

# Mobile Development with Xamarin

Laurent Bugnion  
Senior Cloud Developer Advocate



What is Native?



?



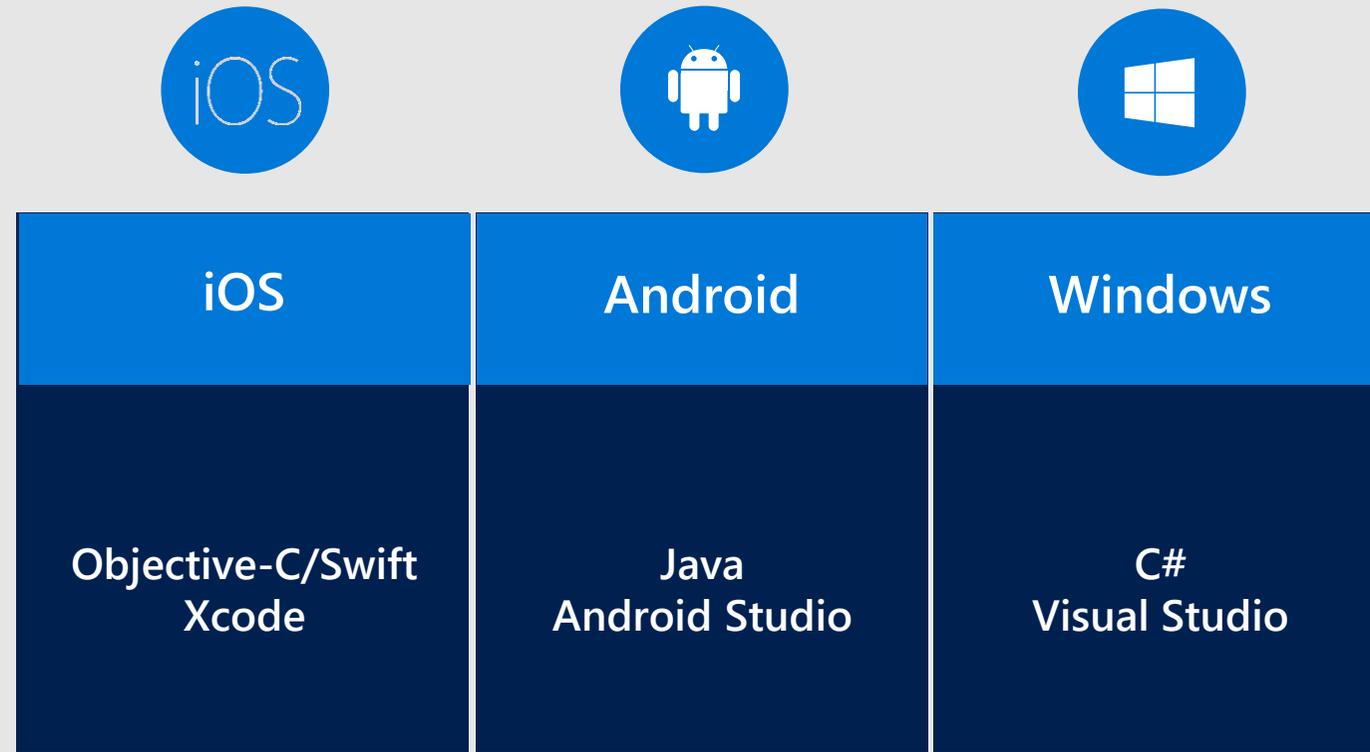
?



?

