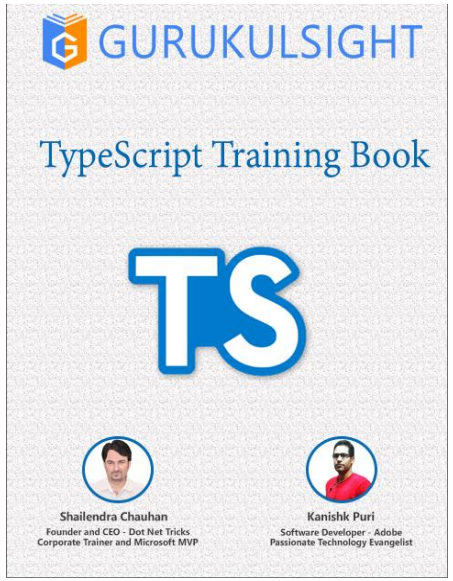
Xamarin + Microsoft Azure

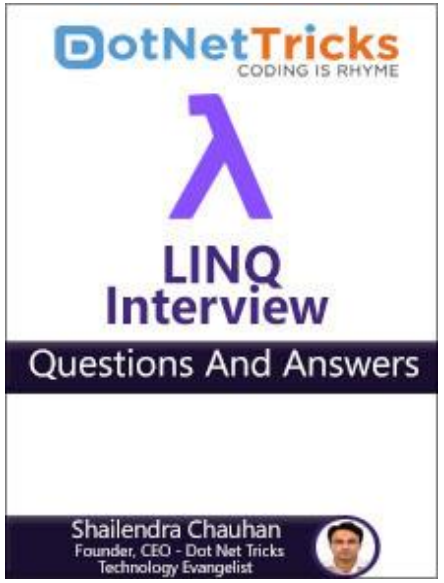
Shailendra Chauhan

Microsoft MVP, Founder & CEO - Dot Net Tricks

DotNetTricks

Author of Most Popular Free e-Books





DotNetTricks

Enterprise Mobile Considerations

Mobile Apps Frameworks

1st Gen

Platform Proprietary


Objective-C / Swift


Android Java


Windows .NET

Native UX
High perfomance
Multi-platform
Unified codebase
Hardware & platform access

2nd Gen

Hybrid HTML & JavaScript Frameworks

Native UX
High perfomance
Multi-platform
Unified codebase
*Hardware & platform access

3rd Gen

Cross- Platform Native


Xamarin


React Native

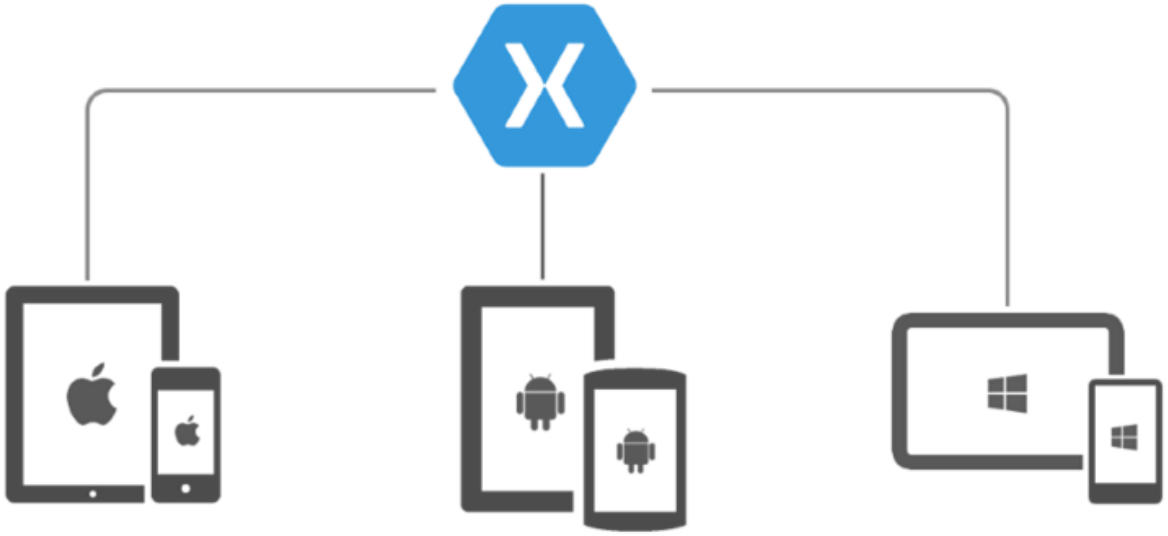

NativeScript

Native UX
High perfomance
Multi-platform
*Unified codebase
Hardware & platform access

DotNetTricks

Xamarin

- An open-source .NET solution run on Mono & .NET framework
- C# based apps run on iOS, Mac, Android and Windows
- Build truly native apps with native performance and Native UI



