

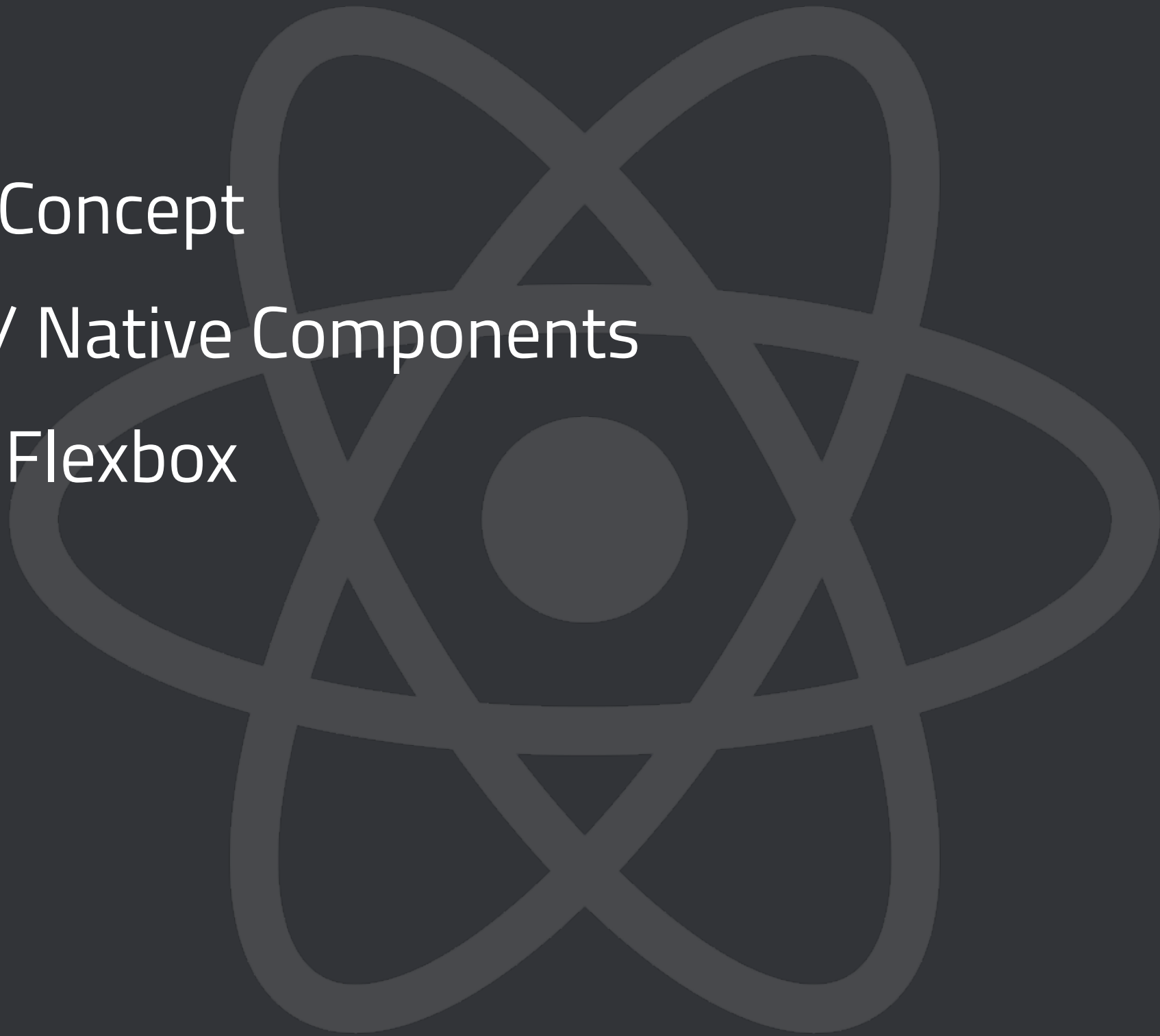
Introduction

react-native

PottJS Meetup, May 4th 2016, Christoph Jerolimov

Agenda

- Motivation & Concept
- Components / Native Components
- Stylesheets / Flexbox
- Demo



Write once, run anywhere
Initially Java, but html5 too



Write once, run anywhere
Learn once, use anywhere



React Native

A FRAMEWORK FOR BUILDING NATIVE APPS USING REACT

React Native enables you to build world-class application experiences on native platforms using a consistent developer experience based on JavaScript and [React](#). The focus of React Native is on developer efficiency across all the platforms you care about — learn once, write anywhere. Facebook uses React Native in multiple production apps and will continue investing in React Native.

