



.NET桌面应用开发技术概述

北京理工大学计算机学院
金旭亮

什么是“桌面应用”？



桌面应用：以PC和平板为主要运行环境，拥有可视化的界面，通过调用PC操作系统（Windows、Linux和Mac OS）API实现各种功能的应用程序。

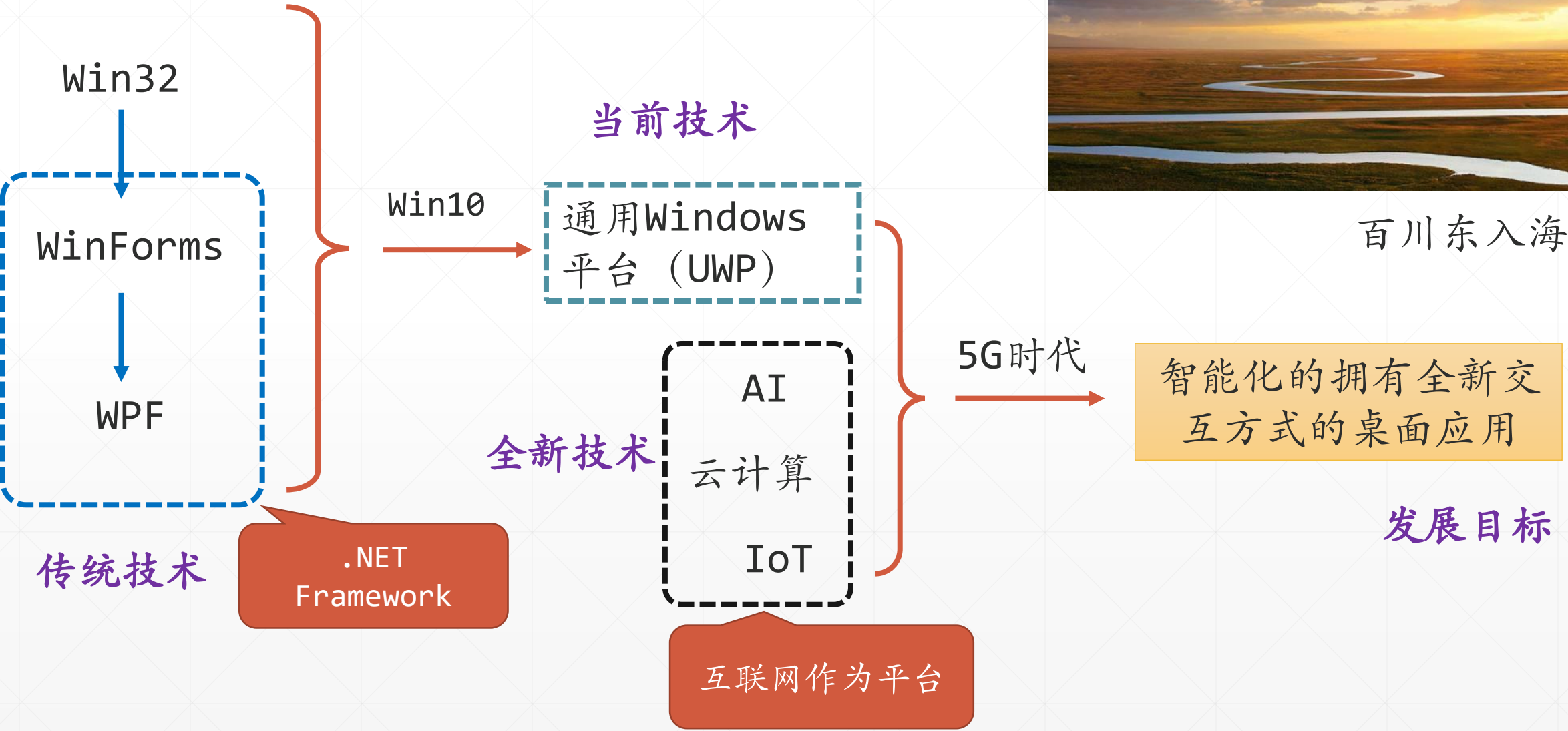


从广义上来说，“黑底白字”的控制台应用程序也属于“桌面应用”，不过由于它的UI交互比较简单，所以本课程将其“排除”在“桌面应用”之外。



现有的Windows桌面应用开发技术

Windows桌面开发技术发展路线图



百川东入海



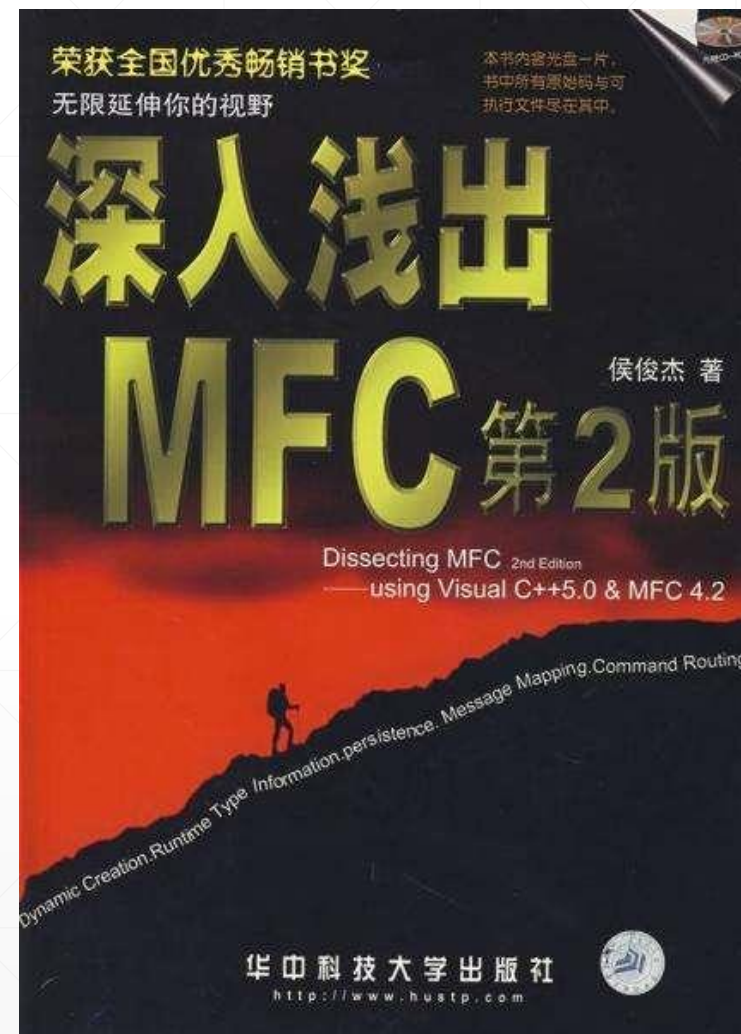
香港四大天王，纵横影视歌三十
余载.....



Windows 桌面应用“四大天王”，长
江后浪推前浪，江湖风云三十年.....

四大天王之“Win32”

- ➔ 主要使用C/C++开发，直接生成本机代码，运行于Windows操作系统之上。
- ➔ 能直接调用Windows操作系统提供的Win32 API，无所不能。
- ➔ 需要了解诸多底层技术细节，可以进行深度性能优化，开发效率与代码质量受开发者自身技术水平影响极大。
- ➔ 学习曲线陡峭，难度大。



四大天王之“Windows Forms”



汲取了VB和Delphi的经验，是.NET Framework早期最主要的桌面应用开发框架，与Windows兼容性好。



拥有一个所见即所得的可视化设计器，拥有“一堆”标准控件（第三方厂商也开发了不少商业控件），易学易用，开发效率高，程序运行性能让人满意。



所有功能依赖于.NET Framework的基类库，在特定场景可以通过“平台调用（P/Invoke）”调用Win32 API，2019年移植到.NET Core，可以调用Win10 API，可以使用UWP控件。



适合于开发拥有标准Windows界面的应用程序，或者是用于快速构建原型，



四大天王之“WPF”