Xamarin - A Better Model for Mobile Development

One Language
One Toolset

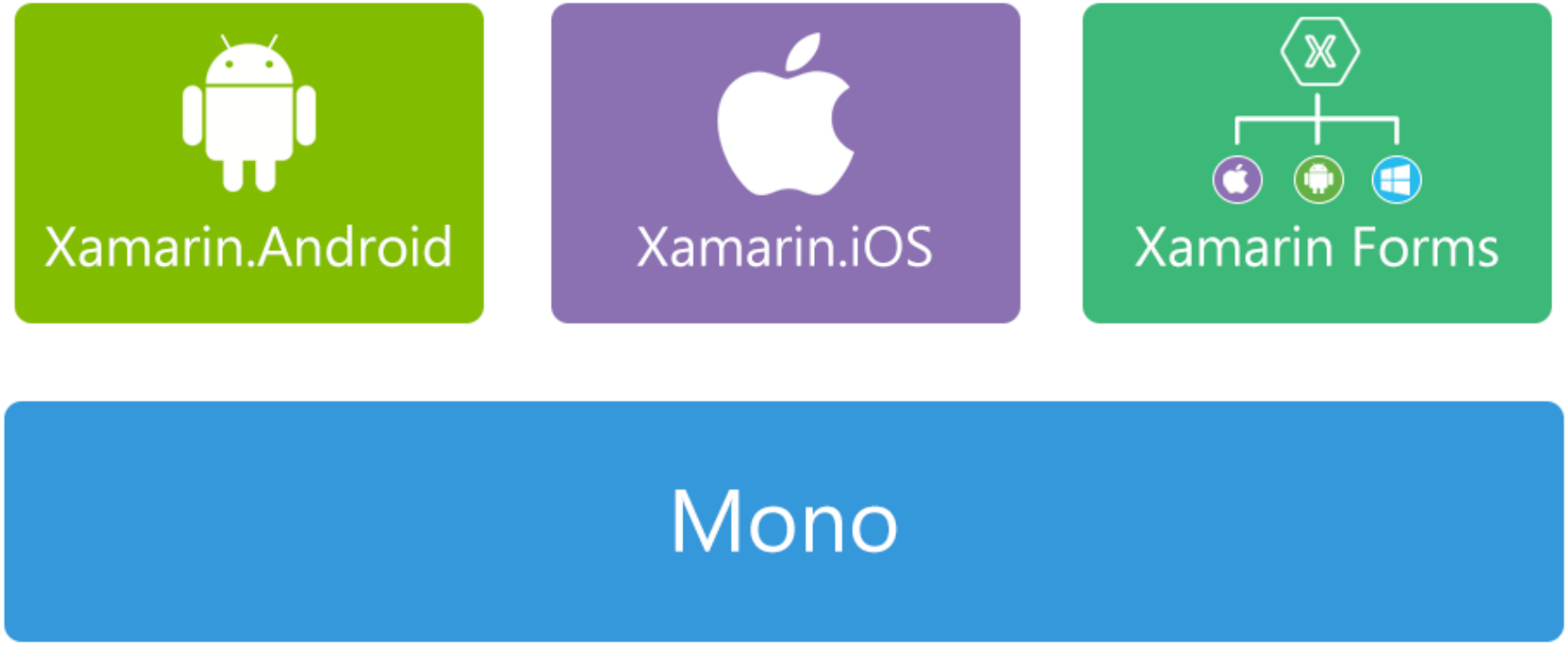
Native API/Performance

Native UI/Performance

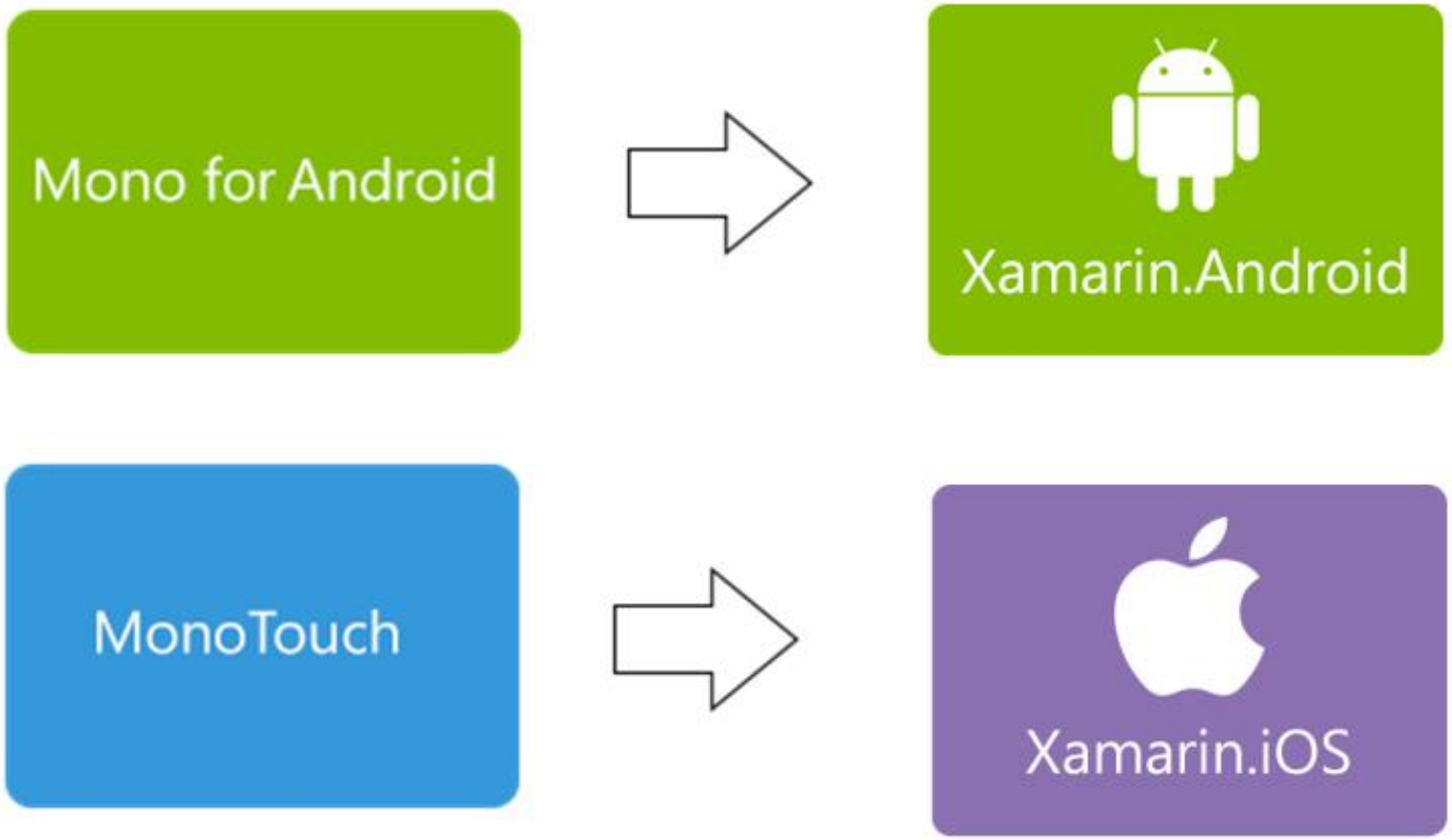
Xamarin Forms - Single UI Framework

Source control Build System Testing Monitoring

Xamarin Architecture



Mono



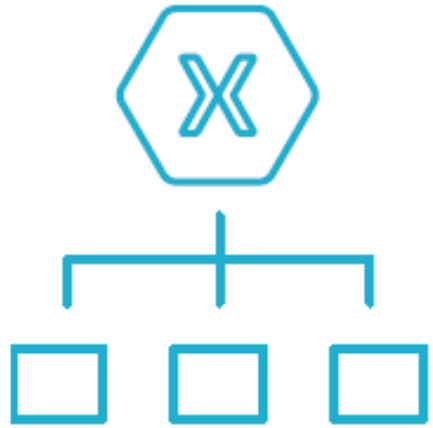
Xamarin Family

Xamarin.Android

Native mobile apps for Android, Android Wear, and Android TV

Xamarin.iOS & Xamarin.Mac

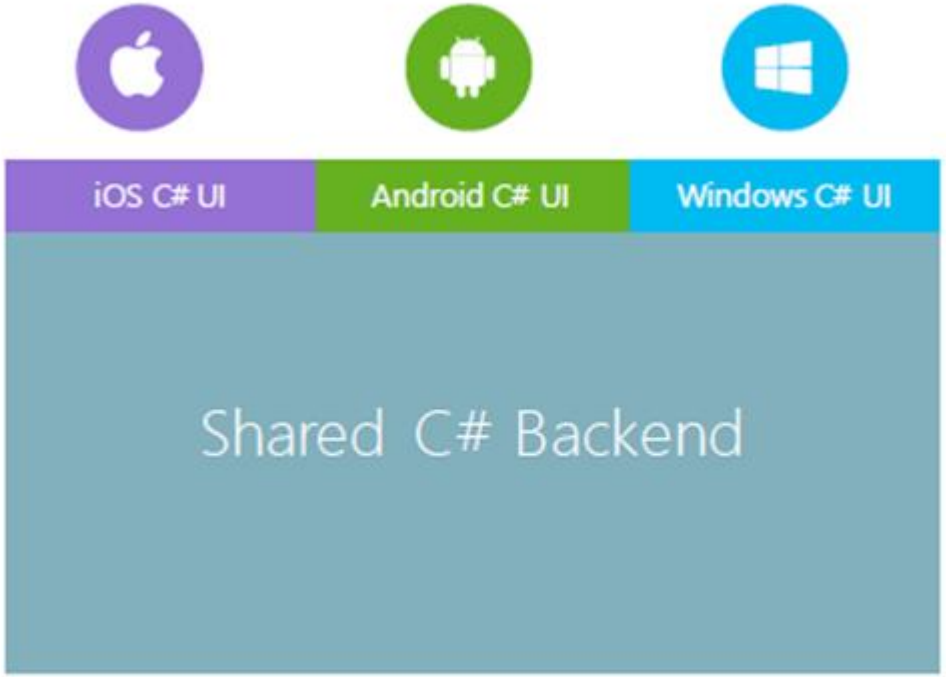
Native mobile apps for iOS, watchOS, tvOS, and OS X



Xamarin Forms

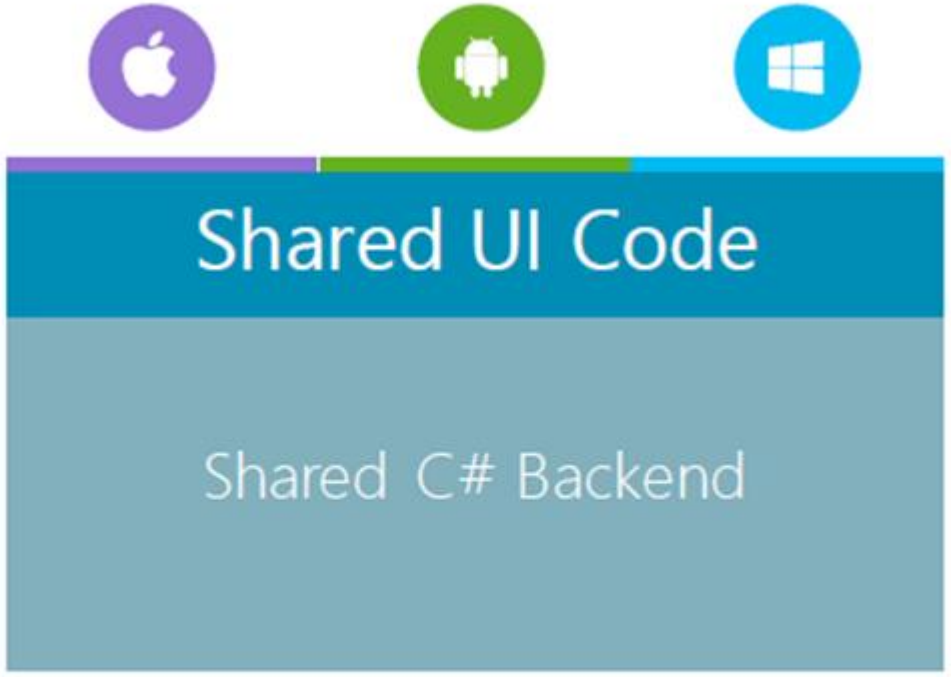
Native UIs for iOS, Mac, Android & Windows from a single, shared codebase

Xamarin App Development Approaches



Xamarin Unique Approach

100% Native API Access



Xamarin Forms

More Code Sharing, 100% Native

Which Xamarin approach is best for your app?

