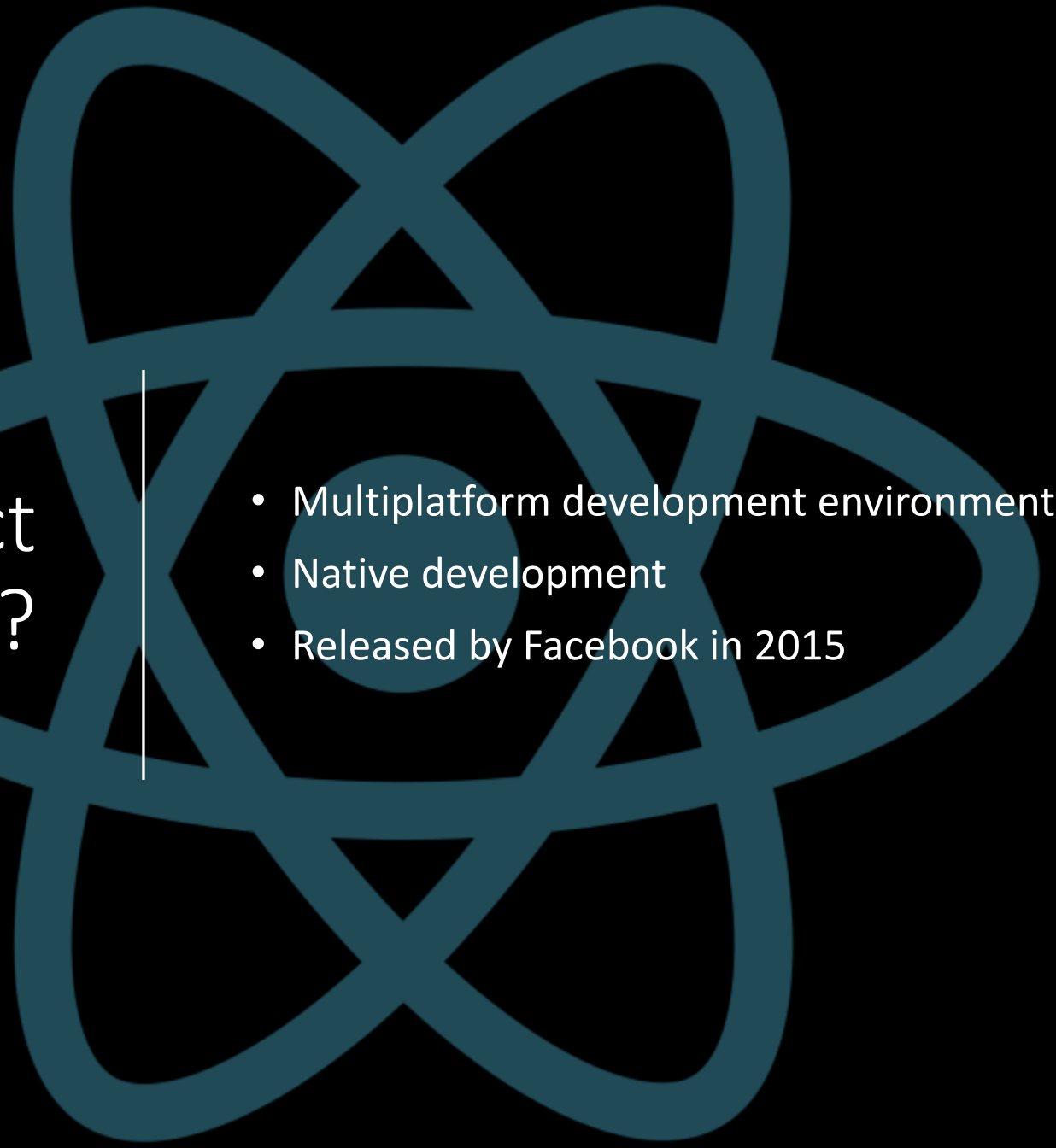
React-Native

- García Álvarez, Pelayo
- González Meléndez, Alejandro
- Gutiérrez Fernández, Íñigo
- Megido García, Lucía

What is React Native?

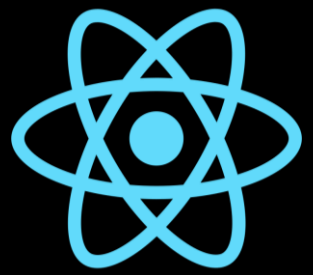


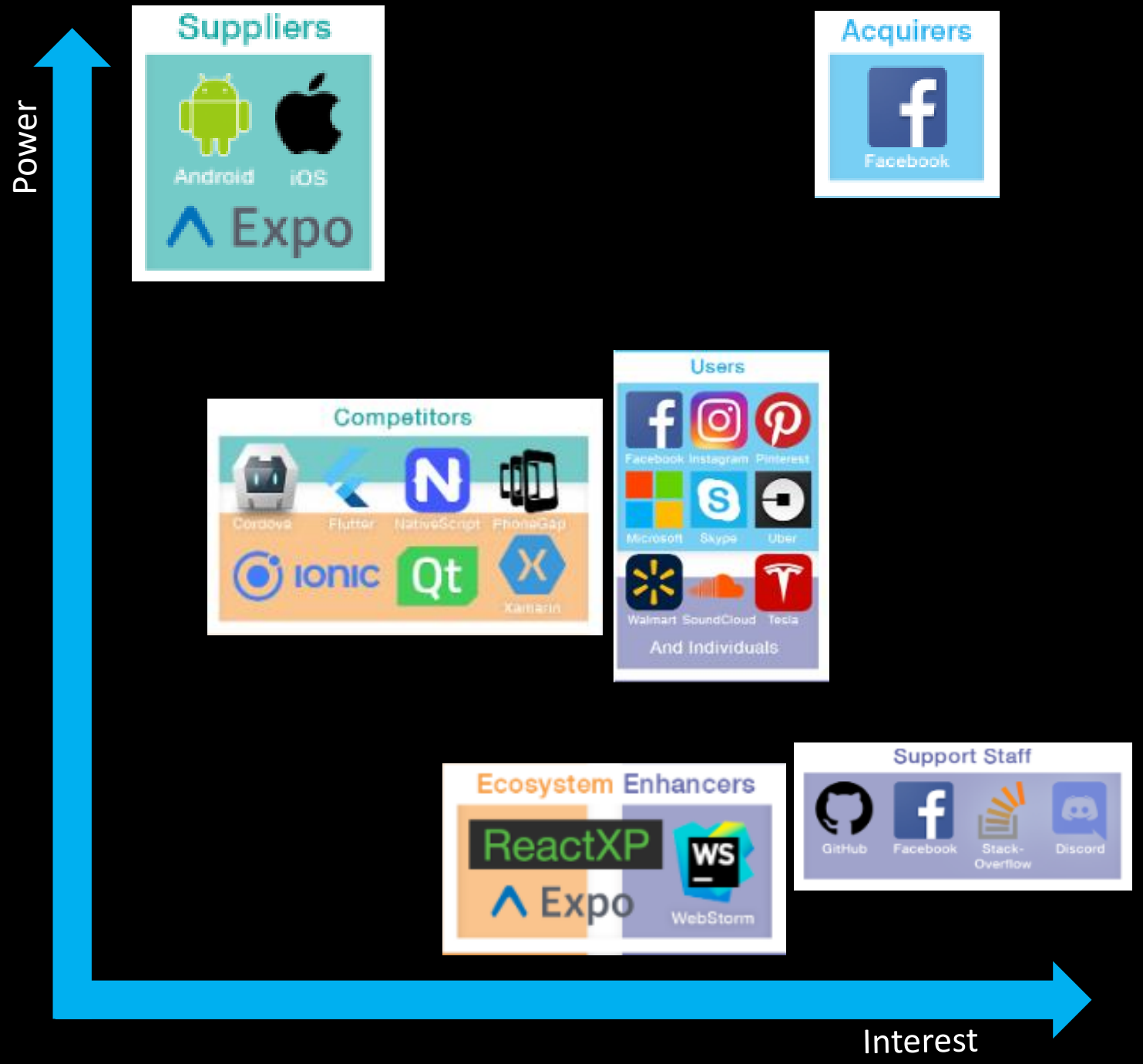
- Multiplatform development environment
- Native development
- Released by Facebook in 2015

How does it work?

- React framework + Native development
- React primitives render to native platform UI
- Core components mapping to native UI building blocks:
  - View
  - ScrollView
  - ImageView
  - Text
  - …

