

Il futuro delle applicazioni Windows con .NET 5

Matteo Pagani

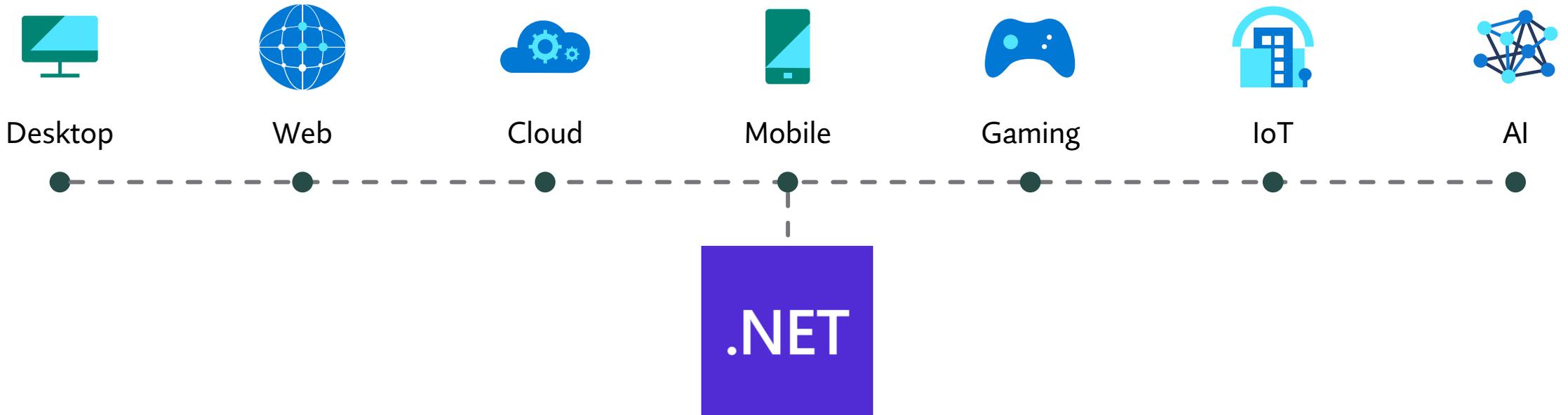
Windows AppConsult Engineer @ Microsoft
matteo.pagani@microsoft.com | @qmatteoq

.NET Conference
Italia 2020

.NET

.NET

Your platform for building anything

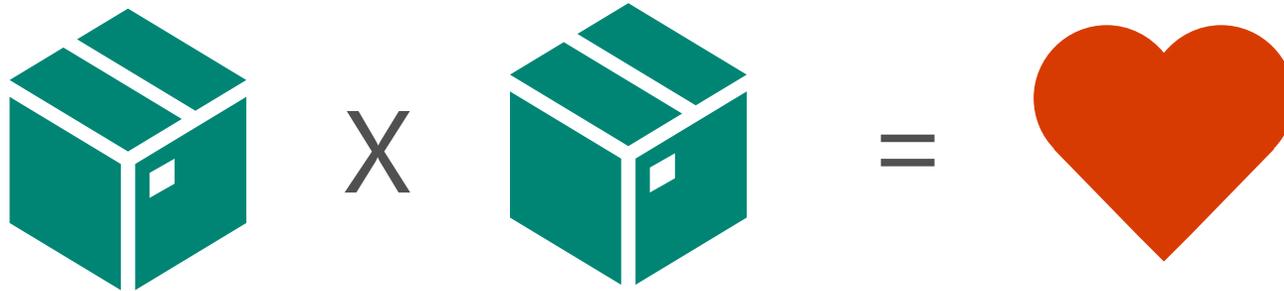


The future of .NET Framework

- .NET Framework 4.8 is the last major version of .NET Framework on Windows
- Support policy remains the same:
 - Will always be in Windows
 - Will be patched with Windows
 - Will be supported with Windows
- Keep existing mature applications on .NET Framework
- Recommend .NET Core for new applications or existing applications which will continue to be evolved

Why Windows desktop on .NET 5.0?

