Introduction to Xamarin.Forms Cross Platform Mobile Application Development

By Brandon Ward
Department of Electrical and Computer Engineering – University of Utah

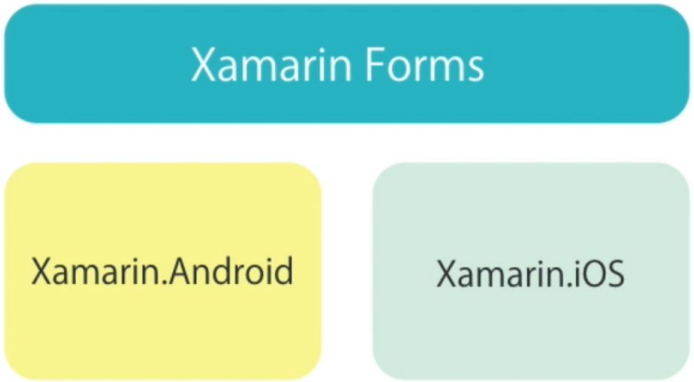
Salt Lake City, UT, USA

What is Xamarin.Forms?

- Xamarin.Forms is an extension of the Xamarin mobile development framework
- It's an open source framework that allows the building of cross-platform mobile apps for iOS, Android, and Windows with .NET from a single shared codebase.

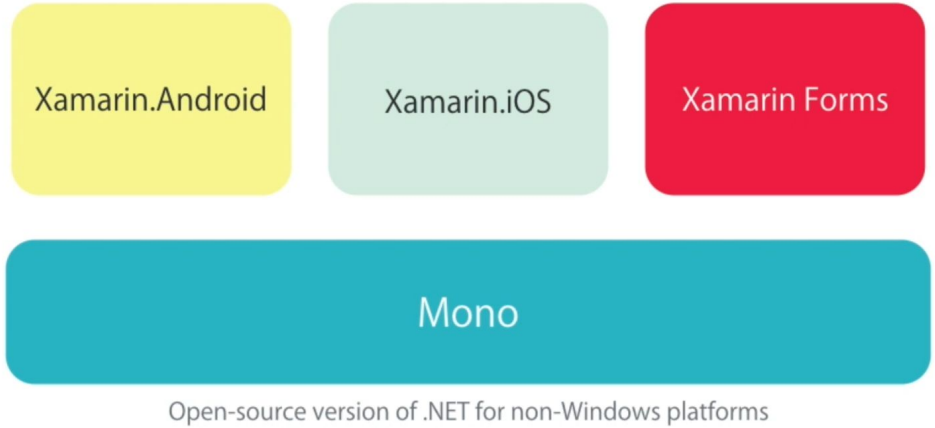
What is Xamarin.Forms? (cont.)

- Xamarin.Forms is built on two libraries, which allows for working with a single api for creating user interfaces.
- This also allows for the ability to create platform-specific code.



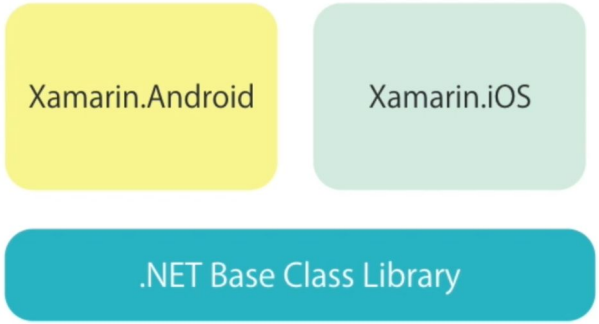
How Xamarin Works

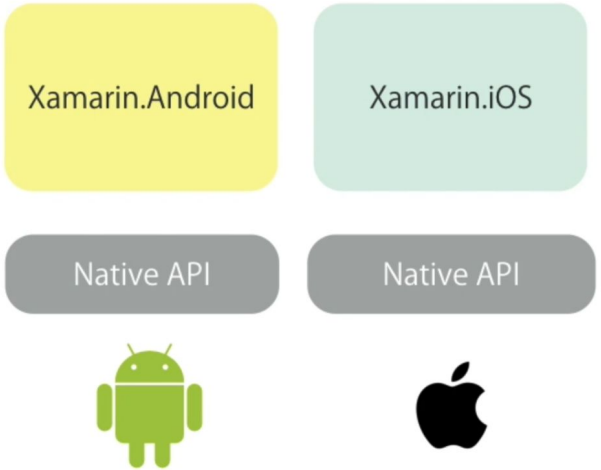
- Xamarin is built on top of Mono.
- Mono is open source version of .NET that runs on non-Windows platforms like Unix and Linux.