Stakeholders





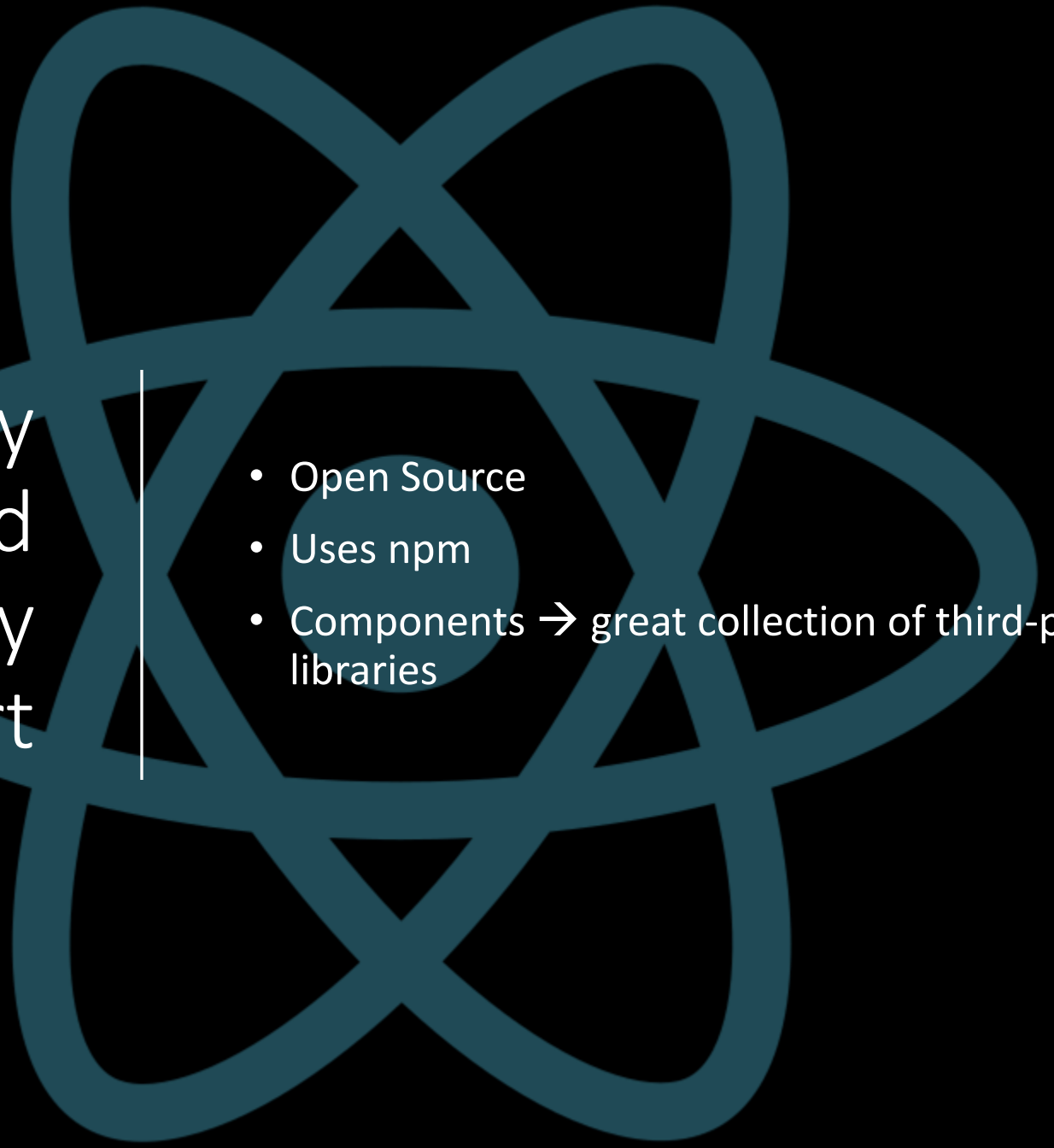
Quality Attributes

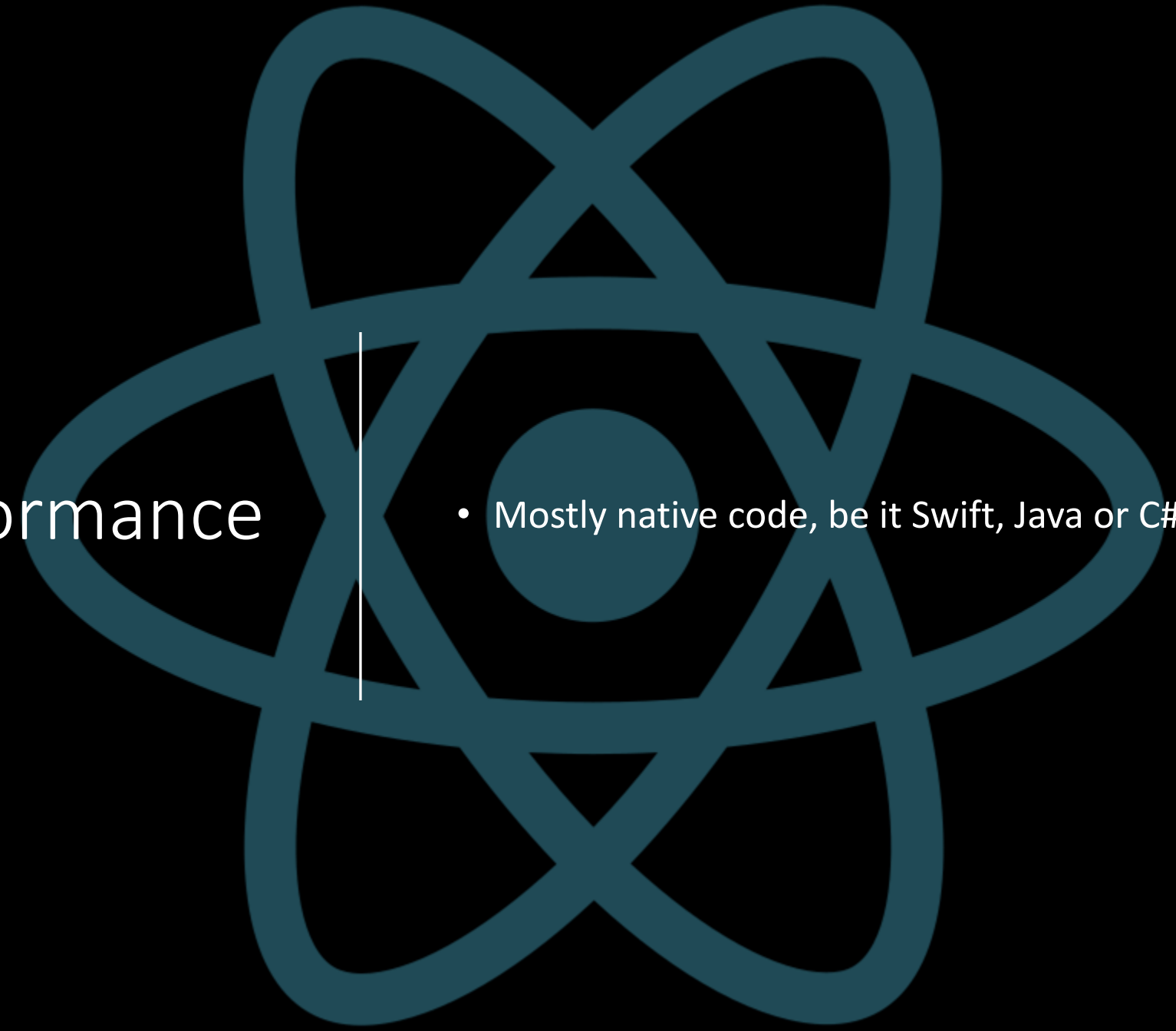
Reusability and Learnability

- One code base → multiple platforms (IOS, Android, UWP, …)
- Written in JSX, syntax extension to JavaScript

Modularity and Community Support

- Open Source
- Uses npm
- Components → great collection of third-party libraries

