

Visual Studio **LIVE!** | San Diego
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Flying High with Xamarin.Forms

Sam Basu | @samidip
Developer Advocate
Progress

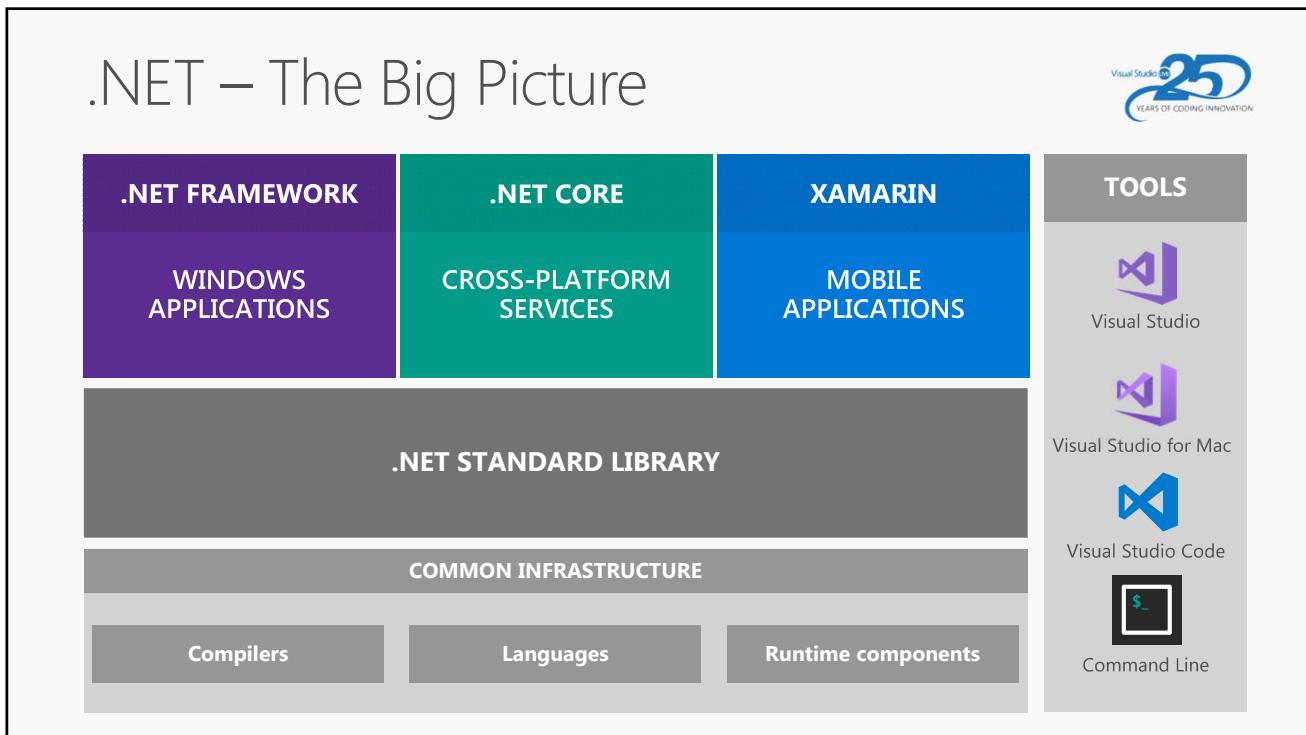
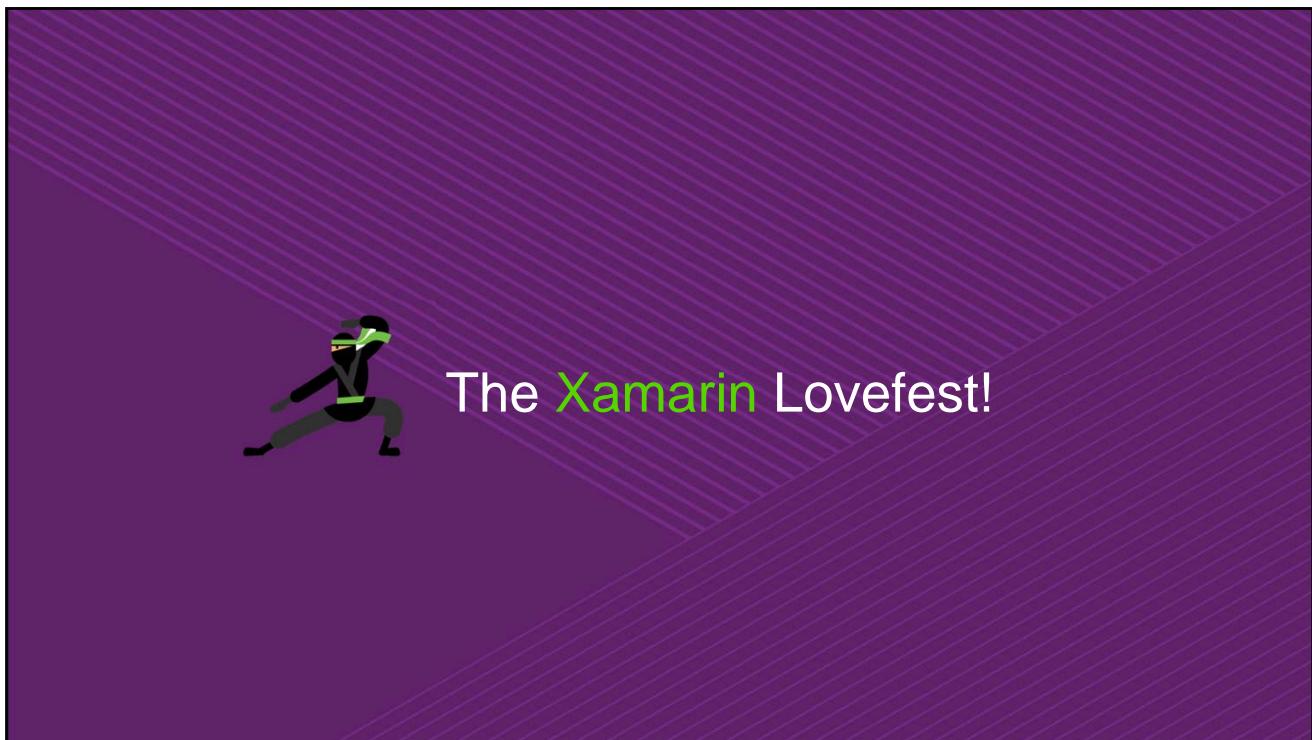
Code Again for the First Time!