Xamarin.Forms is best for:

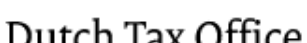
- Apps that require little platform-specific functionality
- Apps where code sharing is more important than custom UI
- Developers comfortable with XAML

Xamarin.iOS & Xamarin.Android are best for:

- Apps with interactions that require native behavior
- Apps that use many platform-specific APIs
- Apps where custom UI is more important than code sharing

Who are using Xamarin?

Over 15,000 companies rely on Xamarin.



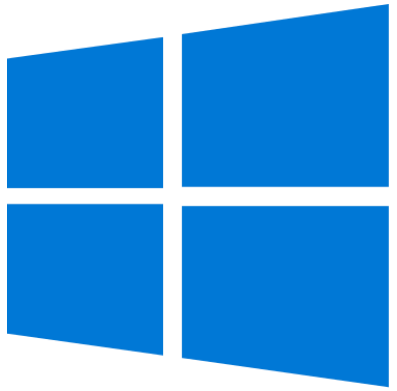
Xamarin Apps Show Case?

Development IDEs

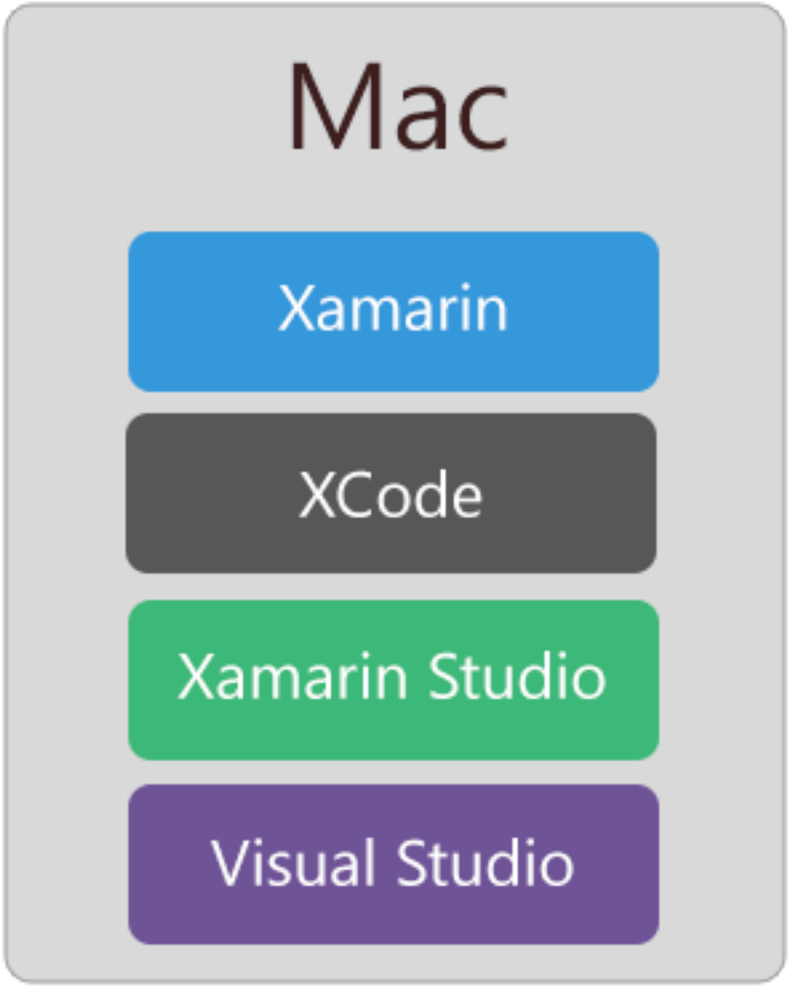
Visual Studio for Mac

Visual Studio for Windows

Development SDKs



Development with Mac



Android

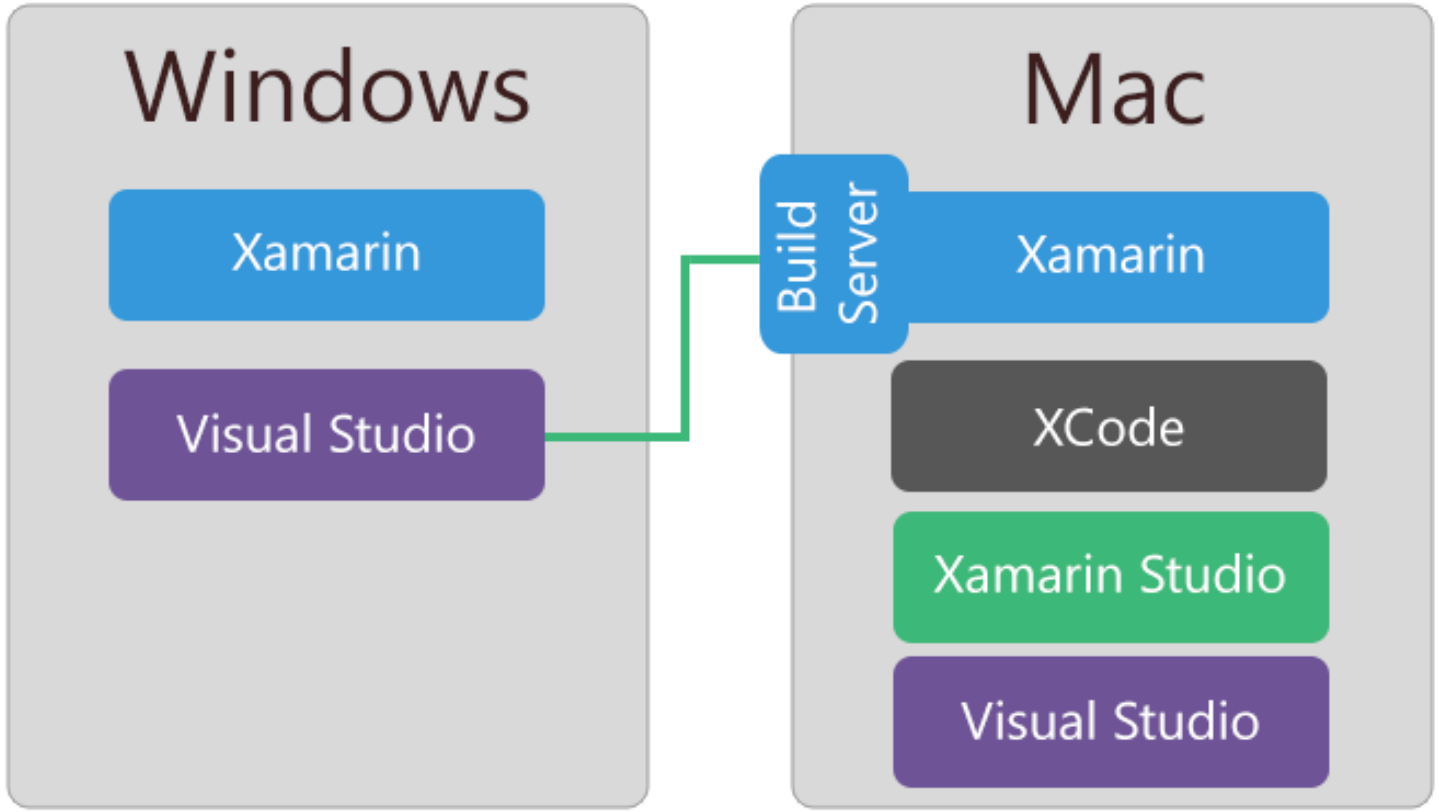
iOS

Development with Windows

Android

UWP

Development for All Devices



Android

iOS

UWP

Why Xamarin?

