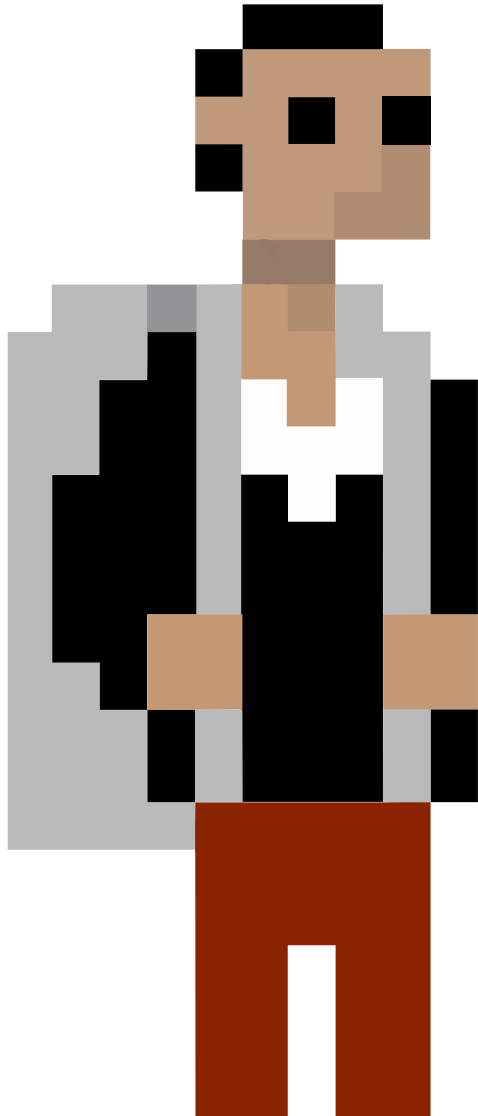# Using and Developing Plugins for Xamarin

James Montemagno
@JamesMontemagno
Developer Evangelist, Xamarin

Xamarin

Who's this guy?

James
Montemagno
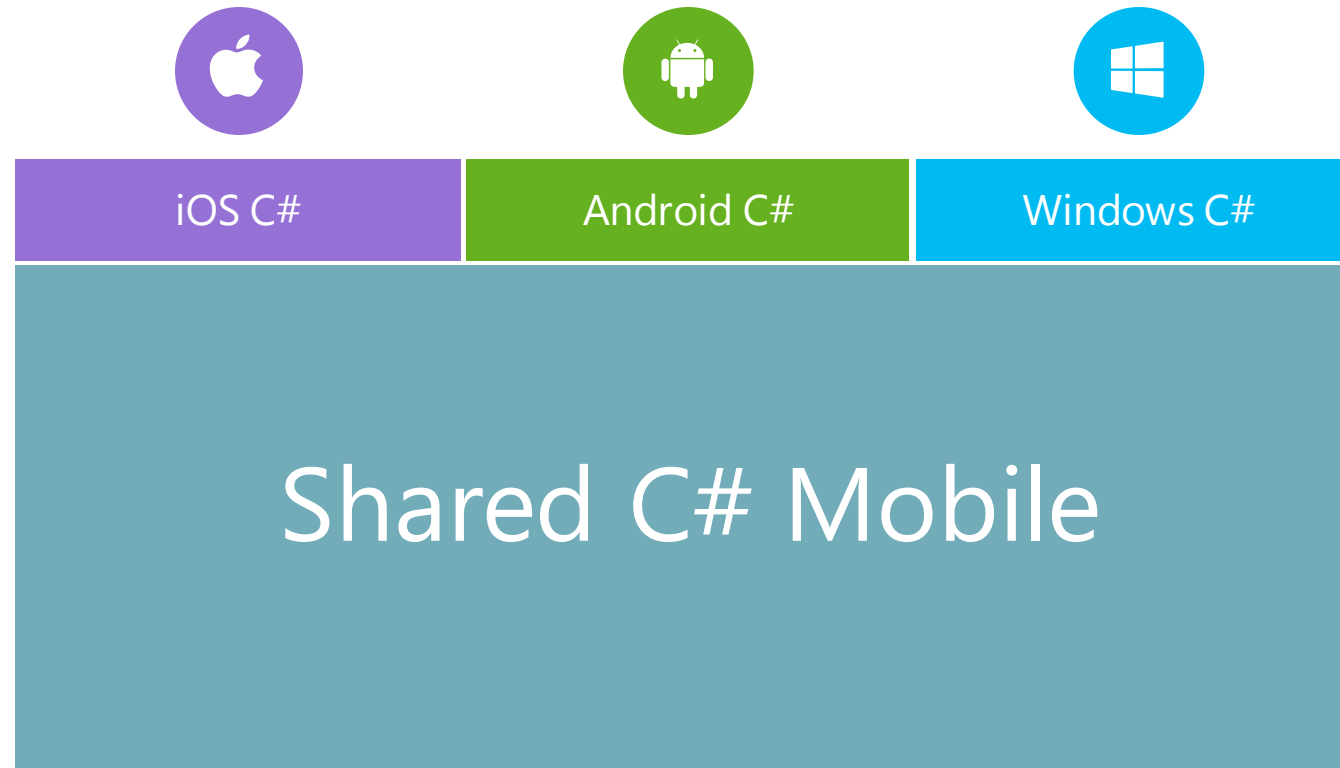Developer Evangelist, Xamarin

james@xamarin.com   **http://motzcod.es**   @JamesMontemagno

Xamarin's Unique Approach

iOS C#	Android C#	Windows C#

Shared C# Mobile

Shared C# codebase • 100% native API access • High performance

Windows APIs

Microsoft.Phone	Microsoft.Networking	Windows.Storage	Windows.Foundation	Microsoft.Devices
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