Visual Studio 25 YEARS OF CODING INNOVATION

Mobile Your Way

It's 2018 | Developers have choices

1	Mobile Web / PWA
2	Native
3	Hybrid
4	JS Native
5	X-Compiled

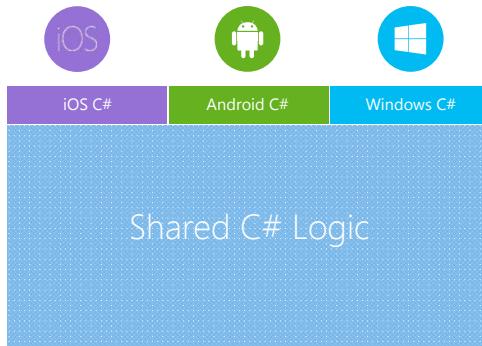


Xamarin Democratizes .NET Mobile Development Truly Native | Truly Cross-Platform

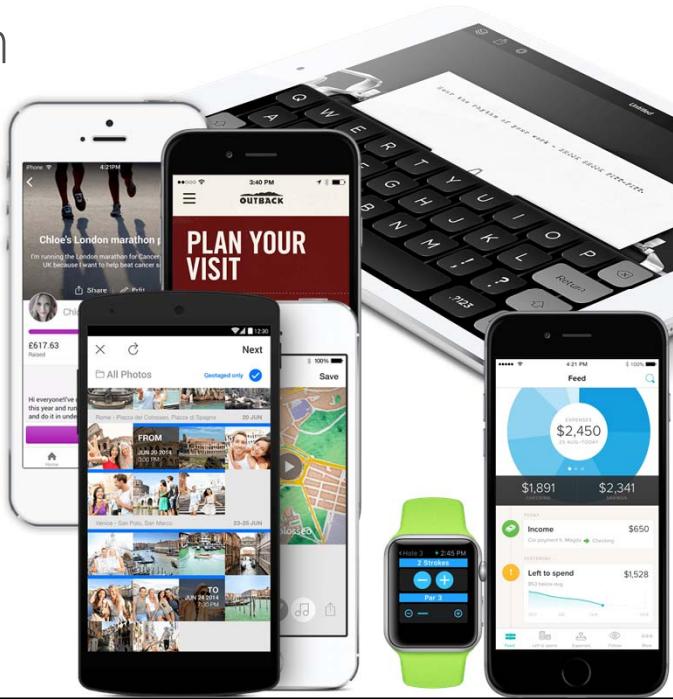


Xamarin Technology Stack

Traditional UI approach