Get started with React Native

Native Components

With React Native, you can use the standard platform components such as `UITabBar` on iOS and `Drawer` on Android. This gives your app a consistent look and feel with the rest of the platform ecosystem, and keeps the quality bar high. These components are easily incorporated into your app using their React component counterparts, such as `TabBarIOS` and `DrawerLayoutAndroid`.

```
// iOS

var React = require('react-native');
var { TabBarIOS, NavigatorIOS } = React;

var App = React.createClass({
  render: function() {
    return (
      <TabBarIOS>
        <TabBarIOS.Item title="React Native" selected={true}>
```

Why?

- Reuse react.js (web) to render the view hierarchy
- Renders native views (no WebView!)
- Polyfills for networking (fetch), Geolocation, ...
- (Easy) Integration options in *both* directions

Why react?

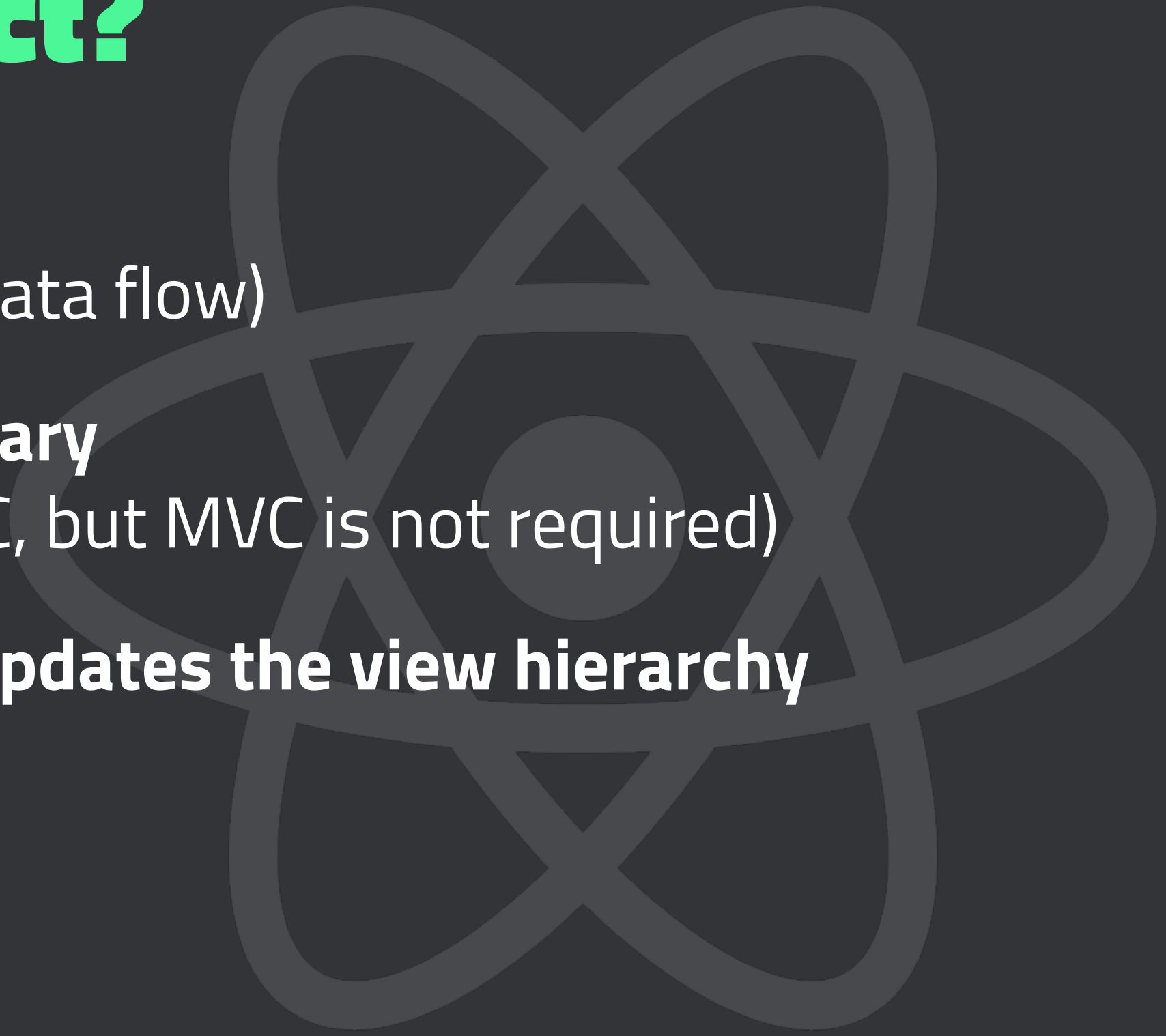
Declarative UI

(Unidirectional data flow)

A view-only library

(the view in MVC, but MVC is not required)

Automatically updates the view hierarchy



Developer Experience?

- "HTML- & CSS-like" => JSX + Flexbox
- Hot reloading (⌘R) & Live Reload
- Debugger, UI Inspector, Profiling



