

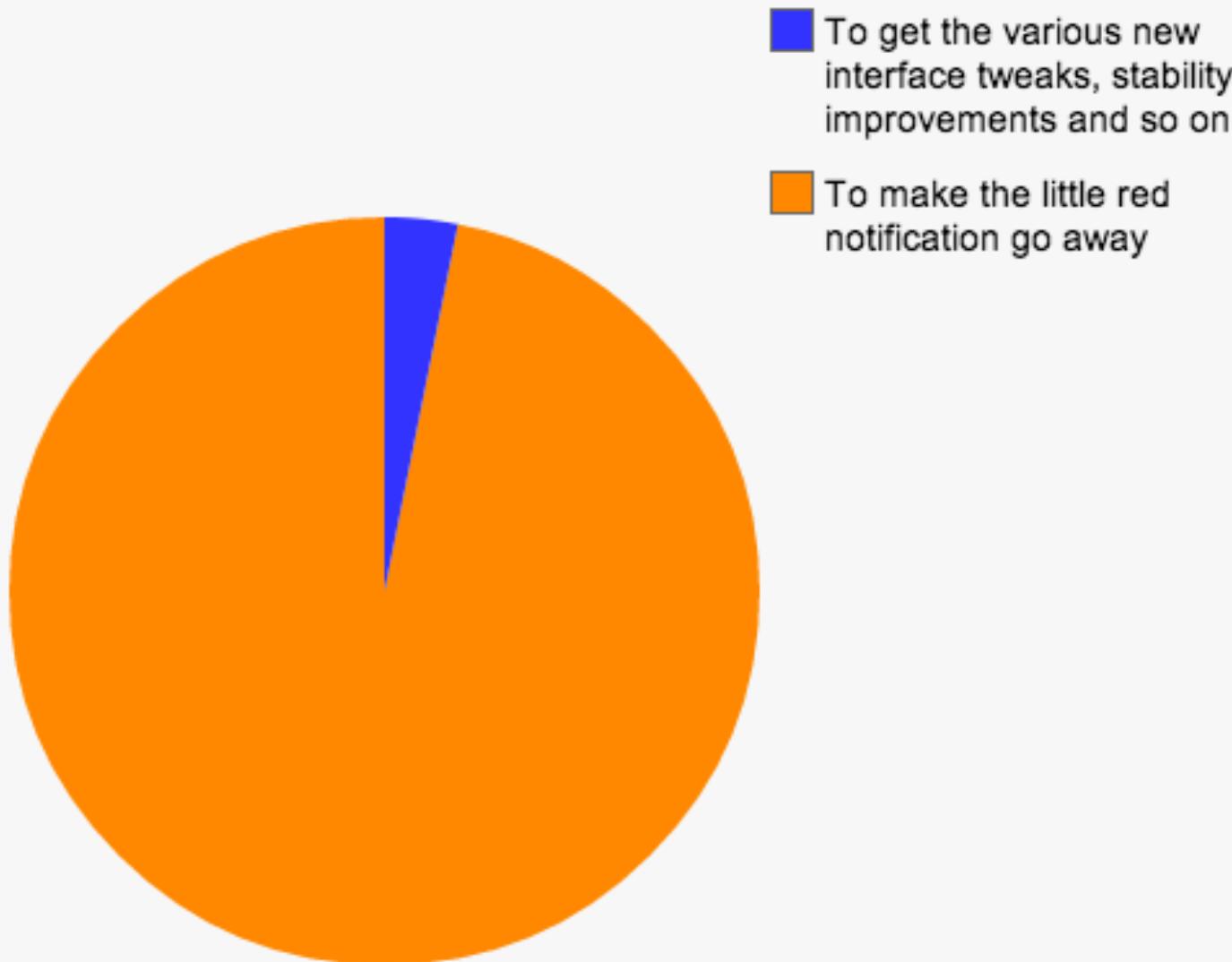
Go X-Platform with **Xamarin**

Have your C# and mobile too



August 12, 2014
@DennisWelu
DennisWelu@MotisConsulting.com

Reasons why I update my apps

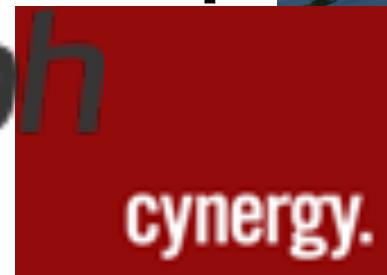




**Dubuque .Net
Users Group**

Cartegraph

Mood board



Motis





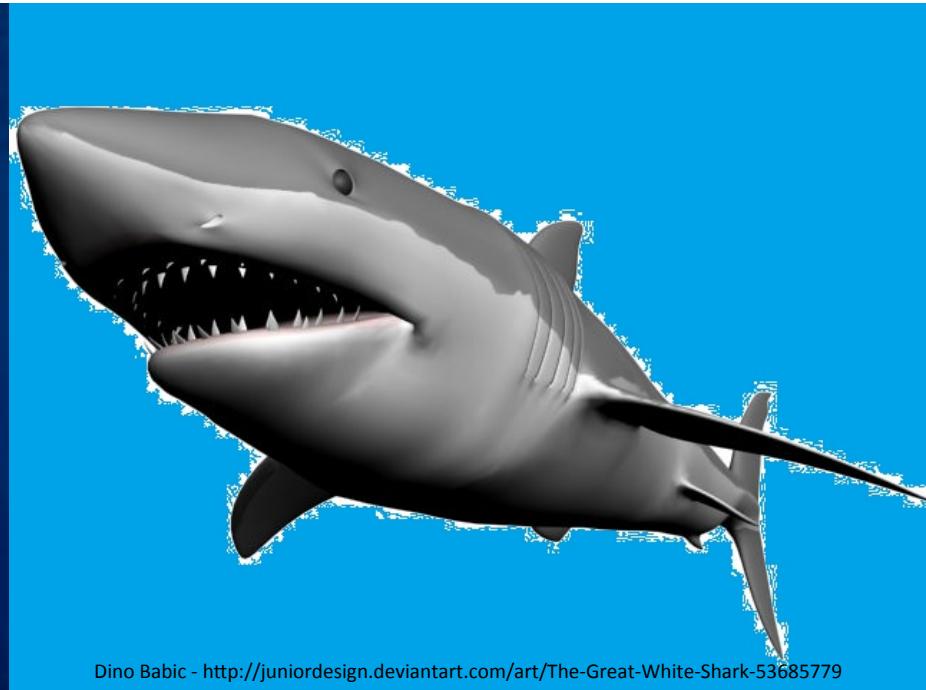
Way Cool App



y Cool App



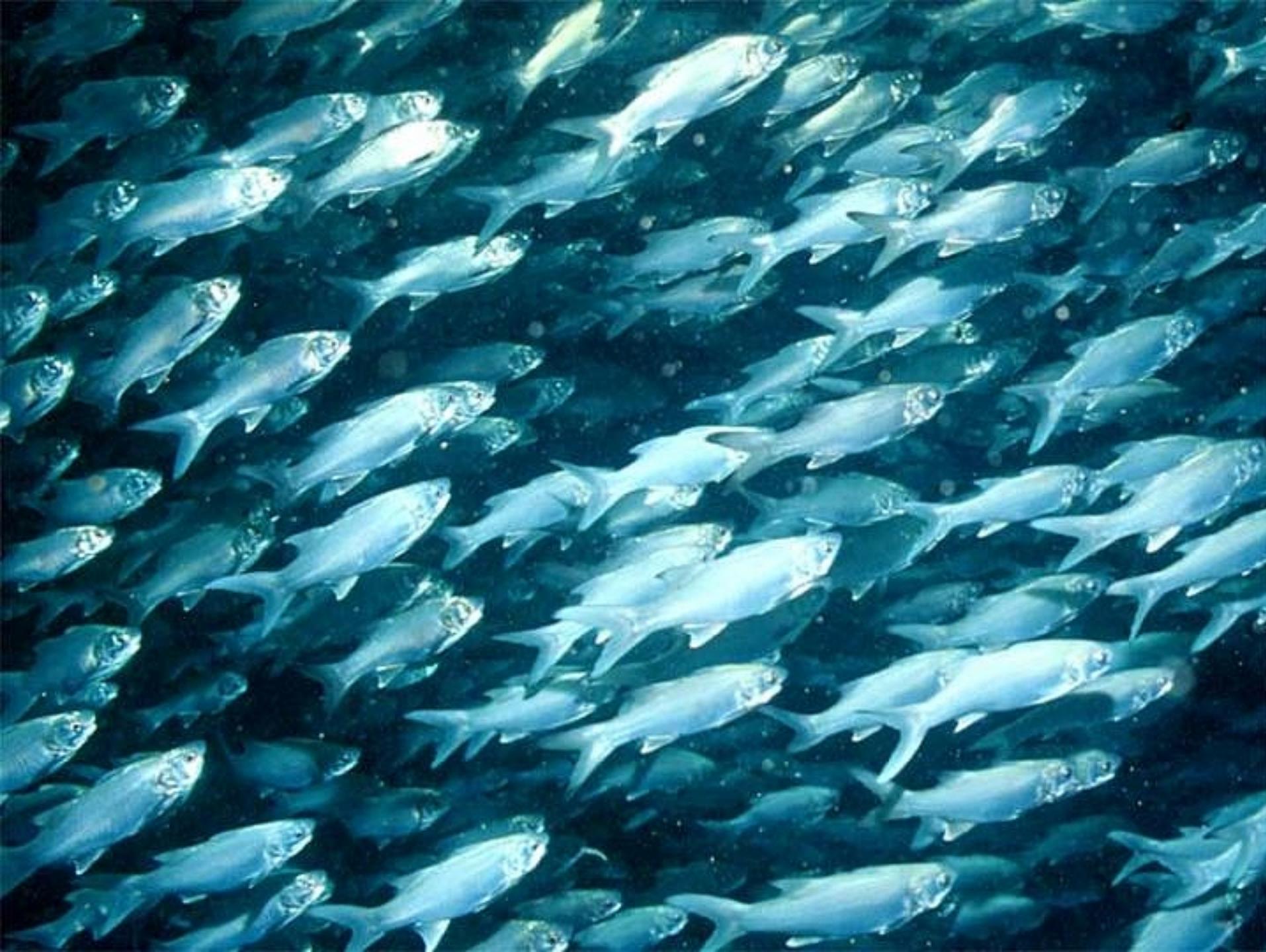
Ken Bondy - <http://www.fotopedia.com/items/flickr-2145763683>



