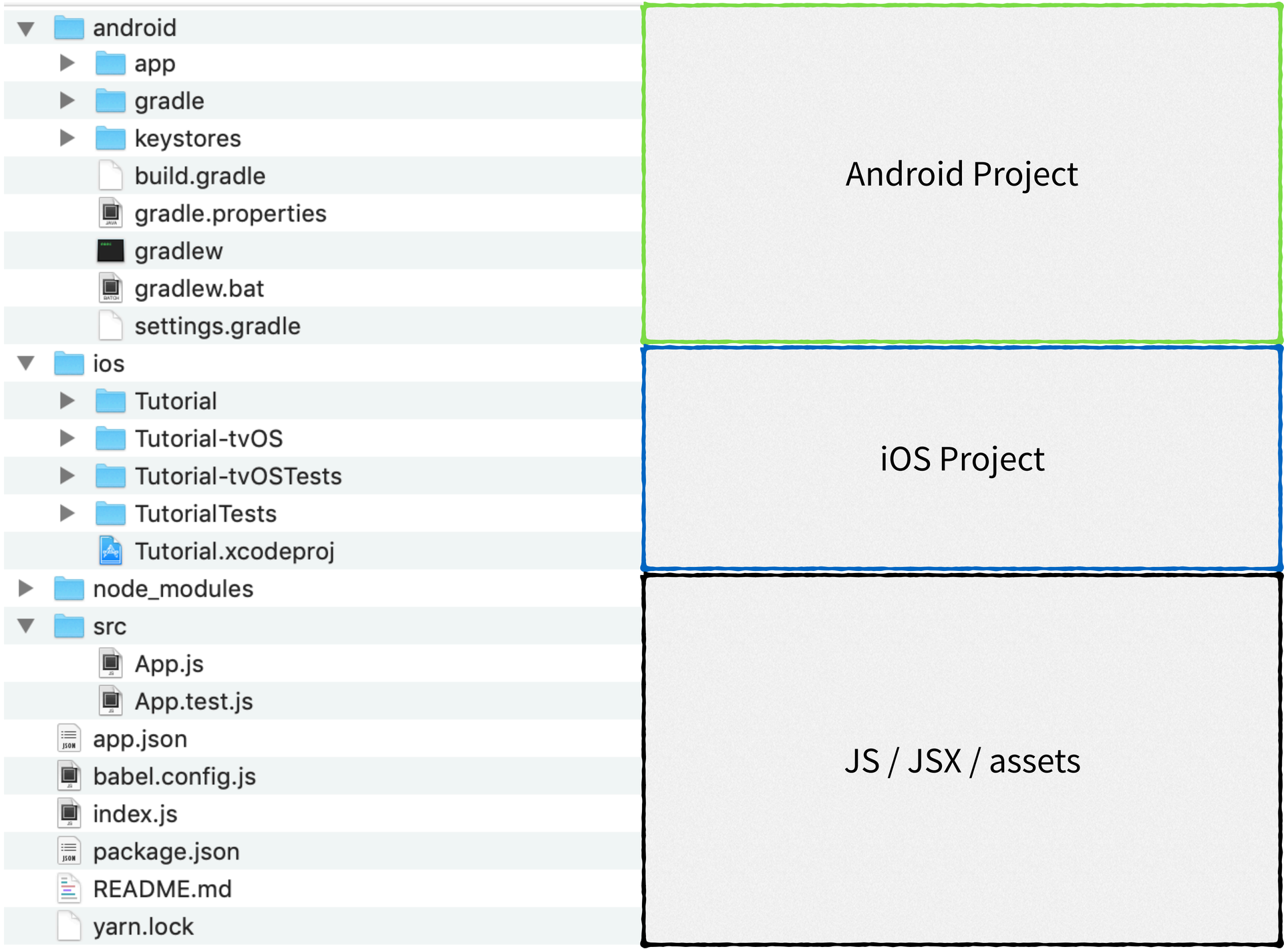
Build-in View Components

View, Text, TextInput, Image, **Switch, Slider,**
ScrollView, **WebView, ListView**, PickerIOS,
ActivityIndicator, Modal, RefreshControl,
ActivityIndicator, **DatePickerIOS**,
DrawerLayoutAndroid, SegmentedControlIOS,
TouchableHighlight, **TouchableOpacity,**
TouchableWithoutFeedback, ...