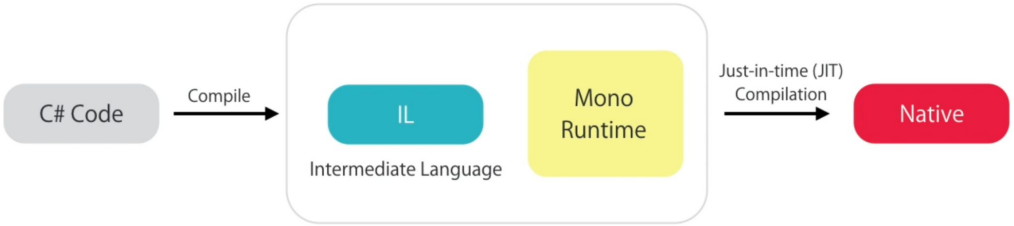
How Xamarin Works (cont.)

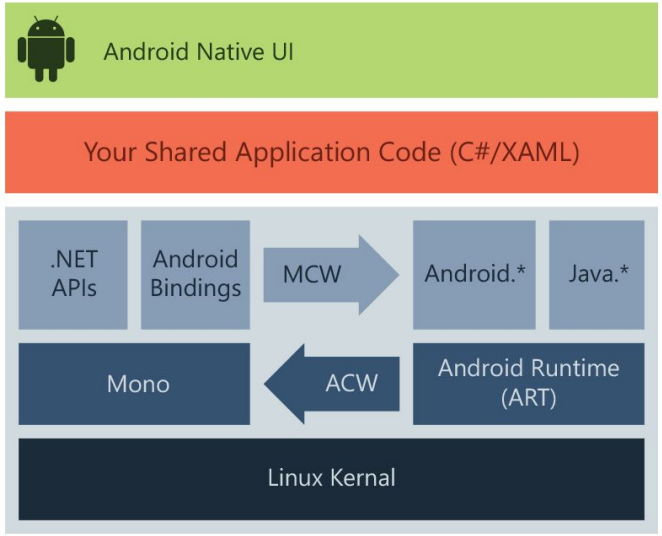
- Two releases of Mono
  - Mono for Android -> Now Xamarin.Android
  - MonoTouch -> Now Xamarin.iOS