Hello world

React Native: Development

Reload

Debug in Chrome

Disable Live Reload

Start Systrace

Show Inspector

Show Perf Monitor

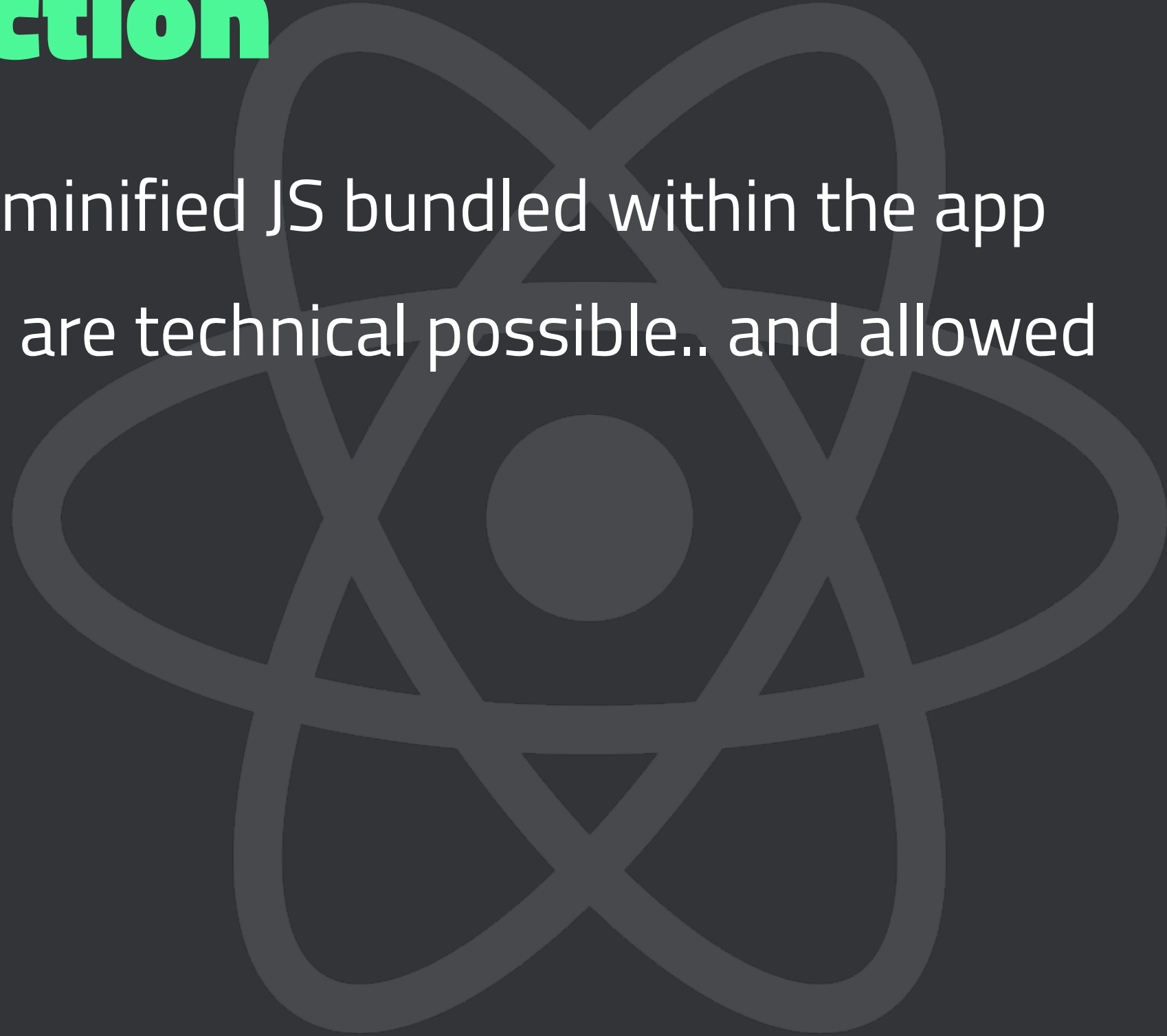
Cancel

In development

- You write "**modern**" javascript in your favorited editor
- Babel transform the sources (ES6 and more...)
- App communicates with a local http server

In production

- Precompiled, minified JS bundled within the app
- Code updates are technical possible.. and allowed



Technical

- Based on a **minimal JS VM: JavaScriptCore**
- JS controls the native UI
- JS renders the "virtual DOM" as JSON
- JS <-> Native bridge (multithreaded)
- Native part renders UI based on this JSON

Supported platforms

- Android 4.1+, $\geq 93\%$ ¹
- iOS 7+, $\geq 96\%$ ^{2 3}

¹ <https://developer.android.com/about/dashboards/index.html>

² <https://david-smith.org/iosversionstats/>

³ <https://developer.apple.com/support/app-store/>

Getting Started⁴

- Requires Node.js 4+, nvm is recommended
- OSX is the common dev platform (at FB)
- Linux and Windows should work⁵
- Android SDK⁶ for Android / Xcode 7+ for iOS

⁴ <http://facebook.github.io/react-native/docs/getting-started.html>

⁵ <http://facebook.github.io/react-native/docs/linux-windows-support.html>

