A Career in Mobile Development

Shawn Castrianni

shawn@spcware.com

MY BACKGROUND

My Software Development Career Timeline

- **June 1990:** Graduated high school from Plano East Senior High
- **June 1994:** Graduated B.S. Electrical Engineering from Texas A&M University
- **December 1996:** Graduated M.S. in Computer Science from Texas A&M University
- **January 1997:** Software Engineer at Schlumberger
  - C/C++ and Motif UI for Solaris
- **July 2000:** Systems Architect at Halliburton
  - Java UI for Windows/Linux/Solaris
- **January 2008:** Chief Configuration Management Architect at Halliburton
  - Java/Ant build automation
- **January 2012:** Principal Mobile Architect at IHS
  - C#/Xamarin UI for iOS/Android

My Mobile Development Timeline

- **December 2009**: Developed first Android app using pure Java
- **March 2010**: Developed second Android app using pure Java
- **July 2010**: Discovered Xamarin.iOS (aka MonoTouch)
- **November 2010**: Developed first iOS app using Xamarin
- **January 2011**: Developed third Android app using pure Java
- **April 2011**: Developed second iOS app using Xamarin with inapp purchases
- **December 2011**: Published Android app to Amazon store
- **April 2012**: Developed first Windows Phone app using pure C#
- **April 2013**: Helped to develop first app to use Xamarin for both iOS and Android platforms
- **May 2014**: Helped to convert cross platform app to MvvmCross architecture
- **May 2014**: Created prototype using new Xamarin.Forms architecture

MOBILE DEVELOPMENT CAREER

Job Duties

- Develop front-end mobile UI
- Develop back-end web services
- Agile User Story management
- Bug fixing
- Work estimation
- Automated builds
- Automated testing
- Demos

Mobile Industry Perks

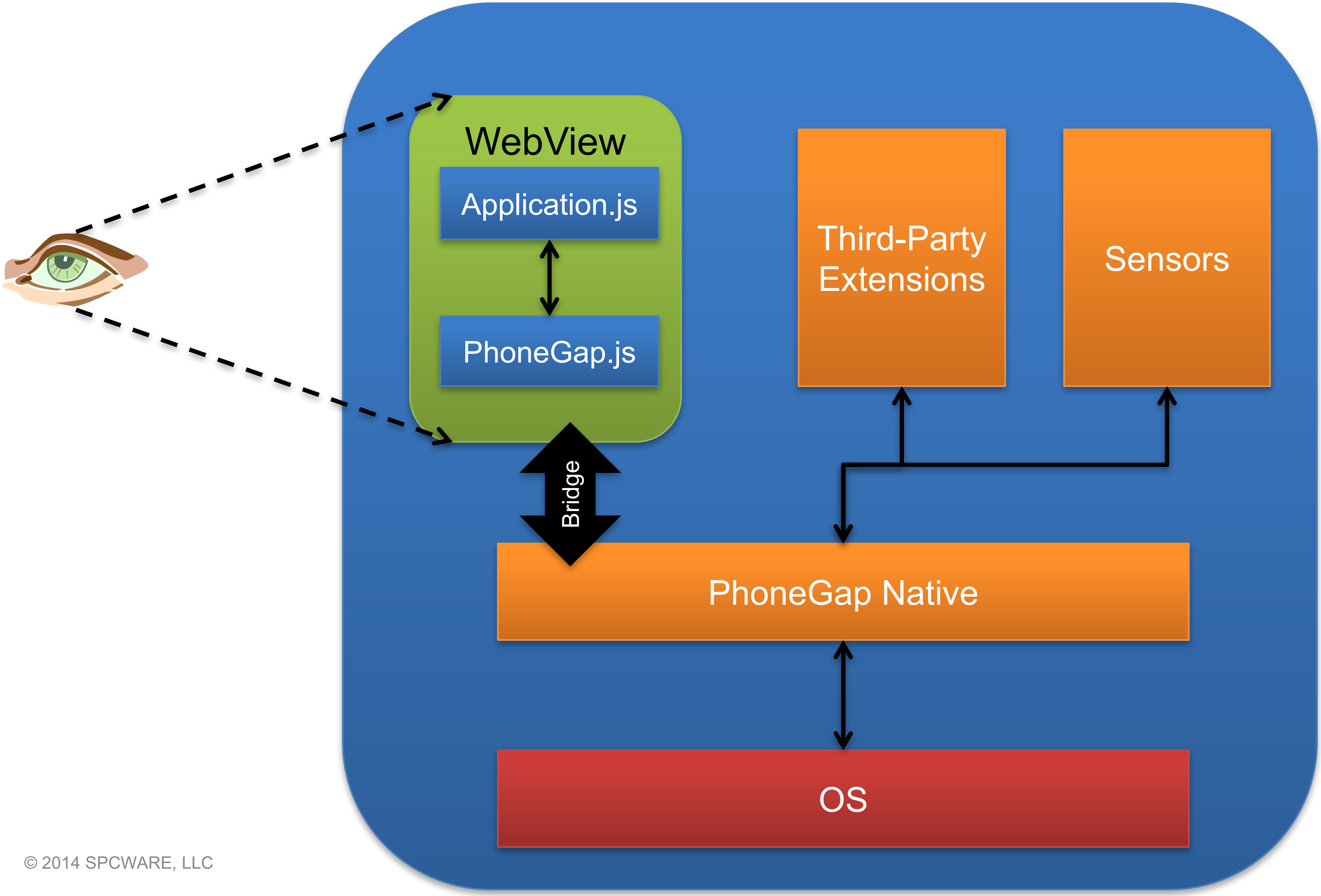
- Multiple free smart phones
- Multiple free tablets
- Free cellular data plan
- Can work from home or anywhere
- Free knowledge on the Internet

