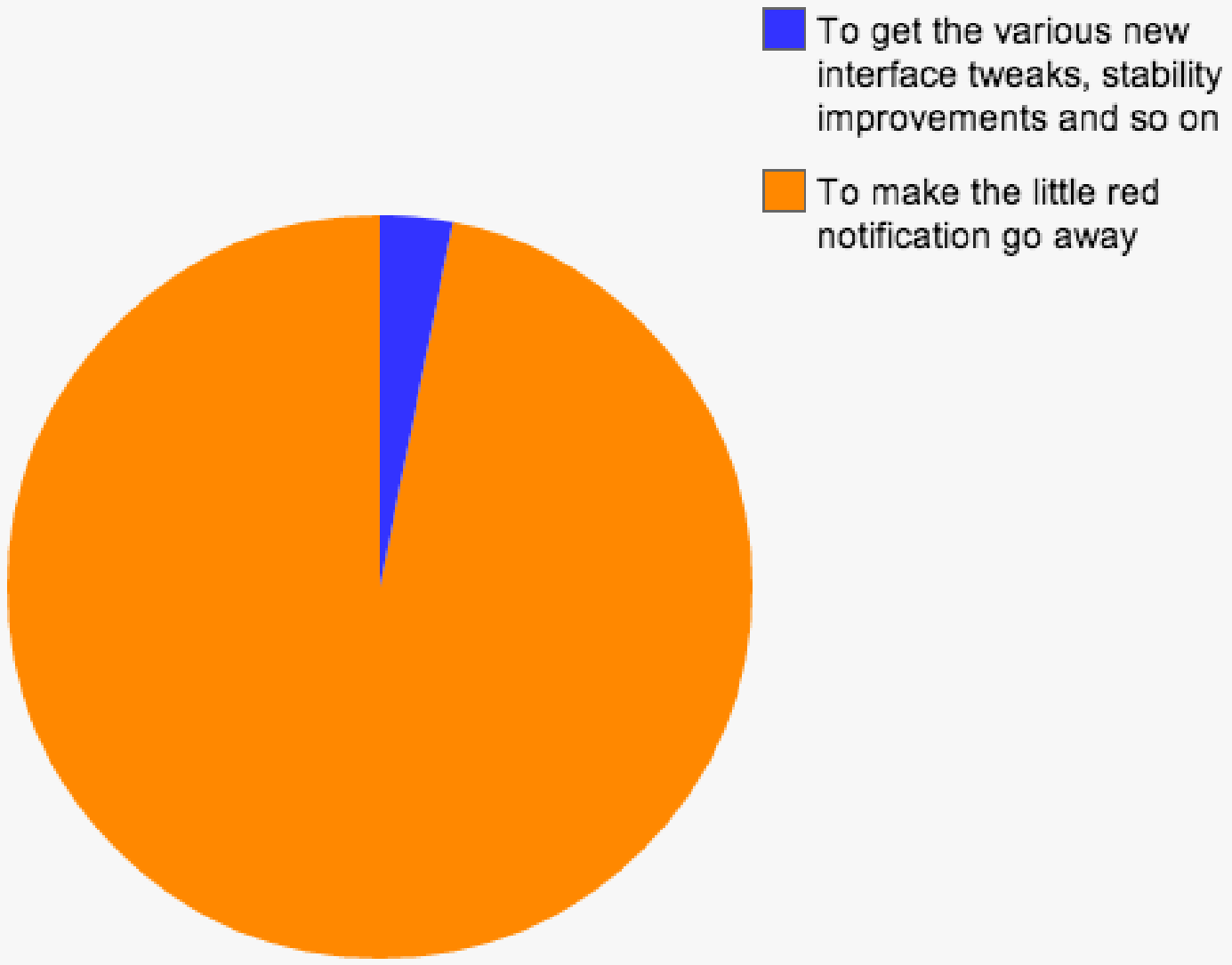
Go X-Platform with

Have your C# and mobile too

Iowa Code Camp – July 19, 2014
@DennisWelu
DennisWelu@MotisConsulting.com

Reasons why I update my apps



Clarke
UNIVERSITY

EAGLE POINT

Dubuque .Net
Users Group

Motis

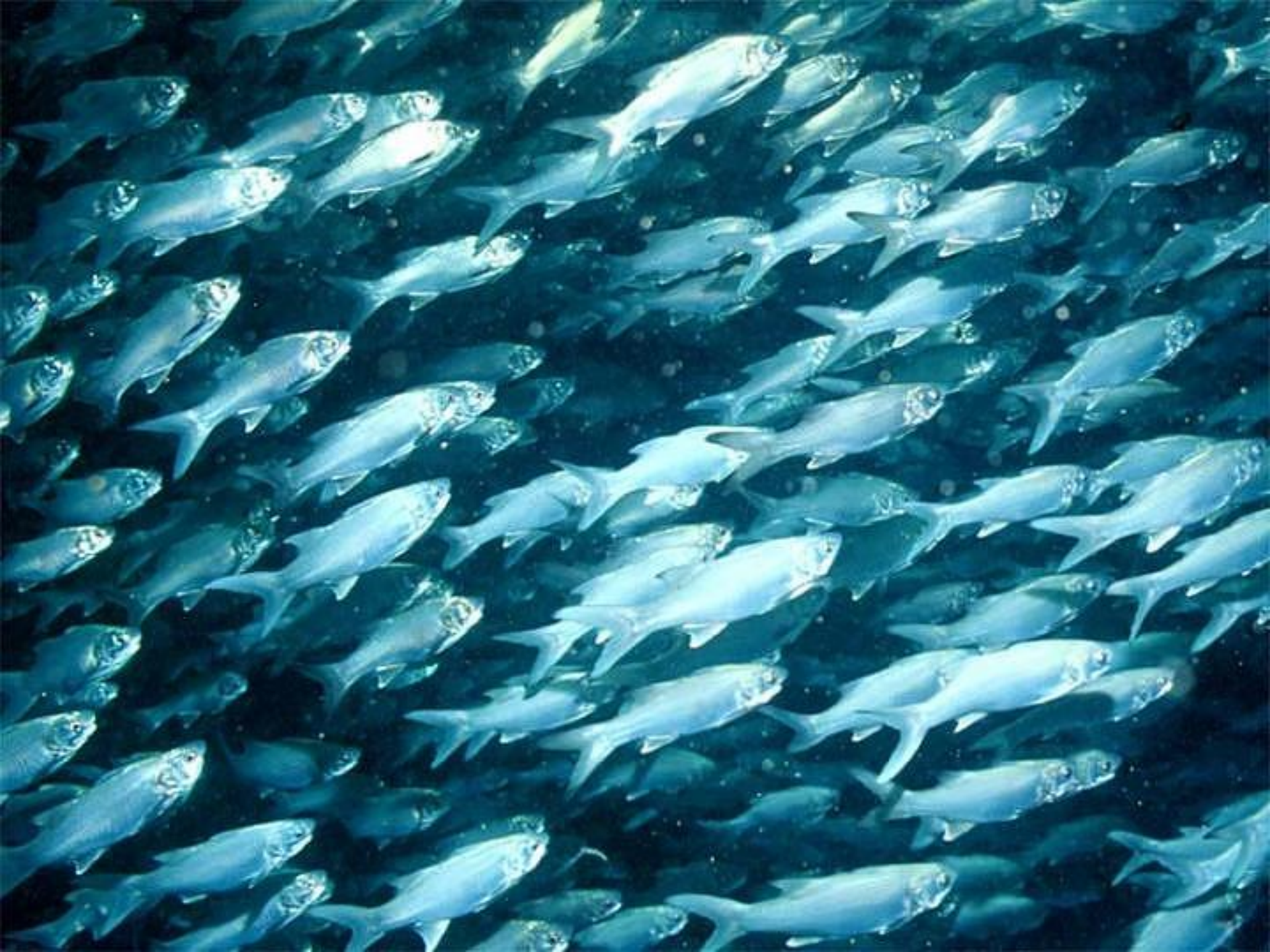
Cartegraph

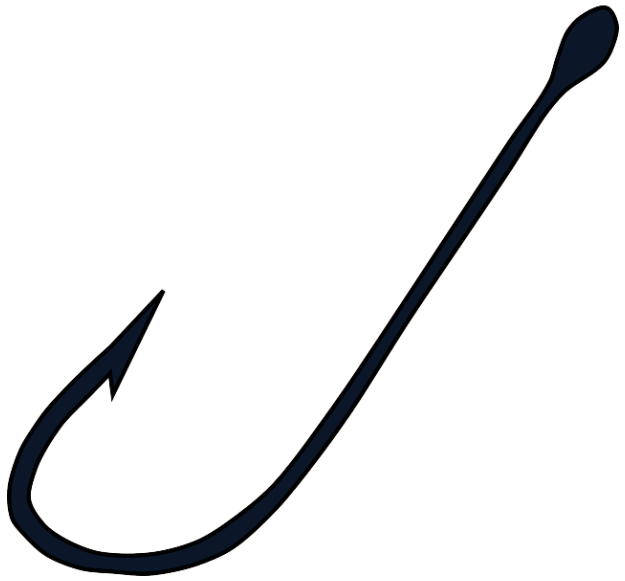
cynergy.

Mood board


Way Cool App

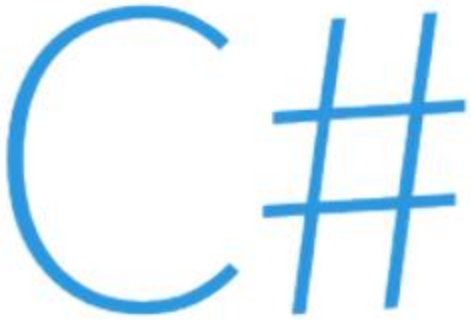

y Cool App





//TODO:

- ❑ **Xamarin**
- ❑ HelloWorld++
- ❑ Architecture
- ❑ Build it bigger
- ❑ Gotchas
- ❑ Resources



on 2.5 Billion Devices

GLASS

kindle fire

amazon fireTV

Lopadotemachoselachogaleokranio-leipsanodrimhypotrimmatosilphioparao-melitokatakechymenokichlepikossypho-phattoperisteralektryonoptekephalliokigklo-peleiolagoiosiraiobaphetraganopterygon

A dish consisting of fish, poultry, other meat.

siebenhundertsiebenundsiebzigtausendsiebenhundertsiebenundsiebzig

777,777

http://en.wikipedia.org/wiki/Longest_words

C# Goodies

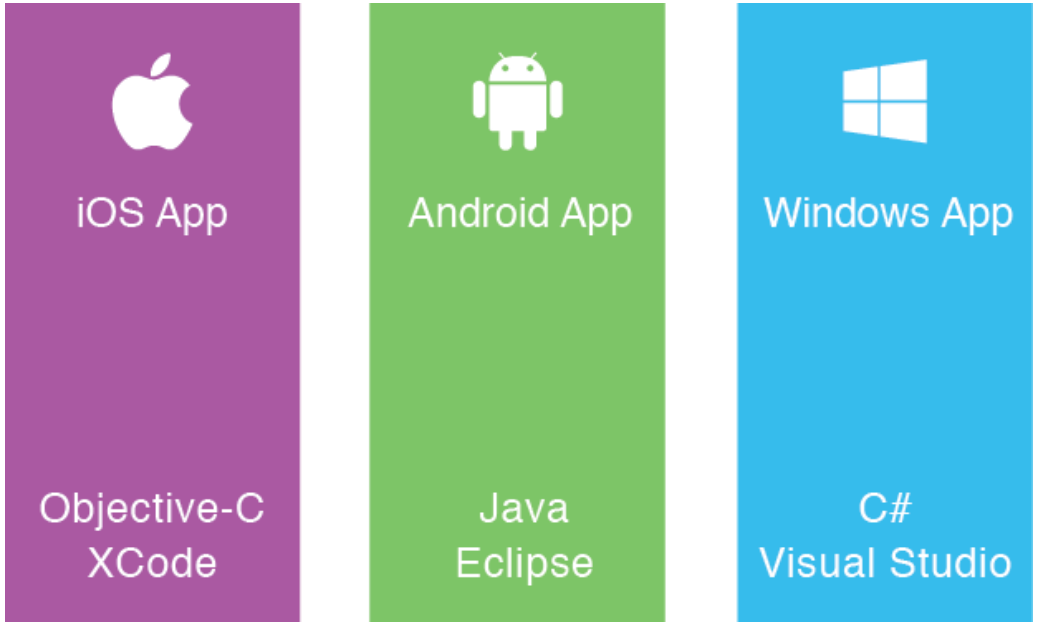
- Linq
- Lambdas
- Events
- Delegates
- Generics
- Async/await
- Etc. etc.