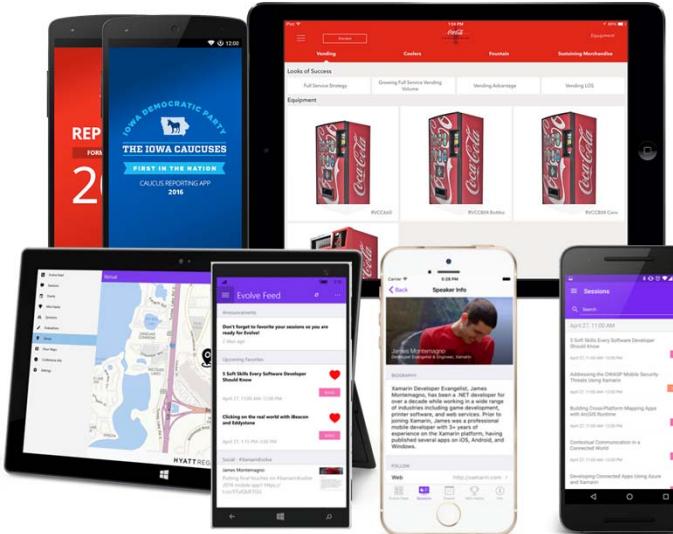
3 Native User Interfaces
Shared App Logic



Xamarin.Forms approach



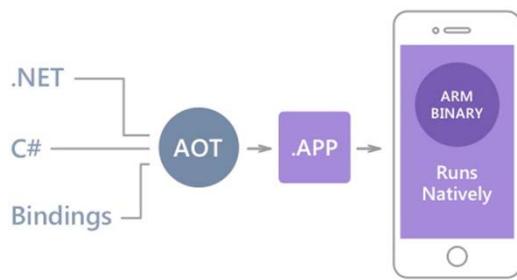
Shared User Interface
Shared App Logic



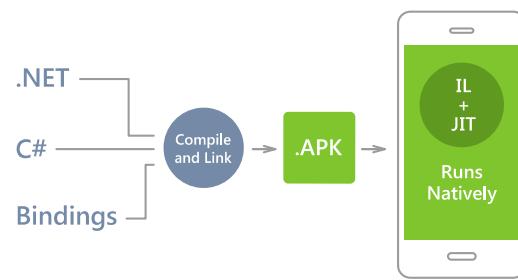
Xamarin.Forms Going Places



Native Performance



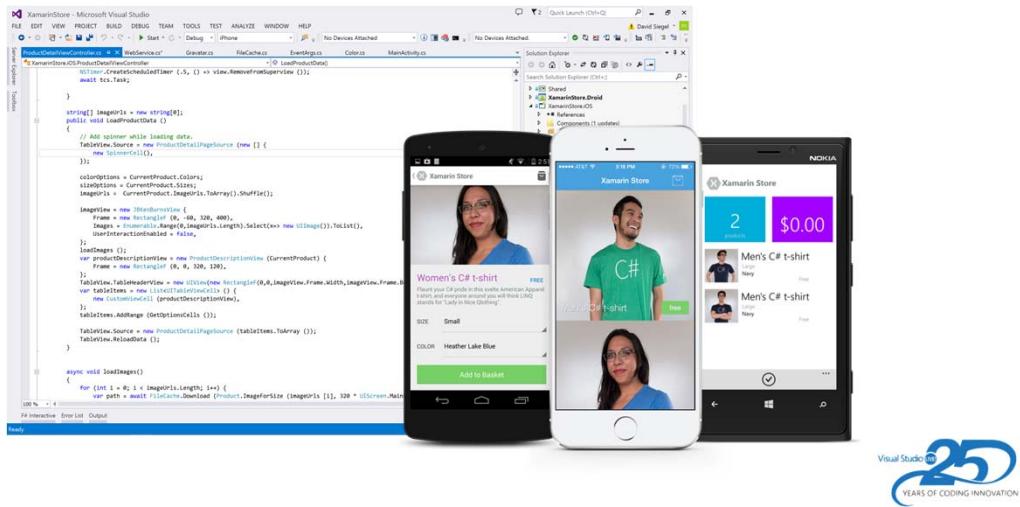
Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store.