C#

iOS – 100% API Coverage

MapKit	UIKit	iBeacon	CoreGraphics	CoreMotion
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

C#

Android – 100% API Coverage

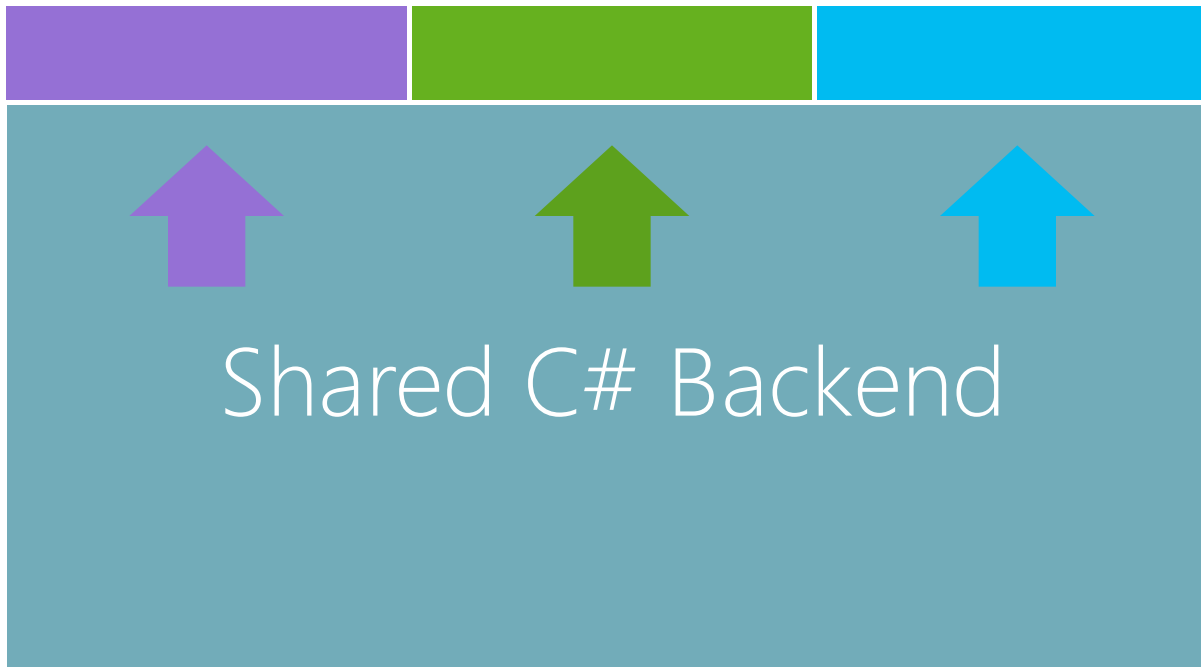| Text-to-speech	ActionBar	Printing Framework	Renderscript	NFC
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

C#

Shared C# Backend

Platform Specific Code

UI+APIs

Battery
GPS
Lights
Notifications
Settings
Text To Speech

UI + APIs

Battery
GPS
Lights
Notifications
Settings
Text To Speech

UI + APIs

Battery
GPS
Lights
Notifications
Settings
Text To Speech

Text To Speech

AVSpeechSynthesizer

```
public void Speak (string text)
{
    var speechSynthesizer = new AVSpeechSynthesizer ();
    var speechUtterance = new AVSpeechUtterance (text)
    {
        Rate = AVSpeechUtterance.MaximumSpeechRate/4,
        Voice = AVSpeechSynthesisVoice.FromLanguage ("en-US"),
        Volume = 0.5f,
        PitchMultiplier = 1.0f
    };
    speechSynthesizer.SpeakUtterance (speechUtterance);
}
```

Text To Speech

TextToSpeech

```csharp
TextToSpeech speaker;
string toSpeak;
public void Speak (string text)
{
  toSpeak = text;
  if (speaker == null)
  {
    speaker = new TextToSpeech (ctx, this);
  }
  else
  {
    var p = new Dictionary<string,string> ();
    speaker.Speak (toSpeak, QueueMode.Flush, p);
  }
}
public void OnInit (OperationResult status)
{
  //... more stuff
}
```

Text To Speech

SpeechSynthesizer

```
public async void Speak(string text)
{
    SpeechSynthesizer synth = new SpeechSynthesizer();
    await synth.SpeakTextAsync(text);
}
```

Common Interface?

ITextToSpeech

```csharp
public interface ITextToSpeech
{
    void Speak (string text);
}
```

What if we didn't have to write all this code?

Plugins for Xamarin

github.com/xamarin/plugins

Common API
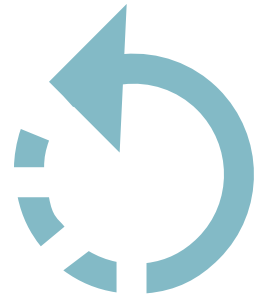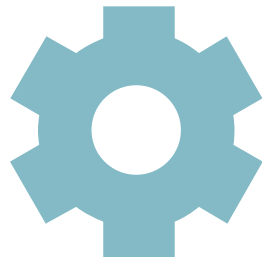
This is too good to be true!
Show me the Plugins!

How do they work?

Portable Class Libraries

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

Install additional frameworks...

