

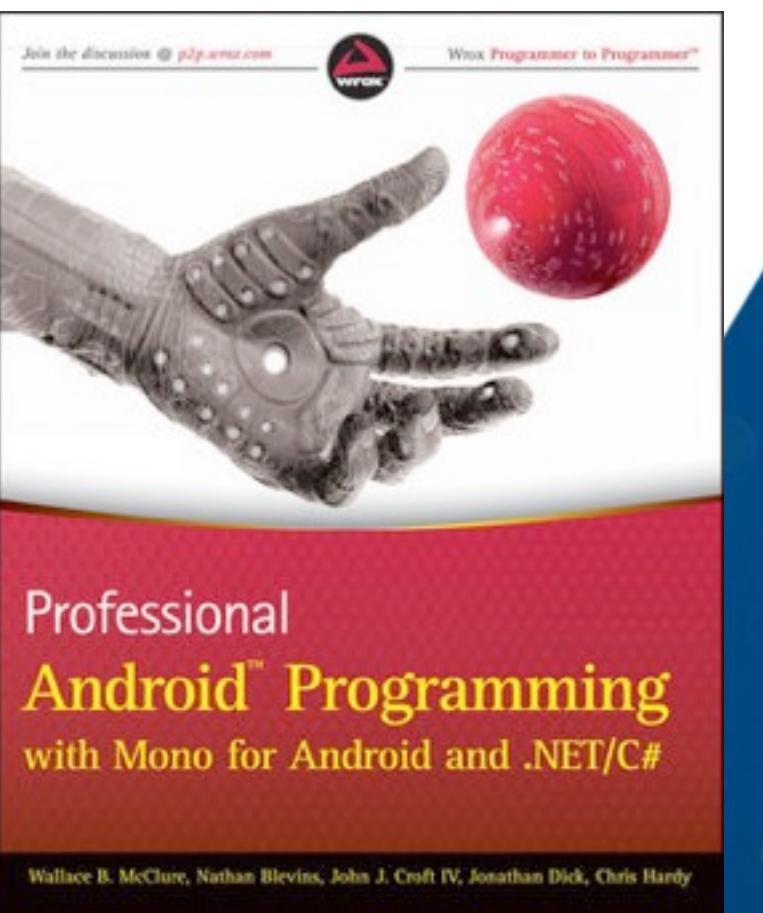


Native Cross Platform Mobile with Xamarin and C#

Jonathan Dick

DBA / Developer
Xamarin Insider

jondick@gmail.com
[@redth](https://twitter.com/redth)





I've Evolved.



Xamarin Studio



Visual Studio



Xamarin.iOS



Xamarin.Android



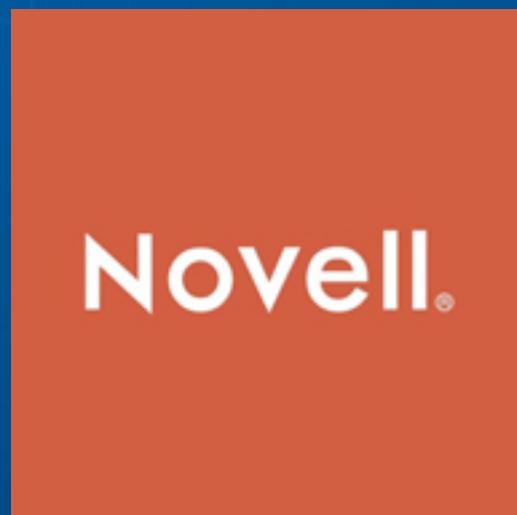
Xamarin.Mac



Delight Developers!

Brief History Lesson

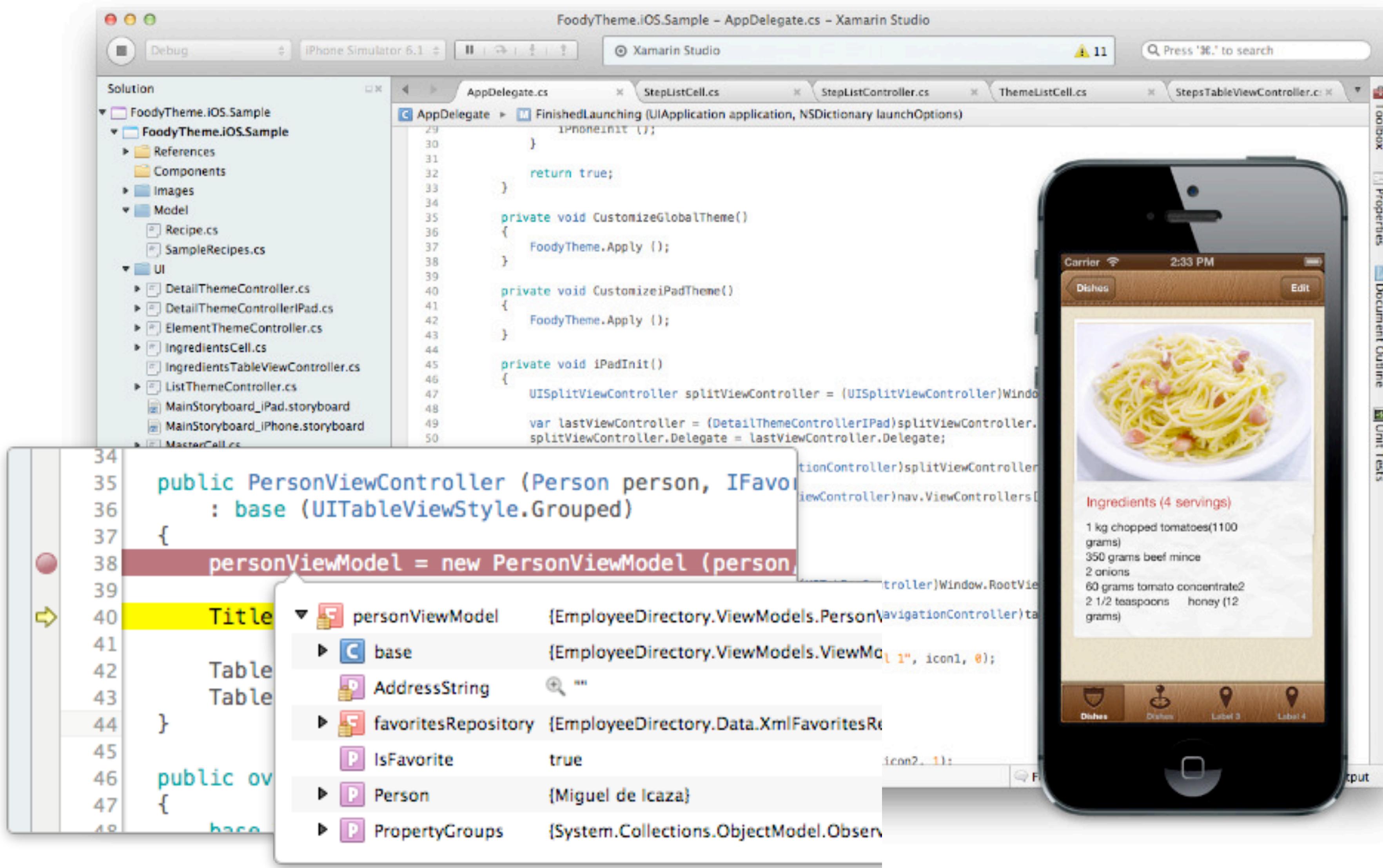
- Mono Project
 - Open Source .NET for ALL
- Novell originally owned:
 - MonoTouch
 - Mono for Android
- Attachmate bought Novell
- Xamarin Founded
 - Awesomeness ensued