Dino Babic - <http://juniordesign.deviantart.com/art/The-Great-White-Shark-53685779>

Real

Fake





//TODO:

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

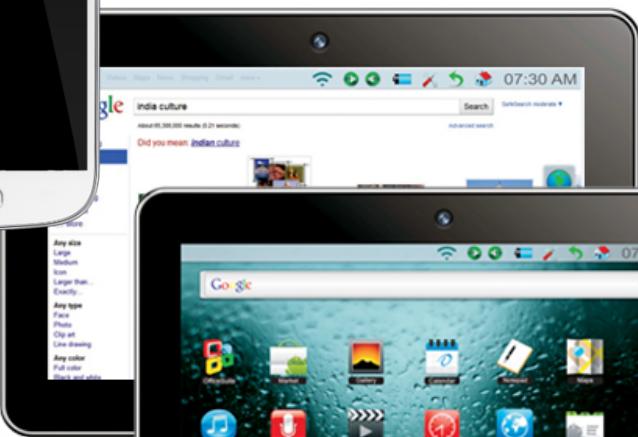
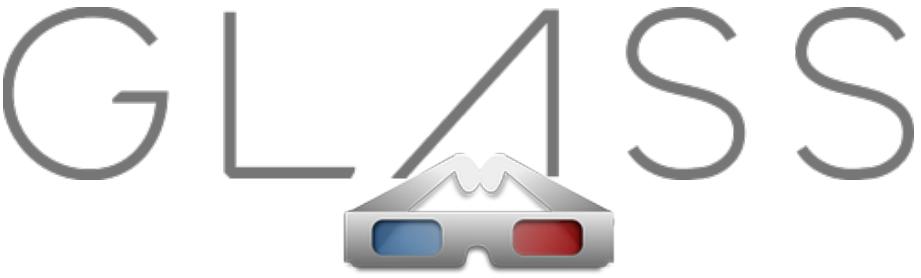


