

React Native for Windows

Glamping in a swarm of bugs

Aaron Wirth

MADE WITH
beautiful.ai

*Join Us
Next Year!*

WI  24
THAT[®]
CONFERENCE

JULY 29TH - AUG. 1ST



SPECIAL THANKS TO ALL OUR AWESOME CAMP SPONSORS!



Unspecified
SOFTWARE CO





Aaron Wirth

- Software Developer at Modus
- Professional human
- Amateur Golfer / Dog Enthusiast / Traveler
- University of Wisconsin - Eau Claire

Agenda

1 | Getting the fire started

Quick look back on how we went from WinRT to RNW (and the dozens of other acronyms we passed on the hiking trail)

2 | Toasting some marshmallows

Demo time: Let's "build" a basic React Native for Windows app and experience the marshmallow-y goodness. Things might get sticky

3 | Deciding if it's worth making the s'more

Review time: React Native is great but it's not without its short comings.

Also, let's be honest, sometimes the graham crackers are on the other side of the fire and it's just not worth getting out of the camp chair

4 | Reflecting on our dessert choices

Question time. Maybe some answers.

An Adventure of Sorts

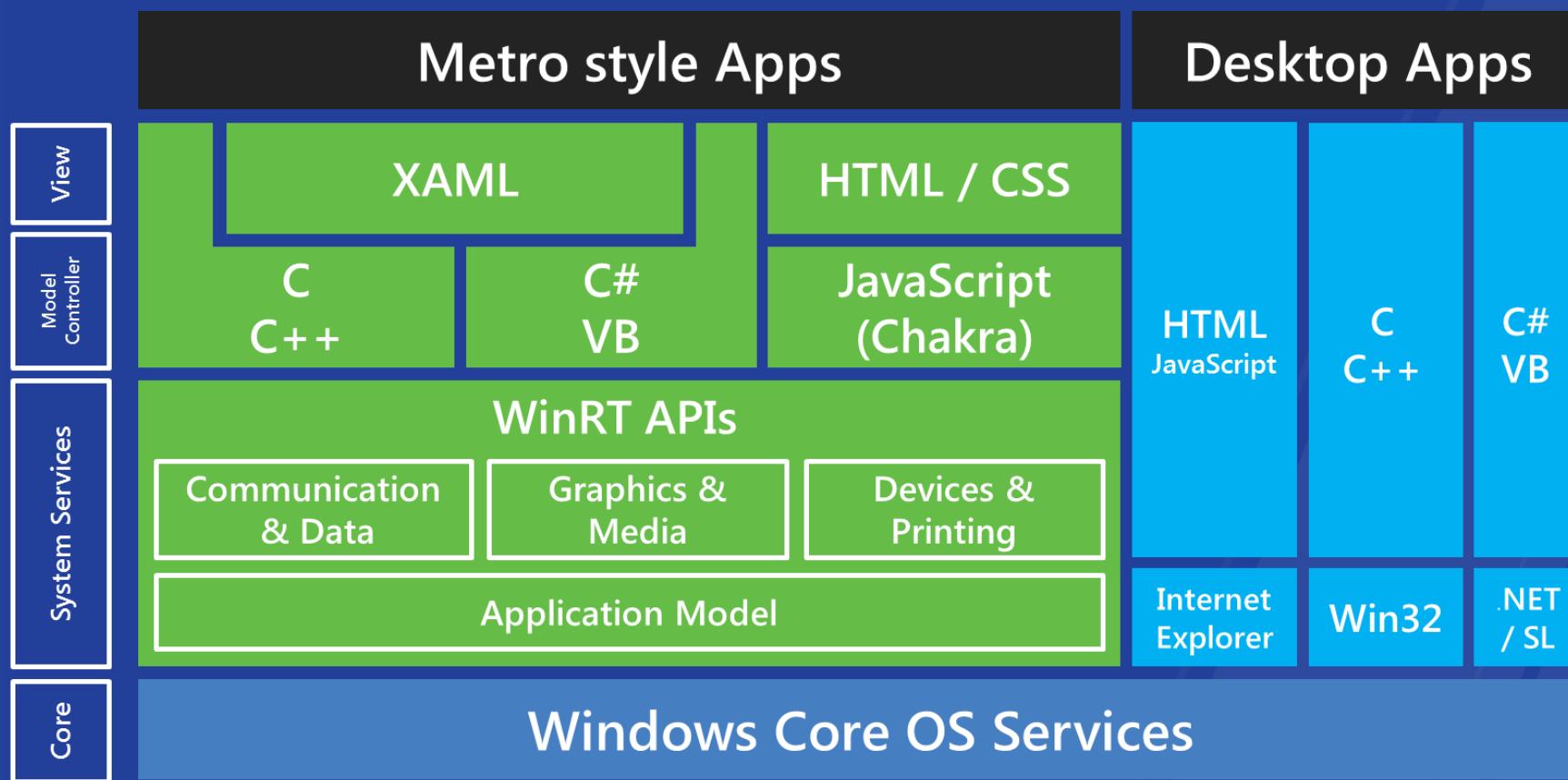
The Short but Perilous Journey of Windows Application Development