Build-in modules

ActionSheetIOS, **Alert**, **Animated**, AppRegistry, AppState, AppState, AsyncStorage, **BackAndroid**, CameraRoll, Dimensions, IntentAndroid, InteractionManager, LayoutAnimation, Linking, **NetInfo, PanResponder, PushNotificationIOS, StatusBar, StyleSheet, ToastAndroid, Vibration, ...**

HOW


android
app
gradle
keystores
build.gradle
gradle.properties
gradlew
gradlew.bat
settings.gradle
ios
Tutorial
Tutorial-tvOS
Tutorial-tvOSTests
TutorialTests
Tutorial.xcodeproj
node_modules
src
App.js
App.test.js
app.json
babel.config.js
index.js
package.json
README.md
yarn.lock
Android Project
iOS Project
JS / JSX / assets

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)

How it works?

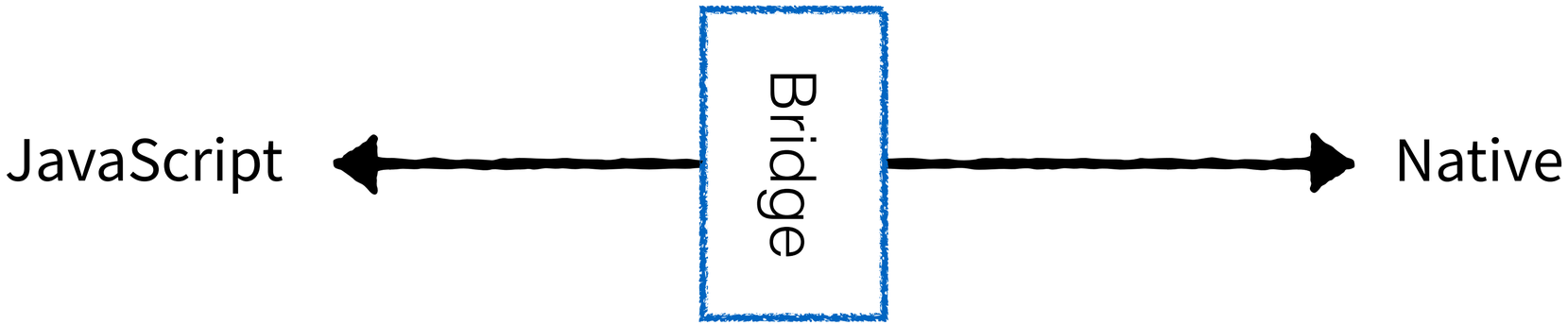
- “**Components**” wrap native UI elements
  <View … />

- “**CSS**” Reimplementation, incl. **Flexbox**
  style={{ width: 100, height: 100, backgroundColor: ‘red’, borderRadius: 10 }}

- “**Modules**” wrap native APIs

- Provide some **polyfills**
  (fetch for Network requests, geolocation for Location tracking, etc.)