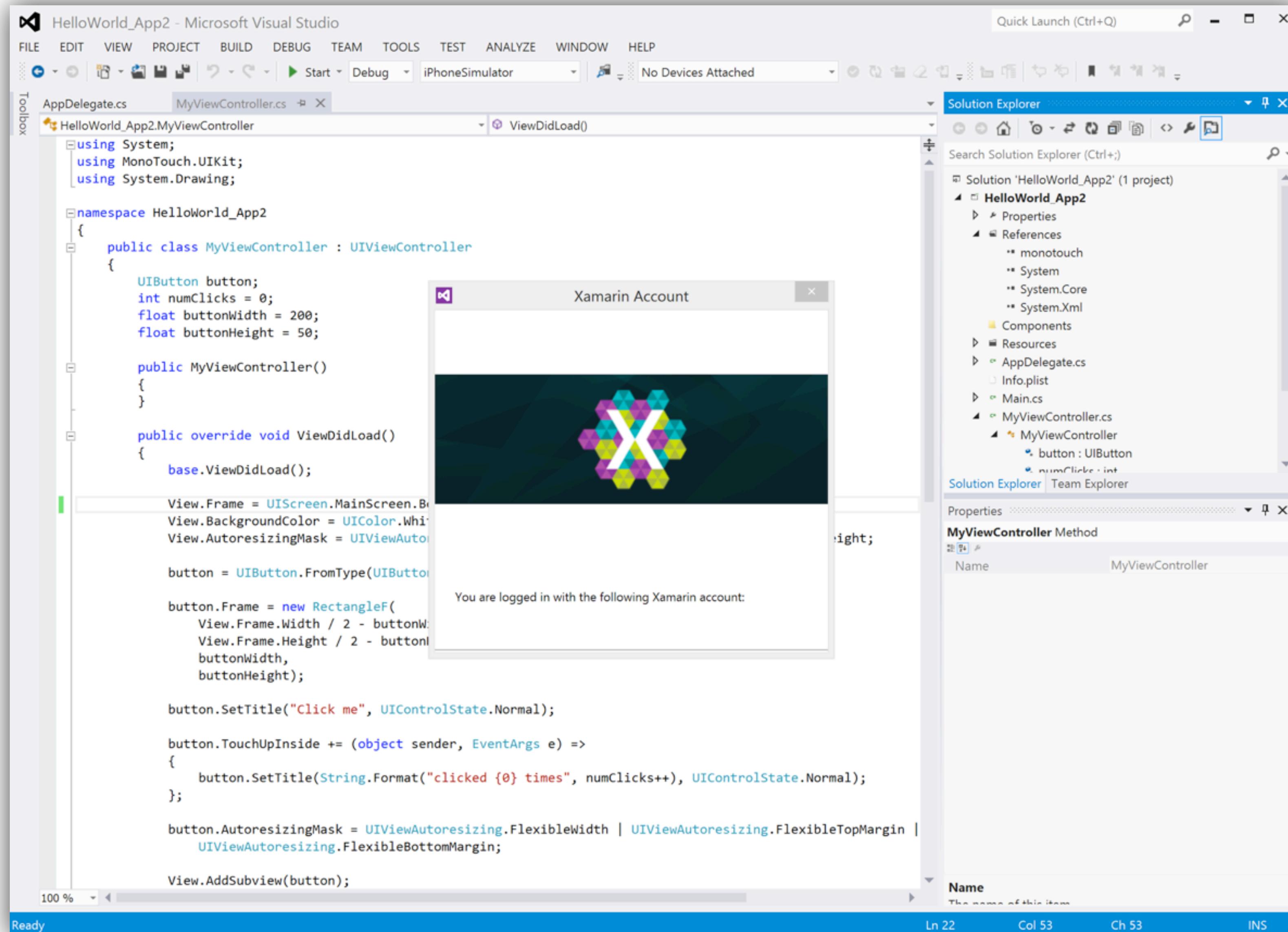
Why Xamarin?

- C# all the things!
- *REAL* Native Apps
- Native UI, Native API's
- High Performance
- Code Sharing / Skill Reuse
- Debugging
- Excellent Documentation & Resources