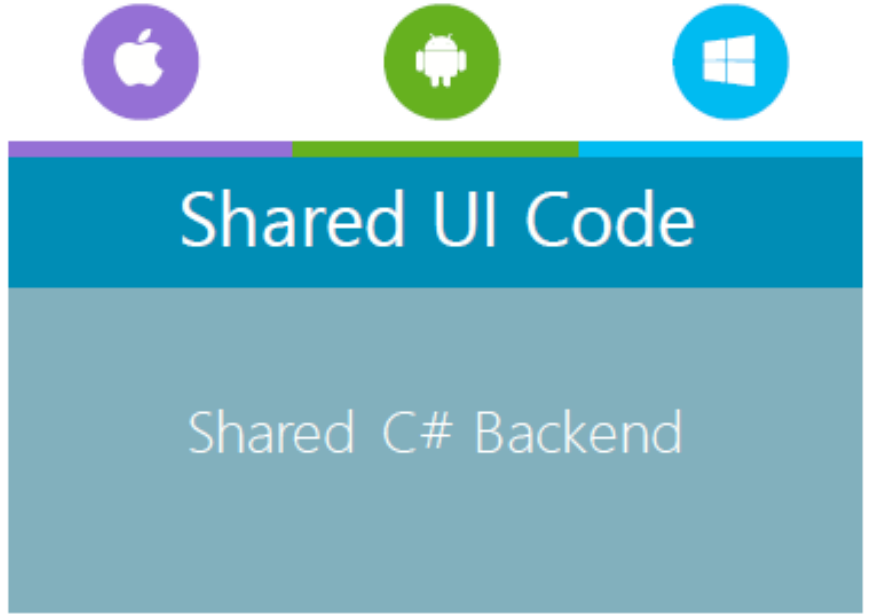
- Target iOS, Android, MacOS & UWP from single code base
- Reach Smart Watches and Smart TVs
- Reuse existing skills in C#/XAML
- Familiar software paradigms
- Consistent cross-platform APIs
- Top-notch tooling/IDEs – Visual Studio on Windows or Mac
- Great deployment & testing tools
- Lots of frameworks, utilities & Standardization efforts
- Support for full DevOps workflows
- Fantastic developer community

Xamarin Limitations

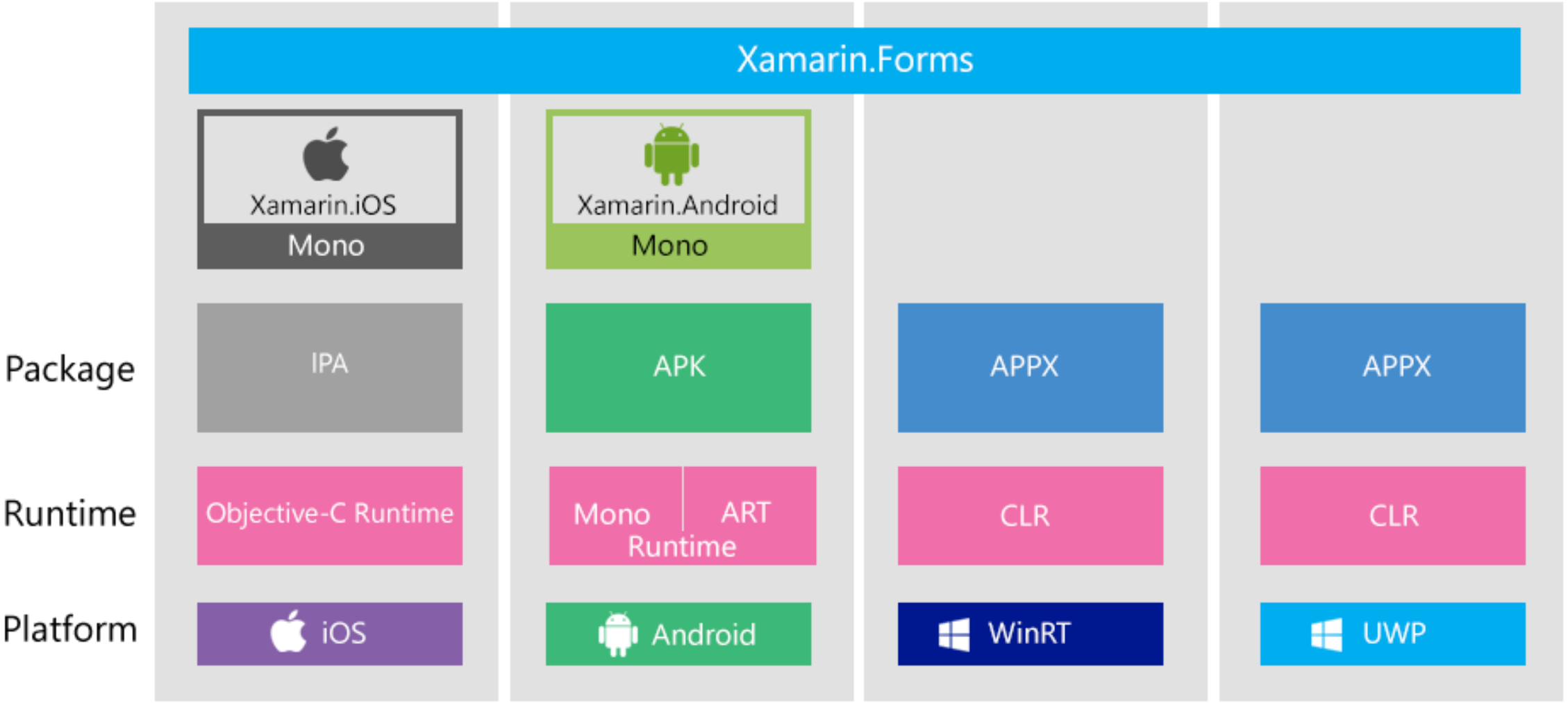
- Limited Dynamic Language Support
- Limited Generics Support
- Limited Sharing of UI Code Across iOS, Mac and Android
- Limited .NET API Support

Xamarin Forms

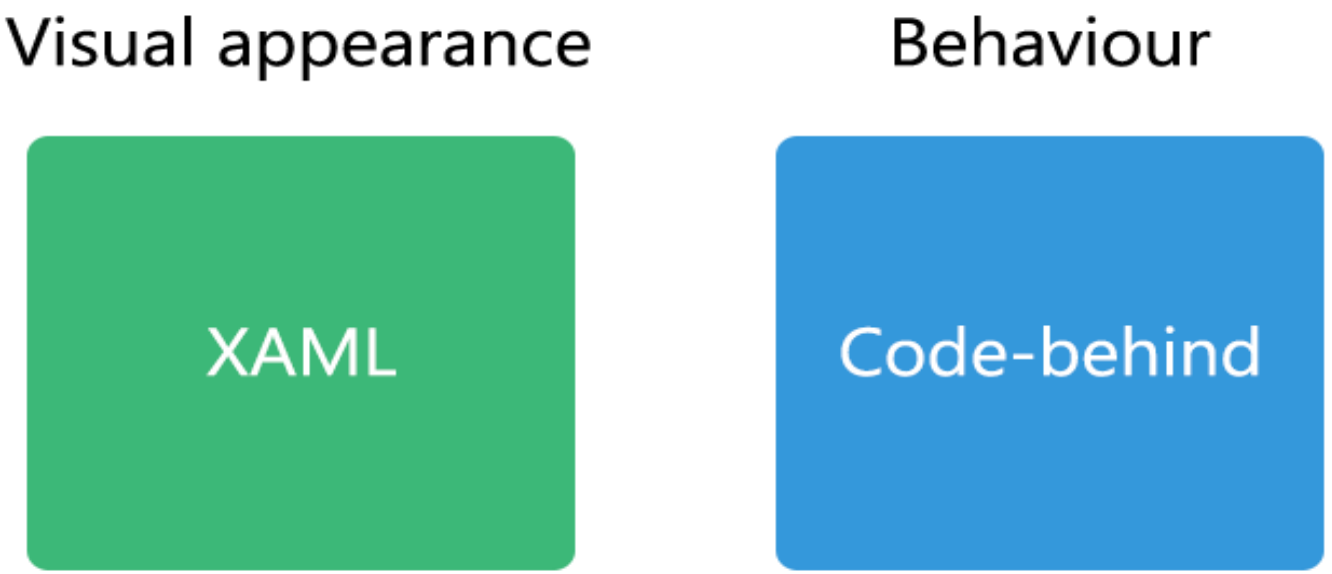
- A UI library which is compiled for iOS, Android, and Windows phones devices to develop native cross platform app
- Provides Pages, Layouts, Views and Cells to develop UI
- Allows you to access the platform specific features through injection with UI customization and custom renderers



