

Cross-platform mobile development using C# and Xamarin Forms

Runar Ovesen Hjerpbakk

<https://hjerpbakk.com>

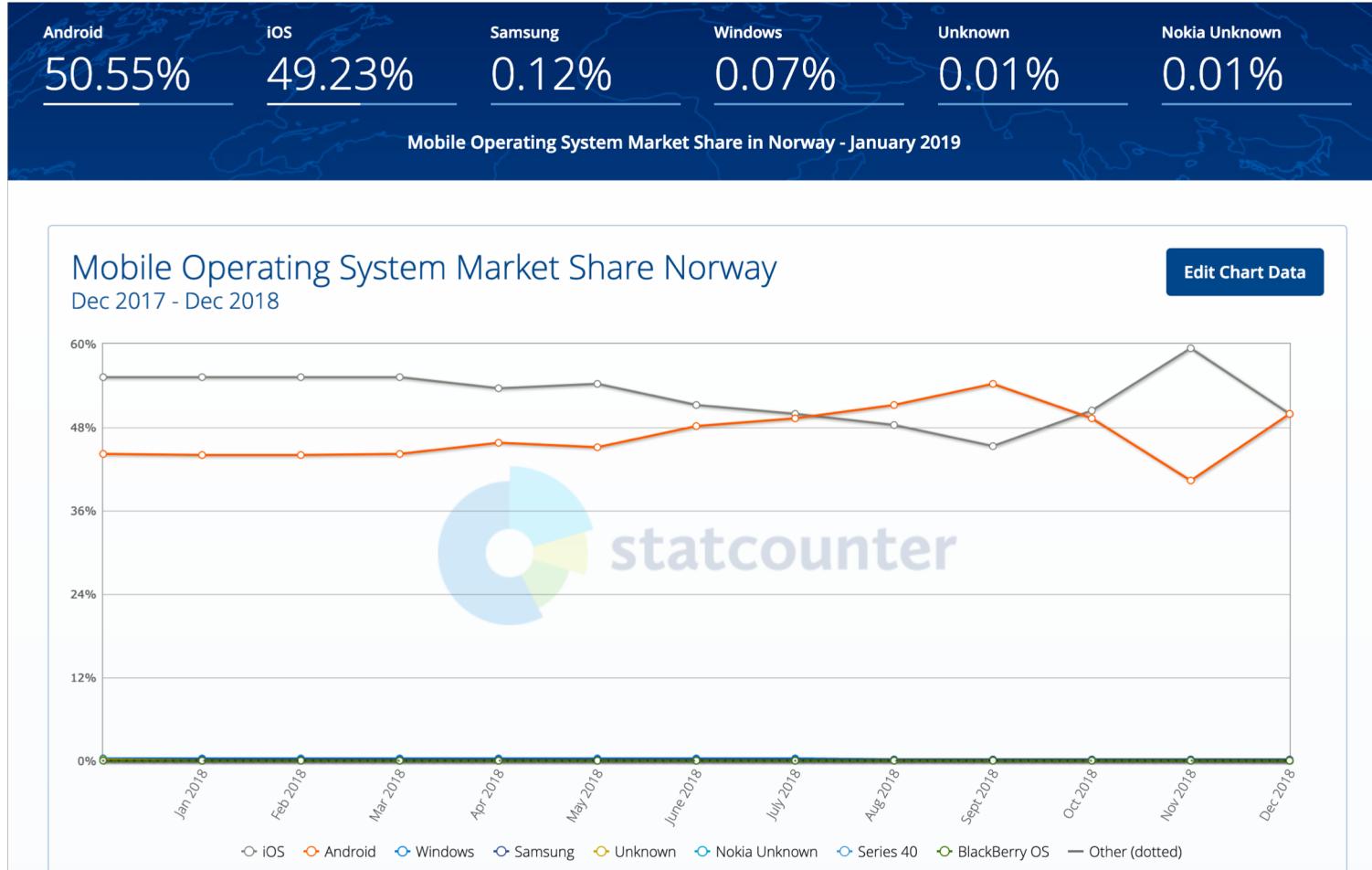
@hjerpbakk

Agenda

- Motivation
- Why Xamarin?
- How Xamarin?
- Patterns and practices
 - Code sharing through standard libraries
 - SOLID
 - MVVM
 - ValueConverters
 - Dependency Injection
 - Testing

