Build great desktop experiences for Windows and macOS with React Native

Matteo Pagani – Windows App Consult Engineer @ Microsoft
matteo.pagani@microsoft.com

@qmatteoq qmatteoq

{codemotion}

Online Tech Conference

- Italian edition -

23-24-25 Marzo, 2021

What is React?

- React is a UI library which allows to build user interfaces with JavaScript
- It isn’t a full-fledged framework, like Angular
- It’s built by Facebook
- It’s open source https://github.com/facebook/react

Components

- Each section of a page is ideally a component
- The component defines both the UI and the logic
- You can use **props** to pass data from one component to another
- You can use **state** to store data in a component

Define the UI with JSX

- JSX is a syntax expression to JavaScript based on XML
- You can mix UI components with JavaScript code

```
const MyComponent = () => {

  const name = "John";

  return (
    <p>Hello {name}</p>
  );
}
```

```
const MyComponent = () => {

  const name = "John";

  return (
    <p>Hello {formatName(name)}
    </p>
  );
}
```

What is React Native?

- It's a React implementation in which JSX renders native controls for mobile platforms
- The UI isn't rendered through a web view, but it generates a native UI using the controls offered by the platform
- You can build native modules to leverage native APIs of the platform

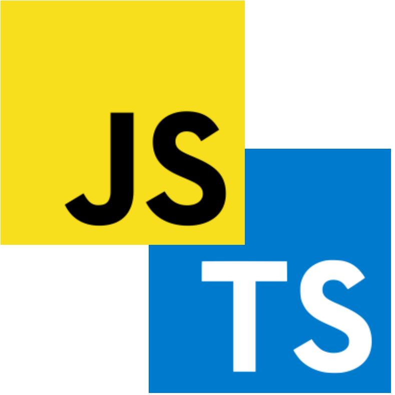
Microsoft helps make React Native great for desktop

More input devices

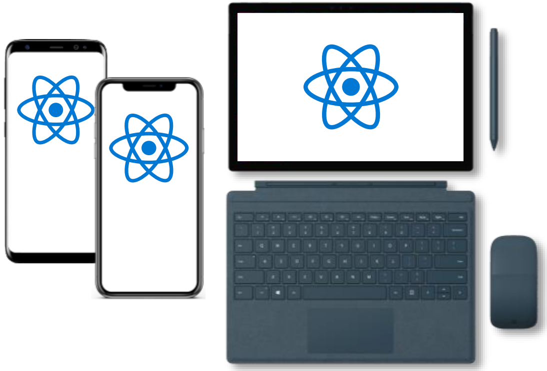
Better performance

Open source on GitHub

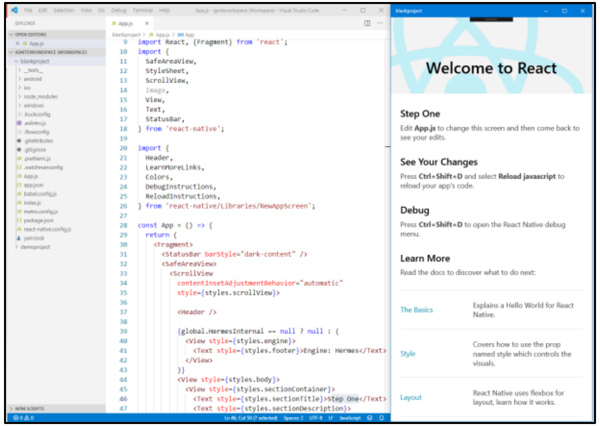
Developer Benefits of React Native



App written in
JavaScript or TypeScript



Build native apps
with native experiences



```
import React, {Fragment} from 'react';
import {
  SafeAreaView,
  StyleSheet,
  ScrollView,
  Image,
  View,
  Text,
  StatusBar,
} from 'react-native';

import {
  Header,
  LearnMoreLinks,
  Colors,
  DebugInstructions,
  ReloadInstructions,
} from 'react-native/Libraries/NewAppScreen';

const App = () => {
  return (
    <Fragment>
      <StatusBar barStyle="dark-content" />
      <SafeAreaView>
        <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}>
```

Welcome to React

Step One

Edit **App.js** to change this screen and then come back to see your edits.