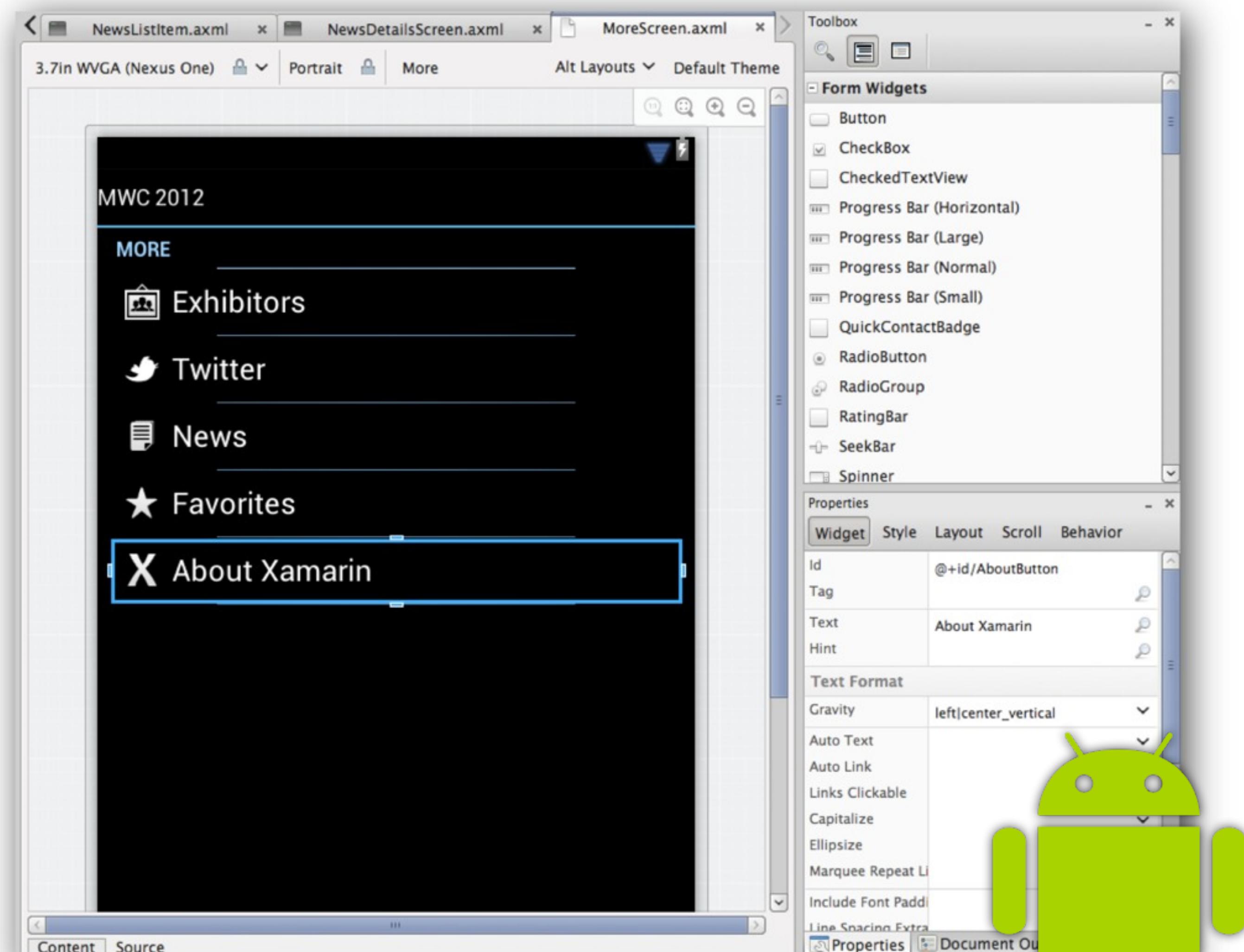
Xamarin Studio



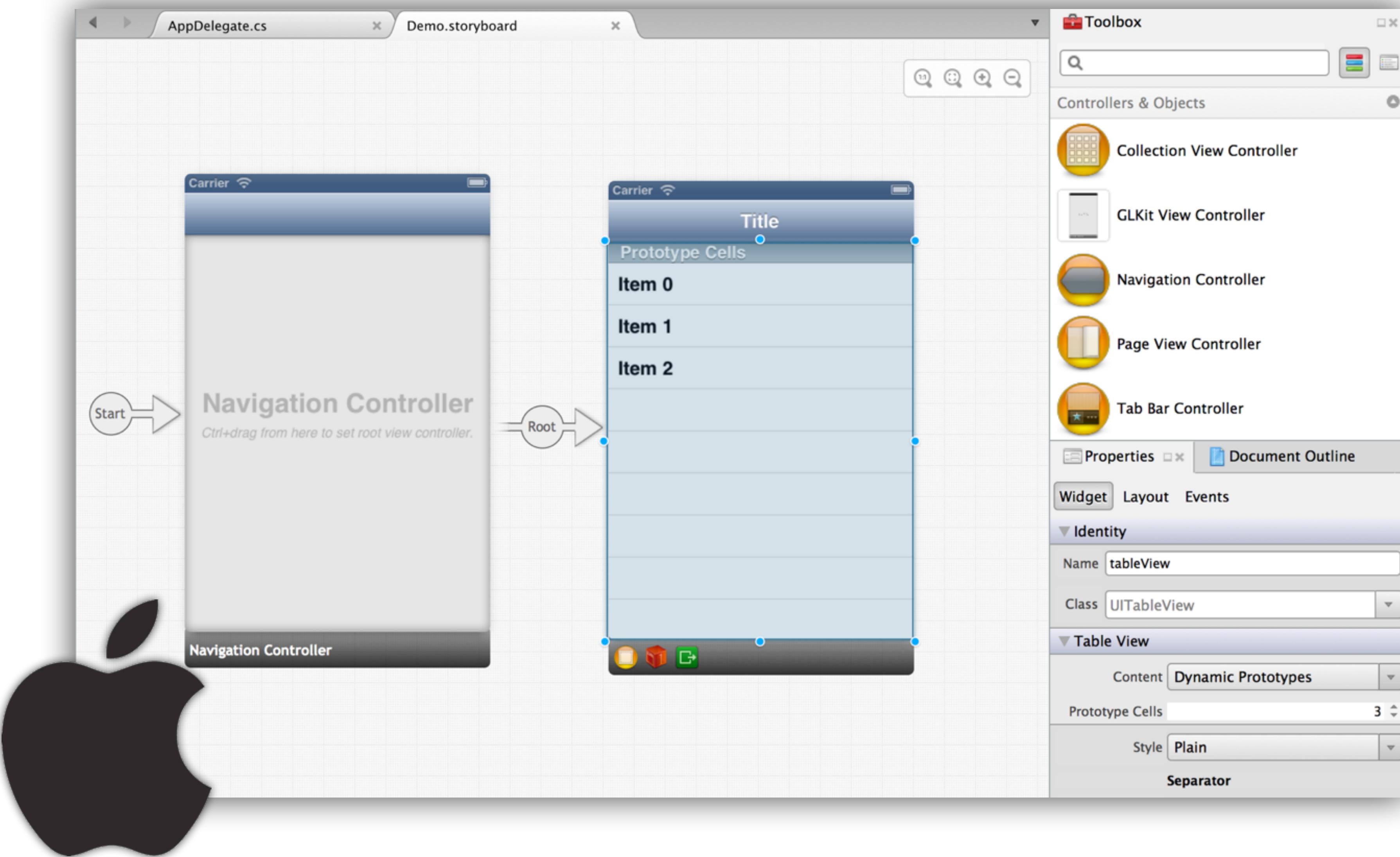
Visual Studio Support



Android Designer



Xamarin iOS Designer



Why C# is Awesome

- Generics
- Linq
- Async / Await
- Anonymous Methods
- Action / Func
- Lambda's

```
var hahs = from c in kittens  
           where c.KittenType == KittenType.Lolcat  
           && c.HazCheezburger  
           orderby c.FunninessLevel  
           select c;
```

```
async void SpewCatz()  
{  
    var http = new HttpClient();  
  
    var resp = await http.GetAsync("http://icanhas.cheezburger.com");  
  
    resp.EnsureSuccessStatusCode();  
  
    var html = await resp.Content.ReadAsStringAsync();  
  
    using (var stream = new StreamWriter(File.OpenWrite(@"C:\catz.html")))  
        await stream.WriteAsync (html);  
}
```

C#... it just feels right!

Creating attributed strings

Objective-C

```
CFStringRef keys[] = {  
    kCTFontAttributeName,  
    kCTForegroundColorAttributeName  
};  
  
CFTypeRef bval[] = {  
    cfListLineCTFontRef,  
    CGColorGetConstantColor(kCGColorBlack)  
};  
  
attr = CFDictionaryCreate (kCFAllocatorDefault,  
    (const void **) &keys, (const void **) &bval,  
    sizeof(keys) / sizeof(keys[0]), &kCFTypeDictionaryKeyCallBacks,  
    &kCFTypeDictionaryValueCallBacks);  
  
astr = CFAttributedStringCreate(kCFAllocatorDefault, CFSTR("Hell
```

C# with Xamarin

```
var attrs = new CFStringAttributes {  
    Font = listLineCTFont,  
    ForegroundColor = UIColor.Black.CGColor  
};  
  
var astr = new NSAttributedString ("Hello World", attrs);
```

C#... it just feels right!

