Build For Multiple Devices

Presenter Name

Expense reporting backlog

- ~~Improve architecture, maintainability, and quality~~
  - ~~Adopt a services architecture~~
- ~~Improve accessibility, scalability, and operations~~
  - ~~Move to the cloud~~
- ~~Update the user experience~~
  - ~~Build a more modern-looking user experience~~
- Expand device support
  - Companion apps for Windows Store and Windows Phone

Agenda

Windows 8 and Windows Store apps
Windows Phone apps
Building for both

Microsoft

Windows 8

Windows reimagined

New fast and fluid Start screen

Touch, mouse, keyboard

Everything great about Windows 7 we made even better

Great experience across all hardware

Tablets to laptops to all-in-ones

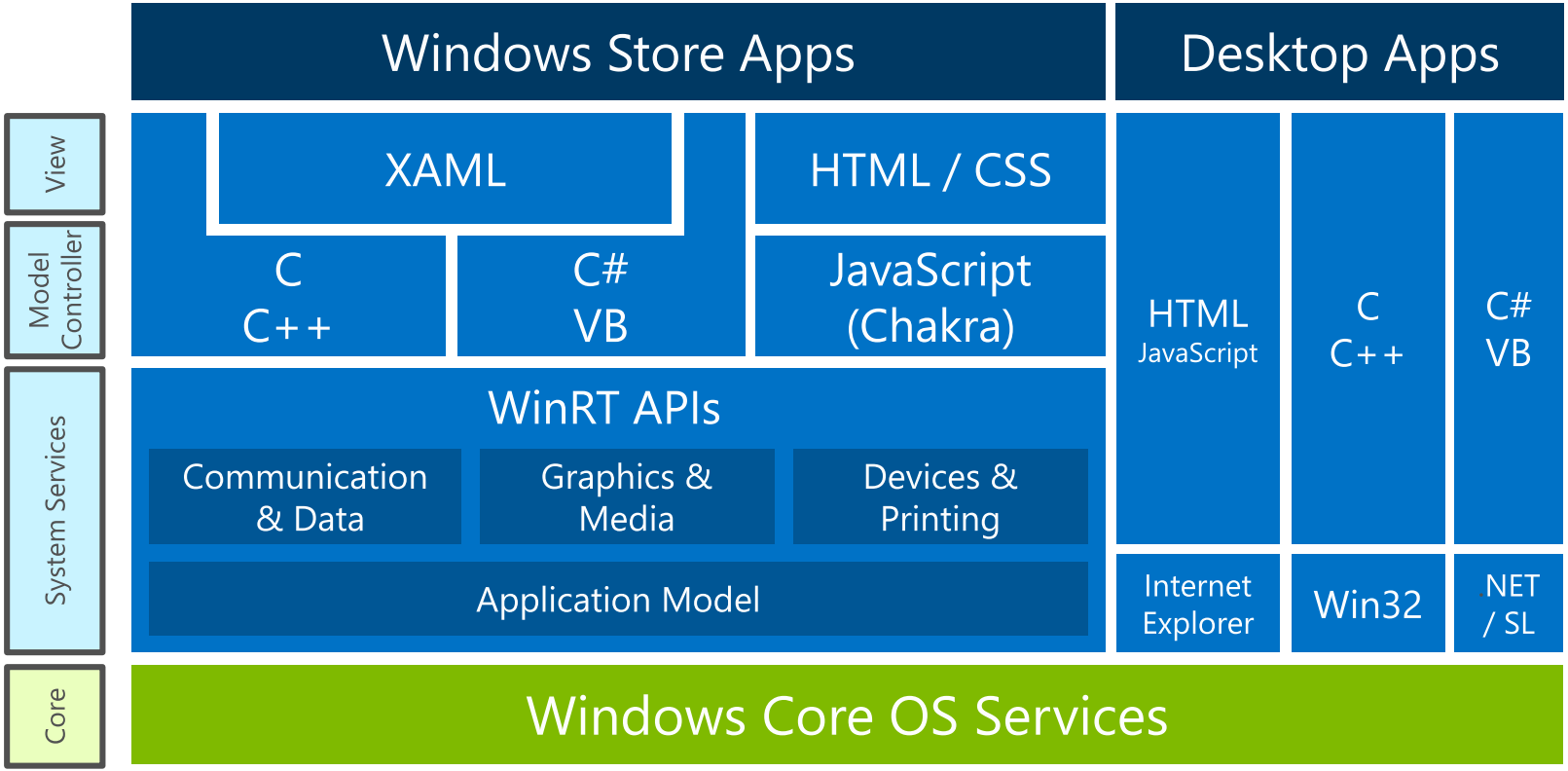
Touch, mouse, keyboard

Highest power to the most efficient chipset

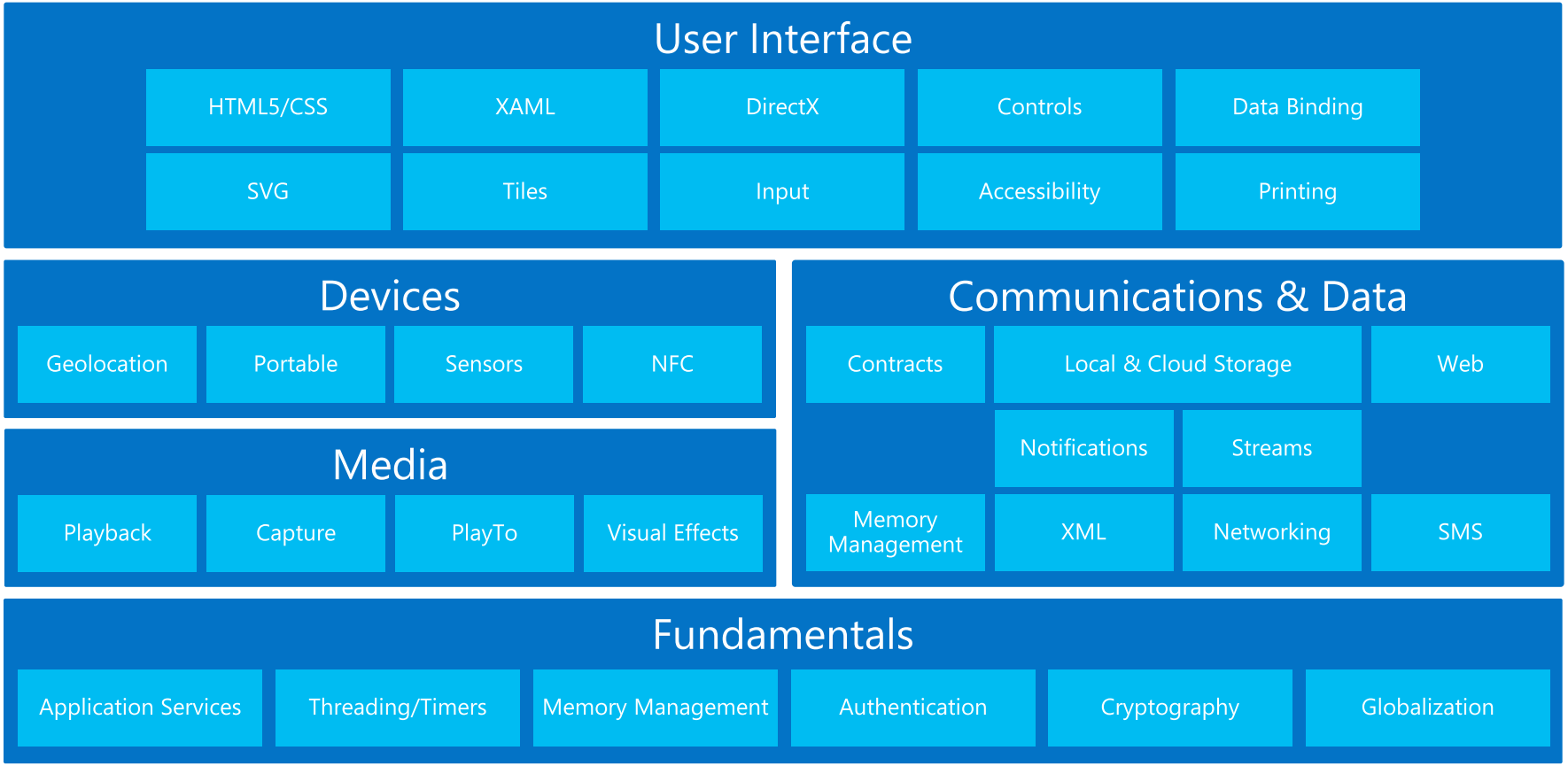
Windows 8 Experience

- Fast and fluid user experience
  - Responsive, alive, beautiful
- Immersive and full-screen
  - Focuses attention on your apps
- Touch-first with full keyboard and mouse
  - Enables your choice of interaction
- Web of apps working together
  - Apps are connected to each other and the cloud
- Experience for all PC devices and architectures
  - No compromise across new form factors, desktops and laptops