See Your Changes

Press **Ctrl+Shift+D** and select **Reload javascript** to reload your app's code.

Debug

Press **Ctrl+Shift+D** to open the React Native debug menu.

Learn More

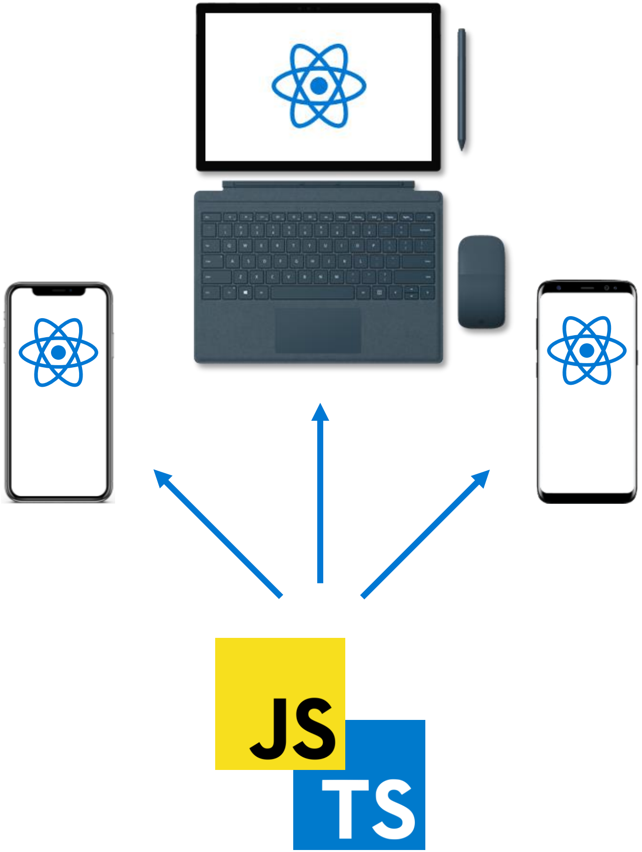
Read the docs to discover what to do next:

The Basics	Explains a Hello World for React Native.
Style	Covers how to use the prop named style which controls the visuals.
Layout	React Native uses flexbox for layout, learn how it works.

Fast, familiar,
web-like dev loop

Benefits of React Native

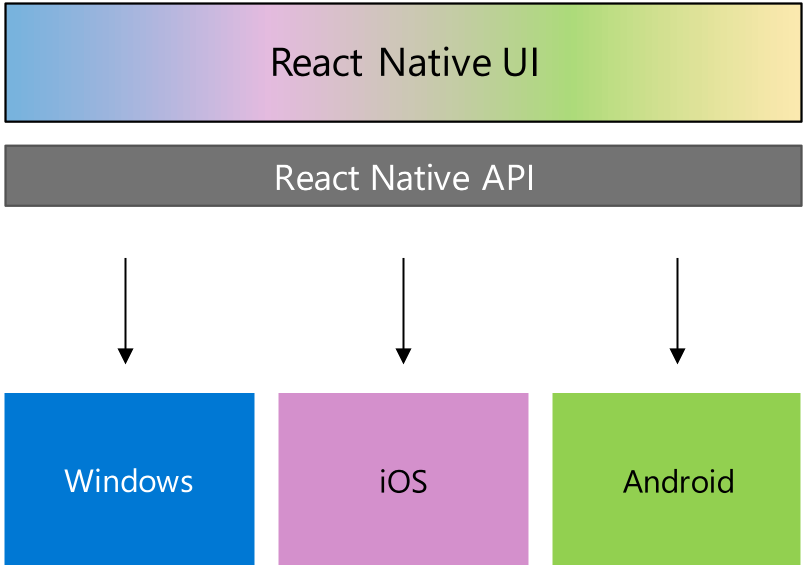
- **Build apps for Windows and more using JavaScript or TypeScript**