MOBILE DEVELOPMENT TECHNIQUES

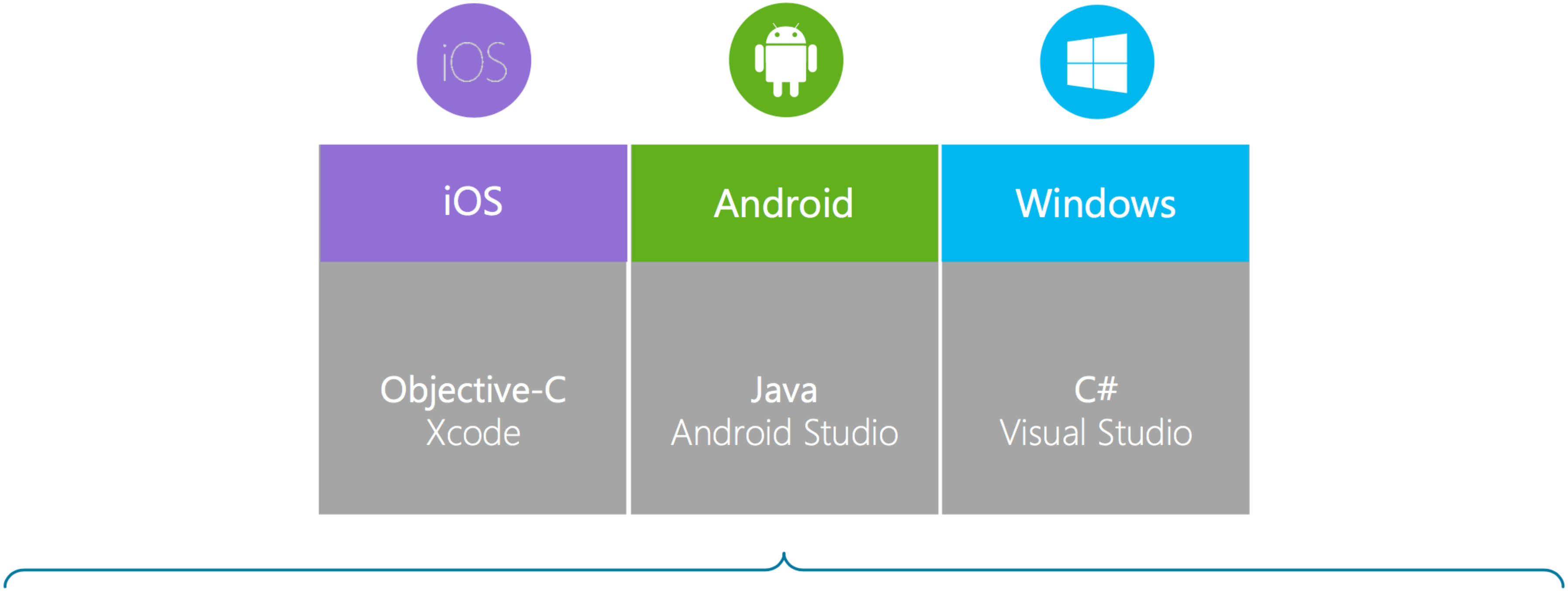
Mobile Techniques

- Several techniques to choose from:
  - Desktop-Style web app (XYZ.com)
  - Mobile web app (m.XYZ.com)
  - Embedded mobile web app (PhoneGap/Cordova)
  - Cross platform native (Xamarin)
  - Pure native (Java, Objective-C/Swift, C#)
- Techniques that do not use native UI controls try to imitate native look and feel