Activity Registration / Button Click

Java

```
<activity android:name=".MainActivity" android:label="@string/app_name">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

findViewById(R.id.button).setOnClickListener(
    new View.OnClickListener()
    {
        @Override
        public void onClick(View v)
        {
            //handle click
        }
    }
);
```

C# with Xamarin

```
[Activity(Label="App Name", MainLauncher = true)]

FindViewById<Button>(Resource.Id.button).Click += delegate
{
    //handle click
};
```

About the Alternatives

	Xamarin	Appcelerator	Sencha
Native Performance	✓		
Full Native UI & Apps	✓		
Write once run everywhere		✓	✓
HTML 5 Support	✓	✓	✓



Write once, run anywhere

**“The biggest mistake we made as a company was
betting too much on HTML5 rather than native”**



Mark Zuckerberg

“There are a few things that are critically missing”

“It’s not that HTML5 isn’t ready; it’s that the ecosystem doesn’t support it”

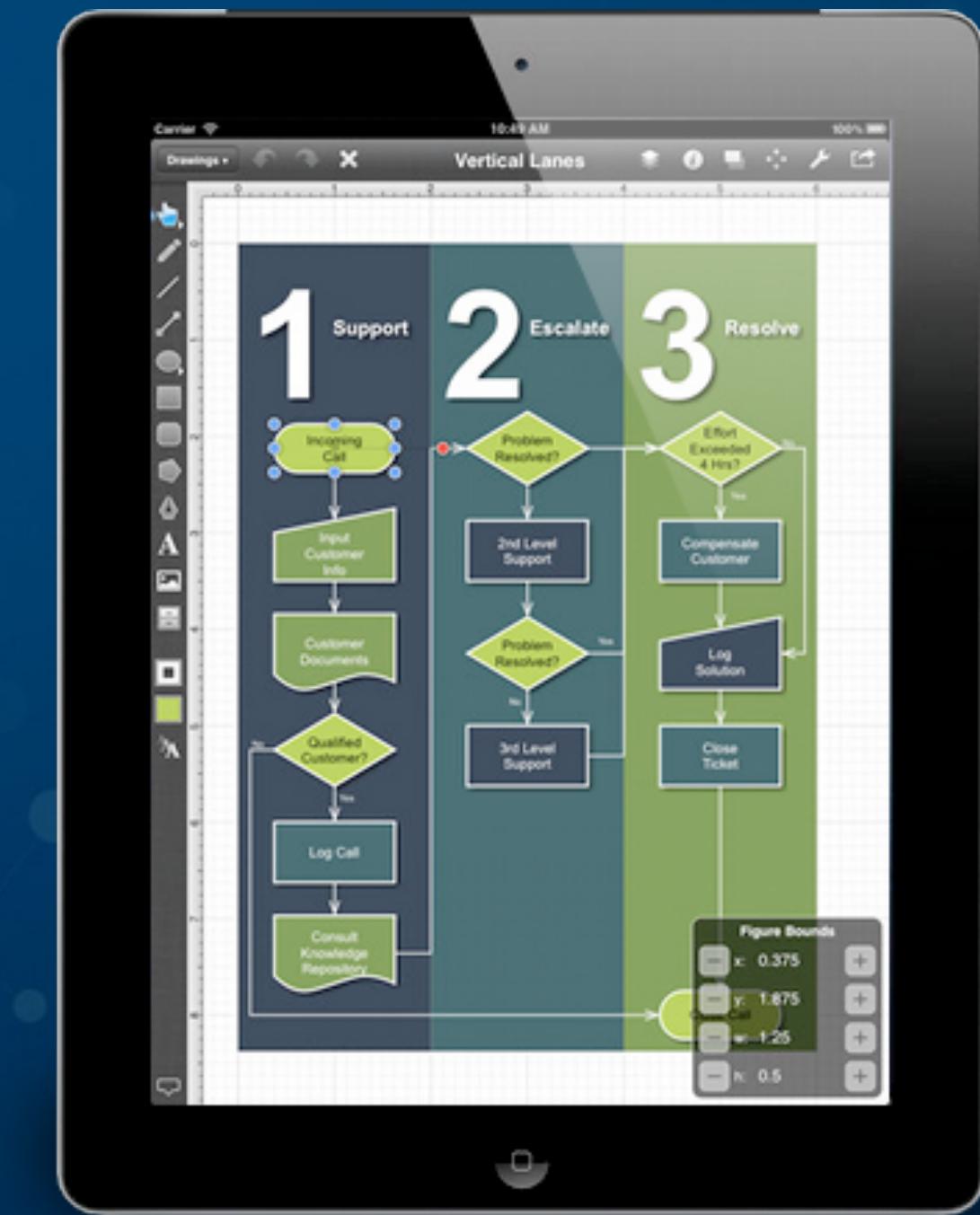
“...getting that smoothness, we felt like we needed native to really do that well.”



Kiran Prasad

Do the Smart Thing

Reuse code where it makes sense, but
leverage each platform's native UX





HOW DOES IT WORK?