C#

on 2.5 Billion Devices



amazon fireTV



kindle fire

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



iOS App

Objective-C
XCode



Android App

Java
Eclipse

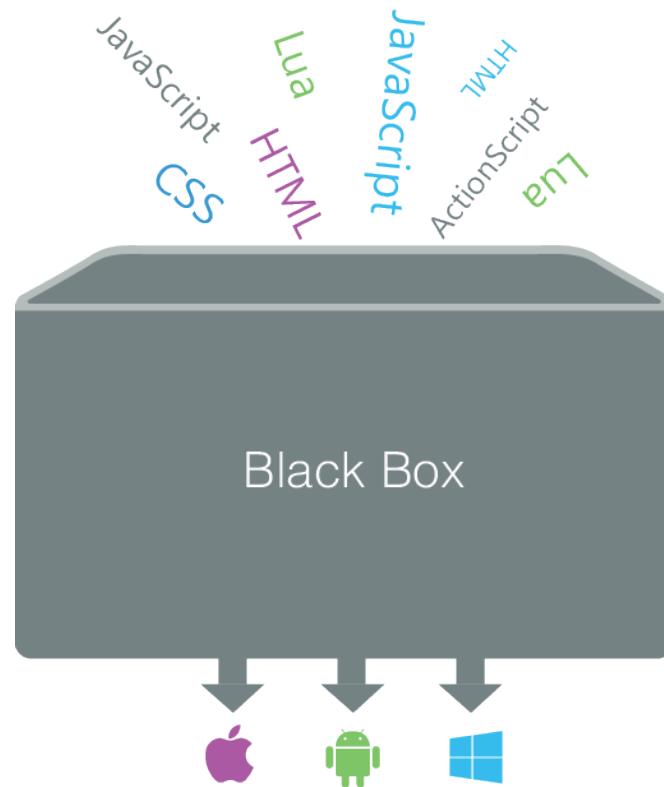


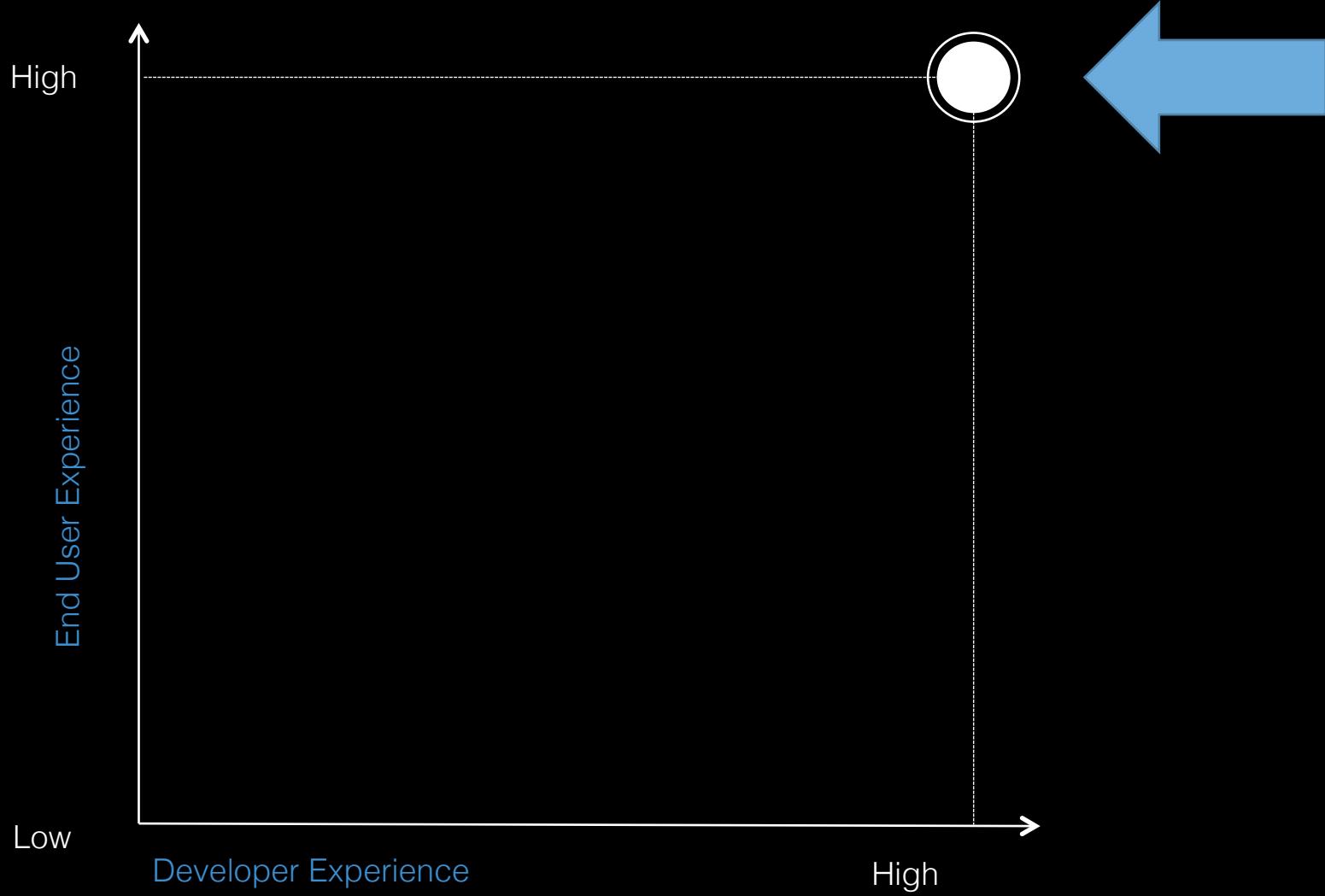
Windows App

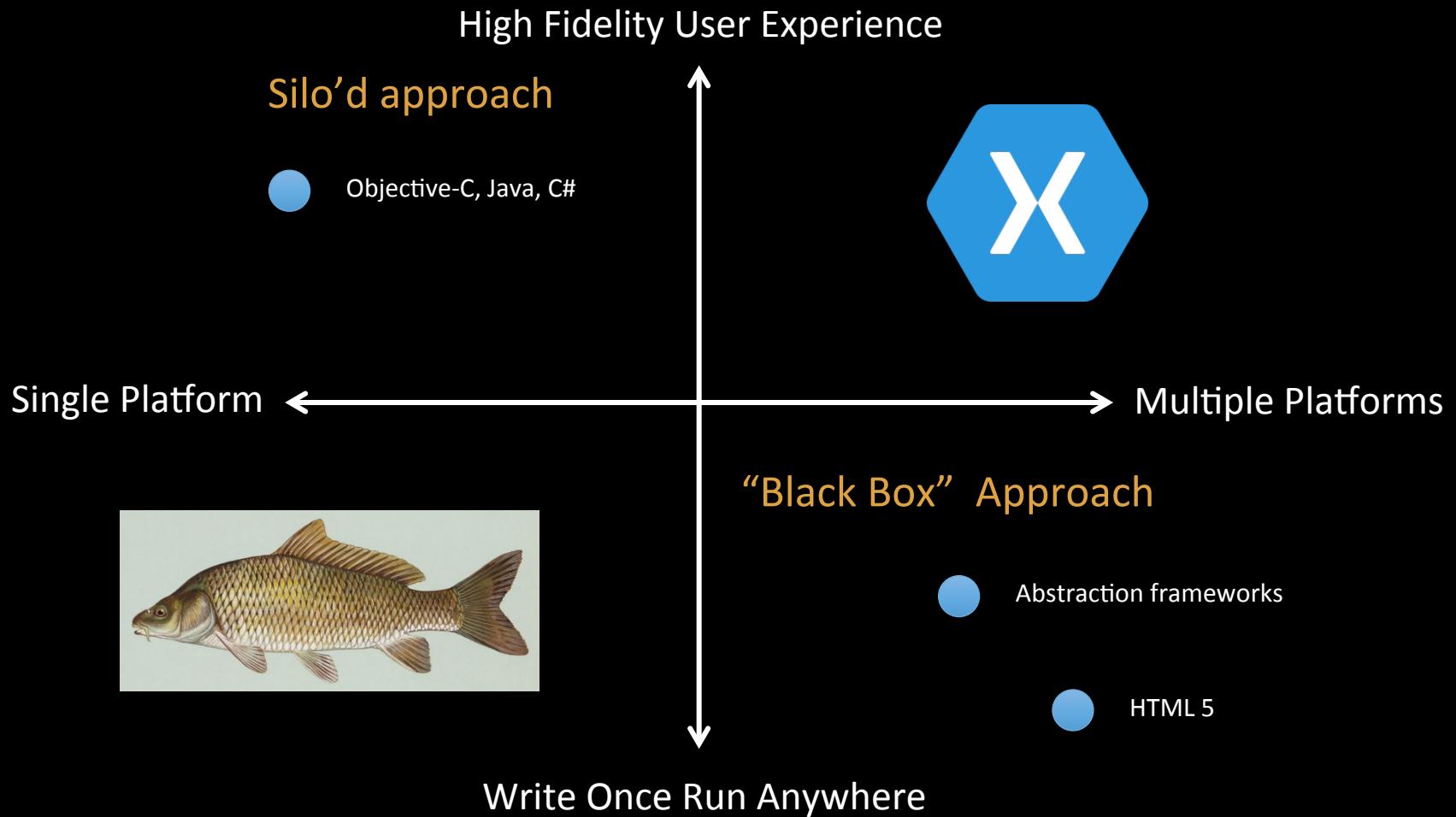
C#
Visual Studio

Write-Once-Run-Anywhere Approach

- UX?
- Device services?
- API coverage?

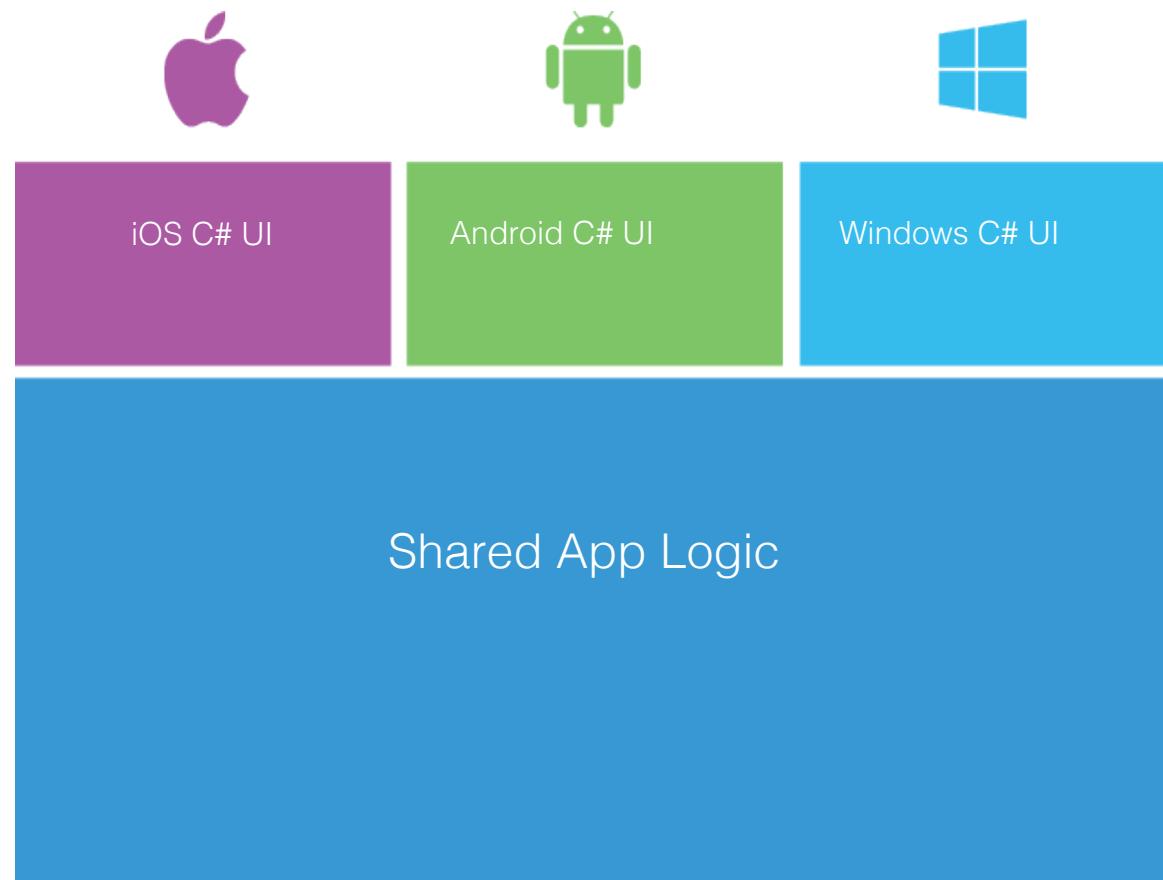




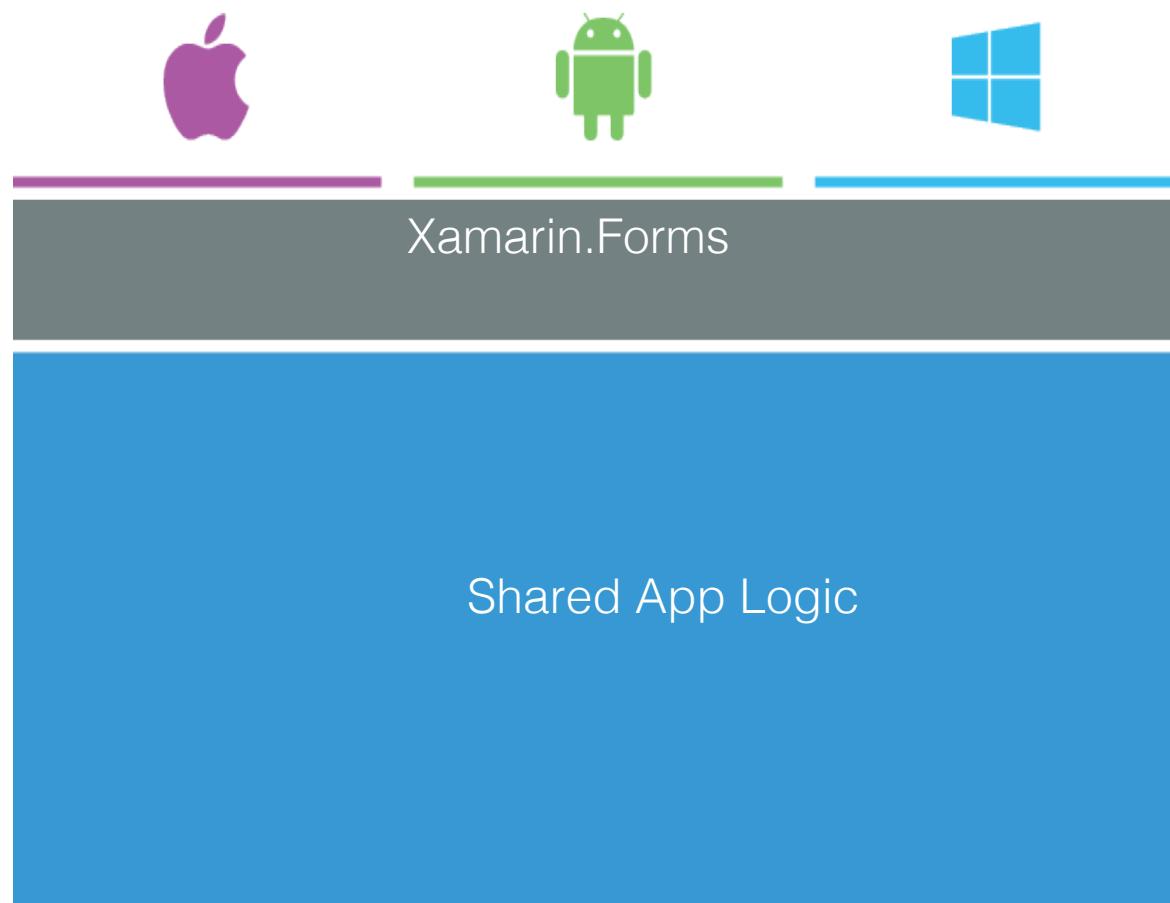


Code Sharing

“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)