OK   Cancel
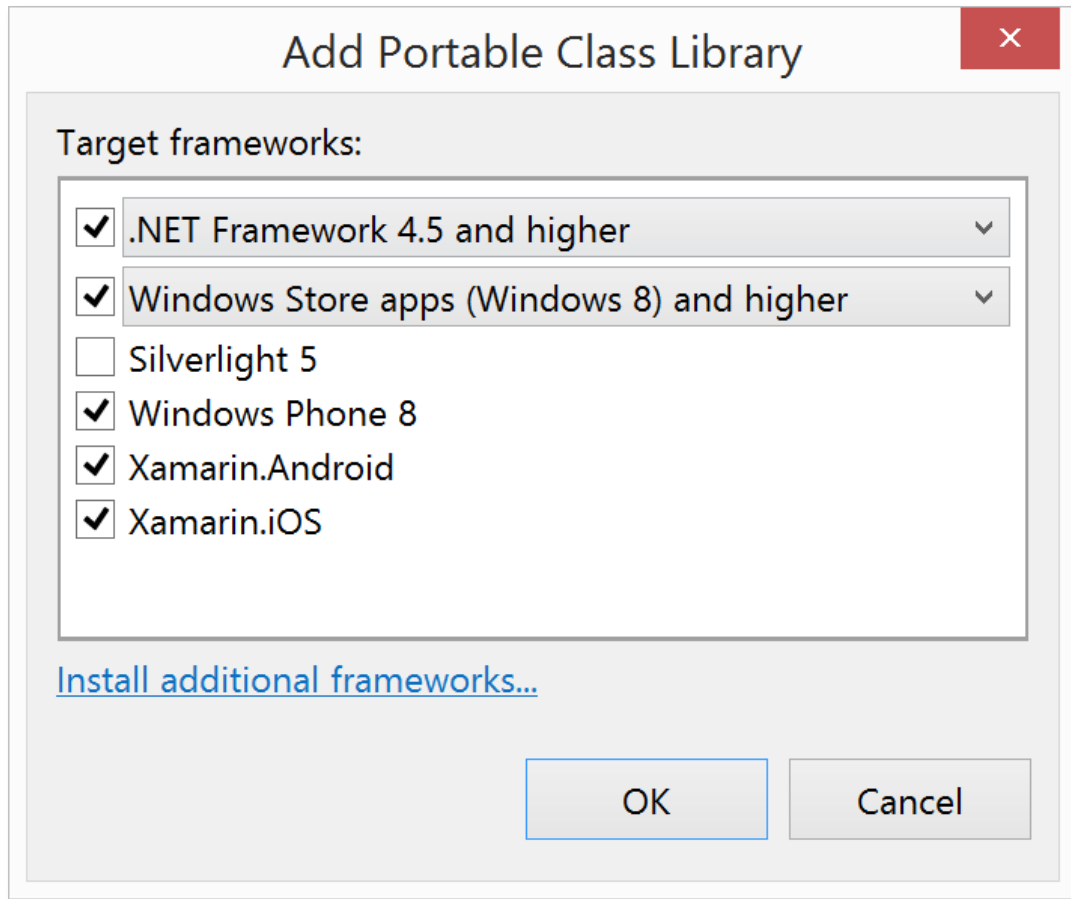
1 Assembly
Multiple Platforms

Including:
Xamarin.iOS and Xamarin.Android

What is a PCL?

Part 1:

- Reference Assembly
  - API Contract

Part 2:

- Implementation

Native iOS App

Views

Call Into View Model

Native Android App

Views

Call into View Model

Shared Code

Models

View Models

VM Calls Method

Reference

Reference

Portable Class Library

API Contract

Built in IoC
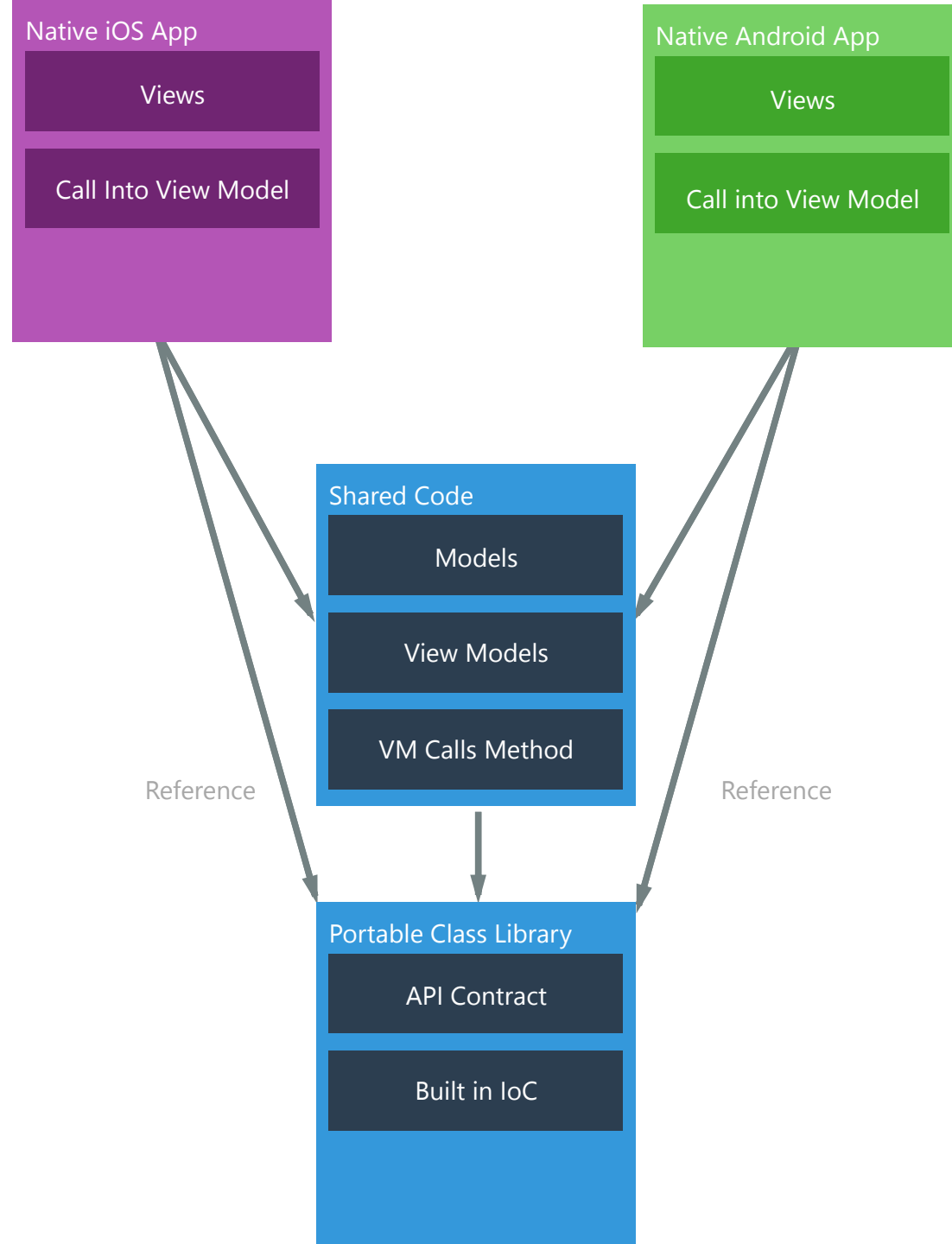
Standard PCL – Profile Only

