ICT365

Software Development Frameworks

Dr Afaq Shah

Murdoch UNIVERSITY

XAMARIN
Cross-platform Mobile Apps

Murdoch
UNIVERSITY

Aims

XAMARIN

Cross-platform Mobile Apps

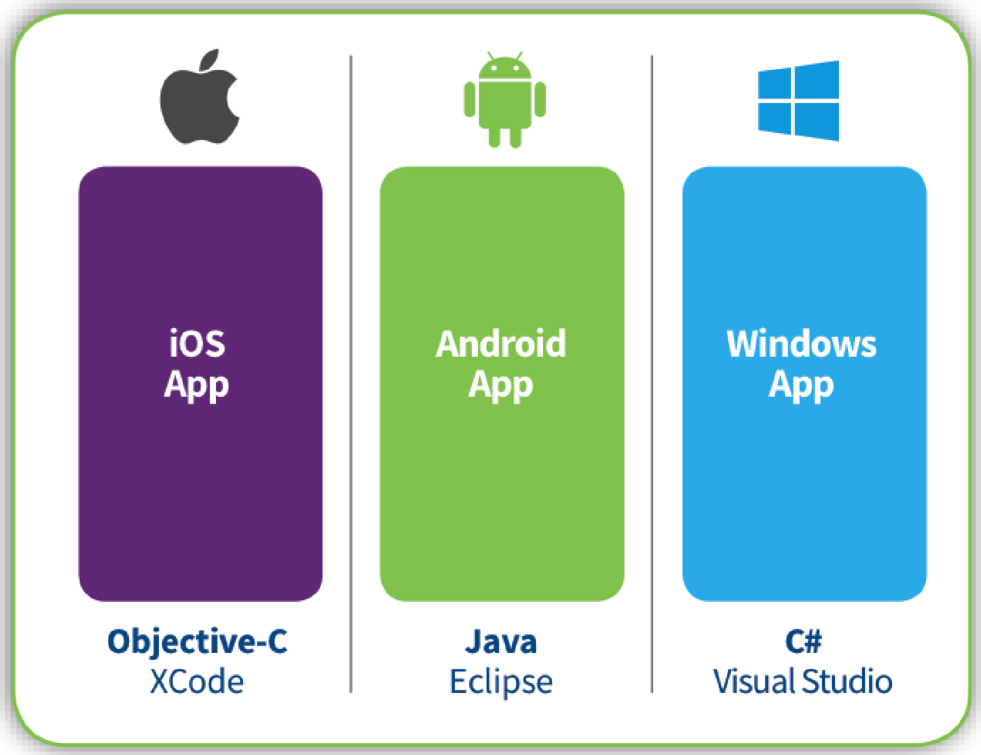
Mobile Development Approaches

What are the top 2 mobile browsers?

Safari

Android Webkit

Silo'd Approach



- Multiple Teams
- Multiple Code Bases
- Expensive & Slow
- Positive = Great apps delivered to user's platform
- Negative = Development hampered by multiple code bases & fragmentation

Silo – Write App on every Target

Benefits

Full native experience

Total access to the device as provided by SDK

Share Web API

Negatives

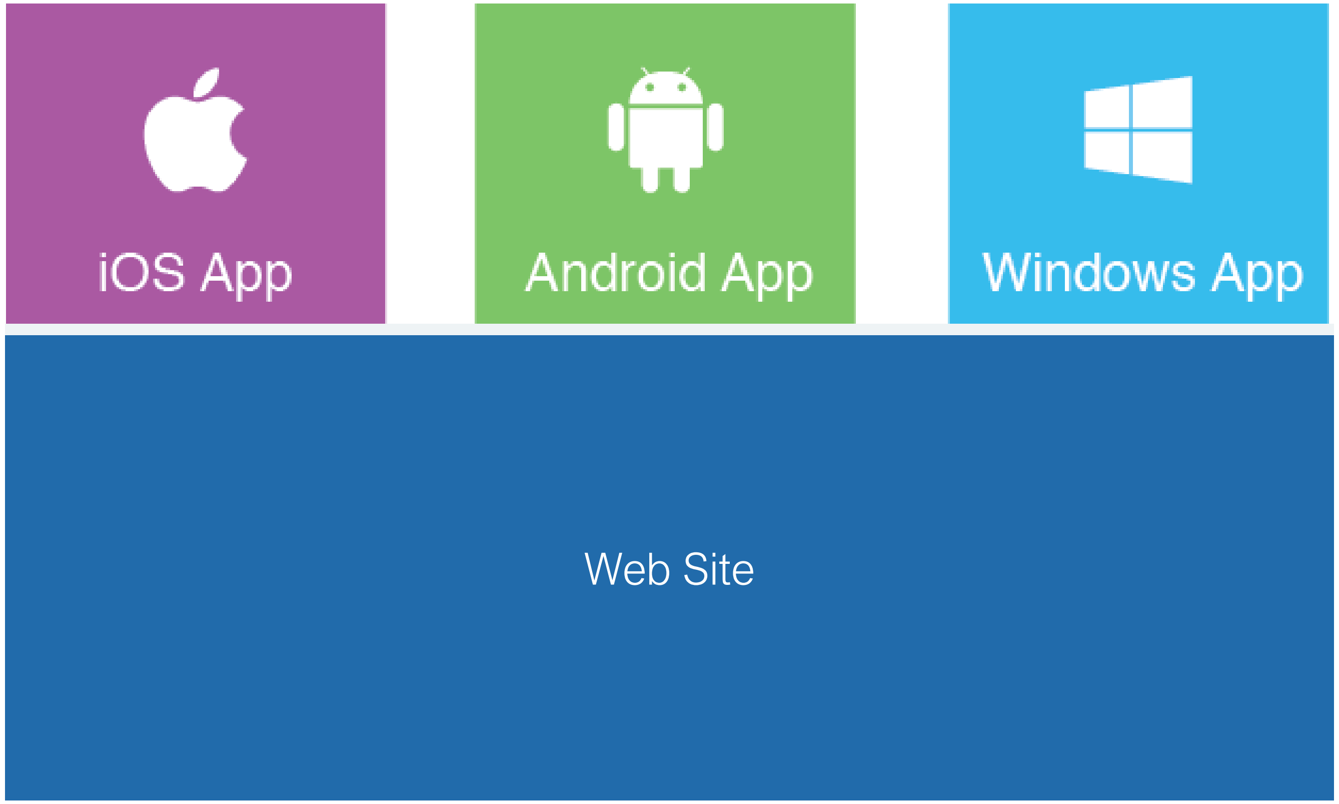
Minimal re-use mostly on back end Web API

Higher development cost from multiple teams (silo teams) or expensive multi-device developers

Multiple codebases to maintain and extend

One platform rules the others are subservient

Target Browser Not OS



Didn’t HTML5 solve this?

Complicates build for some use cases.

Patchy or varied support for some features cross-browser or webview.

HTML5 lessons learned

"I think the biggest mistake that we made, as a company, is betting too much on HTML5 as opposed to native..." – Mark Zuckerberg, Facebook

"We have definitely shifted from HTML5 to native. The primary reason for that is .. people are spending more time in the app, and the app is running out of memory…" – Kiran Prasad, LinkedIn

HTML – Write App using Mobile Web

Benefits

Provide consistent experience regardless of target

Cheap as it is just HTML

Single codebase to maintain and extend

No need for revenue sharing as no need to be in app stores

Negatives

User experience tends to be webish and not native

Need to still test and debug multiple targets

Features tend to be a subset common to all targets

HTML – Write App using Mobile Web

- Tools

  HTML5

  jQuery Mobile

  Sencha Touch

  ASP.NET

  J2EE

Who cares about Cross Platform?

For most developers cross platform was just talk.

Prior to 2010 70+% of all computers ran a version of Windows.

MS was very good on backwards compatibility.

XP was kept alive by .NET

Picking Windows or Internet Explorer was a no brainer. Or more correctly WinForms/WPF or Internet Explorer was a no brainer.

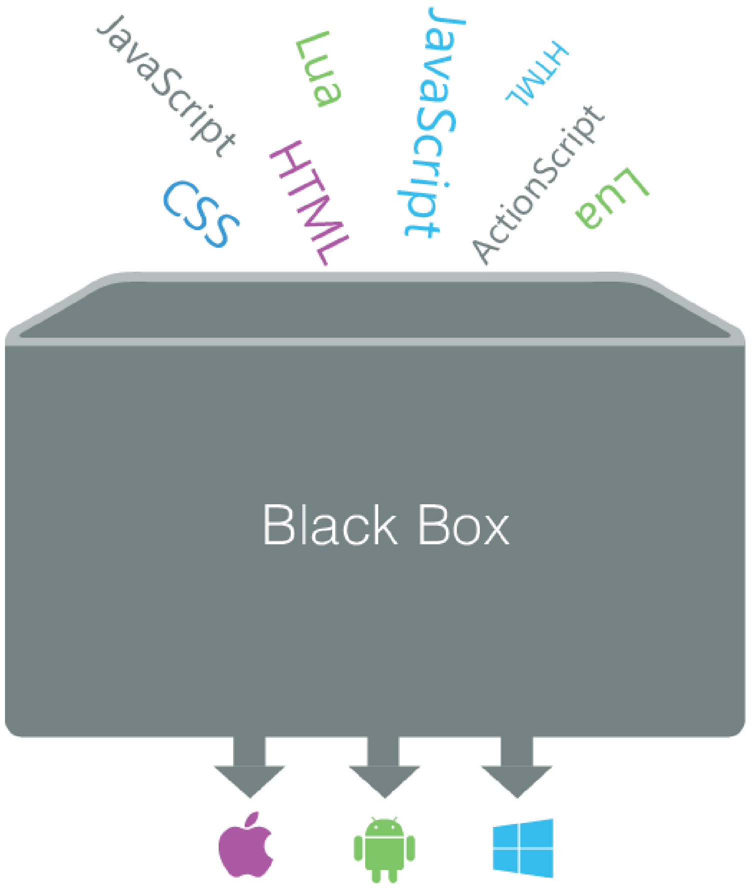
Who cares about Cross Platform?

2013 App Economy was 68 billion USD according to DeveloperEconomics.com or roughly 10 USD per person

2016 App Economy was 143 billion USD according to DeveloperEconomics.com or roughly 20 USD per person

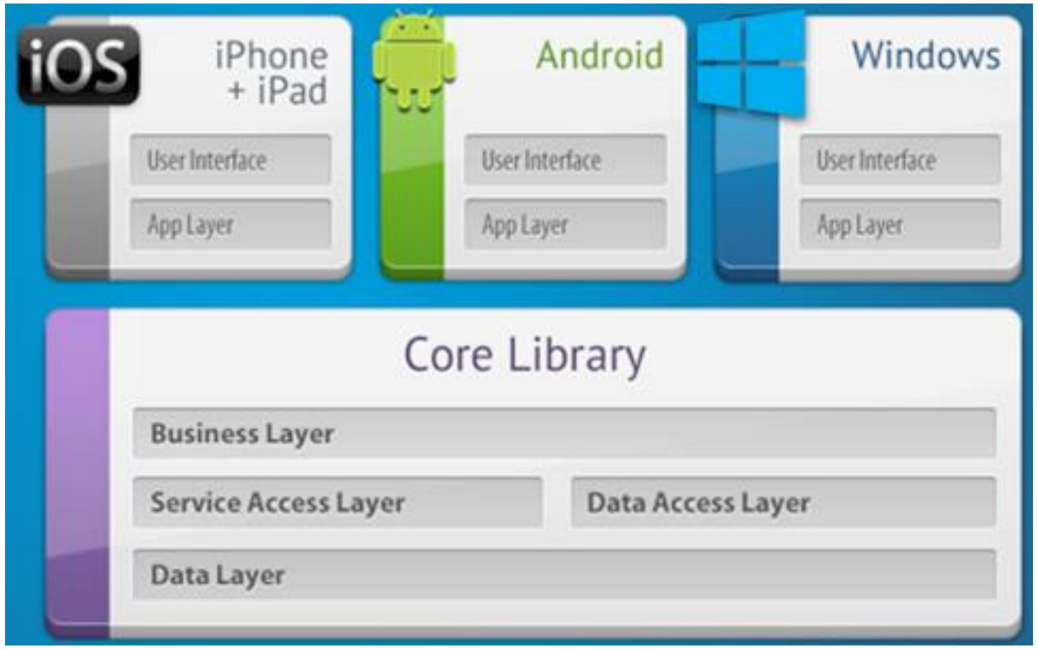
Problem is that there is no OS monopoly. What is a developer to do?

Target Developer Platform



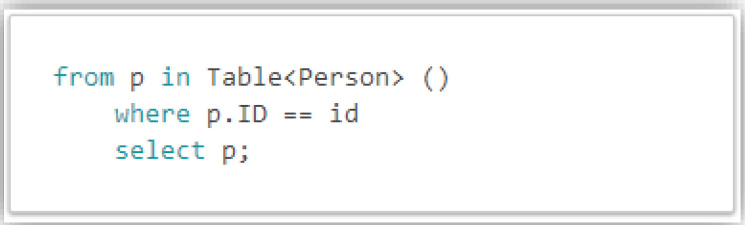
Write Once,
Run Anywhere Approach

Xamarin's Unique Approach



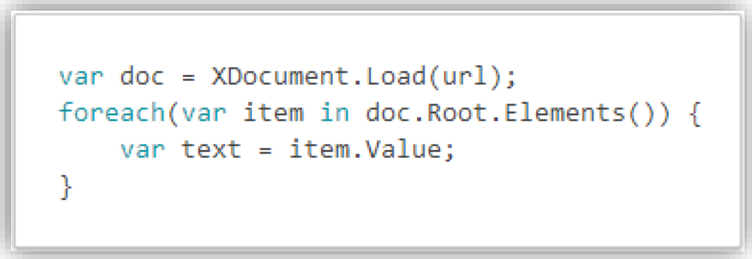
UI build natively per platform, leveraging C#
C# + XAML
C# + XML
C# + XIB
One shared app logic code base, iOS, Android, Mac, Windows Phone, Windows Store, Windows

C# ..