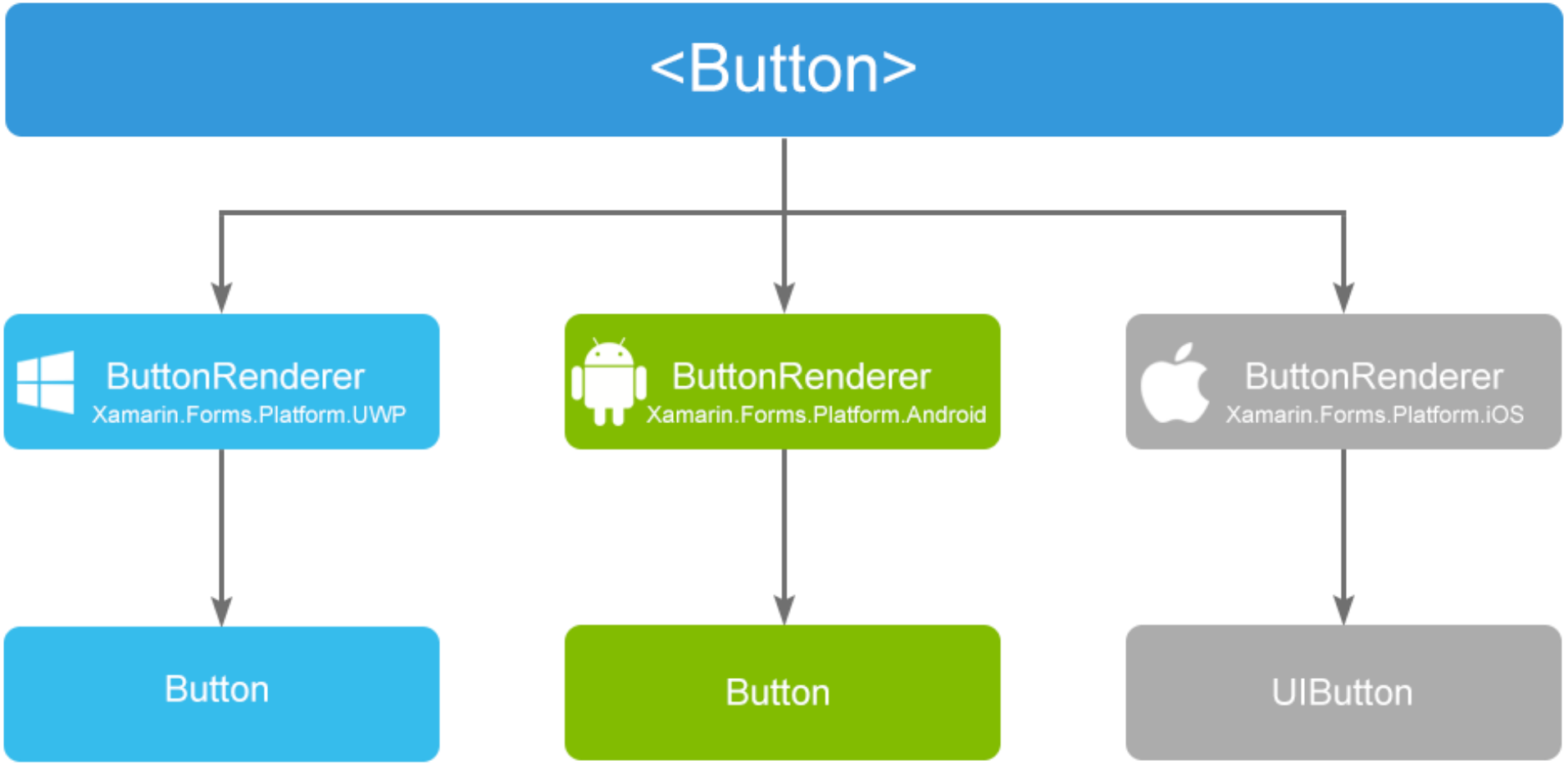
Xamarin Forms Architecture



Xamarin Forms UI

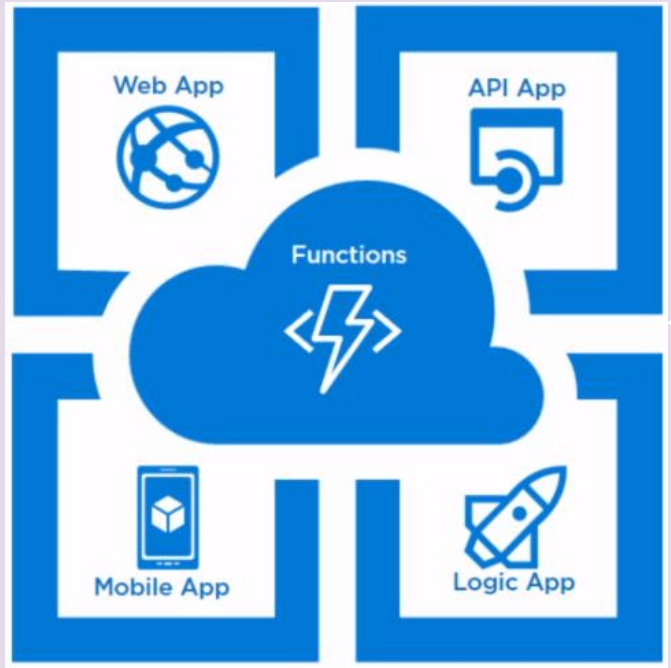


Xamarin Forms UI Rendering Process



Mobile + Cloud

Azure App Service



Build and scale great cloud apps

.NET, Java, Node.js, PHP, Python

Auto patching

Auto scale

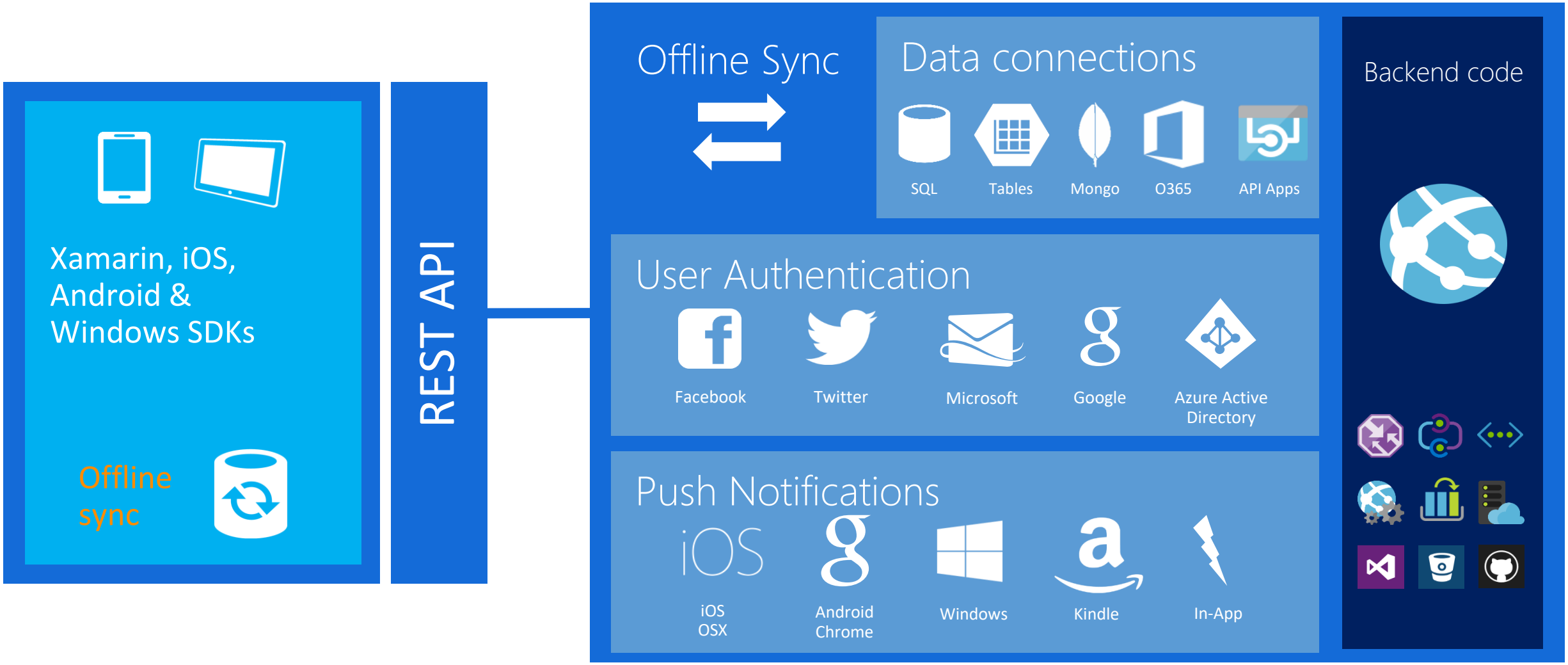
Integration with existing apps