- Deployment flexibility
 - Side-by-side support
 - Machine global or app local framework
 - Self-contained EXEs
- Core runtime and API improvements
- Performance



Target framework for Windows apps

```
<Project Sdk="Microsoft.NET.Sdk">
```

```
<PropertyGroup>
```

```
<OutputType>WinExe</OutputType>
```

```
<TargetFramework>net5.0-windows</TargetFramework>
```

```
<UseWPF>>true</UseWPF>
```

```
</PropertyGroup>
```

```
</Project>
```

- Support to Windows 7, 8 and 10
- Access to specific Windows APIs like:
 - Registry
 - Event Viewer
 - WCF services
 - Windows Services

.NET Framework and .NET Core 3.x

- The WinRT projection are included in the compiler
- Package available on NuGet to reference the correct metadata
- Get access to all the UWP APIs from a .NET application



Microsoft.Windows.SDK.Contracts

10.0.18362.2005 

The Windows 10 WinRT API Pack enables you to add the latest Windows Runtime APIs support to your .NET Framework 4.5+ and .NET Core 3.0+ libraries and apps.

This package includes all the supported Windows Runtime APIs up to Windows 10 version 1903. If you are targeting earlier platforms, consider only offering functionality available on the detected platform version. For further details, see:

<https://docs.microsoft.com/en-us/windows/uwp/debug-test-perf/version-adaptive-code>

Supported platforms:

- .NET Framework 4.5+
- .NET Core 3.0+

<https://www.nuget.org/packages/Microsoft.Windows.SDK.Contracts>

C# / WinRT

- NuGet-packaged toolkit that provides Windows Runtime (WinRT) projection support for the C# language
- It enables .NET 5.0 application to access to Windows Runtime APIs
 - Geolocation
 - Bluetooth
 - Biometric authentication
 - Notifications
 - And much more...
- It enables to consume Windows Runtime components in a .NET 5.0 application

New approach in .NET 5

- It leverages the C# / WinRT project
- The goal is to make .NET completely agnostic, by lifting the WinRT projection support out of the compiler
- No more NuGet packages to reference
- Just target the specific TFM

```
<Project Sdk="Microsoft.NET.Sdk">
```

```
  <PropertyGroup>
```

```
    <OutputType>WinExe</OutputType>
```

```
    <TargetFramework>net5.0-windows10.0.19041.0</TargetFramework>
```

```
    <UseWPF>>true</UseWPF>
```

```
  </PropertyGroup>
```

```
</Project>
```

Supported versions

- 10.0.17763.0
- 10.0.18362.0
- 10.0.19041.0

Using Windows Runtime APIs

Demo

Consuming Windows Runtime components

- You can use C# / WinRT to create a projection library for a Windows Runtime Component
- You can pack this library into a NuGet component or just distribute the DLL
- Use the library from a .NET 5.0 application

```
<PropertyGroup>  
  <CsWinRTIncludes>GeolocationComponent</CsWinRTIncludes>  
  <CsWinRTGeneratedFilesDir>$(OutDir)</CsWinRTGeneratedFilesDir>  
</PropertyGroup>
```

Consuming a Windows Runtime component

Demo

Making apps *great* for the people who use them



Support new hardware



Modern user experience



App deployment and management



Reliability, security, and privacy



System performance and battery life

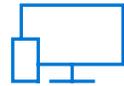
Starting where you are

 Compatibility with all your code – WinForms, WPF, MFC...