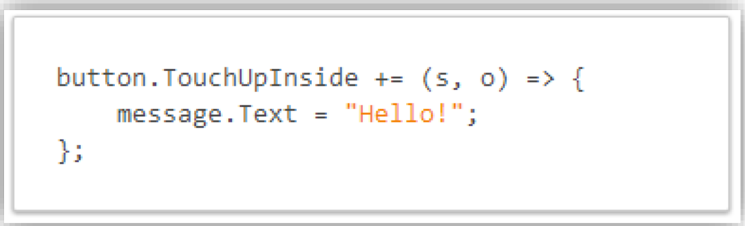
```
from p in Table<Person> ()
    where p.ID == id
    select p;
```

- LINQ Support



```
var doc = XDocument.Load(url);
foreach(var item in doc.Root.Elements()) {
    var text = item.Value;
}
```

- Work With XML Easily XDocument

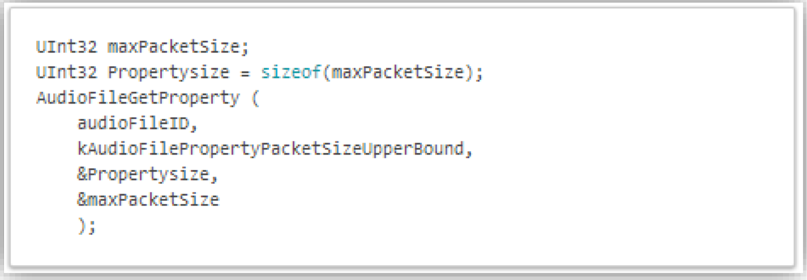


```
button.TouchUpInside += (s, o) => {
    message.Text = "Hello!";
};
```

- Event Handling & Delegates

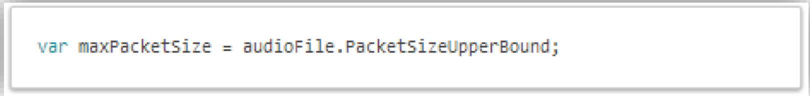
Probing for properties on an AudioFile

C



```
UInt32 maxPacketSize;
UInt32 Propertysize = sizeof(maxPacketSize);
AudioFileGetProperty (
    audioFileID,
    kAudioFilePropertyPacketSizeUpperBound,
    &Propertysize,
    &maxPacketSize
    );
```

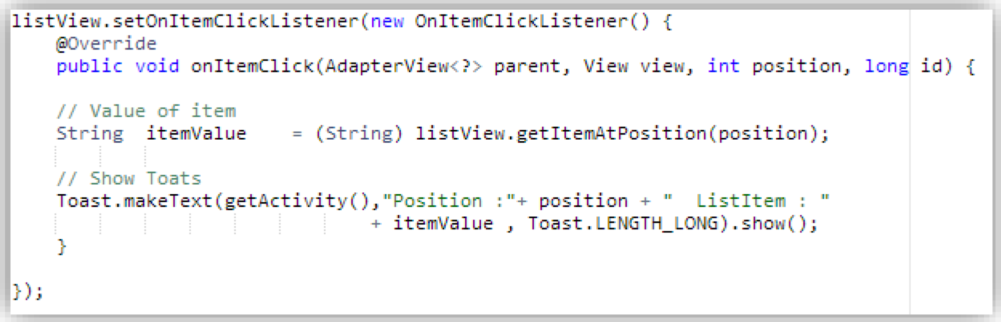
C# with Xamarin



```
var maxPacketSize = audioFile.PacketSizeUpperBound;
```

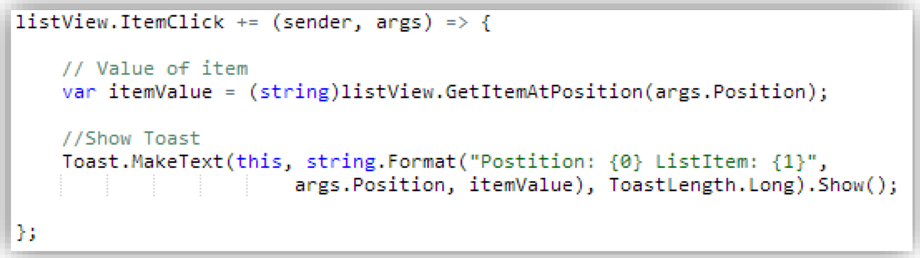
Different – Android ItemClick

Java



```
listView.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

    // Value of item
    String  itemValue    = (String) listView.getItemAtPosition(position);

    // Show Toats
    Toast.makeText(getActivity(),"Position :"+ position + "  ListItem : "
                            + itemValue , Toast.LENGTH_LONG).show();
    }

});
```

C# with Xamarin



```
listView.ItemClick += (sender, args) => {

    // Value of item
    var itemValue = (string)listView.GetItemAtPosition(args.Position);

    //Show Toast
    Toast.MakeText(this, string.Format("Postition: {0} ListItem: {1}",
                    args.Position, itemValue), ToastLength.Long).Show();

};
```

Here we can see how easy it is just to do a += for an event and not have to implement a bunch of listeners every time. Easy to read, string.Format, using args, etc

Write Everything in C#

- Take advantage of everything great about C# and now write code that can be shared across all platforms
- iOS
- Android,
- Mac
- Windows (WPF, Store, Phone, ASP.NET, etc)

2.5+ Billion Devices!

Xamarin History

Over a Decade of Enterprise Production Use

Murdoch UNIVERSITY

2000
Ximian Founded

2001
Mono Started

2003
Ximian Acquired by Novell

2009
First iOS product (now Xamarin.iOS) launches

2011
Xamarin Founded

First Release of Xamarin.Android

2012
Reach 200,000 Developer Mark

First release of Xamarin.Mac

Launch Partner Program

2013
500,000+ Developers
100+ Partners
100+ Components

Xamarin 2.0

Component Store

Evolve 2013

Xamarin Test Cloud

Microsoft Partnership

Xamarin History

- Xamarin is a descendant of Mono

  Mono is created in 2000 as a ".NET for Linux and MAC"

  Later developed Mono Touch and Mono for Android for development of apps for iOS, OS X and Android

- Xamarin is cross-platform

  The C# code is compiled to native code for each OS

    It works pretty fast

Xamarin History (2)

- In 2014 Xamarin introduced Xamarin.Forms

  A common UI for mobile platforms

  A way of reusing ~90% of the code for all mobile platforms

  Contains a Xamarin XAML that is much like the MS XAML

    Supports data-binding, dependency and attached properties

  It all compiles to native code

Murdoch UNIVERSITY

Microsoft and Xamarin Partner Globally

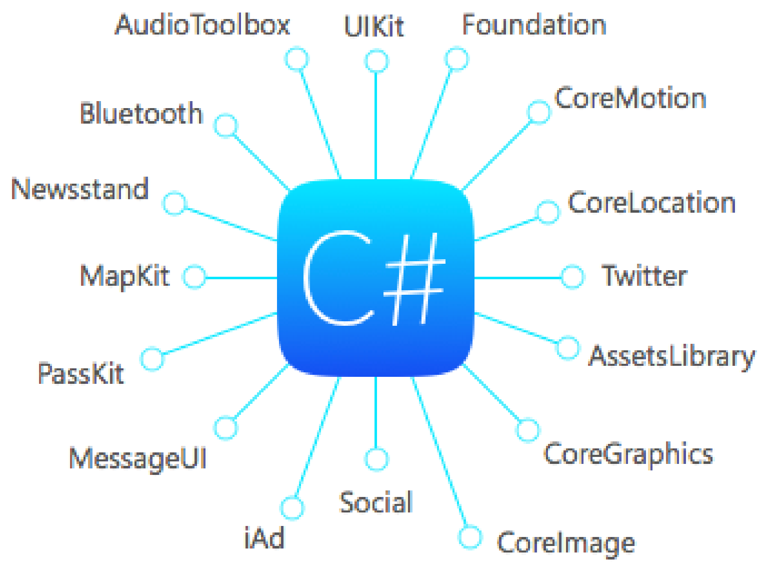
+ = C#

on 2.5 Billion Devices

With Xamarin, developers combine all of the productivity benefits of C#, Visual Studio 2013 and Windows Azure with the flexibility to quickly build for multiple device targets.”

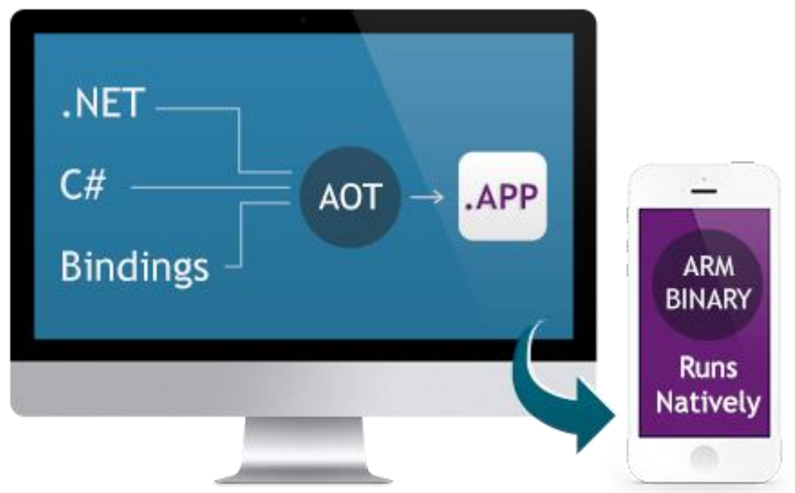
S. Somasegar, Corporate Vice President, Microsoft

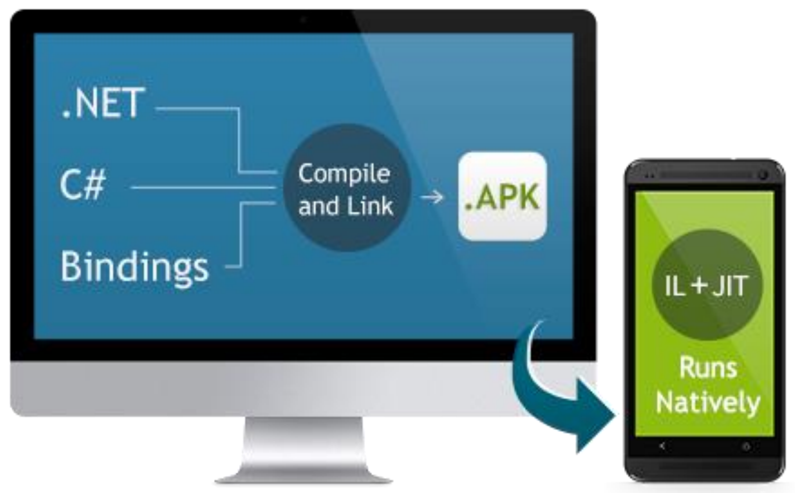
100% API Coverage



Anything you can do in Objective-C or Java can be done in C# and Visual Studio with Xamarin!

Native Performance





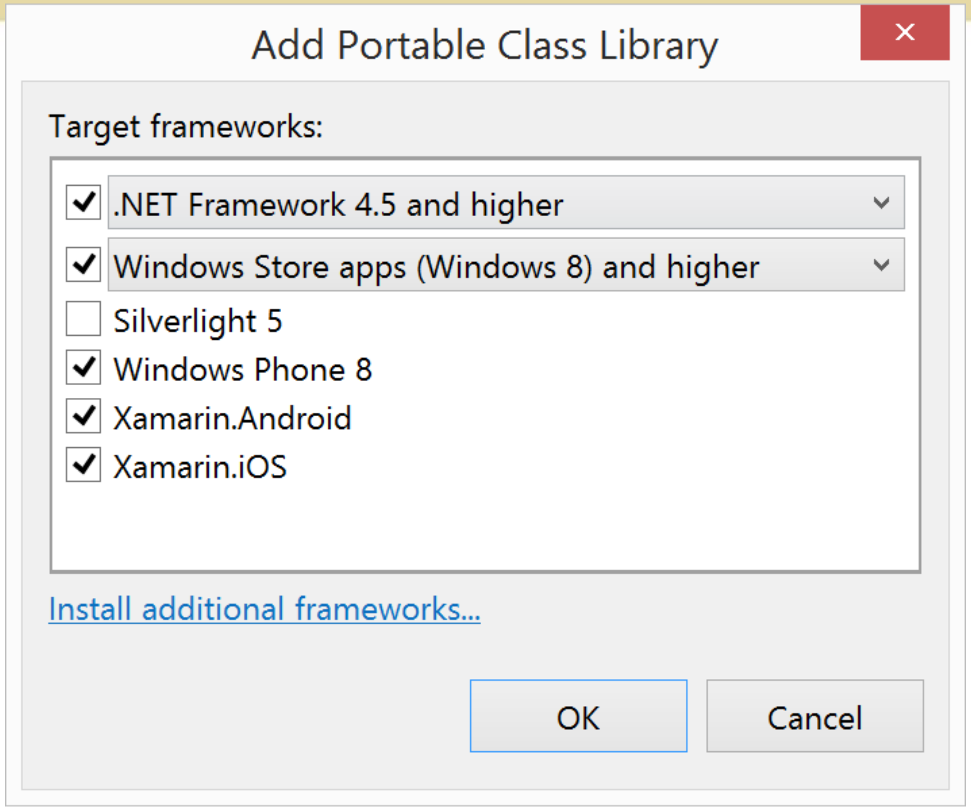
Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store.

Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device.

- There is no compromise on performance.
- Xamarin apps look and feel native because they are native.

Portable Class Libraries

- 1 Assembly
- Multiple Platforms
- **Including:**
- **Xamarin.Android**
- **Xamarin.iOS**



Add Portable Class Library

Target frameworks:

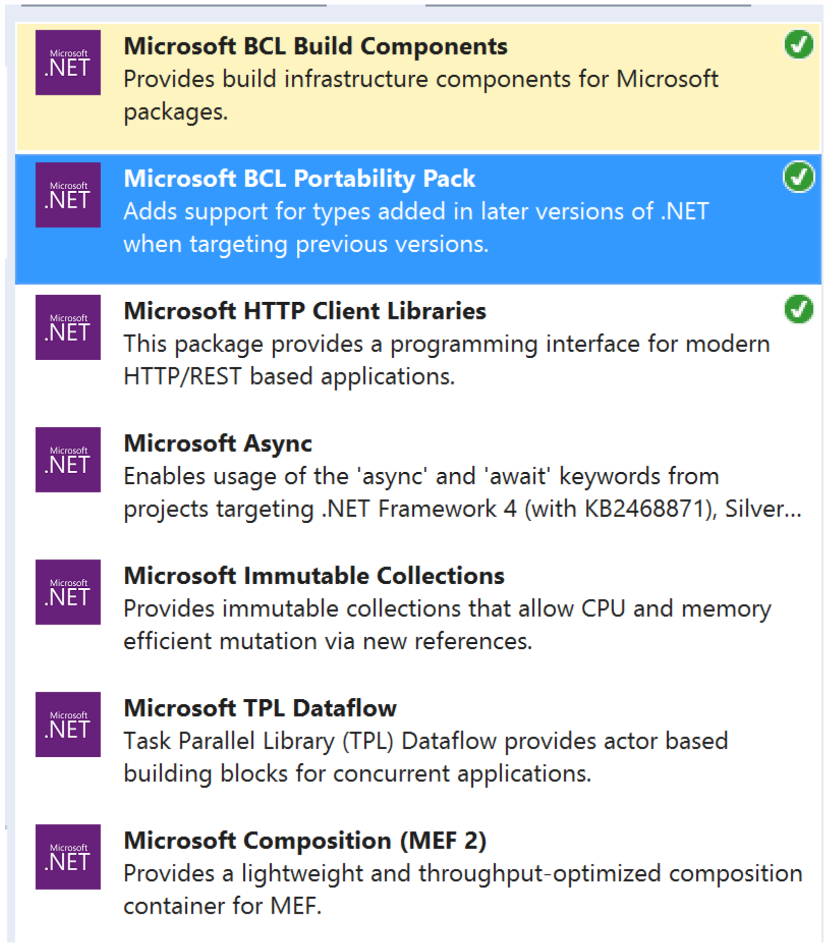
- [✓] .NET Framework 4.5 and higher
- [✓] Windows Store apps (Windows 8) and higher
- [ ] Silverlight 5
- [✓] Windows Phone 8
- [✓] Xamarin.Android
- [✓] Xamarin.iOS

- Write all of your C# code in one assembly and share across all platforms
- Before the Xamarin & Microsoft Partnership PCLs were limited ONLY to Windows Platforms
- Now add official support to create and use PCLs in Visual Studio and Xamarin Studio

Portable Class Libraries Features

- Centralized Code Sharing
  - How you expect it to work
  - Debug seamlessly into and out of PCL

- Project/Assembly Sharing
  - NuGet



Microsoft BCL Build Components ✓
Provides build infrastructure components for Microsoft packages.

Microsoft BCL Portability Pack ✓
Adds support for types added in later versions of .NET when targeting previous versions.

Microsoft HTTP Client Libraries ✓
This package provides a programming interface for modern HTTP/REST based applications.

Microsoft Async
Enables usage of the 'async' and 'await' keywords from projects targeting .NET Framework 4 (with KB2468871), Silver...

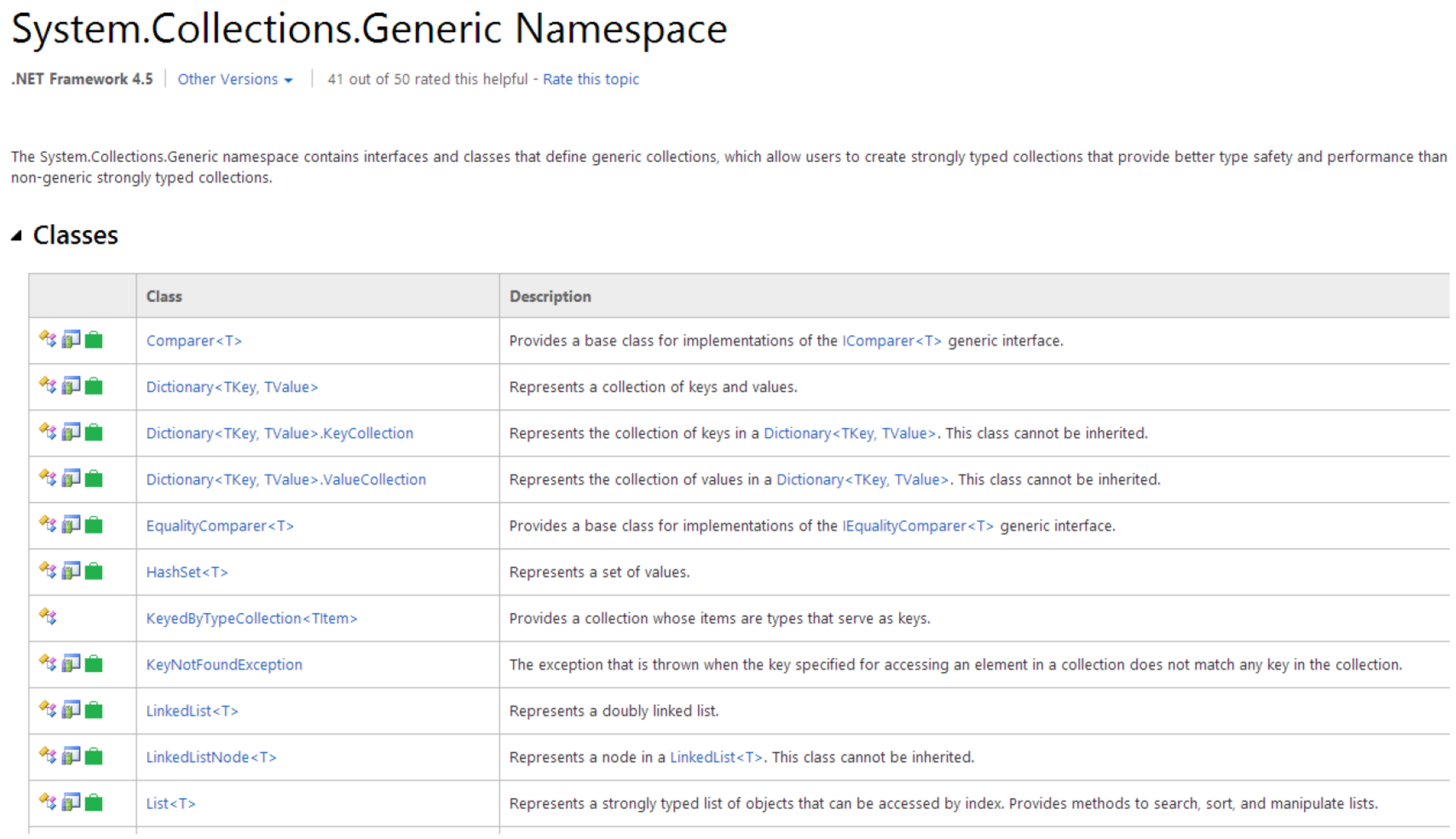
Microsoft Immutable Collections
Provides immutable collections that allow CPU and memory efficient mutation via new references.

Microsoft TPL Dataflow
Task Parallel Library (TPL) Dataflow provides actor based building blocks for concurrent applications.

Microsoft Composition (MEF 2)
Provides a lightweight and throughput-optimized composition container for MEF.

- Centralize all code how you want it to work and share across platforms
- Take advantage of NuGet to create and use libraries to your projects
- Easier to Create and Easier to consume in apps
- Create small reusable PCLs to share across all of your projects

PCLs – Well Documented



System.Collections.Generic Namespace

.NET Framework 4.5 | Other Versions ▾ | 41 out of 50 rated this helpful - Rate this topic

The System.Collections.Generic namespace contains interfaces and classes that define generic collections, which allow users to create strongly typed collections that provide better type safety and performance than non-generic strongly typed collections.

◢ Classes

	Class	Description
✦ ▣ 🟩	Comparer<T>	Provides a base class for implementations of the IComparer<T> generic interface.
✦ ▣ 🟩	Dictionary<TKey, TValue>	Represents a collection of keys and values.
✦ ▣ 🟩	Dictionary<TKey, TValue>.KeyCollection	Represents the collection of keys in a Dictionary<TKey, TValue>. This class cannot be inherited.
✦ ▣ 🟩	Dictionary<TKey, TValue>.ValueCollection	Represents the collection of values in a Dictionary<TKey, TValue>. This class cannot be inherited.
✦ ▣ 🟩	EqualityComparer<T>	Provides a base class for implementations of the IEqualityComparer<T> generic interface.
✦ ▣ 🟩	HashSet<T>	Represents a set of values.
✦	KeyedByTypeCollection<TItem>	Provides a collection whose items are types that serve as keys.
✦ ▣ 🟩	KeyNotFoundException	The exception that is thrown when the key specified for accessing an element in a collection does not match any key in the collection.
✦ ▣ 🟩	LinkedList<T>	Represents a doubly linked list.
✦ ▣ 🟩	LinkedListNode<T>	Represents a node in a LinkedList<T>. This class cannot be inherited.
✦ ▣ 🟩	List<T>	Represents a strongly typed list of objects that can be accessed by index. Provides methods to search, sort, and manipulate lists.