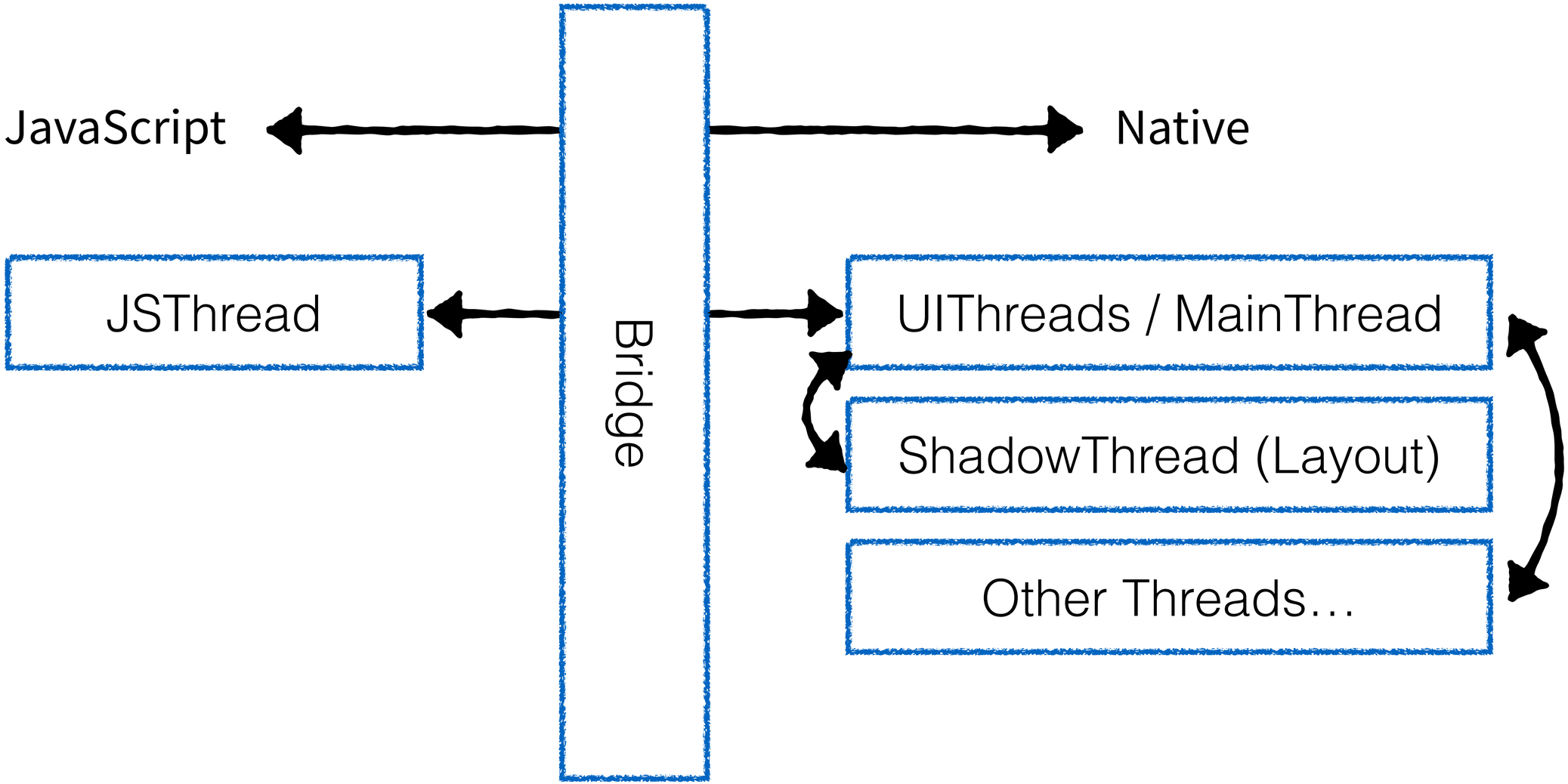
How does it work?