Continuous deployment

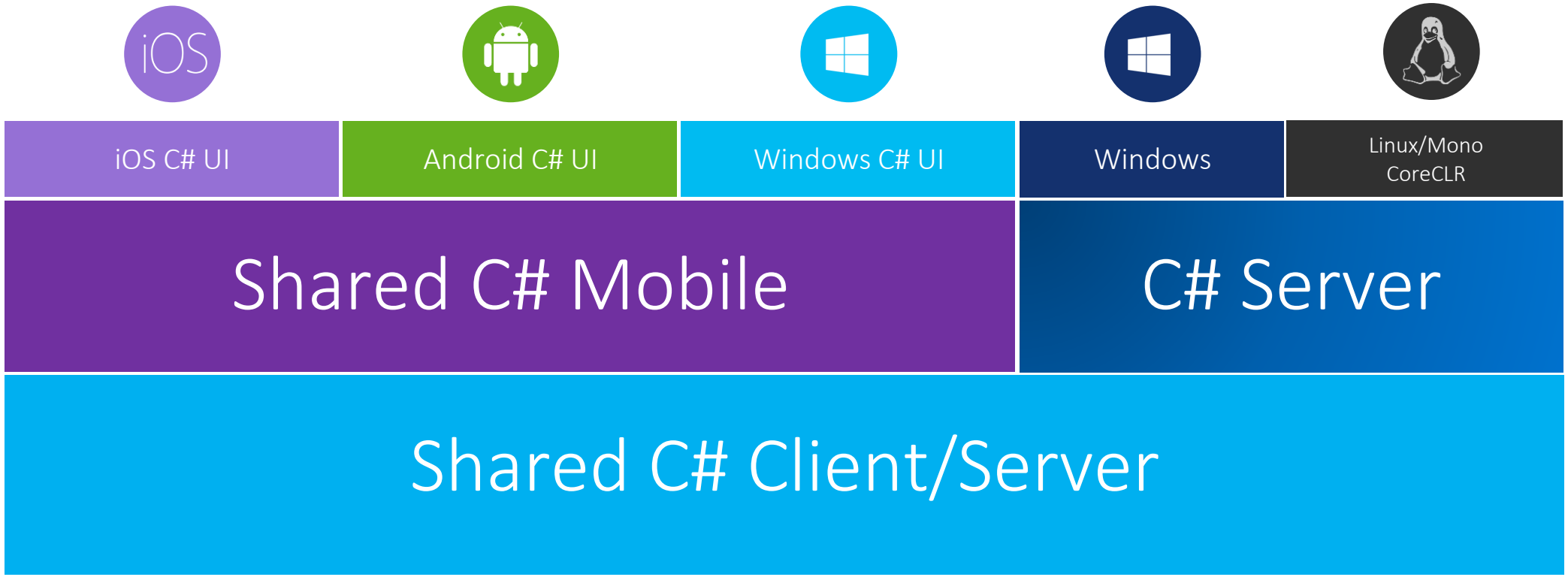
Why Azure Mobile Apps?

- Extremely powerful
- Flexible
  - Easy Tables (Node.js Backend)
  - App Service (ASP.NET Backend)
- Cross-platform client SDK
  - iOS, Android, and Windows
- Open source C#
  - Backend and Client SDKS on GitHub

Azure Mobile Apps

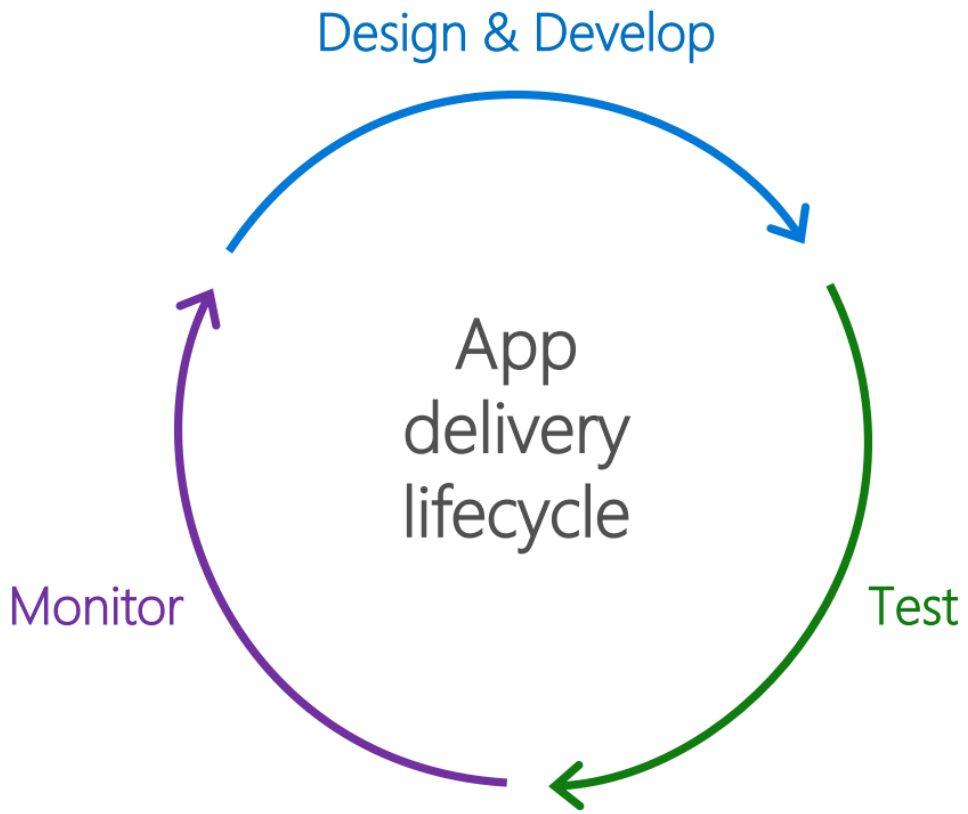


Mobile + Server (Xamarin and C#)