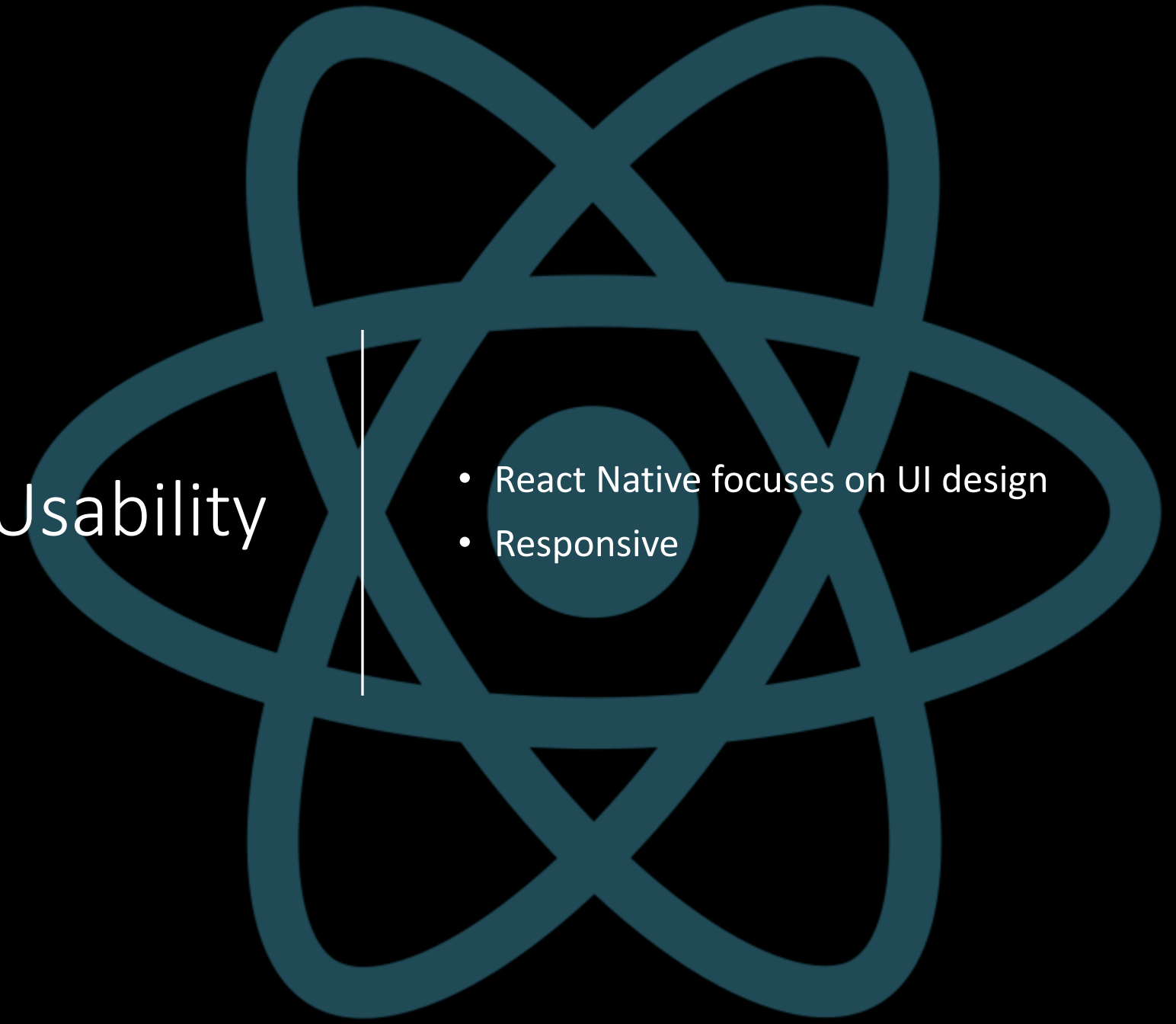




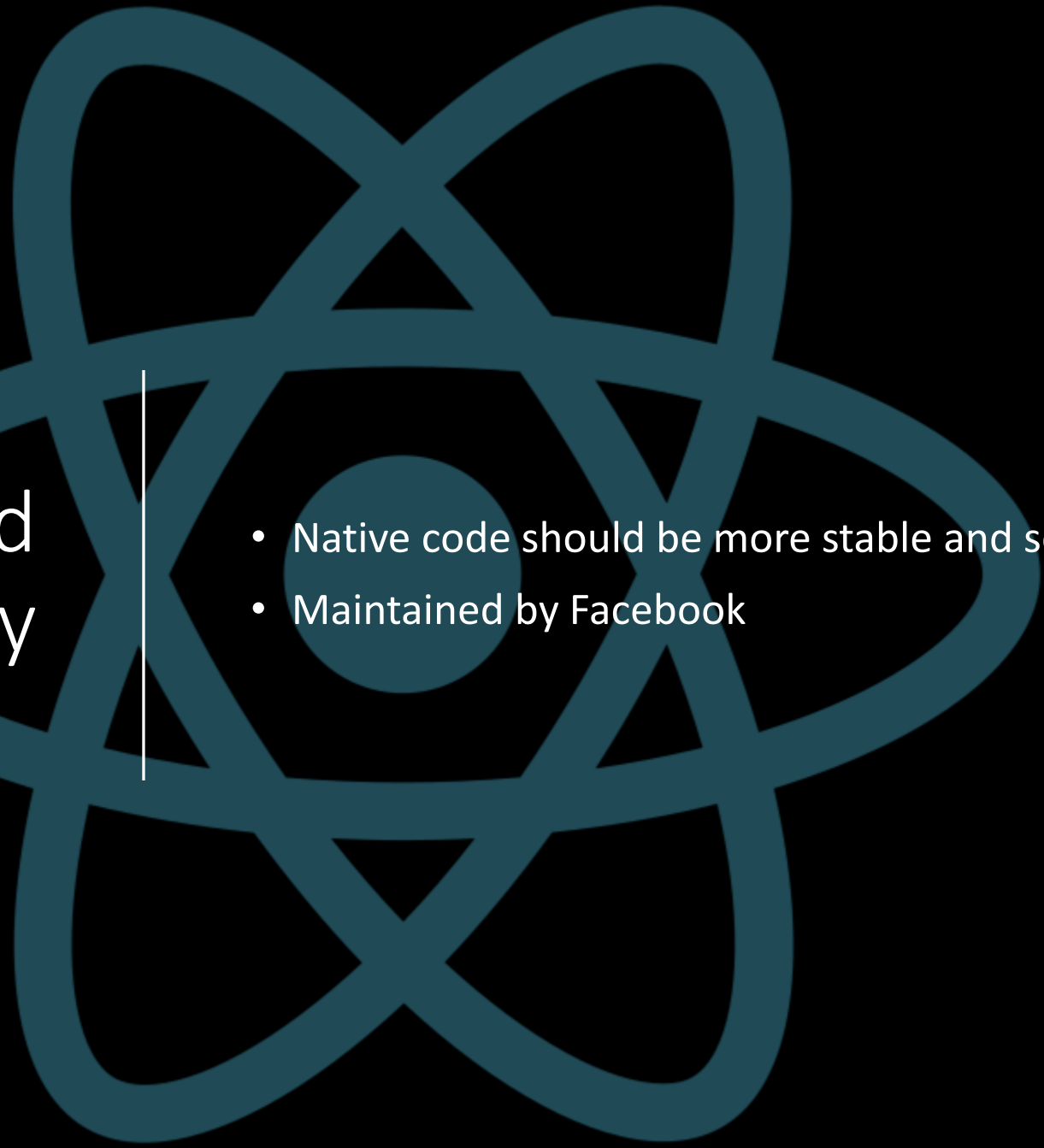
Performance

- Mostly native code, be it Swift, Java or C#

Usability

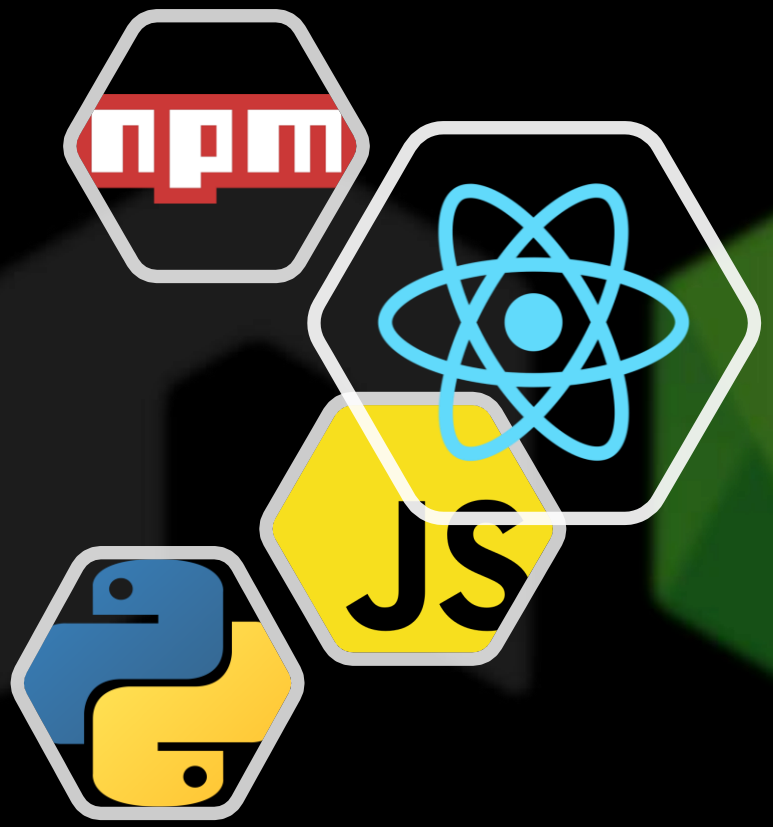


- React Native focuses on UI design
- Responsive



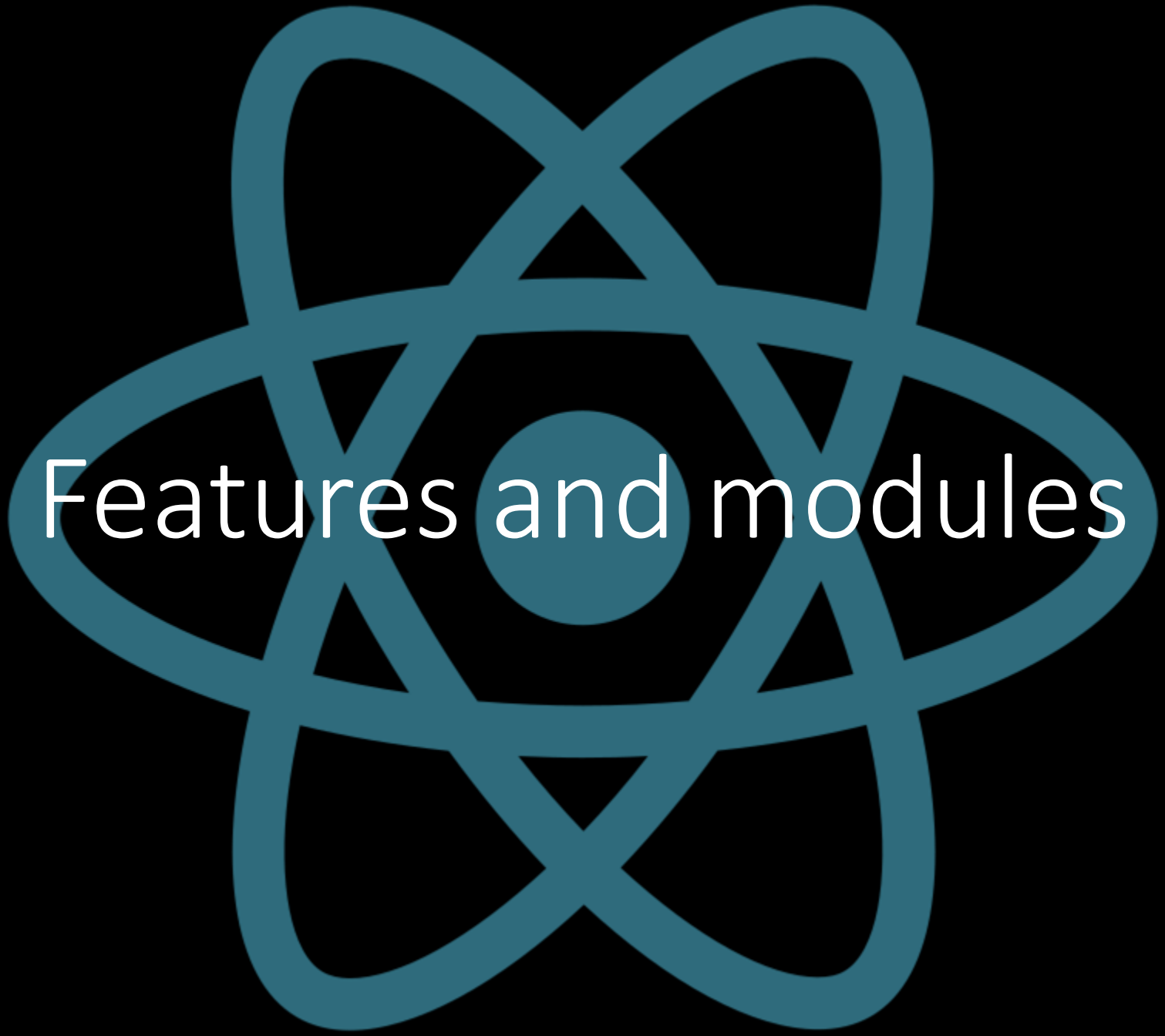
Security and Reliability

- Native code should be more stable and secure
- Maintained by Facebook



Constraints

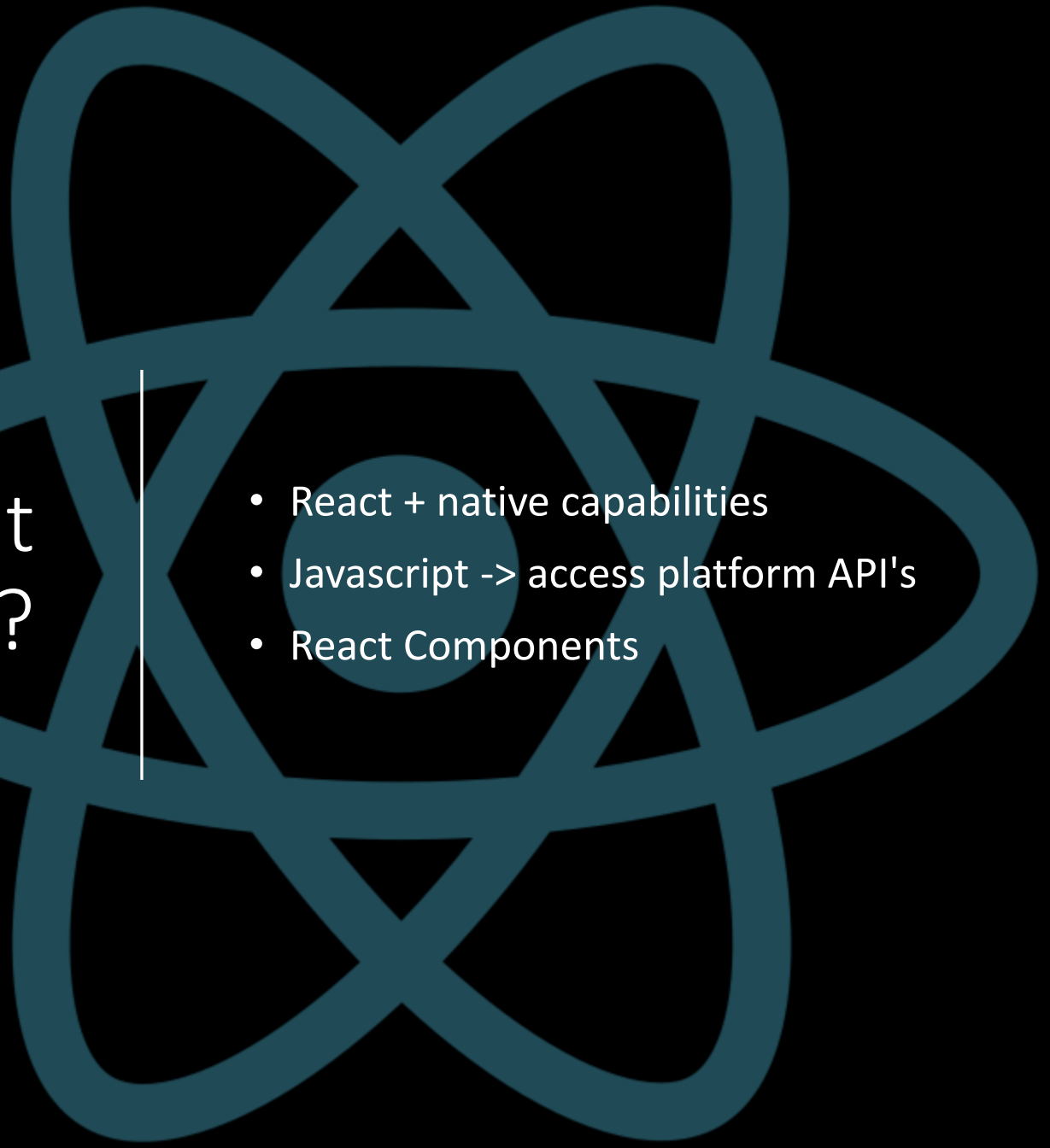
- JavaScript
- NodeJs
- Package Manager (npm, yarn…)