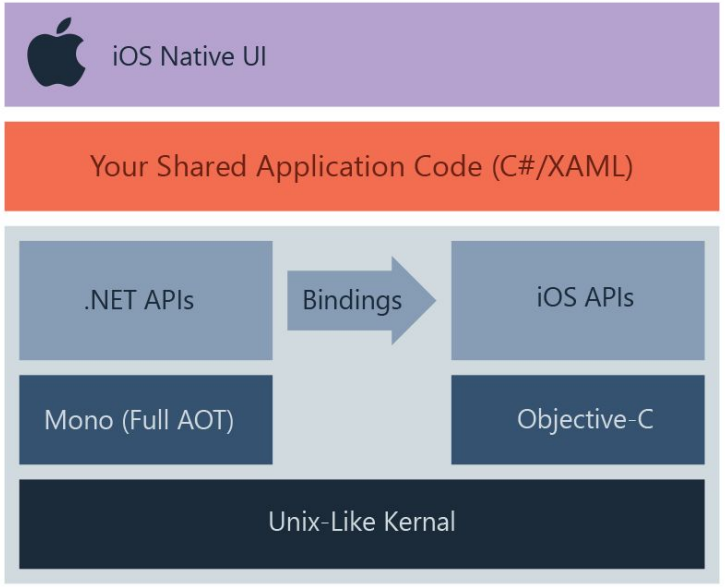
Xamarin.Android



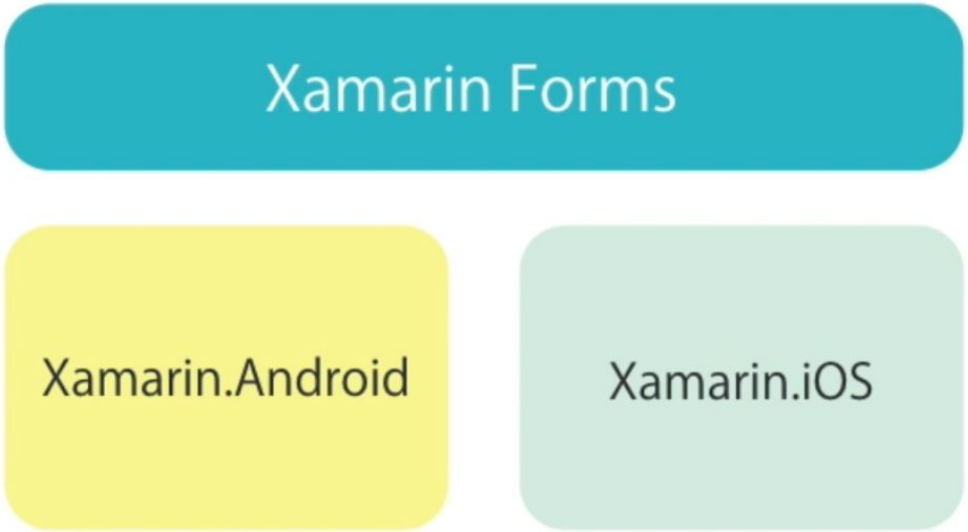


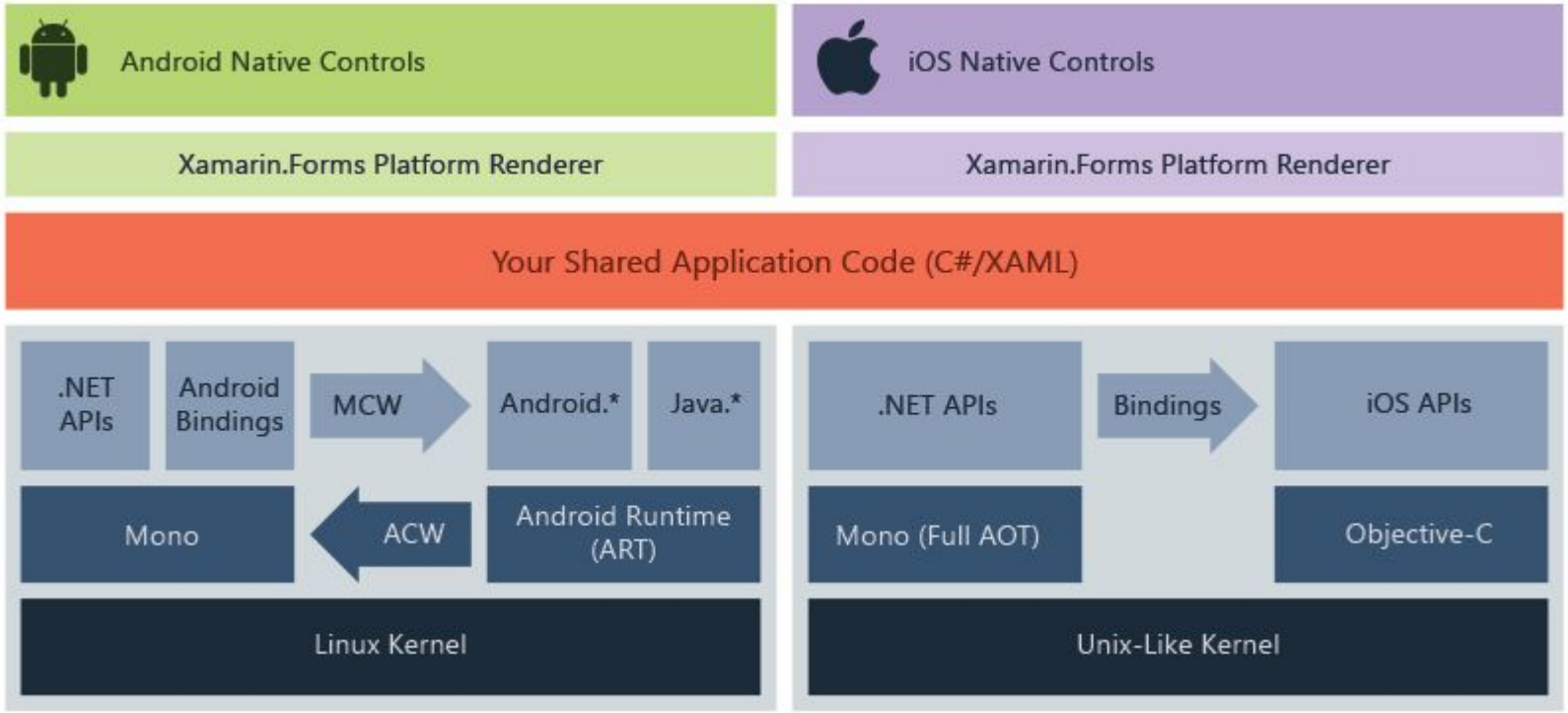
Xamarin.iOS

C# Code
Compile
IL
Ahead-of-time (AOT)
Compilation
Native


iOS Native UI
Your Shared Application Code (C#/XAML)
.NET APIs
Bindings
iOS APIs
Mono (Full AOT)
Objective-C
Unix-Like Kernal

Xamarin.Forms




Android Native Controls
iOS Native Controls
Xamarin.Forms Platform Renderer
Xamarin.Forms Platform Renderer
Your Shared Application Code (C#/XAML)
.NET APIs
Android Bindings
MCW
Android.*
Java.*
Mono
ACW
Android Runtime (ART)
Linux Kernel
.NET APIs
Bindings
iOS APIs
Mono (Full AOT)
Objective-C
Unix-Like Kernel

User Interfaces

The 4 main control groups used for Xamarin.Forms user interfaces:

- Pages
- Layouts
- Views
- Cells

See all of them:

https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/controls/

Data Binding

- Xamarin.Forms recommends using the MVVM design architecture.
- Despite each XAML file having a code-behind, you need a view model
- You can’t unit test if you don’t implement a view model

INotifyPropertyChanged

- To inform, items (labels, list, views, etc.) that the property values of the data it is bound to has changed, use INotifyPropertyChanged.
- Implement it in every view model in your app.

ListView and ObservableCollection

- For displaying a list of data, Xamarin.Forms uses ListViews
- ListViews are populated using ObservableCollection
- UI will automatically reflect the changes if the collection items are added, removed, and changed.