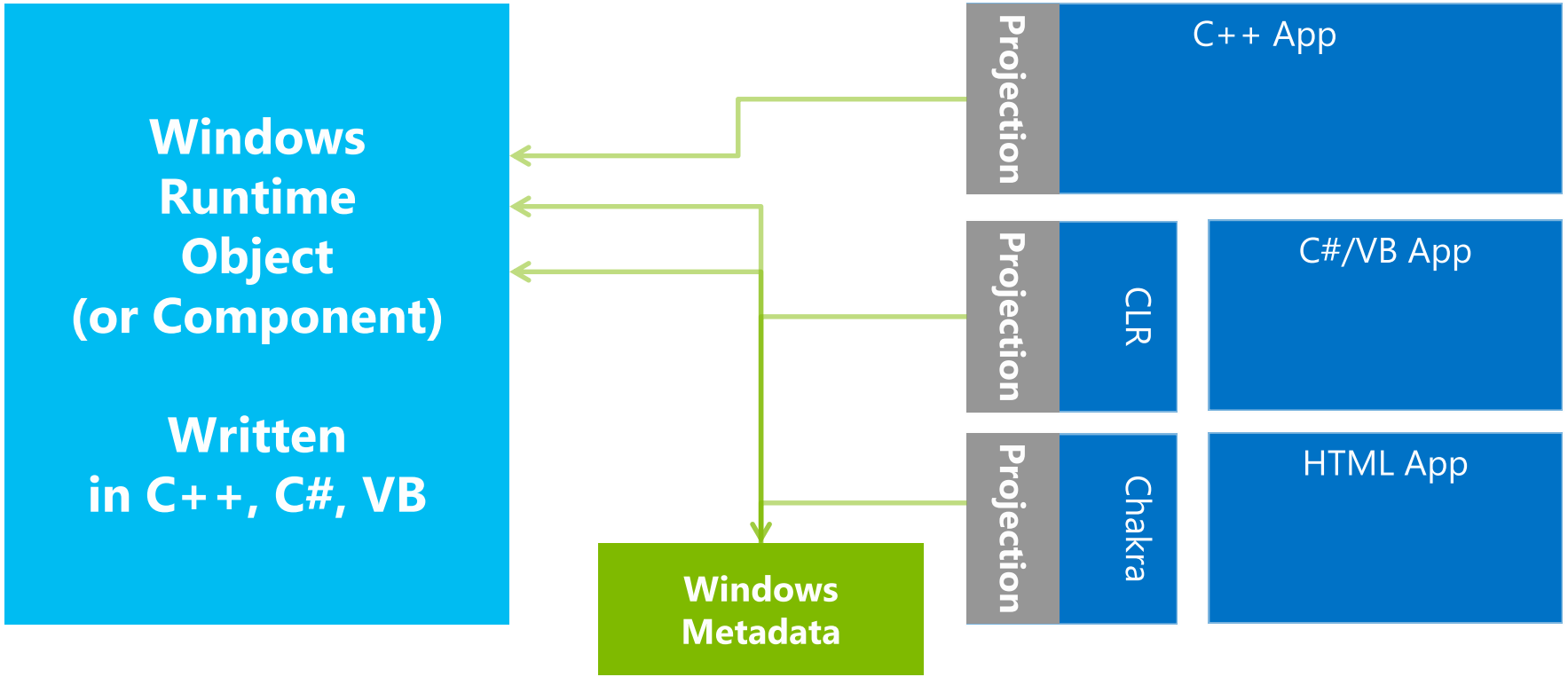
Windows 8 Platform



Windows Runtime APIs



Language projections



The C# code you have to write today…

```
[DllImport("avicap32.dll", EntryPoint="capCreateCaptureWindow")]
static extern int capCreateCaptureWindow(
  string lpszWindowName, int dwStyle,
  int X, int Y, int nWidth, int nHeight,
  int hwndParent, int nID);

[DllImport("avicap32.dll")]
static extern bool capGetDriverDescription(
  int wDriverIndex,
  [MarshalAs(UnmanagedType.LPTStr)] ref string lpszName,
  int cbName,
  [MarshalAs(UnmanagedType.LPTStr)] ref string lpszVer,
  int cbVer);

// more and more of the same
```