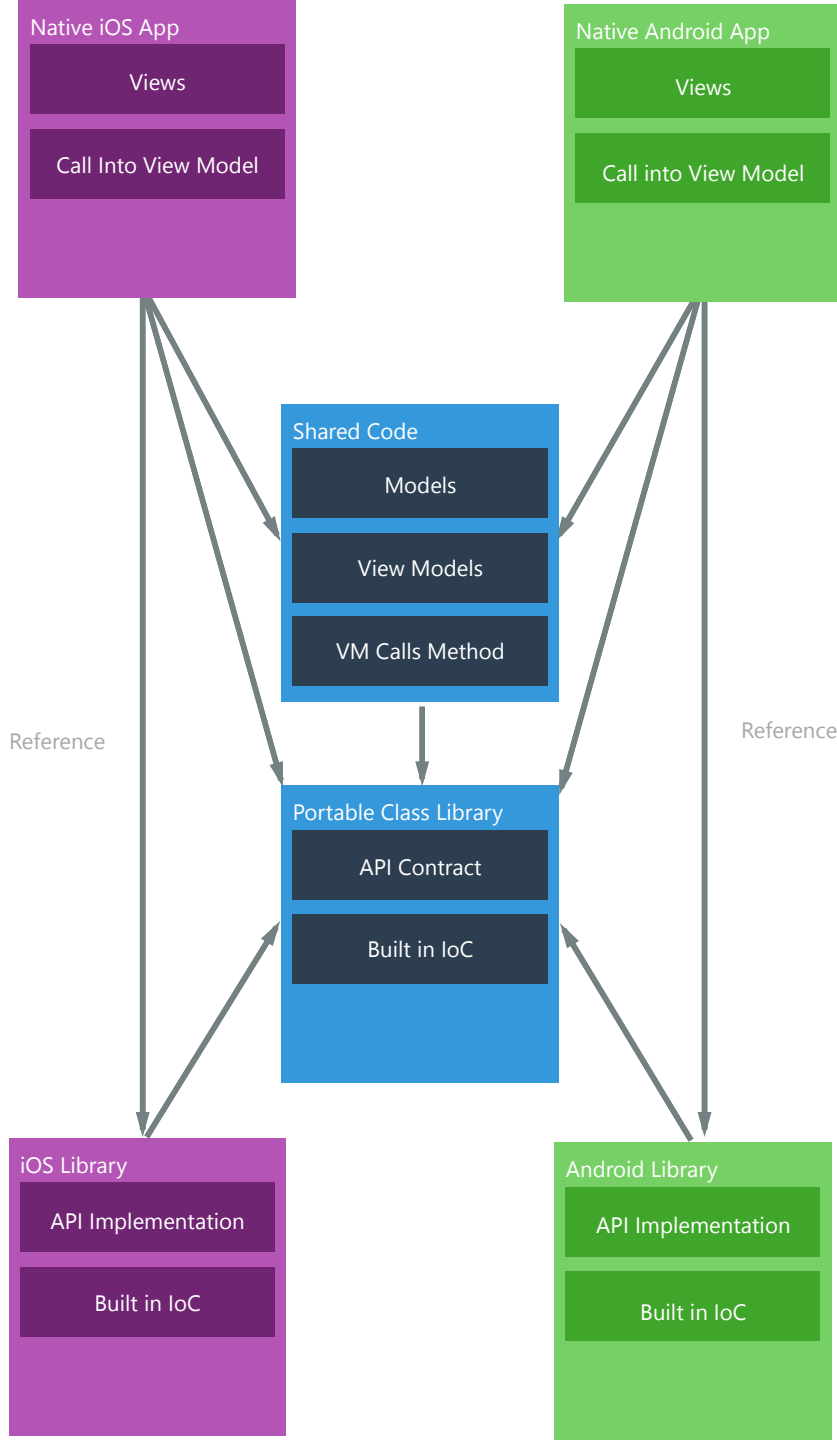- Limited to PCL Subset
- Easy To Manage
- Easy to Share

How to Tap into Platform Specific Code

Part 1:

- Reference Assembly

  - API Contract
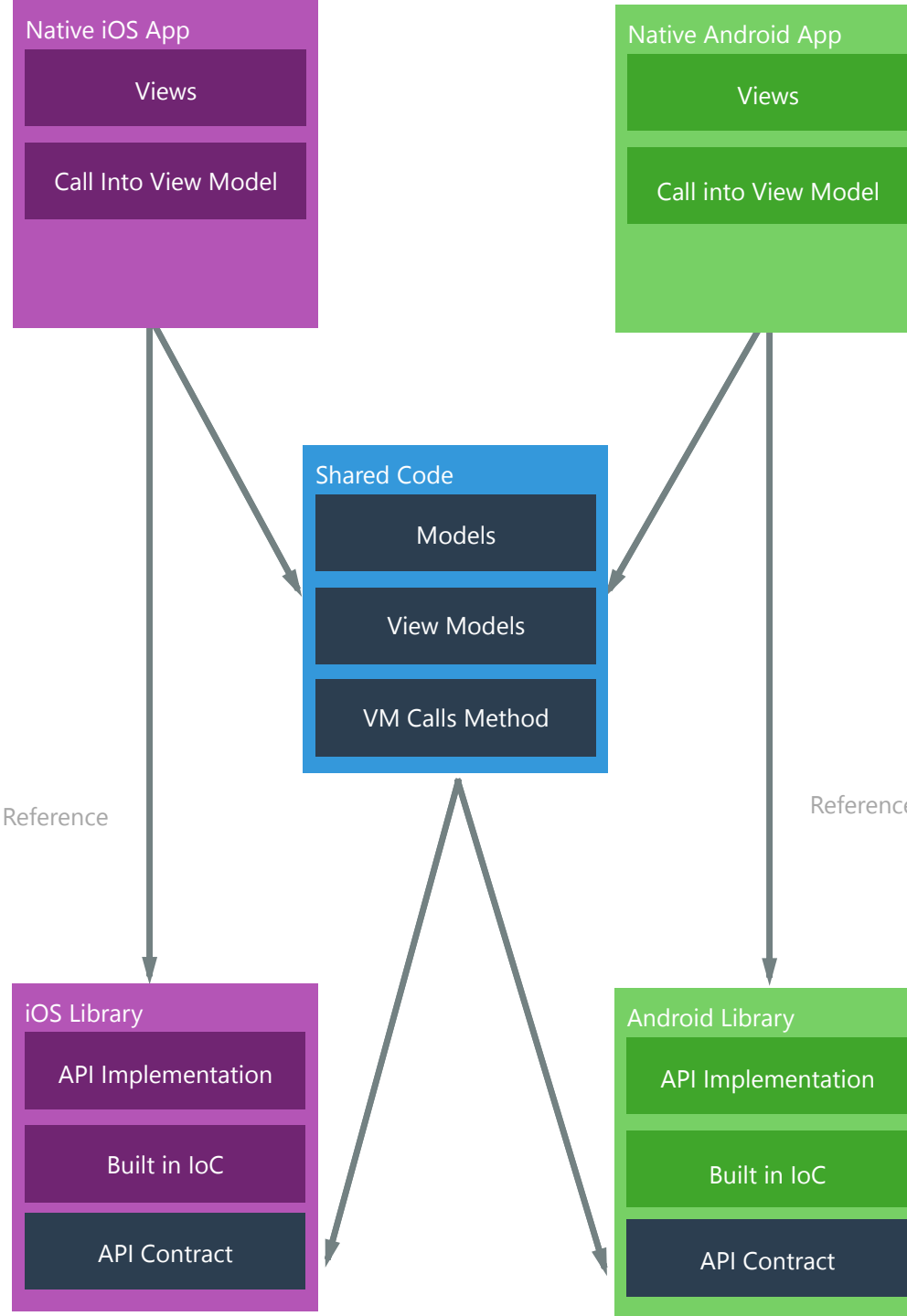

Part 2:

- Implementation

Leverage Bait & Switch

If PCL & Platform Binary Matches:

- Assembly Name
- Version
- Class Structure

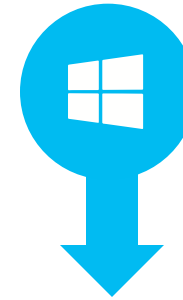Then PCL Binary is Replaced with Platform Binary

How do you create them?