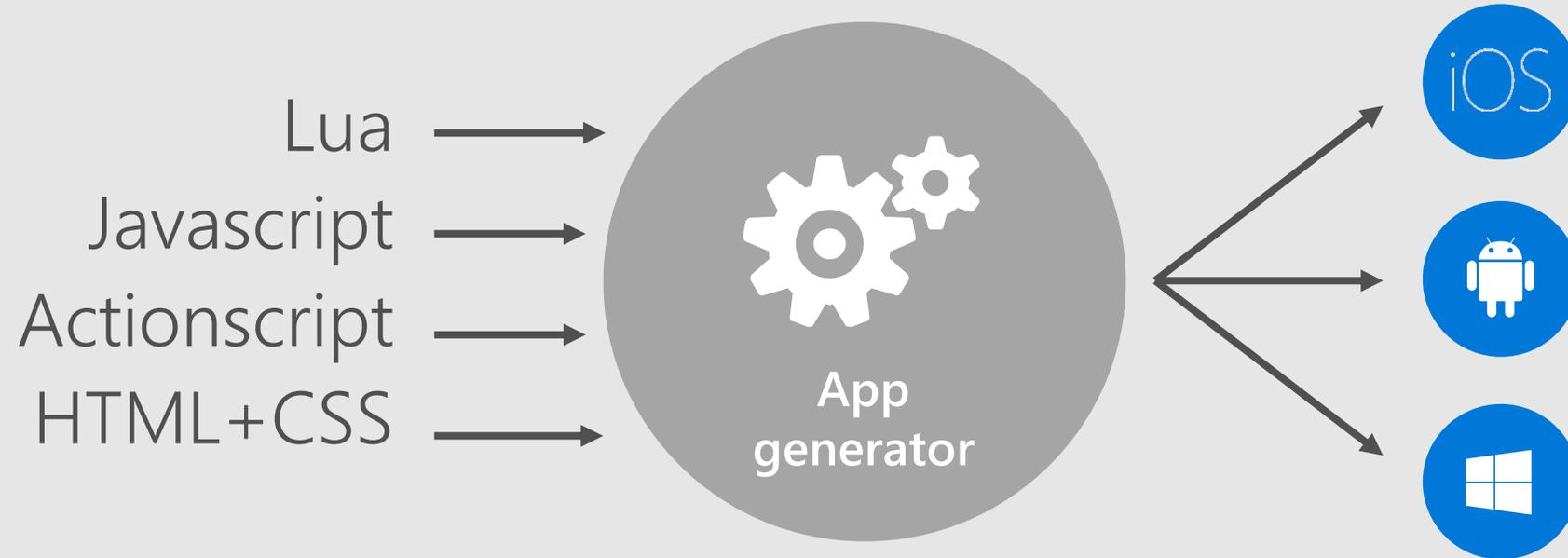
# Architecting Mobile Apps

# Silo approach



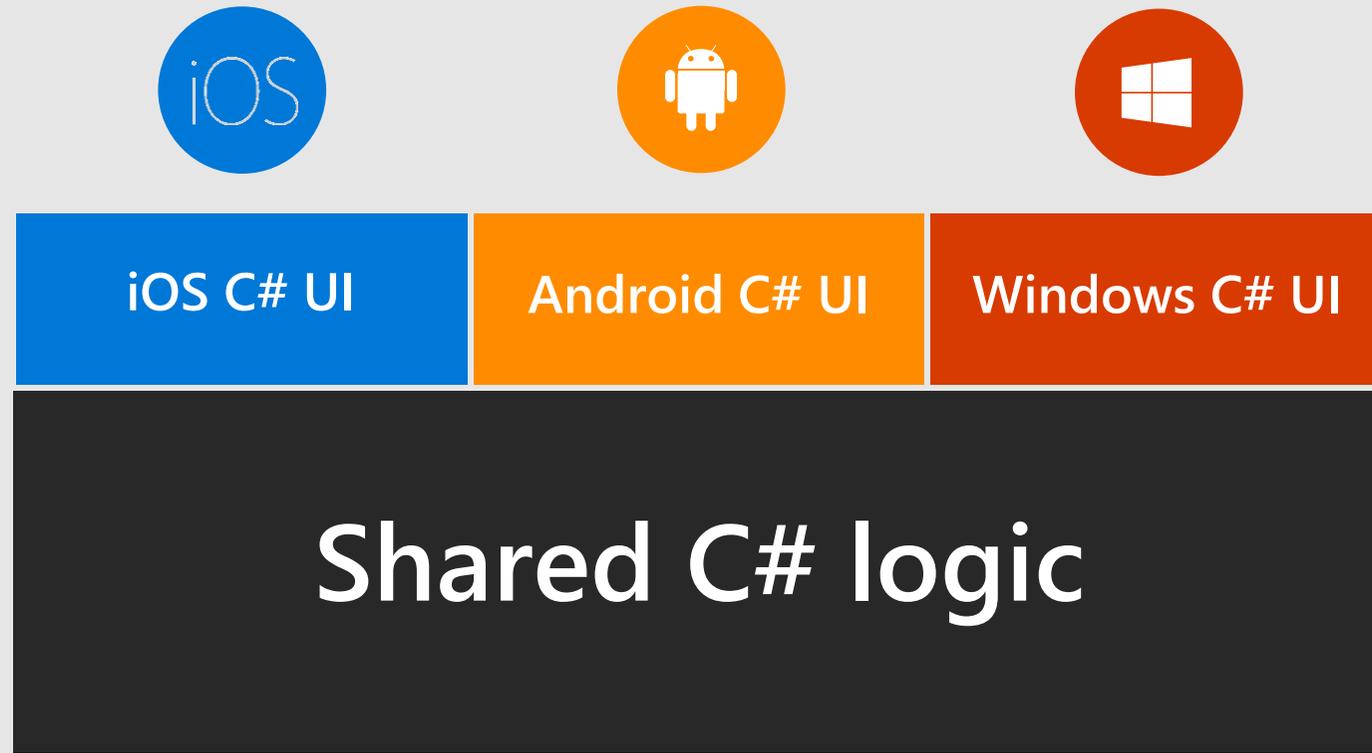
No shared code • Many languages and development environments • Multiple teams

