Cross Platform Development

Is there a modern technology available for using the same codebase to produce native apps on all of the currently popular platforms?

- *iOS (iPhone/iPad/iPod Touch)*
- *Android (Smart Phone and Tablet)*
- *Windows (Desktop and phone),*
- *Mac*
- *Linux*

A Brief History of Cross Platform Desktop UI Toolkits

- In the 1980's the problem of a cross platform desktop user interface was for the most part solved by the X Window system, known as X11, (1984-).
- In 1989 Motif (a widget toolkit) was built on top of X11, and supported all the major platforms at that time:
  - Microsoft Windows
  - Various flavours of Unix, like Sun Solaris, HP-UX, IBM's AIX etc.
  - Later came Linux along and maintained full support for X11 and Motif
- Developers who wanted to target multiple platforms would usually write in C/C++ and use X11 and/or Motif.
- The UI code for the most part could remain the same and the application just needed to be recompiled for each platform.

But Motif has pretty much faded into the background and been replaced with newer widget toolkits.

Free Cross Platform Toolkits For The Desktop

- Motif has pretty much faded into the background and been replaced with newer widget toolkits, still built on top of X11:
  - Qt (1991-),
  - wxWidgets (1992-)
  - GTK+ (1998-).
  - These now run on many different platforms and bindings are available for many languages.

- These three free open source toolkits have been successful for the desktop application case but looking towards mobile and tablet platforms, these toolkits don't currently have the support there to take them into the future.

Qt

- **Native language**: C++
- **Platforms**: Windows, Mac, Linux, Symbian and Meego.

- Probably the C++ based Qt widget toolkit is the most well established way of writing desktop cross platform applications.
- Smartphone support isn't really there yet, but there is an unofficial port to iOS.
- Many significant apps have been developed with Qt, including Autodesk Maya, VLC media player, Mathematica and Skype.

GTK+

- **Native language**: C
- **Platforms**: Windows, Mac, Linux and HTML5 (unofficial).

- The C-based GTK+ widget toolkit is also a very well established way of writing cross platform desktop apps.
- Smartphone support isn't there, but there is an unofficial port to HTML5.
- Significant apps have been developed with GTK+, including The GNOME desktop environment and GIMP.

wxWidgets

- **Native language**: C++
- **Platforms**: Windows, Mac, Linux and iOS .

- A C++ native mode toolkit that provides a thin abstraction to a platform's native widgets.
- It was originally developed as a desktop GUI toolkit in the same vein as GTK+ and Qt and support for iOS is still beta.
- There are a number of notable apps developed with wxWidgets including BitTorrent, TortoiseCVS and RapidSVN.

WRITE ONCE, RUN ANYWHERE

Java, Flash and Silverlight

Java

- Sun Microsystems released the first public implementation in 1995.
- It promised "Write Once, Run Anywhere" providing no-cost run-times (VM) on popular platforms.
- Major web browsers soon incorporated the ability to run Java applets within web pages, and Java quickly became popular.
- For GUI development Sun's created Swing UI Library (1997-).

- Although technically successful Swing attempted to replace the 'native' look of apps developed on a particular platform and instead imposed its own look and feel based on graphical primitives, which although being more powerful for developers, users on the whole didn't like.

Flash

- Adobe Flash is a multimedia platform used to add animation, video, and interactivity to web pages on different platforms.
- Flash is frequently used for advertisements, games and flash animations for broadcast.
- Started as FutureSplash in 1995 but was acquired by Macromedia and released as Flash in 1996 and later acquired by Adobe.
- Flash Player for smart phones was made available to handset manufacturers at the end of 2009, but have never succeeded to get on ios.
- In November 2011 Adobe announced the end of Flash for mobile platforms, instead focusing on HTML5 for browser content and Adobe AIR for the various mobile AppStores

Silverlight

- Microsoft Silverlight (2007-) is an application framework for writing and running rich Internet applications, with features and purposes similar to those of Adobe Flash.
- The run-time environment for Silverlight is available as a plug-in for web browsers running under Microsoft Windows and Mac OS X.
- Like Flash Silverlight has had a hard time getting on Smart Phone platforms, and it seems like Microsoft is focusing on HTML5 and JavaScript for cross-platform development.