Benefits of React Native

- Build apps for Windows and more using JavaScript, TypeScript

- **Fully native apps with support for fully native experiences**



Benefits of React Native

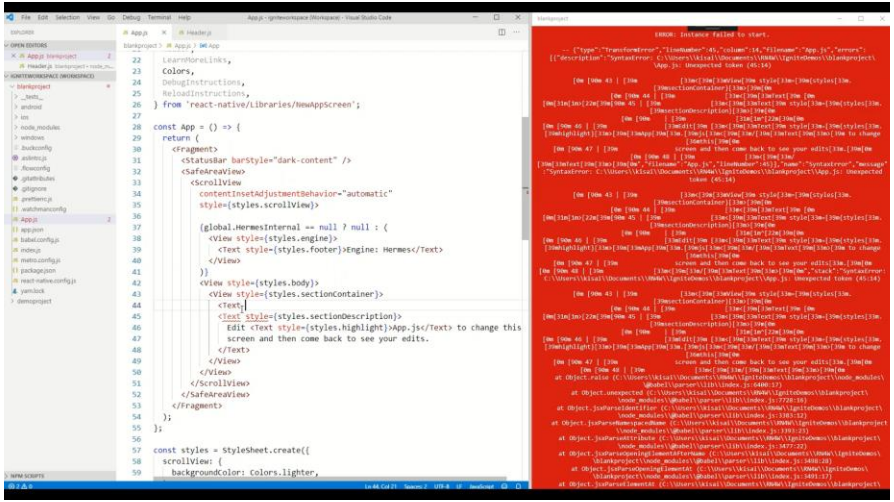
```
<Picker selectedValue={this.state.color} style={{height: 50, width: 200}}
   onValueChange={(itemValue, itemIndex) => this.setState({color: itemValue})}>
   <Picker.Item label="Blue" value="blue" />
   <Picker.Item label="Green" value="green" />
   <Picker.Item label="Yellow" value="yellow" />
</Picker>
```

Benefits of React Native

- Build apps for Windows and more using JavaScript, TypeScript

- Fully native apps with support for fully native experiences

- **Fast, familiar, web-like developer experience and ecosystem**



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

const App = () => {
  return (
    <Fragment>
      <StatusBar barStyle="dark-content" />
      <SafeAreaView>
        <ScrollView
          contentInsetAdjustmentBehavior="automatic"
          style={styles.scrollView}>

          {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
              <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>
        </ScrollView>
      </SafeAreaView>
    </Fragment>
  );
};

const styles = StyleSheet.create({
  scrollView: {
    backgroundColor: Colors.lighter,
```

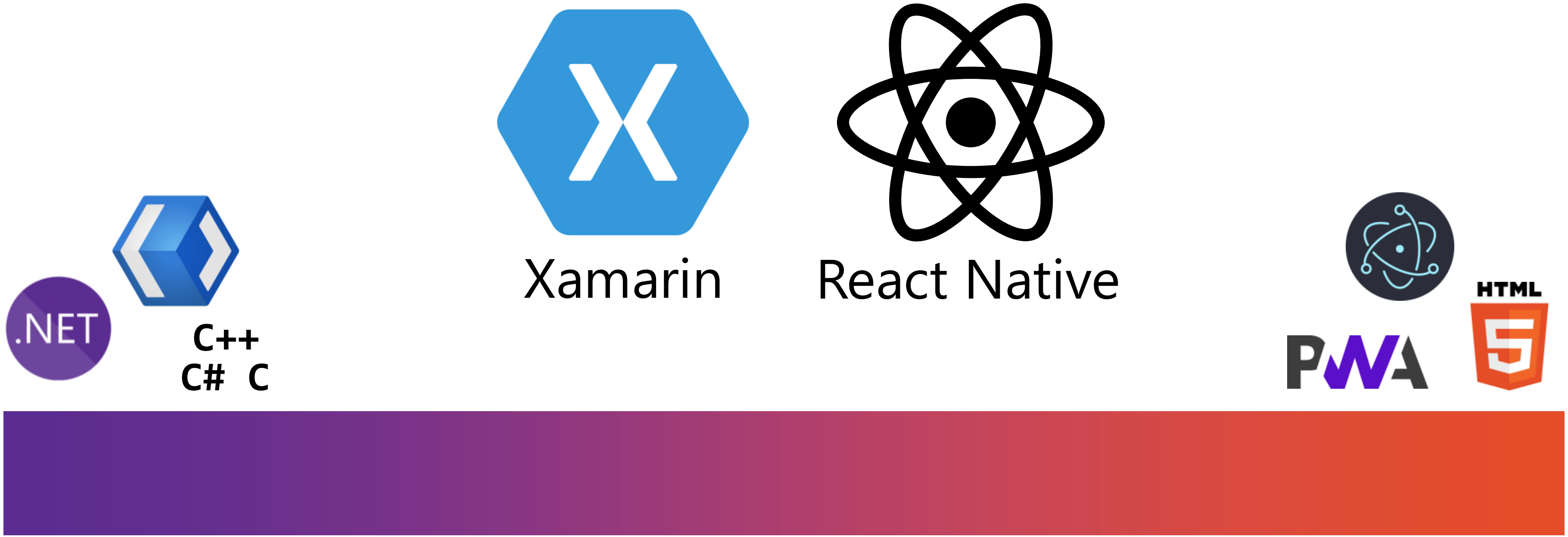
```
ERROR: Instance failed to start.