# Write once, run anywhere



Limited native API access • Slow performance • Poor user experience

# Xamarin's unique approach



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

# How Xamarin Works

# 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#

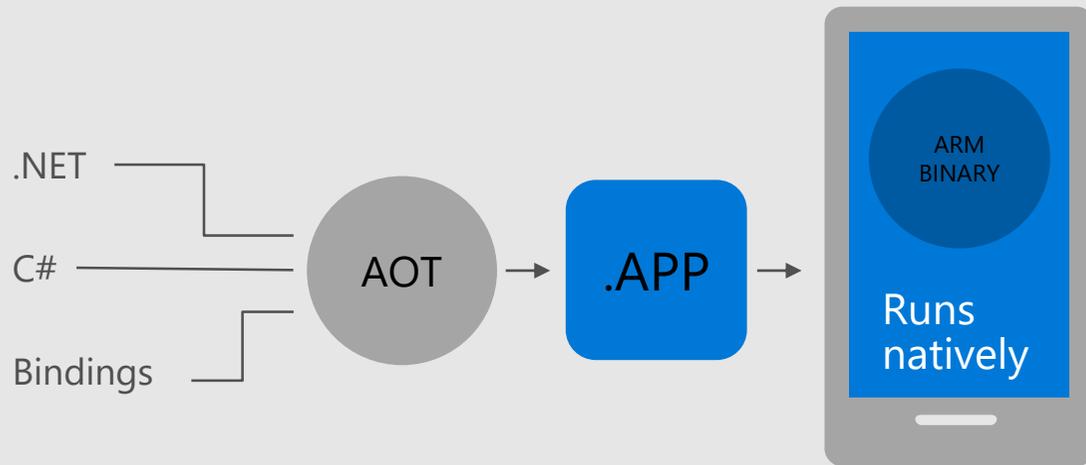
# Android—100% API coverage

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

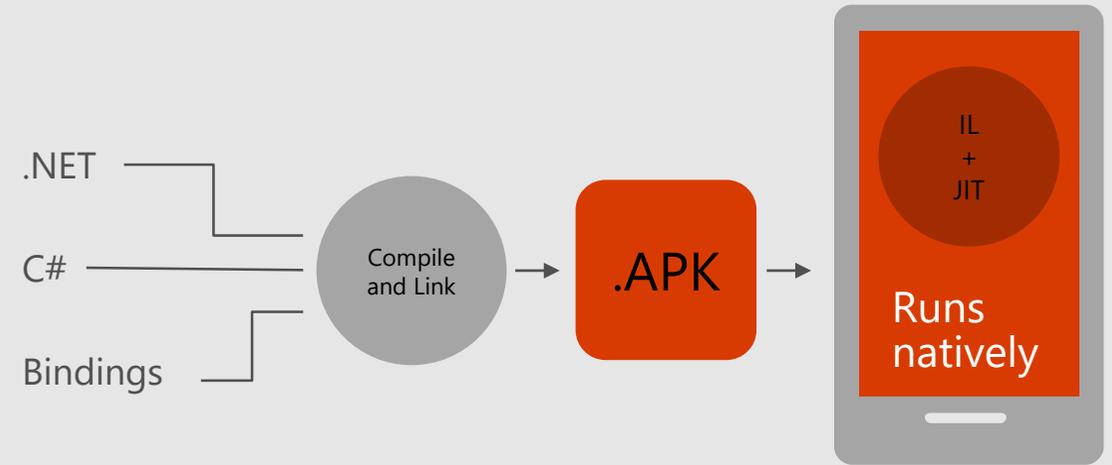