THE RISE OF WEB APPLICATIONS AND THE APP STORE

Smartphones And Tablets

The two big changes since the desktop UI toolkit days:

- the rise of browser-based web applications and
- the rise of downloadable apps running on smartphones and tablets
  - These new class of devices offer a quick method of downloading, paying for and installing apps.

- Microsoft is getting in on the App Store action a little late with its 'Windows Store' for Windows 8.
  - Windows Phone 8 OS is based on the same technology as Windows 8 and hence will unify the development platform for Microsoft's desktop, tablet and smartphone operating systems.
  - Windows 8's new 'Metro' style apps on the Windows Store will run on desktops and tablets and look like those on the Windows phones.

Web Applications

- if you don't need much access to hardware and can remain restricted to the browser, then HTML/CSS and JavaScript is the way to go.
- Browser based web applications have their own set of problems however:
  - the browser incompatibilities,
  - varying support for the HTML/CSS web standards
  - and differences in the Document Object Model (DOM).

- The main advantage of the browser application is that its everywhere and supported by every device, so when HTML5 is fully supported on all platforms and browsers, then it could become the defacto standard for developing apps of any kind, desktop or not.

Cross Platform Mobile Development SDK

- Until HTML5 matures we may use on of the Cross Platform Mobile Development frameworks:
  - Mono
  - PhoneGap
  - Appcelerator Titanium
  - Rhodes
  - MoSync
  - Moai
  - Corona SDK
  - JUCE
  - …

Mono / Monotouch / Mono for Android

- **Native language**: C#
- **Platforms**: iOS, Android, Windows, Mac, Linux, Browser (e.g. via ASP.NET WebForms or MVC).
- **Cost:** Windows, Mac & Linux: free.
  Android and iOS: from US$399 each.
- **License:** LGPL/GPL/X11 combination. Commercial license available.
- **Source**: Mono for Windows/Mac/Linux is open source.
  Mono for iOS/Android is closed source.
- Mono targets both mobile and desktop apps
  - The idea is that you build your software using the MVC (Model View Controller) pattern, so that the Model and Controller components (both written in C#) can be shared across all platforms without any changes. Only the View component needs to be re-written for each platform (again, in C#).

PhoneGap / Apache Cordova

- **Native language**: HTML5/JavaScript
- **Platforms**: iOS, Android, Windows Mobile, Blackberry, Symbian, Palm.
- **Cost:** Free
- **License**: Apache License, Version 2.0
- **Source**: Open source
- PhoneGap is an open source framework that helps you develop apps for smartphones using web development languages such as JavaScript and HTML5.
- It also allows for access to hardware features including GPS/location data, accelerometer, camera, sound and more through javascript.

JUCE

- **Native language**: C++
- **Platforms**: Windows, MacOS, Linux, iOS, Android (work in progress).
- **Cost:** Free for GPL license. Commercial license from £399 per product.
- **License**: GPL. Commercial license available.
- **Source**: Open source
- JUCE is an all-encompassing C++ class library for developing cross-platform software. Its designed to contain everything you're likely to need to create most applications, and is particularly well-suited for building highly-customised GUIs, and for handling graphics and sound.
- Founded by Julian Storer in 1999, who is still the primary developer today.

Appcelerator Titanium

- **Native language**: JavaScript, Python and Ruby
- **Platforms**: iOS, Android, Windows, Mac, Linux.
- **Cost:** Community Edition is free.
  Indie Edition (extra APIs, help, support): US$499 per year.
- **License**: Apache Public License v2
- **Source:** Open source
- Allows you to use web development languages (HTML, JavaScript, CSS, Python and Ruby) to build mobile and desktop apps.
- App data can be stored in the cloud or on the device, and apps can take full advantage of hardware, particularly camera and video camera capability.