- Python
- Reusability



Features and modules

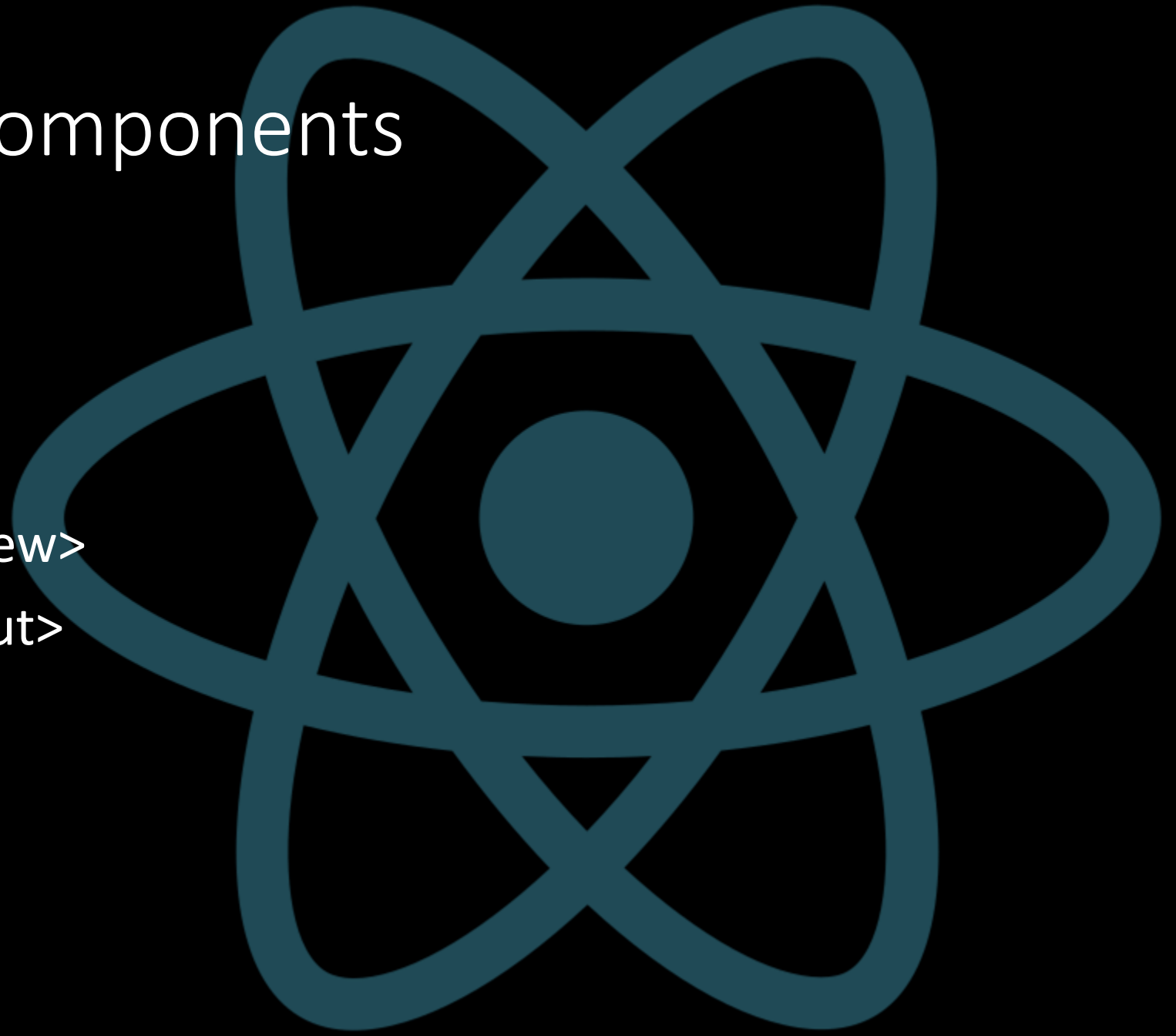
How does it work?

- React + native capabilities
- Javascript -> access platform API's
- React Components



Core Components

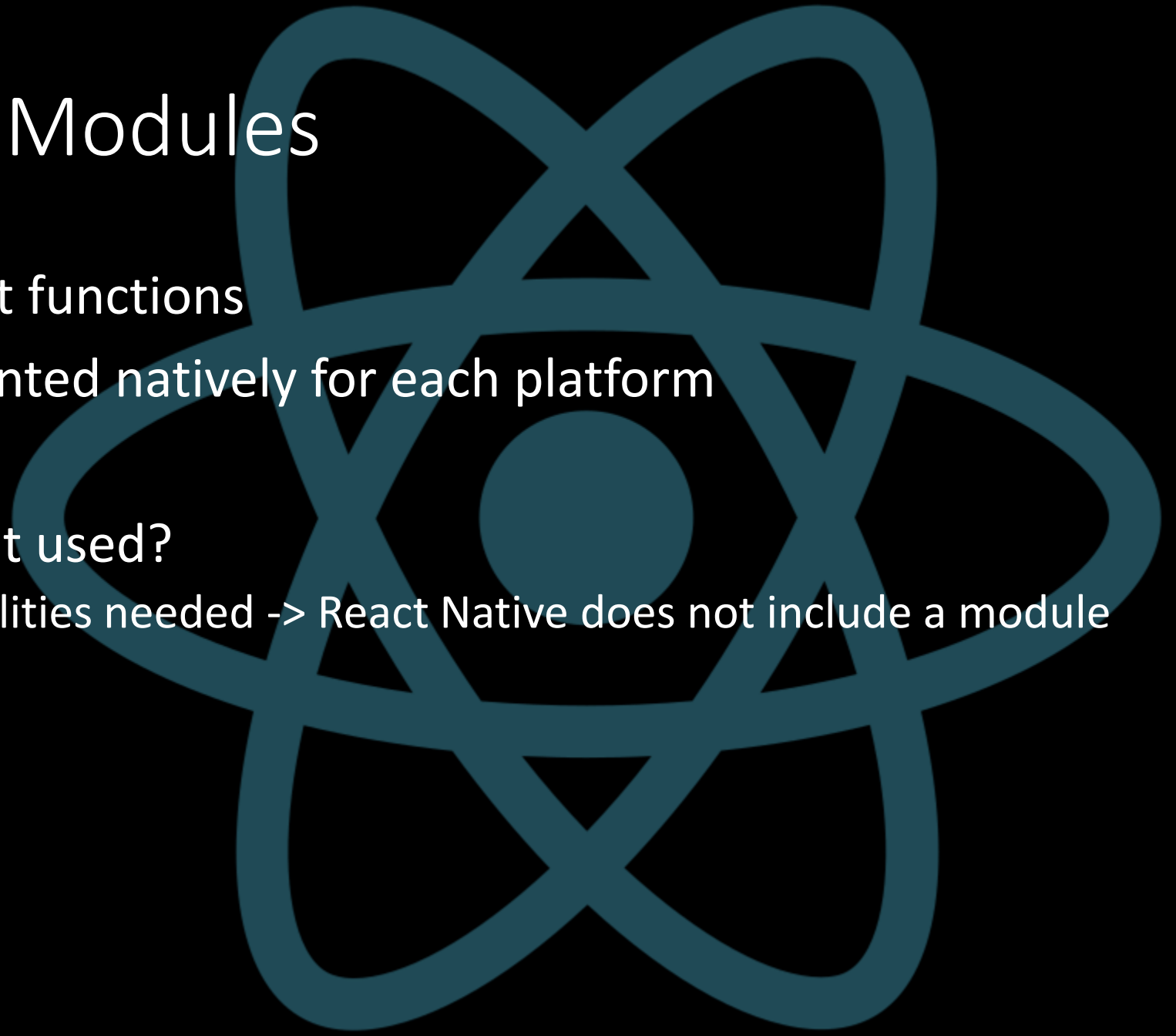
- <View>
- <Text>
- <Image>
- <ScrollView>
- <TextInput>



Native Modules

- Javascript functions
- Implemented natively for each platform

- When is it used?
  - Capabilities needed -> React Native does not include a module