-- {"type":"TransformError","lineNumber":45,"column":14,"filename":"App.js","errors":
[{"description":"SyntaxError: C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\\blankproject\
\App.js: Unexpected token (45:14)

[0m [90m 43 | [39m          [33m<[39m[33mView[39m style[33m=[39m{styles[33m.
[39msectionContainer}[33m>[39m[0m
[0m [90m 44 | [39m            [33m<[39m[33mText[39m [0m
[0m[31m[1m>[22m[39m[90m 45 | [39m            [33m<[39m[33mText[39m style[33m=[39m{styles[33m.
[39msectionDescription}[33m>[39m[0m
[0m [90m    | [39m             [31m[1m^[22m[39m[0m
[0m [90m 46 | [39m              [33mEdit[39m [33m<[39m[33mText[39m style[33m=[39m{styles[33m.
[39mhighlight}[33m>[39m[33mApp[39m[33m.[39mjs[33m<[39m[33m/[39m[33mText[39m[33m>[39m to change
[36mthis[39m[0m
[0m [90m 47 | [39m              screen and then come back to see your edits[33m.[39m[0m
[0m [90m 48 | [39m            [33m<[39m[33m/
[39m[33mText[39m[33m>[39m[0m","filename":"App.js","lineNumber":45}],"name":"SyntaxError","message"
:"SyntaxError: C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\\blankproject\\App.js: Unexpected
token (45:14)

[0m [90m 43 | [39m          [33m<[39m[33mView[39m style[33m=[39m{styles[33m.
[39msectionContainer}[33m>[39m[0m
[0m [90m 44 | [39m            [33m<[39m[33mText[39m [0m
[0m[31m[1m>[22m[39m[90m 45 | [39m            [33m<[39m[33mText[39m style[33m=[39m{styles[33m.
[39msectionDescription}[33m>[39m[0m
[0m [90m    | [39m             [31m[1m^[22m[39m[0m
[0m [90m 46 | [39m              [33mEdit[39m [33m<[39m[33mText[39m style[33m=[39m{styles[33m.
[39mhighlight}[33m>[39m[33mApp[39m[33m.[39mjs[33m<[39m[33m/[39m[33mText[39m[33m>[39m to change
[36mthis[39m[0m
[0m [90m 47 | [39m              screen and then come back to see your edits[33m.[39m[0m
[0m [90m 48 | [39m            [33m<[39m[33m/[39m[33mText[39m[33m>[39m[0m","stack":"SyntaxError:
C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\\blankproject\\App.js: Unexpected token (45:14)