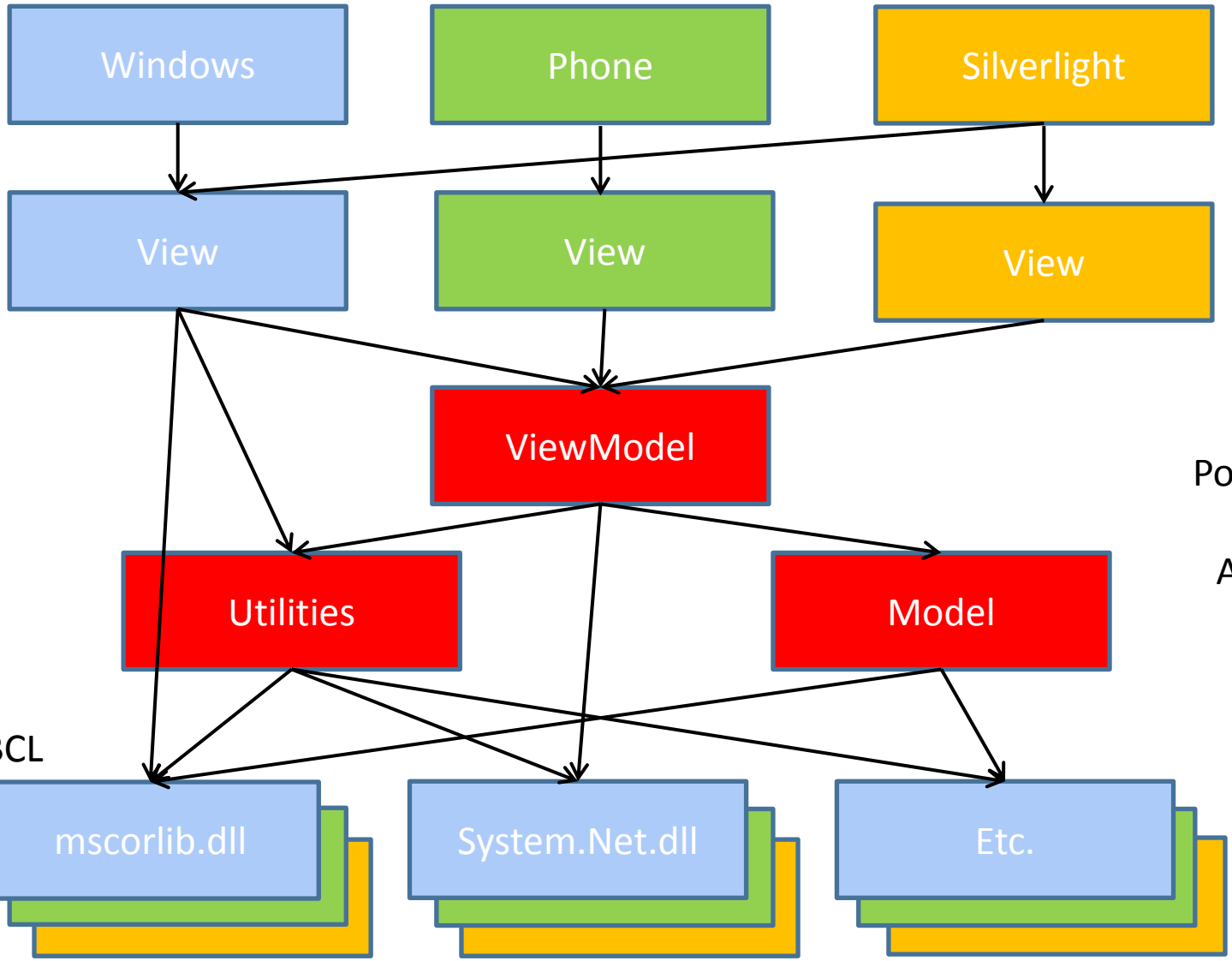
MICROSOFTS SUPPORT FOR CROSS PLATFORM DEVELOPMENT

Portable Class Libraries

- Even though C# compiles to IL that may be JIT compiled to different platforms it has not really been possibly to utilise this until recently (2011).
- The problem is that most (all) code uses classes (packed in assemblies) from the .Net framework. And these assemblies are specific for each platform.

- But in 2011 MS released The Portable Class Library project that enables you to write and build managed assemblies that work on more than one .NET Framework platform.
- You can create classes that contain code you wish to share across many projects, such as shared business logic, and then reference those classes from different types of projects.