.NET Framework 3.0引入，将UI设计与功能代码分离，使用XAML设计UI界面。



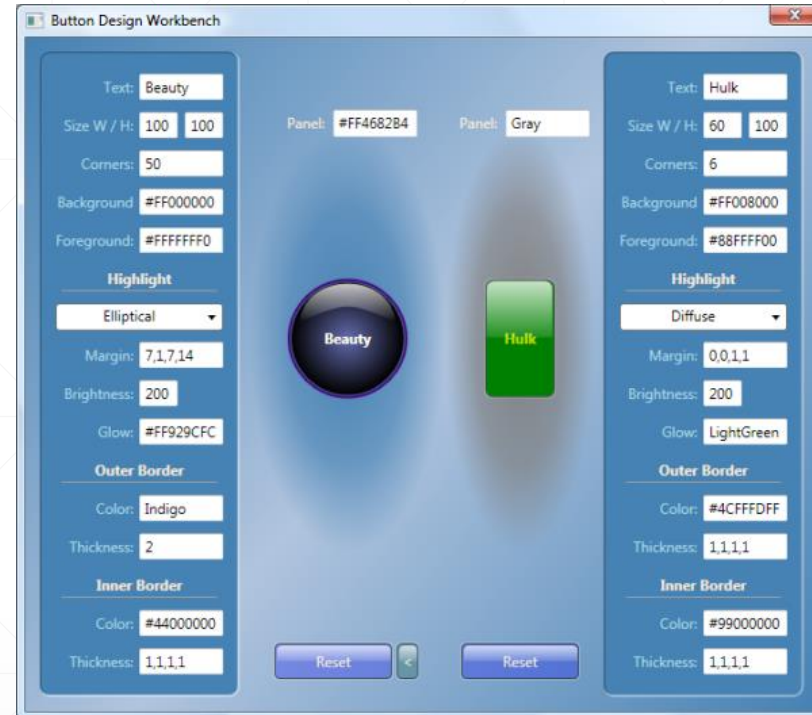
拥有丰富的技术特性和高度的UI界面可定制性，可以开发出界面非常Cool、功能庞大的复杂桌面应用（比如Visual Studio）。



学习曲线比较陡峭，需要比较扎实的.NET面向对象编程技能。



WPF应用与WinForms一样，依赖于.NET Framework基类库，支持Windows 7，并于2019年起一起被移植到.NET Core，同样可以集成UWP控件和调用Win10 API。



四大天王之“UWP”



微软最新的桌面应用开发技术框架，只能运行于Windows 10之上。



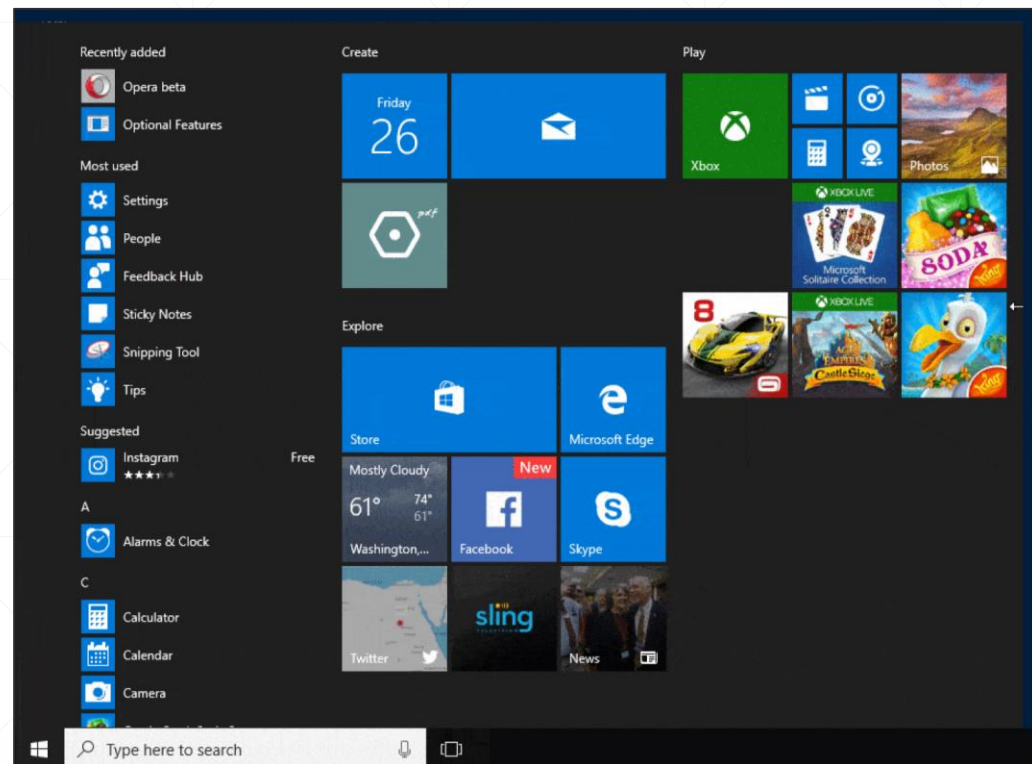
App编译生成原生代码，可以上传到微软应用商店，跑在所有Win10设备上。



支持多种编程语言，可以直接调用Win10 API，不依赖于.NET Framework和.NET Core运行环境。



集成许多现代计算机技术成果，拥有诸多新特性，是Windows平台当前和未来的主要编程框架。



Windows桌面应用的“主力军”与“游击队”