Ecosystem Goodies

- Visual Studio
- NuGet
- Productivity plugins…ReSharper, CodeRush, etc.

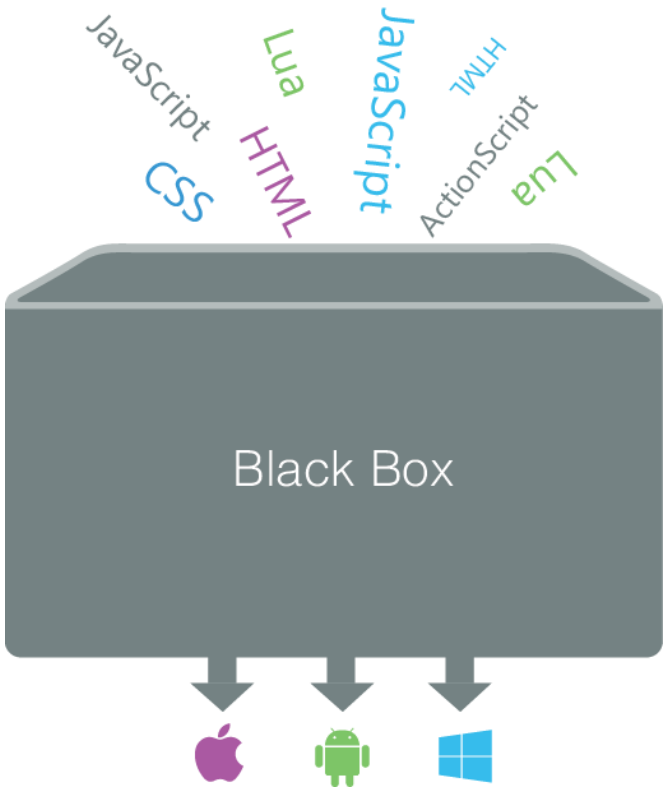
Cross-platform Approach

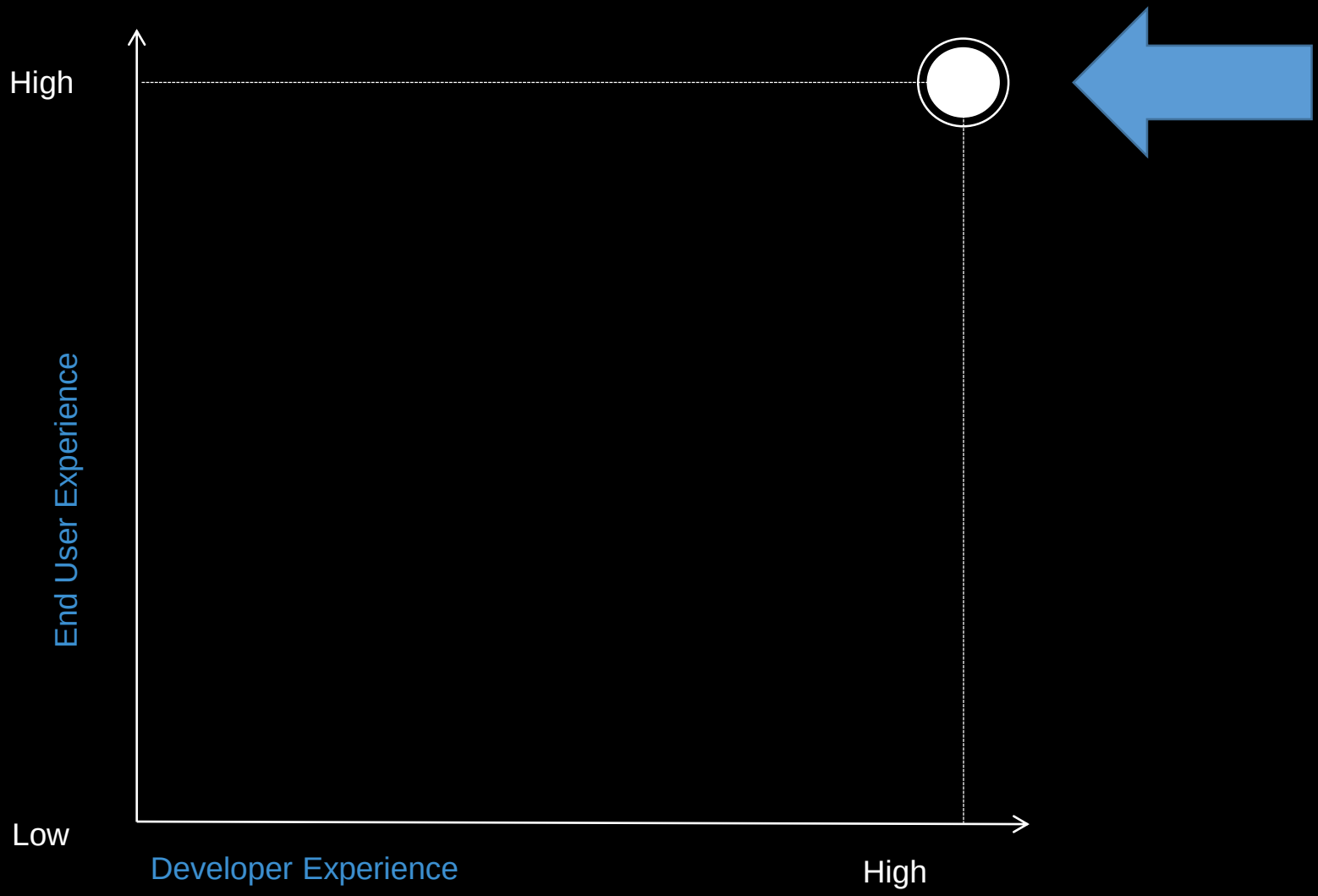
Native Tools Approach



Write-Once-Run-Anywhere Approach

- Ux?
- Device services?
- API coverage?