A large circular dashed line forms a path or track, starting from the bottom left, curving upwards and to the right, then downwards and to the left, creating a loop-like shape.

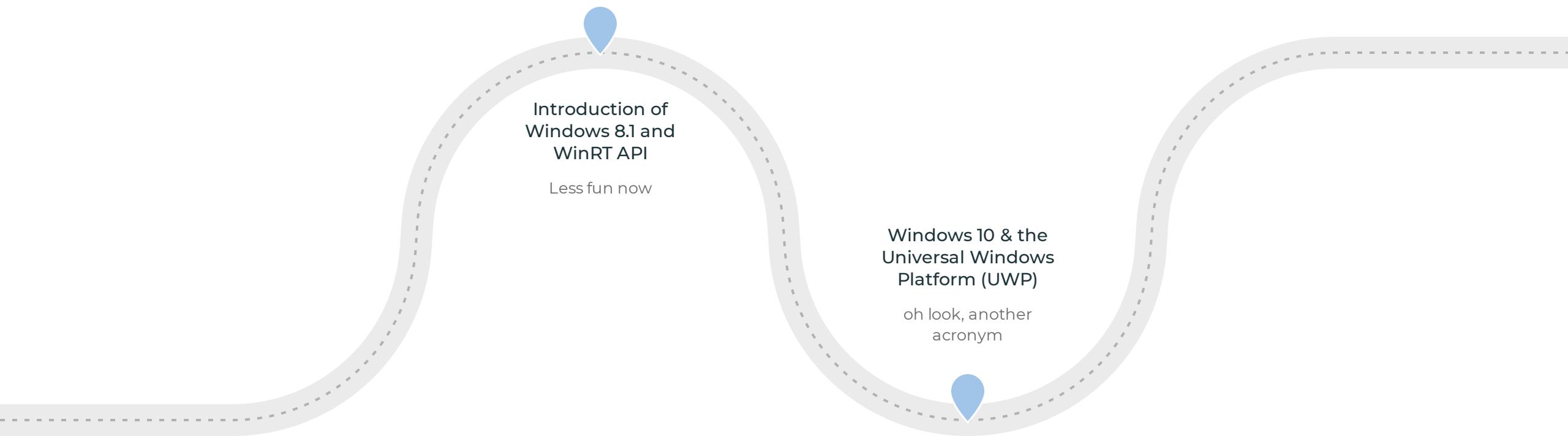
Introduction of
Windows 8.1 and
WinRT API

Runtime, more like
Fun-time.....