Distribute Everywhere

Murdoch UNIVERSITY

Available on the App Store

ANDROID APP ON Google play

amazon apps Available at amazon

Available as a nook app

Windows Store

Windows Phone

Xamarin for Visual Studio

Xamarin.iOS

Xamarin Studio

Xamarin.Android

Xamarin Test Cloud

Xamarin.Mac

Component Store

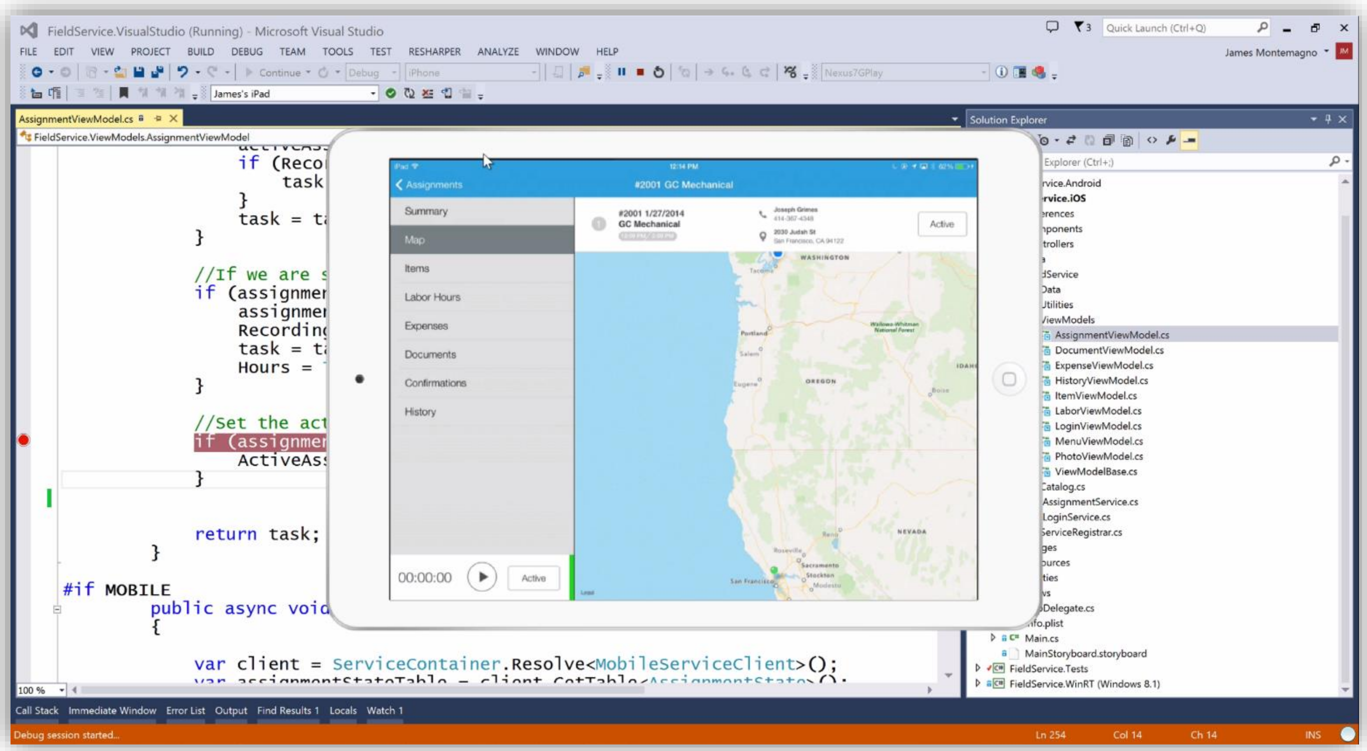
.NET Mobility Scanner

Development Environment

Xamarin Studio
PC or Mac

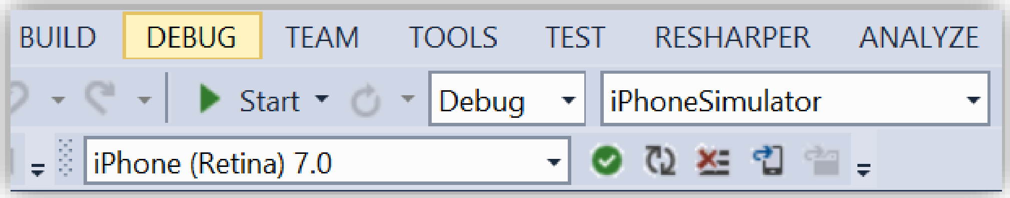
Visual Studio Plugin
VS 2010/2012/2013

Visual Studio Integration

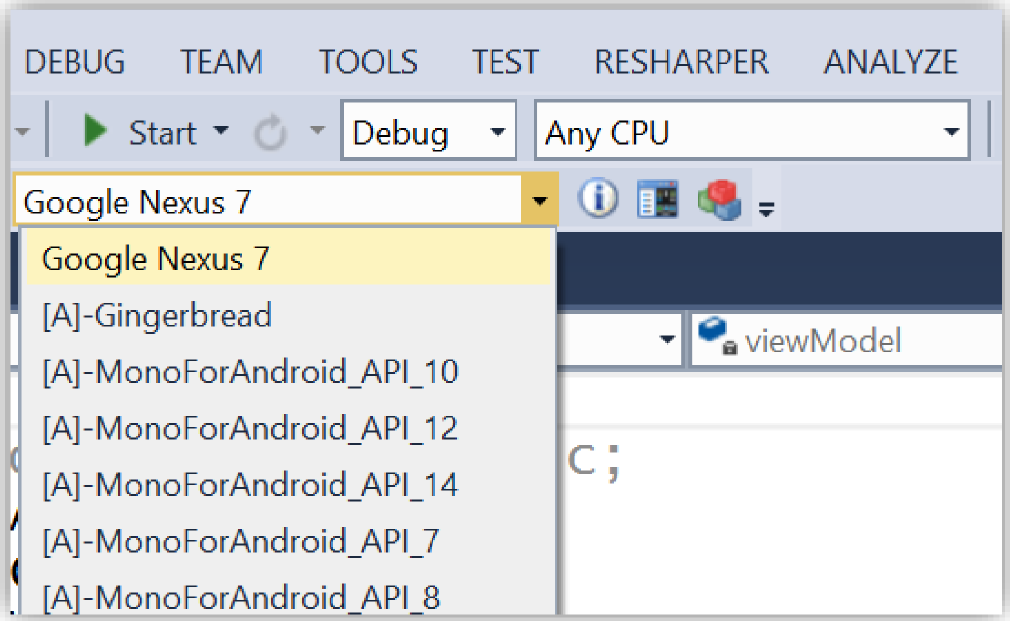


```
                if (Reco
                    task
                }
                task = ta
            }

            //If we are s
            if (assignmen
                assignmen
                Recording
                task = ta
                Hours = 
            }

            //Set the act
            if (assignmen
                ActiveAss
            }


            return task;
        }

#if MOBILE
        public async void
        {

            var client = ServiceContainer.Resolve<MobileServiceClient>();
```

Visual Studio Integration



Debug
iPhoneSimulator
iPhone (Retina) 7.0



Debug
Any CPU
Google Nexus 7

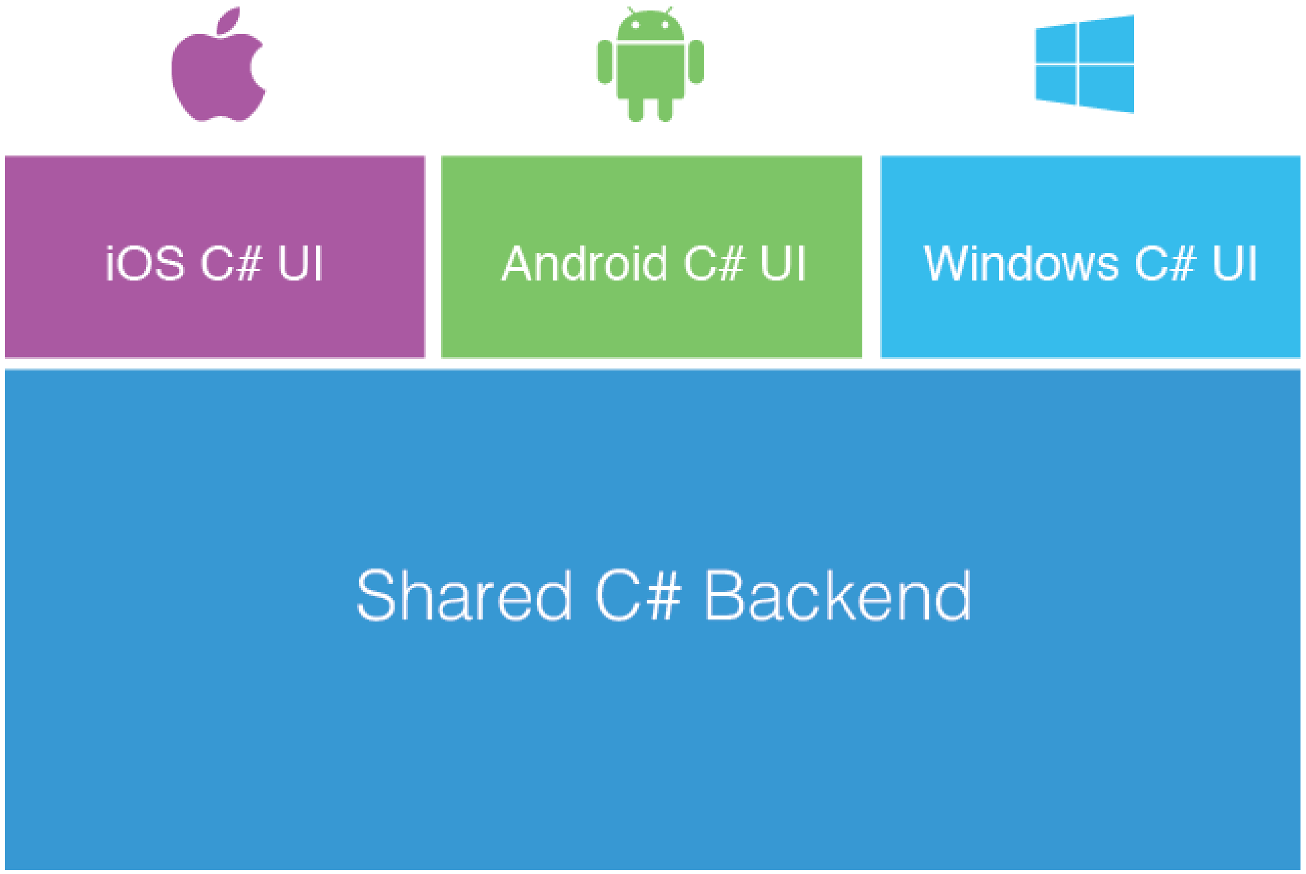
Google Nexus 7
[A]-Gingerbread
[A]-MonoForAndroid_API_10
[A]-MonoForAndroid_API_12
[A]-MonoForAndroid_API_14
[A]-MonoForAndroid_API_7
[A]-MonoForAndroid_API_8

Debug to:
- Emulators
- Devices

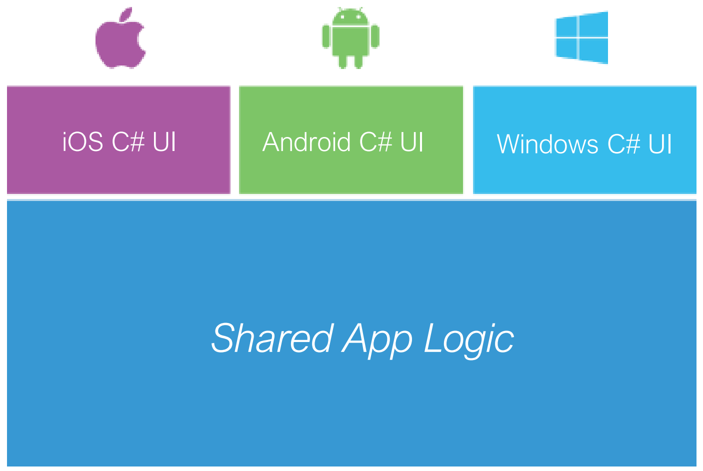
Integrated into toolbar
- Status
- Logs
- List of devices

Just Click Start Debugging!

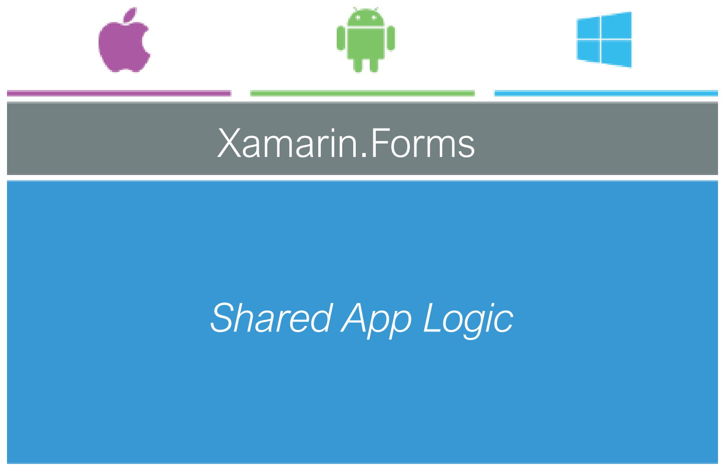
Xamarin Approach



Xamarin Approach



Traditional



Xamarin.Forms

Xamarin

Benefits

Re-use .NET skills

Leverage existing .NET technology

- JSON.NET
- OAUTH.NET
- SignalR

High code re-use 80+%

Tailor UI/UX to target

Negatives

Need to still test and debug multiple targets

Multiple codebase for UI

No sharing of UI

Vendor risk and lock in although Xamarin is a strategic partner for MS

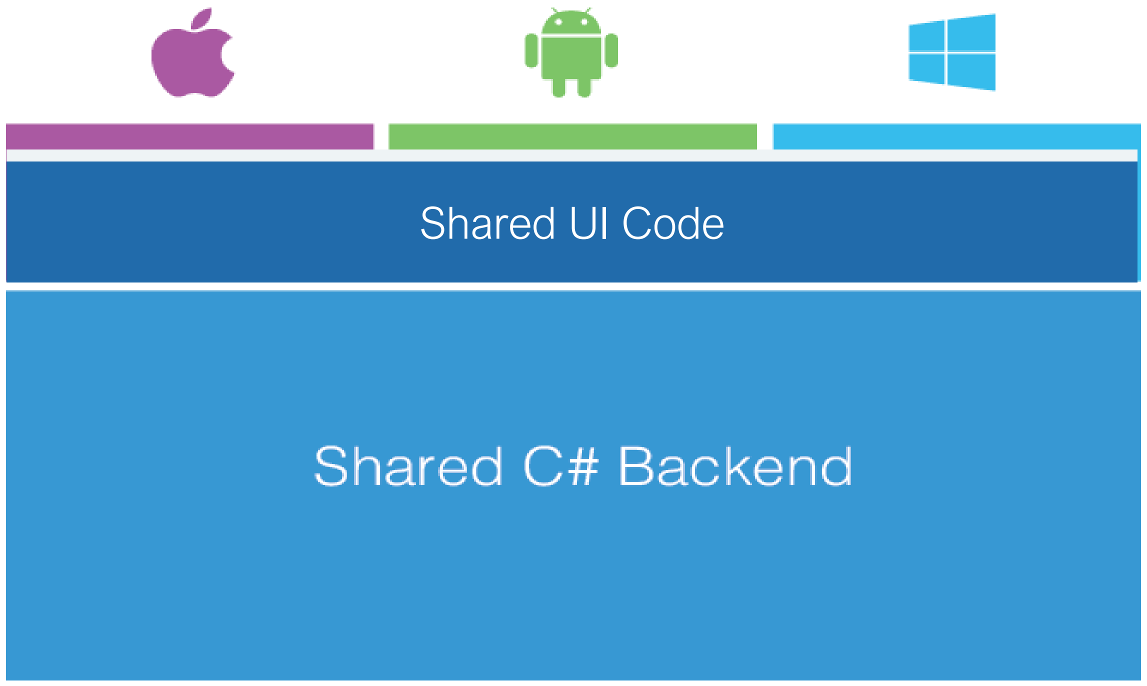
May have to wait on new targets like Android