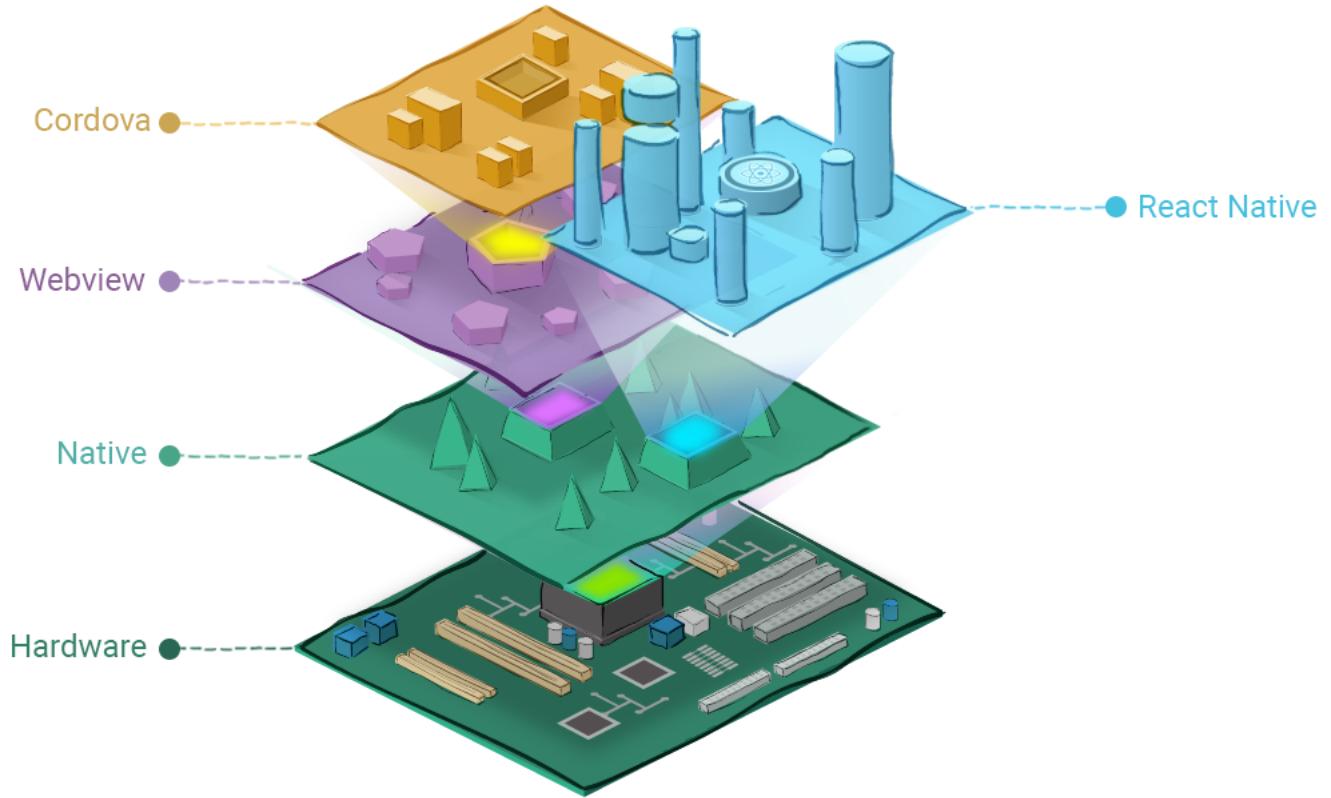


Norwegian market share



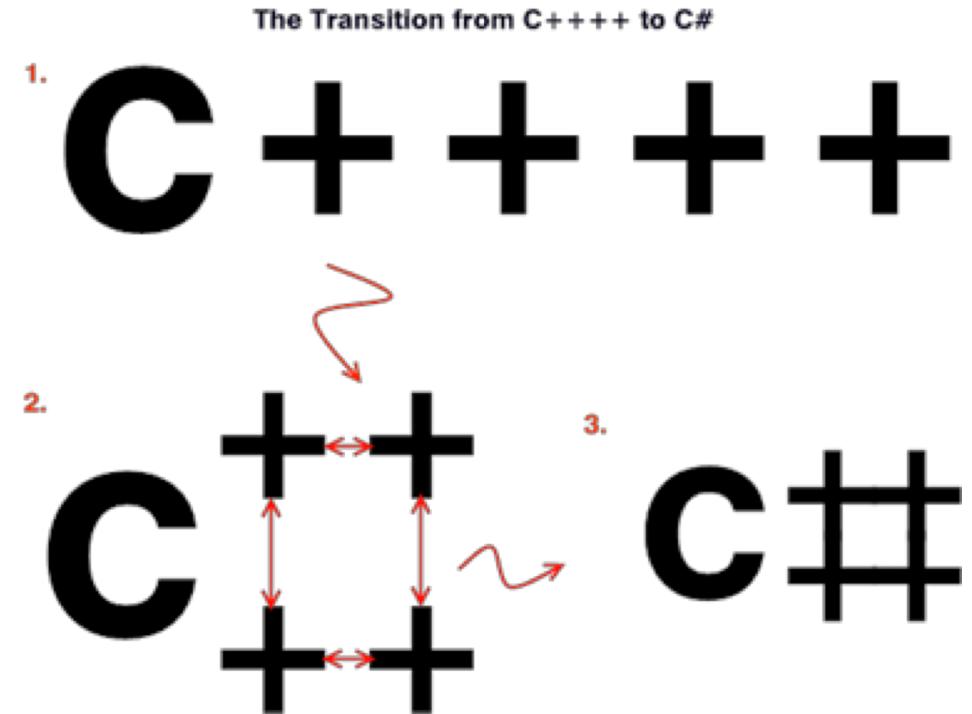
3 popular approaches

- Native
- Using webviews (Cordova, PhoneGap, Ionic)
- Native with different languages (Xamarin, React Native)