Windows-first

Cross platform



UWP

WPF

Xamarin.Forms

Windows forms

PWA

React Native

Electron

.NET程序员

Web程序员



桌面应用的“现代化”趋势

时代的大背景



→ PC时代早已落幕，以手机为代表的移动互联时代已经进入稳定期，智能时代正在揭幕.....

→ 桌面应用技术领域市场在日益缩小，但绝不会消失，因为PC不会消亡。

→ Windows的整个生态系统正在向Windows 10过渡，这是非常明确的趋势。

→ Windows平台的传统桌面开发技术，正处于一个升级换代的时间节点上。

Windows桌面应用的“现代化”发展历程

新一代智能桌面应用软件

C++和Win32

- 远古时代技术
- 直接调用Win32 API

Windows Forms

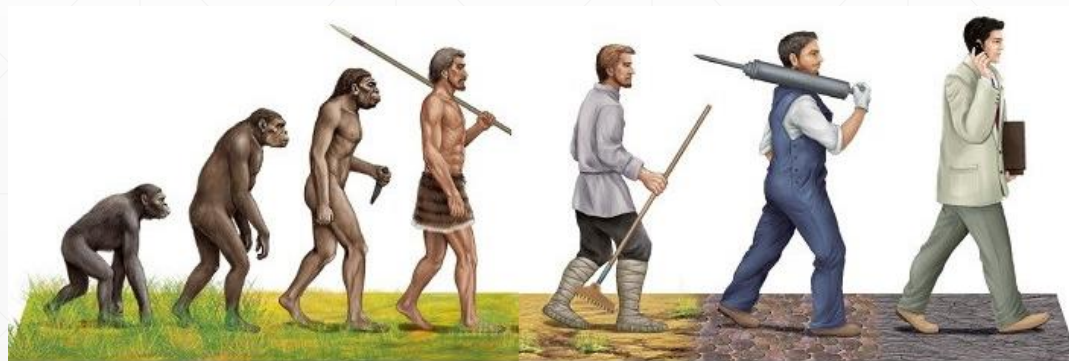
- 从.NET 1.0就有
- 底层封装的是GDI

WPF

- .NET 3.0开始
- 底层封装了 DirectX

UWP

- Win8开始演化
- 底层运行环境是 WinRT

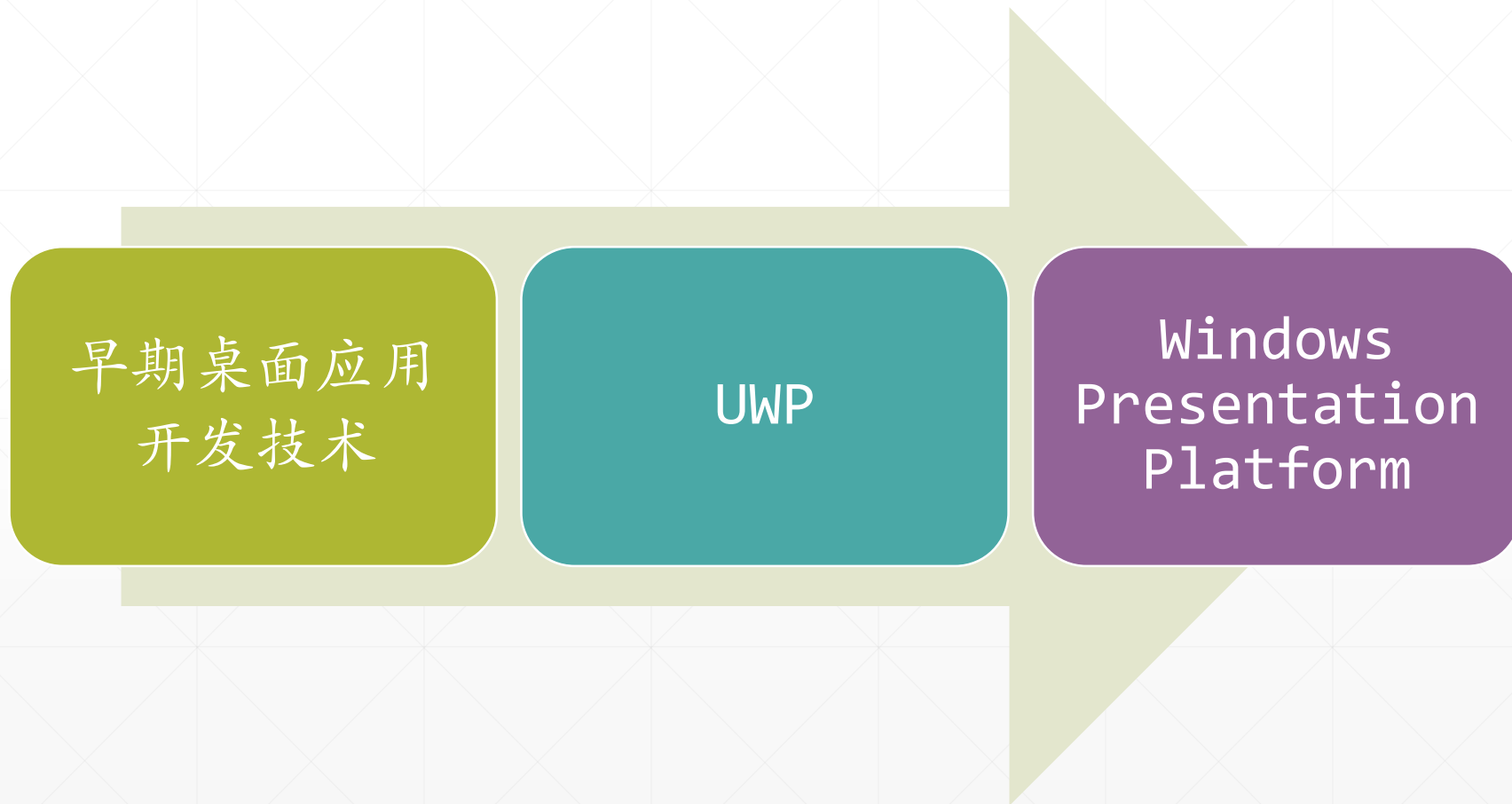


新一代智能桌面应用软件的特点

- 随时在线
- 新的输入手段
- 支持操控新的硬件
- 使用新的服务
- 更加智能化
- 物联网的一个节点
-




近几年Windows桌面开发技术的发展方向



WinUI是Windows Presentation Platform的主要成员

Windows UI Library

 Follow WinUI on Twitter 6.9k

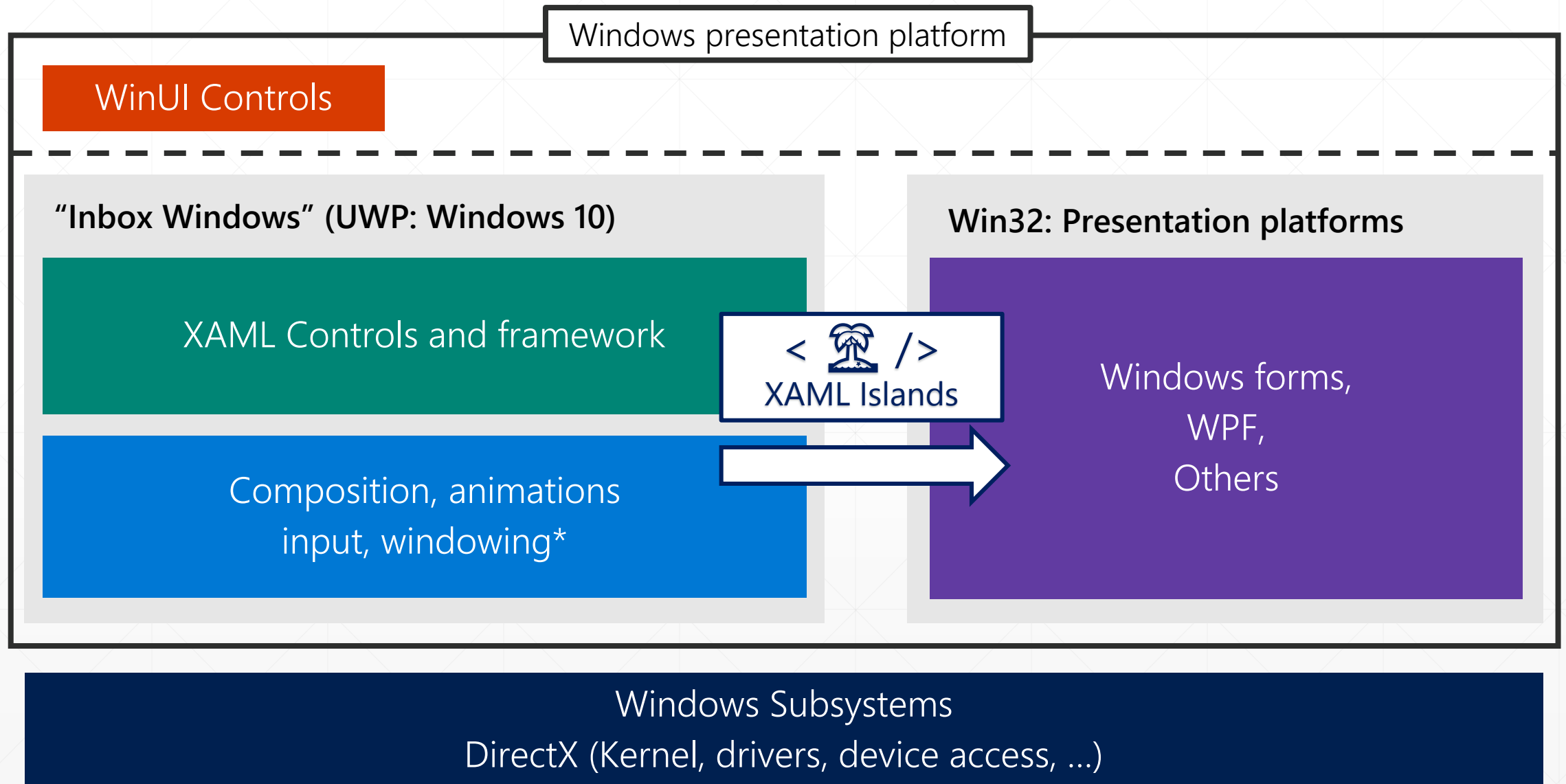