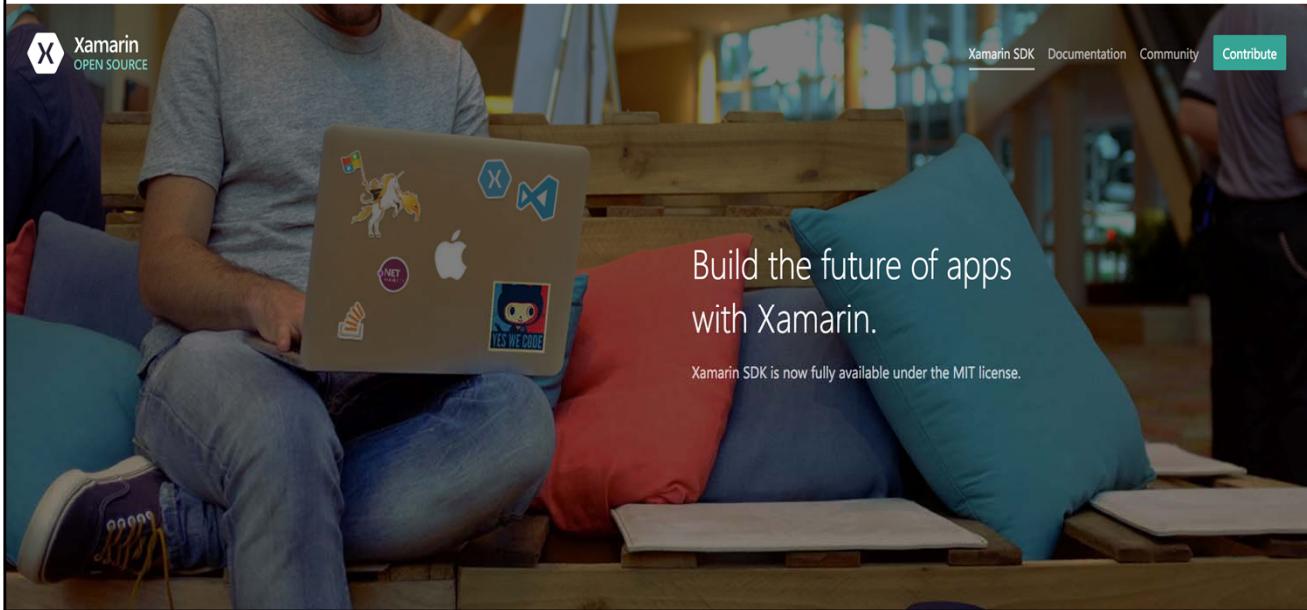
Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device.