C#

# 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.

# ✓ Always up-to-date

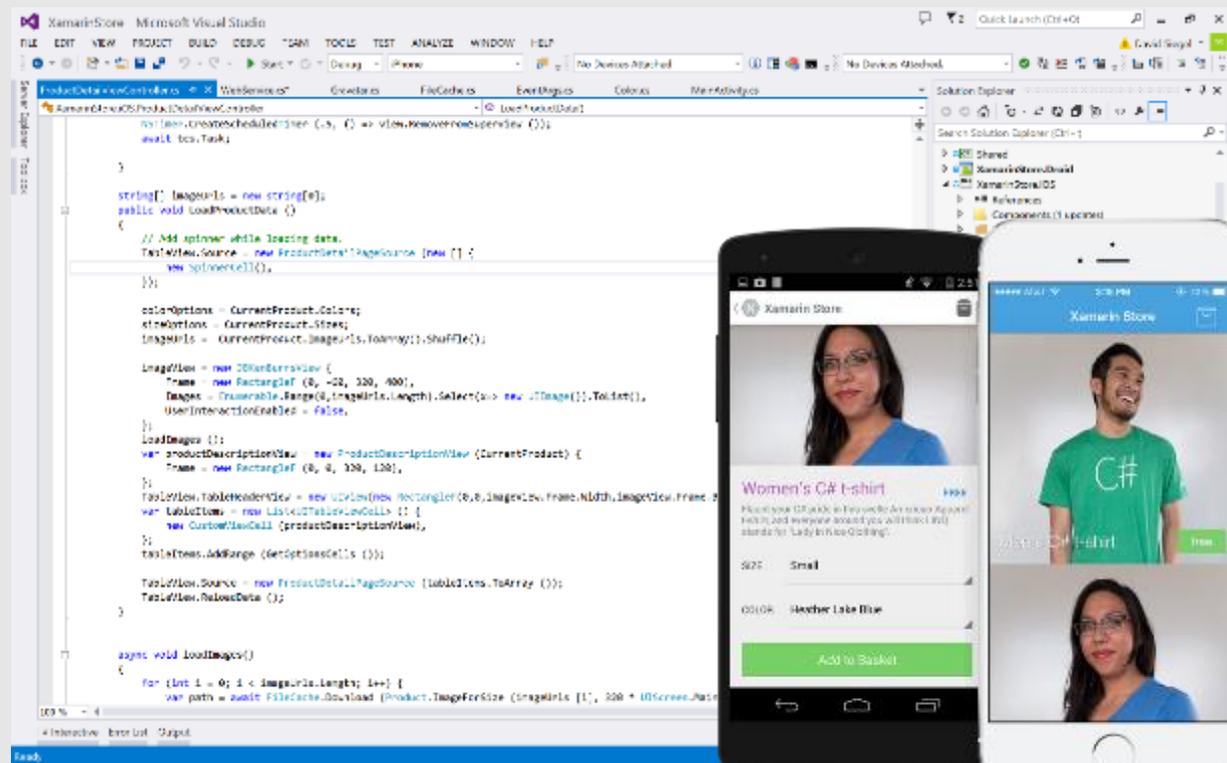
## **Same-day support:**

- iOS 5
- iOS 6
- iOS 7
- iOS 8
- iOS 9
- iOS 10
- iOS 11

## **Full support for:**

- Apple Watch
- Apple TV
- Android Wear
- Amazon Fire TV
- Google Glass
- and much more

Anything you can do in Objective-C, Swift, or Java can be done in C# with Xamarin