The C# code you get to write on Windows 8

```
using Windows.Media.Capture;

var ui = new CameraCaptureUI();
ui.PhotoSettings.CroppedAspectRatio = new Size(4, 3);

var file = await ui.CaptureFileAsync(CameraCaptureUIMode.Photo);

if (file != null)
{
    var bitmap = new BitmapImage() ;
    bitmap.SetSource(await file.OpenAsync(FileAccessMode.Read));
    Photo.Source = bitmap;
}
```

Flexibility of the Windows experience

Maximum reach to your users across platforms

Native apps

Build for new, mobile scenarios

Leverage search, share and new capabilities

Websites

Broad reach across OSes

Desktop apps

Keep software investments already made

Windows Phone

Windows Phone 8 Hardware

- Beautiful new hardware from Nokia, HTC, Samsung and more…

Modern Smartphone Platform

- New multicore chipset
- New graphics processor
- Increased RAM: 1GB or 512MB
- More Screen resolutions
- Removable, encryptable storage
- NFC

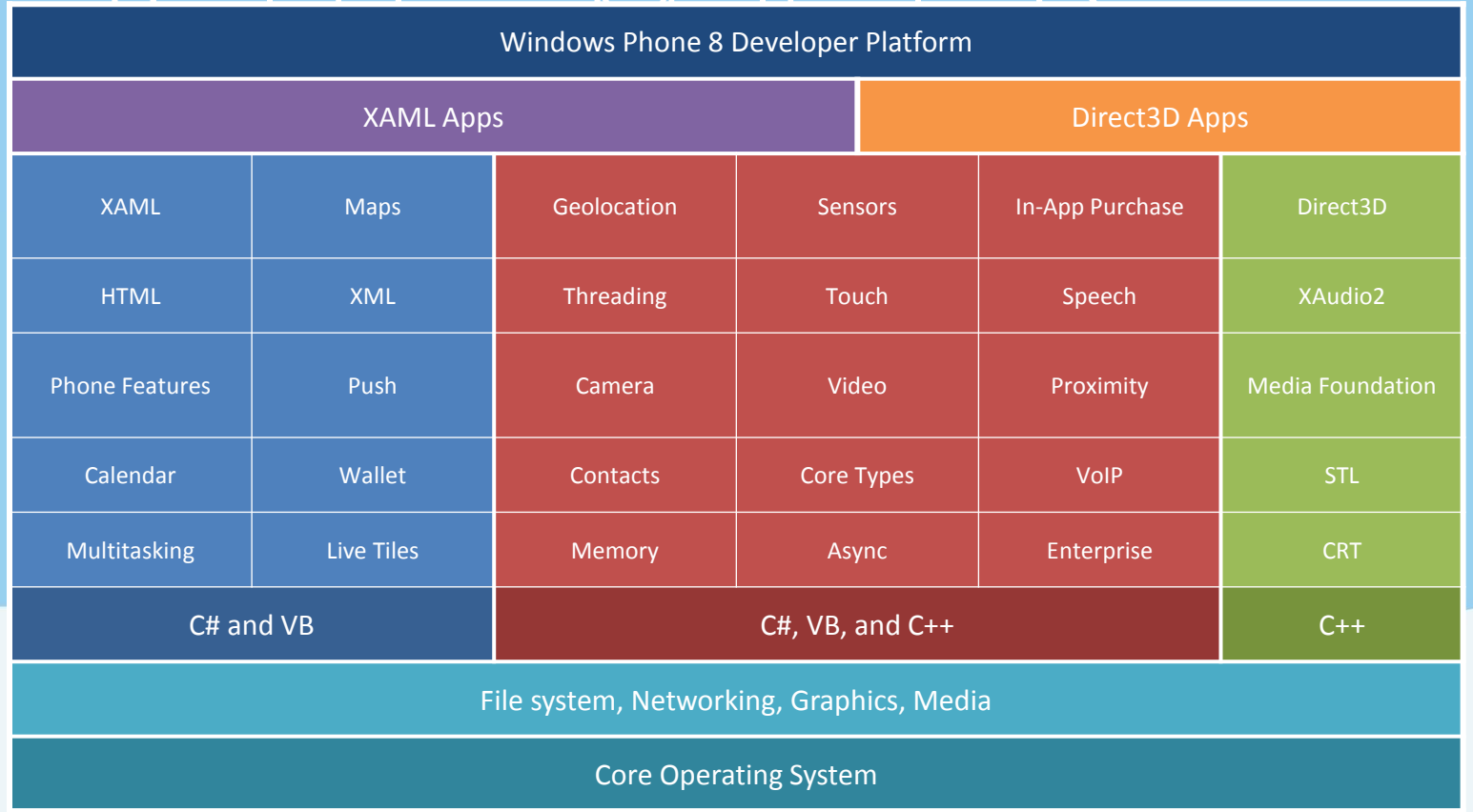
Shared Windows Core

- Windows 8 and Windows Phone 8 share many components at the operating system level

Shared Windows Core

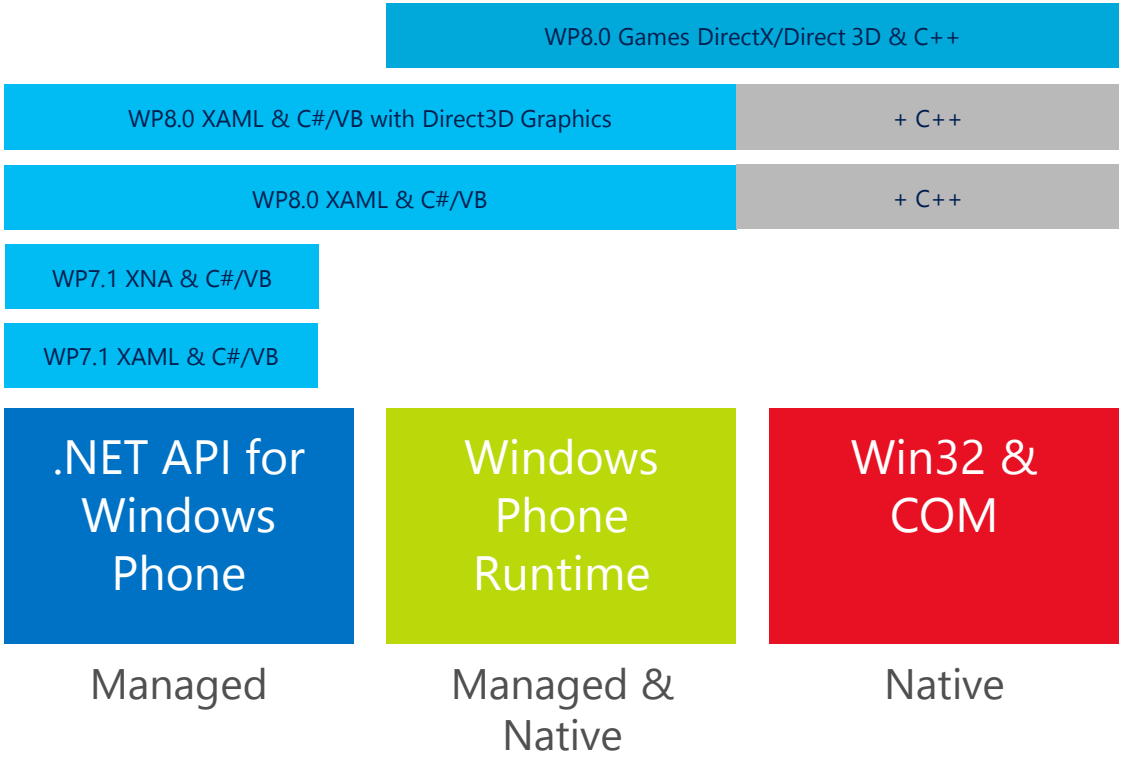
- Shared Core means
  - OS components such as the kernel, networking, graphics support, file system and multimedia are the same on both Windows 8 and Windows Phone 8
  - Hardware manufacturers work with the same driver model on both platforms
  - Windows Phone gets the support for multi-core and other hardware features that Windows has had for years
  - These solid, common foundations makes it easier to extend the Windows Phone platform into the future
- It doesn't mean
  - Windows 8 and Windows Phone 8 developers work to exactly the same APIs
    - (though you will see more commonality as new features are introduced to both platforms in the future)