Creating a Native Module

- Create the application

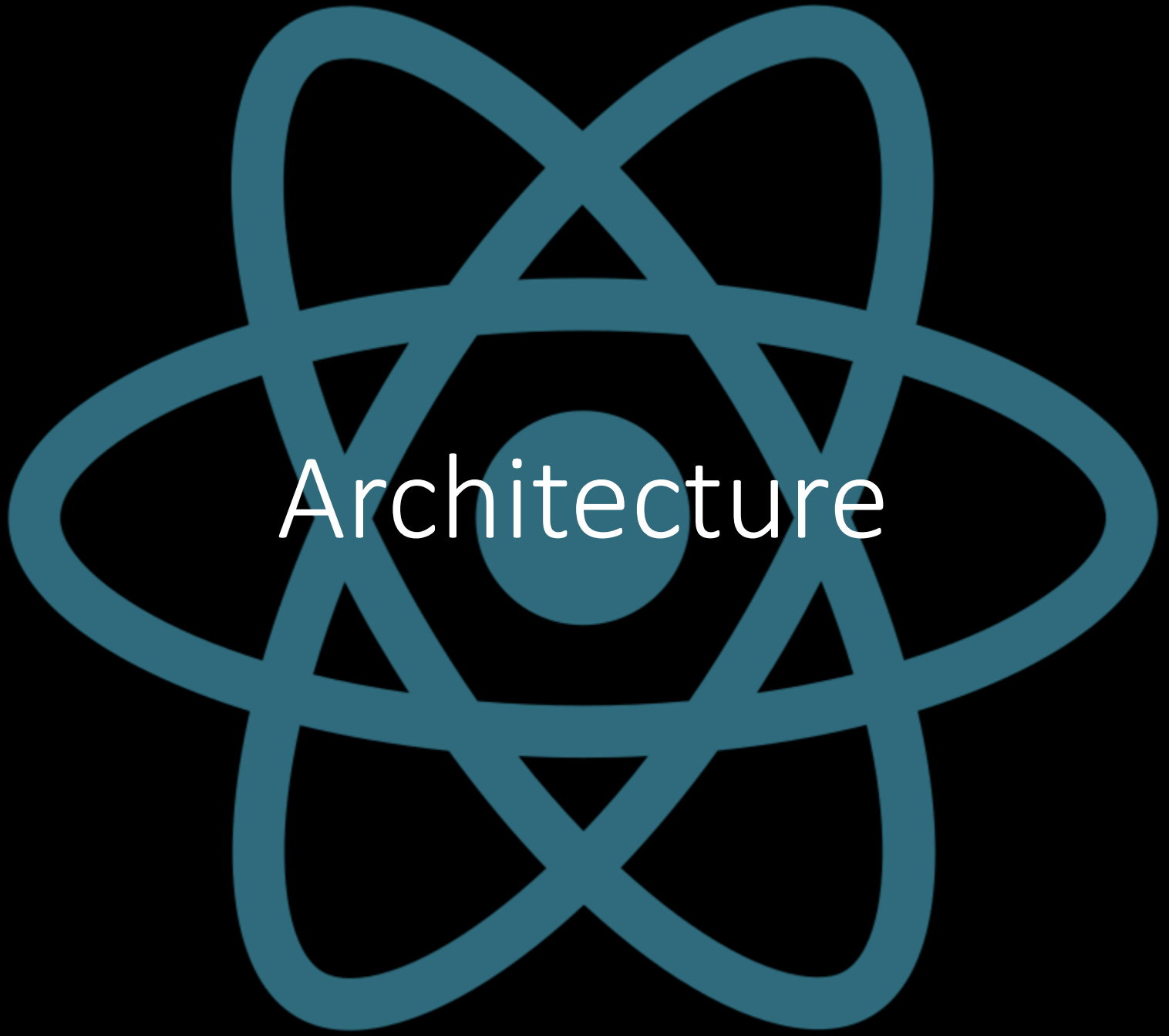
```
C:[redacted]\react-native-example>npx react-native init MyApp
```

- Create the module

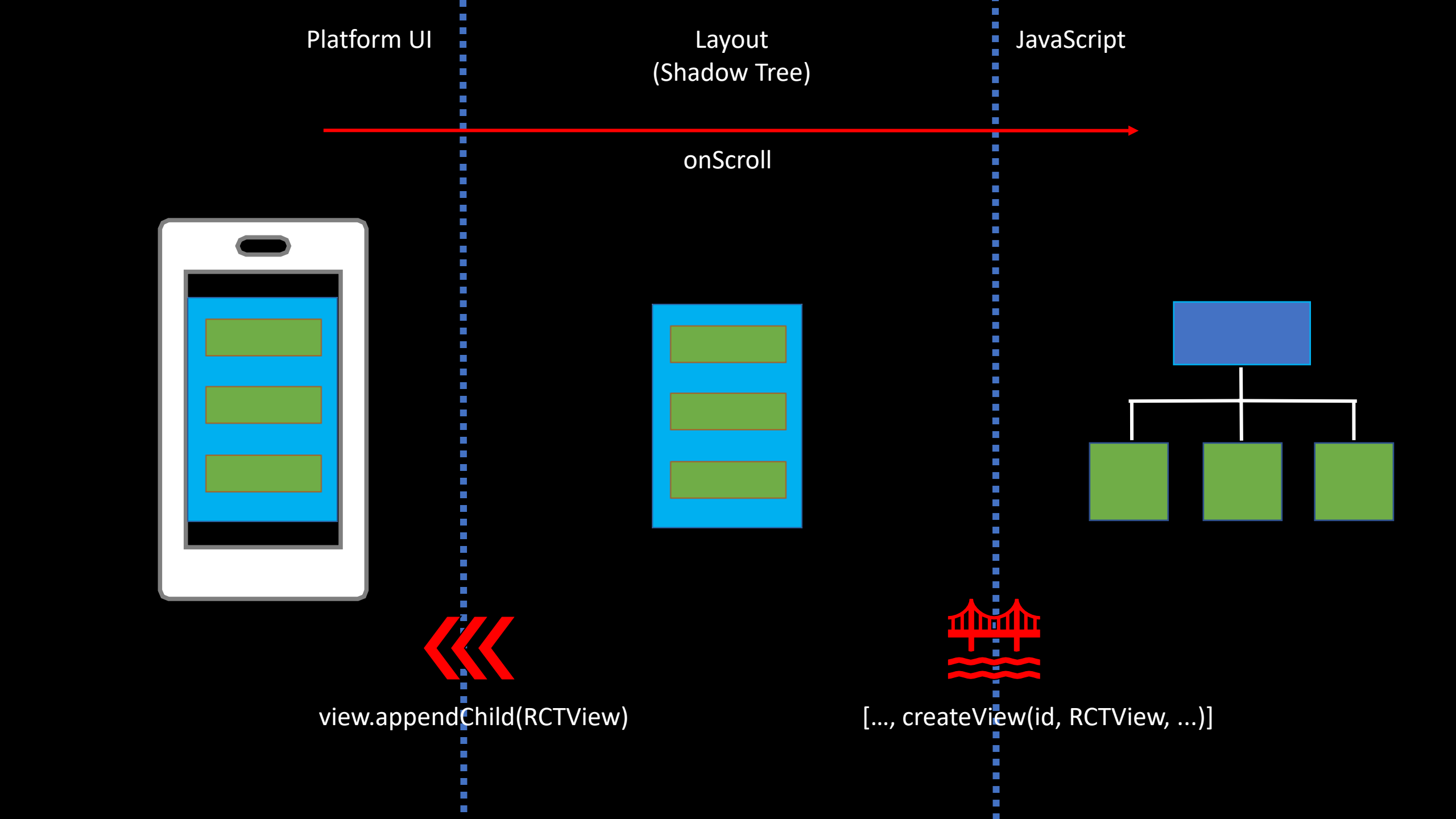
```
C:[redacted]\react-native-example>yarn global add create-react-native-module
```

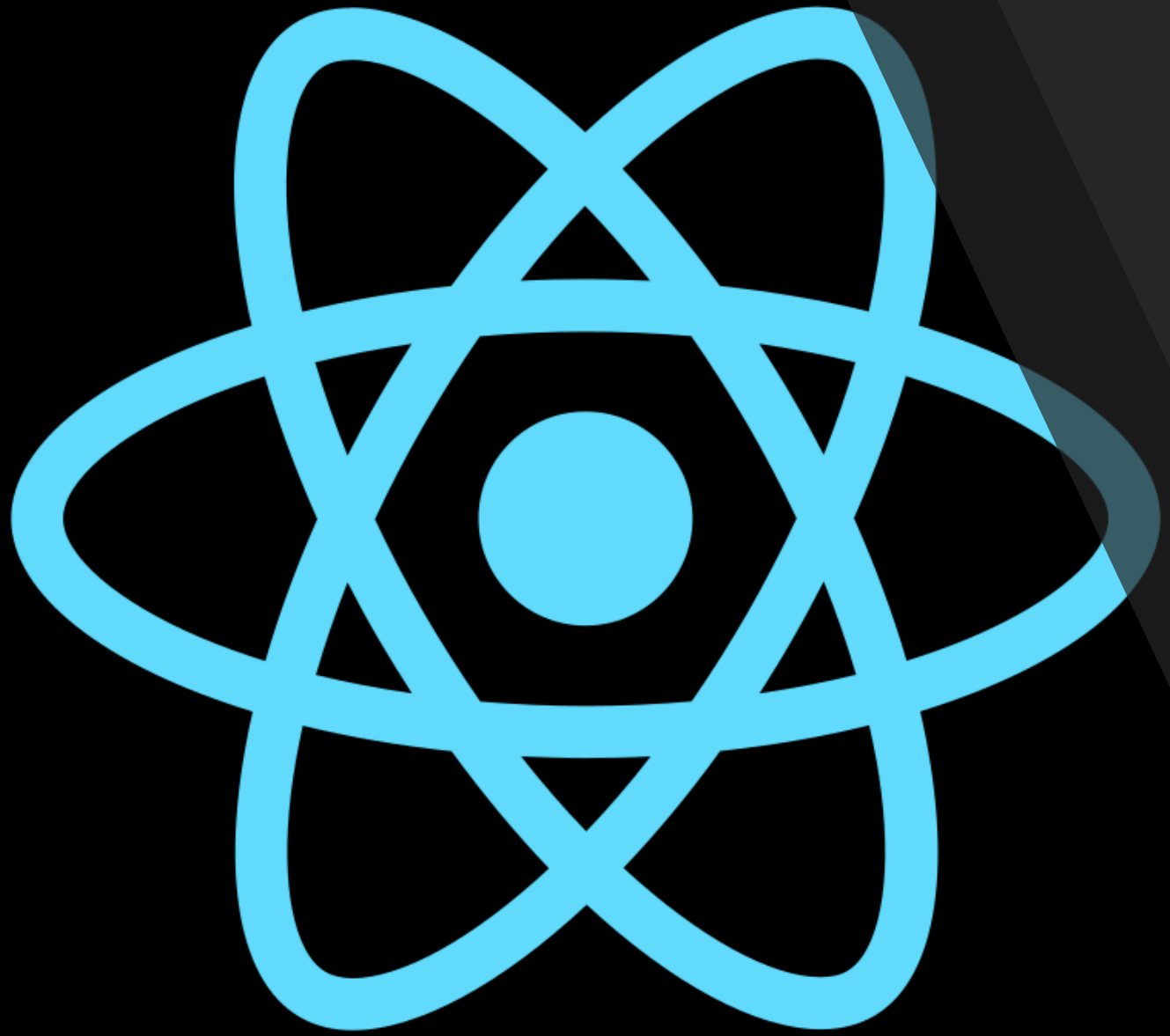
```
C:\[redacted]\react-native-example>create-react-native-module MyModule
```

- Add module in package.json



Architecture


Platform UI
Layout
(Shadow Tree)
JavaScript
onScroll
view.appendChild(RCTView)
[..., createView(id, RCTView, ...)]