⁶ <http://facebook.github.io/react-native/docs/android-setup.html>

Hello world



JS + JSX

```
class HelloWorld extends Component {  
  render() {  
    return <Text>Hello World</Text>;  
  }  
}
```

```
AppRegistry.registerComponent('MyApp', () => HelloWorld);
```


Stylesheets

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

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

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

View components

View, Text, TextInput, Image, Switch, ScrollView, PickerIOS, ProgressBarAndroid, ProgressViewIOS, WebView, ListView, Navigator, NavigatorIOS, Modal, MapView, RefreshControl, TabBarIOS, ActivityIndicatorIOS, DatePickerIOS, DrawerLayoutAndroid, PullToRefreshViewAndroid, SegmentedControlIOS, SliderIOS, TouchableHighlight, TouchableOpacity, TouchableWithoutFeedback, ...

Other APIs / modules

ActionSheetIOS, **Alert**, AlertIOS, **Animated**, AppRegistry, AppState, AppStateIOS, AsyncStorage, **BackAndroid**, CameraRoll, Dimensions, IntentAndroid, InteractionManager, LayoutAnimation, LinkingIOS, **NetInfo**, PanResponder, **PushNotificationIOS**, **StatusBarIOS**, **StyleSheet**, ToastAndroid, VibrationIOS, ...

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.