High Fidelity Ux

Xamarin Creates Native Apps

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

The Native API is Covered

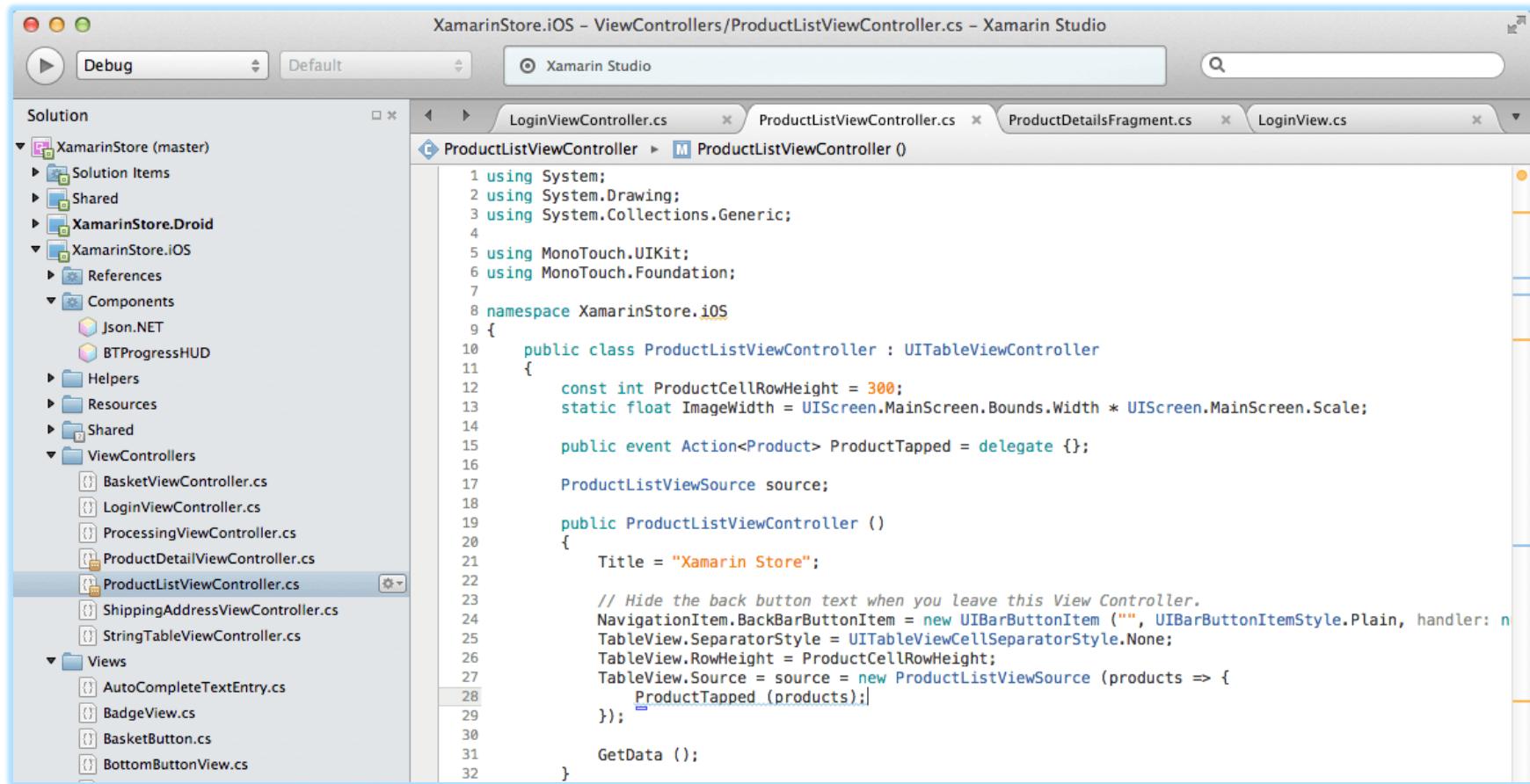
- Anything in the native SDK is possible
- Access to device services
- Same day support

Native SDK++...Visual Studio

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** XamarinStore - Microsoft Visual Studio
- Menu Bar:** FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TEST ANALYZE WINDOW HELP
- Toolbars:** Standard, Debug, Task List, Status Bar.
- Solution Explorer:** Shows the project structure:
 - Shared
 - XamarinStore.Droid
 - References
 - Components (1 updates)
 - Helpers
 - Resources
 - Shared
 - ViewControllers
 - BasketViewController.cs
 - LoginViewController.cs
 - ProcessingViewController.cs
 - ProductDetailViewController.cs
 - ProductListViewController.cs
 - ShippingAddressViewController.cs
 - StringTableViewCell.cs
 - Views
 - AppDelegate.cs
 - Entitlements.plist
 - Solution Explorer Team Explorer Class View
- Properties Window:** Available on the right side of the interface.
- Editor:** Displays the `ProductDetailViewController.cs` file content, which includes code for loading product data and setting up a JBKenBurnsView.
- Status Bar:** Ready, Ln 125, Col 35, Ch 23, INS