High
End User Experience
Low
Developer Experience
High

High Fidelity User Experience

Silo'd approach

Objective-C, Java, C#

Single Platform

Multiple Platforms

"Black Box" Approach

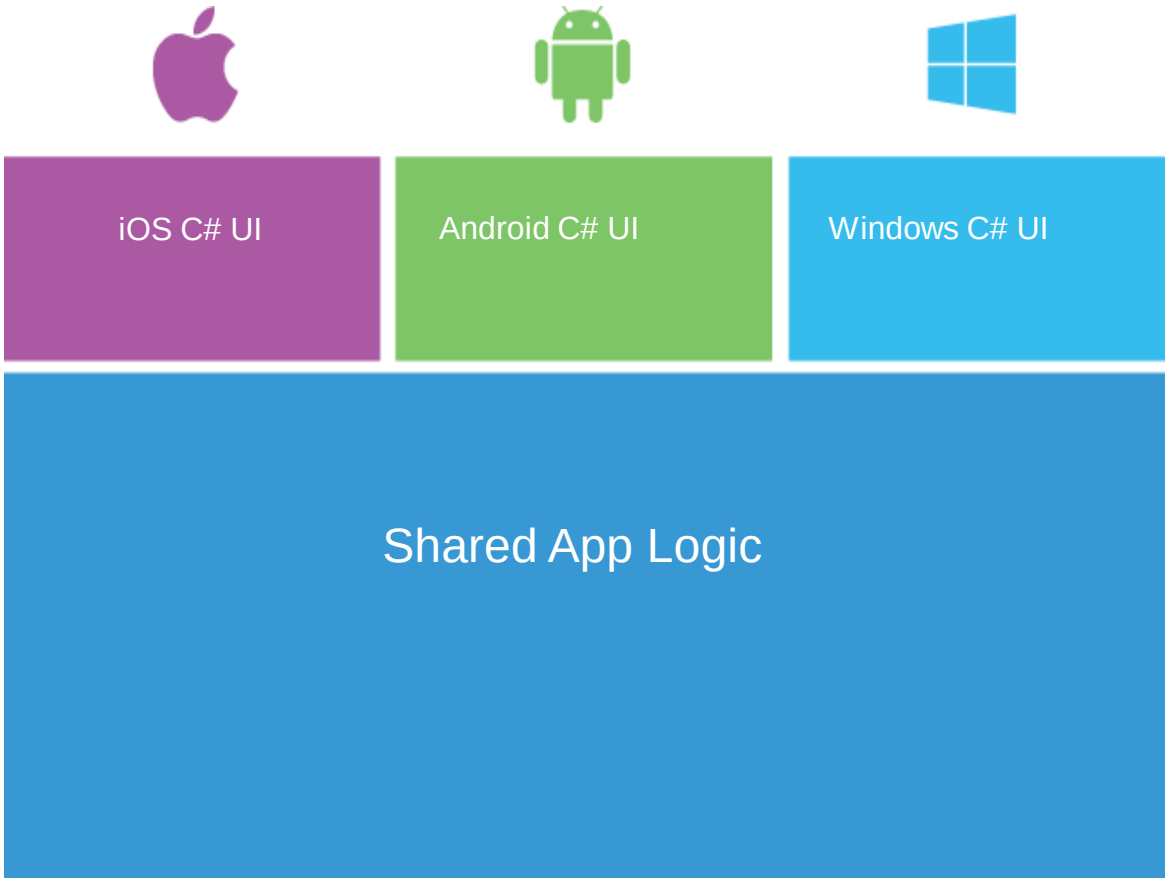
Abstraction frameworks

HTML 5

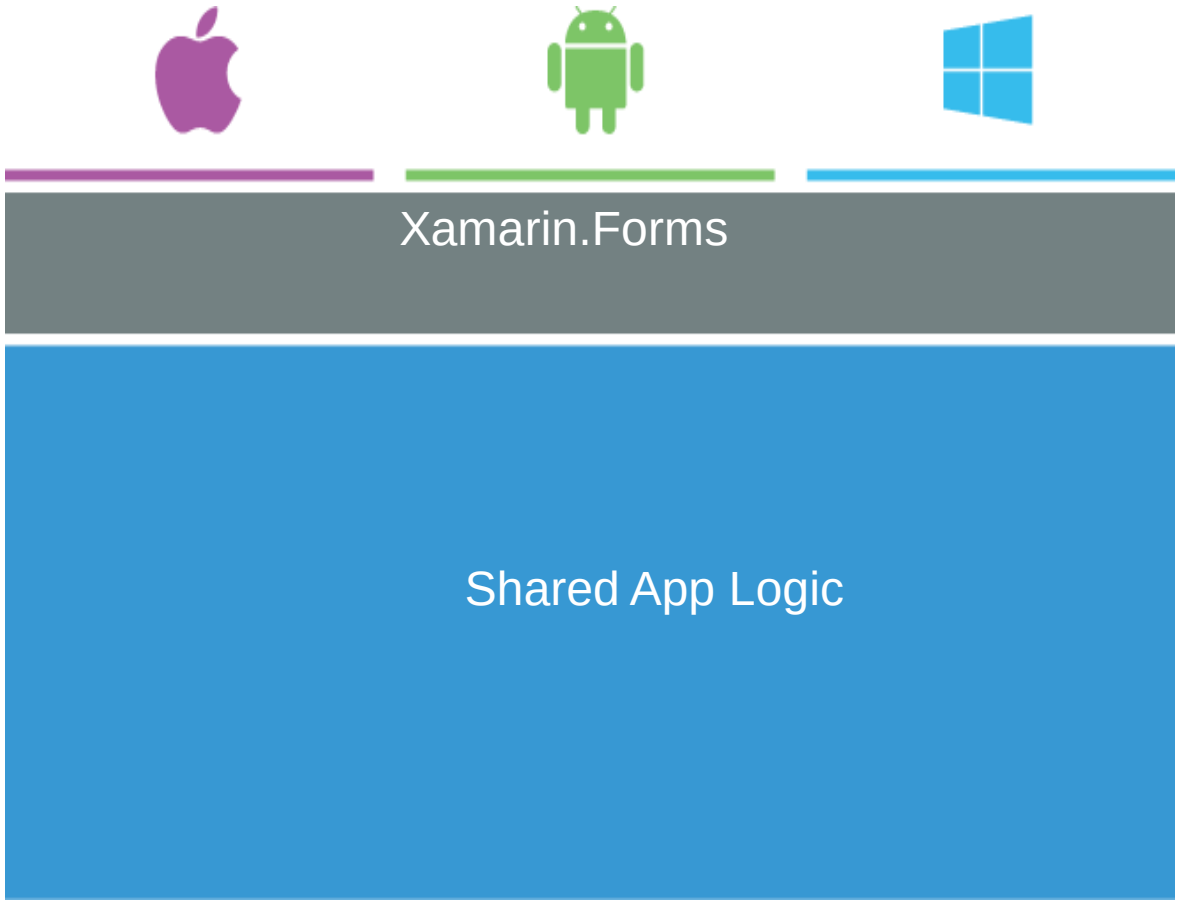
Write Once Run Anywhere

Code Sharing

“Traditional” Xamarin Architecture



“Newfangled” Alternative



Share Code

- Traditionally (Xamarin.iOS and Xamarin.Android)
  - …60-80+% code sharing
- Newfangled (Xamarin.Forms)
  - …90-100% code sharing
- Mix-n-match traditional w/ Xamarin.Forms
- iOS, Android, WP…yes, but also…
  - Xamarin.Mac
  - Amazon Fire, Glass, Mono, etc.
  - WPF, ASP.NET, Win8, etc.
  - Server-based services/apps…(e.g. CSLA)

Native Apps

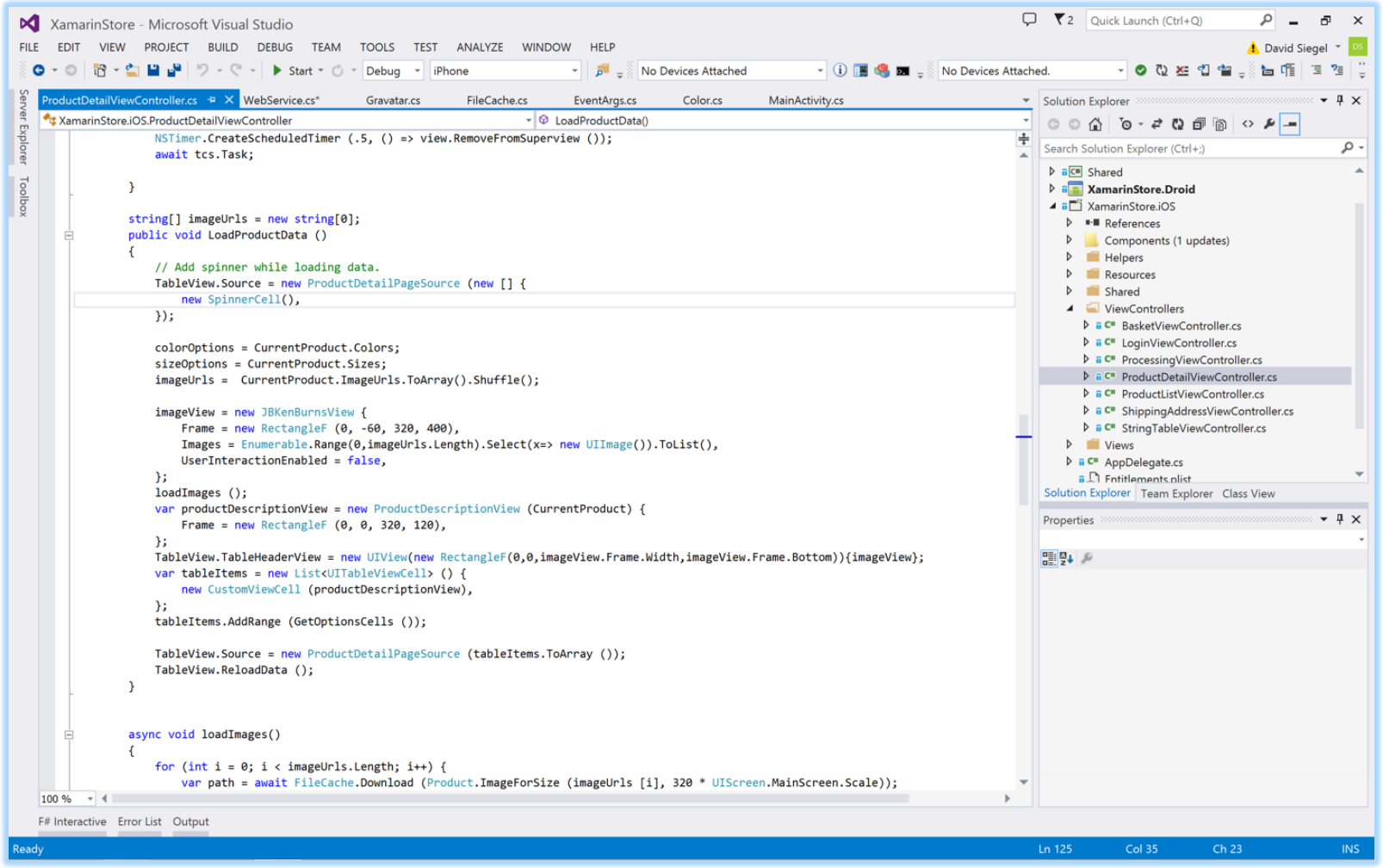
Xamarin Creates Native Apps

- Look and feel
- Performance
- “Hybrid” at design time…native at run time

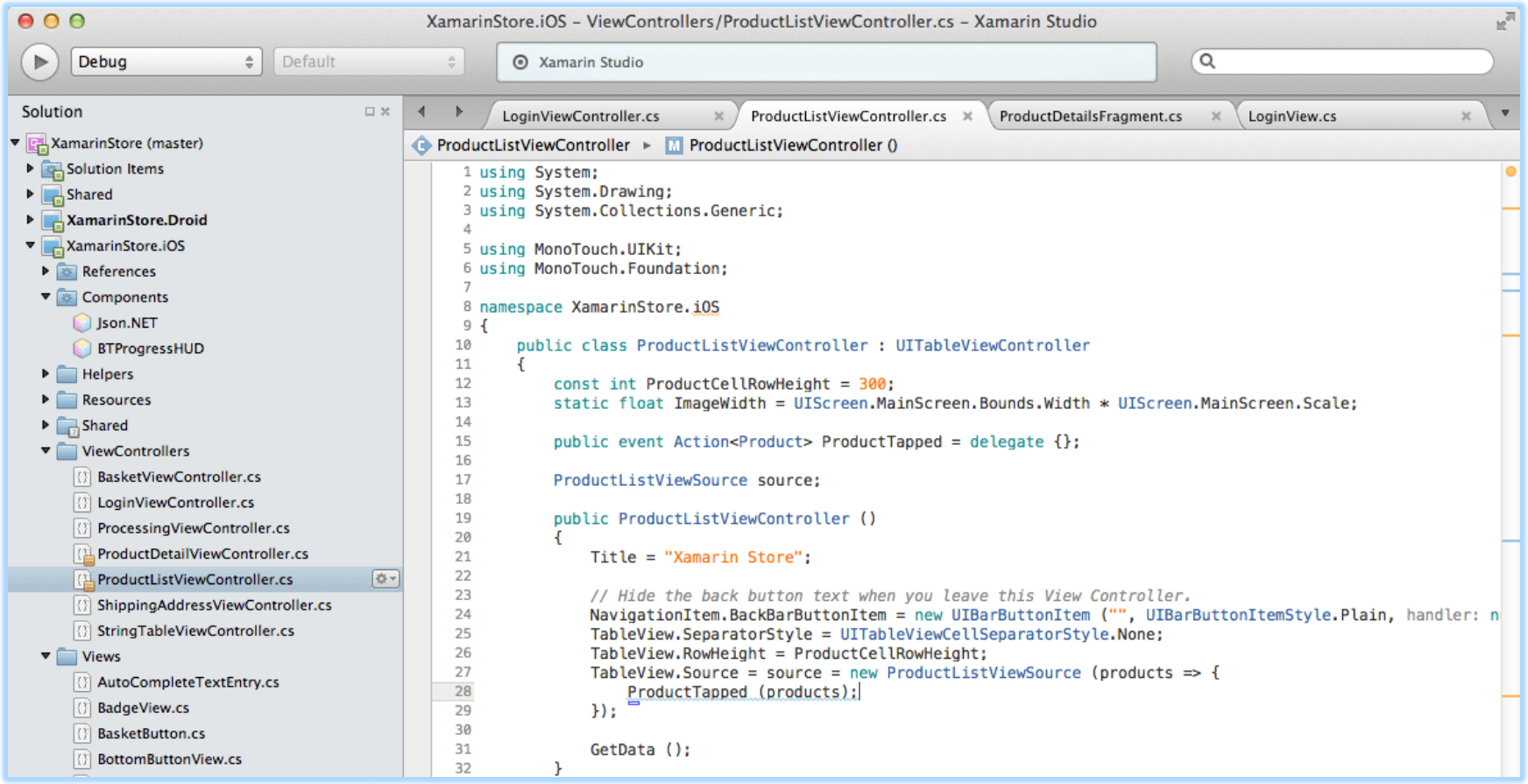
The Native API is Covered

- Anything in the native SDK is possible
- Access to device services
- Same day support