Embedded Mobile Web



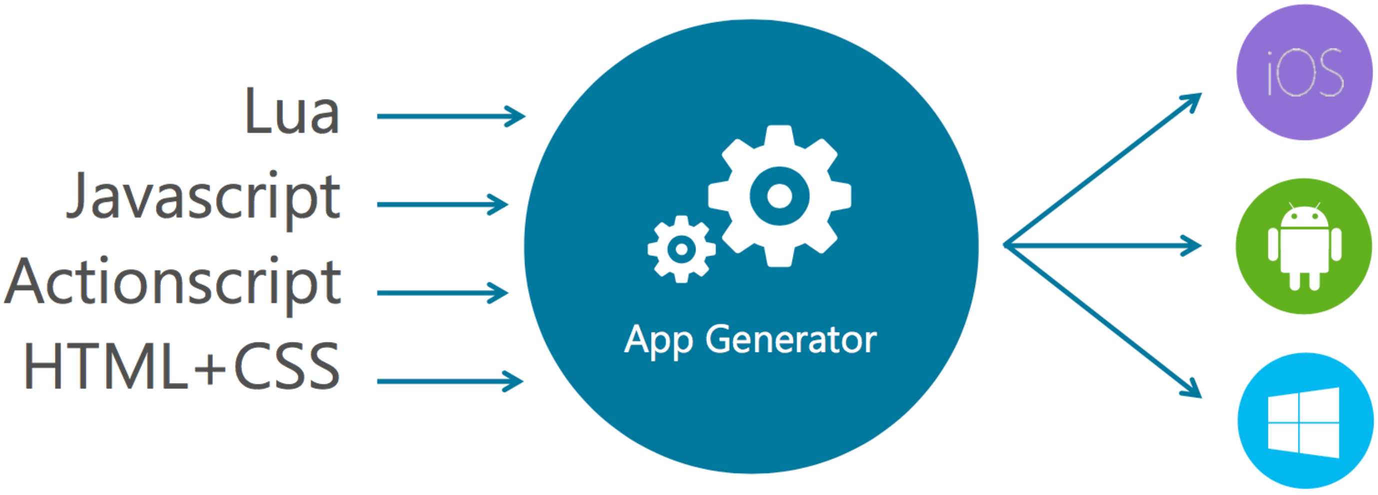
Silo Approach



iOS	Android	Windows
Objective-C Xcode	Java Android Studio	C# Visual Studio

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

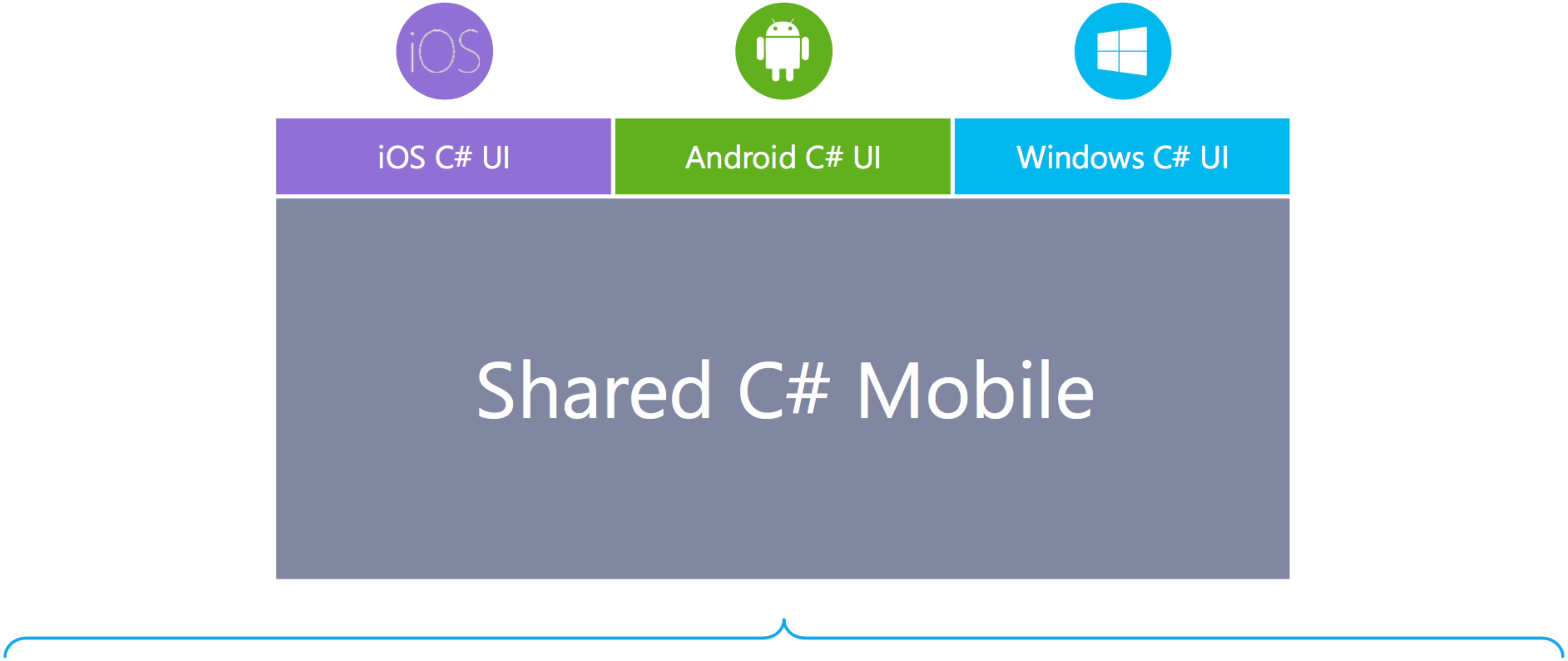
Write Once, Run Anywhere



Limited native API access • Slow performance • Poor user experience

XAMARIN

Xamarin Platform



iOS C# UI

Android C# UI

Windows C# UI

Shared C# Mobile

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

Xamarin

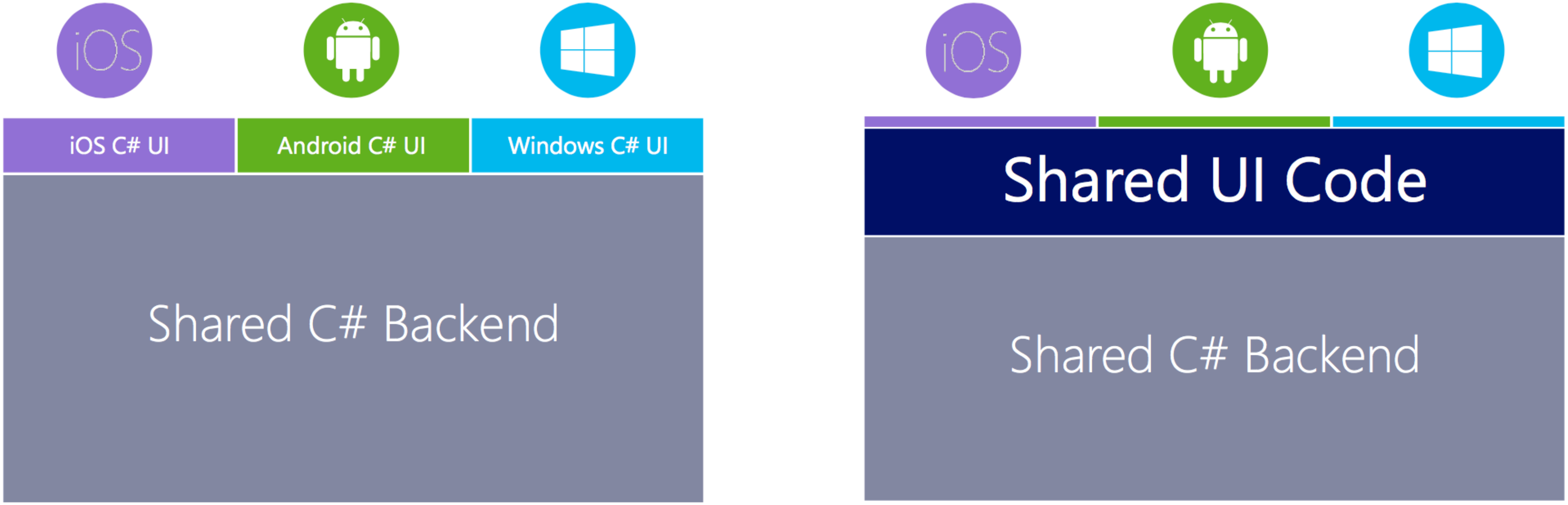
- Provides a .NET runtime for mobile platforms (and Mac too)
- Allows C# development using Visual Studio
- Provides 1-to-1 C# bindings for underlying native mobile SDK, updated same day as native SDK releases
- Can use Xamarin Studio on Mac platform
- Full power and performance of underlying native platform but with an average of 65% (see upcoming slides for more details) code reuse across all platforms

Xamarin .NET – Advantages

- Common network access (REST and SOAP)
- Common local DB access
- Common XML/JSON parsing
- Common I/O
- Full Object Oriented language design with C#:
    - Garbage collection
    - Lambda expressions
    - LINQ
    - Async/await parallel API
    - Generics
    - Portable Class Libraries (PCL)
    - Native interop