An Adventure of Sorts

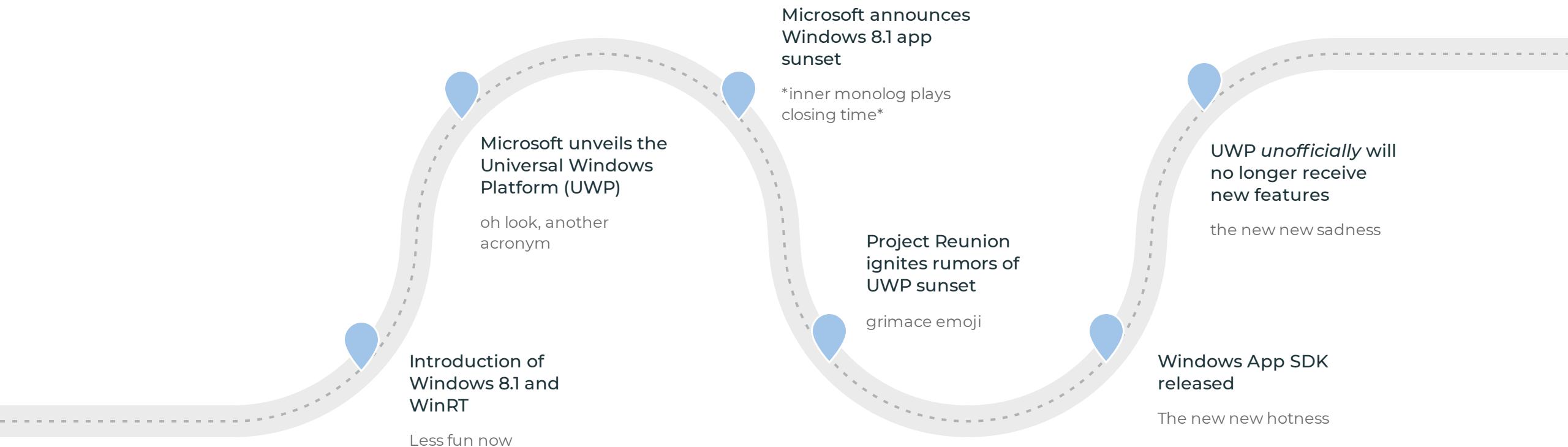
The Short but Perilous Journey of Windows Application Development





An Adventure of Sorts

The Short but Perilous Journey of Windows Application Development



Windows App SDK - Calling all Windows developers!

Previously known as Project Reunion.

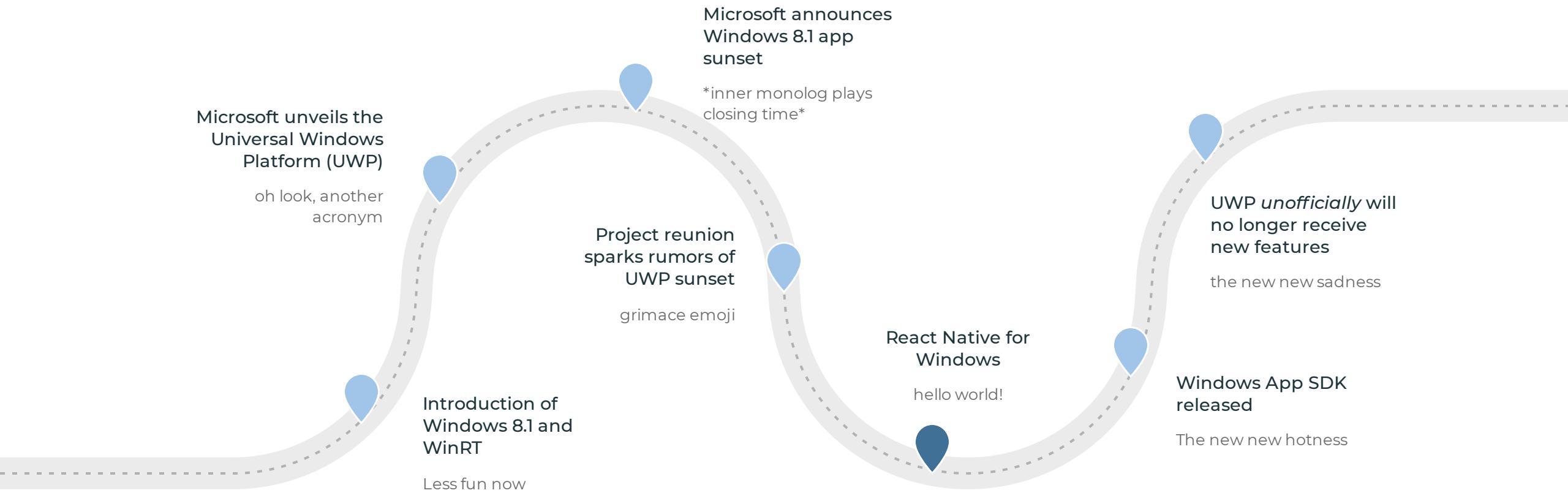
Delight users with modern UI	Access new platform features	Backwards compatible
 Powerful WinUI 3 UI	 Localization + power status + more	 Down to Windows 10 1809