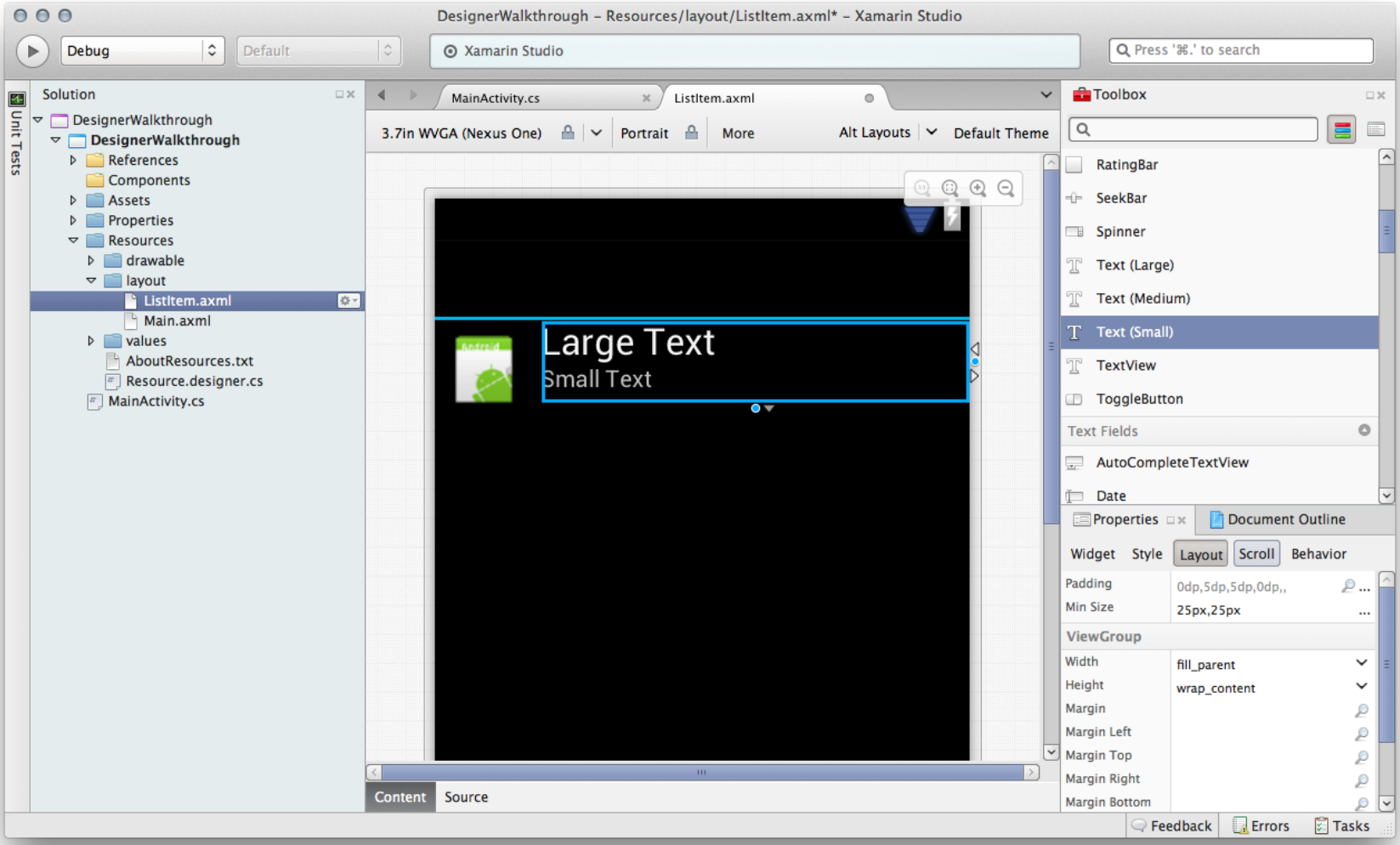
Native SDK++…Visual Studio



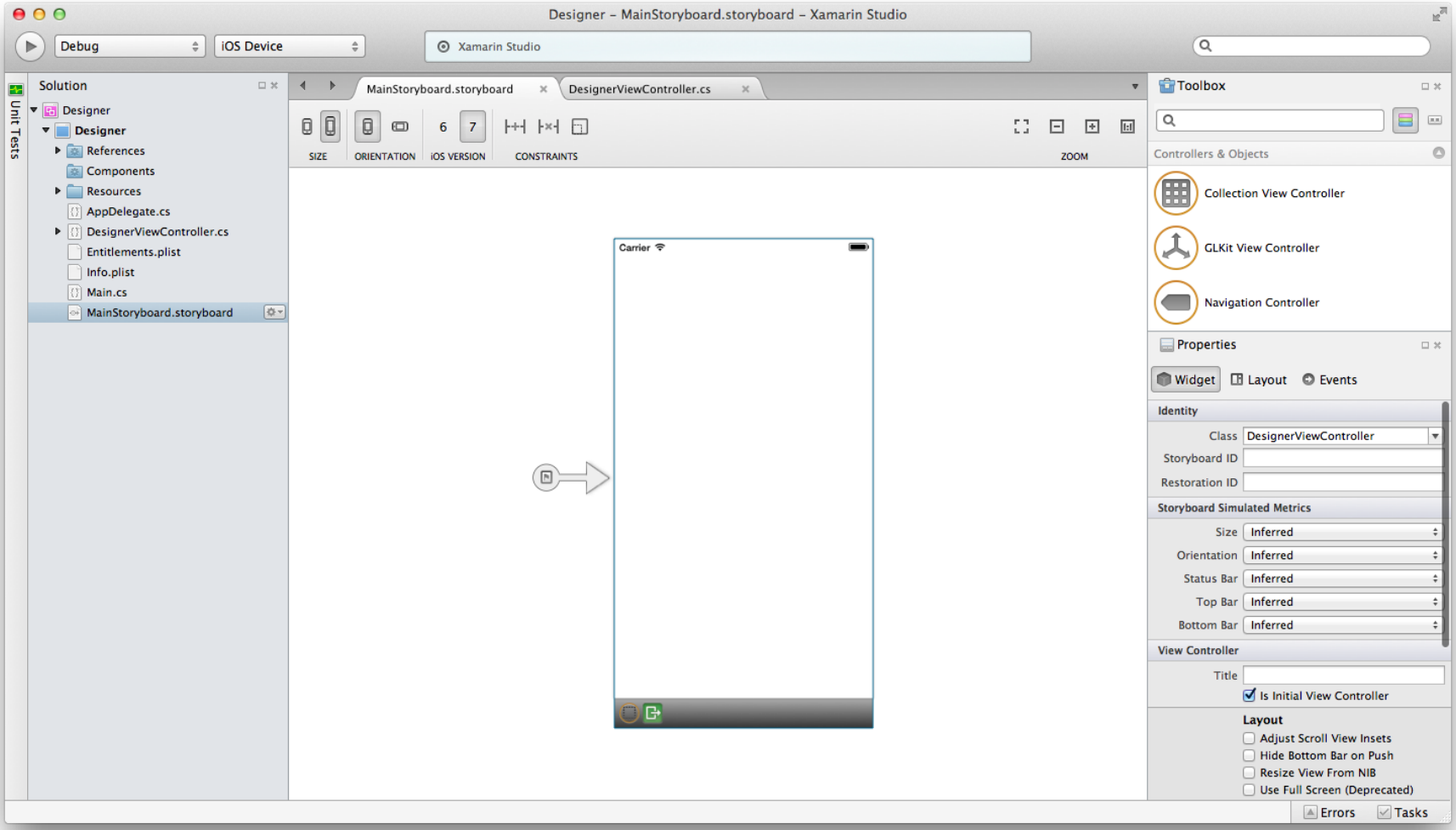
Native SDK++…Xamarin Studio



Native SDK++…Android Designer



Native SDK++…iOS Designer




Xamarin

Xamarin: Explosive Growth in 3 Years

10

In production use for 10 years

120

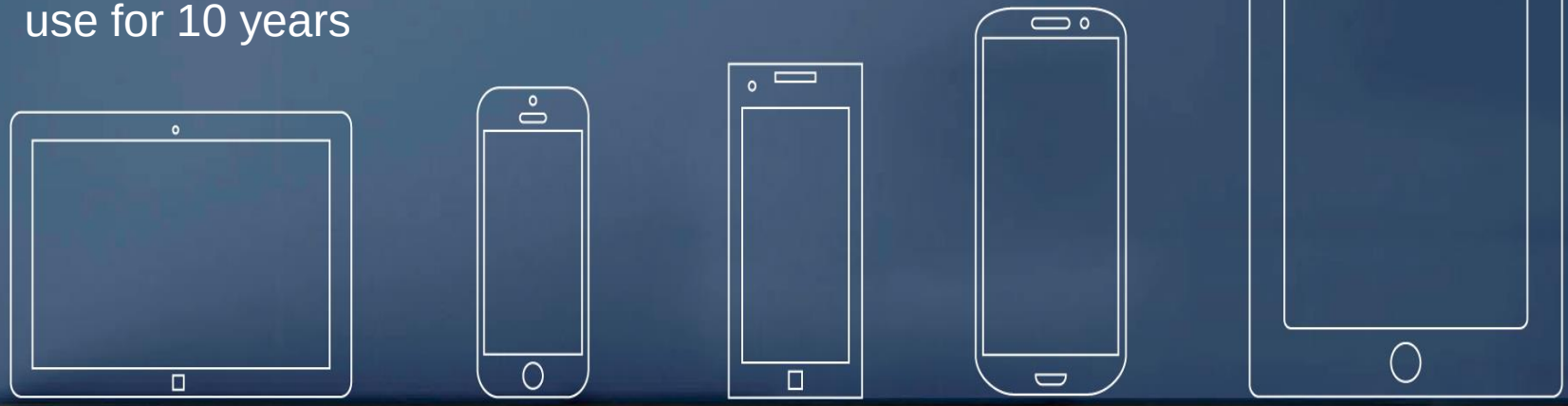
Customers in 120 countries

30,000

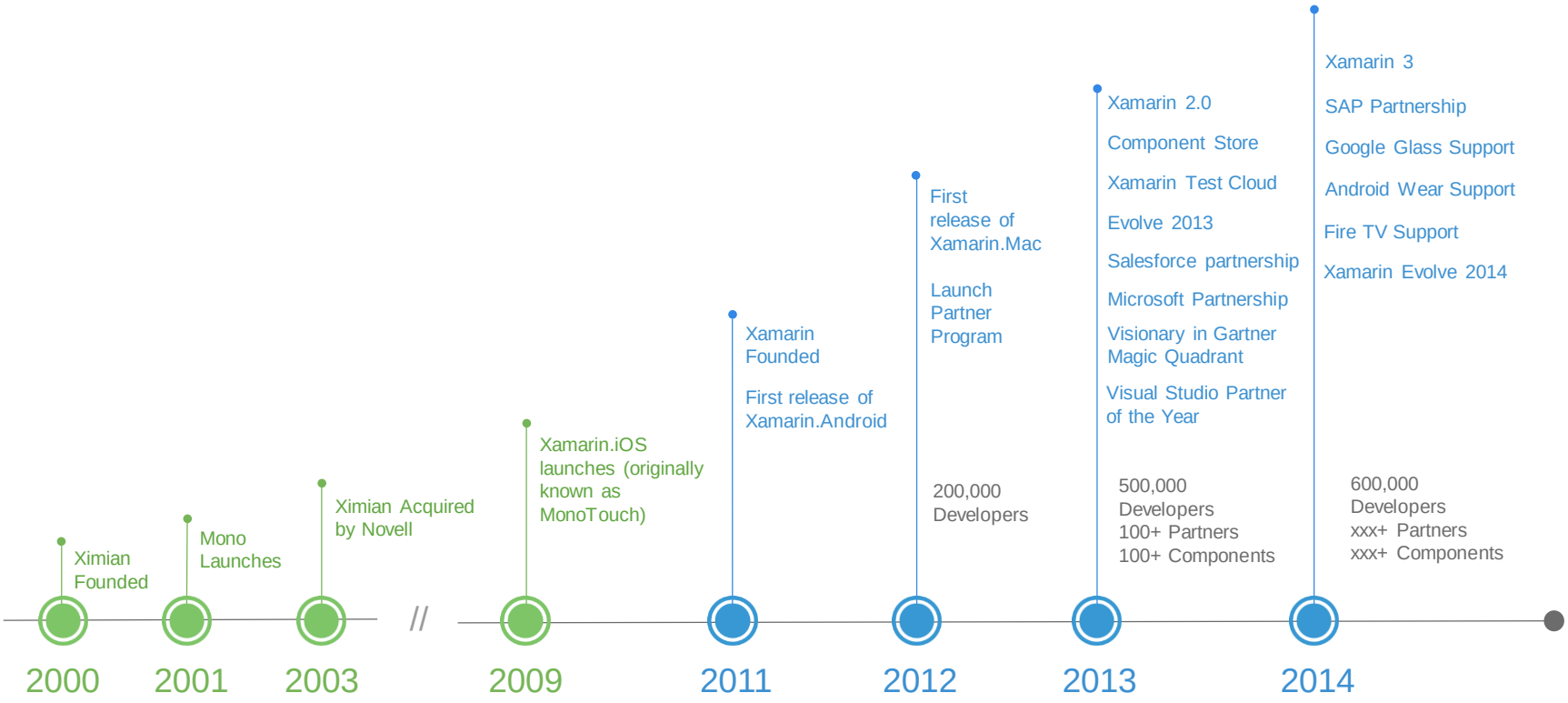
Adding over 30,000 developers a month

600,000

600,000 registered developers in just 3 years



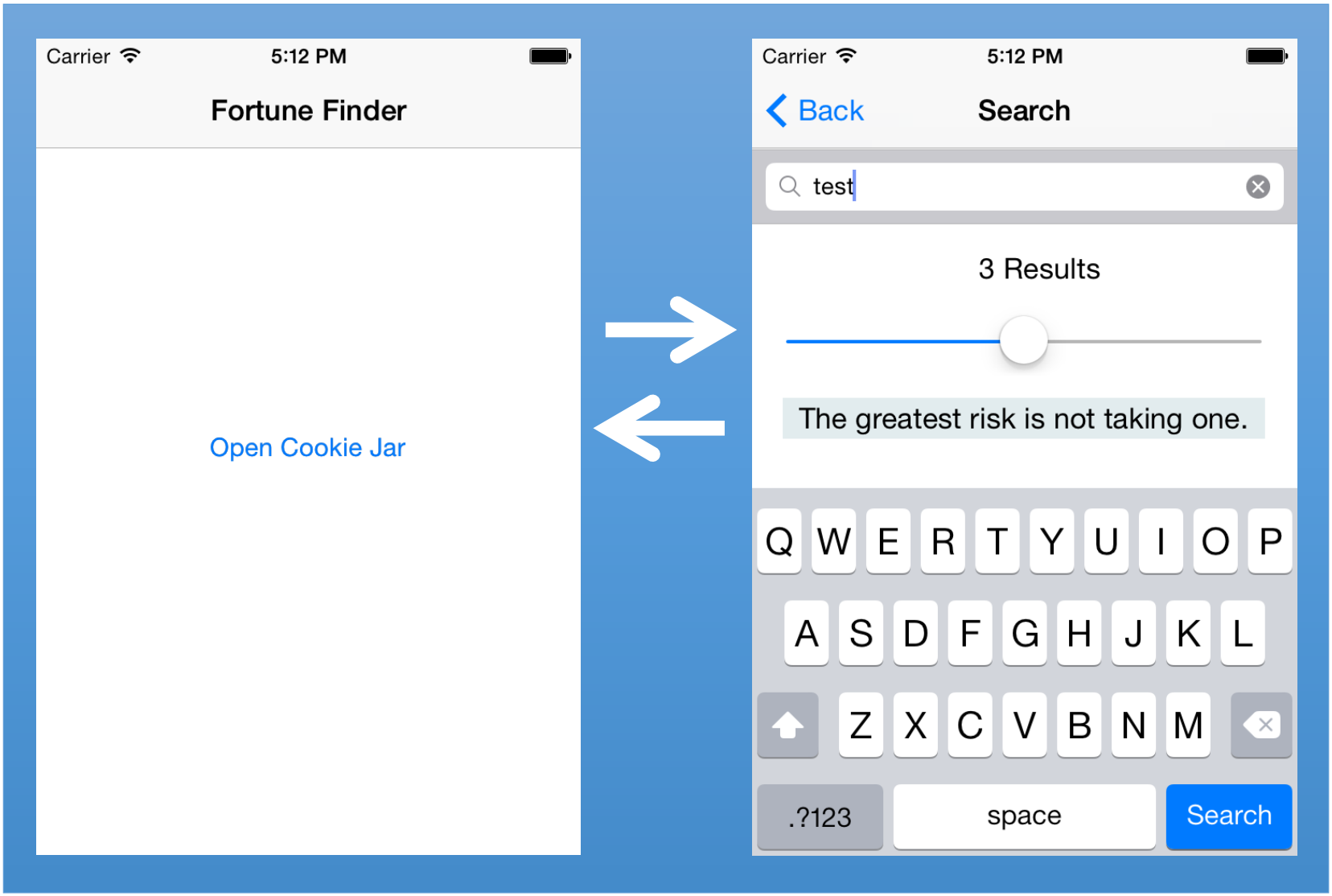
Xamarin and Microsoft Global Partnership


Ximian Founded
Mono Launches
Ximian Acquired by Novell
Xamarin.iOS launches (originally known as MonoTouch)
Xamarin Founded
First release of Xamarin.Android