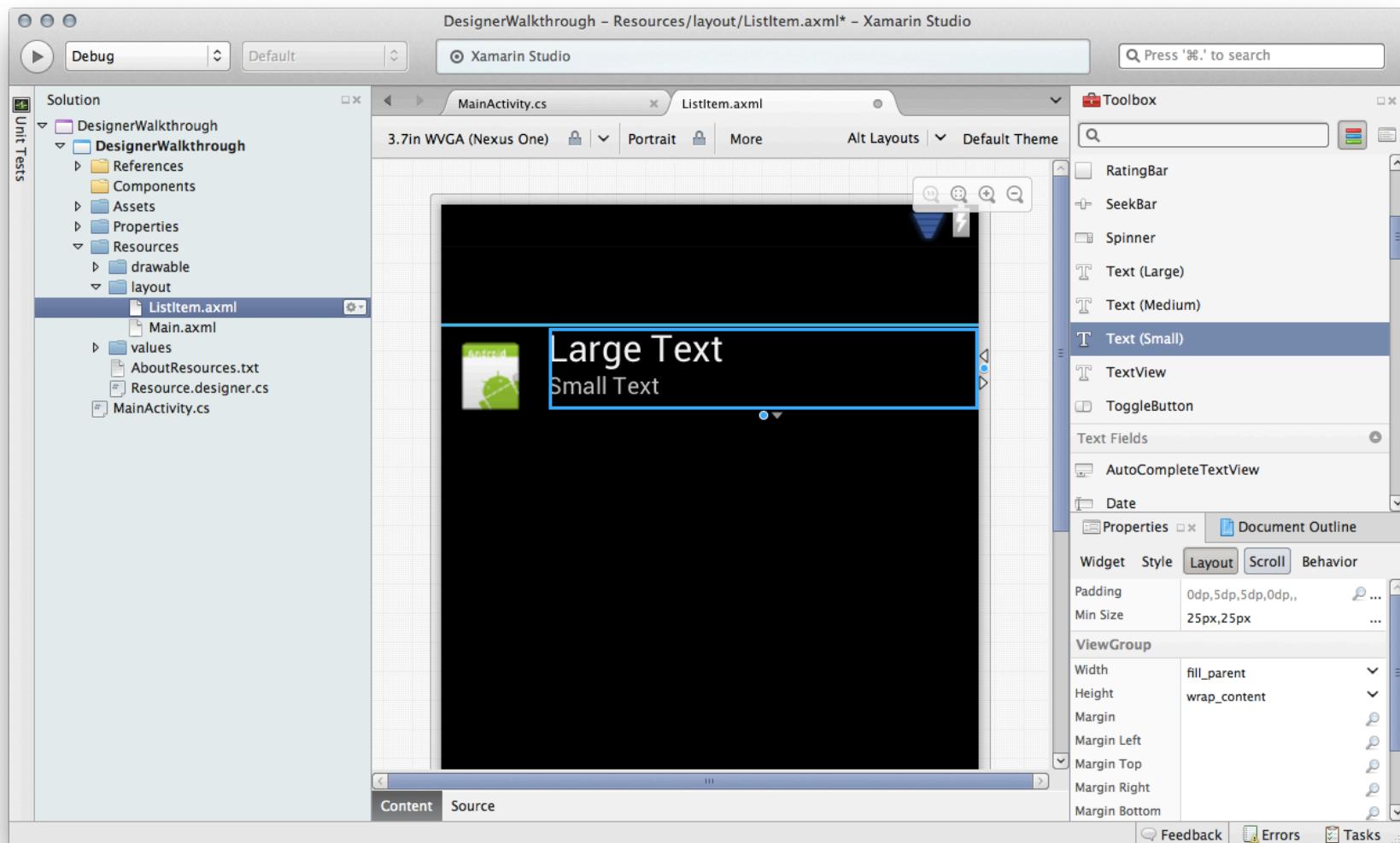
Native SDK++...Xamarin Studio



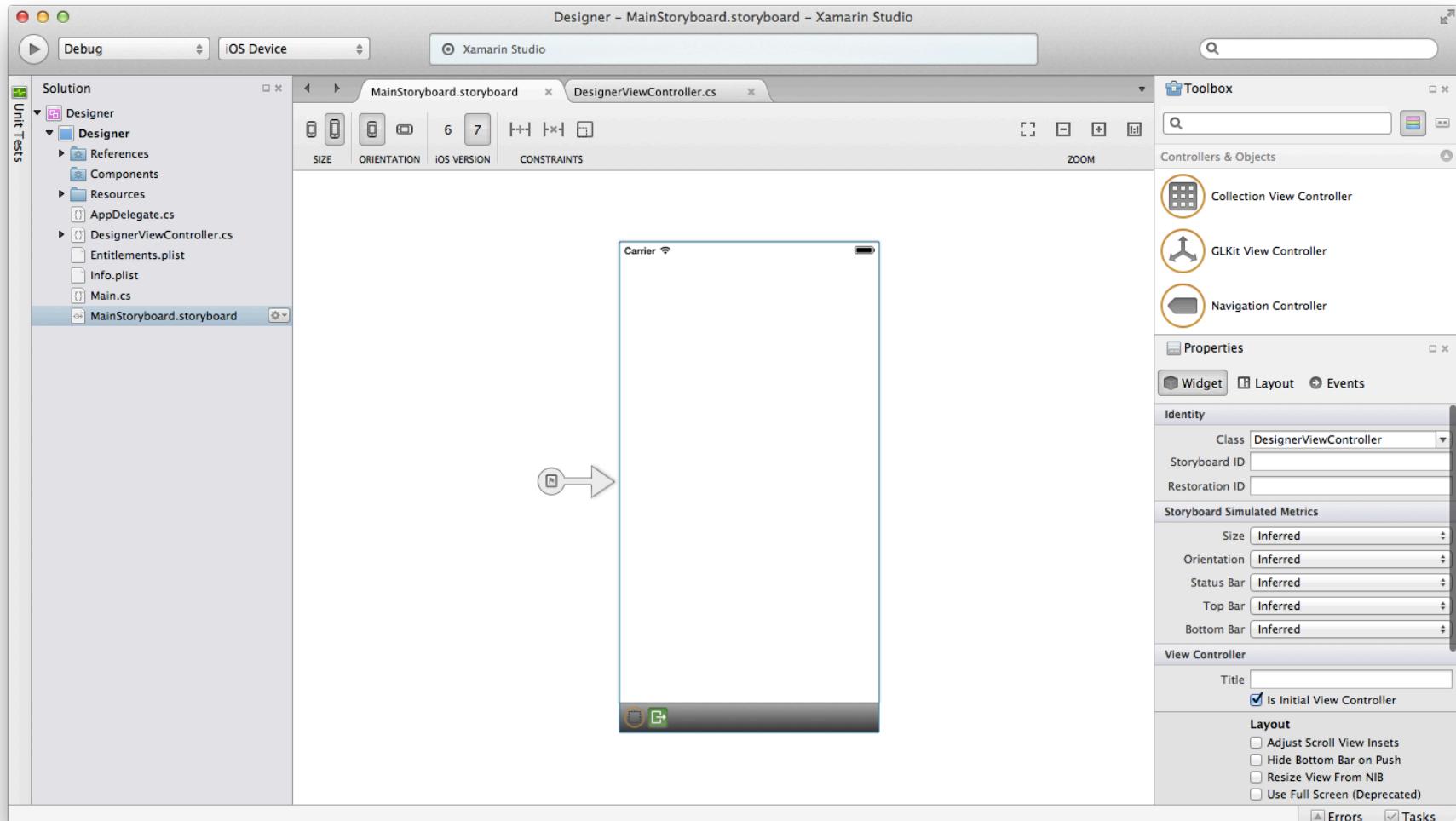
The screenshot shows the Xamarin Studio interface with the title bar "XamarinStore.iOS - ViewControllers/ProductListViewController.cs - Xamarin Studio". The toolbar includes "Debug" and "Default" buttons, and the search bar contains "Xamarin Studio". The main window has a "Solution" sidebar on the left listing project components like XamarinStore (master), Shared, XamarinStore.Droid, and XamarinStore.iOS. The "ViewControllers" folder under XamarinStore.iOS contains several files: BasketViewController.cs, LoginViewController.cs, ProcessingViewController.cs, ProductDetailViewController.cs, ProductListViewController.cs (the current file being edited), ShippingAddressViewController.cs, and StringTableViewController.cs. The "Views" folder contains AutoCompleteTextEntry.cs, BadgeView.cs, BasketButton.cs, and BottomButtonView.cs. The central editor area displays the code for ProductListViewController.cs:

```
1 using System;
2 using System.Drawing;
3 using System.Collections.Generic;
4
5 using MonoTouch.UIKit;
6 using MonoTouch.Foundation;
7
8 namespace XamarinStore.iOS
9 {
10    public class ProductListViewController : UITableViewController
11    {
12        const int ProductCellRowHeight = 300;
13        static float ImageWidth = UIScreen.MainScreen.Bounds.Width * UIScreen.MainScreen.Scale;
14
15        public event Action<Product> ProductTapped = delegate {};
16
17        ProductListSource source;
18
19        public ProductListViewController ()
20        {
21            Title = "Xamarin Store";
22
23            // Hide the back button text when you leave this View Controller.
24            NavigationItem.BackBarButtonItem = new UIBarButtonItem ("", UIBarButtonItemStyle.Plain, handler: n
25            TableView.SeparatorStyle = UITableViewCellStyle.SeparatorStyle.None;
26            TableView.RowHeight = ProductCellRowHeight;
27            TableView.Source = source = new ProductListSource (products => {
28                ProductTapped (products);
29            });
30
31            GetData ();
32        }
33    }
}
```