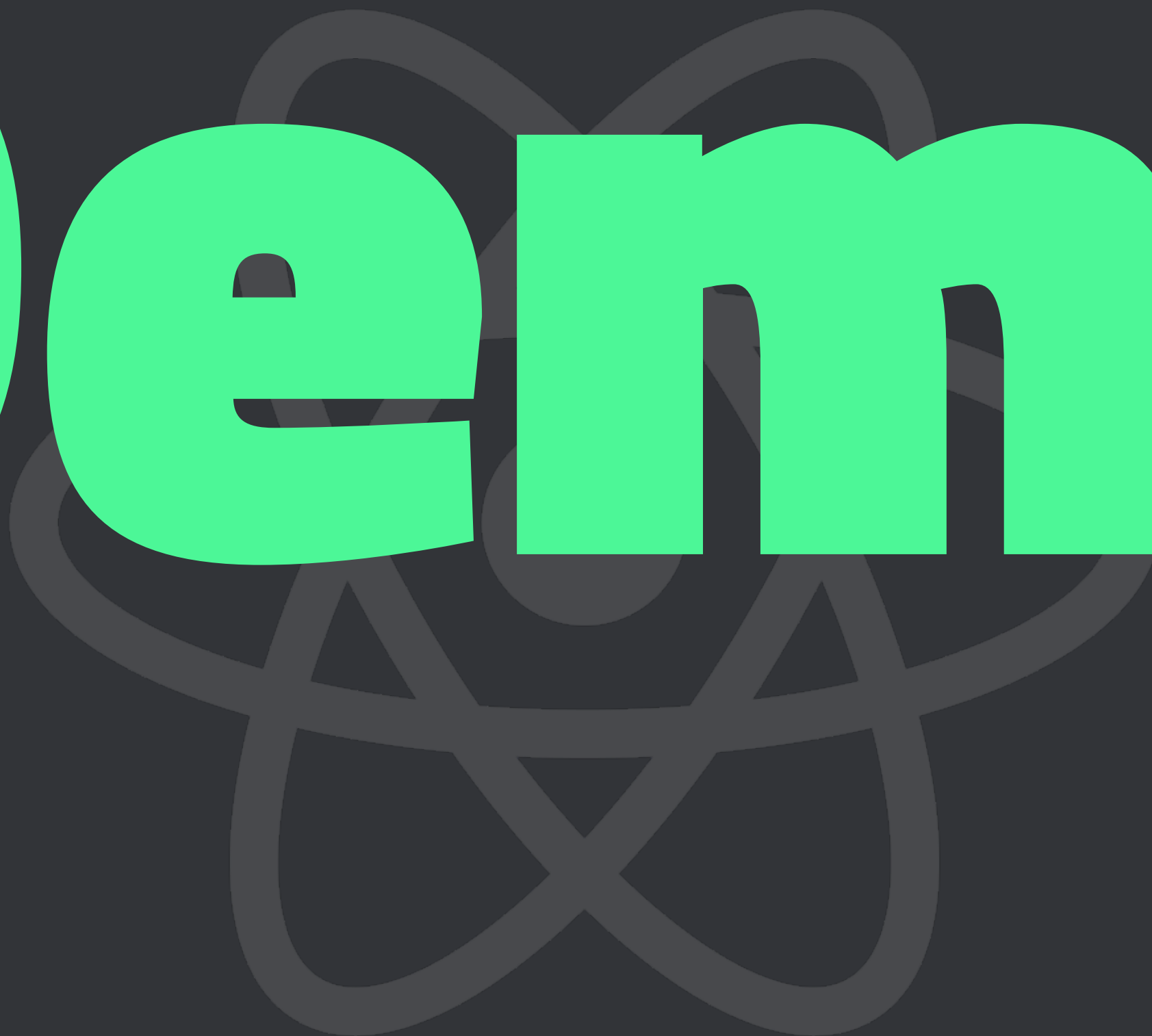
`ios` **tintColor** `color`

The color of the refresh indicator.