 Support for your existing packaging and deployment

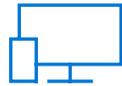
 Features that work across users' versions of Windows 10

101010
010101
101010

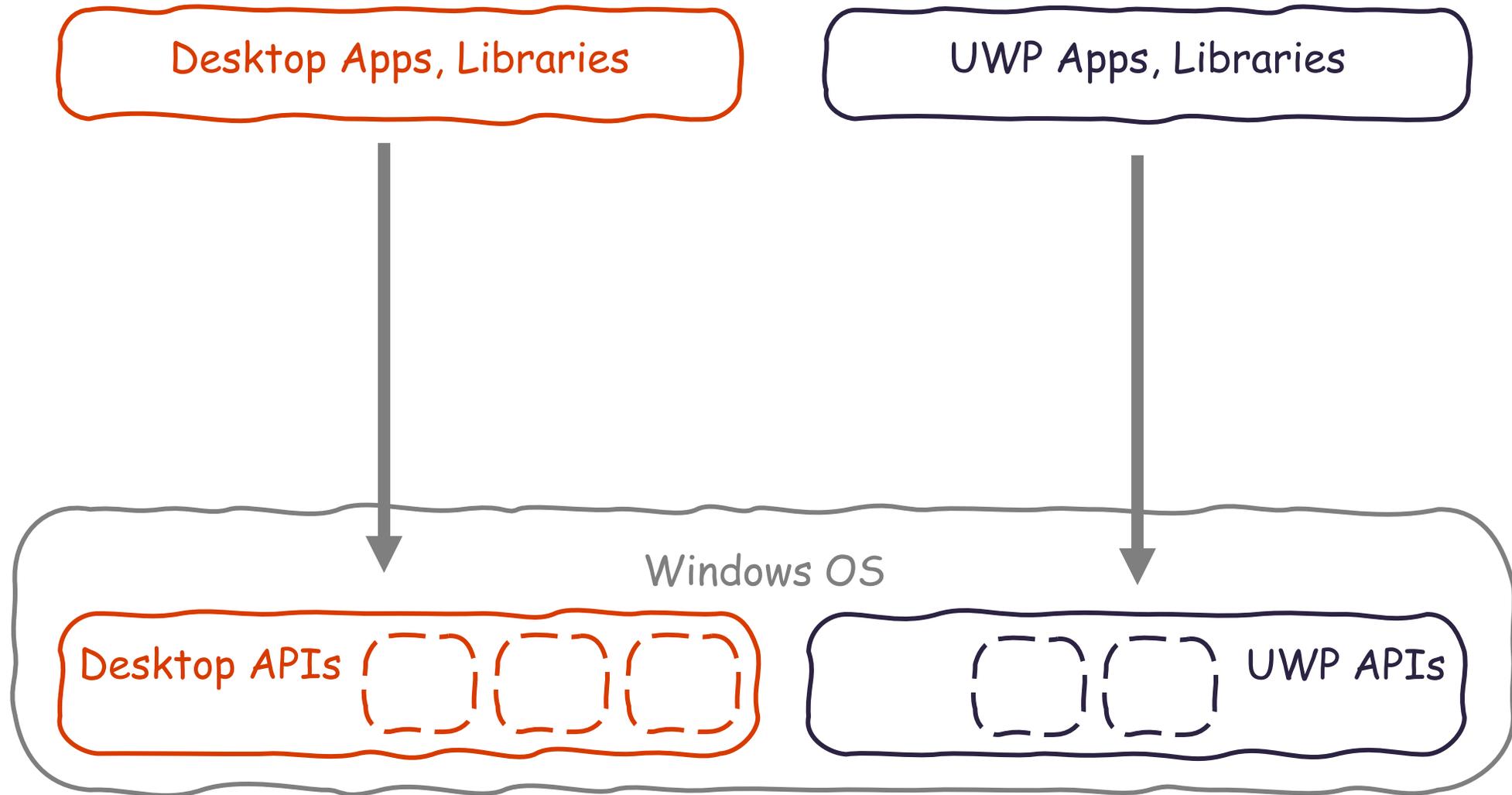


Project Reunion

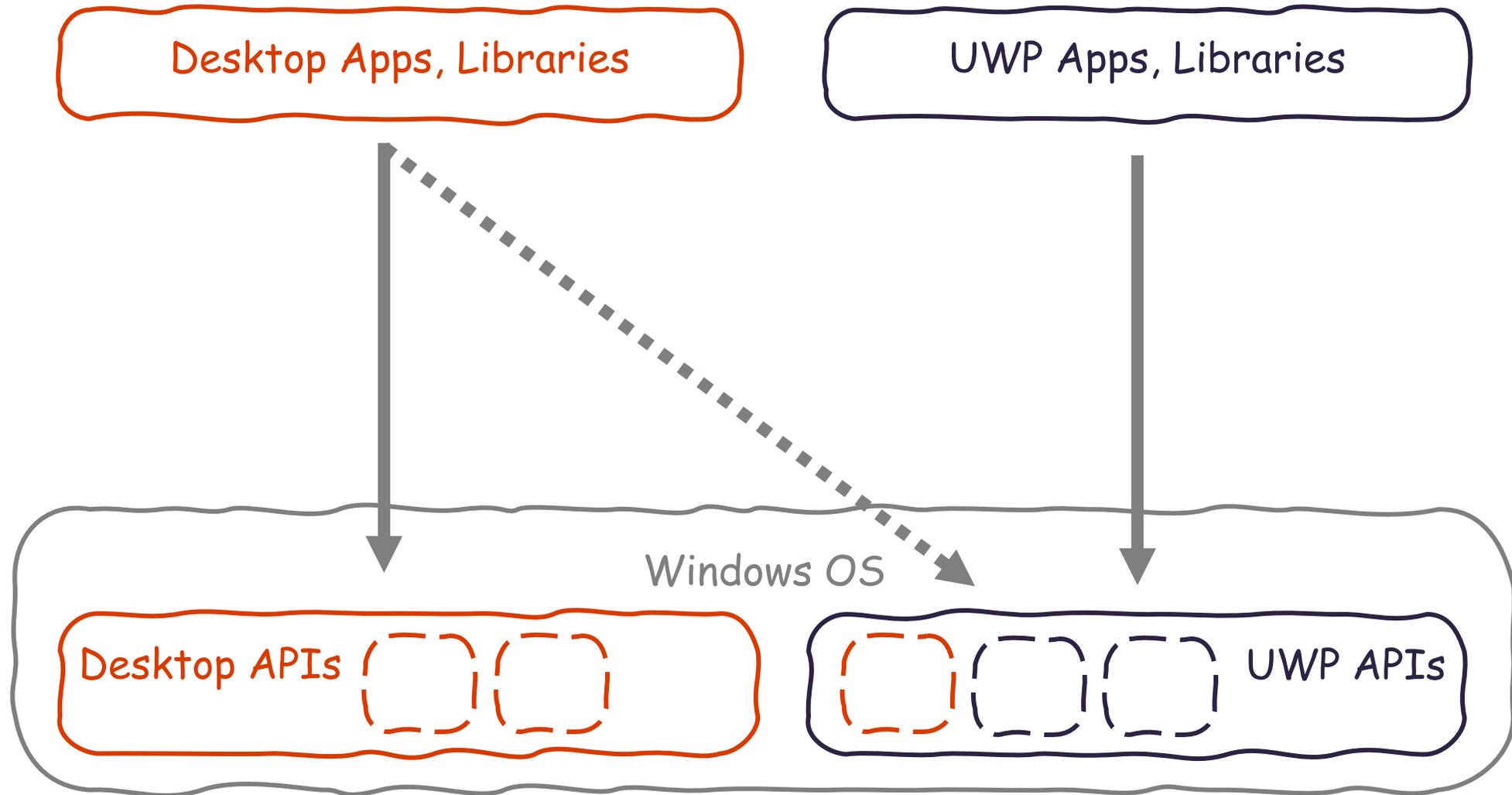
101010
010101
101010