Native SDK++...Android Designer

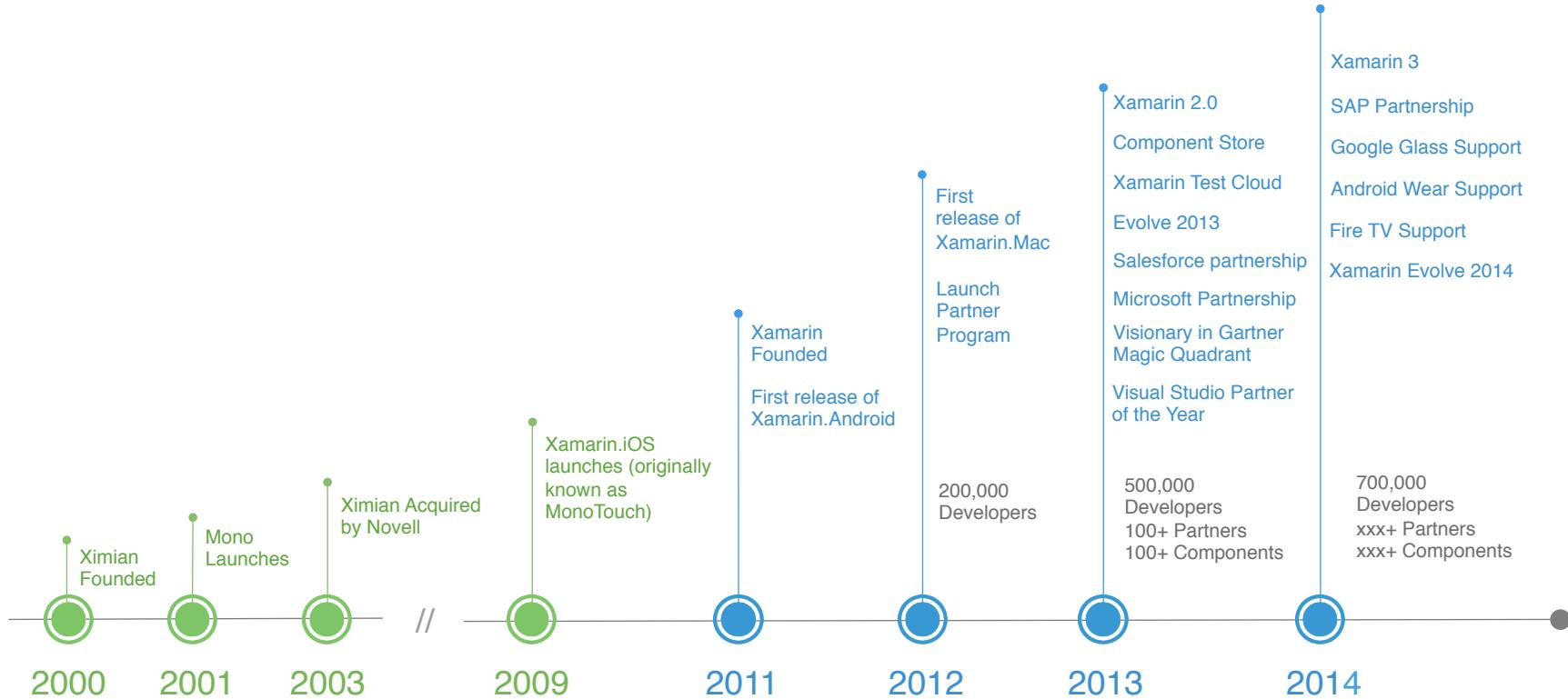


Native SDK++...iOS Designer





Xamarin

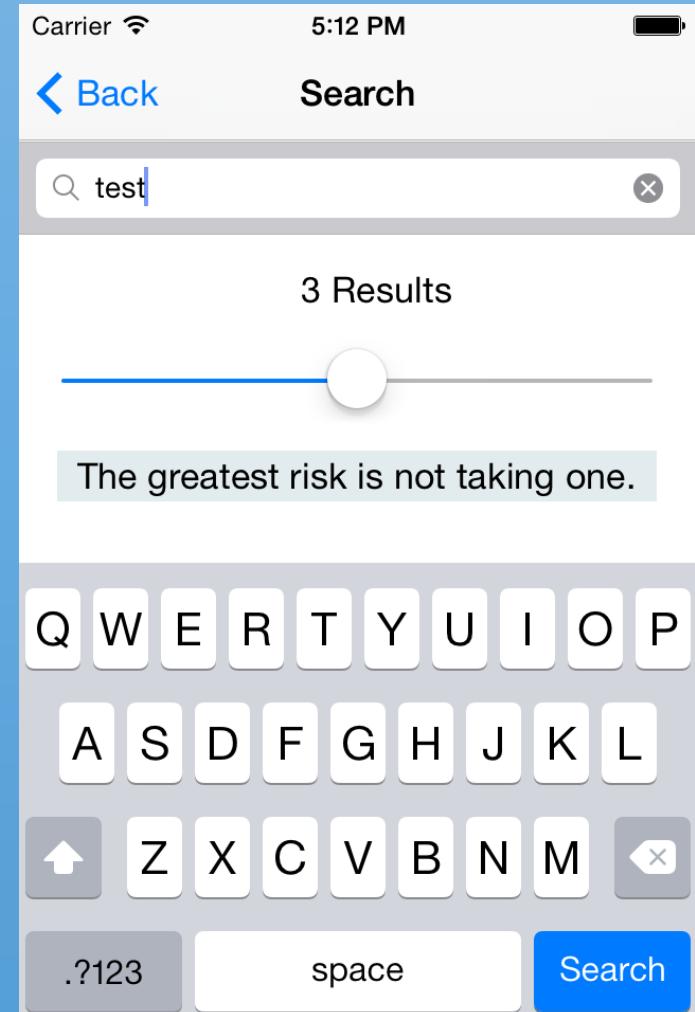
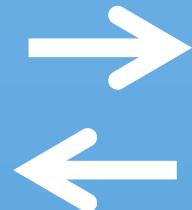
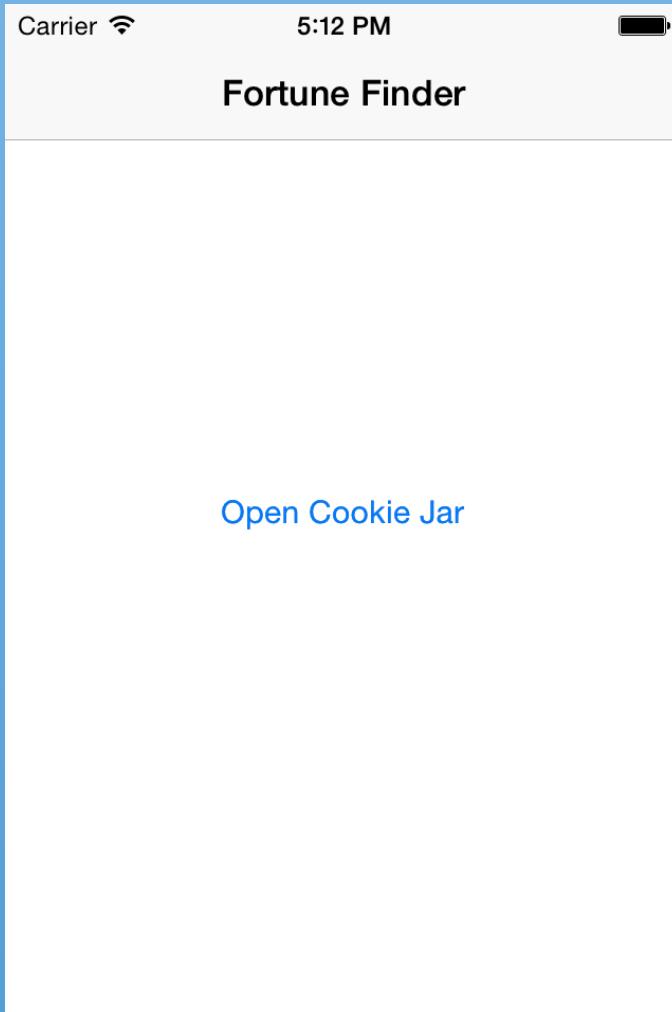


> 700,000 registered developers

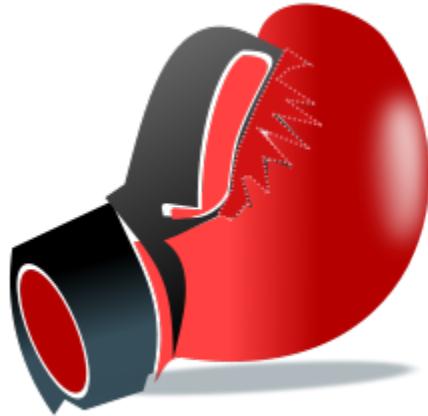
//TODO:

- 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...



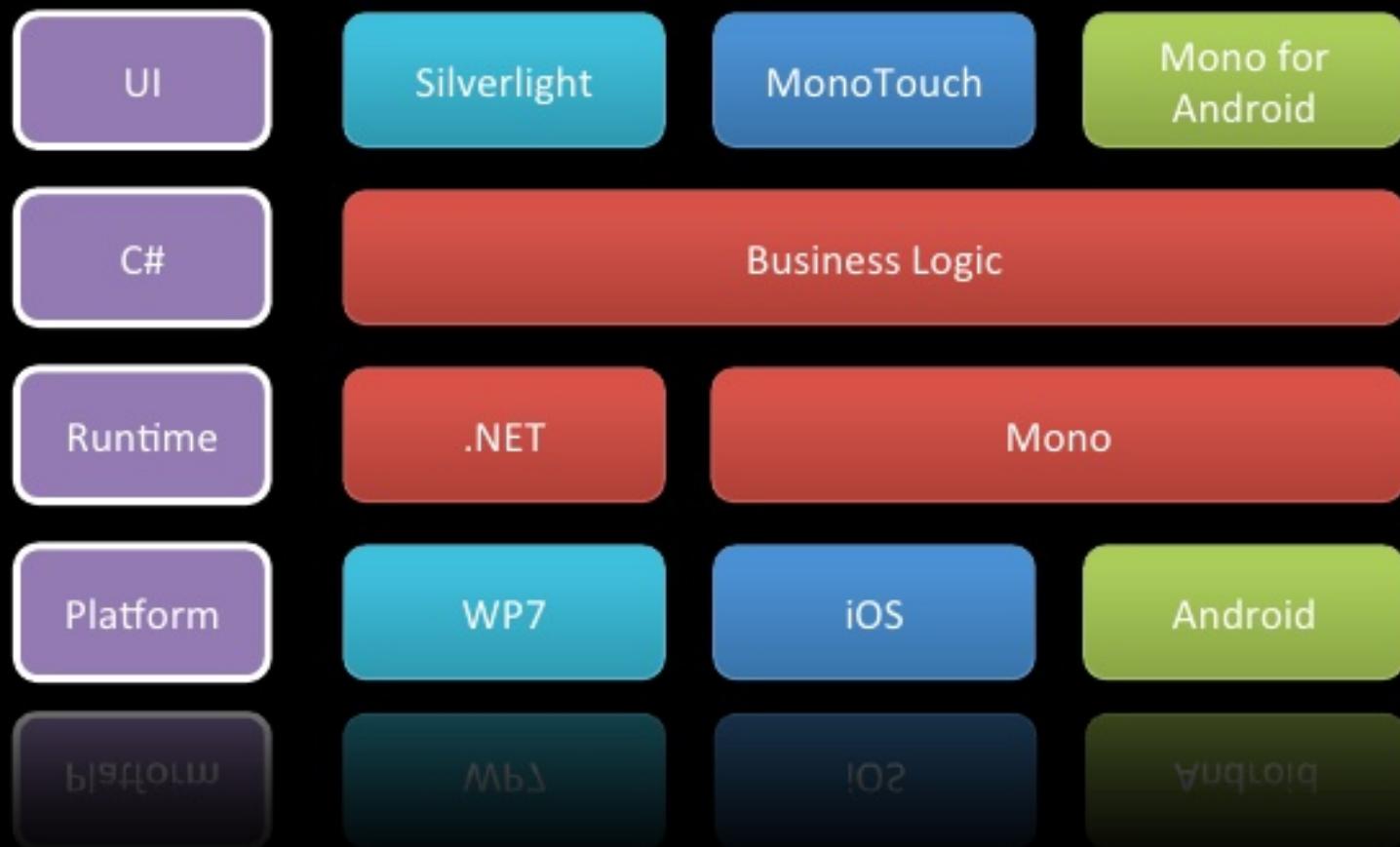
iOS C# UI

Android C# UI

Windows C# UI

Shared App Logic

Architecture

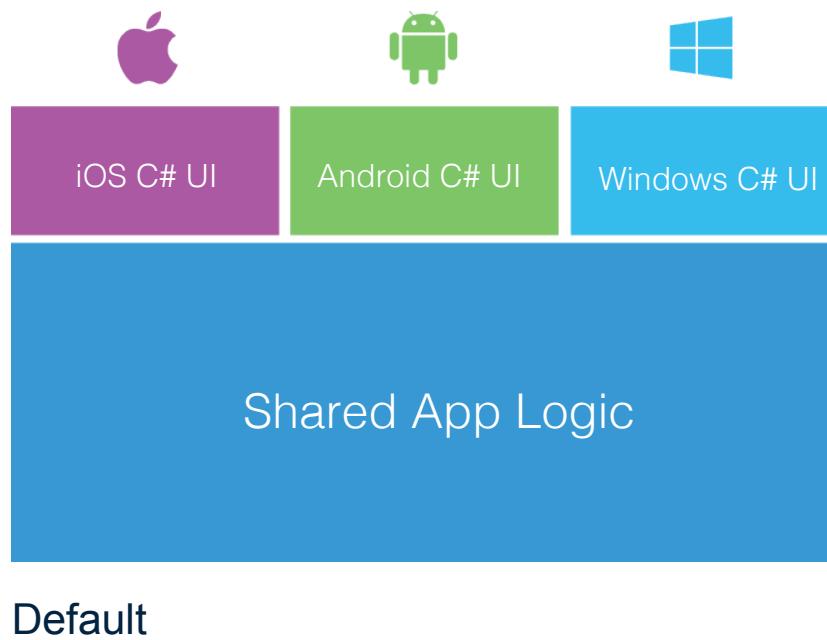


Xamarin

“Traditional” Approach



Default v. 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") }}}
};