[0m [90m 43 | [39m          [33m<[39m[33mView[39m style[33m=[39m{styles[33m.
[39msectionContainer}[33m>[39m[0m
[0m [90m 44 | [39m            [33m<[39m[33mText[39m [0m
[0m[31m[1m>[22m[39m[90m 45 | [39m            [33m<[39m[33mText[39m style[33m=[39m{styles[33m.
[39msectionDescription}[33m>[39m[0m
[0m [90m    | [39m             [31m[1m^[22m[39m[0m
[0m [90m 46 | [39m              [33mEdit[39m [33m<[39m[33mText[39m style[33m=[39m{styles[33m.
[39mhighlight}[33m>[39m[33mApp[39m[33m.[39mjs[33m<[39m[33m/[39m[33mText[39m[33m>[39m to change
[36mthis[39m[0m
[0m [90m 47 | [39m              screen and then come back to see your edits[33m.[39m[0m
[0m [90m 48 | [39m            [33m<[39m[33m/[39m[33mText[39m[33m>[39m[0m
    at Object.raise (C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\\blankproject\\node_modules\
\@babel\\parser\\lib\\index.js:6400:17)
    at Object.unexpected (C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\\blankproject\
\node_modules\\@babel\\parser\\lib\\index.js:7728:16)
    at Object.jsxParseIdentifier (C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\\blankproject\
\node_modules\\@babel\\parser\\lib\\index.js:3383:12)
    at Object.jsxParseNamespacedName (C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\\blankproject