...and more, without having to rewrite your app!

- Use your current installer (*no requirement to use MSIX, but there are reliability/security benefits to using MSIX!*)
- Additive platform APIs (*only add what you need, leave the rest of your app as-is*)
- Works with Win32, WPF, WinForms, and more apps

An Adventure of Sorts

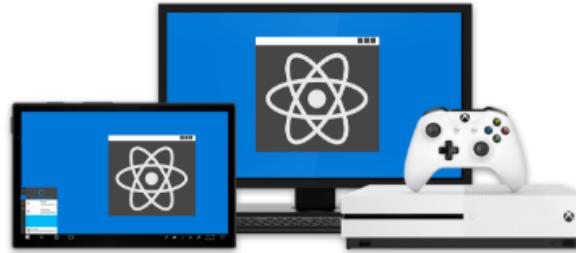
The Short but Perilous Journey of Windows Application Development





Follow @ReactNativeMSFT

React Native for Windows + macOS



Bring your React Native apps to some of the most powerful devices out there

Written in JavaScript—rendered with native code

React primitives render to native platform UI, meaning your app uses the same native platform APIs other apps do.

Many platforms, one React. Create platform-specific versions of components so a single codebase can share code across platforms. With React Native, one team can maintain multiple platforms and share a common technology—React.

```
import React from 'react';
import {Text, View} from 'react-native';
import {Header} from './Header';
import {heading} from './Typography';

const WelcomeScreen = () => (
  <View>
    <Header title="Welcome to React Native"/>
    <Text style={heading}>Step One</Text>
    <Text>
      Edit App.js to change this screen and turn it
      into your app.
    </Text>
    <Text style={heading}>See Your Changes</Text>
    <Text>
      Press Cmd + R inside the simulator to reload
      your app's code.
    </Text>
```

DEMO



Setup

Installing those pesky dependencies

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

C:\WINDOWS\system32> iex (New-Object System.Net.WebClient).DownloadString('https://aka.ms/rnw-vs2022-deps.ps1');
Checking Free space on current drive > 15 GB          OK
Checking Installed memory >= 16 GB      Failed (warn)
Checking Windows version >= 10.0.17763.0          OK
Checking Developer mode is on          OK
Checking Long path support is enabled          OK
Checking Visual Studio 2022 (>= 17.3) & req. components  Failed
you want to install? [y/N]:
```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> iex (New-Object System.Net.WebClient).DownloadString('https://aka.ms/rnw-vs2022-deps.ps1');
>>
Checking Free space on current drive > 15 GB          OK
Checking Installed memory >= 16 GB      OK
Checking Windows version >= 10.0.17763.0          OK
Checking Developer mode is on          OK
Checking Long path support is enabled          OK
Checking Visual Studio 2022 (>= 17.3) & req. components  OK
Checking Node.js (LTS, >= 16.0)          OK
Checking Yarn          OK
Checking .NET SDK (LTS, = 6.0)          OK
All mandatory requirements met.

Press Enter to exit:
```

