


Xamarin.iOS and Xamarin.Android

Adam Kemp
Senior Software Engineer
National Instruments



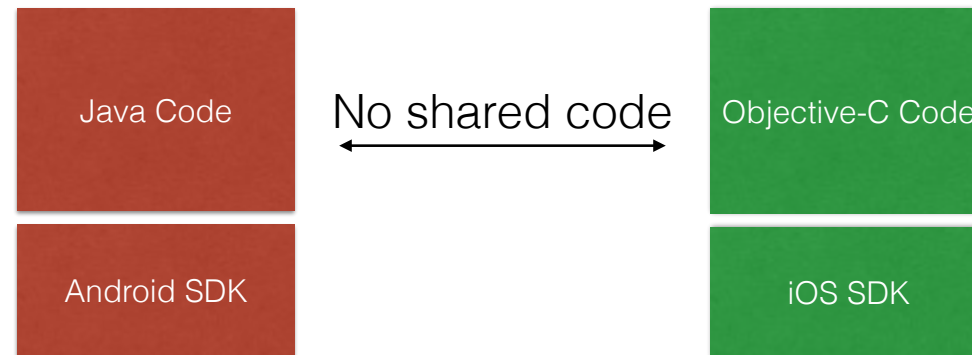
 @TheRealAdamKemp
blog.adamkemp.com

 github.com/TheRealAdamKemp

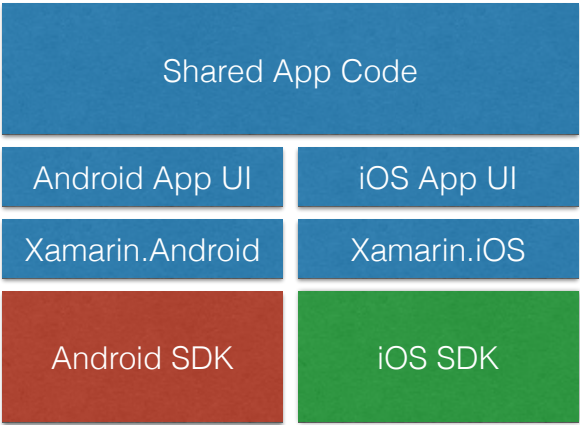
What is Xamarin?

- Xamarin provides frameworks and tools for building iOS, Android, and Mac applications using C#
 - Xamarin.iOS (née MonoTouch)
 - Xamarin.Android (née MonoDroid)
 - Xamarin.Mac (née MonoMac)
- **Xamarin apps are native apps**