The Windows UI Library (WinUI) provides official native Microsoft UI controls and features for Windows [UWP apps](#).

WinUI is the easiest way to build great [Fluent Design](#) experiences for Windows.

WinUI can be used in any Windows 10 UWP XAML app, or in a Xamarin.Forms app running on Windows 10 using [native view embedding](#).

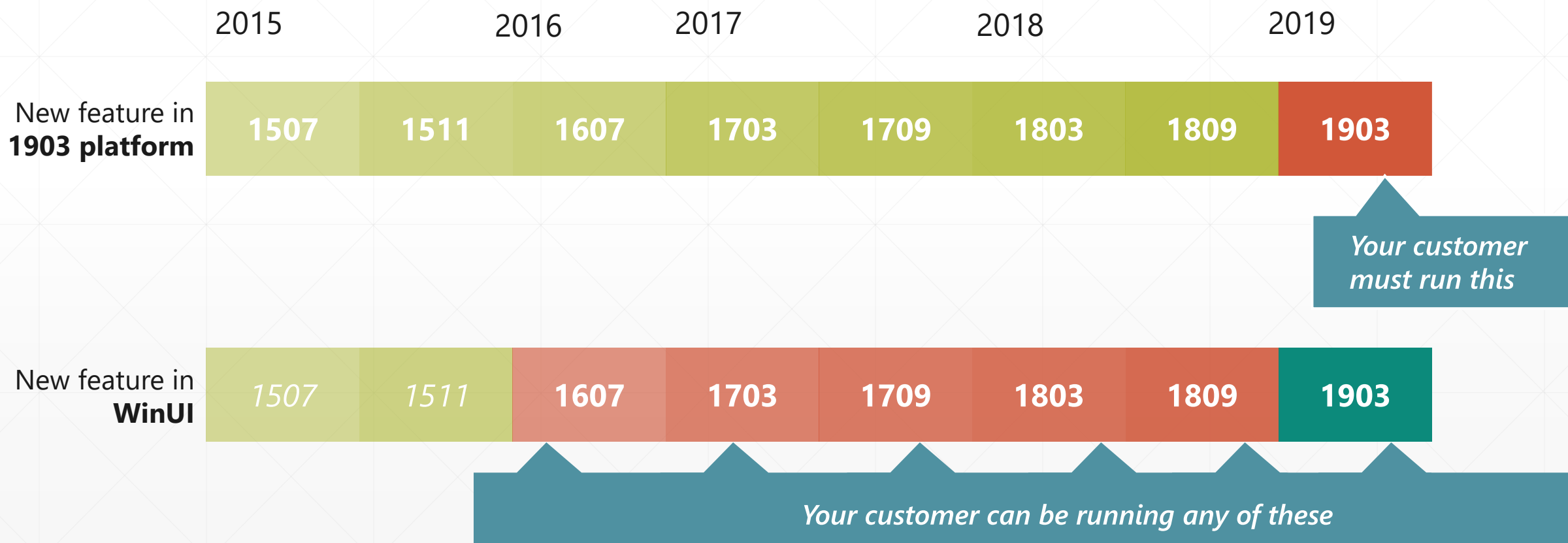
<https://github.com/Microsoft/microsoft-ui-xaml>