Setup

Setup React Native, add Windows, and run

```
npx react-native init <projectName> --template "react-native@^0.72.0"
```



```
npx react-native-windows-init --overwrite --language cs
```



```
npx react-native run-windows
```



Setup

C#-hoosing your project language

	C++ APP	C# APP
C++ Module	✓	✓
C++ Module w/ C++ NuGet Dependency	✓	✓
C++ Module w/ C# NuGet Dependency	✗	✗
C# Module	✗	✓
C# Module w/ C# NuGet Dependency	✗	✓
C# Module w/ C++ NuGet Dependency	✗	✓

Core Components

ready-to-use Native Components

View

The most fundamental component for building a UI.

Text

A component for displaying text.

Image

A component for displaying images.

TextInput

A component for inputting text into the app via a keyboard.

ScrollView

Provides a scrolling container that can host multiple components and views.

StyleSheet

Provides an abstraction layer similar to CSS stylesheets.



iOS



Android

Styling Components

Another way to style: styled-components

```
import React from 'react';
import {Text, View} from 'react-native';
import styled from 'styled-components/native';

const App = () => (
  <Container>
    <Text>React Native</Text>
  </Container>
);

const Container = styled.View`  

  flex: 1;  

  background-color: white;  

  align-items: center;  

  justify-content: center;  

`;
```

Nuget Packages

```
✓ Restoring NuGet packages
✓ Auto-linking...
Success: Auto-linking changes completed. (1520ms)
✓ Found Solution: C:\git\wirthwhiledev\rnw-demo\windows\rnwDemo.sln
i Build configuration: Debug
i Build platform: x64
✗ Building Solution: BVLinearGradient.vcxproj -> C:\git\wirthwhiledev\rnw-demo\windows\x64\Debug\BVLinearGradien...
✗ Build failed with message 6:10>C:\git\wirthwhiledev\rnw-demo\node_modules\react-native-webview\windows\ReactNativeWebView\
ReactNativeWebView.vcxproj(194,5): error : This project references NuGet package(s) that are missing on this computer. Use Nu
Get Package Restore to download them. For more information, see http://go.microsoft.com/fwlink/?LinkId=322105. The missing f
ile is C:\git\wirthwhiledev\rnw-demo\windows\packages\Microsoft.UI.Xaml.2.7.0\build\native\Microsoft.UI.Xaml.targets.. Check
your build configuration.
Command failed. Re-run the command with --logging for more information.
PS C:\git\wirthwhiledev\rnw-demo> []
```