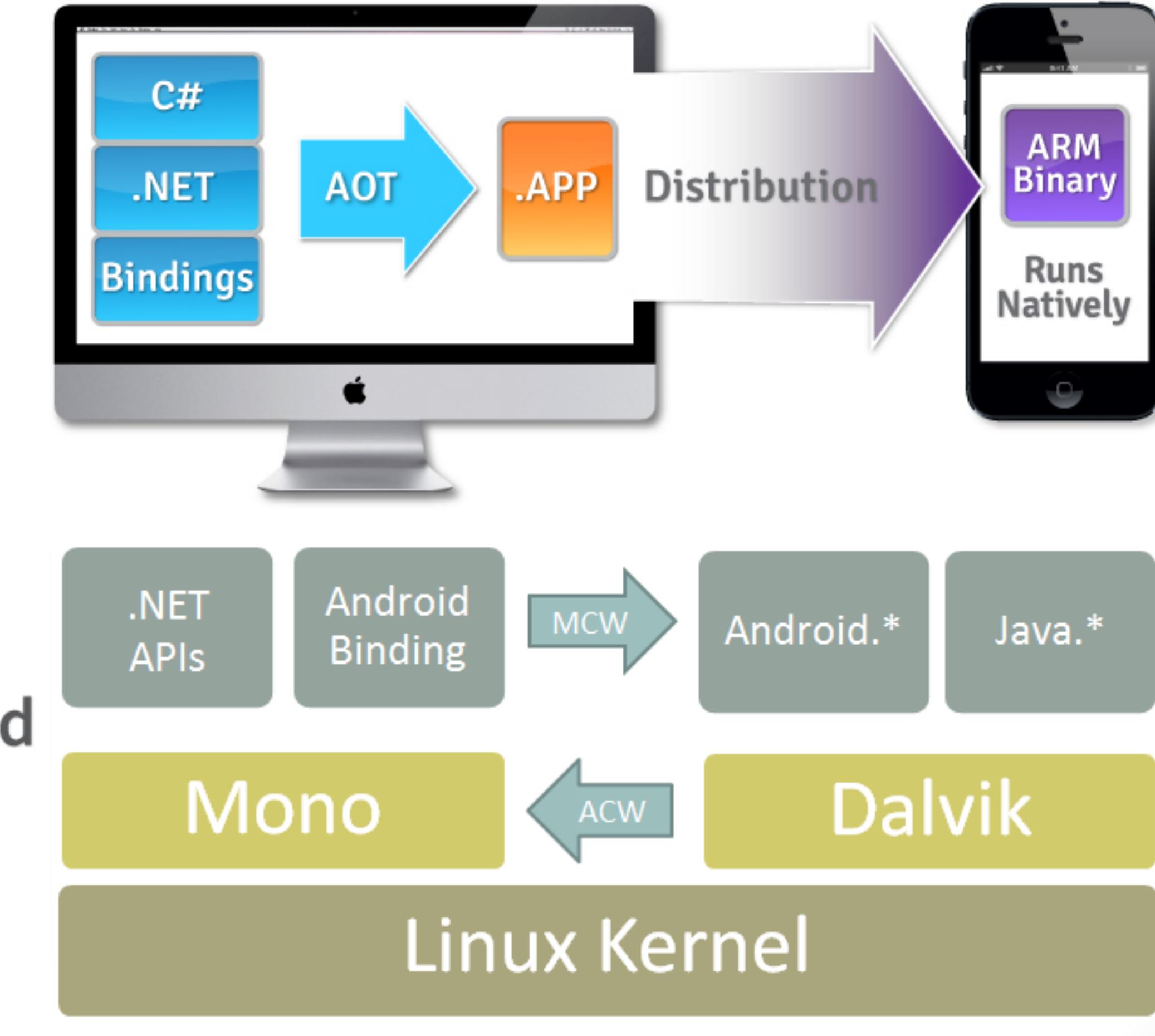
Magic!!!1



But Seriously...



iOS

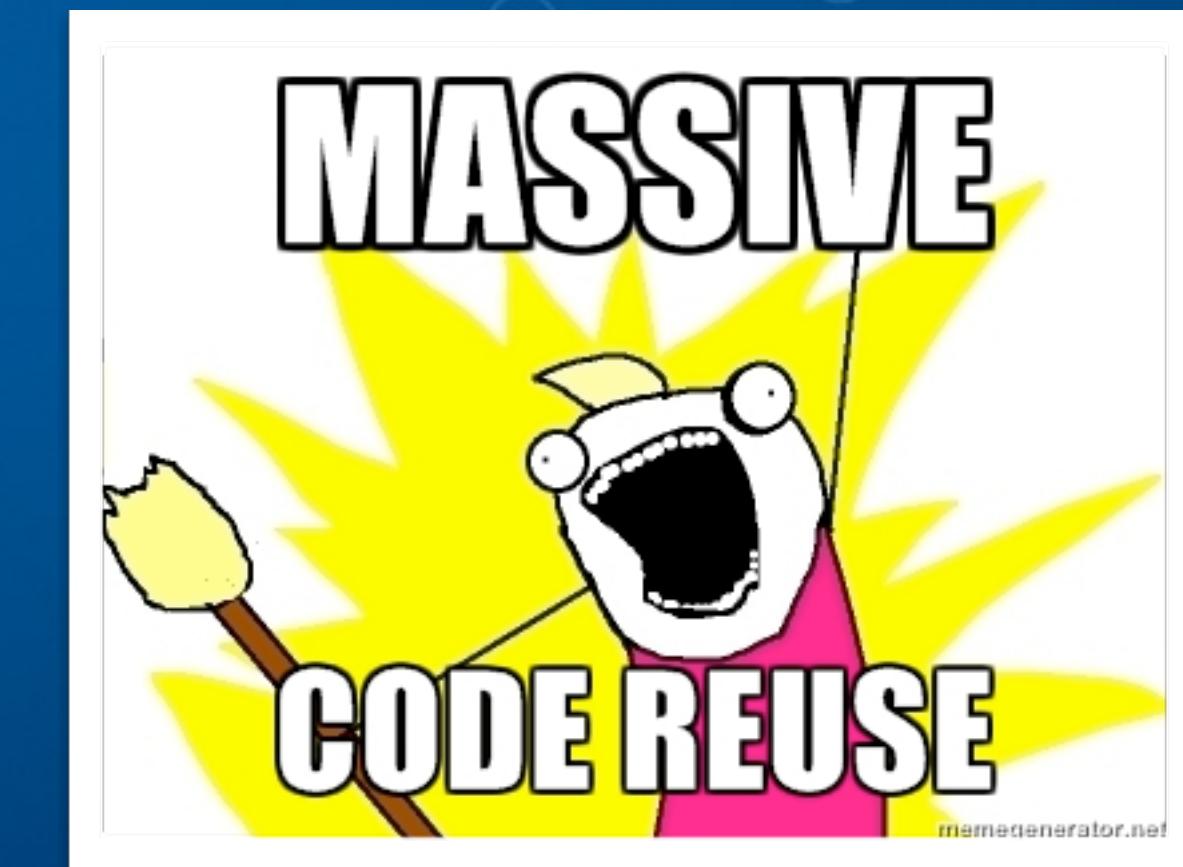


Cross Platform Story



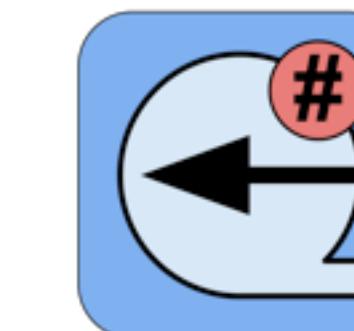
Reuse Code!

- Typical to Reuse 40-60%
 - Web Service calls
 - Business Logic
 - SQLite / Data
- MVVM Pattern
- Games have higher reuse potential



Leverage Existing Awesome

- NuGet
- Open Source



ReactiveUI

Component Store

The screenshot shows the 'Components / Xamarin' page of the Xamarin Developer Center. The main heading is 'Add some awesome to your app.' with the subtext 'Add amazing functionality to your app in minutes with Xamarin Components.' A sidebar on the left contains a search bar and a 'Categories' section with links to 'All Components', 'Cloud Services', 'User Interface', 'Libraries', 'Themes', 'Game Development', and 'Prime Components'. The main content area features a navigation bar with 'Documentation', 'Forums', and 'Components' tabs, along with links to 'Suggest a Component', 'Submit a Component', and 'Login'. Below this is a grid of component cards. The first card is for 'Xamarin.Mobile' by Xamarin, described as a library for common mobile device functionality across iOS, Android, and Windows platforms. The second card is for 'Json.NET' by James Newton-King, a high-performance JSON framework for .NET. The third card is for 'ZXing.Net.Mobile' by Redth, a C#/NET Barcode Scanning Library. Each card includes a star rating, download count (e.g., 9 ratings for Json.NET), and a 'Free' status with platform icons (iOS, Android, Windows).

Components / Xamarin

Xamarin Developer Center

Add some awesome to your app.