First release of Xamarin.Mac
Launch Partner Program
200,000 Developers
Xamarin 2.0
Component Store
Xamarin Test Cloud
Evolve 2013
Salesforce partnership
Microsoft Partnership
Visionary in Gartner Magic Quadrant
Visual Studio Partner of the Year
500,000 Developers
100+ Partners
100+ Components
Xamarin 3
SAP Partnership
Google Glass Support
Android Wear Support
Fire TV Support
Xamarin Evolve 2014
600,000 Developers
xxx+ Partners
xxx+ Components
2000
2001
2003
//
2009
2011
2012
2013
2014

//TODO:

- ❑ Xamarin
- ❑ **HelloWorld++**
- ❑ Architecture
- ❑ Build it bigger
- ❑ Gotchas
- ❑ Resources

Let’s Build This



That's the Plan

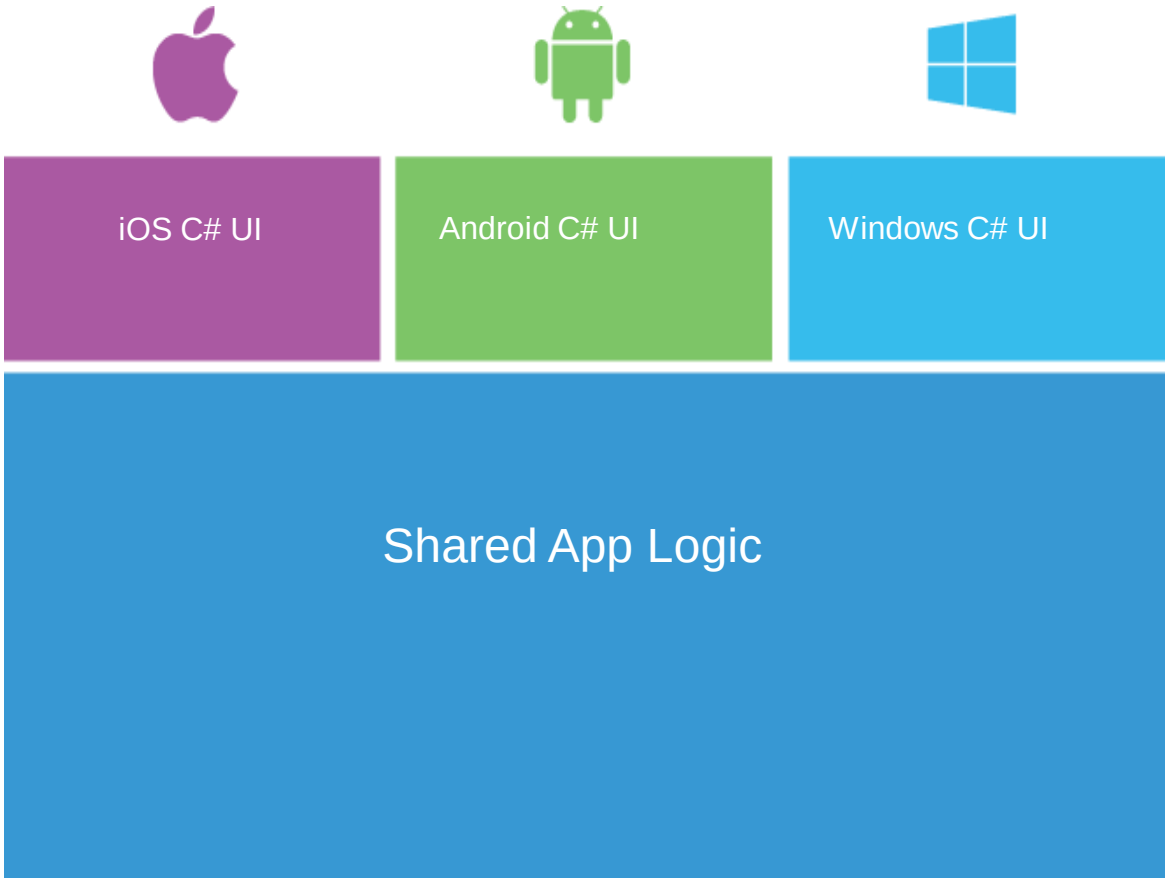
"Everybody has a plan till they get punched in the mouth."