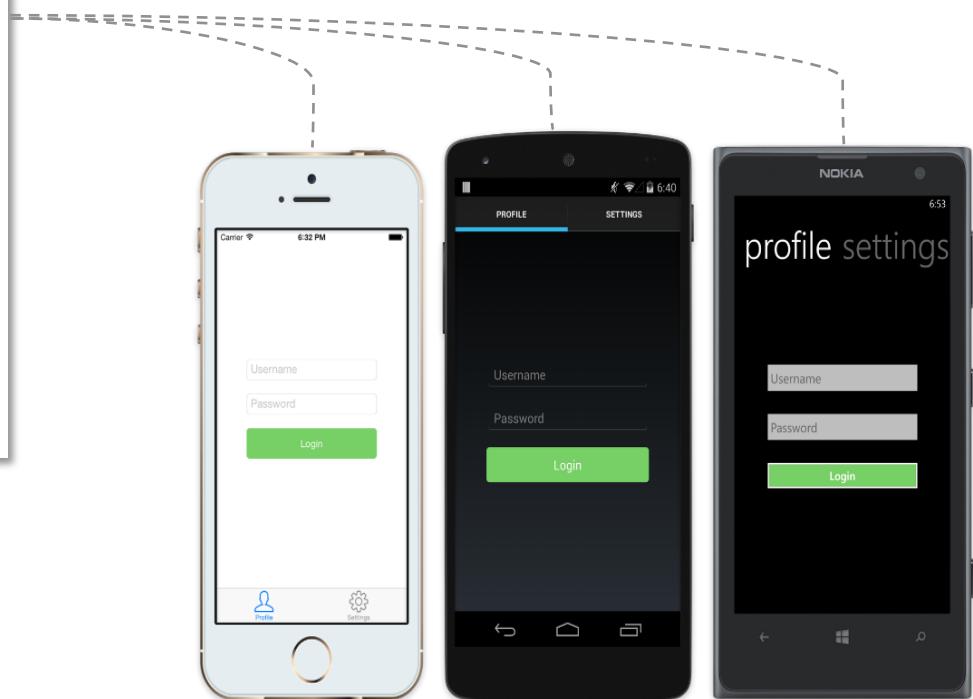
var settingsPage = new ContentPage {
    Title = "Settings",
    Icon = "Settings.png",
    (...)

};

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



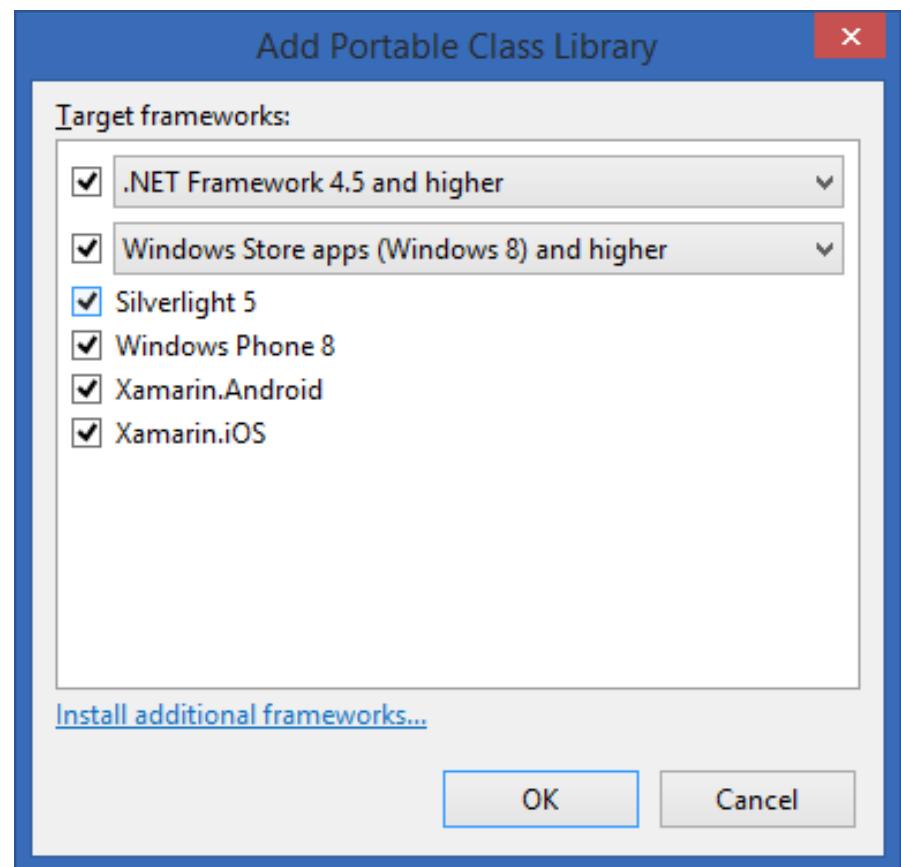
Solution Structuring for Core Code

- 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 supported 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	-	-	-	-	-



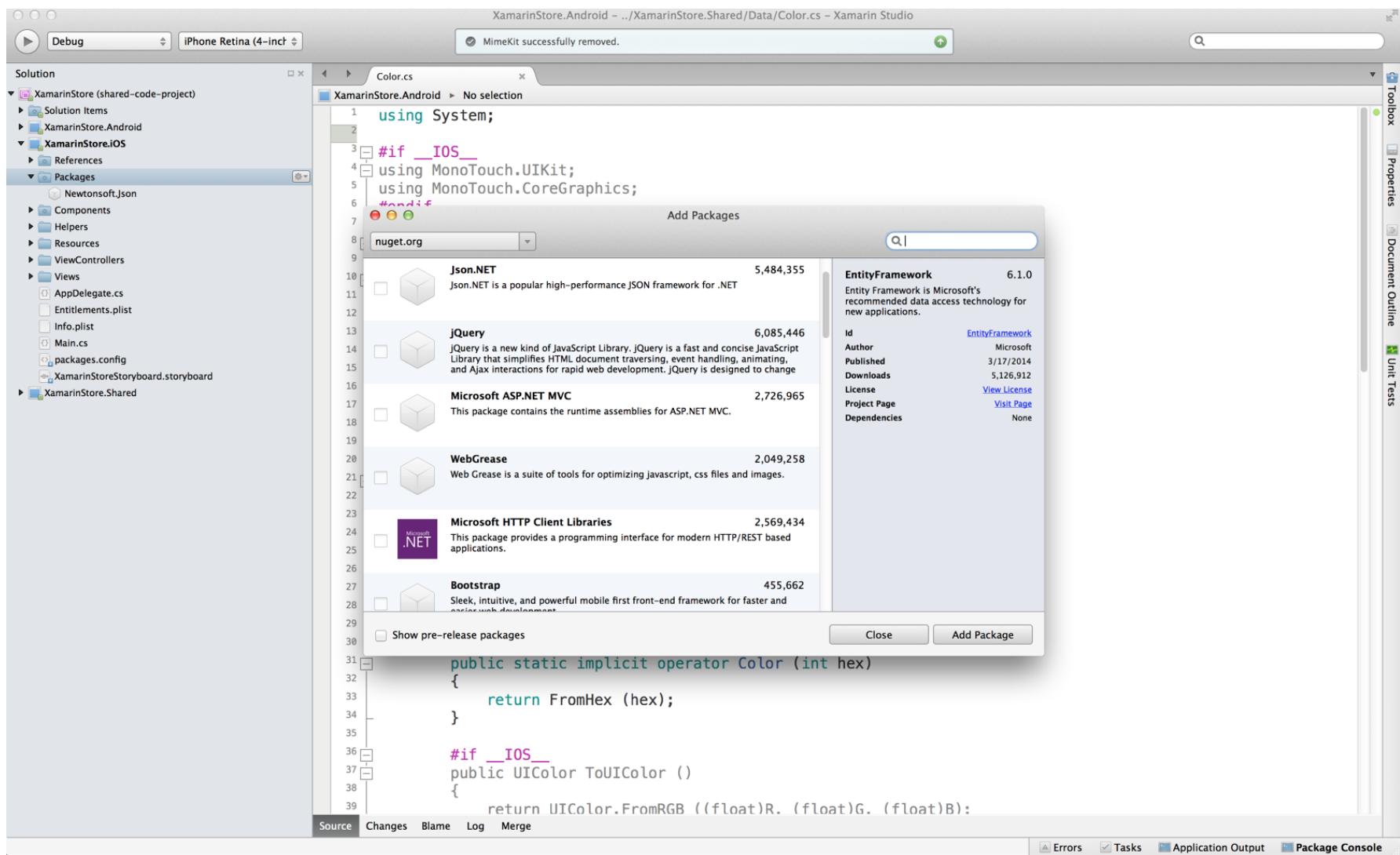
Find Similar UI Patterns

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

The screenshot shows the Xamarin Components Store interface. The main search bar at the top contains the text "Xamarin Components". Below it, there are buttons for "Hide inactive components", "Suggest a Component", and "Submit a Component". On the left, a sidebar titled "All Components" includes a search bar, a "CATEGORIES" section with links to "All Components", "Cloud Services", "User Interface", "Libraries", "Themes", "Game Development", and "Prime Components", and a "TAGS" section with filters for "iOS", "Android", and "Windows". The main content area displays a list of components. The first component listed is "Azure Mobile Services" by Microsoft, which is free and has 14 ratings. The second component listed is "Auth0 SDK" by Auth0, which is also free and has 1 rating. A modal window is open over the list, showing detailed information about the "Salesforce SDK" by Salesforce. The modal includes the component's name, publisher, category, price (free), and tags (Cloud Services). It also shows a screenshot of a mobile application interface for "Authenticate" with a Salesforce logo and user input fields. At the bottom right of the modal, there is a green "Add to App" button.