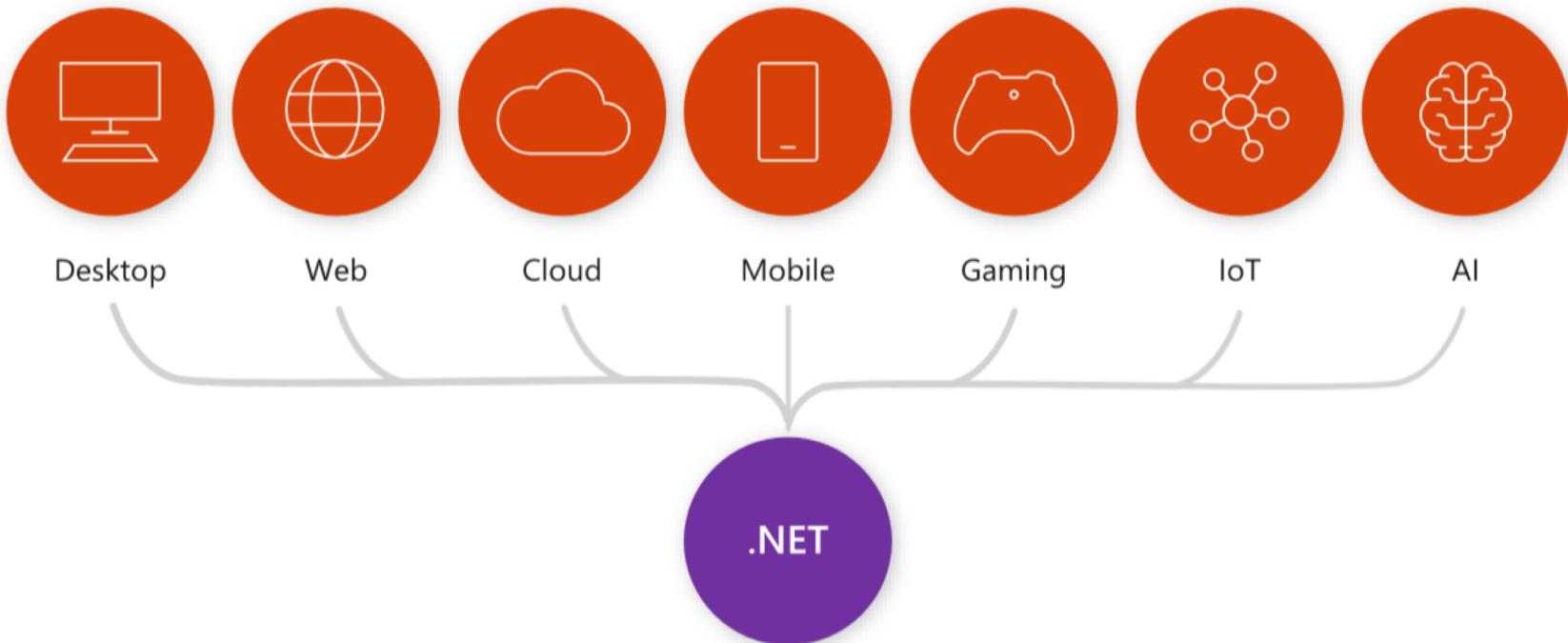
C#

- Wikipedia: «C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines.»
- Runar Ovesen Hjerpbakk:
«C# ligner på Java, alt er bare bedre.»