Project Structure

Solution 'Connectivity' (8 projects)
- ▷ 📁 Solution Items
- ▲ 🔒C# **Connectivity.Plugin (Portable)**
  - ▷ 🔒🔧 Properties
  - ▷ References
  - ▷ 🔒C# CrossConnectivity.cs
- ▲ 🔒C# Connectivity.Plugin.Abstractions (Portable)
  - ▷ 🔒🔧 Properties
  - ▷ References
  - ▷ 🔒C# BaseConnectivity.cs
  - ▷ 🔒C# ConnectionType.cs
  - ▷ 🔒C# IConnectivity.cs
- ▲ 🔒📱 Connectivity.Plugin.Android
  - ▷ 🔒🔧 Properties
  - ▷ References
  - 📦 Components
  - ▷ 📁 Resources
  - ▷ 🔒C# ConnectivityChangeBroadcastReceiver.cs
  - ▷ 🔒C# ConnectivityImplementation.cs
  - ▷ 🔒C# CrossConnectivity.cs
- ▷ 🔒📱 Connectivity.Plugin.iOS
- ▷ 🔒📱 Connectivity.Plugin.iOSUnified
- ▷ 🔒C# Connectivity.Plugin.WindowsPhone8 (Windows Phone 8.0)
- ▷ 🔒C# Connectivity.Plugin.WindowsPhone81 (Windows Phone 8.1)
- ▷ 🔒C# Connectivity.Plugin.WindowsStore (Windows 8.1)
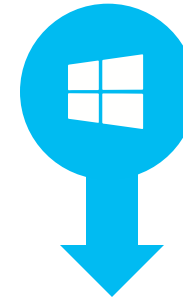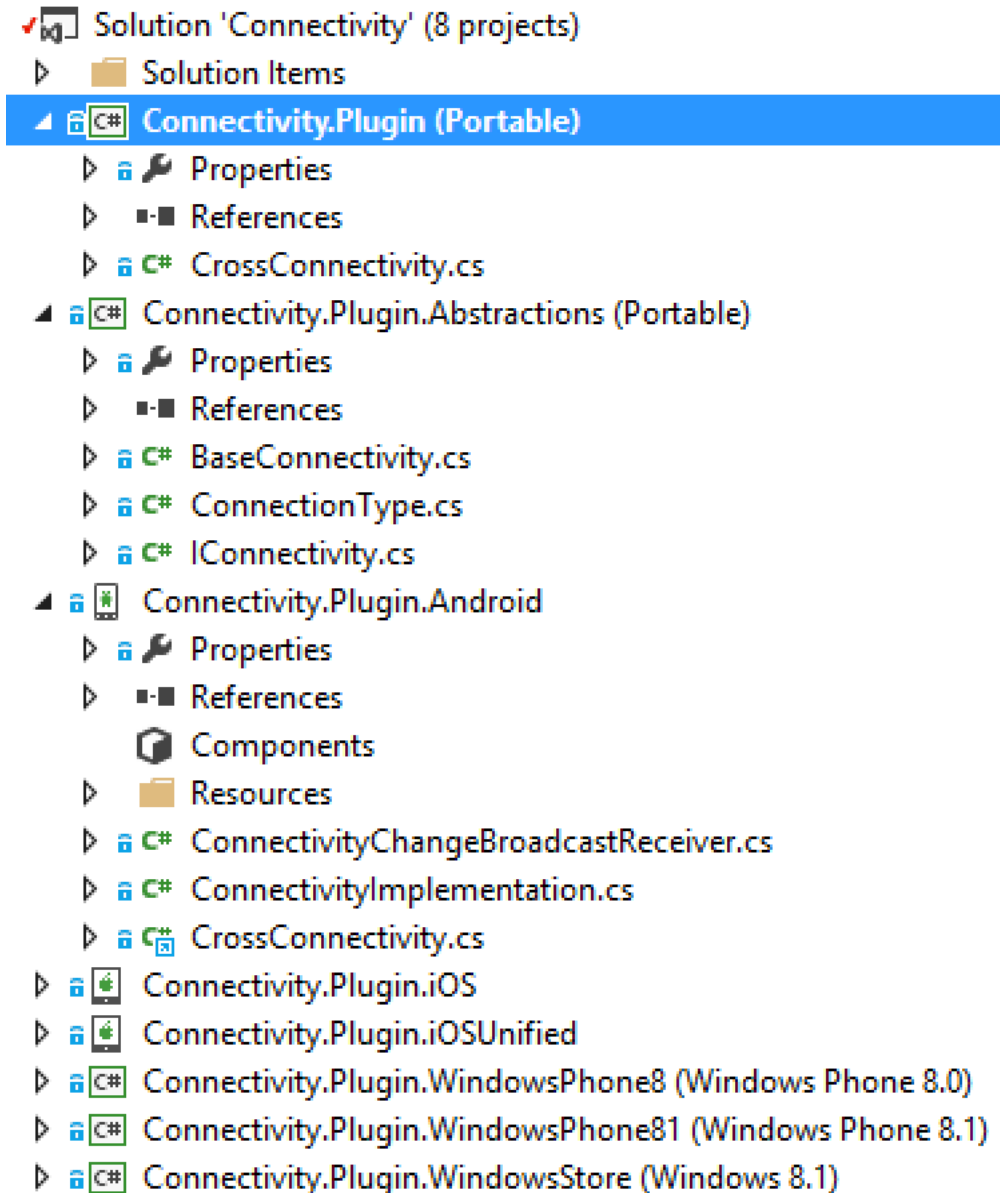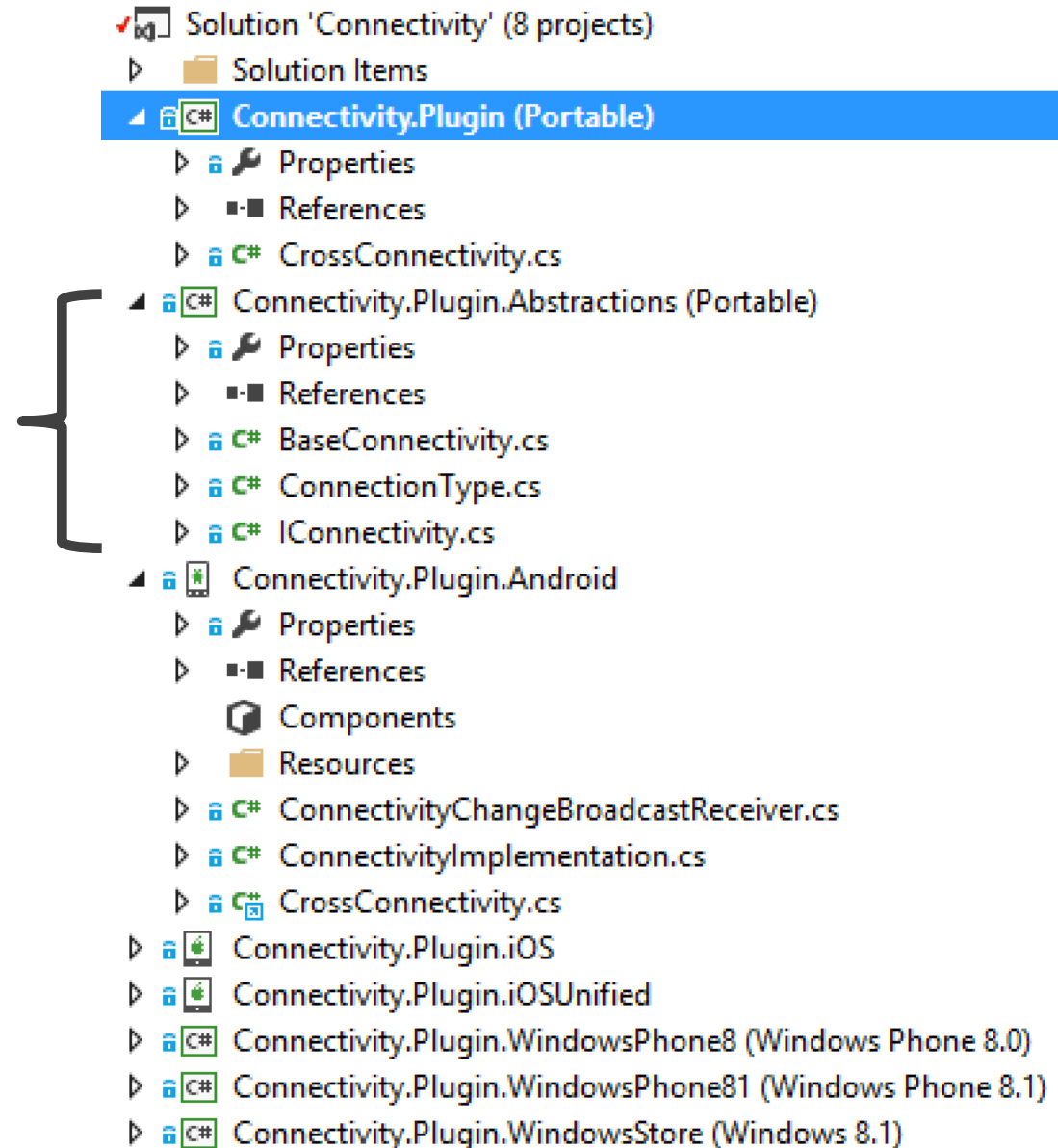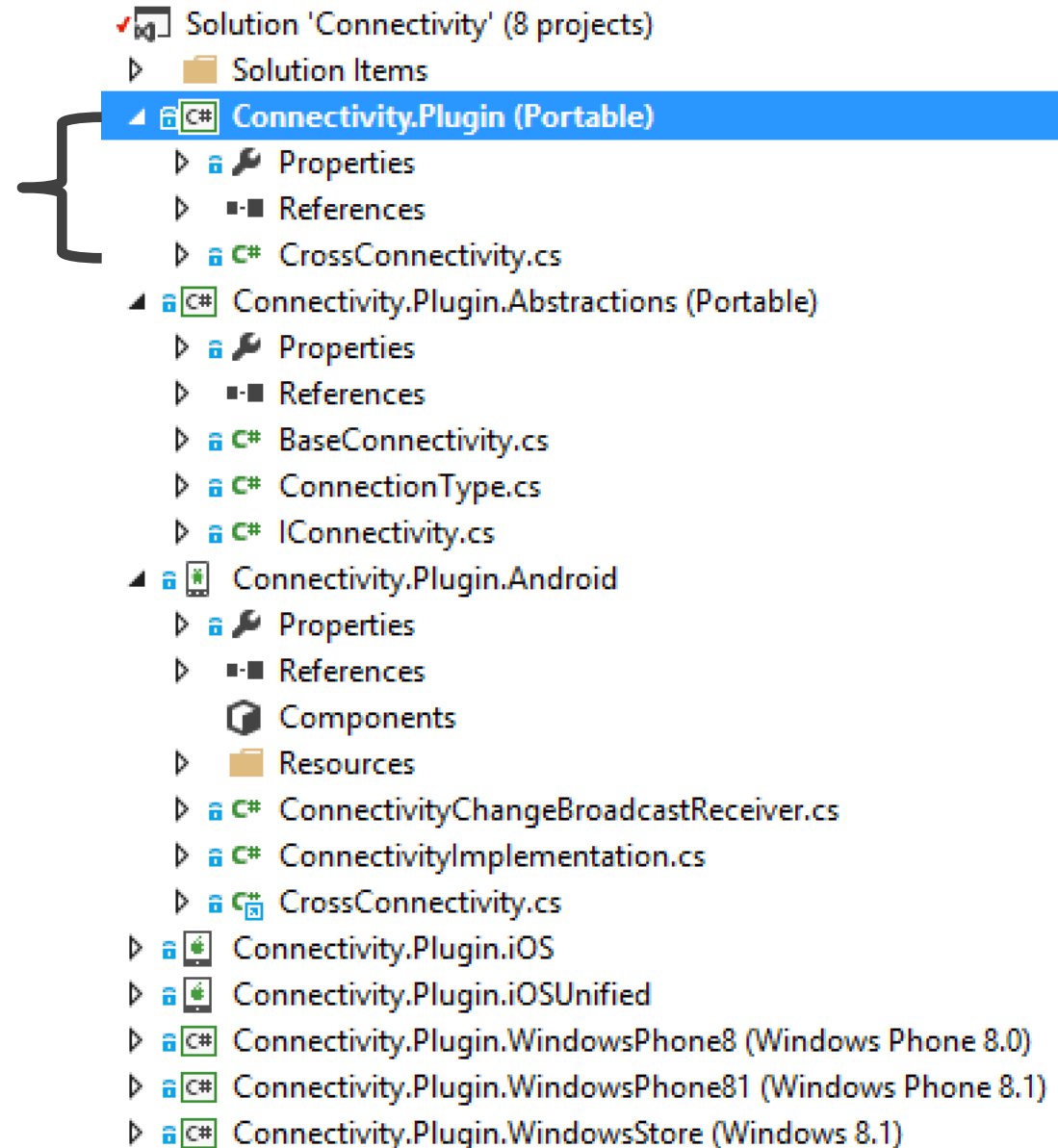
Plugin.Abstractions