# Demo

Xamarin End-to-End

# Sharing User Interface

# Xamarin.Forms—Cross Platform UI

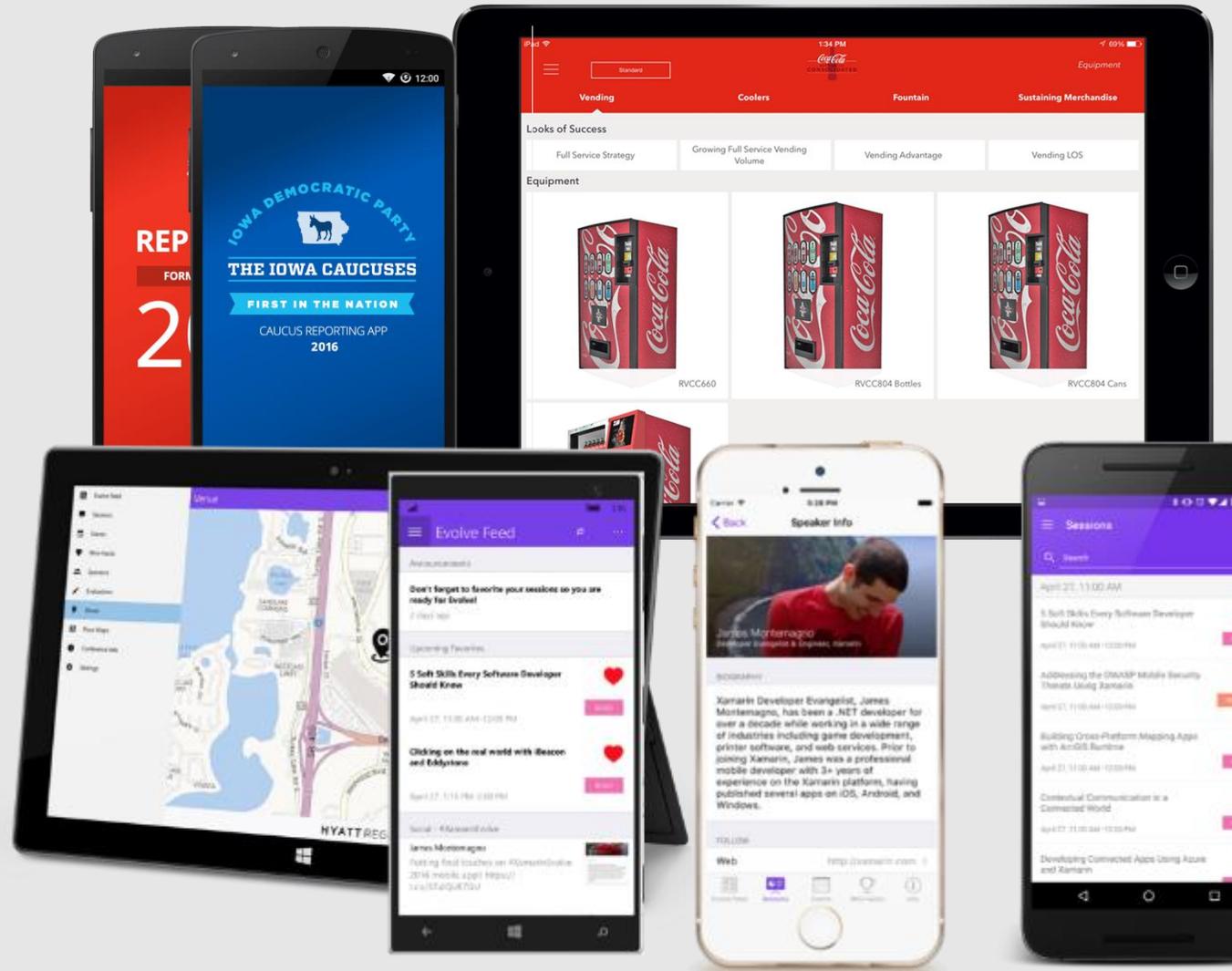


Shared UI Code

Shared C# Logic

Shared Native User Interface

Shared App Logic



# What's included

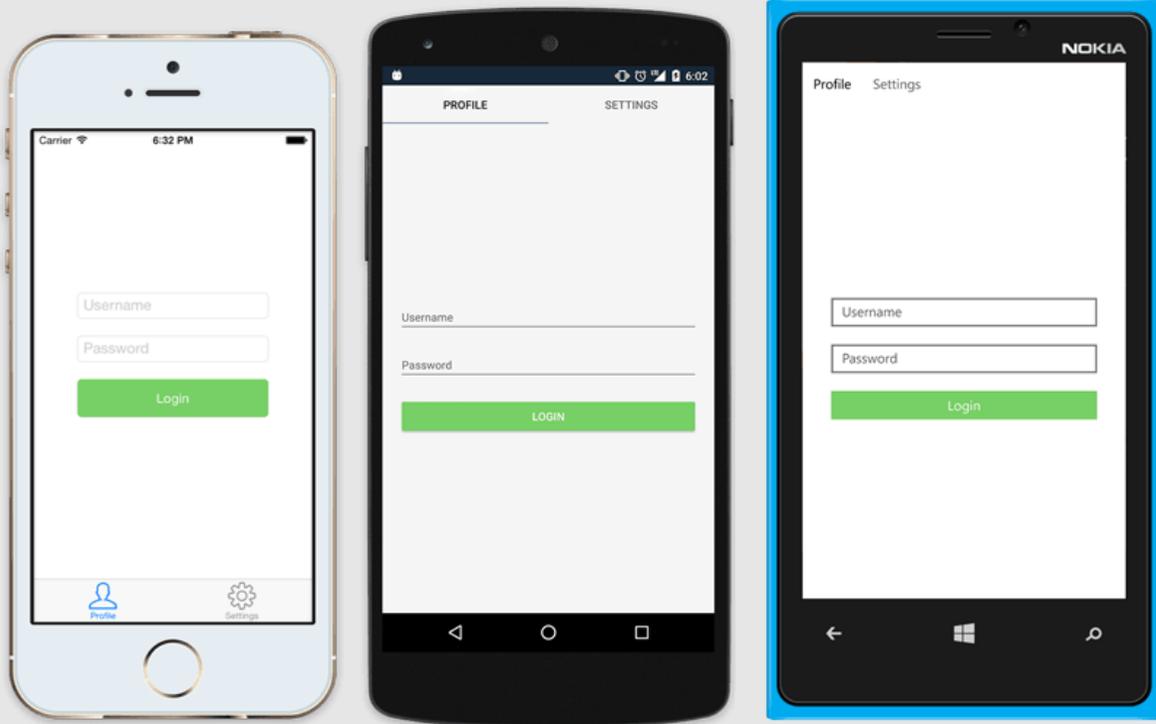


Shared UI Code

Shared C# Logic

- ✓ 40+ Pages, layouts, and controls (Build from code behind or XAML)
- ✓ Two-way data binding
- ✓ Navigation
- ✓ Animation API
- ✓ Dependency Service
- ✓ Messaging Center

# Native UI from shared code



```
<?xml version="1.0" encoding="UTF-8"?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="MyApp.MainPage">
  <TabbedPage.Children>
  <ContentPage Title="Profile" Icon="Profile.png">
    <StackLayout Spacing="20" Padding="20"
      VerticalOptions="Center">
      <Entry Placeholder="Username"
        Text="{Binding Username}"/>
      <Entry Placeholder="Password"
        Text="{Binding Password}"
        IsPassword="true"/>
      <Button Text="Login" TextColor="White"
        BackgroundColor="#77D065"
        Command="{Binding LoginCommand}"/>
    </StackLayout>
  </ContentPage>
  <ContentPage Title="Settings" Icon="Settings.png">
    <!-- Settings -->
  </ContentPage>
</TabbedPage.Children>
</TabbedPage>
```