Sharing Portable Class Libraries



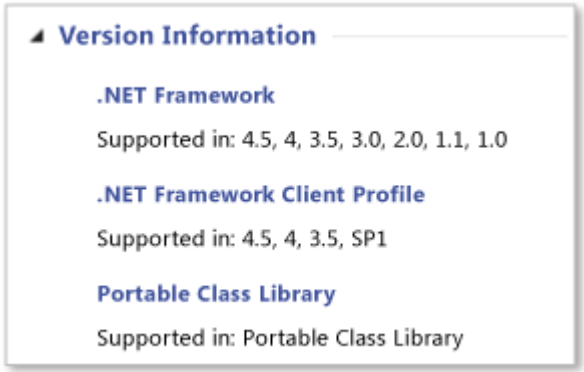
Portable Class Library Assemblies

Supported Features

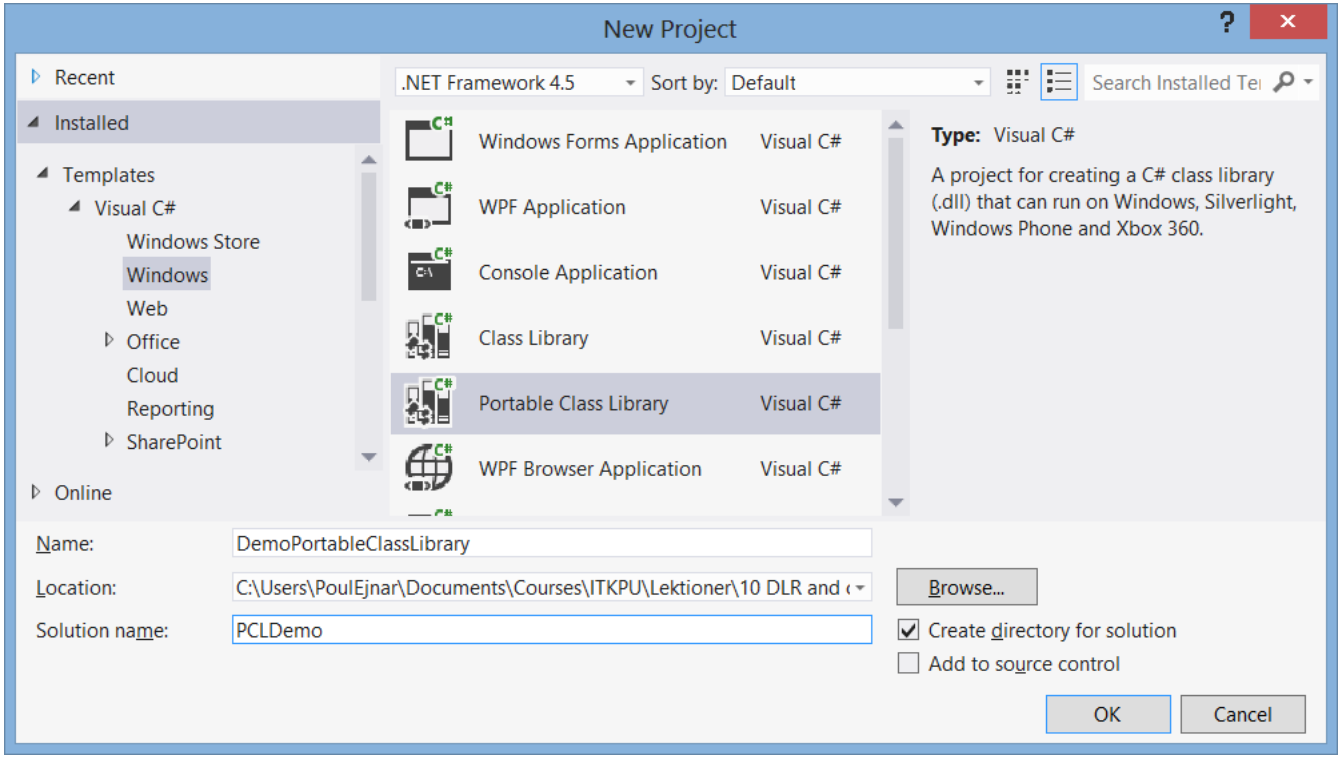
Feature	.NET Framework	Windows Store	Silverlight	Windows Phone	Xbox 360
Core	√	√	√	√	√
LINQ	√	√	√	√	
IQueryable	√	√	√	7.5 and higher	
Dynamic keyword	Only 4.5	√	√		
Managed Extensibility Framework (MEF)	√	√	√		
Network Class Library (NCL)	√	√	√	√	
Serialization	√	√	√	√	
Windows Communication Foundation (WCF)	√	√	√	√	
Model-View-View Model (MVVM)	Only 4.5	√	√	√	
Data annotations	Only 4.0.3 and 4.5	√	√		
XLINQ	Only 4.0.3 and 4.5	√	√	√	√
System.Numerics	√	√	√		