Windows Phone Platform

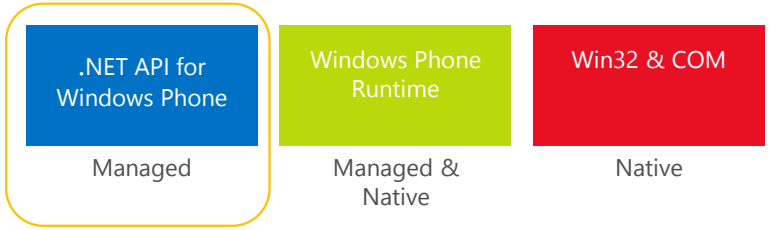


Windows Phone 8 Programming APIs

- Managed app dev using the WP7.1, WP8.0 .NET and WinPRT APIs
- Native app dev using WinPRT and Win32
- Games dev using the WP7.1 XNA framework
- Games dev using Direct3D or DirectX



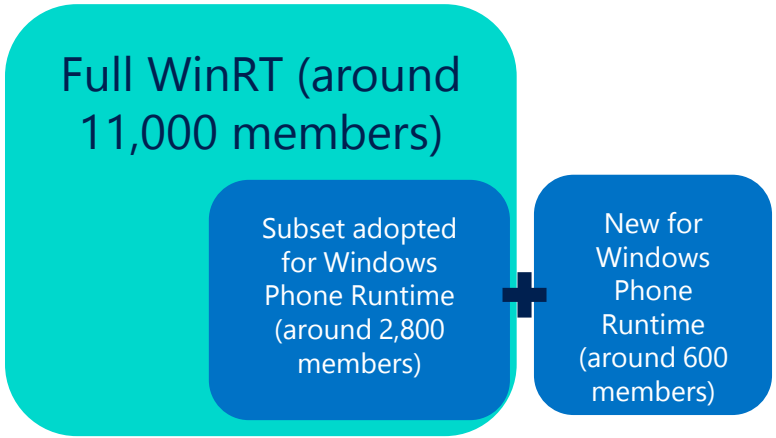
.NET API for Windows Phone



- The .NET API for Windows Phone is the primary managed API
  - Includes *all* the types and APIs from Windows Phone OS 7.1
  - Contains classes and types from the System and Microsoft.Phone namespaces
- New classes added for Windows Phone 8.0
  - Microsoft.Phone.Wallet
  - Microsoft.Phone.Tasks.ShareMediaTask
  - Microsoft.Phone.Tasks.MapsTask
  - Microsoft.Phone.Storage.ExternalStorage
  - Microsoft.Phone.Networking.Voip
  - Many more…!

Windows Phone Runtime API

- Windows Phone Runtime is a subset of the full WinRT, plus some phone-specific additions
  - Windows (Phone) Runtime is implemented in C++ and projected into C#, VB.NET, and C++
  - HTML5/JavaScript projection not available on Windows Phone 8
- Phone-specific additions to Windows Phone Runtime include
  - Speech synthesis and recognition
  - Windows.Phone.Networking.Voip
  - Windows.Phone.PersonalInformation
  - LockScreen and LockScreenManager
  - More…



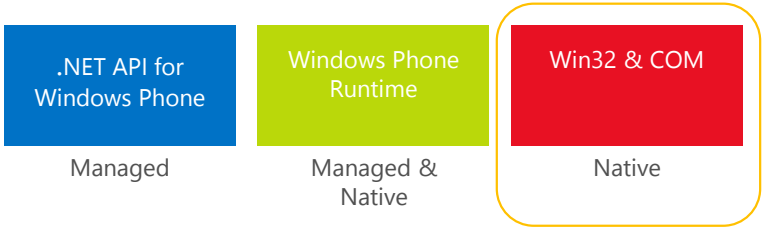
API Choices for Managed Code Developers

- Many of the APIs in Windows Phone Runtime exist to provide new functionality to Windows Phone
- Other APIs exist to expose Windows Phone capabilities to both native and managed code developers and provide equivalent functionality to the .NET APIs

.NET API	Windows Phone Runtime API
System.IO.IsolatedStorage	Windows.Storage
System.NET.Sockets	Windows.Networking.Sockets
System.Threading.ThreadPool	Windows.System.Threading.ThreadPool
Microsoft.Devices.Sensors	Windows.Devices.Sensors
System.Device.Location	Windows.Devices.GeoLocation