# Demo

Xamarin.Forms  
Xamarin Live Player

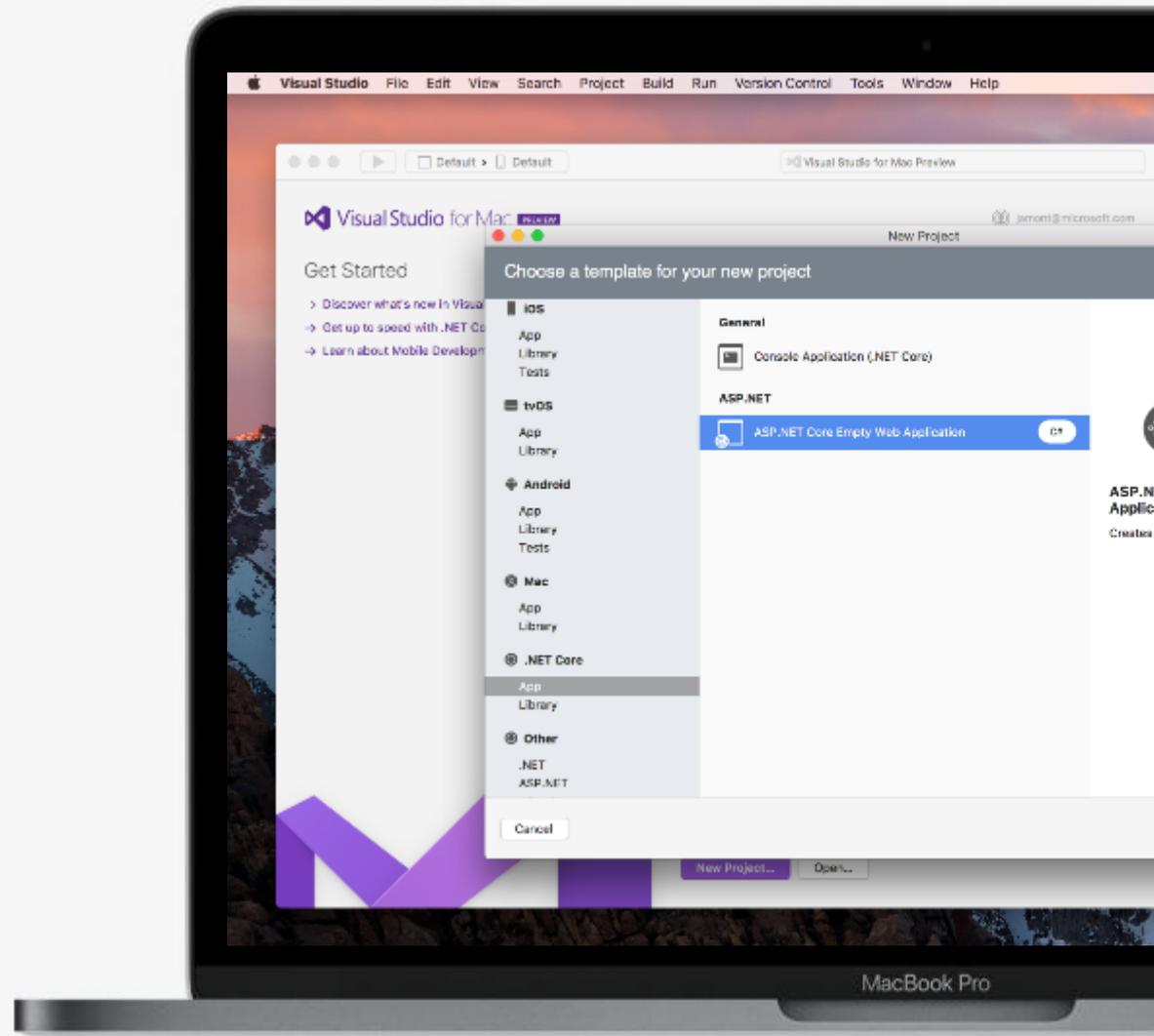
Xamarin Platform is  
included in Visual Studio

Including Community Edition

# Visual Studio



# Visual Studio for Mac



# Xamarin and Xamarin.Forms ecosystem





# Xamarin University

Free self-guided courses

Take free self-guided courses that cover the core Xamarin concepts and count towards certification. Register at:  
[university.xamarin.com/self-guided](https://university.xamarin.com/self-guided)

Get a full subscription for live classes, advanced topics, and to complete certification. Sign up at:  
[xamarin.com/university](https://xamarin.com/university)

[Samples](#) > [Functions](#) > [Xamarin](#) > Implementing a simple Azure Function with a Xamarin.Forms client

# Implementing a simple Azure Function with a Xamarin.Forms client



by [Laurent Bugnion](#)

Last updated: 1/27/2018

[Edit on GitHub](#)



[Browse on GitHub](#)



[Download as .zip](#)

<http://gslb.ch/c84s>

# Resources

All you need at  
<http://gslb.ch/baden18>

- Getting started with Xamarin  
<http://developer.xamarin.com>
- Xamarin University  
<http://university.xamarin.com>
- Xamarin Evolve conference  
<http://evolve.xamarin.com>
- MVVM Light and Xamarin  
<http://gslb.ch/evolve14>
- More MVVM Light and Xamarin  
<http://gslb.ch/evolve16>
- Xamarin Live Player  
<http://gslb.ch/c26s>
- Forms embedding in Native  
<http://gslb.ch/c56s>
- Native embedding in Forms  
<http://gslb.ch/c55s>
- Azure App Center  
<https://appcenter.ms>

# Please Complete your Session Evaluations

## Get your cool IoT Dev Kit!

Fill out your feedback form and turn it in before you leave.