WinUI 2.X



WinUI的现状与下一步目标

WinUI计划让早期版本的Windows用户无需升级Windows也能使用最新版才有的UWP控件，类似于Android应用兼容包在Android开发中所起的作用。



WinUI如何实现“新特性向老版本的扩散”?



“WinUI 新版”

XAML 控件和开发框架
(including new WinUI Controls)

Composition, animations
input, windowing

Windows presentation platform

WinUI Controls

Inbox Windows (service)

XAML 控件和开发框架

Composition, animations
input, windowing

Windows Subsystems (早期版本)
DirectX (Kernel, drivers, device access, ...)

Windows Subsystems (最新版本)
DirectX (Kernel, drivers, device access, ...)



桌面应用开发技术选型与迁移指南

当前桌面应用技术选型指南

如果现在（2019+）要开发一个Windows桌面应用：



简单快捷的选Windows Forms



要功能强大的选WPF



专门针对Win10开发应用，选UWP

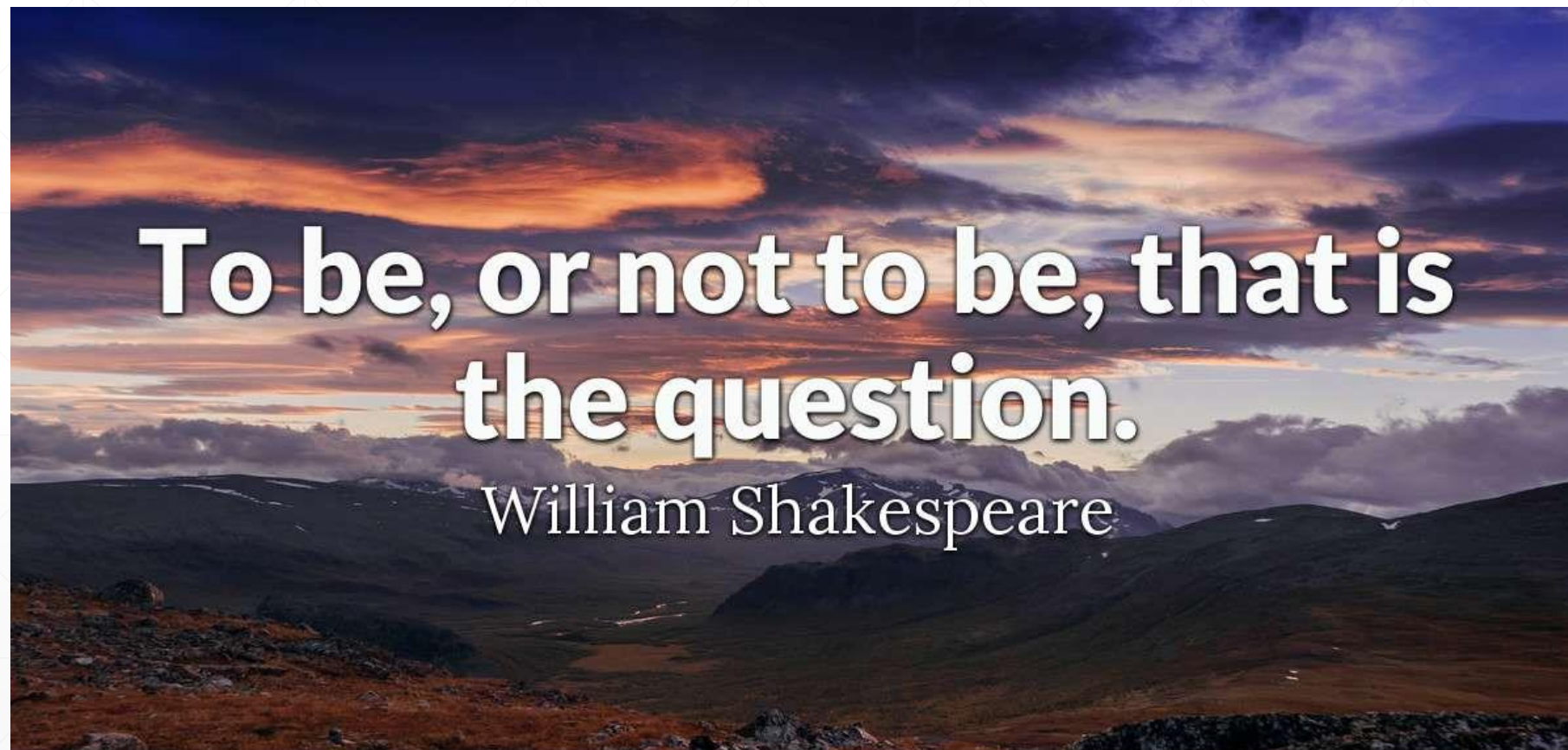


基于.NET Core

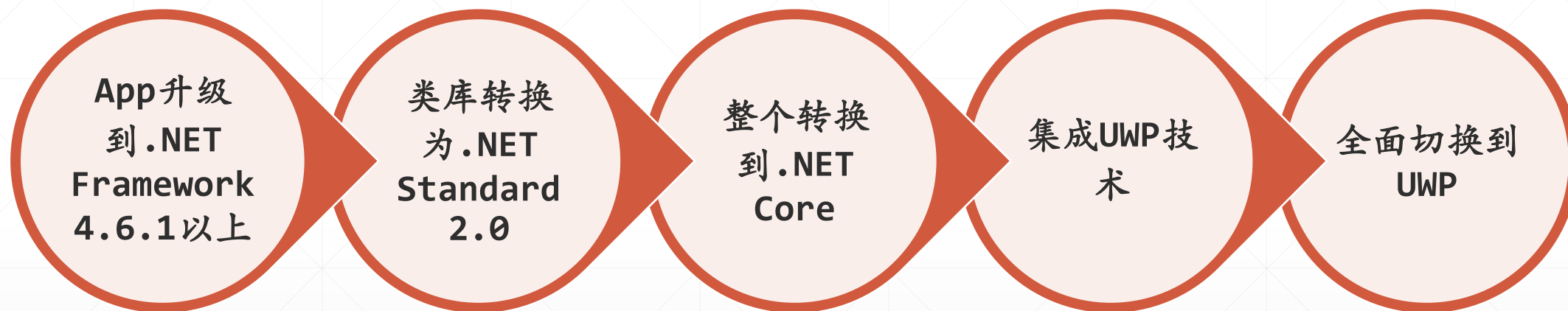
小结：桌面应用技术选型指南

- ➔ 原有程序，只是维护，不需要扩充新功能的，可以不动。
- ➔ 原有程序，需要长期维护并且需要扩充新功能的，应该迁移到 .NET Core。
- ➔ 新写程序，只在PC上用的，推荐使用基于 .NET core 的 Windows Forms 或 WPF，兼容性最好。
- ➔ 只在 Windows 10 上运行，并且需要支持 PC、平板、Xbox 等多种设备，支持触摸屏、语音识别等新的人机交互手段，使用 UWP。

传统WinForm/WPF桌面应用迁移到.NET Core?



早期版本WinForms/WPF应用迁移推荐路线



本模块内容安排

课程安排

(1) 快速把握Windows Forms
编程模型



(2) WPF 编程模型与实例展示



(3) UWP 开发技术走马观花



(4) 传统桌面应用的“现代化”
历程