Supported Types and Members

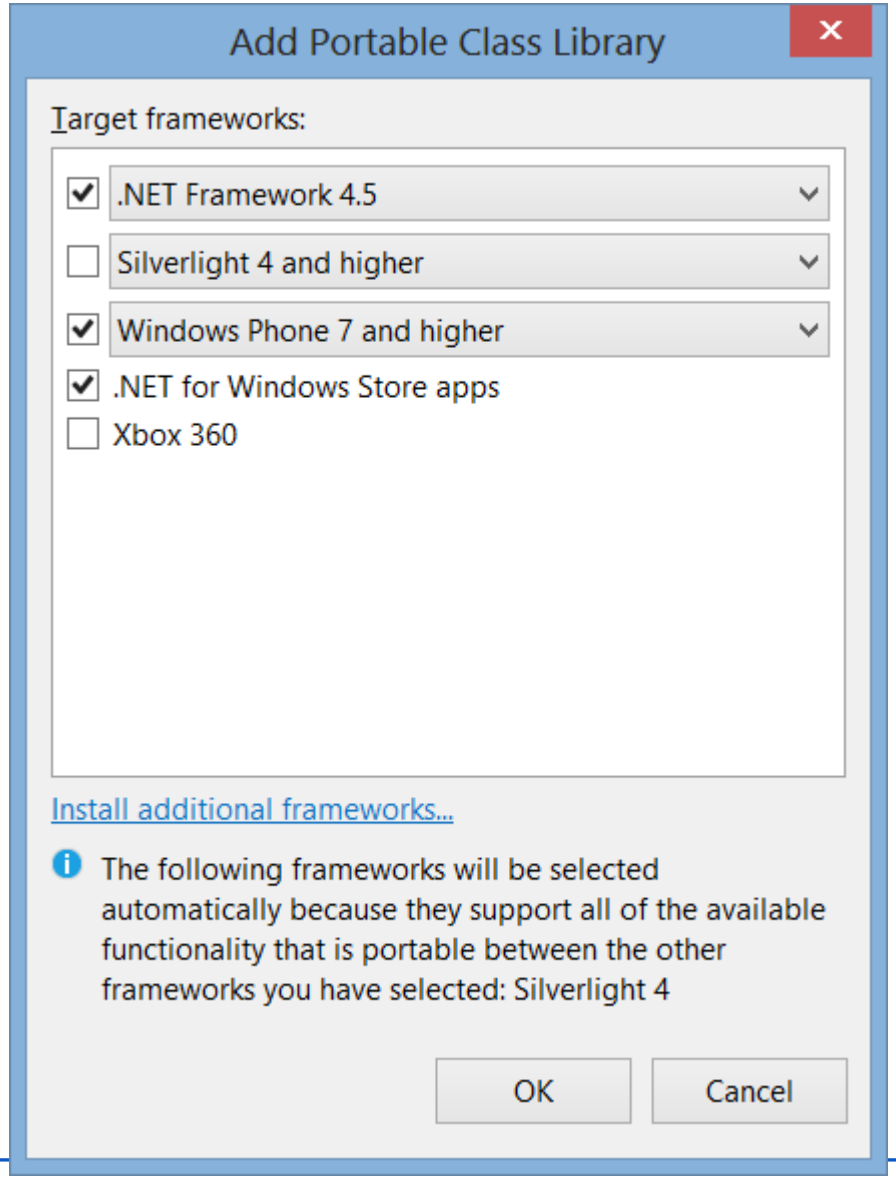
- The types and members that are available in Portable Class Library projects are constrained by several compatibility factors:
  - They must be shared across the target platforms you selected.
  - The must behave similarly across those platforms.
  - They must make sense in a portable environment, especially when supporting members are not portable.
- The Portable Class Library project doesn't contain any UI-related types or members because of behavioral differences between the UIs of different devices.
- You can find which members are supported by the Portable Class Library in the reference topics for the .NET Framework Class Library.



Creating a Portable Class Library Project



Choose platforms to Suppord



Beware of API Differences

- To make Portable Class Library assemblies compatible across all supported platforms, some members have been slightly changed in the Portable Class Library.
- For details about which members were changed and how they were changed, see “API Differences in Portable Class Library”.

References

- Modern Cross Platform Development
  http://www.dodgycoder.net/2012_01_01_archive.html
- Microsoft Portable Class Libraries
  http://msdn.microsoft.com/en-us/library/gg597391.aspx