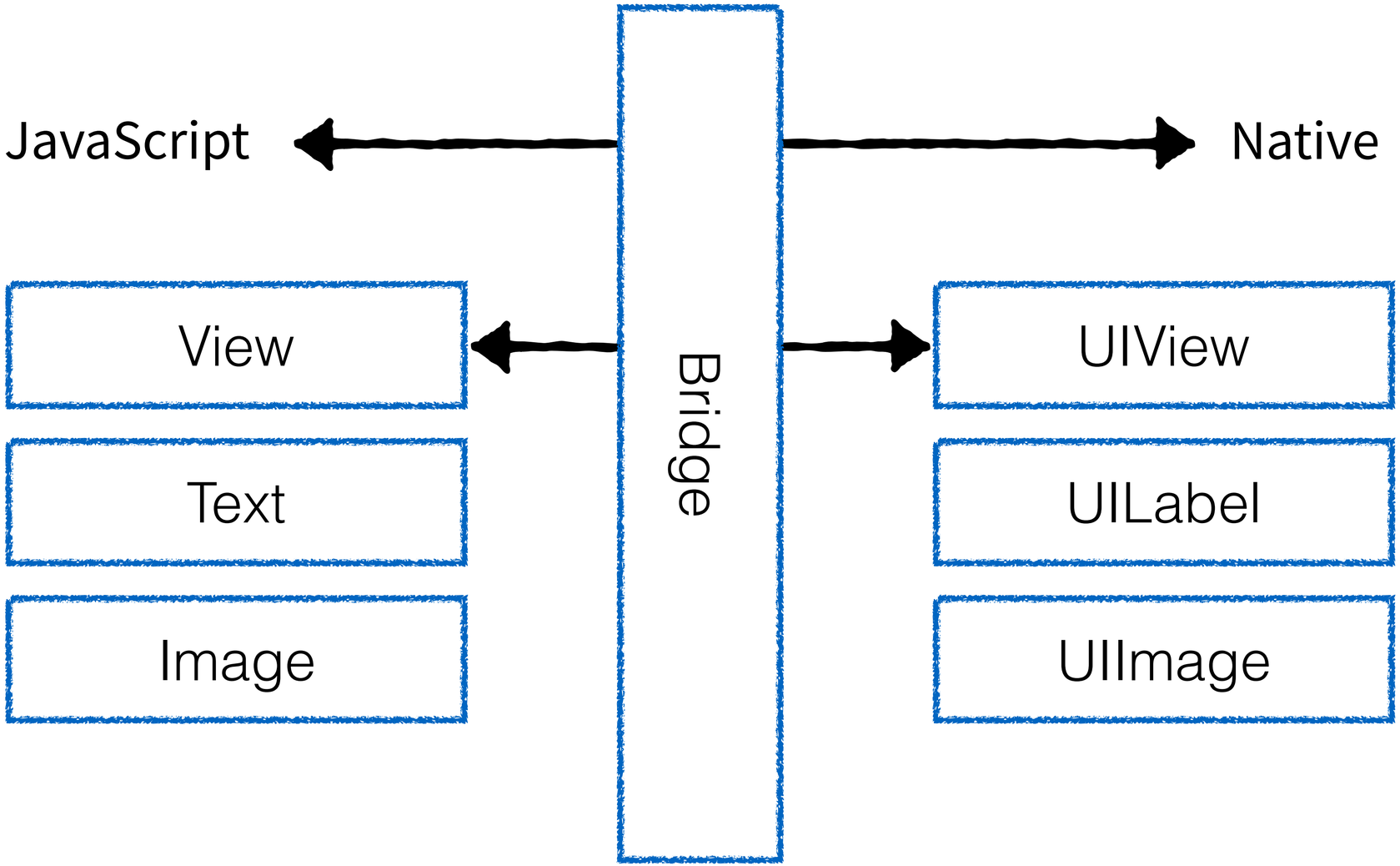
Transfer serializable data (JSON)