\\node_modules\\@babel\\parser\\lib\\index.js:3393:23)
    at Object.jsxParseAttribute (C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\\blankproject\
\node_modules\\@babel\\parser\\lib\\index.js:3477:22)
    at Object.jsxParseOpeningElementAfterName (C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\
\blankproject\\node_modules\\@babel\\parser\\lib\\index.js:3488:28)
    at Object.jsxParseOpeningElementAt (C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\
\blankproject\\node_modules\\@babel\\parser\\lib\\index.js:3491:17)
    at Object.jsxParseElementAt (C:\\Users\\kisai\\Documents\\RN4W\\IgniteDemos\\blankproject\
```

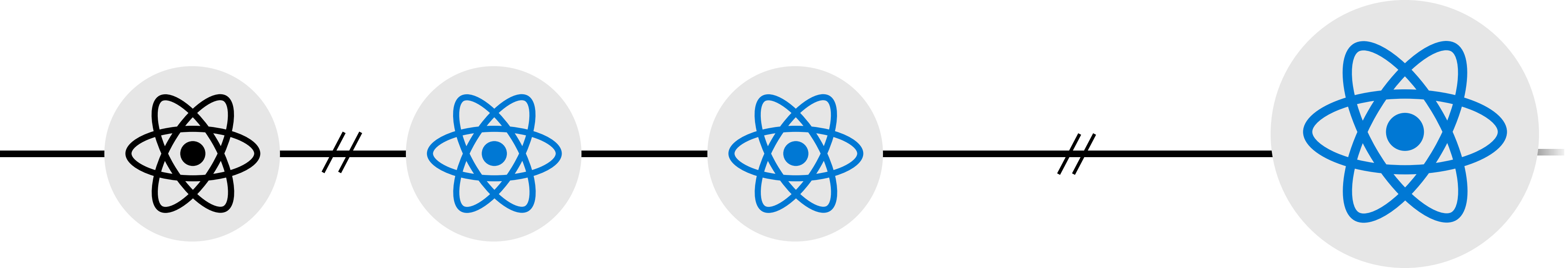


Native client app

Mobile-first, Cross platform

Website, Web app

React Native for Windows



2016

Legacy

Initial C# release

Spring '19

Milestone 1

High performance
C++ implementation

February 2020

Milestone 2

React Native for Windows 0.61 on the high-performance common core for iOS and Android

March 2021

0.64

Same day release for the latest React Native version released by Facebook

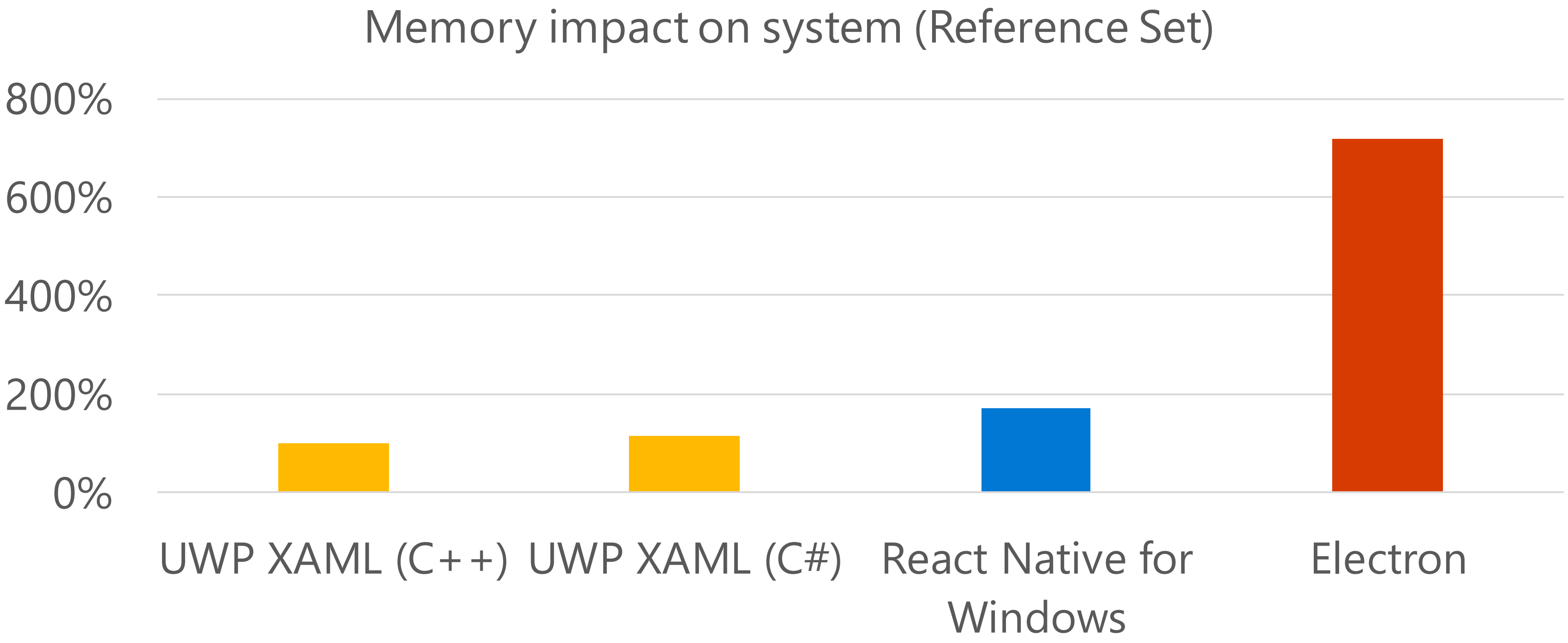
React Native for Windows

- High performant C++ implementation
- Same core as iOS and Android
- The output is a Universal Windows Platform application for Windows 10