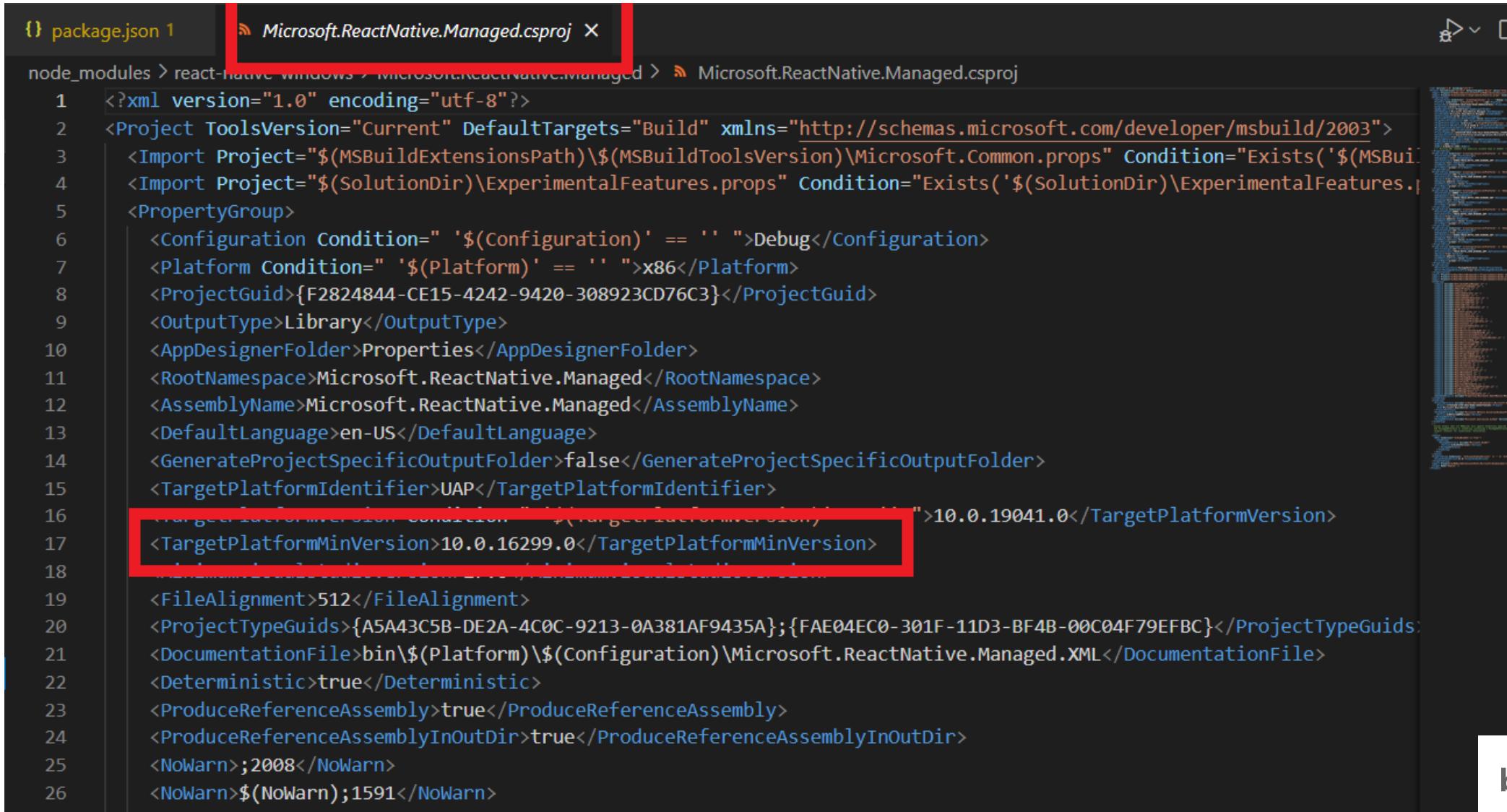
Very Sad Packages

```
53  55      return (
54  -     <NativeLinearGradient
55  -       ref={(component) => { this.gradientRef = component; }}
56  -       {...otherProps}
57  -       startPoint={convertPoint('start', start)}
58  -       endPoint={convertPoint('end', end)}
59  -       colors={colors.map(processColor)}
60  -       locations={locations ? locations.slice(0, colors.length) : null}
61  -       useAngle={useAngle}
62  -       angleCenter={convertPoint('angleCenter', angleCenter)}
63  -       angle={angle}
64  -     />
56 +     <View ref={(component) => { this.gradientRef = component; }} {...otherProps} style={style}>
57 +       <NativeLinearGradient
58 +         style={{ position: 'absolute', top: 0, left: 0, bottom: 0, right: 0 }}
59 +         colors={colors.map(processColor)}
60 +         startPoint={convertPoint('start', start)}
61 +         endPoint={convertPoint('end', end)}
62 +         locations={locations ? locations.slice(0, colors.length) : null}
63 +         useAngle={useAngle}
64 +         angleCenter={convertPoint('angleCenter', angleCenter)}
65 +         angle={angle}
66 +       />
67 +       {children}
68 +     </View>
```