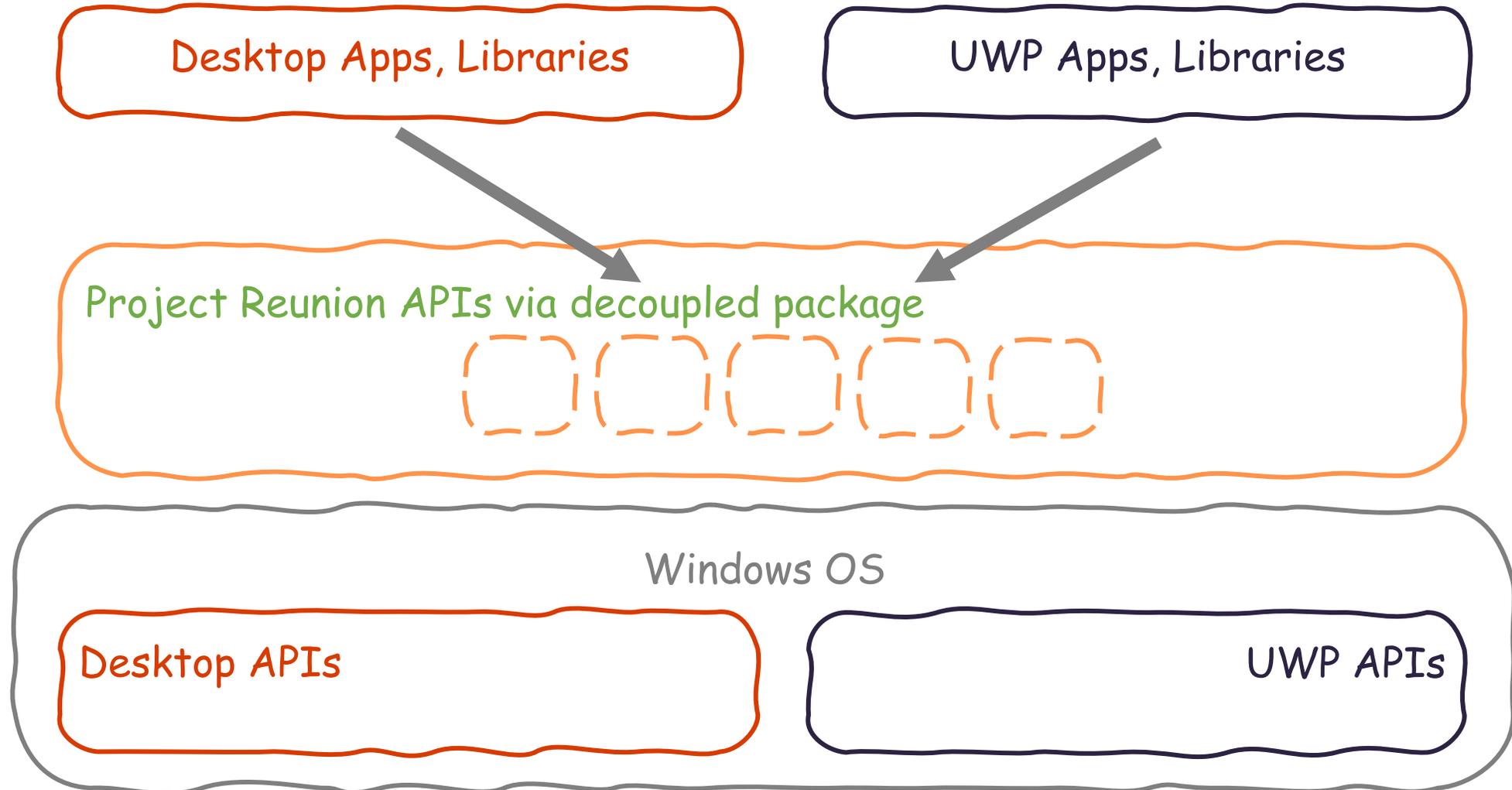
Today



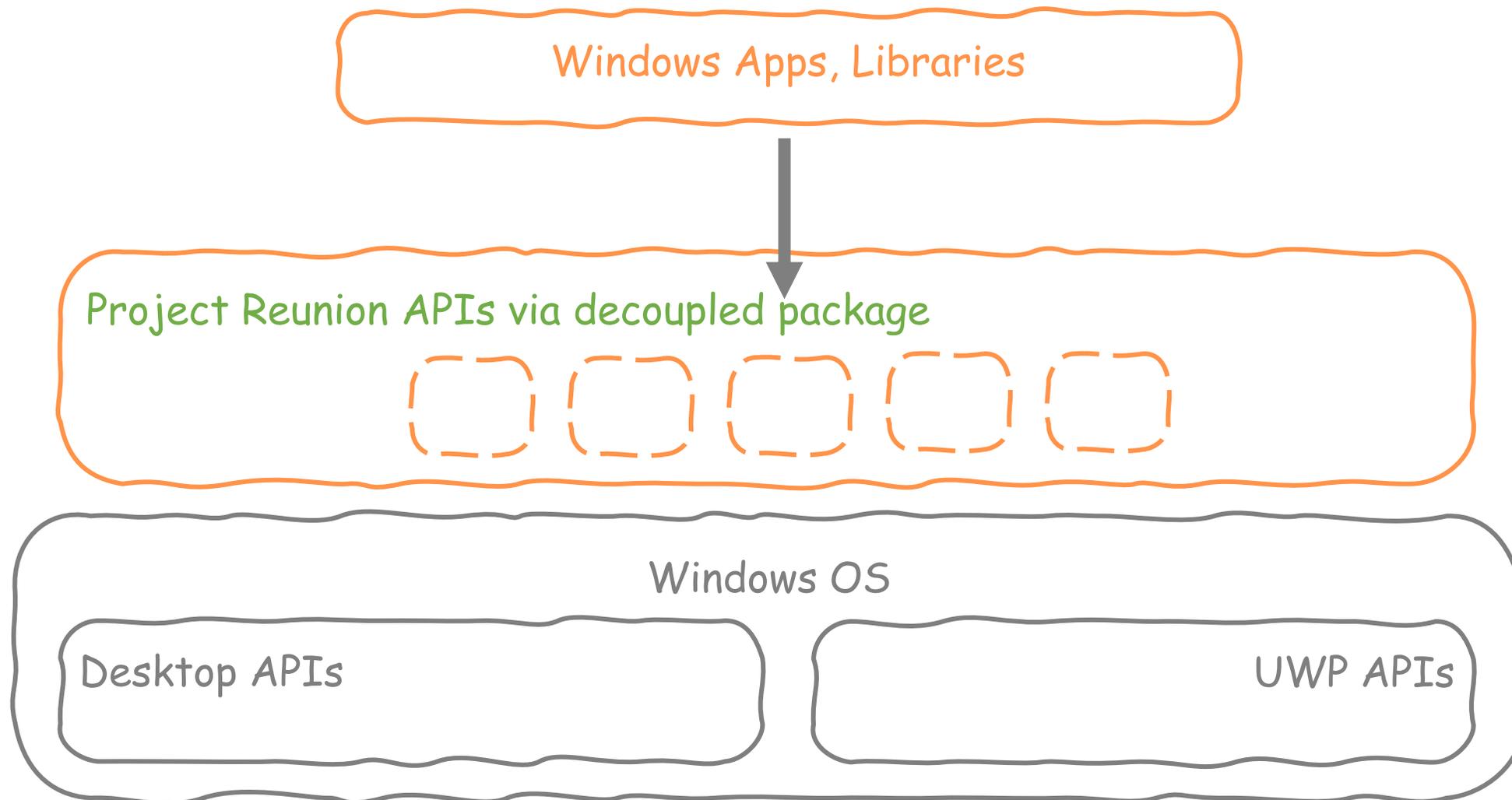
Today



Project Reunion



Project Reunion



Agility through backwards compatibility

New APIs and platform improvements can become instantly available, without waiting for user os updates to catch up.