- Support for Hermes as JavaScript engine

React Native for macOS

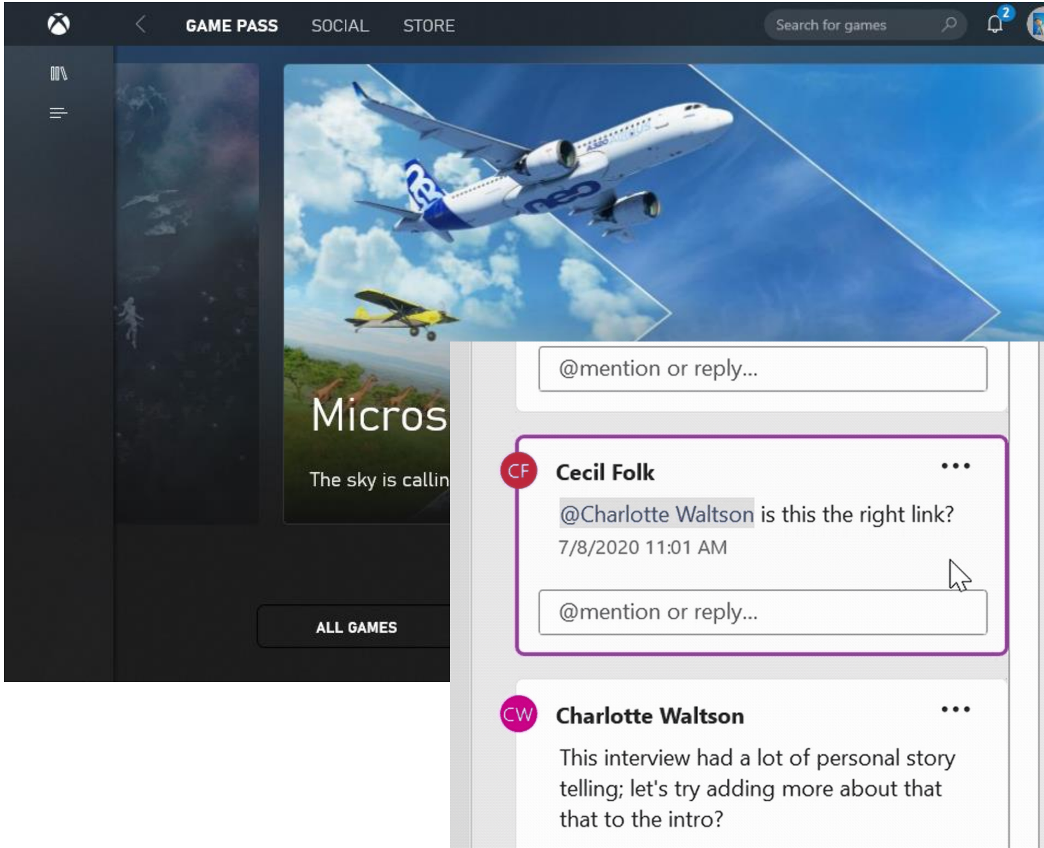
- The Microsoft team is building also a macOS implementation of React Native
- Same optimizations for mouse and keyboard scenarios
- Many iOS modules can be reused “as it is” due to the same framework and language

React Native for Windows memory performance



Modular development model

- Build end-to-end applications
- Include a React Native component in an existing Universal Windows Platform application
- Include a React Native component in an existing Win32 application



CF Cecil Folk

@Charlotte Waltson is this the right link?

7/8/2020 11:01 AM

@mention or reply...

CW Charlotte Waltson

This interview had a lot of personal story telling; let's try adding more about that that to the intro?

Requirements

Windows

- Node.js
- Windows 10 – minimum version 1703 (build 15063)
- Visual Studio 2019
  - Universal Windows Platform workload

Mac

- Node.js
- Xcode 11.3.1 or newer
- Cocoapods

Tools for development and debugging

- Visual Studio Code
- React Native Tools plugin (https://marketplace.visualstudio.com/items?itemName=msjsdiag.vscode-react-native)
- Setup launch.json and add breakpoint and watchers

Building a React Native Windows App

1. Create a new React Native project