- Interfaces
- Enums
- Installed in ALL projects

Plugin

- Internal IoC
- Bait/Switch

✓ Solution 'Connectivity' (8 projects)
  ▷ 📁 Solution Items
  ▲ 🔒C# **Connectivity.Plugin (Portable)**
    ▷ 🔒🔧 Properties
    ▷ ◼◻◼ References
    ▷ 🔒C# CrossConnectivity.cs
  ▲ 🔒C# Connectivity.Plugin.Abstractions (Portable)
    ▷ 🔒🔧 Properties
    ▷ ◼◻◼ References
    ▷ 🔒C# BaseConnectivity.cs
    ▷ 🔒C# ConnectionType.cs
    ▷ 🔒C# IConnectivity.cs
  ▲ 🔒📱 Connectivity.Plugin.Android
    ▷ 🔒🔧 Properties
    ▷ ◼◻◼ References
      📦 Components
    ▷ 📁 Resources
    ▷ 🔒C# ConnectivityChangeBroadcastReceiver.cs
    ▷ 🔒C# ConnectivityImplementation.cs
    ▷ 🔒C# CrossConnectivity.cs
  ▷ 🔒📱 Connectivity.Plugin.iOS
  ▷ 🔒📱 Connectivity.Plugin.iOSUnified
  ▷ 🔒C# Connectivity.Plugin.WindowsPhone8 (Windows Phone 8.0)
  ▷ 🔒C# Connectivity.Plugin.WindowsPhone81 (Windows Phone 8.1)
  ▷ 🔒C# Connectivity.Plugin.WindowsStore (Windows 8.1)

Plugin.Platform

- Implementation
- Bait/Switch IoC
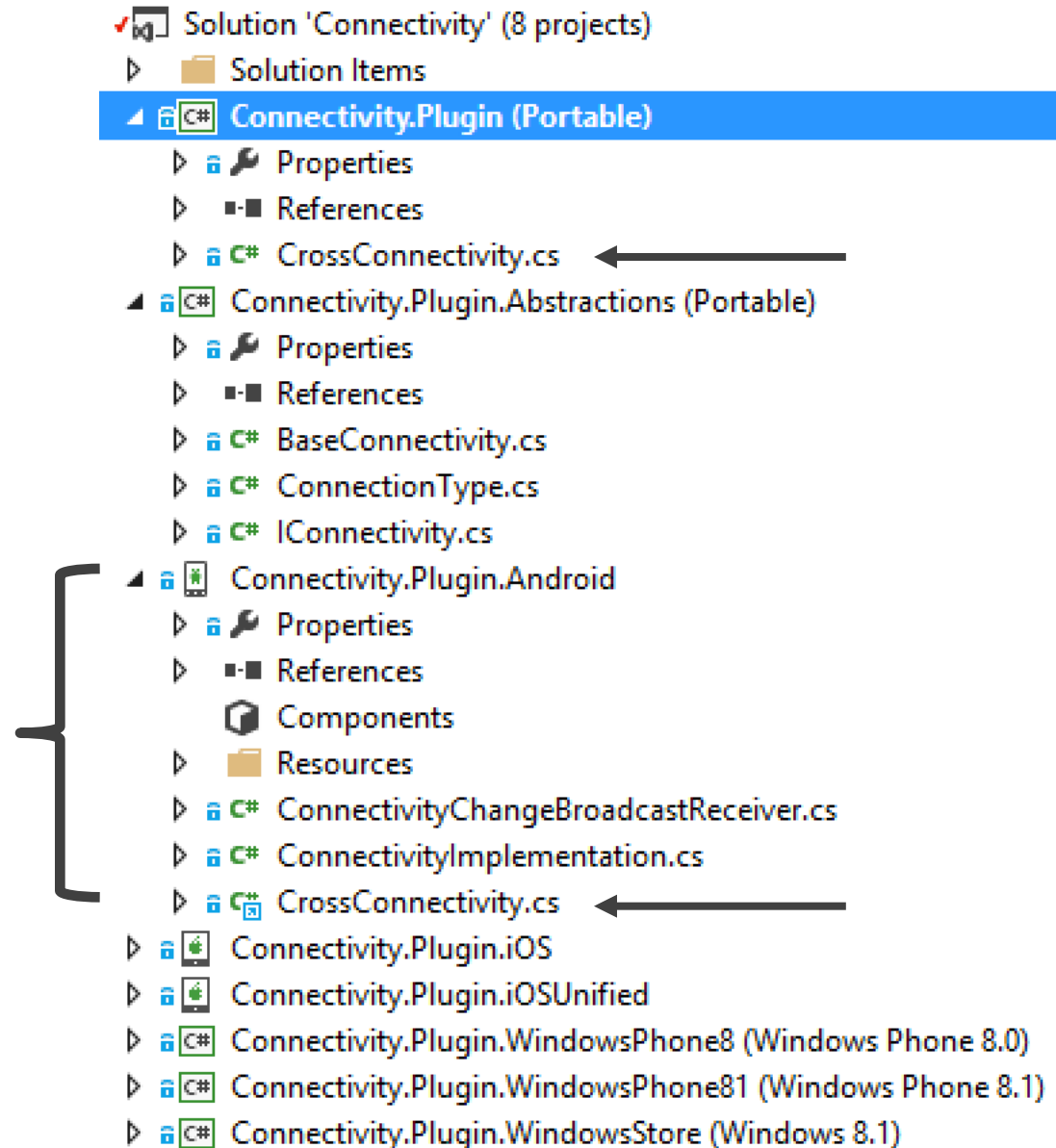
Let's take a look!

Resources

- Plugins for Xamarin
  - http://github.com/xamarin/plugins
- Plugin Templates
  - http://bit.ly/plugin-templates
- Documentation
  - http://developer.xamarin.com
- Bait and Switch
  - http://log.paulbetts.org/the-bait-and-switch-pcl-trick/
- Testing Plugins
  - http://www.michaelridland.com/xamarin/best-practices-using-xamarin-plugins/

Thank you. Questions?

James
Montemagno
Developer Evangelist, Xamarin

james@xamarin.com   http://motzcod.es   @JamesMontemagno