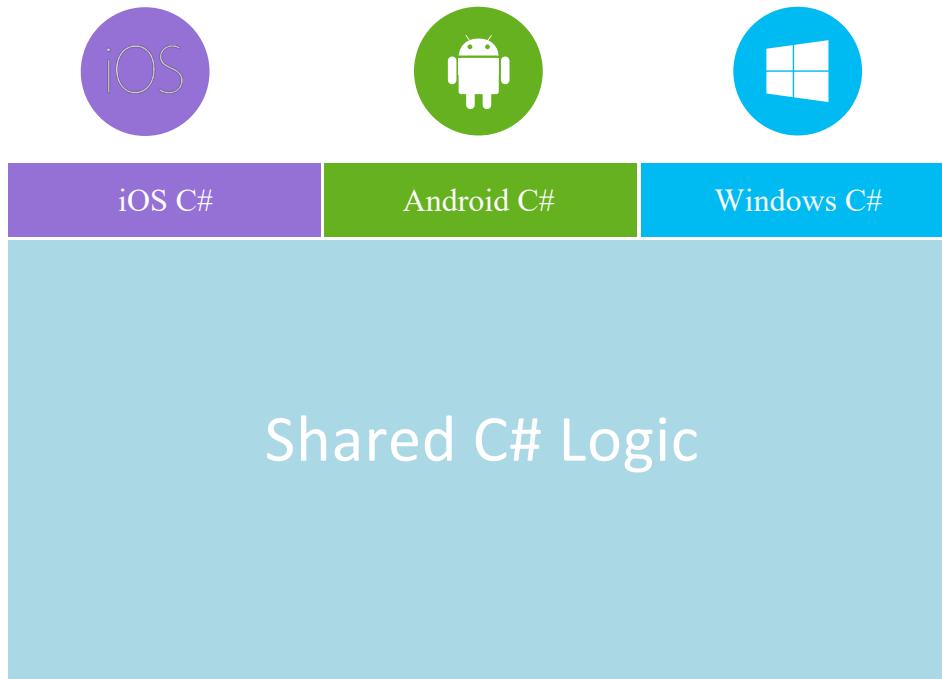


.Net

Your platform for building anything



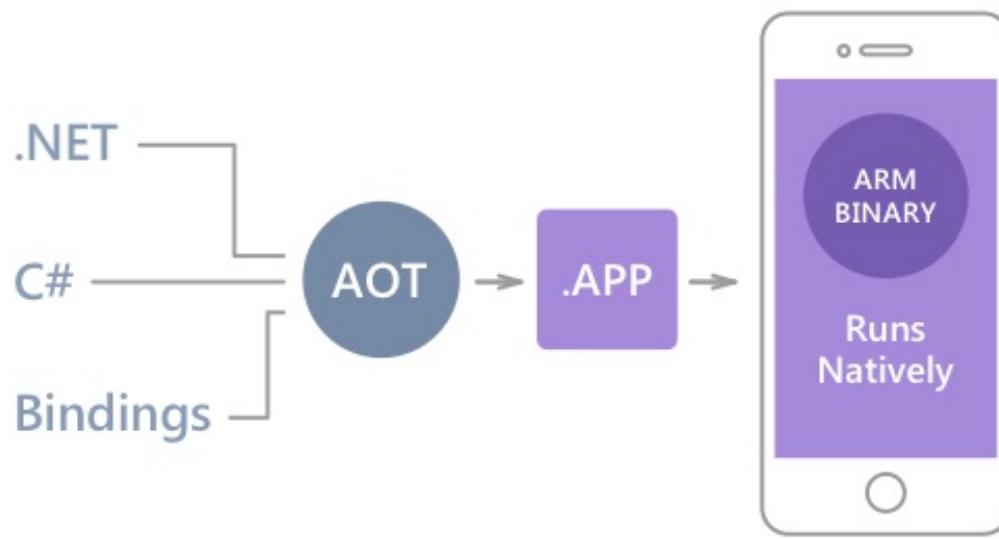
Traditional UI approach



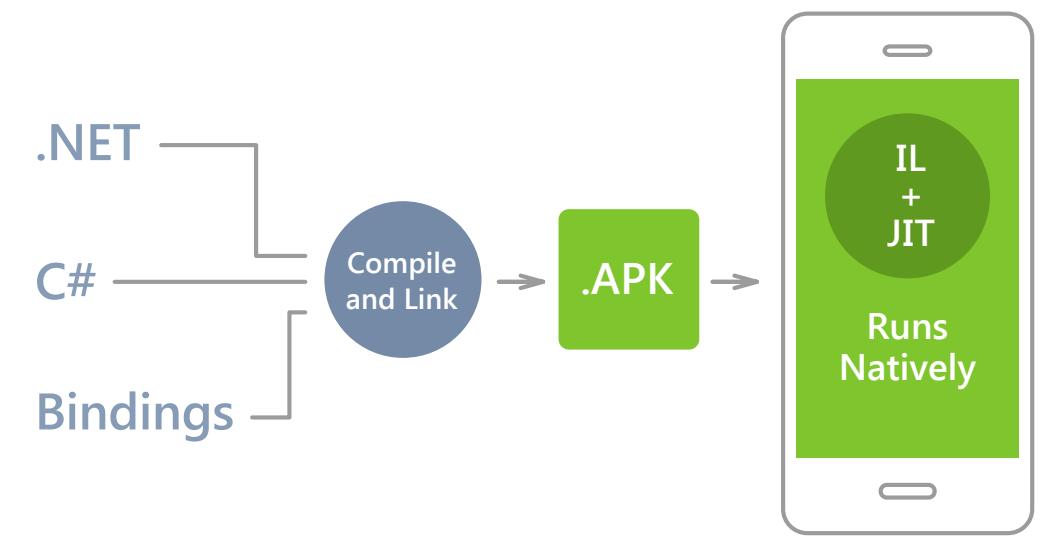
3 Native User Interfaces
Shared App Logic