- Mike Tyson

//TODO:

- ❑ Xamarin
- ❑ HelloWorld++
- ❑ **Architecture**
- ❑ Build it bigger
- ❑ Gotchas
- ❑ Resources

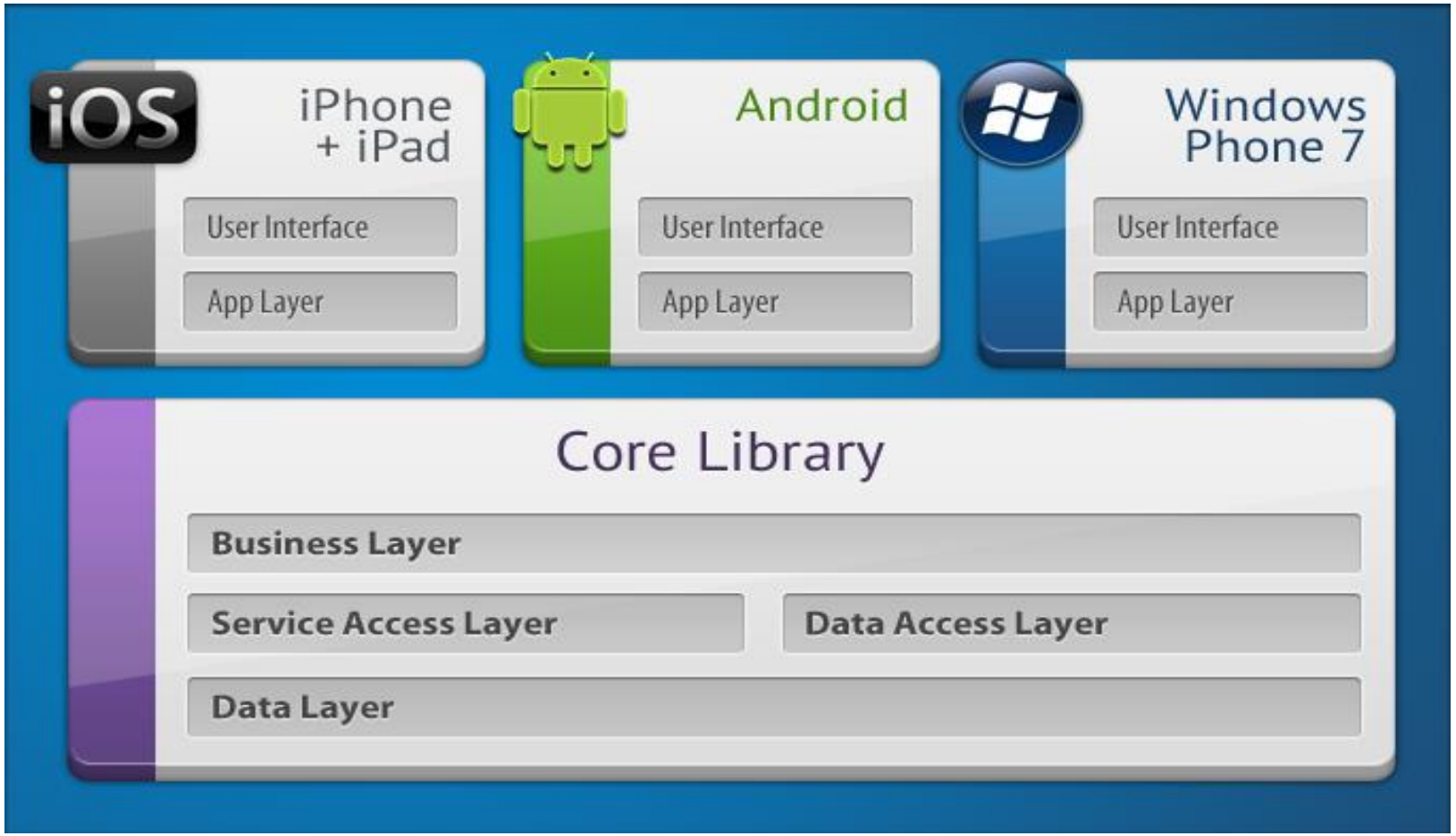
Let’s zoom in a bit further…



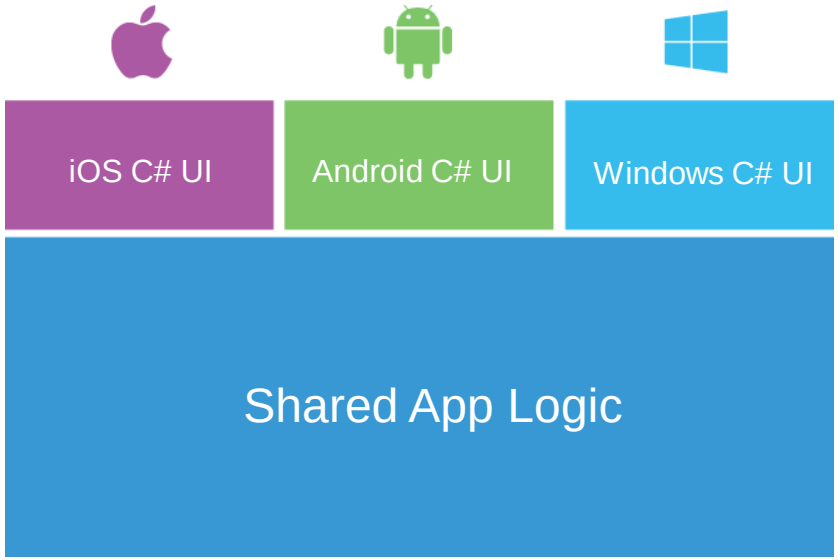
Architecture

Xamarin

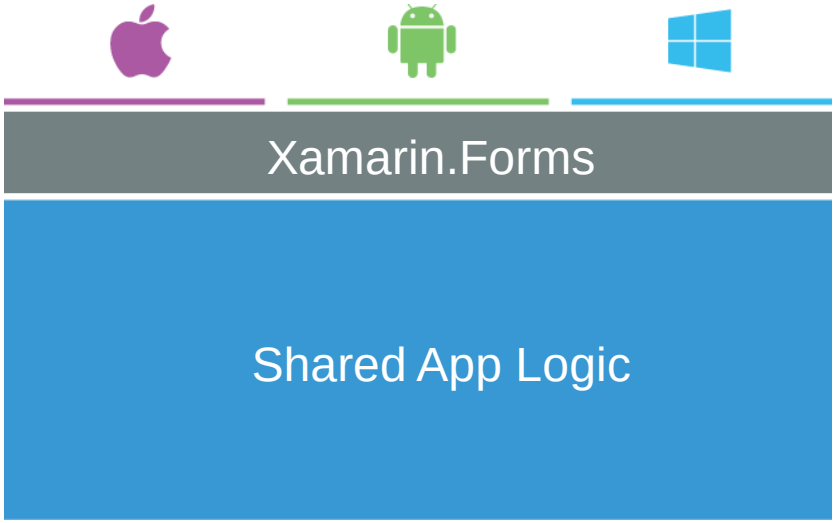
“Traditional” Approach



Default v. Xamarin.Forms



Default



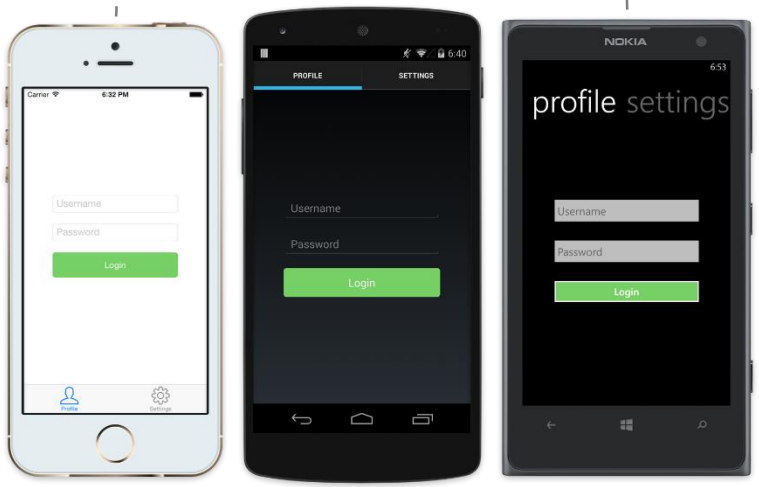
Xamarin.Forms

Xamarin.Forms

```
using Xamarin.Forms;

var profilePage = new ContentPage {
    Title = "Profile",
    Icon = "Profile.png",
    Content = new StackLayout {
        Spacing = 20, Padding = 50,
        VerticalOptions = LayoutOptions.Center,
        Children = {
            new Entry { Placeholder = "Username" },
            new Entry { Placeholder = "Password", IsPassword = true },
            new Button {
                Text = "Login",
                TextColor = Color.White,
                BackgroundColor = Color.FromHex("77D065") }}}
};

var settingsPage = new ContentPage {
    Title = "Settings",
    Icon = "Settings.png",
    (...)
};

var mainPage = new TabbedPage { Children = { profilePage, settingsPage } };
```

A single UI API

At runtime, each Xamarin.Forms page and its controls are mapped to platform-specific native user interface elements



How To Share

How do you share?

- Object-oriented abstraction
  - Interfaces, base classes, etc.
- Design patterns
  - IoC, subject/observer; pub/sub
- Language features
  - Partial classes, partial methods, extension methods
- Compiler and build features
  - Conditional compilation
  - Custom build scripts
- Solution structuring
  - File linking, shared projects, PCL's

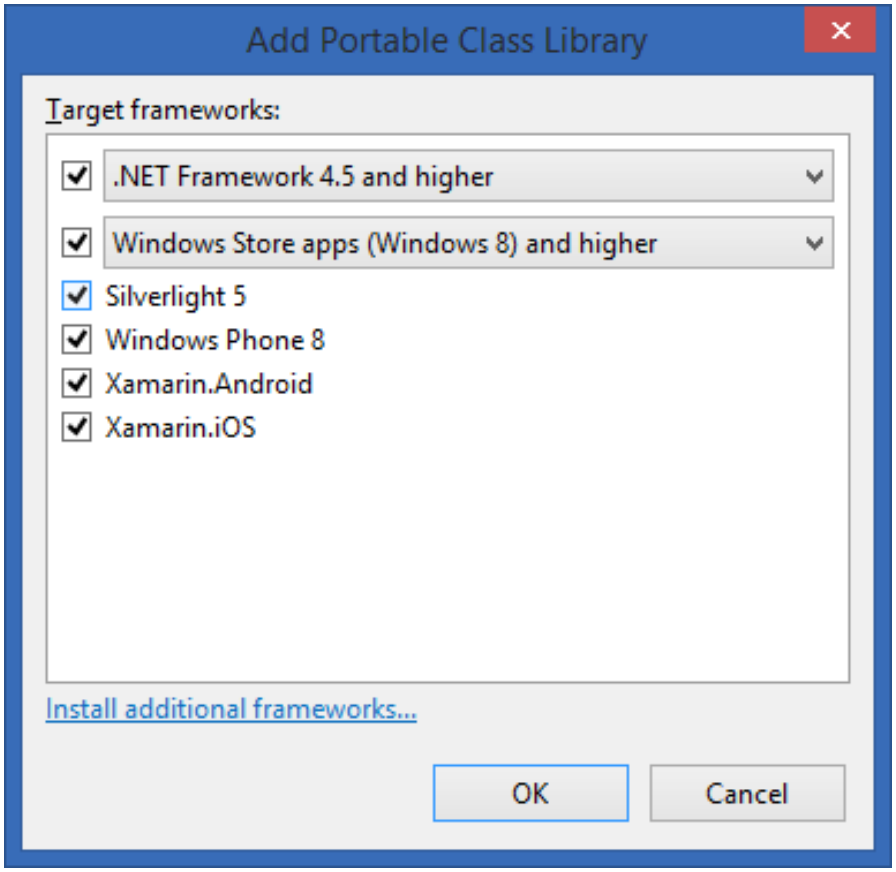
Solution Structuring for Core Code

- OK: File linking approach
  - Multiple “Core” project definitions
  - Use the “Project Linker” component to sync
- GOOD: Project sharing approach
  - One “Core” project
  - Build settings controlled by referencing projects
- EVEN GOODER: Multi-targeting via PCL
  - One “Core” project
  - Built to work x-platform as per PCL profile chosen

Portable Class Libraries

October 2013 – key support for PCL's announced by the .Net team

…including HttpClient, SignalR, and others



Portable Class Libraries

- Portable Class Libraries are specific to a “feature set”
- The support features are described by its “Profile”