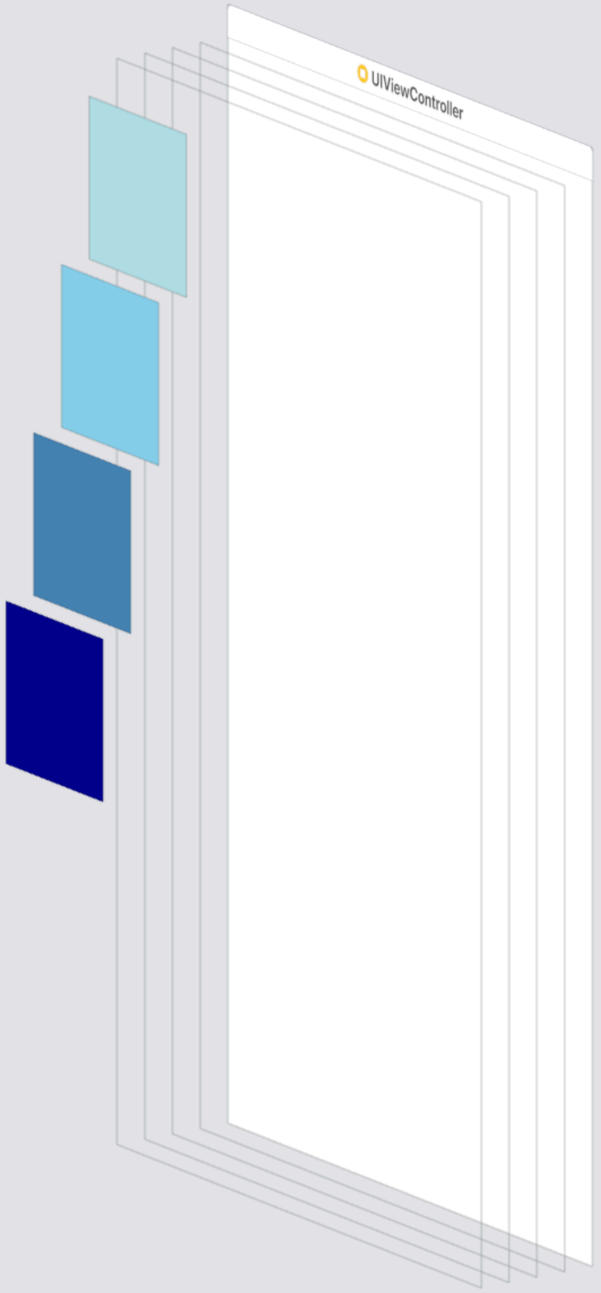
For each view change, touch event, etc.

Threads


JavaScript
Native
JSThread
Bridge
UIThreads / MainThread
ShadowThread (Layout)
Other Threads…

How does it work?




UIViewController

PAST

What changed over the last years

- Support for **Android** 😉
- Also rendering backends available for Windows, Ubuntu and Web
- Shift from “per platform” components to “real” cross platform components
- Mostly stable and high quality components by the community
- Externalization of components:
  - WebView was the component which are extracted
    from the core to an own community project (react-native-webview)

https://github.com/react-native-community/discussions-and-proposals/pull/3

RefreshControl

Edit on GitHub

This component is used inside a ScrollView to add pull to refresh functionality. When the ScrollView is at `scrollY: 0`, swiping down triggers an `onRefresh` event.

Props

View props...

android **colors** [[object Object]]

The colors (at least one) that will be used to draw the refresh indicator.

android **progressBackgroundColor** color

The background color of the refresh indicator.

ios **tintColor** color

The color of the refresh indicator.