- Managed code developers can use whichever API they like
  - Developers targeting WP7.1 and WP8 devices will prefer the .NET API
  - Developers sharing code between WP8 and W8 targets will tend to use the Windows Phone Runtime API

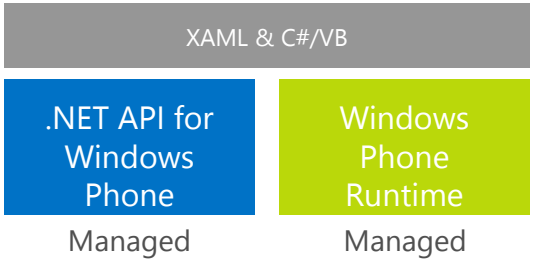
Win32 and COM API

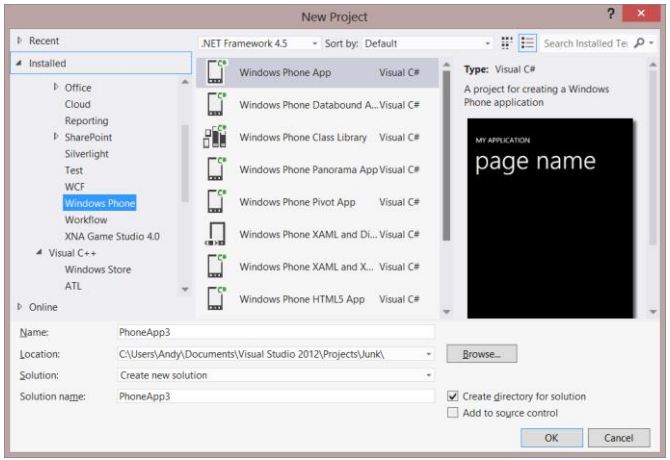


- In addition to .NET and Windows Phone Runtime, you have access to some Win32 APIs
  - Winsock for low-level networking
  - Camera APIs for native code apps
  - COM APIs such as CoInitializeEx, CoTaskMemAlloc, CoTaskMemFree, CreateFile2, ReadFile, WriteFile, HeapAlloc, CreateMutexExW, WaitForSingleObjectW,…many others…
- Mainly of interest to native code developers
- Managed applications could call these by adding a native project to the solution, but there are few situations where this will be used

XAML UI with Managed Code

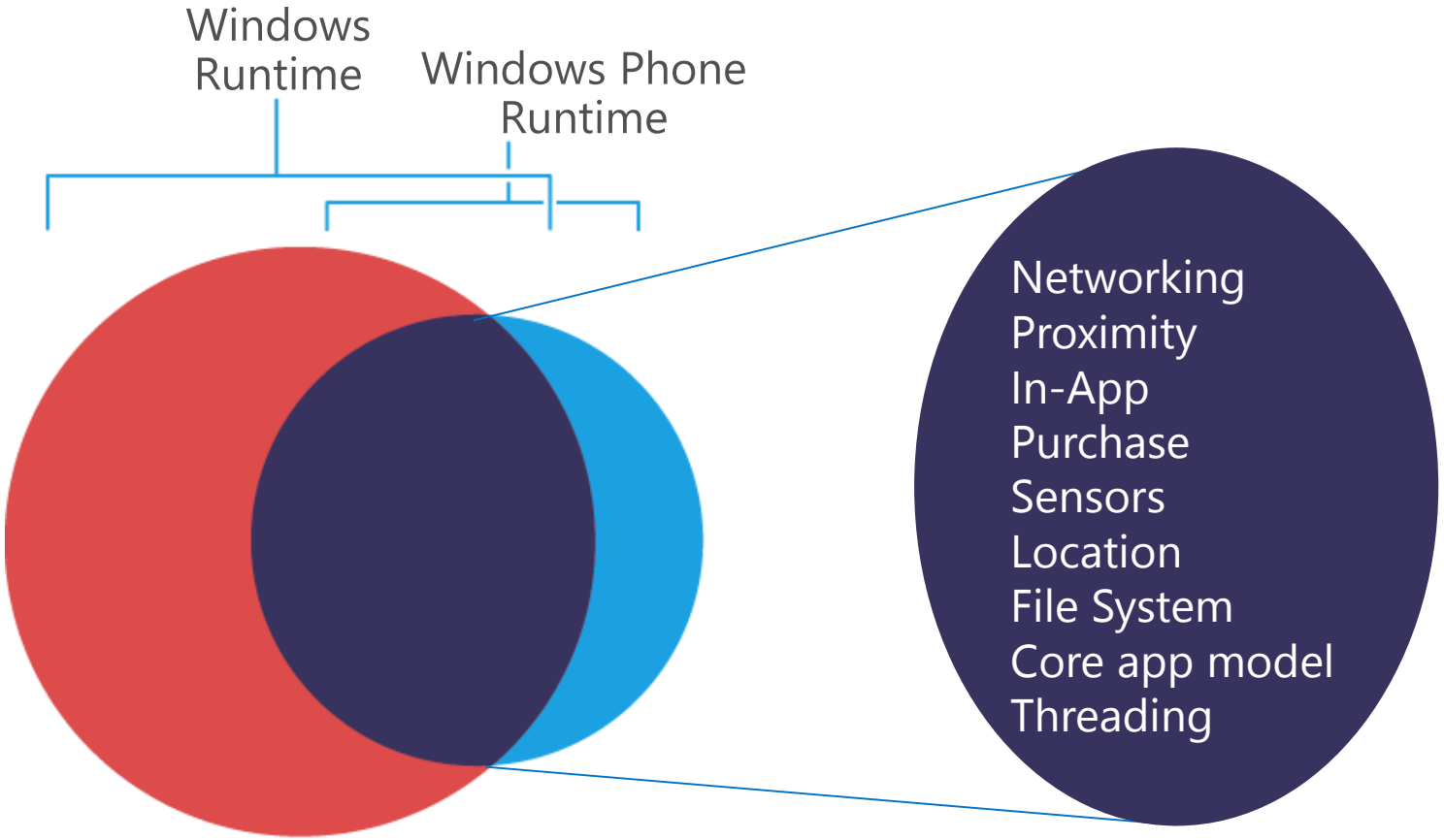
- The most common way to build apps for Windows Phone
- UI defined using XAML
- Logic written using C# or Visual Basic .NET
- Access .NET APIs and Windows Phone Runtime APIs