XAMARIN.FORMS

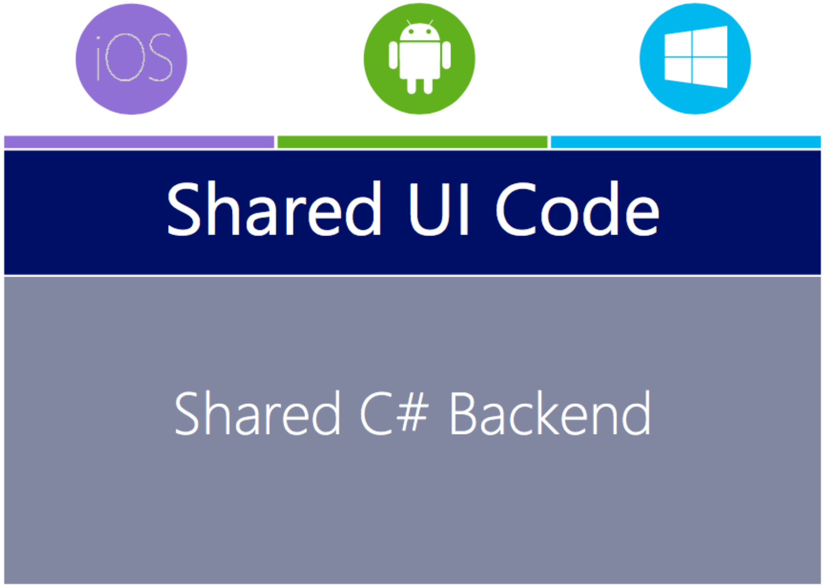
Xamarin.Forms Platform



Traditional Xamarin Approach

With Xamarin.Forms:
More code-sharing, all native

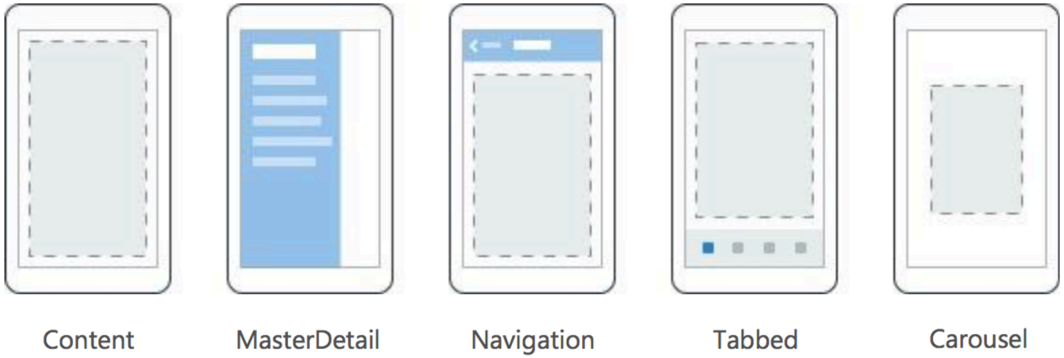
What's Included



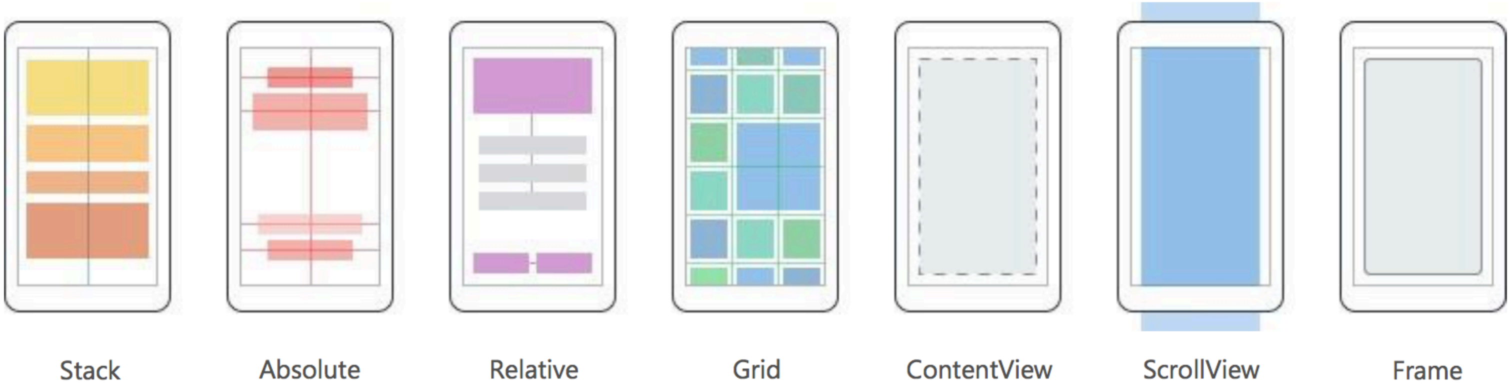
- ✓ 40+ Pages, layouts, and controls
  (Build from code behind or XAML)
- ✓ Two-way data binding
- ✓ Navigation
- ✓ Animation API
- ✓ Dependency Service
- ✓ Messaging Center

Pages and Layouts

Pages



Layouts



Controls

ActivityIndicator	BoxView	Button	DatePicker	Editor
Entry	Image	Label	ListView	Map
OpenGLView	Picker	ProgressBar	SearchBar	Slider
Stepper	TableView	TimePicker	WebView	EntryCell
ImageCell	SwitchCell	TextCell	ViewCell	

DEMO

BOUNCE

Bounce Details

- Uses Xamarin.Forms for UI
- Uses SkiaSharp for 2D graphics
- Uses CocosSharp for game engine/physics
- Media plugin for access to camera and photo library
- Source code: https://github.com/scastria/Bounce