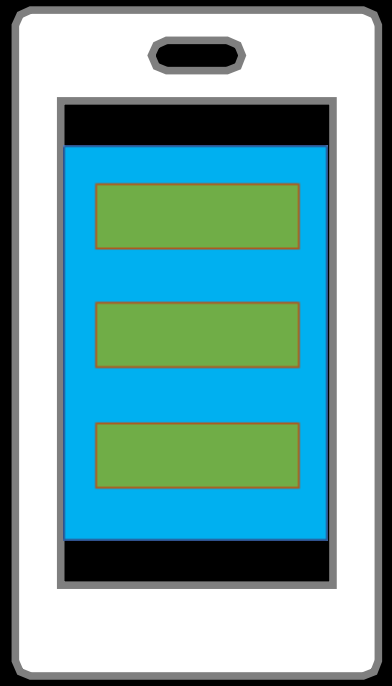


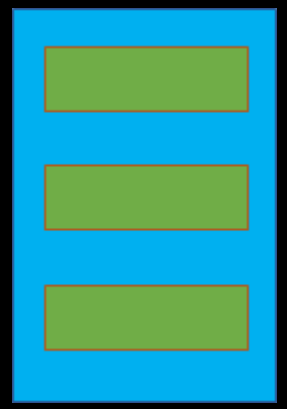
Fabric

Platform UI

Layout

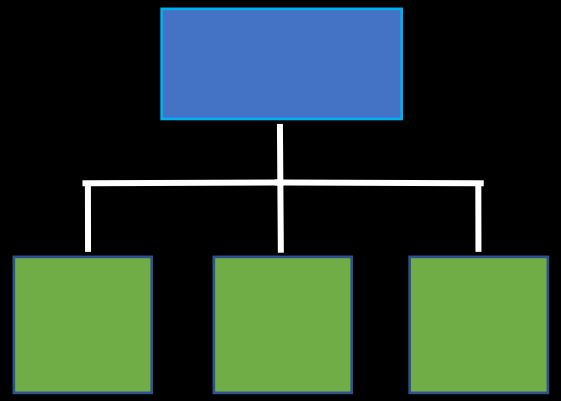
JavaScript





getNativeModule('GPS').getCoordinates()

JavaScript Interface (JSI)



Native Modules

getNativeModule('GPS').getCoordinates()

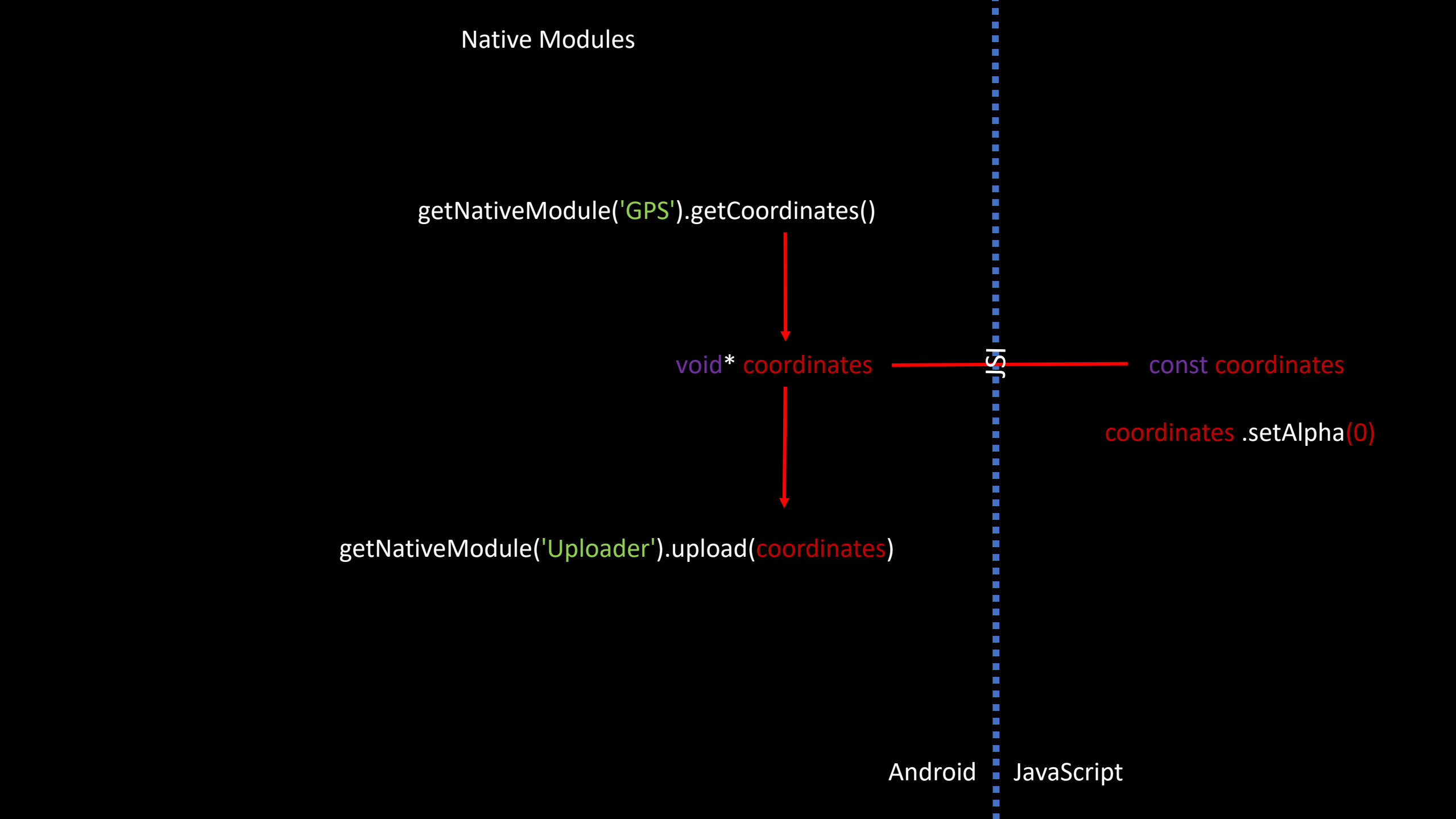
JSON

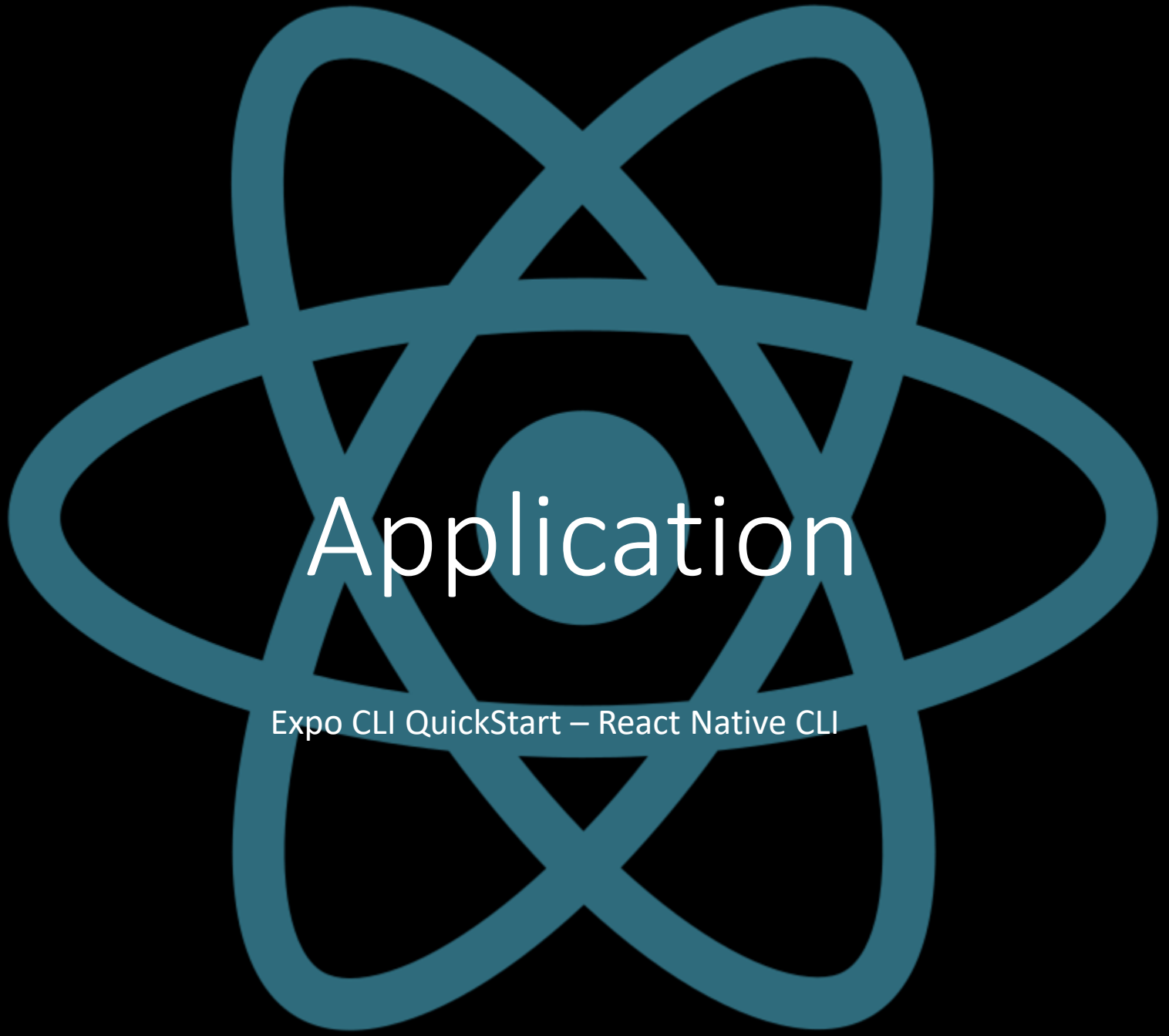
const coordinates

getNativeModule('Uploader').upload(coordinates)