Xamarin + Xamarin.Forms

Quickly and easily build native user interfaces using shared code

Xamarin.Forms elements map to native controls and behaviors

Mix-and-match Xamarin.Forms with native APIs



xamarin forms

Sharing the User Interface

→ Define once

→ Run on supported platforms

Quick prototyping

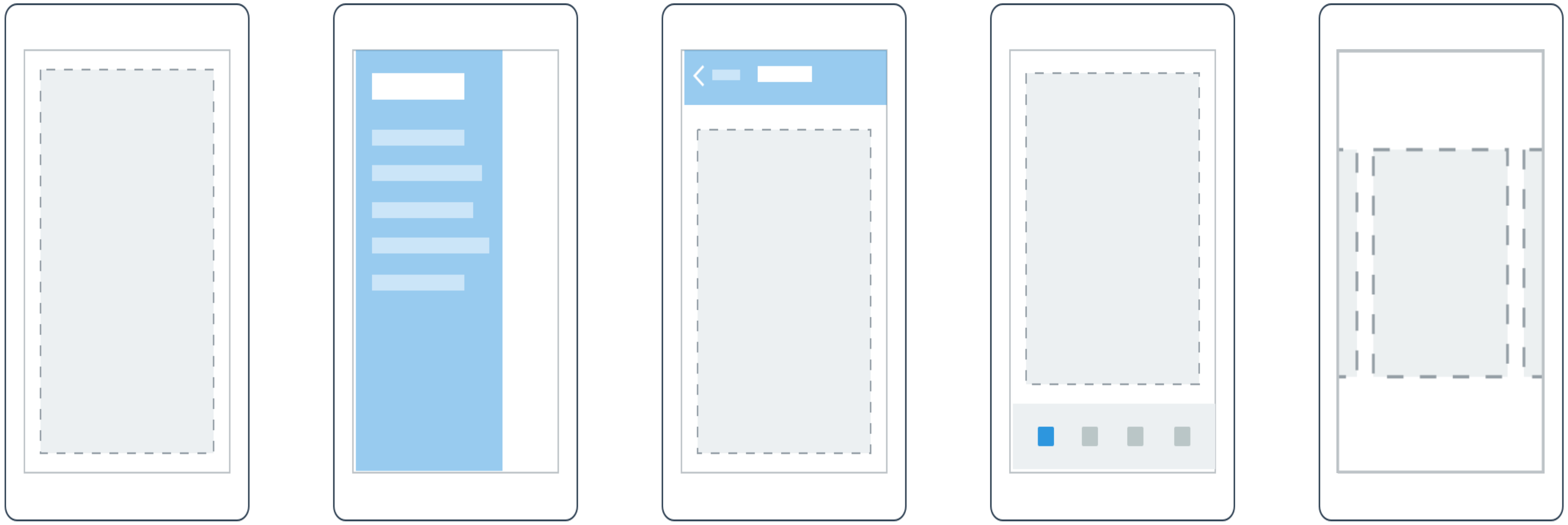
→ Try quickly how the UI works

Evolve your application

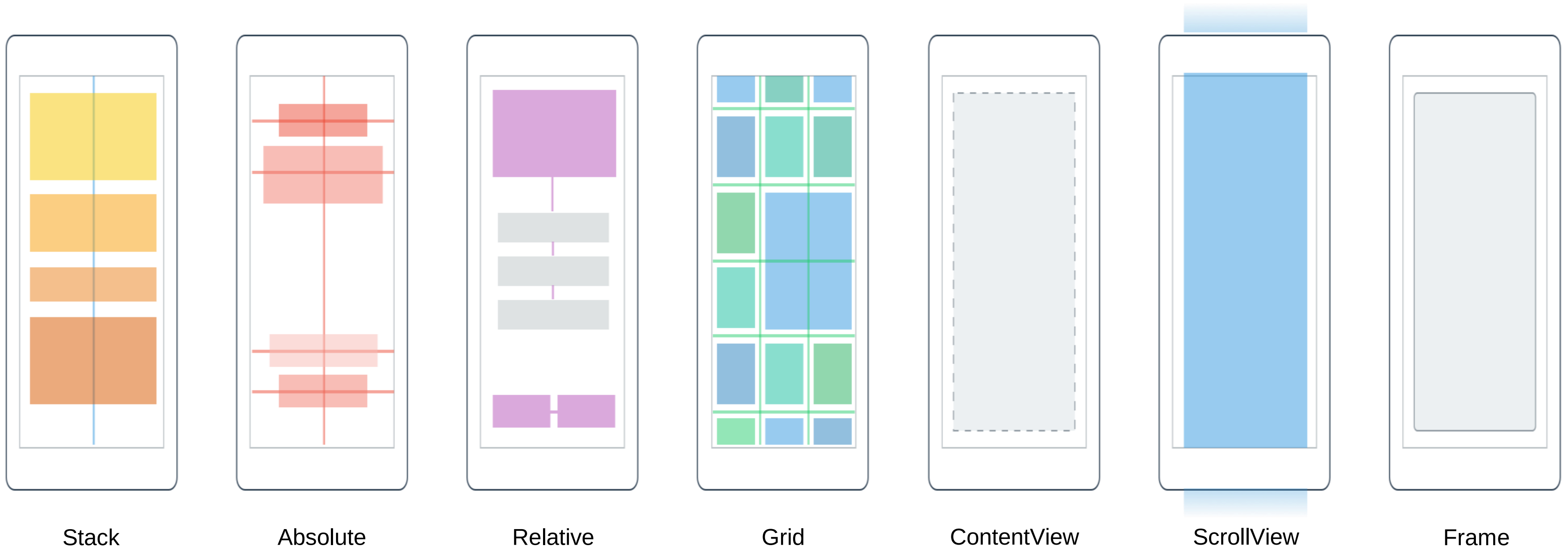
→ Start in Forms

→ Adapt parts to specific platforms

Pages

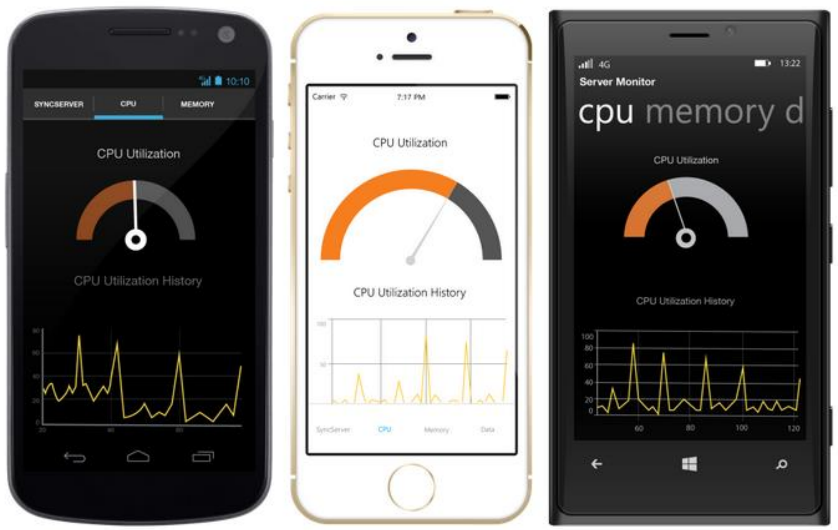


Layouts

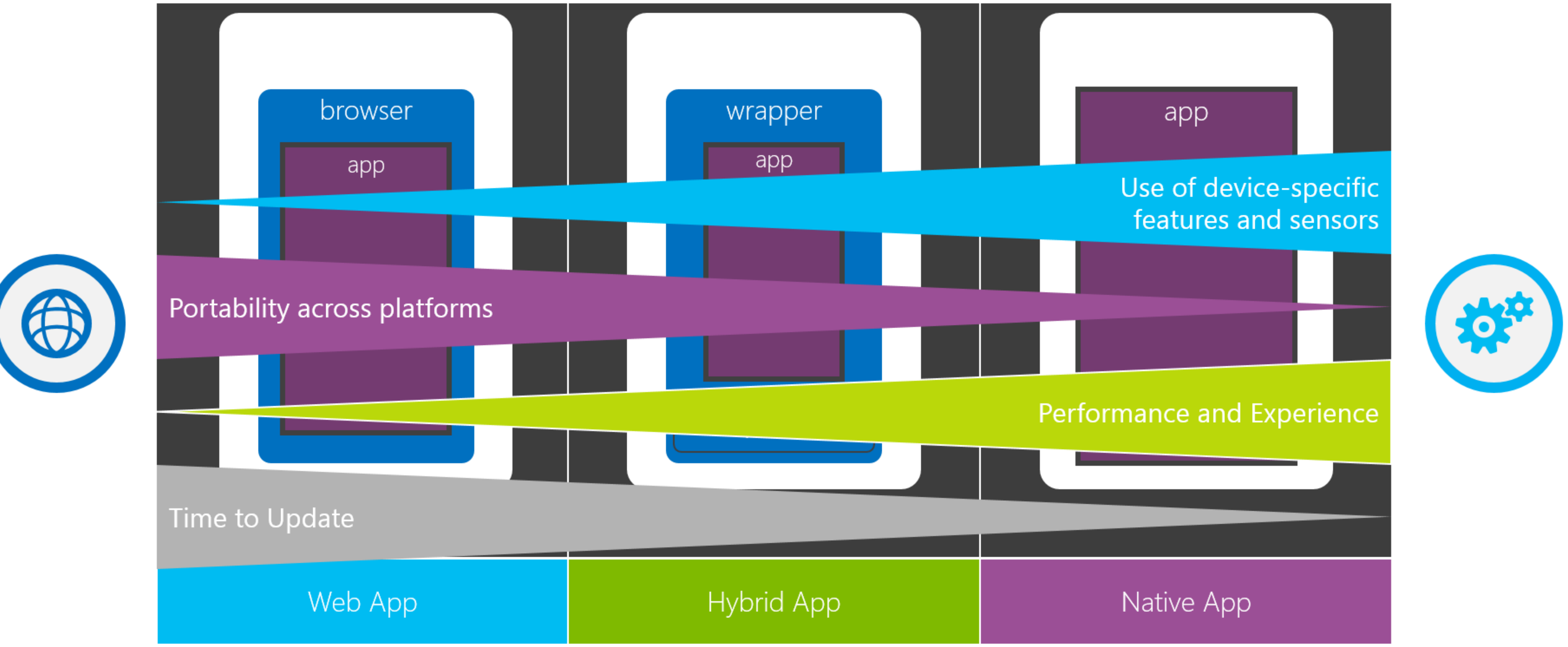

Stack
Absolute
Relative
Grid
ContentView
ScrollView
Frame

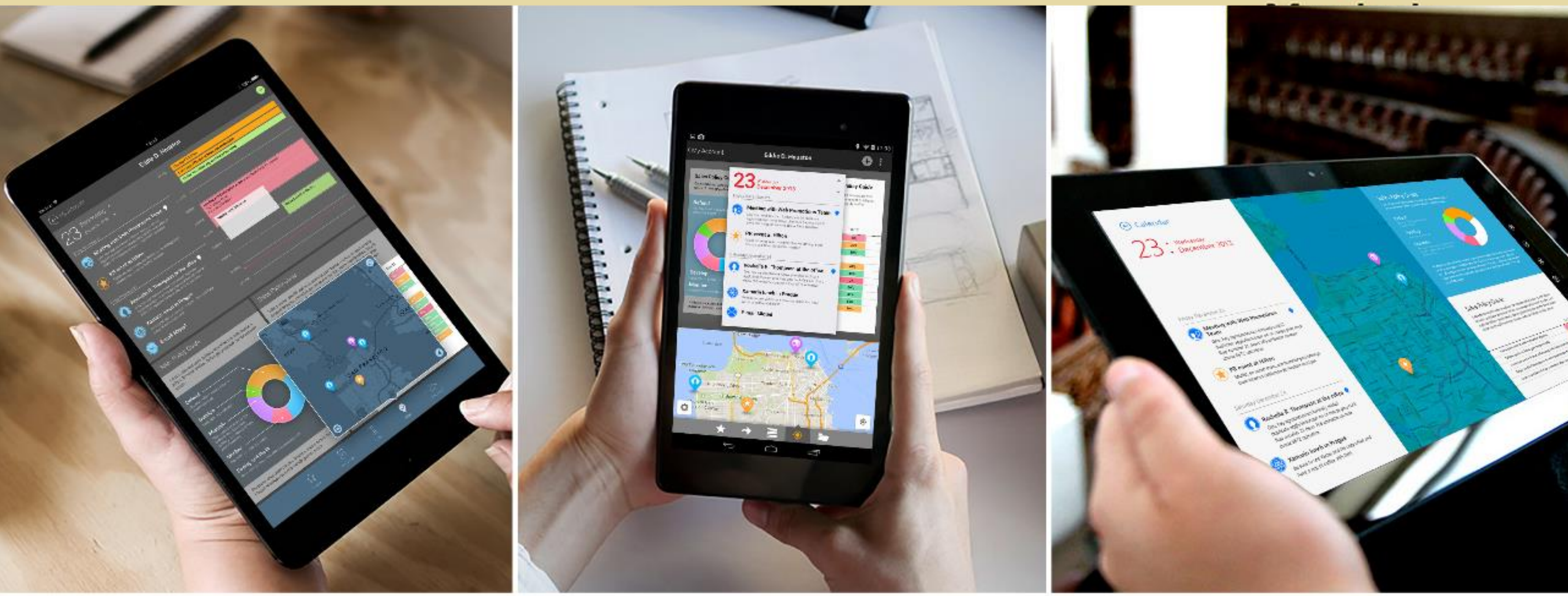
Controls

ActivityIndicator	BoxView	Button	DatePicker	Editor
Entry	Image	Label	ListView	Map
OpenGLView	Picker	ProgressBar	SearchBar	Slider
Stepper	TableView	TimePicker	WebView	EntryCell
ImageCell	SwitchCell	TextCell	ViewCell	



Client Technology Choices


browser
app
wrapper
app
app
Use of device-specific features and sensors
Portability across platforms
Performance and Experience
Time to Update
Web App
Hybrid App
Native App