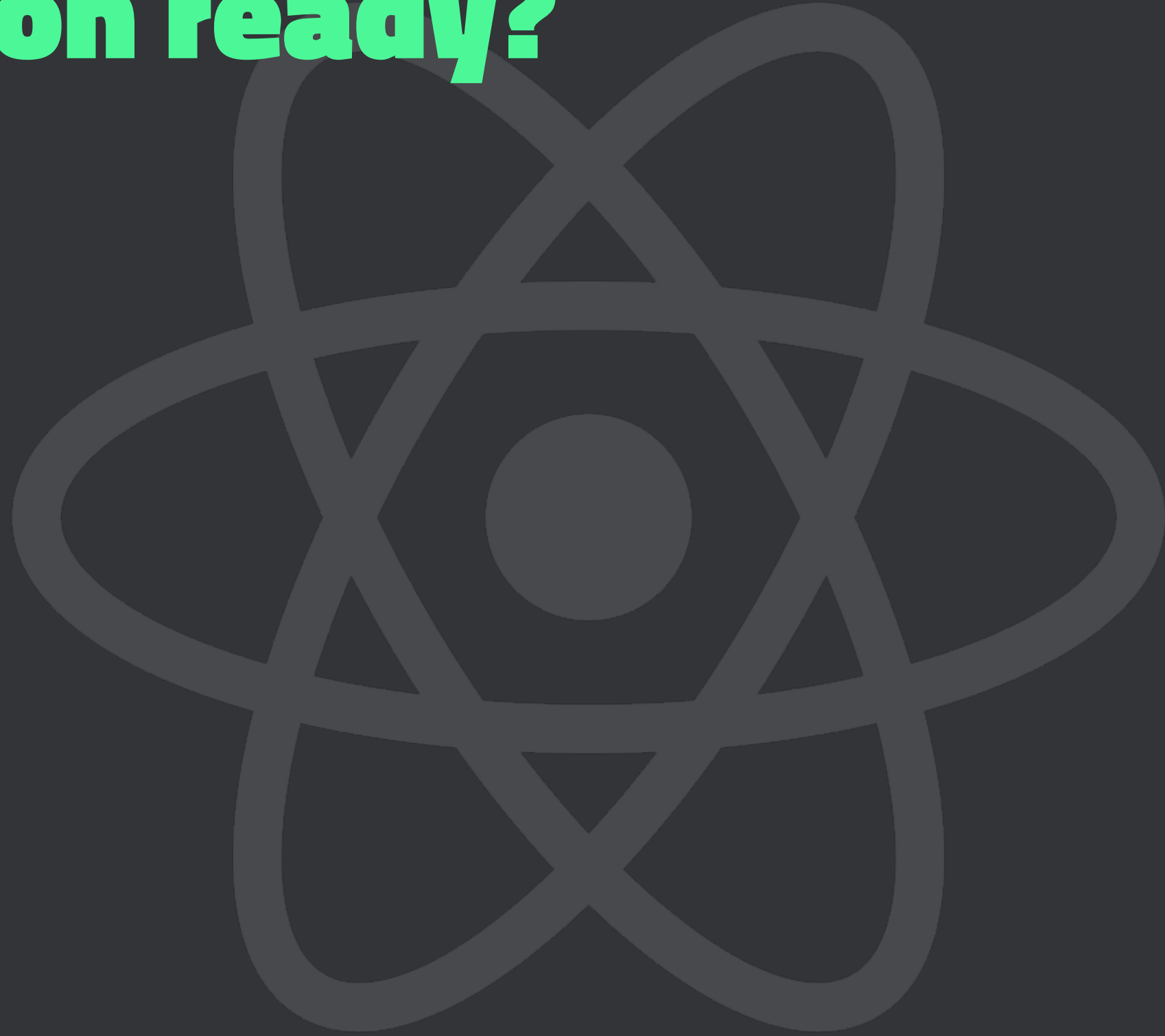
`ios` **title** `string`

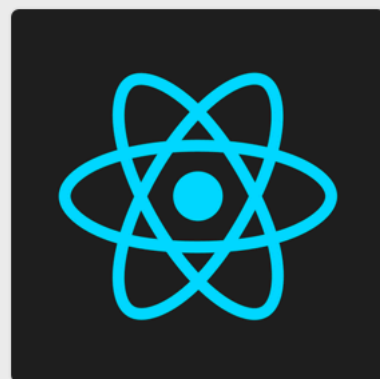
The title displayed under the refresh indicator.

Daimo



Production ready?





React Native

A framework for building native apps with React

🌐 facebook.github.io/react-native

REP



Recent Activity

Top

Fixed



REACT NATIVE

191



Get Android to feature parity with iOS



22



REACT NATIVE

172



React-Native to depend on React



4



REACT NATIVE

115



Offload some animations from JS thread for better perf



8



REACT NATIVE

82



Fix Various ListView Issues



8

Status & Roadmap

- 0.x – But production ready if your brave.
- Some components are not yet available on Android (MapView for example, but community projects are available for all common problems!)
- Android M permissions
- Performance and API improvements

Navigation

pain: Navigator / NavigatorIOS / DrawerLayoutAndroid

better: ExNavigator by James @Ide

upcoming: NavigationExperimental

tip: Make your navigation stack serializable

Performance

- Native UI, e.g. ScrollView
- Smooth animations
- Sometimes laggy, e.g. ListView, missing estimated cell height?
- Never as fast as *optimized* native code

Platform switch

Auto-select component based on a file suffix:

`CustomShoppingCardItem.android.js`

`CustomShoppingCardItem.ios.js`

Or a good old platform switch:

```
import { Platform } from 'react-native';
```

```
if (Platform.OS === 'android') {
```

```
  // ...
```

```
} else {
```

```
  // ...
```

```
}
```

Questions?



Questions?

Thank you!