Feature	.NET Framework	Windows Store	Silverlight	Windows Phone	Xamarin
Core	Y	Y	Y	Y	Y
LINQ	Y	Y	Y	Y	Y
IQueryable	Y	Y	Y	7.5+	Y
Serialization	Y	Y	Y	Y	Y
Data Annotations	4.0.3+	Y	Y	-	Y
System.IO.File	-	-	-	-	-

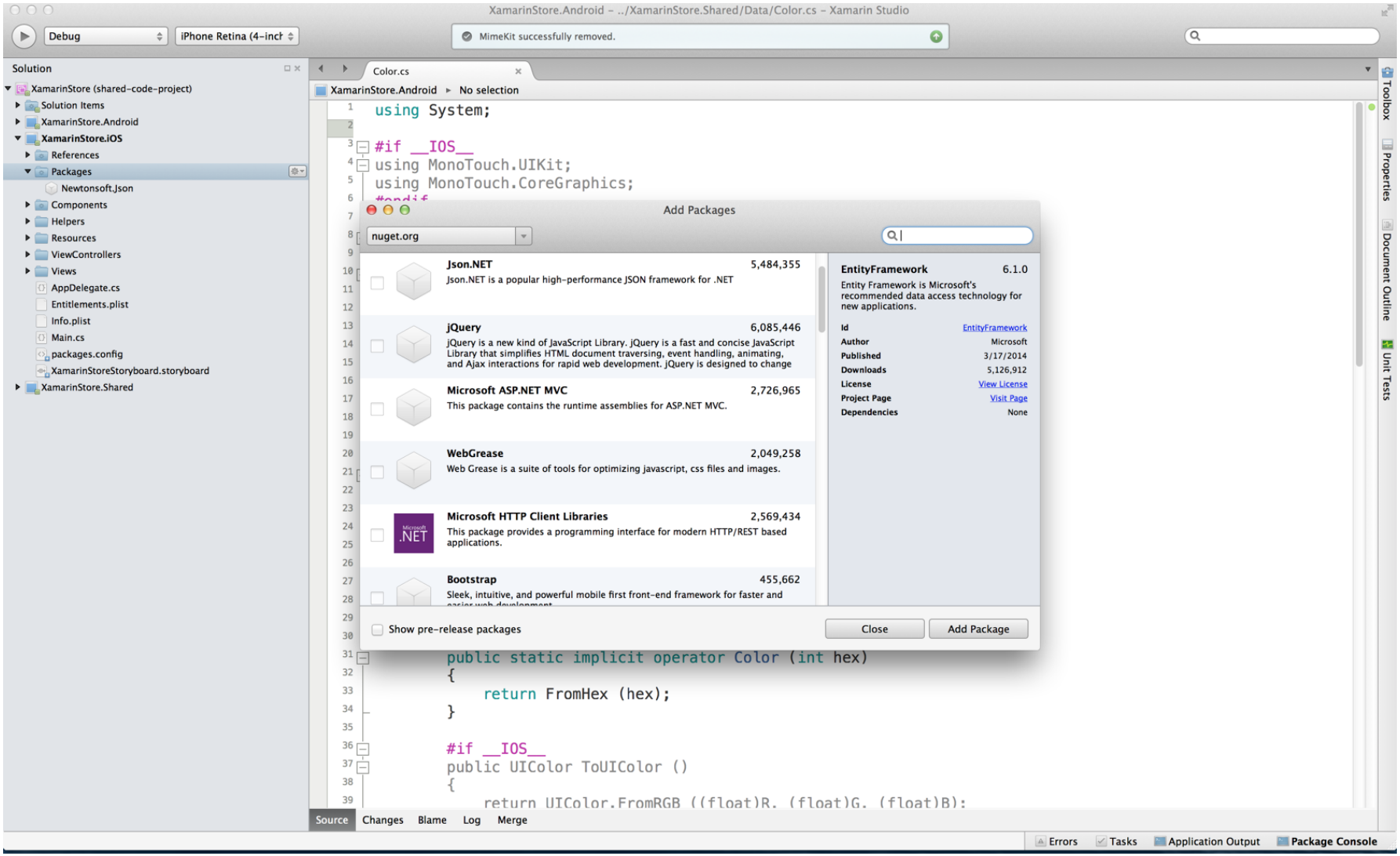
Find Similar UI Patterns

iOS – UIViewController	Android – Activity / Fragment
ViewDidLoad	OnCreate
ViewWillAppear	OnStart
ViewDidAppear	OnResume
ViewWillDisappear	OnPause
ViewDidDisappear	OnStop
UITableView	ListView
UITableViewDataSource	ListAdapter

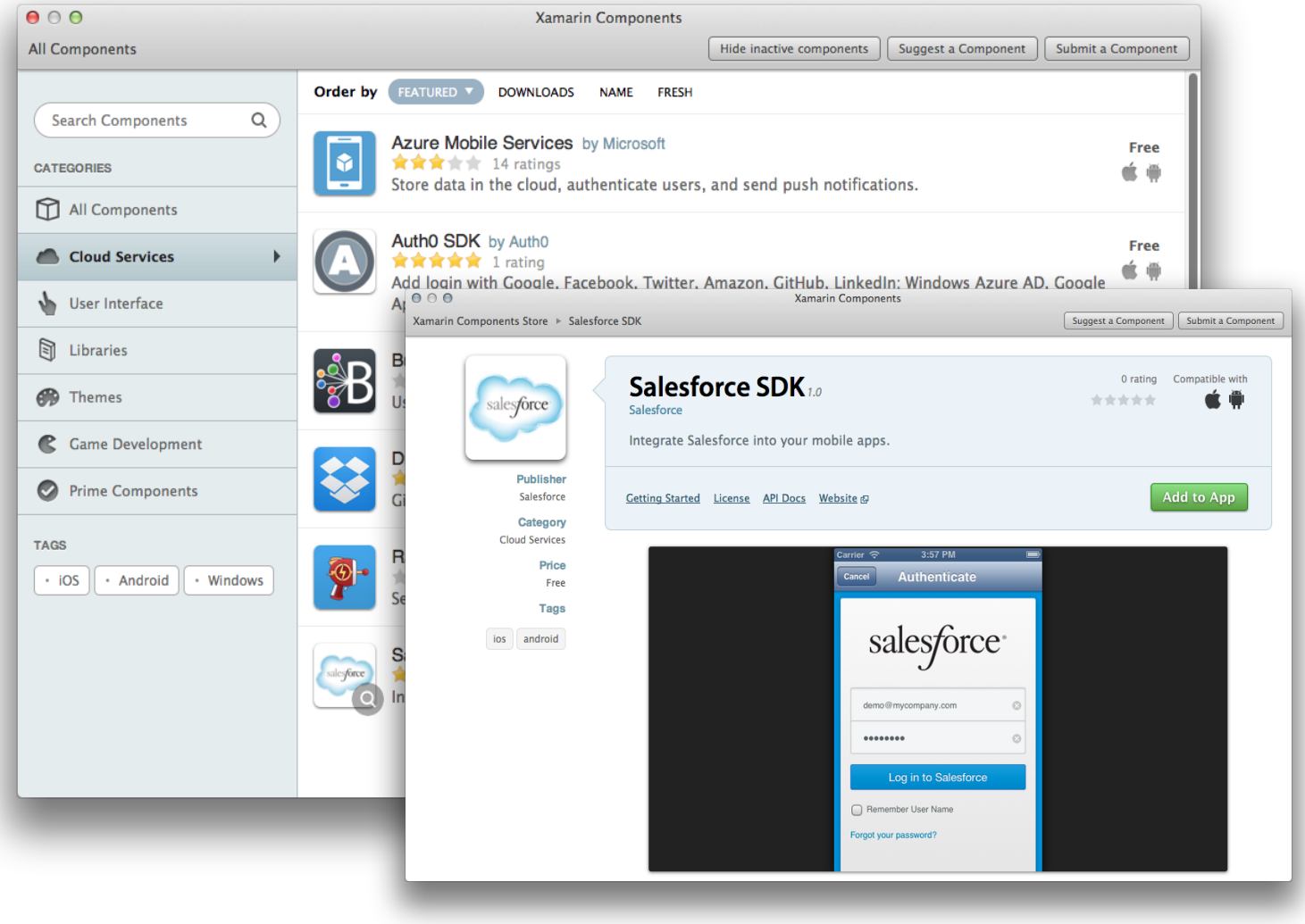
//TODO:

- ❑ Xamarin
- ❑ HelloWorld++
- ❑ Architecture
- ❑ **Build it bigger**
- ❑ Gotchas
- ❑ Resources

NuGet Package Manager



Xamarin Component Store



What Building Blocks Should I Use?

Popular X-Platform Components

From http://components.xamarin.com/

- Json.NET
- Xamarin.Mobile
- Facebook SDK
- Xamarin.Auth
- Xamarin.Social
- Azure Mobile Services
- Various UI Widgets…Bar Chart, Radial Progress, Login Screen, etc.

Xamarin.Mobile

Remote Requests

- BasicHttpBinding: Consume a WCF web service
- RestSharp: Simple REST web client
- HttpClient: Now PCL friendly
- The Paul Betts collection…
  - Fusillade: An opinionated HTTP library
  - Refit: Automatic type-safe REST library
  - ModernHttpClient: HttpClient implementations that use platform-native HTTP clients

Where do I park my data?

- SQLite is “typical” for local structured data
  - Could also use flat files, XML, etc.
- Best easy options for Sqlite
  1. ADO provider using System.Data and Mono.Data.Sqlite
  2. C#-SQLite – File-based C# port of SQLite
  3. Sqlite-net – “Simple, powerful, cross-platform SQLite client and ORM”
     - Strongly-typed queries
     - Single file (1,964 LOC)
- Component store
  - SQLLite.net component
  - SQLCipher component ($500)
    - 256-bit AES encryption

Graphics & Drawing

- OpenTK
  - OpenGL-based for Android, iOS
- MonoGame
  - XNA-API across platforms
- Cocos2D-XNA
  - 2D/3D game framework, XNA API compatible with MonoGame

Some Other Personal Favorites

- TinyIoC and TinyMessenger
  - DI and Pub/Sub
  - drop-in code, one PCL issue
- Rx (Reactive Extensions)
- NetTopologySuite
  - in NuGet as “NTS – Topology Suite”

Full Frameworks

- MonoCross
  - http://monocross.net/
- MvvmCross
  - https://github.com/MvvmCross/MvvmCross
- ReactiveUI
  - https://github.com/reactiveui/ReactiveUI

Native Libraries

- P/Invoke

```
[DllImport (Constants.UIKitLibrary,EntryPoint="UIRectFrameUsingBlendMode")]
public extern static void RectFrameUsingBlendMode (RectangleF rect, CGBlendMode bl
```

- iOS: Objective-C, C++ libraries
  - https://github.com/mono/monotouch-bindings
  - Objective Sharpie
- Android: Java, C, C++ libraries
- Binding Project Template
- Documented guides

//TODO:

- ❑ Xamarin
- ❑ HelloWorld++
- ❑ Architecture
- ❑ Build it bigger
- ❑ **Gotchas**
- ❑ Resources

MEGAFACEPALM

When a single facepalm is not enough.