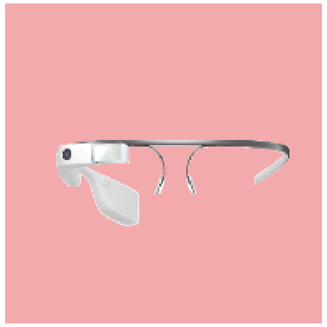


Xamarin exposes 100% of the native APIs for iOS, Android and Windows

Support Native APIs

Same day support:

iOS 5, iOS 6, iOS 7, iOS 7.1, iOS 8



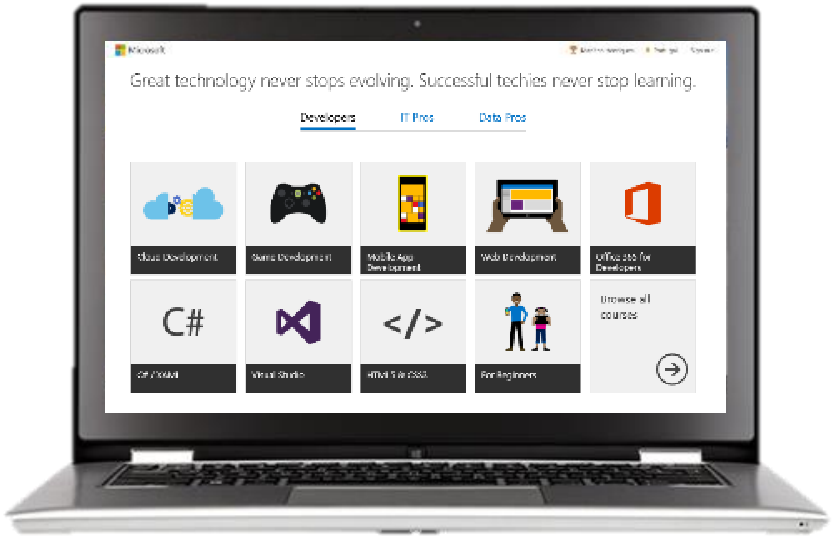
Also:

- Google Glass
- Android Wear
- Amazon Fire TV
- Outros…

MVA

Microsoft Virtual Academy

www.microsoftvirtualacademy.com



Cross-Platform Development with Xamarin & Visual Studio

Cross-Platform Development with Visual Studio

Xamarin Test Cloud

Testing on hundreds of real devices

Scripting the tests

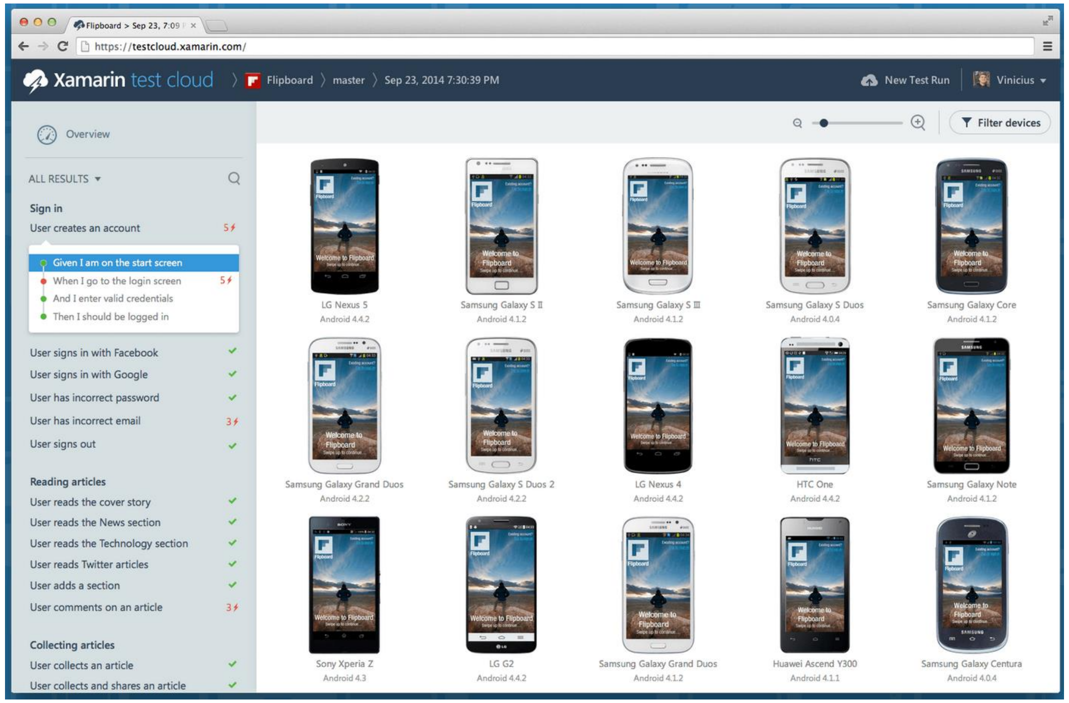
Getting feedback

→ Screenshots (scripted)

→ Crash information

→ Call stack

→ …



https://testcloud.xamarin.com/

Xamarin test cloud 〉 Flipboard 〉 master 〉 Sep 23, 2014 7:30:39 PM

New Test Run | Vinicius

Overview

ALL RESULTS

Sign in
User creates an account 5
Given I am on the start screen
When I go to the login screen 5
And I enter valid credentials
Then I should be logged in
User signs in with Facebook ✓
User signs in with Google ✓
User has incorrect password ✓
User has incorrect email 3
User signs out ✓

Reading articles
User reads the cover story ✓
User reads the News section ✓
User reads the Technology section ✓
User reads Twitter articles ✓
User adds a section ✓
User comments on an article 3

Collecting articles
User collects an article ✓
User collects and shares an article ✓

Filter devices

LG Nexus 5, Android 4.4.2	Samsung Galaxy S II, Android 4.1.2	Samsung Galaxy S III, Android 4.1.2	Samsung Galaxy S Duos, Android 4.0.4	Samsung Galaxy Core, Android 4.1.2
Samsung Galaxy Grand Duos, Android 4.2.2	Samsung Galaxy S Duos 2, Android 4.2.2	LG Nexus 4, Android 4.4.2	HTC One, Android 4.4.2	Samsung Galaxy Note, Android 4.1.2
Sony Xperia Z, Android 4.3	LG G2, Android 4.4.2	Samsung Galaxy Grand Duos, Android 4.1.2	Huawei Ascend Y300, Android 4.1.1	Samsung Galaxy Centura, Android 4.0.4

http://blog.xamarin.com/new-xamarin-test-cloud-features-2/

How does Xamarin work?

- The apps created with Xamarin are compiled into native-code

  For iOS and OS X, Xamarin translates all the .NET code into Objective-C and C code

    This must happen on a MAC machine

  For Android it creates a set of bridges that are installed on the Android device

    Something like a CLR inside an Android device

  For Windows Phone, it just compiles to a Windows Phone app

Installing Xamarin

- Installing Xamarin for all OSes is pretty easy:

  Just download Xamarin Platform Latest:

    http://xamarin.com/platform#download

  This will set up almost everything needed to create apps

    Downloads the necessary JRE, JDK and ADK for Android

    Installs Xamarin Studio and Plugins for Visual Studio, if missing

Installing Xamarin

- For Windows Phone and iOS there are some additional things to do:
  - Install Windows Phone SDK on Windows
    - Windows Phone apps can be created only with Visual Studio
  - Install Xcode on OS X
    - iOS apps can be created only on MAC machine
- For easier work with Android, install Xamarin Android Player
  - Works with VirtualBox

Xamarin.Forms

- Xamarin.Forms is the common UI of Xamarin

  Share ~90% of the code base for each platform

    Only the rest 10% are concrete for the platform

- How to create and run a Xamarin.Forms app?

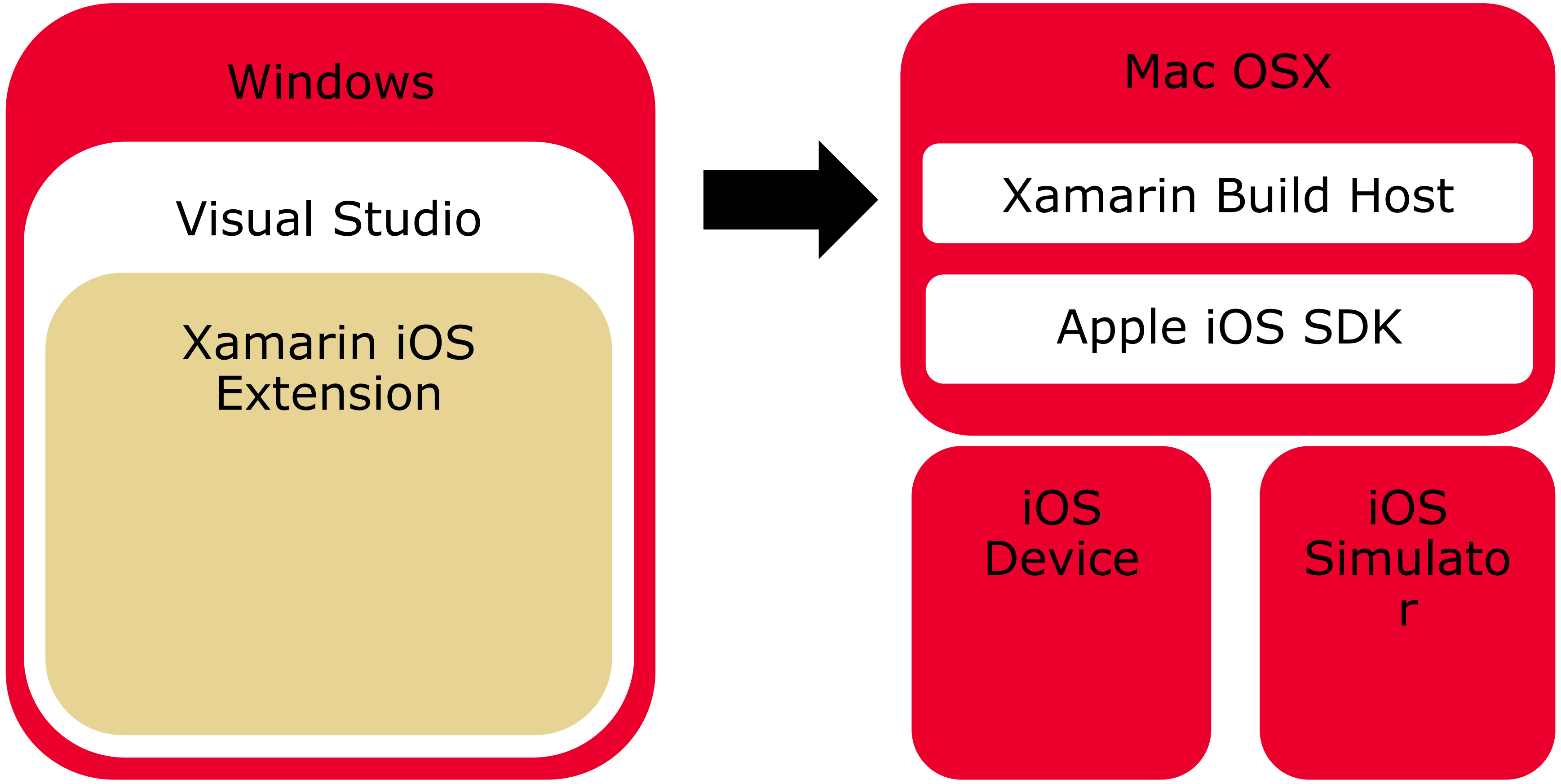
  Create a new project, located at:

    C# -> Mobile Apps -> Blank App

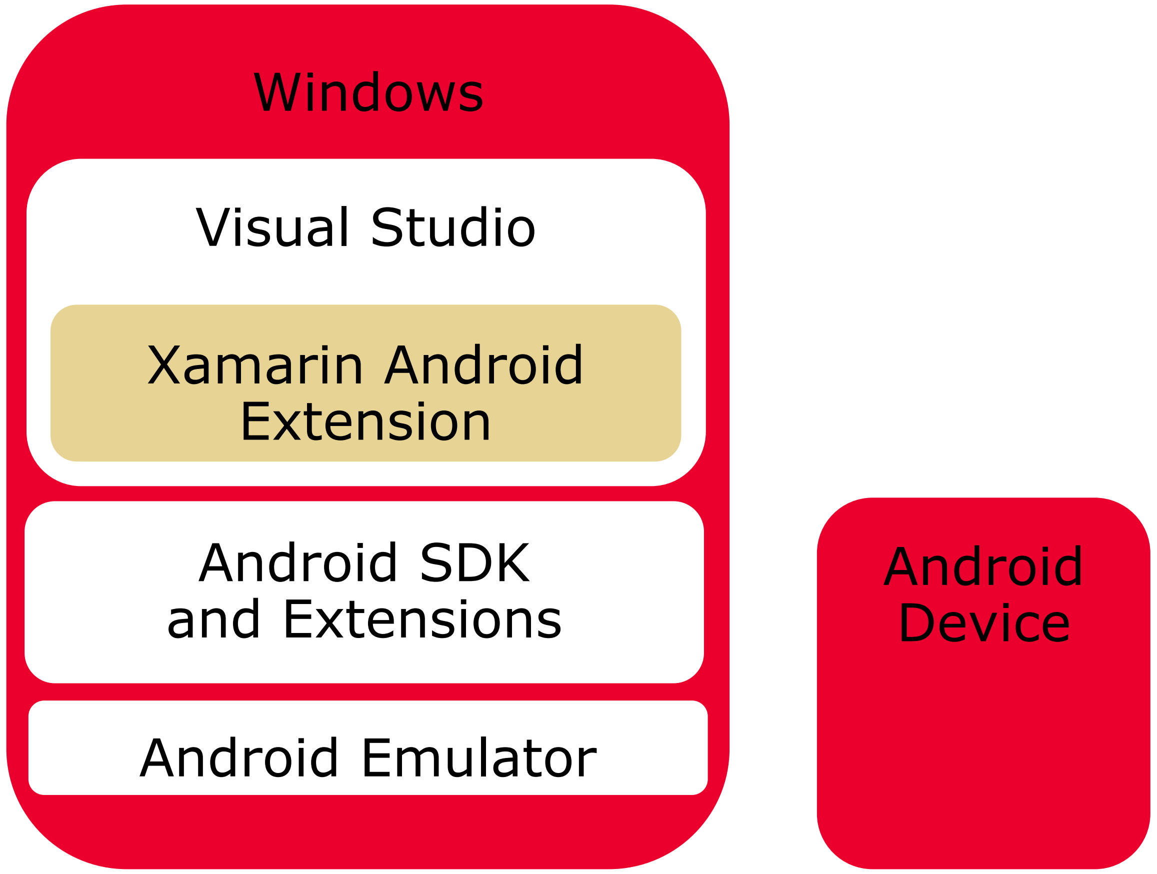
  Write code, select the wanted project as "Startup project"