Native Performance

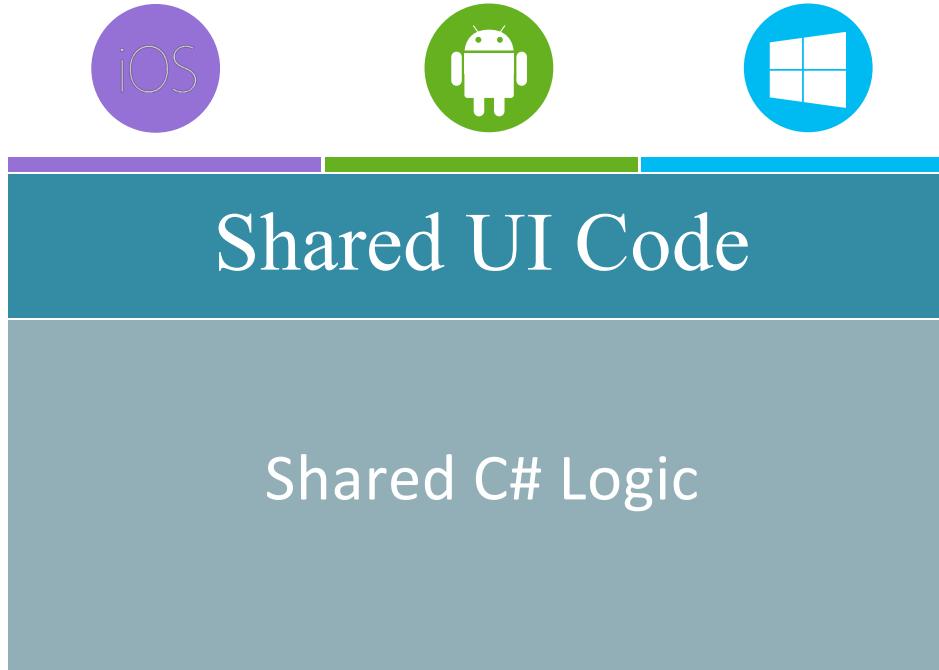


Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store.