Add amazing functionality to your app in minutes with Xamarin Components.

Search Components

CATEGORIES

- All Components
- Cloud Services
- User Interface
- Libraries
- Themes
- Game Development
- Prime Components

Order by

- FEATURED
- DOWNLOADS
- NAME
- FRESH

Xamarin.Mobile by Xamarin

5 stars

Xamarin.Mobile is a library that exposes a single set of APIs for accessing common mobile device functionality across iOS, Android and Windows platforms.

Json.NET by James Newton-King

5 stars 9 ratings

A high-performance JSON framework for .NET.

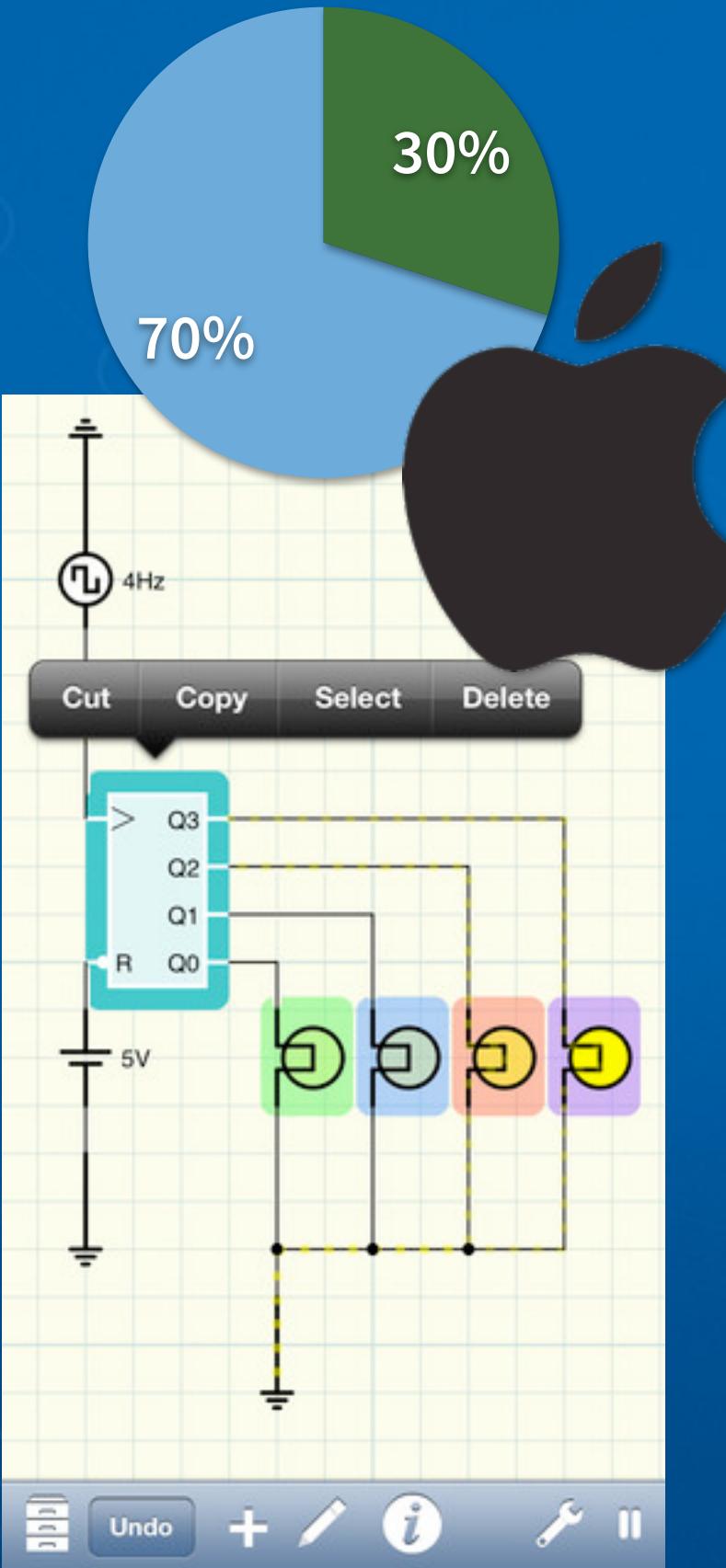
ZXing.Net.Mobile by Redth

5 stars 6 ratings

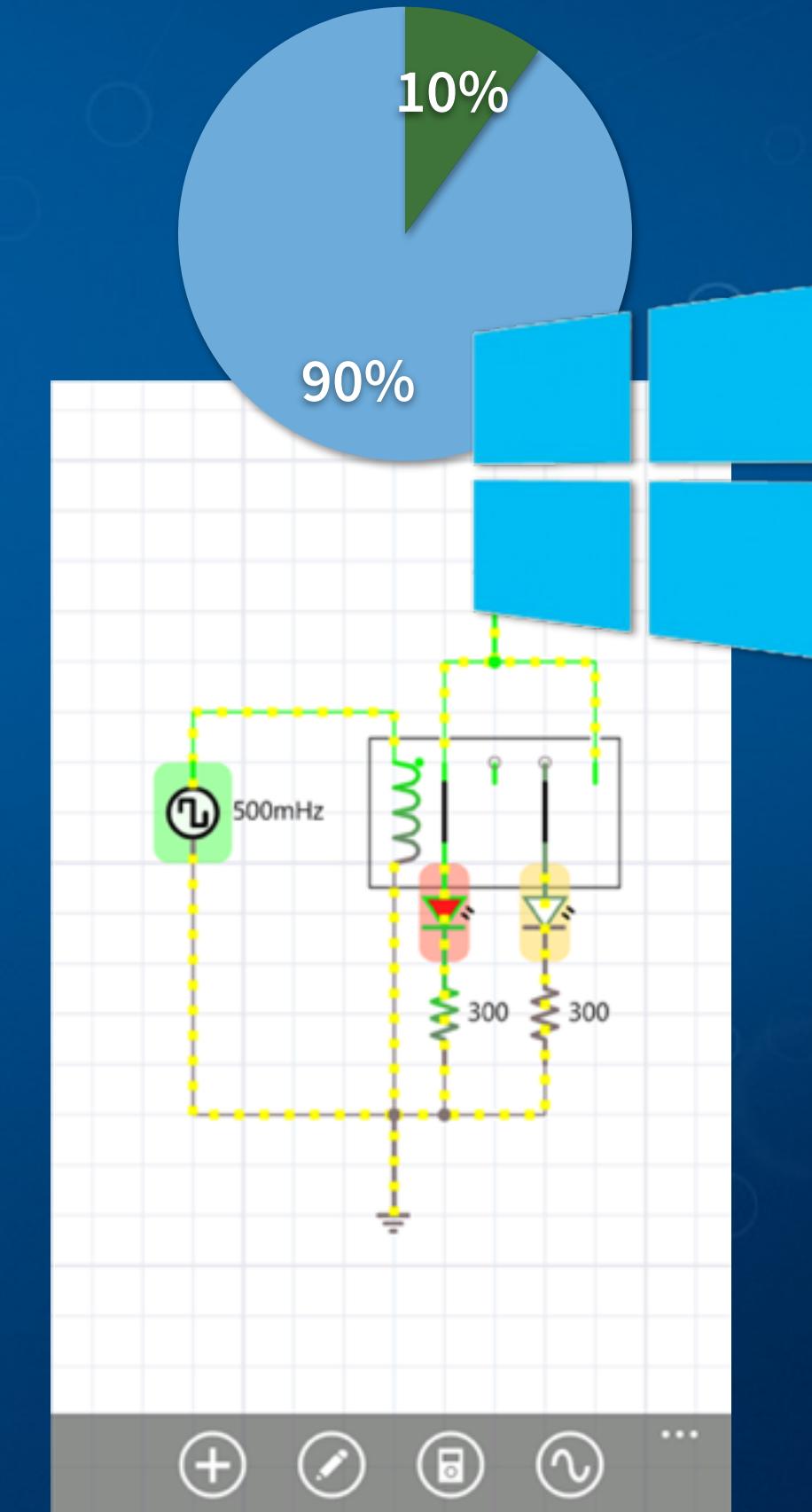
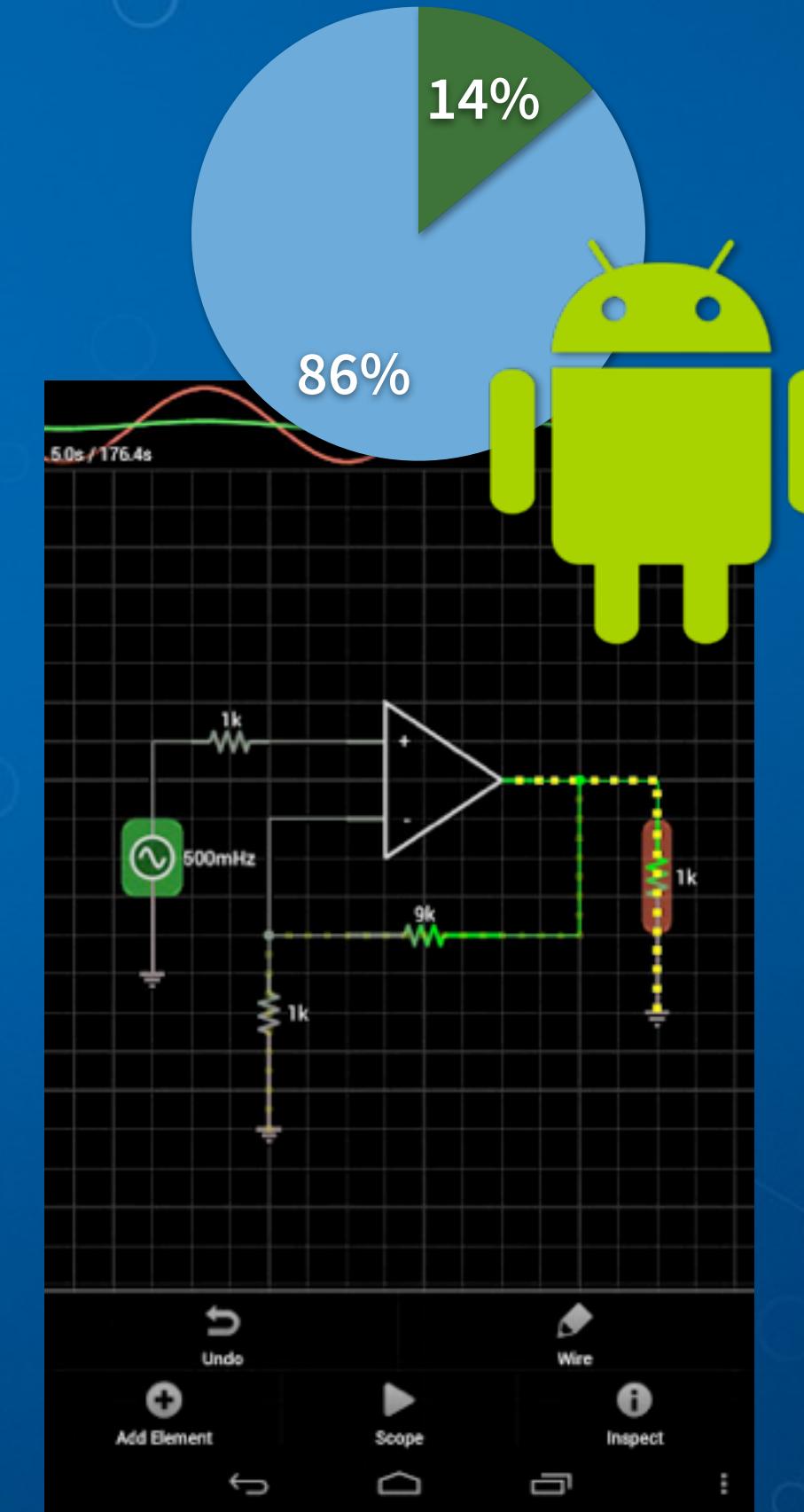
ZXing.Net.Mobile is a C#/NET Barcode Scanning Library.