  Run in the simulator

iOS Development setup

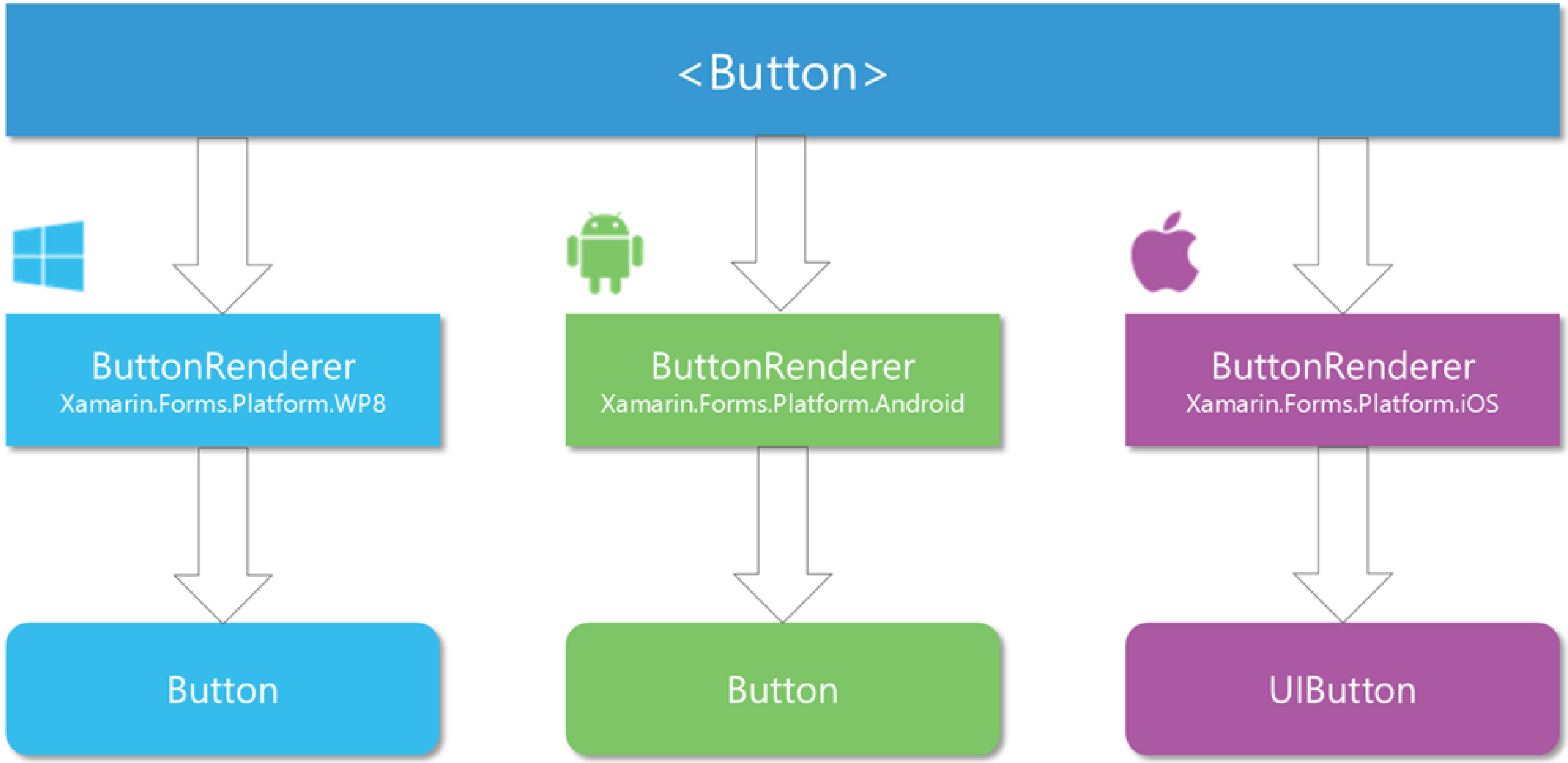


Android Development setup

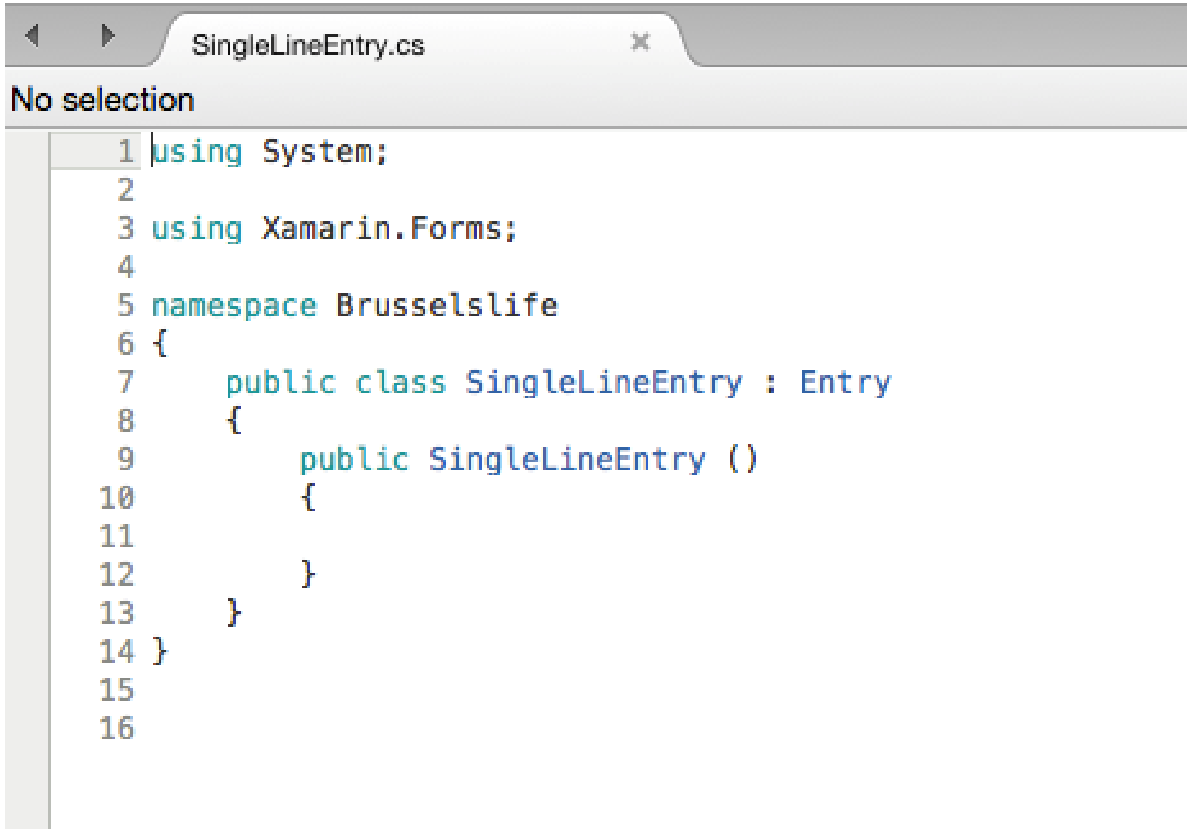


Xamarin forms

How Form controls works ?

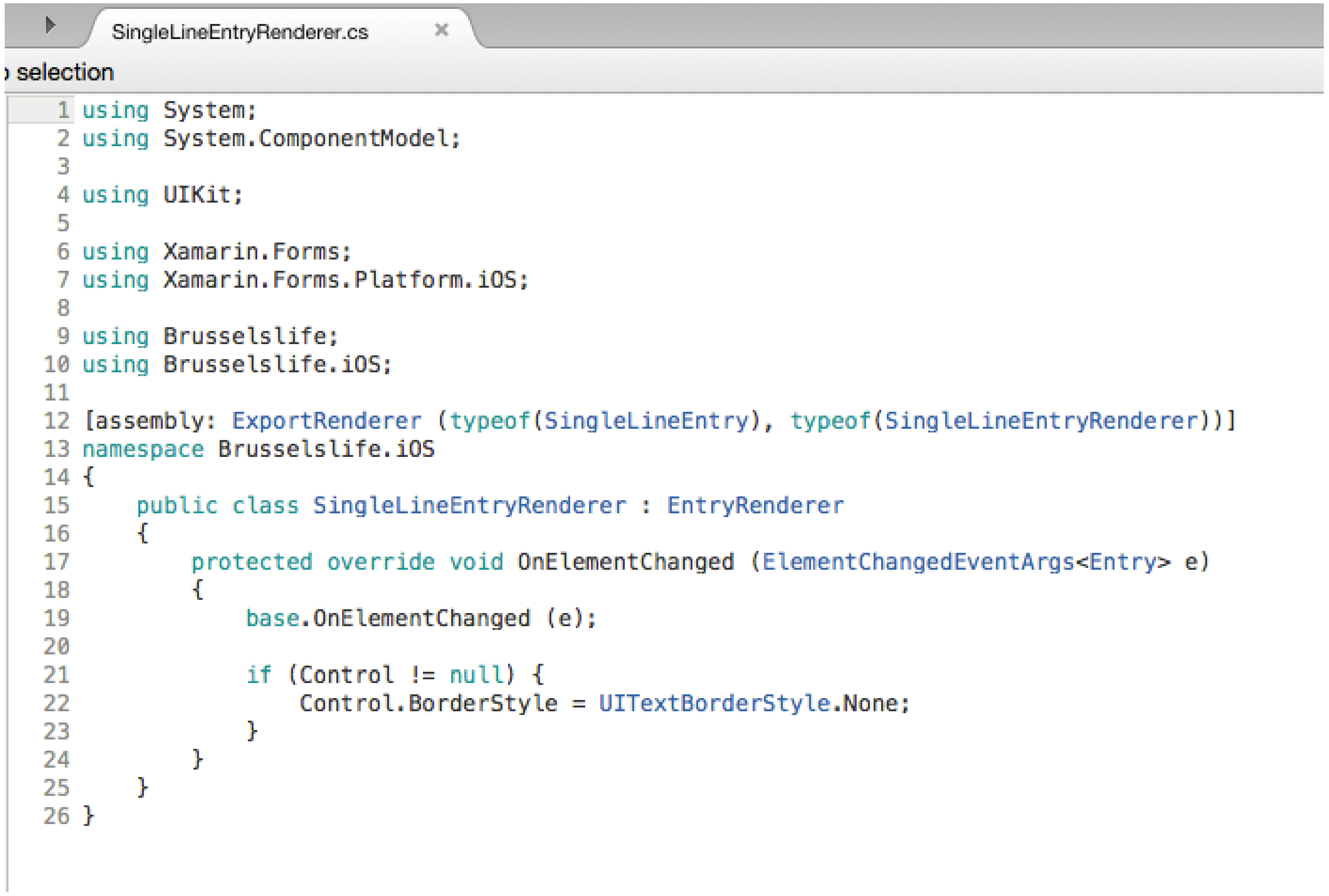

<Button>
ButtonRenderer
Xamarin.Forms.Platform.WP8
ButtonRenderer
Xamarin.Forms.Platform.Android
ButtonRenderer
Xamarin.Forms.Platform.iOS
Button
Button
UIButton