JSON

Android JavaScript


Native Modules
getNativeModule('GPS').getCoordinates()
void* coordinates
JSI
const coordinates
coordinates .setAlpha(0)
getNativeModule('Uploader').upload(coordinates)
Android
JavaScript

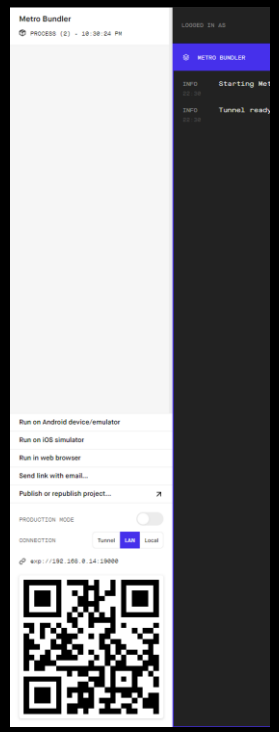


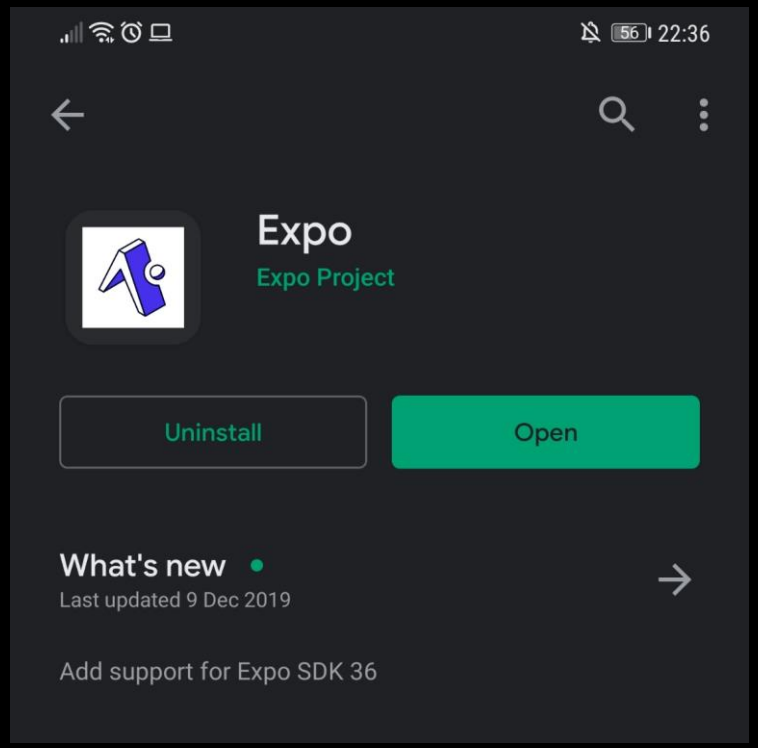
Application

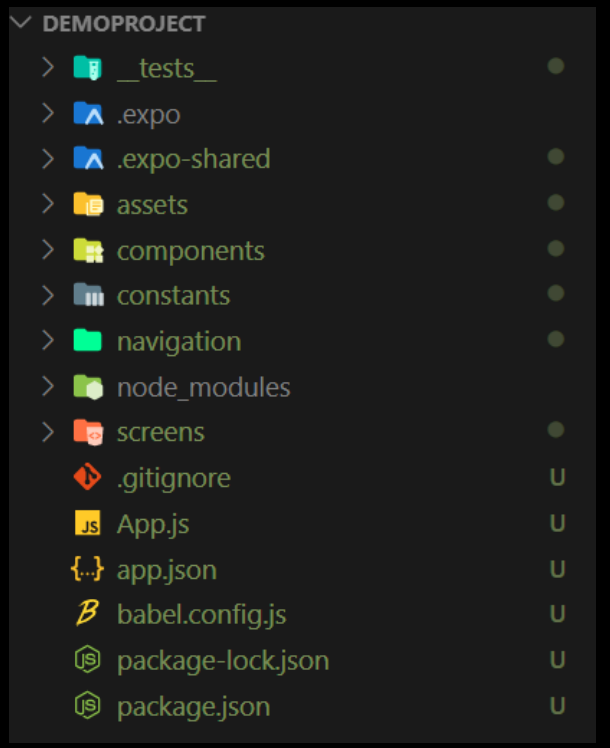
Expo CLI QuickStart – React Native CLI

Expo CLI

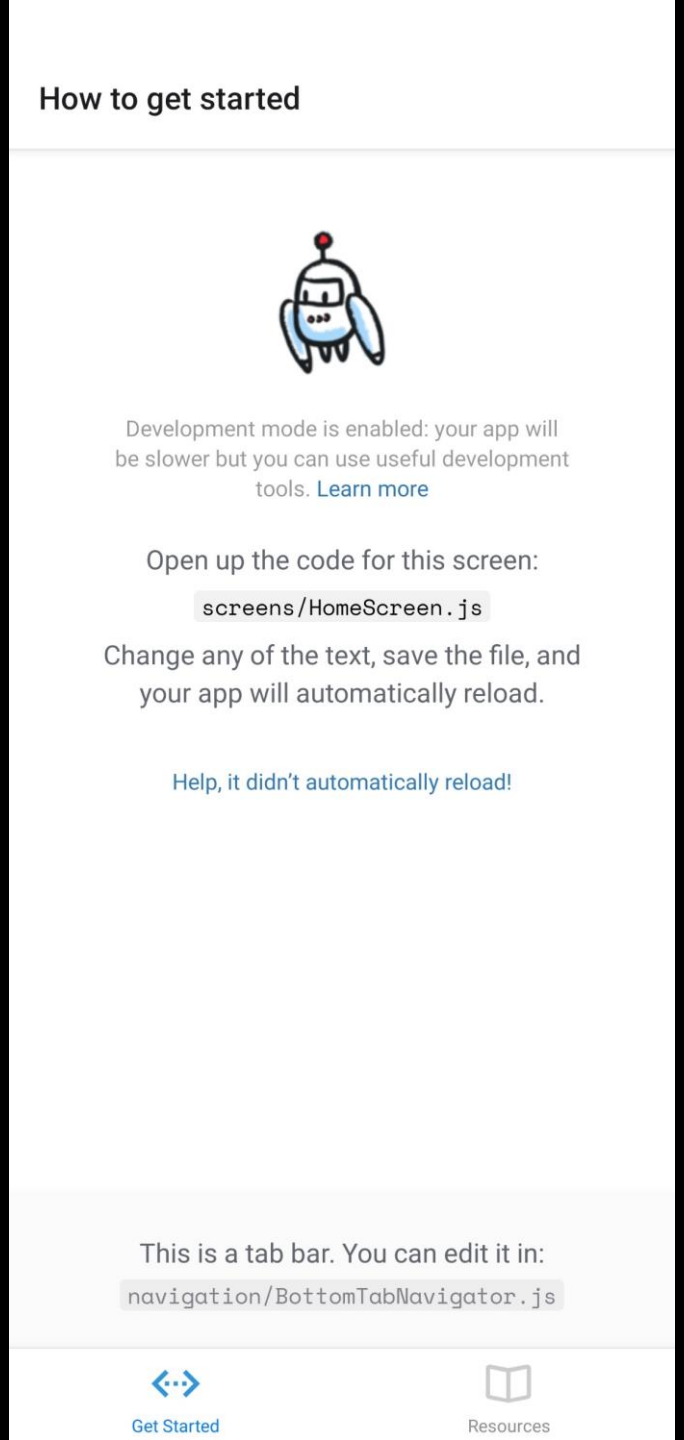
```
C:\Users\AGM-PC\Desktop\AWSDEMO>npm install -g expo-cli
C:\Users\AGM-PC\Desktop\AWSDEMO>expo init demoProject
```