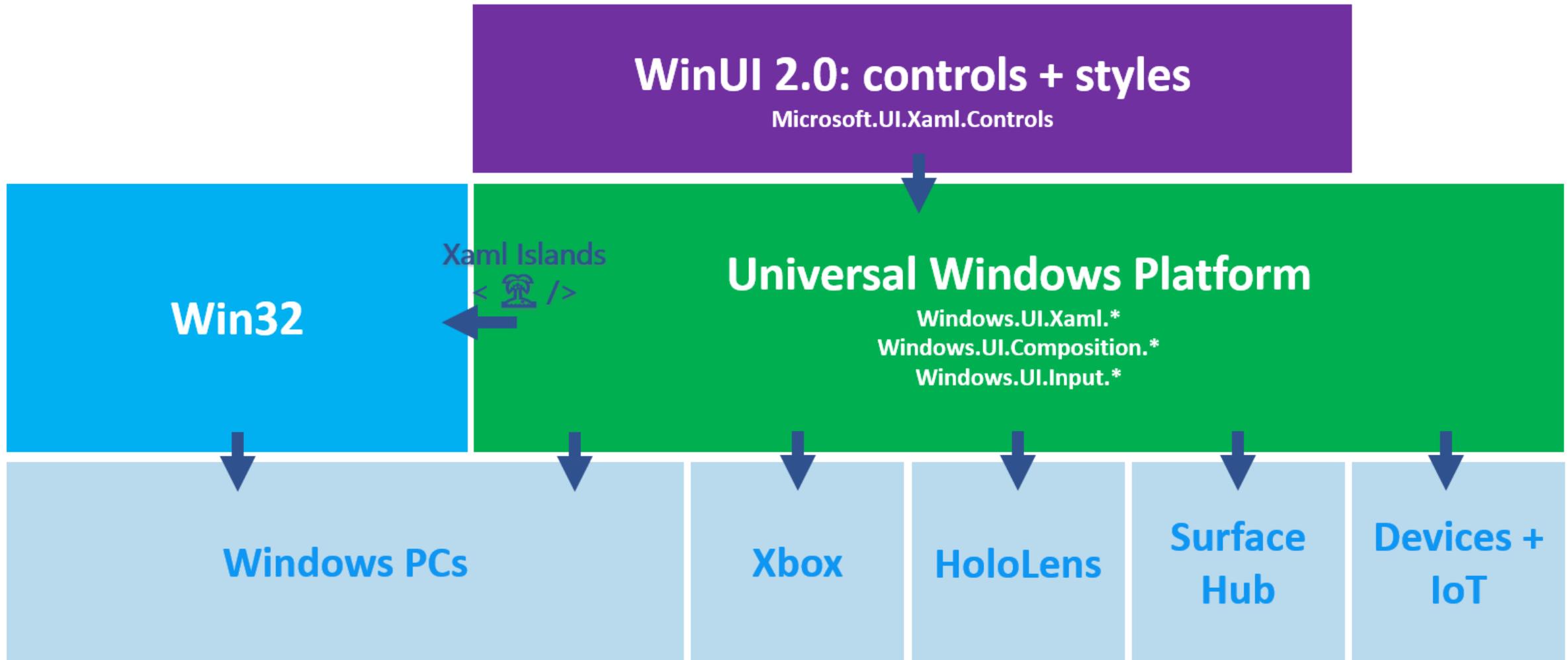
Modern and accessible UI and input

WinUI

- Native UI platform for Windows 10
- It moves controls and resources from the OS to a NuGet package
- It makes easier to leverage UI controls independently from the OS version
- More agility in fixing bugs and adding new features
- Open source on GitHub: <https://github.com/microsoft/microsoft-ui-xaml>



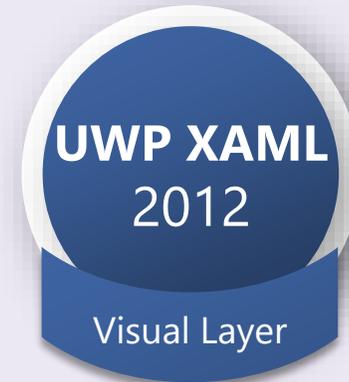
WinUI 2.x



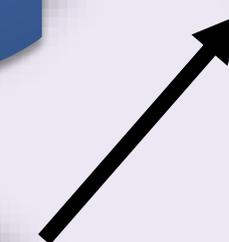
Desktop (Win32) Apps



UWP Apps



+



Desktop (Win32) Apps

UWP Apps



WinUI 3 will provide a state-of-the-art UX framework for **every** Windows developer.