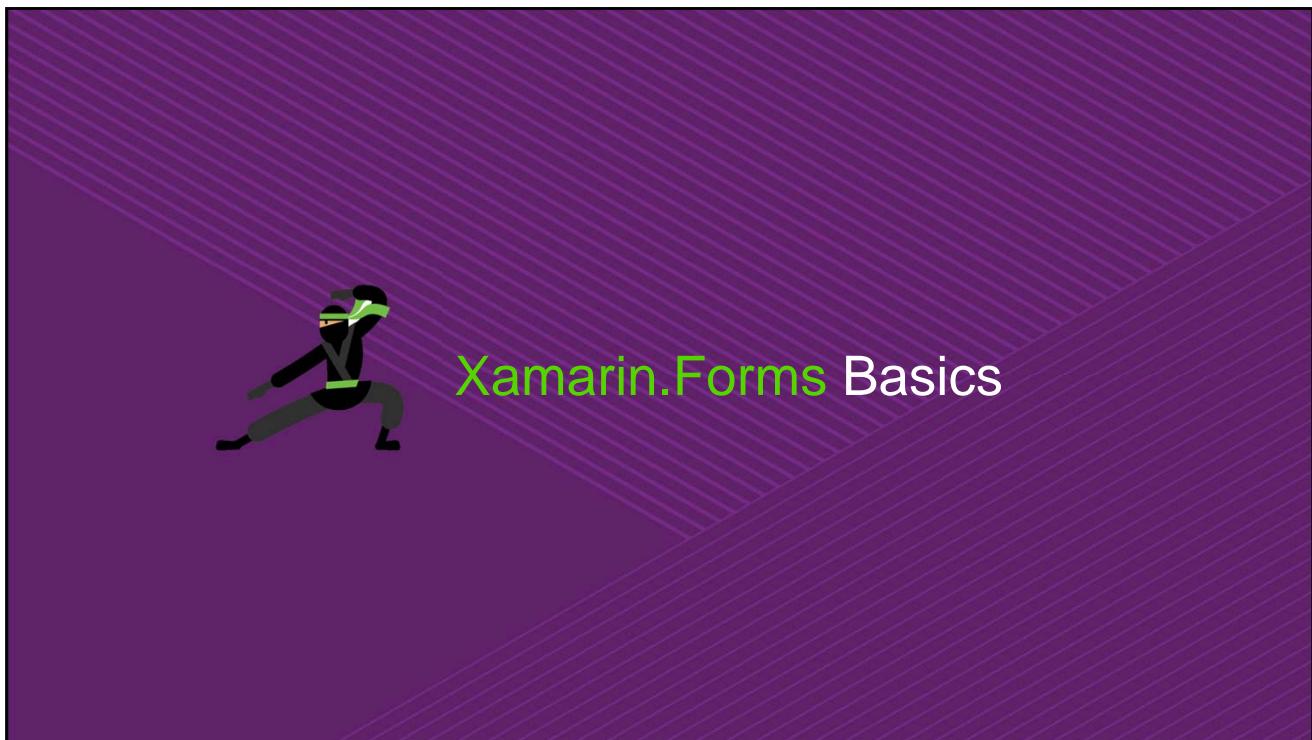


Anything you can do in Objective-C, Swift, or Java
can be done in C# with Xamarin.

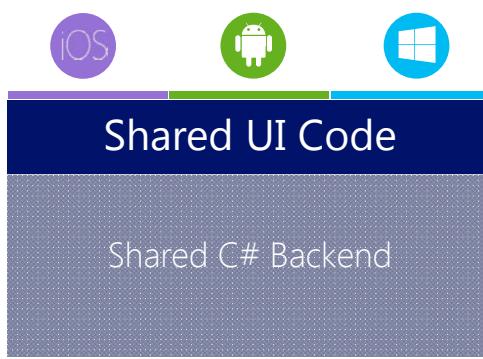


Open Source – open.xamarin.com





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





Windows	Xamarin.Forms
StackPanel	StackLayout
TextBox	Entry
ListBox	ListView
CheckBox	Switch
ProgressBar	ActivityIndicator
Grid	Grid
Label	Label
Button	Button
Image	Image
Date/TimePicker	Date/TimePicker

Know XAML?



Windows	Xamarin.Forms
DataContext	BindingContext
{Binding Property}	{Binding Property}
ItemsSource	ItemsSource
ItemTemplate	ItemTemplate
DataTemplate	DataTemplate

Easy Data Binding

```
<Label Text="{Binding Color.R,
    Converter={StaticResource intConverter},
    ConverterParameter=255,
    StringFormat='R={0:X2}' }" />
```

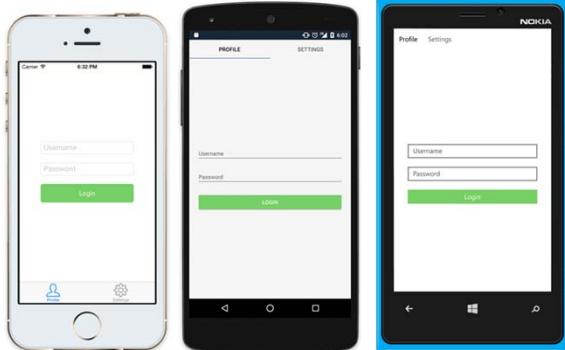


```
<ContentPage.Padding>
    <OnPlatform x:TypeArguments="Thickness">
        <OnPlatform.iOS>
            0, 20, 0, 0
        </OnPlatform.iOS>
        <OnPlatform.Android>
            0, 0, 0, 0
        </OnPlatform.Android>
        <OnPlatform.WinPhone>
            0, 0, 0, 0
        </OnPlatform.WinPhone>
    </OnPlatform>
</ContentPage.Padding>
```