```
> npx react-native init <projectName> --template react-native@^0.63.2
```

2. Add Windows support

```
> npx react-native-windows-init --overwrite
```

3. Run your app!

```
> react-native run-windows
```

Steps are documented on GitHub: https://microsoft.github.io/react-native-windows

Building a React Native MacOS App

1. Create a new React Native project

```
> npx react-native init <projectName> --template react-native@^0.63.2
```

2. Add MacOS support

```
> npx react-native-macos-init
```

3. Run your app!

```
> react-native run-macos
```

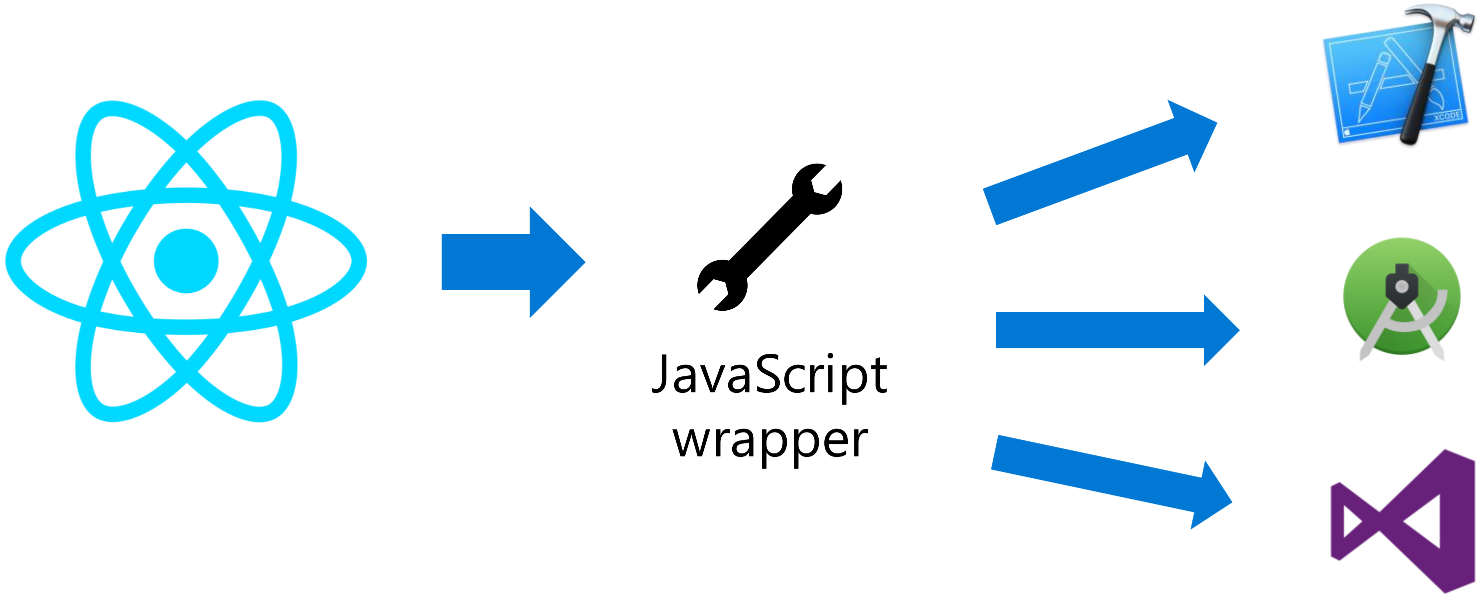
Steps are documented on GitHub: https://microsoft.github.io/react-native-windows

React Native for Windows

Demo

Native modules

- When you need to use a native API, JavaScript isn't enough
- A native module is a library built with the native API of the platform and exposed to JavaScript



3rd party modules

- Everything is a module, React Native provides only the basic UI framework
- Huge catalogue of 3rd party modules available on NPM
- Check on https://reactnative.directory/ if your favorite module already supports Windows and MacOS

Building native modules for Windows

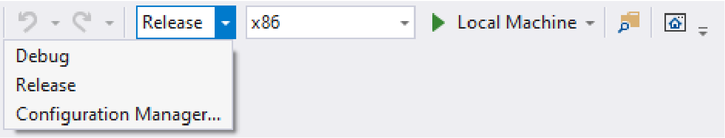
- SDK for C# and C++ to expose functions and objects to JavaScript
- Simple attributes that you can use to decorate your code
- General guidance:
  - If the module is internal only, choose between C# and C++
  - If the module is for the community or general consumption, stick to C++

Native modules

Demo

Deploying React Native applications

- Each platform will produce a package using its own format:
  - APK / ABB for Android
  - IPA for iOS (you will need a Mac)
  - MSIX for Windows
- The package contains a special bundle, which doesn't require the packager up & running like during the development
- In Windows:
  - Open the Visual Studio solution in the **windows** folder
  - Compile it in **Release** mode
  - Publish the package on the Microsoft Store or deploy it with sideloading, Intune, SSCM, etc.



Thanks!

Connect with me on LinkedIn

linkedIn.com/in/matteopagani

Matteo Pagani – Windows App Consult Engineer @ Microsoft
matteo.pagani@microsoft.com

@qmatteoq qmatteoq

{codemotion}

Online Tech Conference

- Italian edition -

23-24-25 Marzo, 2021