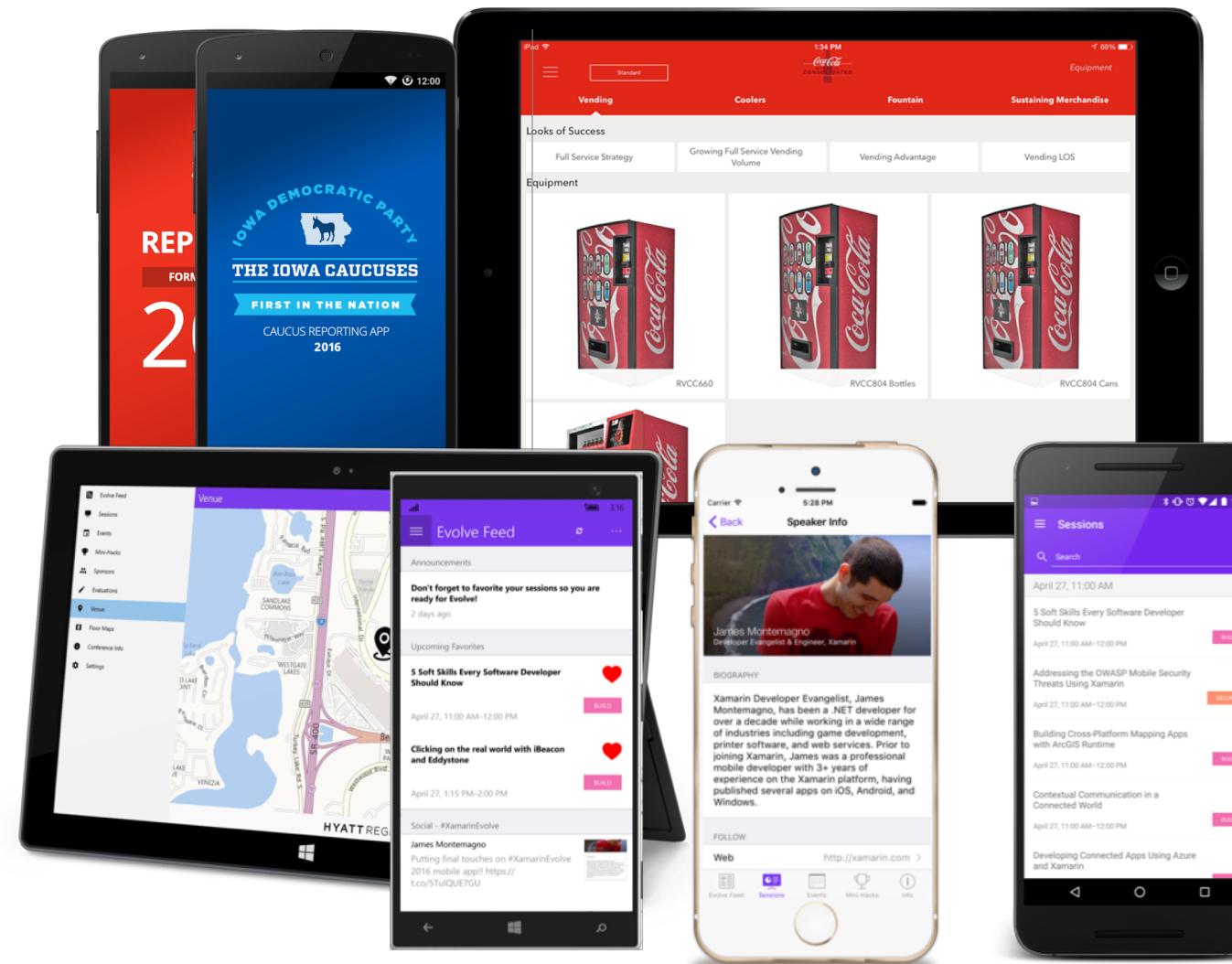


Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device.

Xamarin.Forms approach

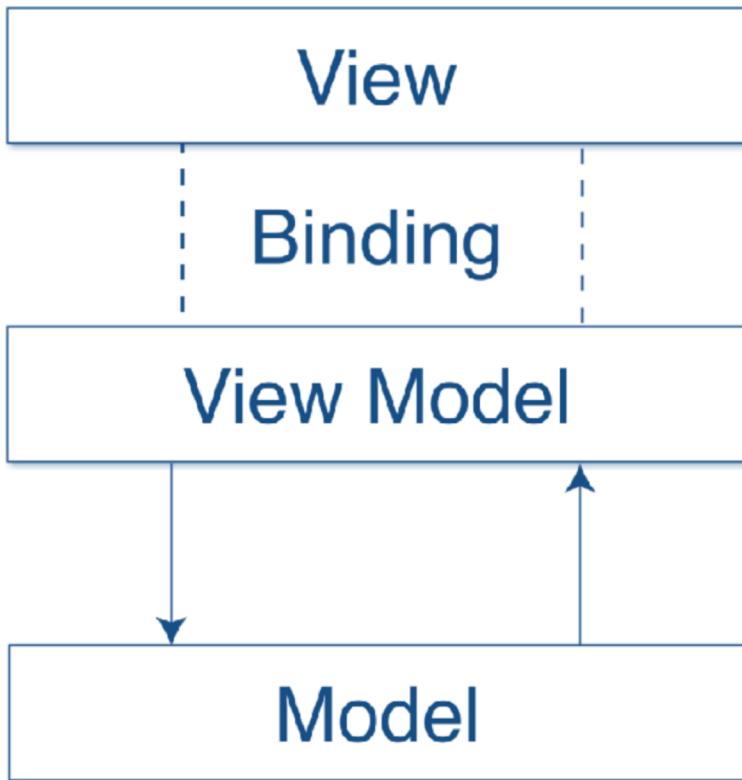


Shared User Interface
Shared App Logic





What is MVM?



- Design pattern used for building UI based apps - originally invented by Microsoft for WPF
- Model is business logic
- View is pure UI, no logic
- View Model is UI logic
- Binding wires up view to view model