ios **title** string

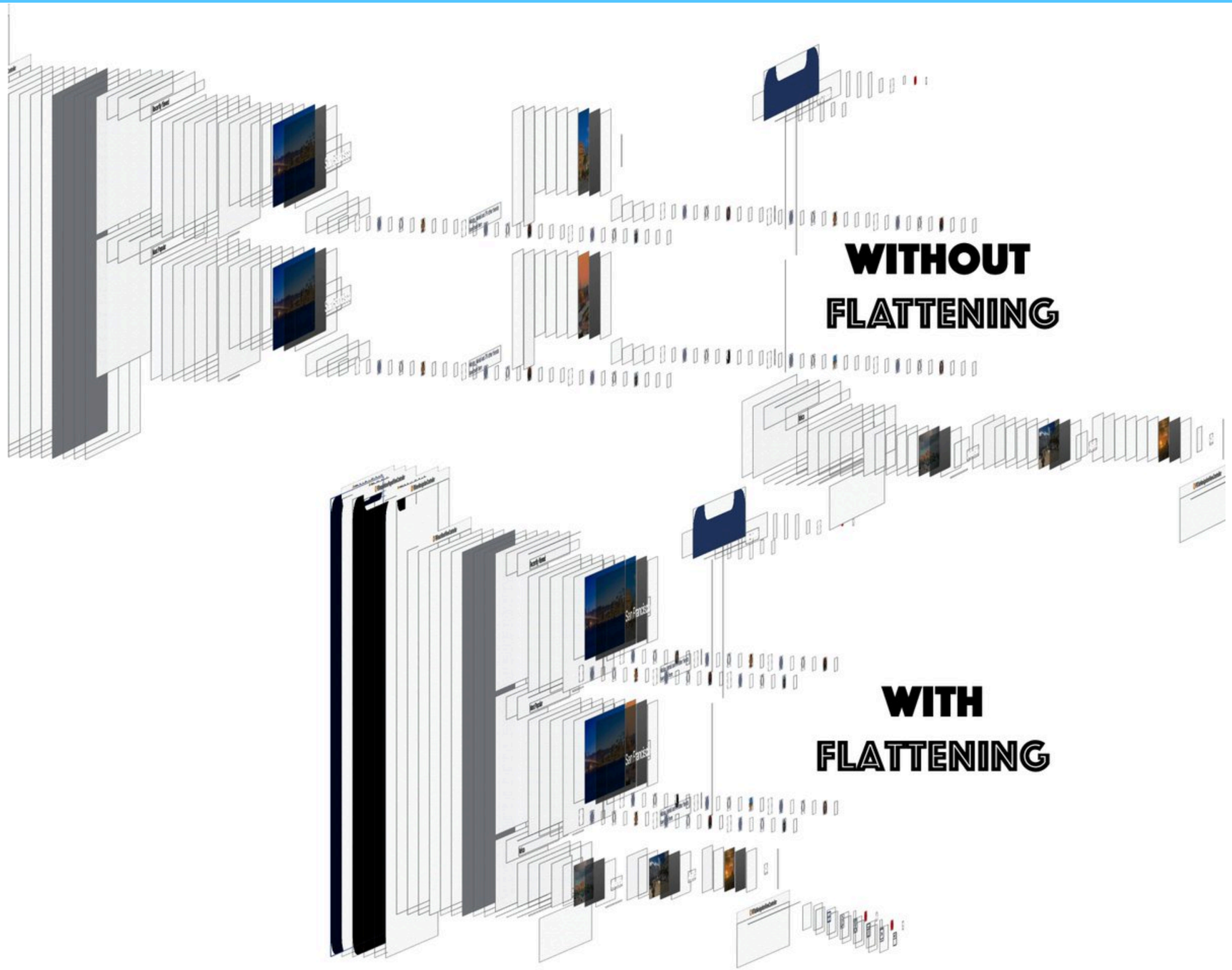
The title displayed under the refresh indicator.

NEXT

React Native Fabric

- Unify more APIs, for example DatePickeriOS and DatePickerAndroid
- Allows **synchronous** bridge communication (great for example for WebView shouldOpenURL(…)
- Auto **Flattening** native View Hierarchy
- Rewrite (part of) the core in C++
- Make apps lighter and faster (esp. time to interaction)
  lazy native class initialization
- Easier setup of multiple React Native Surfaces
- Support for React Fiber Priorities (Async, Batched and Deferred rendering)

https://github.com/react-native-community/discussions-and-proposals/issues/4
http://facebook.github.io/react-native/blog/2018/06/14/state-of-react-native-2018



From Valentin Shergin (React Native Core Developer): https://twitter.com/shergin/status/1058393187079704576

GETTING STARTED

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 (or two) projects than in two (or three)
- 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-GS0M8Q95RIc2lOM6nc77q1IY
- React Conf 2017
  https://www.youtube.com/playlist?list=PLb0IAmt7-GS3fZ46IGFirdqKTIxlws7e0
- “awesome-react-native” link list
  https://github.com/jondot/awesome-react-native

LIVE DEMO

Q&A