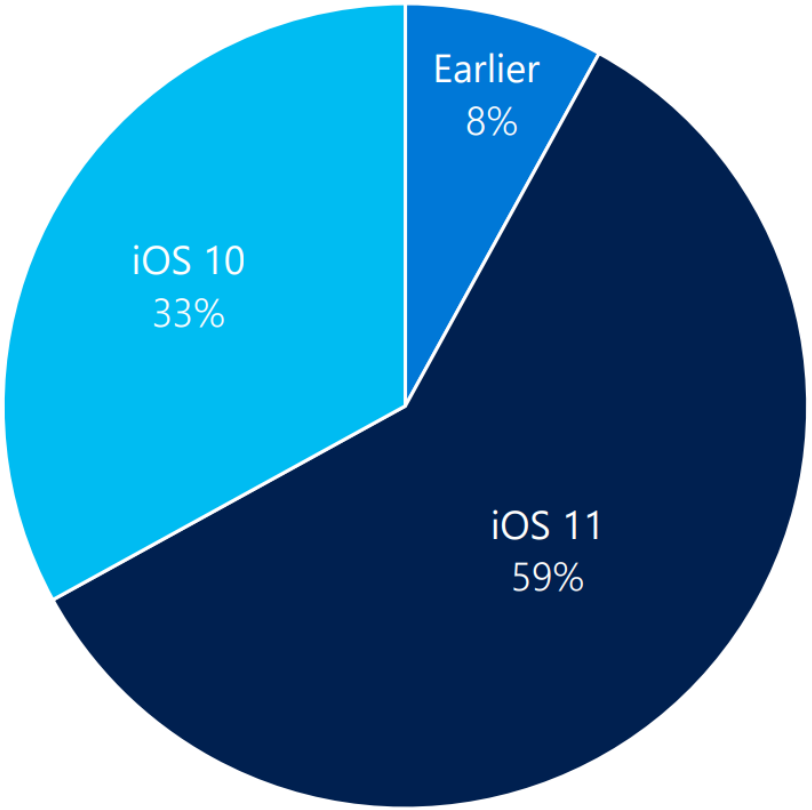


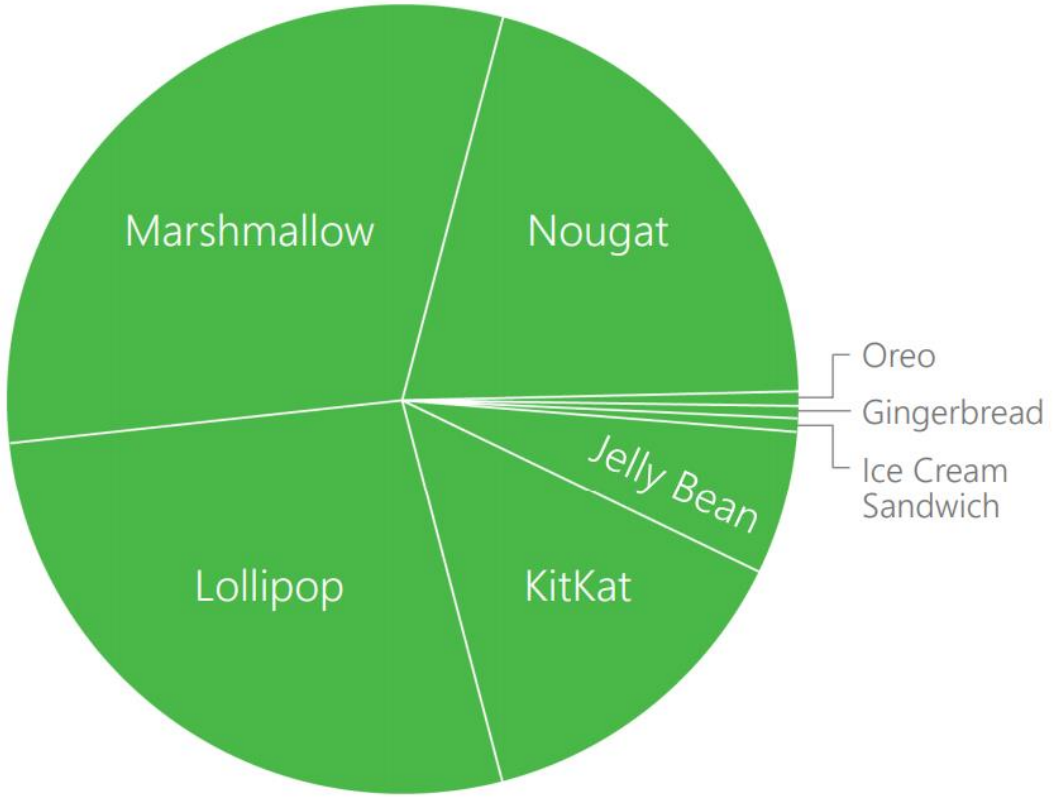
Shared C# codebase • 100% native API access • High performance

Mobile App Delivery Cycle



Testing on different OS Versions





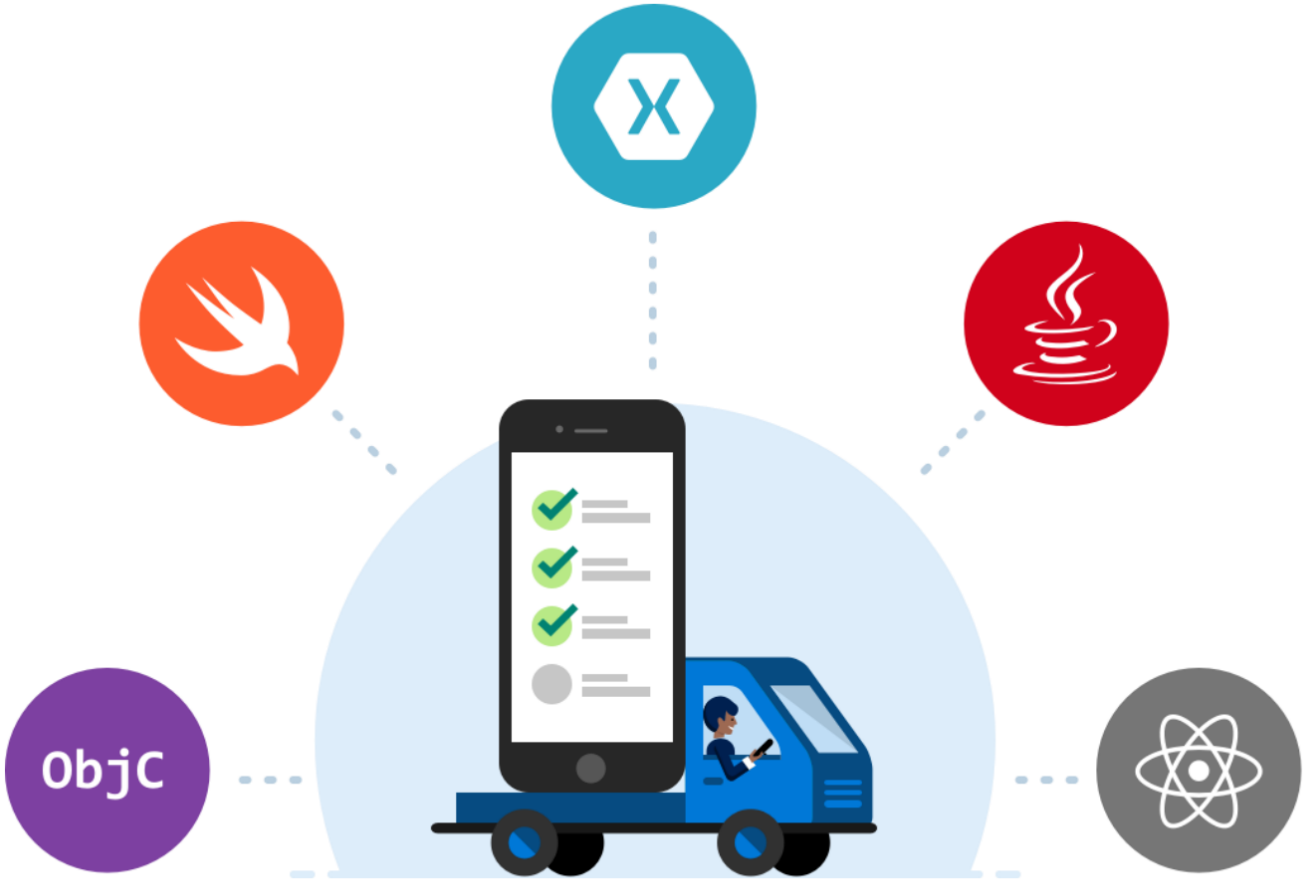
Introducing Visual Studio App Center

The Platforms you need

- ✓ Android
- ✓ iOS
- ✓ UWP
- ✓ macOS (Preview)