Forms control code



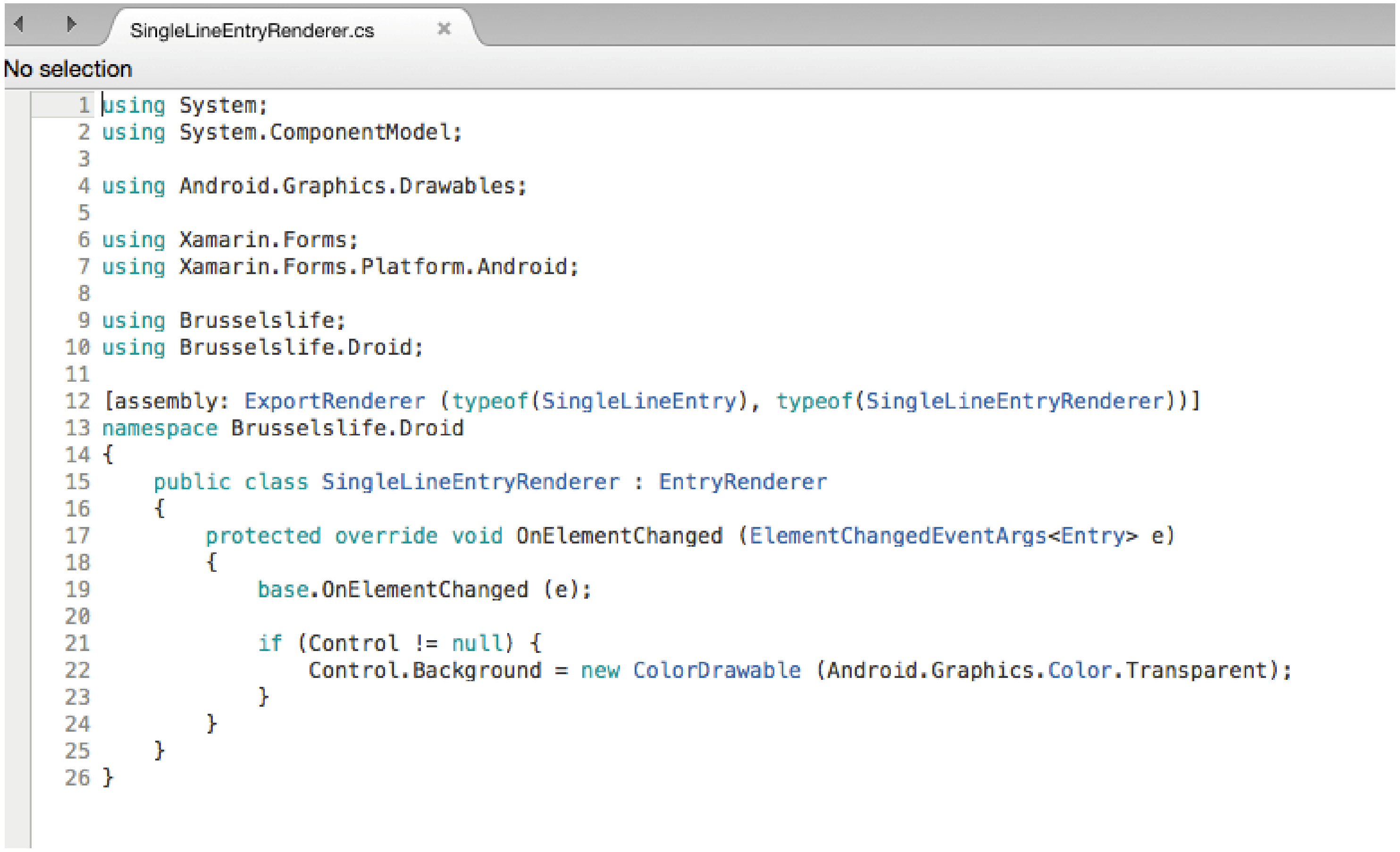
```
using System;

using Xamarin.Forms;

namespace Brusselslife
{
    public class SingleLineEntry : Entry
    {
        public SingleLineEntry ()
        {

        }
    }
}


```

iOS control code



```
using System;
using System.ComponentModel;

using UIKit;

using Xamarin.Forms;
using Xamarin.Forms.Platform.iOS;

using Brusselslife;
using Brusselslife.iOS;

[assembly: ExportRenderer (typeof(SingleLineEntry), typeof(SingleLineEntryRenderer))]
namespace Brusselslife.iOS
{
    public class SingleLineEntryRenderer : EntryRenderer
    {
        protected override void OnElementChanged (ElementChangedEventArgs<Entry> e)
        {
            base.OnElementChanged (e);

            if (Control != null) {
                Control.BorderStyle = UITextBorderStyle.None;
            }
        }
    }
}
```

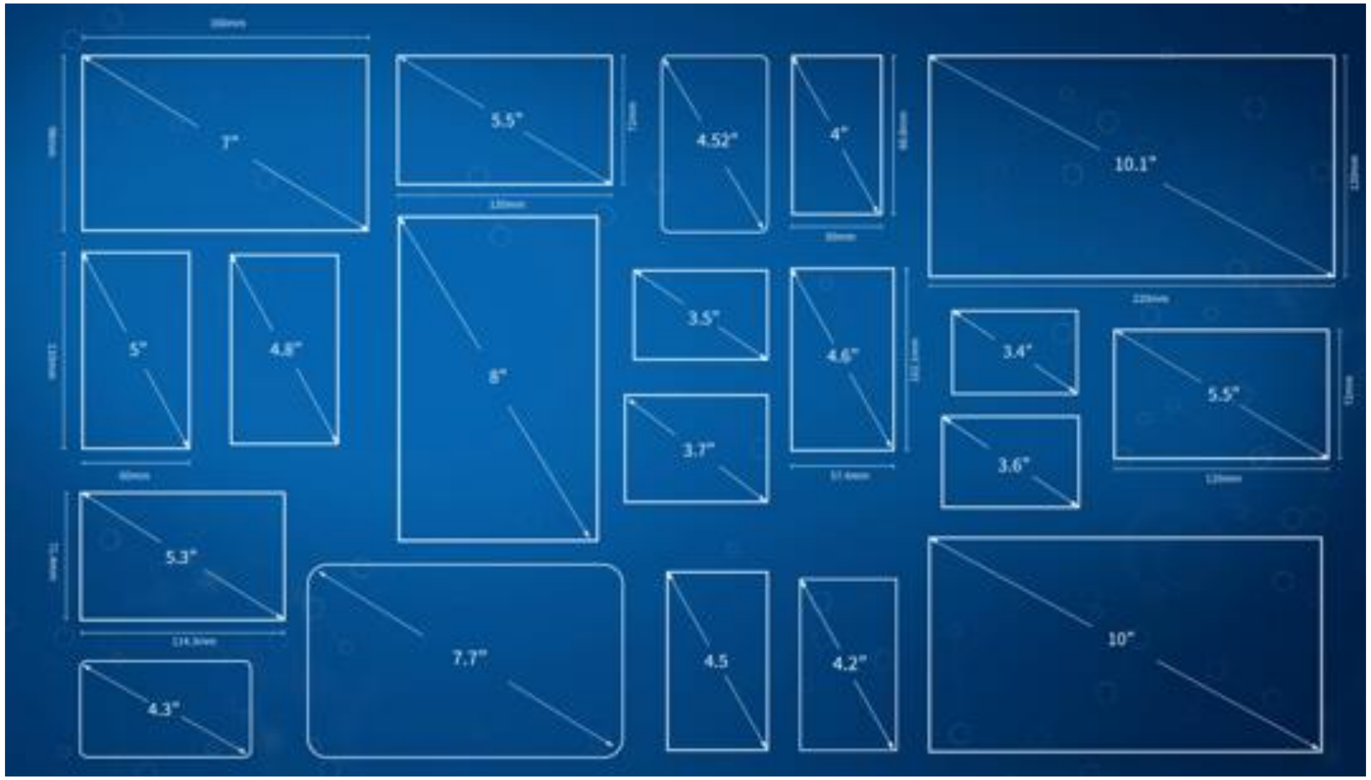
android control code



```
using System;
using System.ComponentModel;

using Android.Graphics.Drawables;

using Xamarin.Forms;
using Xamarin.Forms.Platform.Android;

using Brusselslife;
using Brusselslife.Droid;

[assembly: ExportRenderer (typeof(SingleLineEntry), typeof(SingleLineEntryRenderer))]
namespace Brusselslife.Droid
{
    public class SingleLineEntryRenderer : EntryRenderer
    {
        protected override void OnElementChanged (ElementChangedEventArgs<Entry> e)
        {
            base.OnElementChanged (e);

            if (Control != null) {
                Control.Background = new ColorDrawable (Android.Graphics.Color.Transparent);
            }
        }
    }
}
```

Cross-Platform Mobile Development

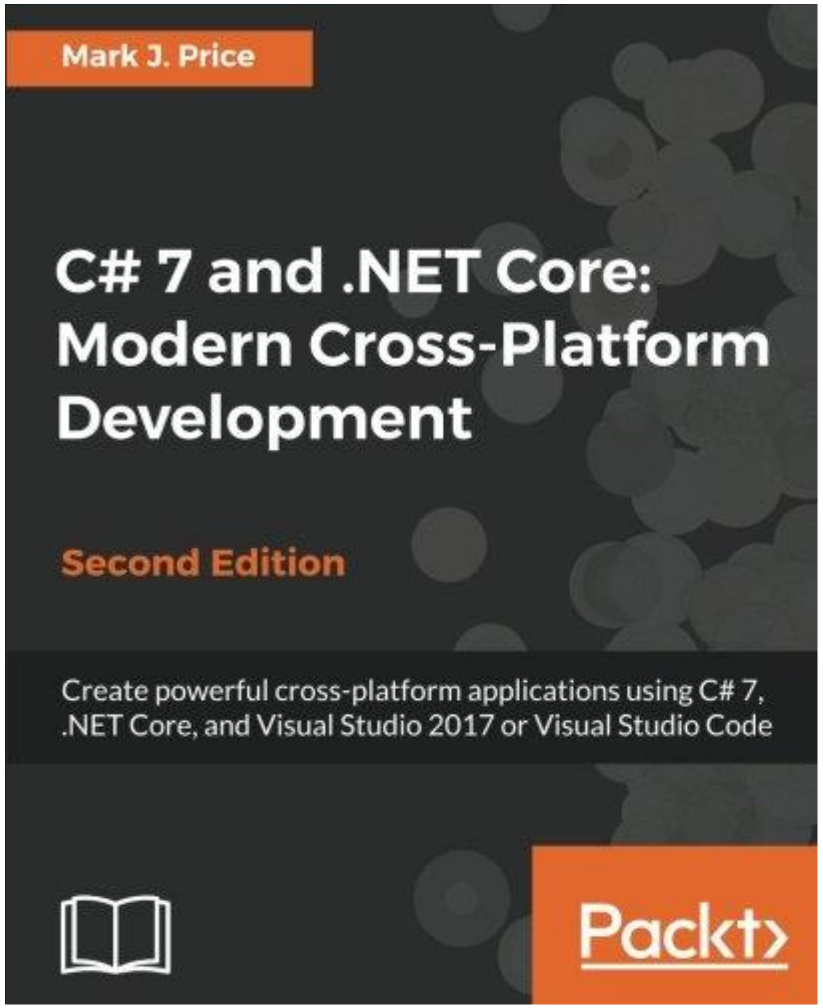
- **Building high-quality Apps is hard:**
  - Different presentation styles, interaction styles and software stacks
  - Devices have different screen sizes, input modes and hardware capabilities
  - New devices and OS versions are introduced multiple times per year
  - Network connectivity and power levels fluctuate widely in typical usage scenarios
  - New consumer applications regularly extend and revise the standards and set the bar higher for good mobile applications



Reading/ reference

http://prospero.murdoch.edu.au/record=b2962782~S1

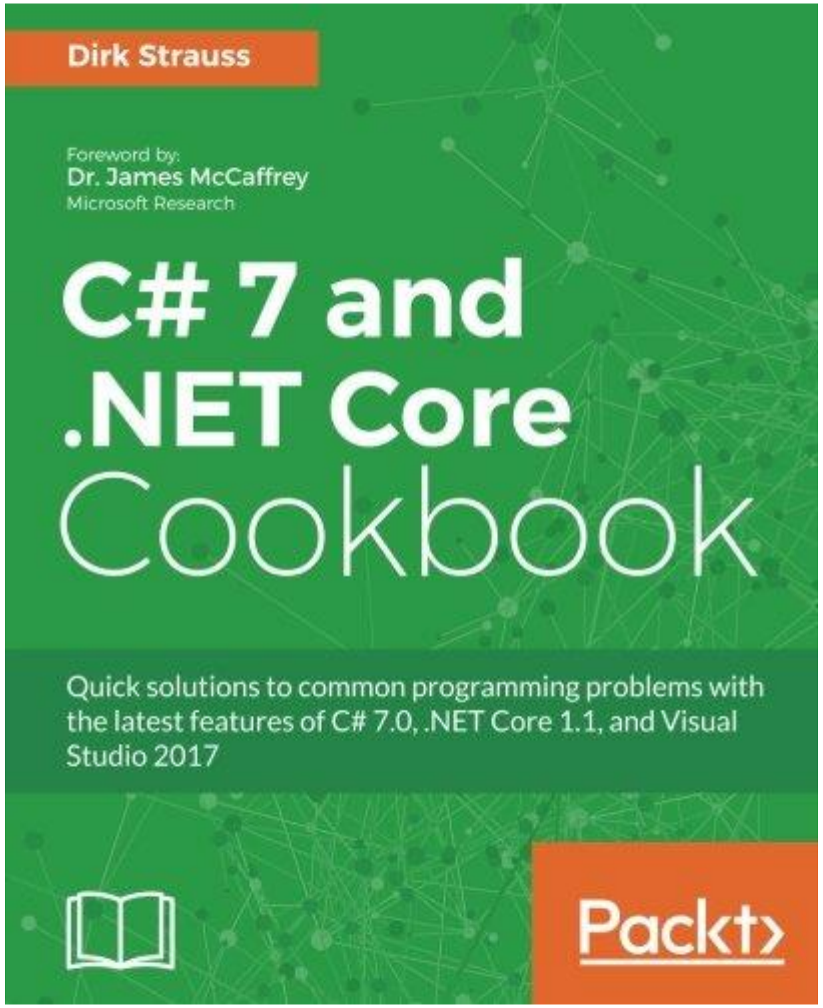
Chapter 15. Building Mobile Apps Using Xamarin.Forms and ASP.NET Core Web API


Mark J. Price
C# 7 and .NET Core: Modern Cross-Platform Development
Second Edition
Create powerful cross-platform applications using C# 7, .NET Core, and Visual Studio 2017 or Visual Studio Code
Packt

Reading/ reference

http://prospero.murdoch.edu.au/record=b2962781~S1

Chapter: CREATING A MOBILE APPLICATION IN VISUAL STUDIO


Dirk Strauss
Foreword by:
Dr. James McCaffrey
Microsoft Research
C# 7 and .NET Core Cookbook
Quick solutions to common programming problems with the latest features of C# 7.0, .NET Core 1.1, and Visual Studio 2017
Packt

MSDN

XAML again

https://docs.microsoft.com/en-us/windows/uwp/get-started/create-a-hello-world-app-xaml-universal

Summary

- Strategies

  Silo – Pure Native

  Mobile Web – Web Apps

- Cross Platform is not theory or option. It is the new reality.
- .NET is a viable platform using MS on MS tech and Xamarin to reach non-MS tech (iOS, Android, Mac, Linux, Google Glass, etc.)