Always contains latest Fluent controls & styles



Native UI performance in any app (C++, or .NET)

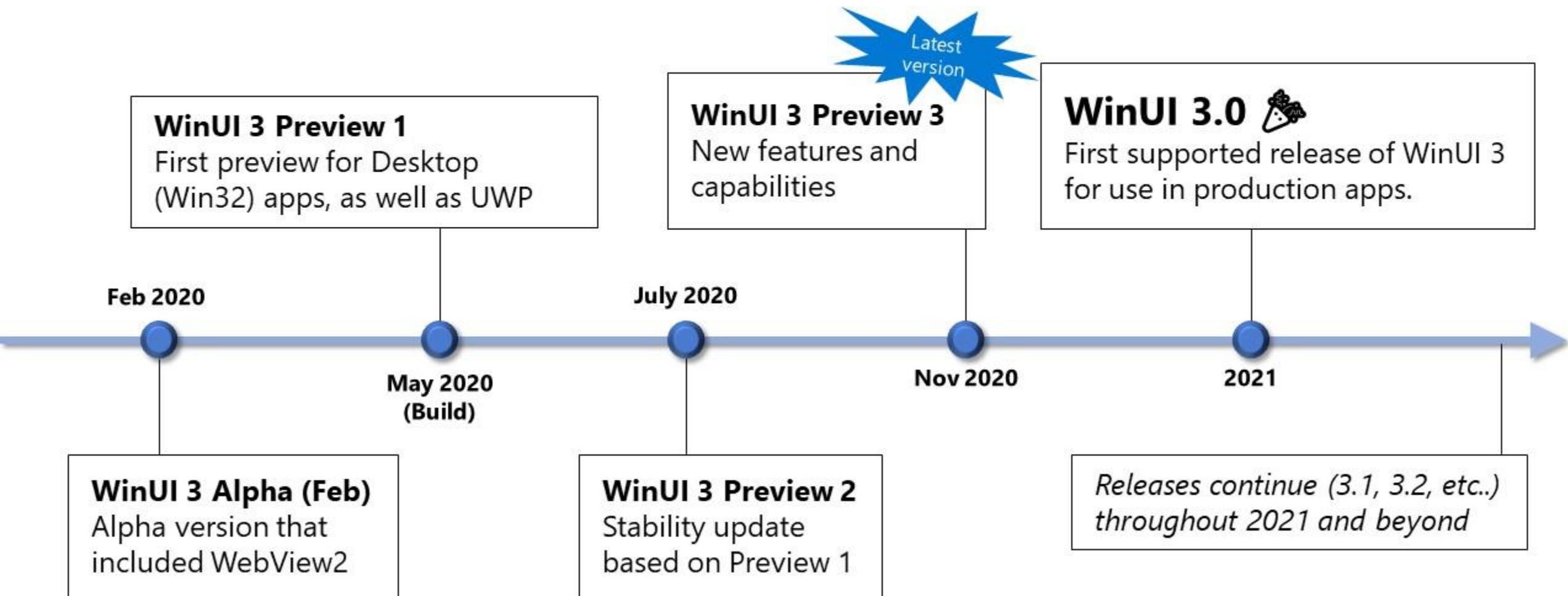


You choose when to upgrade!

Demo

WinUI 3.0 Preview 3

WinUI 3 Release Roadmap



ClickOnce is back!

- But I wouldn't use it 😊
- MSIX is the way forward to deploy Windows applications:
 - Clean install and uninstall
 - Automatic updates
 - Differential updates and storage optimizations
 - Easy integration with CI/CD pipelines
- Add a Windows Application Packaging Project to your solution to package your .NET 5.0 app as MSIX

<https://aka.ms/msix>

@qmatteoq
matteo.pagani@microsoft.com

Slide e materiale su
<https://aspit.co/netconfit-20>

<https://github.com/qmatteoq/NetConf2020-AspItalia>

.NET Conference
Italia 2020

.NET