UBC ASIC200 class of 2011

//GOTCHA: Reference Cycles

Garbage collection quietly coexists with native reference counting. Watch for cycles.

```
public class MyViewController : UIViewController
{
    public override void ViewDidLoad()
    {
        base.ViewDidLoad();

        Add(new MyView());
    }
}


public class MyView : UIView
{
    UIViewController parent;

    public MyView(MyViewController parentViewController)
    {
        parent = parentViewController;
    }
}
```

//GOTCHA: Reference Cycles

Workarounds

1. Design around it
2. Use WeakReference →
3. Set “parent” to null
4. Call Dispose

```
public class MyView: UIView
{
    WeakReference _parent;

    public MyView(MyViewController parentViewController)
    {
        _parent = new WeakReference(parentViewController);
    }

    void DoSomething()
    {
        var parent = _parent.Target as MyViewController;
        if (parent != null)
        {
            // now safe to use parent
        }
    }
}
```

//GOTCHA: Surprise References

- Event handlers are notorious for this

```
myTextField.EditingChanged +=
    (sender, e) => ViewModel.HelloText = myTextField.Text;
```

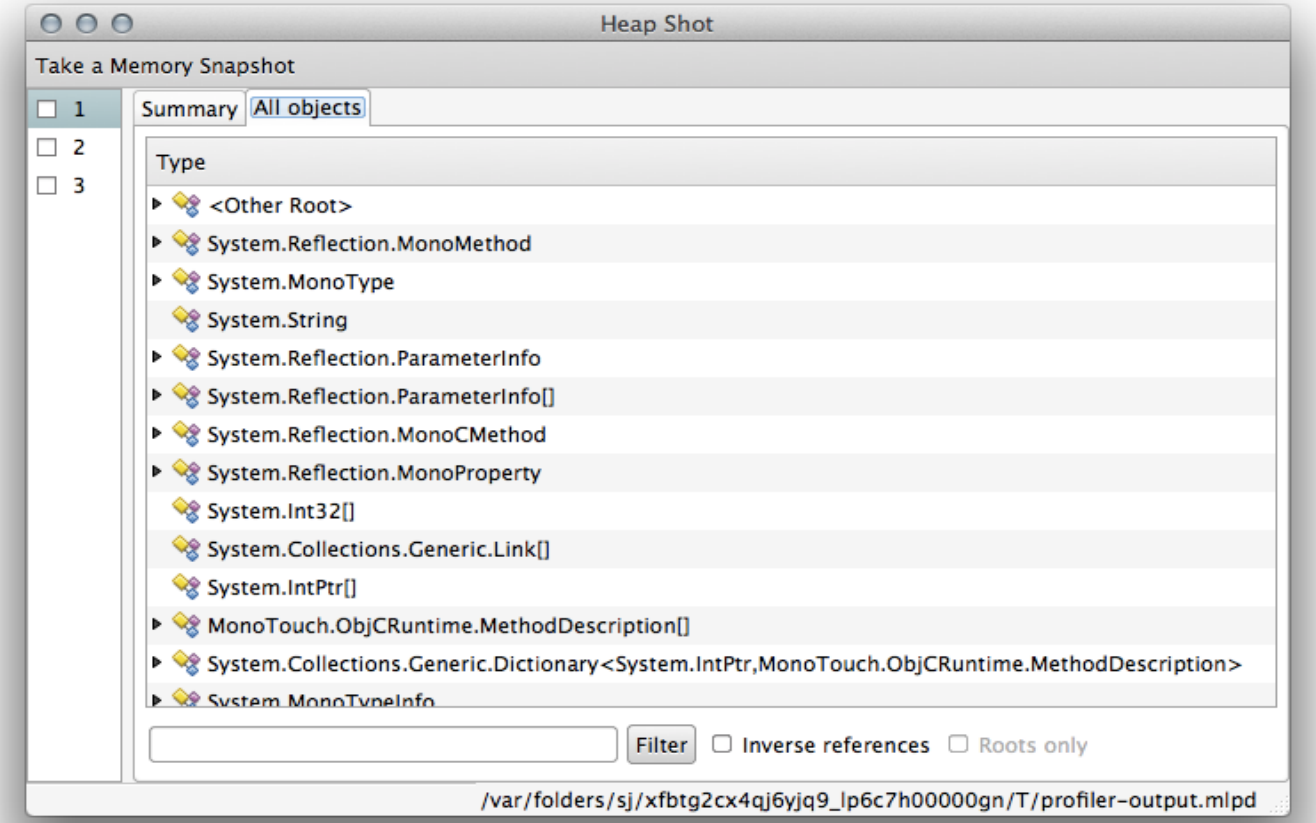
- Manual alternative

```
myTextField.EditingChanged += HandleValueChanged;
myTextField.EditingChanged -= HandleValueChanged;
```

- Something that's easier…an event wrapper

```
myTextField.EditingChanged += new WeakEventHandler(
    (sender, e) => ViewModel.HelloText = myTextField.Text
).Handler;
```

Use The Profiler…

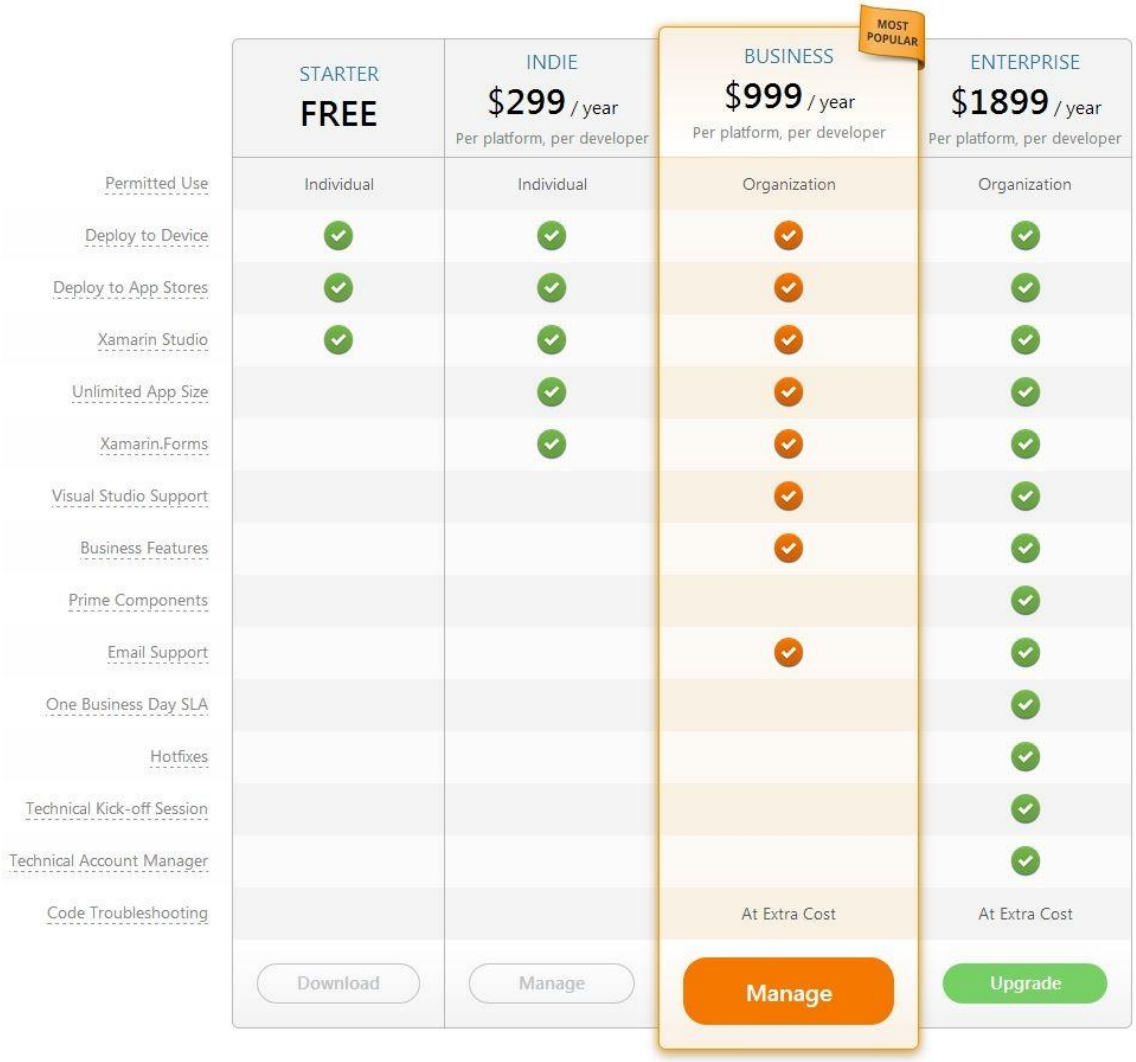


Android: [Consult the guide for directions]

//TODO:

- ❑ Xamarin
- ❑ HelloWorld++
- ❑ Architecture
- ❑ Build it bigger
- ❑ Gotchas
- ❑ **Resources**

What you'll need



	STARTER FREE	INDIE $299 / year Per platform, per developer	BUSINESS $999 / year Per platform, per developer MOST POPULAR	ENTERPRISE $1899 / year Per platform, per developer
Permitted Use	Individual	Individual	Organization	Organization
Deploy to Device	✓	✓	✓	✓
Deploy to App Stores	✓	✓	✓	✓
Xamarin Studio	✓	✓	✓	✓
Unlimited App Size		✓	✓	✓
Xamarin.Forms		✓	✓	✓
Visual Studio Support			✓	✓
Business Features			✓	✓
Prime Components				✓
Email Support			✓	✓
One Business Day SLA				✓
Hotfixes				✓
Technical Kick-off Session				✓
Technical Account Manager				✓
Code Troubleshooting			At Extra Cost	At Extra Cost
	Download	Manage	Manage	Upgrade

What you'll need

Development OS	Mac OS X	Windows	
IDE	Xamarin Studio	Xamarin Studio	Visual Studio
iOS	Y	-	Y
Android	Y	Y	Y
Windows Phone	-	-	Y

For iOS you need a Mac to build on

even if you use Windows to develop on

Check out this link

Pretty much everything starts here

http://developer.xamarin.com/

- Getting started, videos, tutorials, samples, recipes, API, advanced topics, etc.
- Special section covering cross-platform topics

Helpful Tips

- Pick one platform to do first
- Don’t just Google for “Xamarin”
  - The entire native knowledge base is usable
- Understand MonoTouch and MonoDroid monikers
  - Historical evolution of naming
  - MonoTouch == Xamarin.iOS
  - MonoDroid == Mono for Android == Xamarin.Android

Tools for the Mac Tool Belt

(for first time Mac'ers)

- Charles (like Fiddler): http://www.charlesproxy.com/
- Graphics work
  - Screen captures/clips
    - Grab (built-in utility)
    - Entire desktop: **Command-Shift-3**
    - Portion of desktop: **Command-Shift-4**
    - Specific application window: **Command-Shift-4**, then **Spacebar**
    - Add **Control** to send to clipboard instead of file on desktop
  - DigitalColor Meter (built-in utility)
  - PaintCode: http://www.paintcodeapp.com/
- SourceTree: http://www.sourcetreeapp.com/
- SizeUp: https://www.irradiatedsoftware.com/sizeup/

Go **X**-Platform with

Presentation materials available on GitHub

https://github.com/DennisWelu/presentations

https://github.com/DennisWelu/csharp-utils

@DennisWelu

DennisWelu@MotisConsulting.com