Code Sharing: iCircuit

Shared



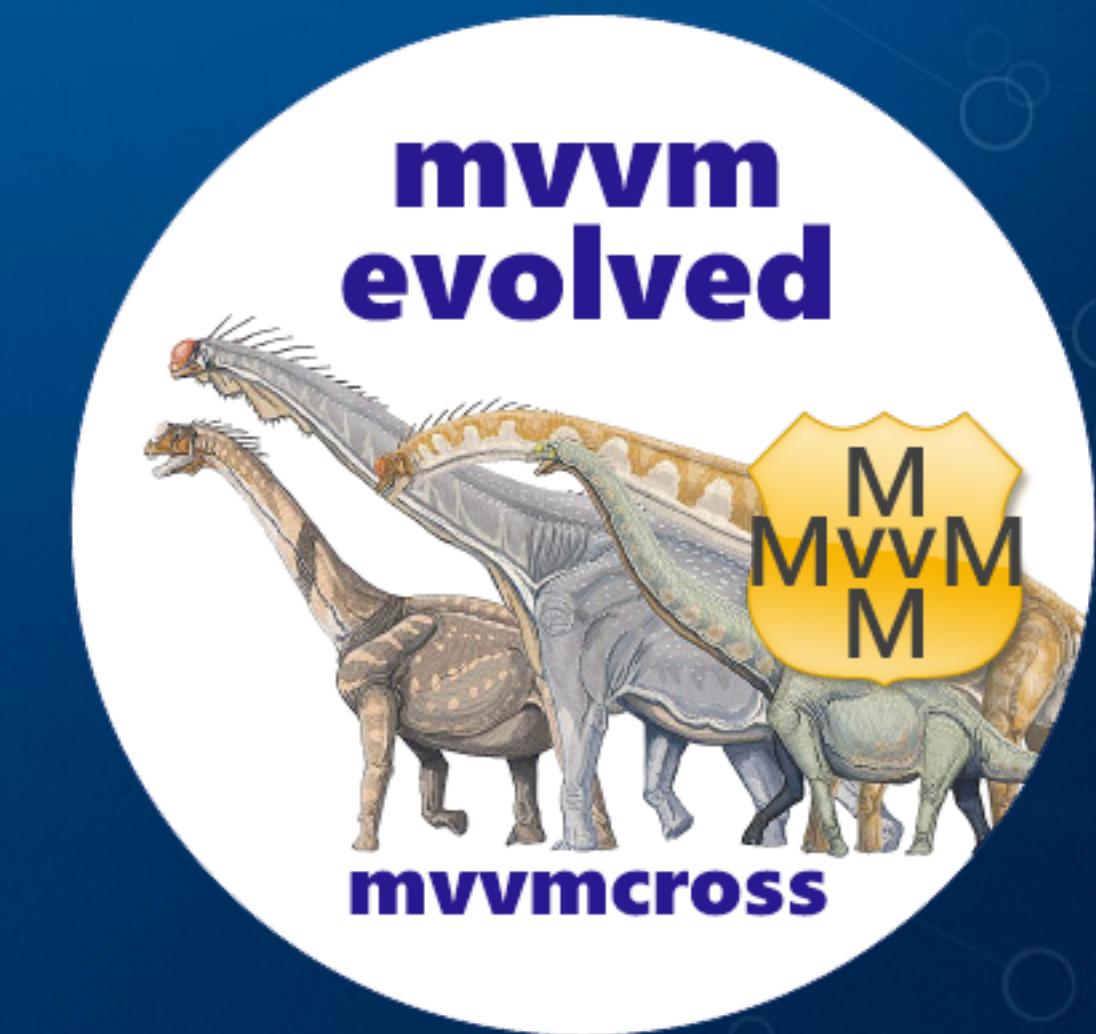
Not Shared



<http://tinyurl.com/icircuit-code-sharing>

#MVVMCross

- Model View View Model Pattern
- Binding to Views
- <https://github.com/slodge/MvvmCross>
- Stuart Lodge (aka ‘Hodges’ or @slodge)



MonoGame

- Open Source Implementation of XNA 4.x
- OpenGL / DirectX
- Runs Everywhere. Really
 - Windows, iOS, Android
 - OUYA, Playstation Mobile
 - Xbox360
- Bastion
- Draw a Stickman Epic
- Infinite Flight



Xamarin.Mobile

- Contacts
- Camera / Image Chooser
- Geo Location
- Available on Component Store



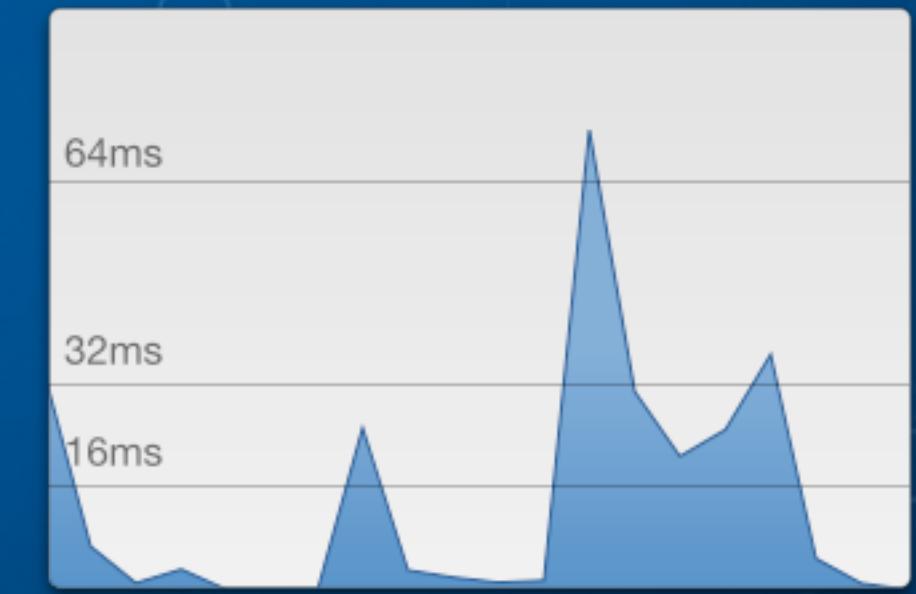


Xamarin Test Cloud



To the (Test) Cloud!

- LessPainful++
- Any iOS or Android App
- Test on 100's of Physical Devices
- Performance Monitoring
- Visual Test Results
- App Explorer Mode
- Behavior Driven Dev
- Write Tests in C# or Ruby



Xamarin Test Cloud

http://test.xamarin.com

Xamarin Test Cloud

My Apps Employee Directory April 16th, 2013, 2:33 AM Upload New Test

Filter tests...

App Launched

Summary 1⚡

Login button pressed in Login screen 1⚡

- App Launched
- Entered "TestUser" into Username entry
- Entered "Password" into Password entry
- Pressed the Login button
- Employees screen appeared.

Ok button pressed in Need Help screen ✓

Xander text pressed in Employees screen ✓

Miguel text pressed in Employees screen ✓

Twitter button pressed in Details screen ✓

Chris text pressed in Employees screen ✓

Phone button pressed in Details screen ✓

Samsung Galaxy S III

Samsung Galaxy S II

Samsung Galaxy Note II

Droid Razr Hd

Samsung Galaxy S

Samsung Galaxy Note

Displaying all devices

Xamarin Test Cloud

http://test.xamarin.com

Xamarin Test Cloud

My Apps Employee Directory April 16th, 2013, 2:33 AM Upload New Test

Filter tests...

Summary 1⚡

Login button pressed in Login screen 1⚡

Ok button pressed in Need Help screen ✓

Xander text pressed in Employees screen ✓

- App Launched
- Entered "TestUser" into Username entry
- Entered "Password" into Password entry
- Pressed the Login button**
- Employees screen appeared.
- Scrolled down
- Tapped text "Xander".
- Xander A. Monkey screen appeared.

Miguel text pressed in Employees screen ✓

Twitter button pressed in Details screen ✓

Chris text pressed in Employees screen ✓

Pressed the Login button

Device Name: Samsung Galaxy S III

Brand: Samsung

Operating System: Android 4.1.2

Screen Resolution: 720 x 1280 pixels

Screen Size: 4.8" (306 ppi)

Release Year: 2012

Worldwide Market Share: 9.9%

Memory Usage Responsiveness

DEMO

Q&A



All the things

- Me - <http://redth.info> - <http://github.com/Redth> - @redth
- Xamarin - <http://xamarin.com> - @xamarinhq
- <http://xamarin.com/test-cloud> - <http://components.xamarin.com>
- iCircuit - <http://tinyurl.com/icircuit-code-sharing> - @praclarum
- MVVMCross - <http://github.com/slodge/mvvmcross> - @slodge
- Greg Shackles - Get his book! - @gshackles
- MonoGame - <http://www.monogame.net>
- Evolve Videos: <http://xamarin.com/Evolve/2013/>

THANK YOU