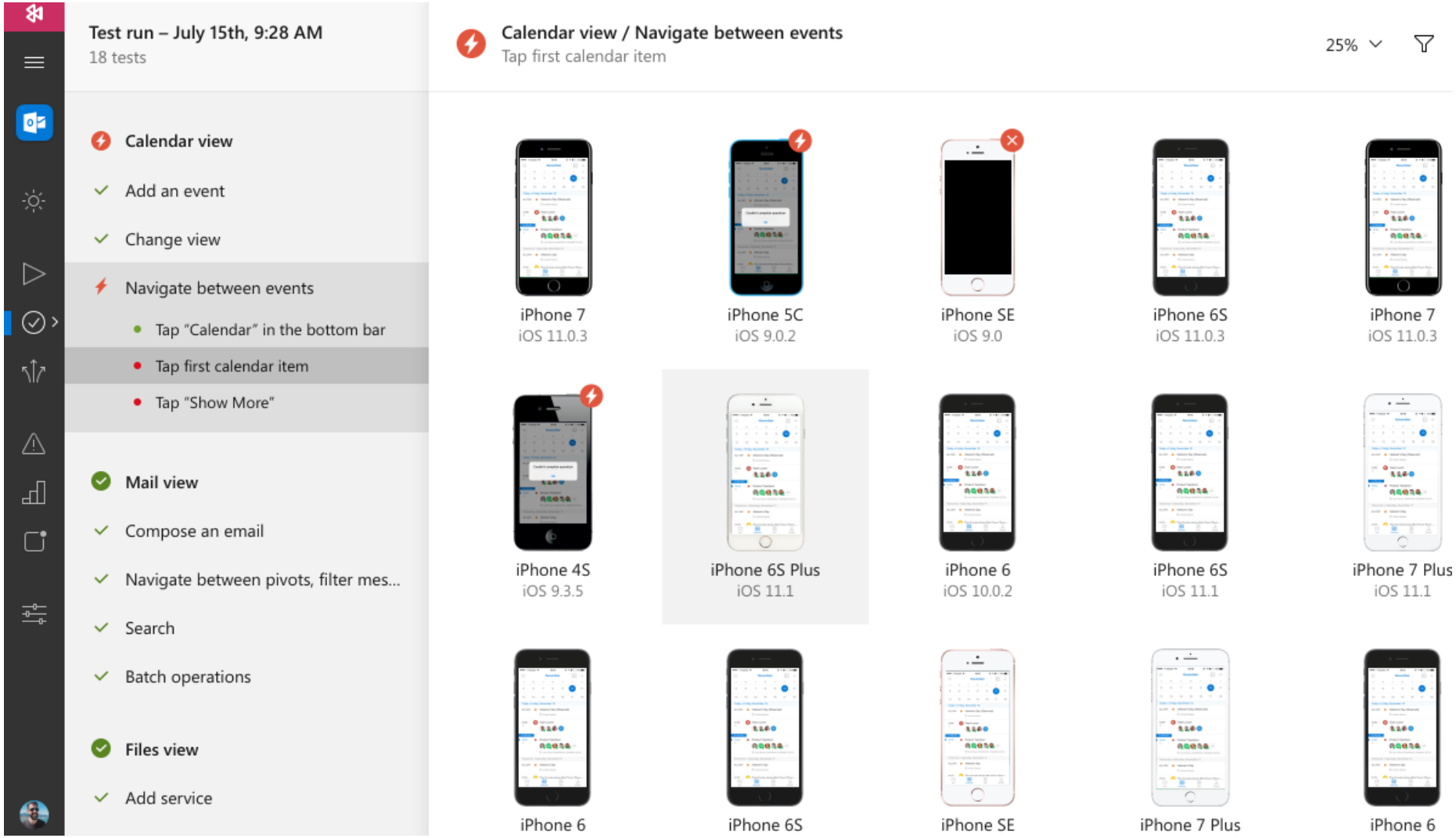
The Frameworks you love

- ✓ Xamarin
- ✓ Java
- ✓ Obj-C/Swift
- ✓ React Native



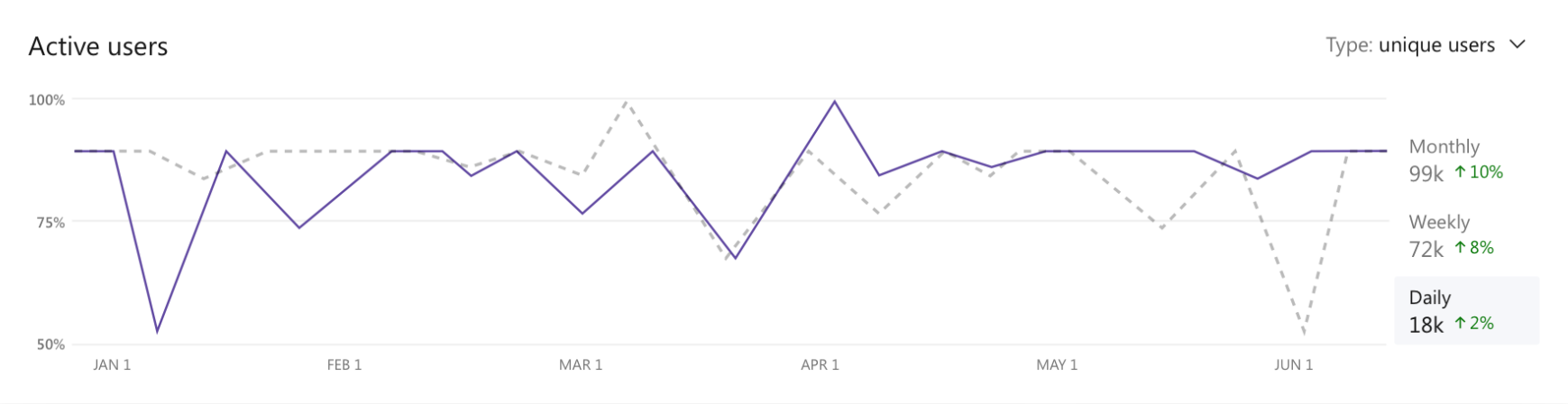
What you can do with App Center?

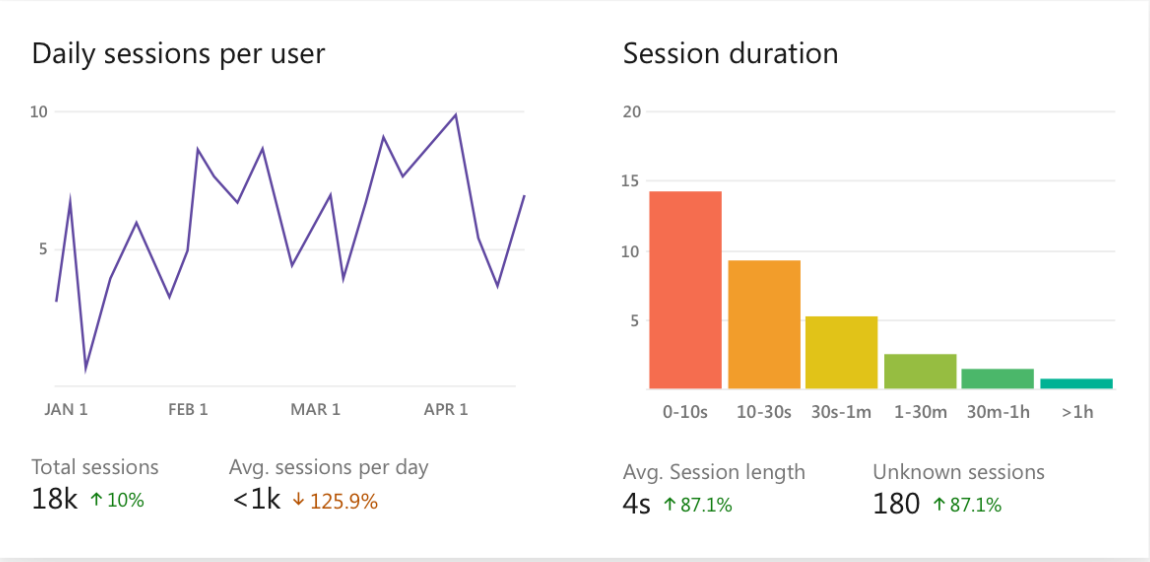
Test the App on Real Devices

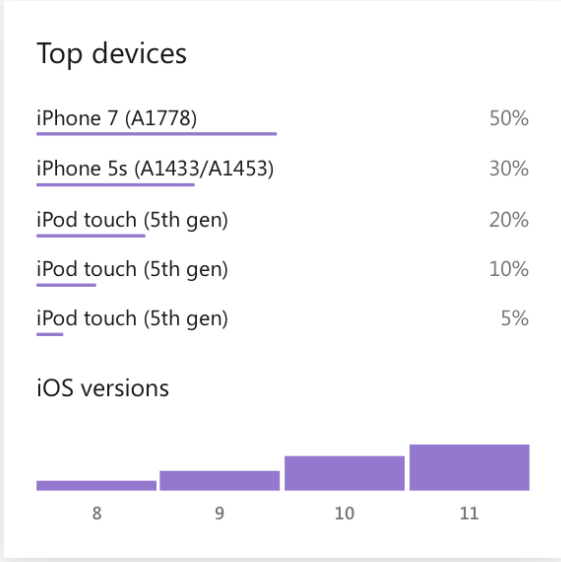


DotNetTricks

App analytics, made for developers







DotNetTricks

Coding is Rhyme

DotNetTricks