Frankenstein's Monster

```
✓ Found Solution: C:\git\wirthwhiledev\that-conference-demo - Copy\windows\thatConferenceDemo.sln
i Build configuration: Debug
i Build platform: x64
✗ Building Solution: C:\Users\aron\.nuget\packages\microsoft.ui.xaml\2.8.0\build\Common.targets(10,5): error : ...
✗ Build failed with message 5:7>C:\Users\aron\.nuget\packages\microsoft.ui.xaml\2.8.0\buildTransitive\Common.targets(10,5): error : Microsoft.UI.Xaml nuget package requires TargetPlatformMinVersion >= 10.0.17763.0 (current project is 16299) [C:\git\wirthwhiledev\that-conference-demo - Copy\node_modules\react-native-windows\Microsoft.ReactNative.Managed\Microsoft.ReactNative.Managed.csproj]. Check your build configuration.
  command failed. Re-run the command with --logging for more information.
S C:\git\wirthwhiledev\that-conference-demo - Copy>
```

Frankenstein's Monster...it's alive



The screenshot shows a code editor with two tabs: 'package.json' and 'Microsoft.ReactNative.Managed.csproj'. The 'Microsoft.ReactNative.Managed.csproj' tab is active and highlighted with a red box. The code in the editor is an XML build configuration file. A second red box highlights a specific section of the XML code where target platform versions are defined.

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="Current" DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props" Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props')"/>
  <Import Project="$(SolutionDir)\ExperimentalFeatures.props" Condition="Exists('$(SolutionDir)\ExperimentalFeatures.props')"/>
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">x86</Platform>
    <ProjectGuid>{F2824844-CE15-4242-9420-308923CD76C3}</ProjectGuid>
    <OutputType>Library</OutputType>
    <AppDesignerFolder>Properties</AppDesignerFolder>
    <RootNamespace>Microsoft.ReactNative.Managed</RootNamespace>
    <AssemblyName>Microsoft.ReactNative.Managed</AssemblyName>
    <DefaultLanguage>en-US</DefaultLanguage>
    <GenerateProjectSpecificOutputFolder>false</GenerateProjectSpecificOutputFolder>
    <TargetPlatformIdentifier>UAP</TargetPlatformIdentifier>
    <TargetPlatformVersion>10.0.19041.0</TargetPlatformVersion>
    <TargetPlatformMinVersion>10.0.16299.0</TargetPlatformMinVersion>
    <FileAlignment>512</FileAlignment>
    <ProjectTypeGuids>{A5A43C5B-DE2A-4C0C-9213-0A381AF9435A};{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}</ProjectTypeGuids>
    <DocumentationFile>bin\$(Platform)\$(Configuration)\Microsoft.ReactNative.Managed.XML</DocumentationFile>
    <Deterministic>true</Deterministic>
    <ProduceReferenceAssembly>true</ProduceReferenceAssembly>
    <ProduceReferenceAssemblyInOutDir>true</ProduceReferenceAssemblyInOutDir>
    <NoWarn>;2008</NoWarn>
    <NoWarn>$(NoWarn);1591</NoWarn>
  </PropertyGroup>
```



Takeaways

- ✓ Easy to get started
- ✓ Dev experience
 - Hot reloading. Chef's kiss
- ✓ Native performance
- ✓ Large community
- ✓ Actively Maintained
 - RNW team is good at keeping it up to date with latest RN version
- ✓ Staying Power
- ✗ It's still very new
 - i.e. buggy
- ✗ Compile and build times are long
- ✗ Microsoft hasn't proven very trustworthy as of late
 - We've been burned before
- ✗ 3rd party libraries are messy for windows
 - The biggest pain point by far



Questions?

Session Feedback



<https://that.land/43ij8lj>



Find me



LinkedIn

<https://www.linkedin.com/in/aaronwirth/>



GitHub

<https://github.com/wirthwhiledev>



Survey Code

<https://that.land/43iJ8lj>