All Components

Xamarin Components

Order by **FEATURED** DOWNLOADS NAME FRESH

Hide inactive components Suggest a Component Submit a Component

Search Components

CATEGORIES

All Components

Cloud Services

User Interface

Libraries

Themes

Game Development

Prime Components

TAGS

• iOS • Android • Windows

Azure Mobile Services by Microsoft
★★★☆☆ 14 ratings
Store data in the cloud, authenticate users, and send push notifications. Free

Auth0 SDK by Auth0
★★★★★ 1 rating
Add login with Google, Facebook, Twitter, Amazon, GitHub, LinkedIn; Windows Azure AD, Google Free

Salesforce SDK by Salesforce
0 rating Compatible with iOS Android
Integrate Salesforce into your mobile apps.

Getting Started License API Docs Website Add to App

Carrier 3:57 PM

Authenticate

salesforce

demo@mycompany.com

Log in to Salesforce

Remember User Name

Forgot your password?



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, & Gaming

- 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, CGBle
```

- iOS: Objective-C, C++ libraries
 - <https://github.com/mono/monotouch-bindings>
- 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.

//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(UIViewController 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;
```

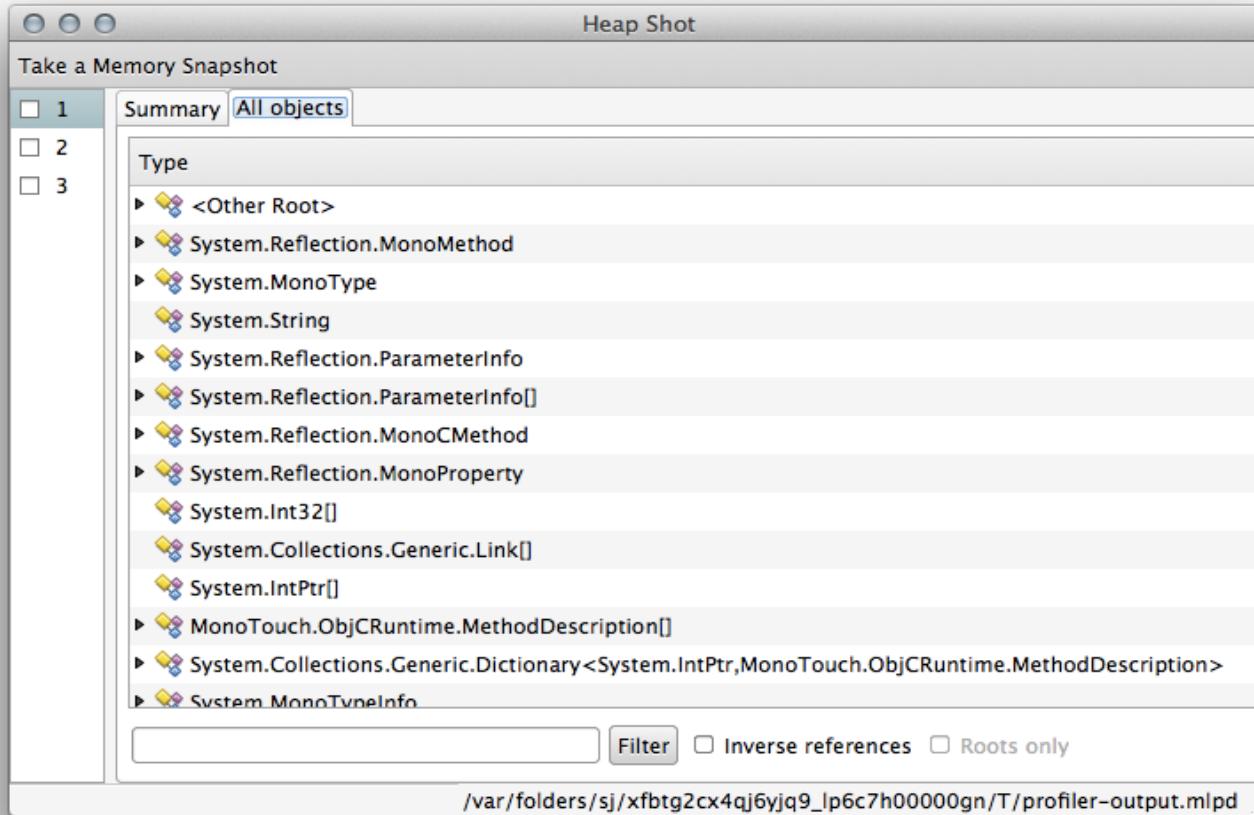
- Manual alternative

```
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 \$25 / month paid monthly	BUSINESS \$83 / month paid annually (\$999 / year)	ENTERPRISE \$158 / month paid annually (\$1899 / year)
Permitted Use	Individual	Individual	Organization	Organization
Subscription Type	N/A	Monthly	Annual	Annual
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
(a \$600 Mac mini works just fine!)

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

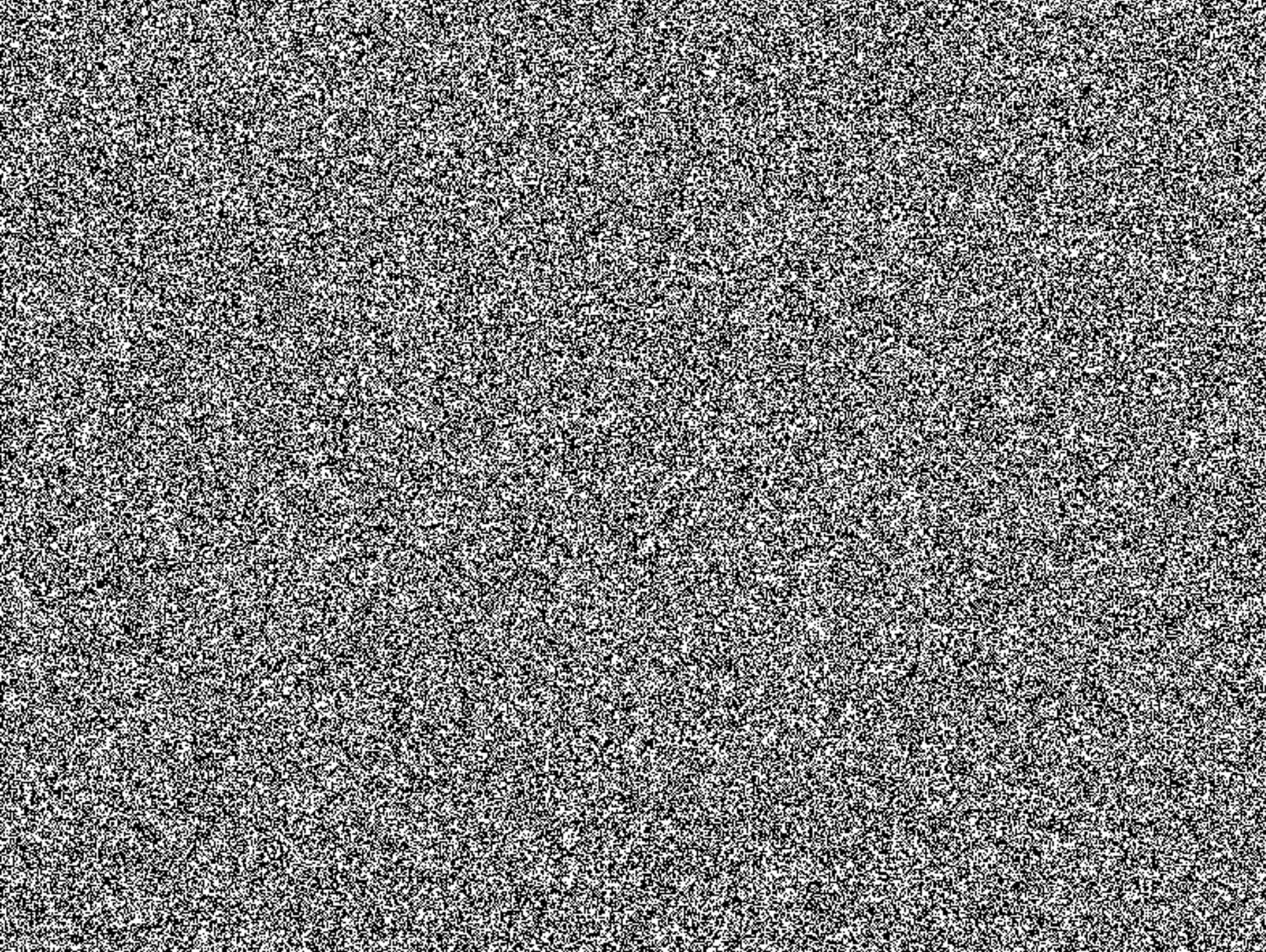
Helpful Tips

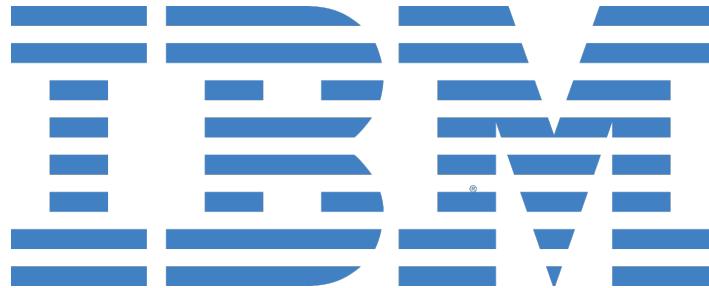
- Pick one platform to do first
- Don't just Google for "Xamarin"
 - The entire native knowledge base is usable
- Understand MonoTouch and MonoDroid monikers
 - Historical evolution of naming
 - MonoTouch == Xamarin.iOS
 - MonoDroid == Mono for Android == Xamarin.Android

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 **Xamarin**

Related links listed on the ThatConference session page
<https://www.thatconference.com/sessions/session/2240>

Sample code and slides on GitHub

<https://github.com/DennisWelu/presentations>

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

@DennisWelu - DennisWelu@MotisConsulting.com