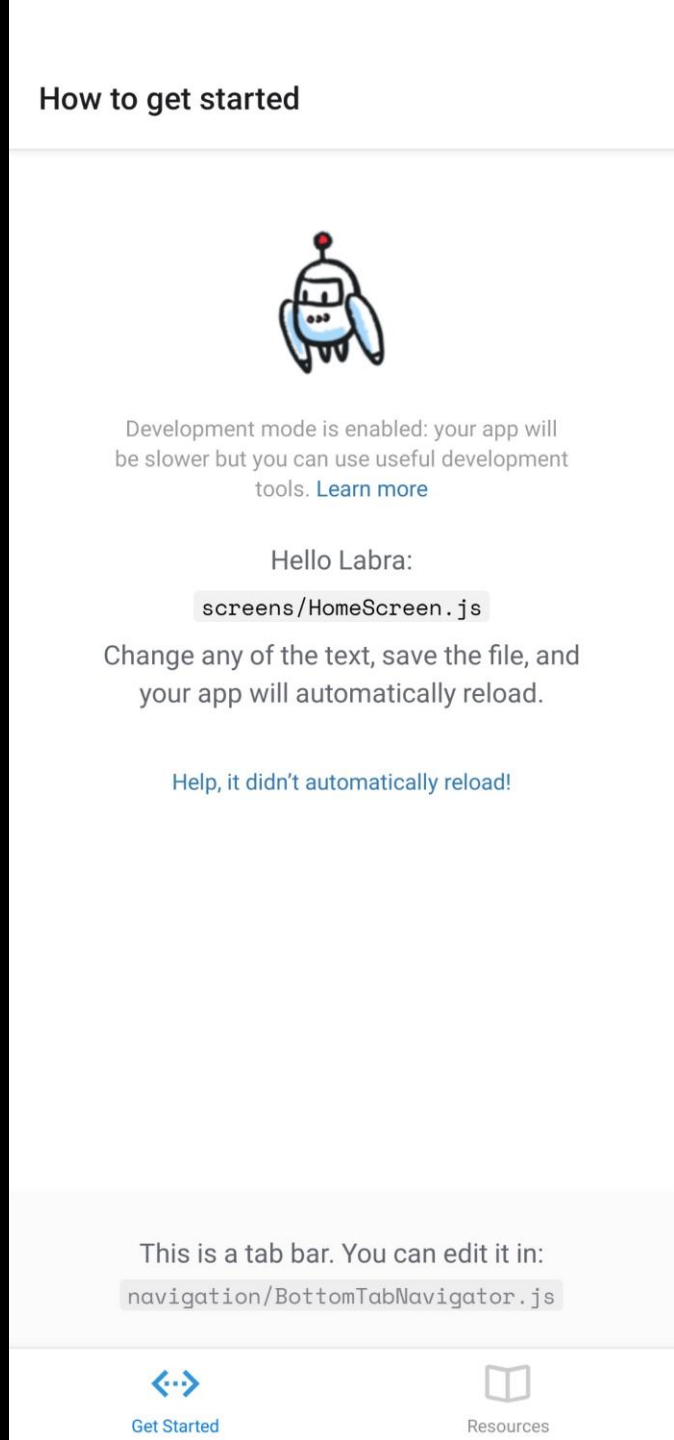






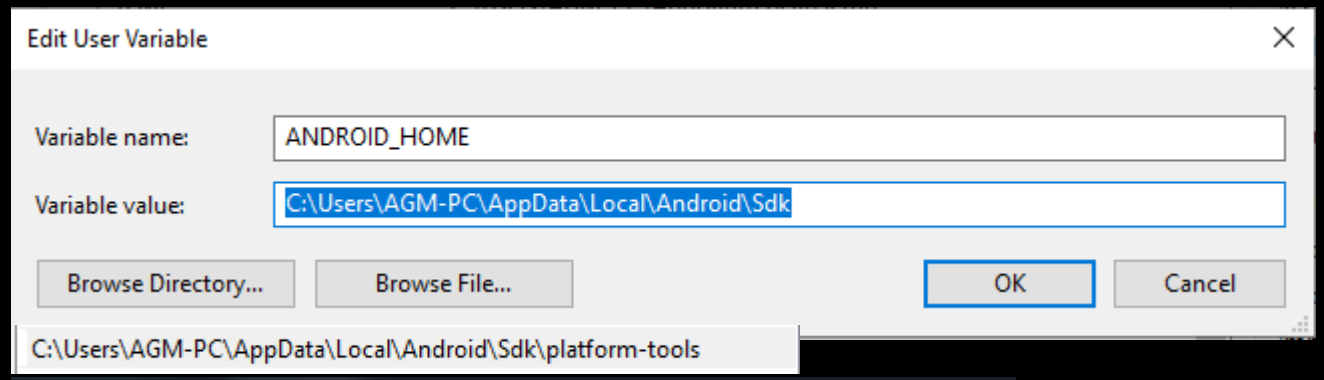
```jsx
import * as React from "react";
import {
  Image,
  Platform,
  StyleSheet,
  Text,
  TouchableOpacity,
  View
} from "react-native";
import { ScrollView } from "react-native-gesture-handler";
import * as WebBrowser from "expo-web-browser";

import { MonoText } from "../components/StyledText";

export default function HomeScreen() {
  return (
    <View style={styles.container}>
      <ScrollView
        style={styles.container}
        contentContainerStyle={styles.contentContainer}
      >
        <View style={styles.welcomeContainer}>
          <Image
            source={
              __DEV__
                ? require("../assets/images/robot-dev.png")
                : require("../assets/images/robot-prod.png")
            }
            style={styles.welcomeImage}
          />
        </View>

        <View style={styles.getStartedContainer}>
          <DevelopmentModeNotice />

          <Text style={styles.getStartedText}>Hello Labra:</Text>
```



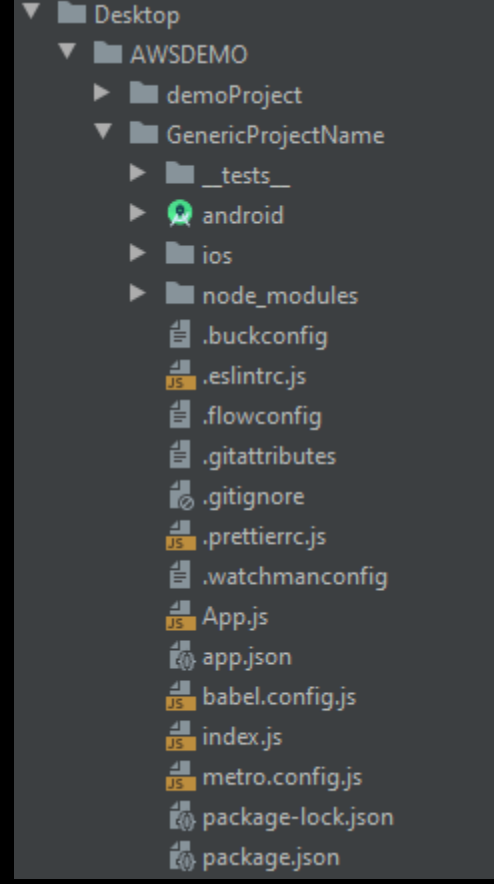


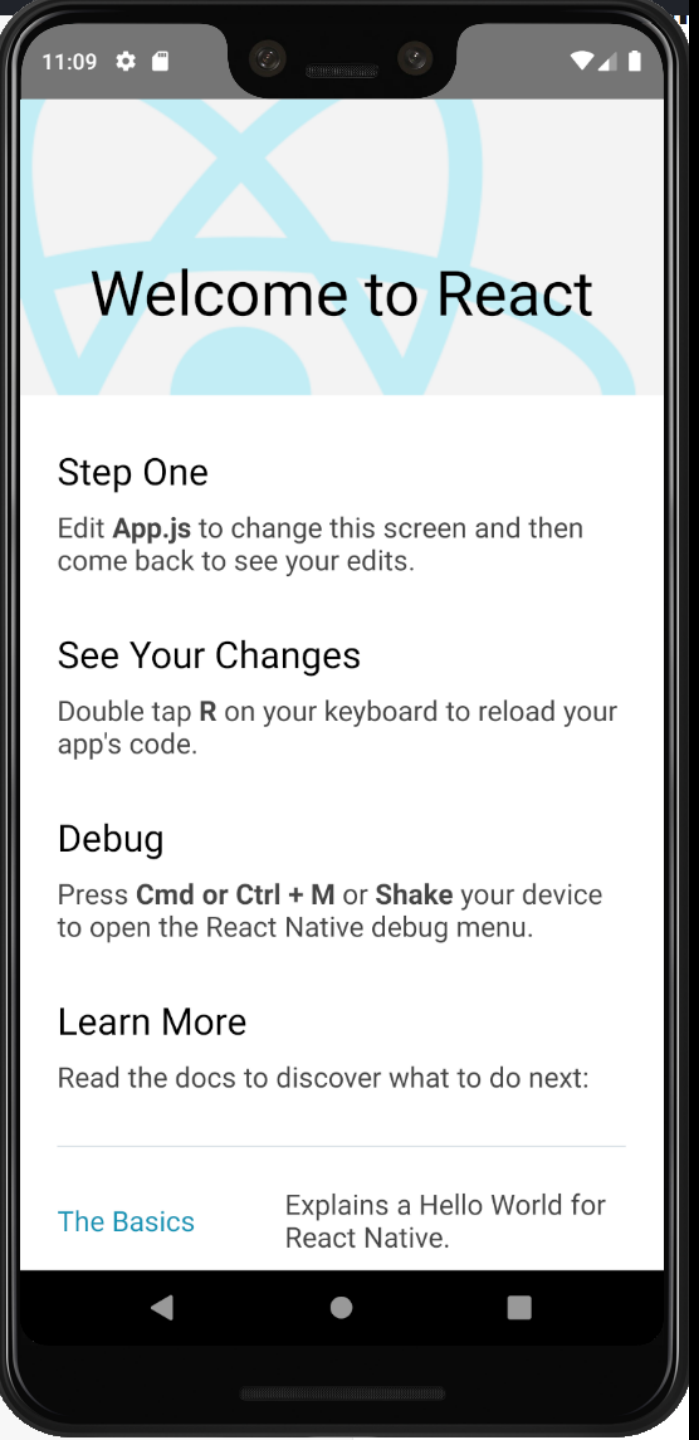
React CLI

- Requirements: Node, python2 and jdk
- Android Studio:
  - Android SDK
  - Android SDK Platform
  - Intel HAXM (or AMD equivalent)
  - Android Virtual Device




```
C:\Users\AGM-PC\Desktop\AWSDEMO>npx react-native
C:\Users\AGM-PC\Desktop\AWSDEMO>npx react-native init GenericProjectName

C:\Users\AGM-PC\Desktop\AWSDEMO\GenericProjectName>npx react-native start

C:\Users\AGM-PC\Desktop\AWSDEMO\GenericProjectName>npx react-native run-android
```





EXPLORER

OPEN EDITORS
- App.js 1

GENERICPROJECTNAME
- __tests__
- android
- ios
- node_modules
- .buckconfig
- .eslintrc.js
- .flowconfig
- .gitattributes
- .gitignore
- .prettierrc.js
- .watchmanconfig
- App.js 1
- app.json
- babel.config.js
- index.js
- metro.config.js
- package-lock.json
- package.json

OUTLINE
NPM SCRIPTS
MAVEN PROJECTS
SPRING-BOOT DASHBOARD

App.js

App.js > ...

```
    DebugInstructions,
    ReloadInstructions,
} from 'react-native/Libraries/NewAppScreen';

const App: () => React$Node = () => {
  return (
    <>
      <St (alias) class ScrollView tent" />
      <Sa import ScrollView
        <ScrollView
          contentInsetAdjustmentBehavior="automatic"
          style={styles.scrollView}>
          <Header />
          {global.HermesInternal == null ? null : (
            <View style={styles.engine}>
              <Text style={styles.footer}>Engine: Hermes</Text>
            </View>
          )}
          <View style={styles.body}>
            <View style={styles.sectionContainer}>
              <Text style={styles.sectionTitle}>Step One</Text>
              <Text style={styles.sectionDescription}>
                Edit <Text style={styles.highlight}>App.js</Text> to change this
                screen and then come back to see your edits.
              </Text>
            </View>
            <View style={styles.sectionContainer}>
              <Text style={styles.sectionTitle}>See Your Changes</Text>
              <Text style={styles.sectionDescription}>
                <ReloadInstructions />
              </Text>
            </View>
            <View style={styles.sectionContainer}>
              <Text style={styles.sectionTitle}>Debug</Text>
              <Text style={styles.sectionDescription}>
                <DebugInstructions />
              </Text>
            </View>
            <View style={styles.sectionContainer}>
              <Text style={styles.sectionTitle}>Learn More</Text>
              <Text style={styles.sectionDescription}>
                Read the docs to discover what to do next:
              </Text>
            </View>
            <LearnMoreLinks />
          </View>
        </ScrollView>
      </SafeAreaView>
```

Thanks for your attention!