Building For Both

Shared Platform



Form Factors

- Windows 8
  - 1024x768+
  - Landscape, Portrait, Snapped, Filled
  - 10 inch screens+
- Windows Phone 8
  - 800x480, 1280x720, 1280x768
  - Portrait, Landscape
  - <5 inch screen most common

User experience considerations
Design a native UX for each platform!

- Windows 8
  - One or two-handed touch, mouse
  - No guarantee of any specific hardware, must check at runtime
  - Rows and columns of content can work well
  - Scroll horizontally for more content
  - Significant room on the app bar
  - On-screen back button
  - Semantic zoom
- Windows Phone 8
  - One-handed touch most common
  - Guaranteed hardware, such as camera and accelerometer
  - Avoid multiple columns of content
  - Scroll vertically for more content
  - Very limited room on the app bar
  - Hardware back button
  - No semantic zoom

XAML

- Avoid reusing XAML across Windows Store and Windows Phone apps
- Major differences in the platforms make this difficult anyway:
  - User experience
  - Screen space
  - Page layout / orientation
  - XAML namespaces
  - XAML controls

Resources

- Windows 8 Camp Training Kit
  - http://www.microsoft.com/en-us/download/details.aspx?id=29854
- Build your first Windows Store app
  - http://channel9.msdn.com/Series/Build-your-first-Windows-Store-app
- Windows Store apps for Absolute Beginners with C#
  - http://channel9.msdn.com/Series/Windows-Store-apps-for-Absolute-Beginners-with-C-
- Windows Store apps for Absolute Beginners with JavaScript
  - http://channel9.msdn.com/Series/Windows-Store-apps-for-Absolute-Beginners-with-JavaScript

Resources

- Windows Phone 8 Training Kit
  - http://www.microsoft.com/en-us/download/details.aspx?id=38782
- Windows Phone 8 Development for Absolute Beginners
  - http://channel9.msdn.com/Series/Windows-Phone-8-Development-for-Absolute-Beginners
- Building Apps for Windows Phone 8 Jump Start
  - http://channel9.msdn.com/Series/Building-Apps-for-Windows-Phone-8-Jump-Start

Resources

- Building Apps for Both Windows 8 and Windows Phone 8 Jump Start
  - http://channel9.msdn.com/Series/Building-Apps-for-Both-Windows-8-and-Windows-Phone-8-Jump-Start