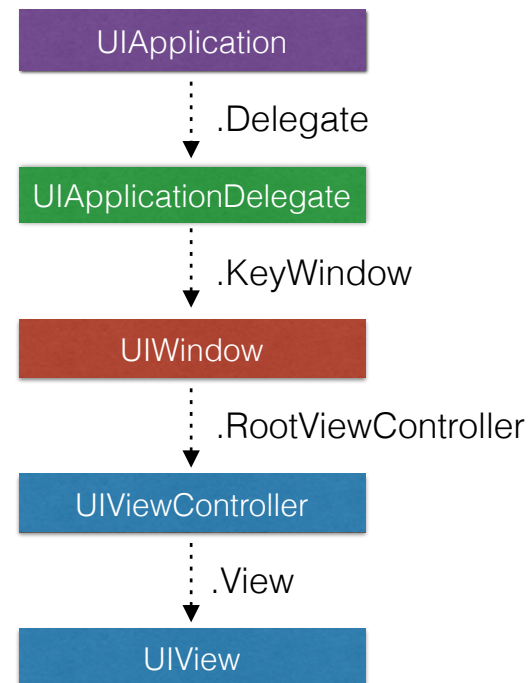
Old and Busted



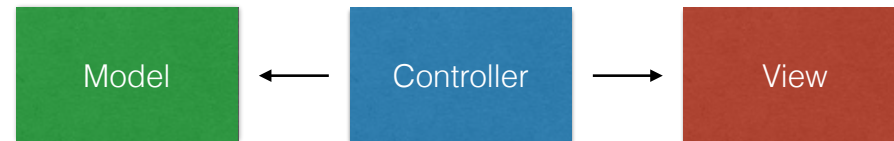
New Hotness



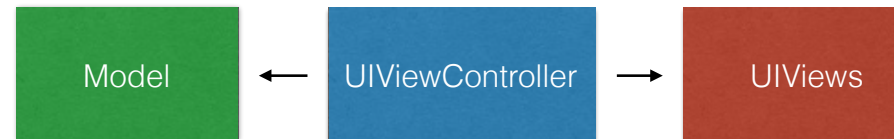
iOS App Components



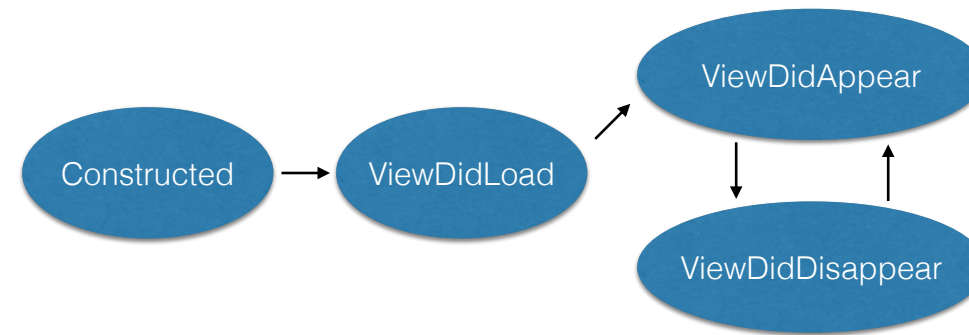
MVC



MVC (iOS)



UIViewController Lifecycle



Storyboards

- Don't use them
- Seriously

.xib

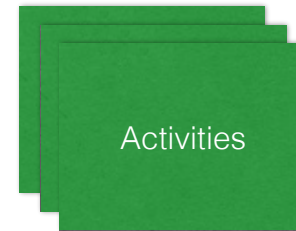
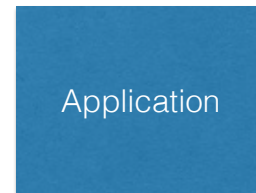
- Similar to XAML in purpose
 - Inflate a bunch of objects (mostly views)
 - Connect them together
- Can be associated with a view controller
- Edit with Xcode's Interface Builder
 - (Mostly) not human-editable

.xib

- Outlets
 - Connect an object in the .xib to a property of another object in the .xib or the “owner”
- Actions
 - Connect an event to a method on another object in the .xib or the “owner”

iOS Demo

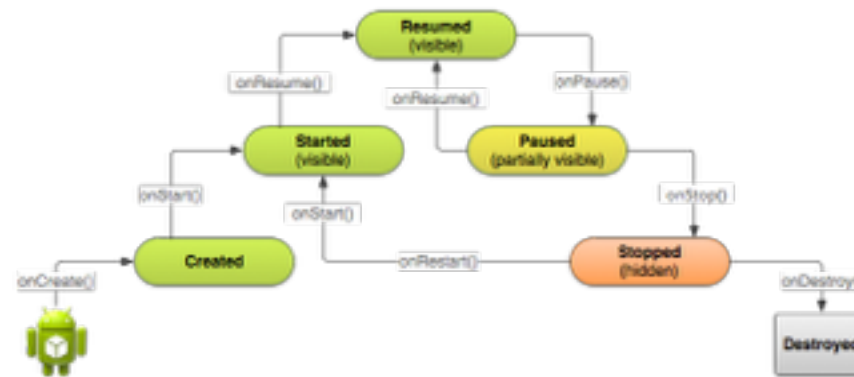
Android App Components



Activities

- Activities are designed to stand alone
- Transition from one to another using Intent objects
- Sending data from one activity to another is a pain
 - Encode in the Intent object (strings and URIs)
 - Can also use ContentProvider (which is a pain)

Activity Lifecycle



<http://developer.android.com/training/basics/activity-lifecycle/starting.html>

Image is CC Attribution (<http://creativecommons.org/licenses/by/2.5/>), found here: <http://developer.android.com/training/basics/activity-lifecycle/starting.html>

.axml

- Also similar to XAML in purpose
- Must be inflated in code
- Can be edited in Xamarin Studio using designer or XML source

.axml

- Views are found by id
 - Resource.designer.cs auto-generated
 - Resource.Layout.myView
 - Syntax: `android:id="@+id/myView"`

Android Demo

Dealing With Rotation

- Rotation is a configuration change
- Configuration changes cause activities to be destroyed and recreated
- All state in an activity must be saved and restored somehow
- **Yes, this is crazy**

Dealing With Rotation

- Approach 1: Disable this behavior
- **Google says don't do this**

Dealing With Rotation

- Approach 2: Save/restore state in Bundle
- onSaveInstanceState/OnRestoreInstanceState
- This is a pain

Dealing With Rotation

- Approach 3: Use a retained Fragment
- Move all logic and state into the Fragment
- Set RetainInstance = true
- **This is Google's recommended approach**

Fragment Demo