Platform Customization

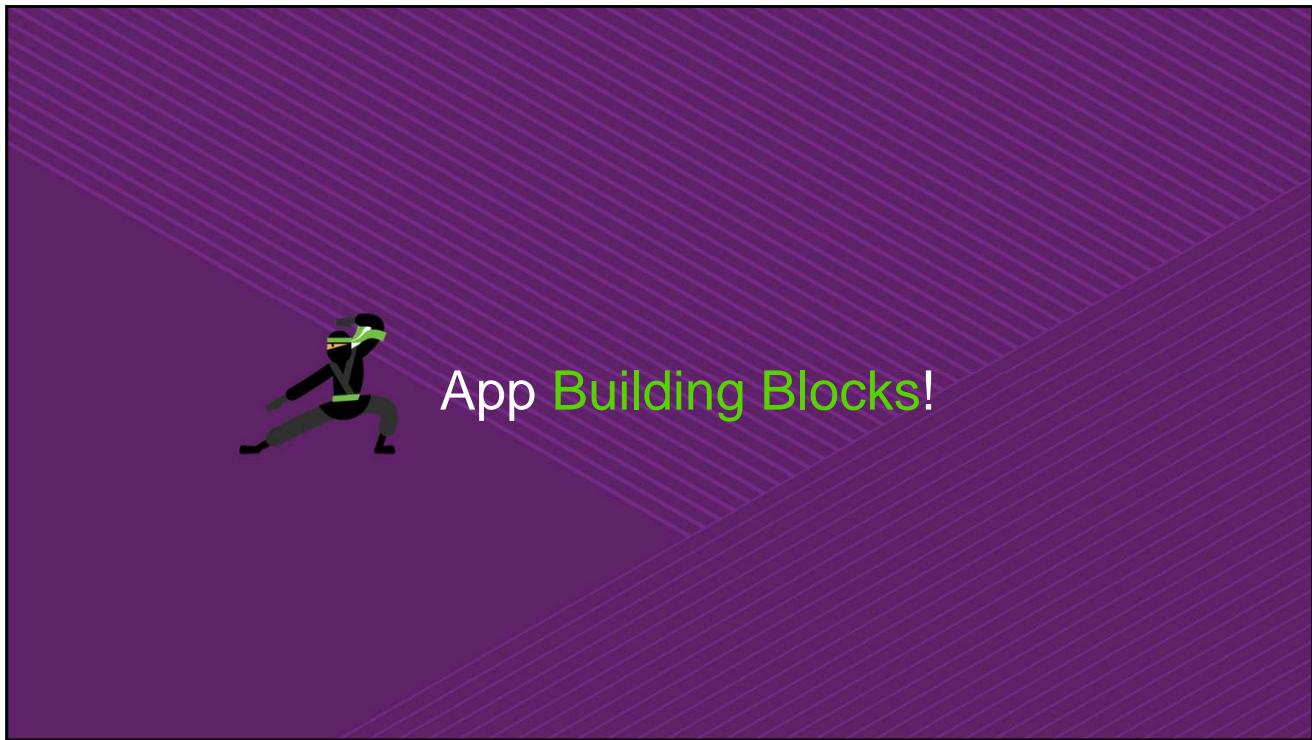


Native UI from shared code



```
<?xml version="1.0" encoding="UTF-8"?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="MyApp.MainPage">
    <TabbedPage.Children>
        <ContentPage Title="Profile" Icon="Profile.png">
            <StackLayout Spacing="20" Padding="20"
                         VerticalOptions="Center">
                <Entry Placeholder="Username"
                      Text="{Binding Username}"/>
                <Entry Placeholder="Password"
                      Text="{Binding Password}"
                      IsPassword="true"/>
                <Button Text="Login" TextColor="White"
                       BackgroundColor="#7D0065"
                       Command="{Binding LoginCommand}"/>
            </StackLayout>
        </ContentPage>
        <ContentPage Title="Settings" Icon="Settings.png">
            <!-- Settings -->
        </ContentPage>
    </TabbedPage.Children>
</TabbedPage>
```





A screenshot of the Visual Studio "New Project" dialog showing the "Choose a template for your new project" step. It highlights the "Blank App" template under the "Forms App" category. To the right, two detailed configuration dialogs are shown: "New Cross Platform App - AmazingHelloWorldApp" and "New iOS App - App10". Both dialogs show the selected "Blank App" template and various configuration options like platform selection, UI technology, and device support.

Visual Studio Live! San Diego 2018

The screenshot shows the NuGet website homepage. At the top, there are navigation links for 'nuget', 'Packages', 'Upload', 'Statistics', 'Documentation', 'Downloads', and 'Blog'. On the right, there are 'Sign in' and 'Register' buttons. The main heading is 'Create .NET apps faster with NuGet'. Below it is a search bar with the placeholder 'Search for packages...'. A central graphic features a blue circuit board with three 3D cubes representing package versions, unique packages, and package downloads. Text overlays on the circuit board indicate '991,756 package versions', '4,353,111,256 package downloads', and '92,117 unique packages'. Below the graphic, the heading 'What is NuGet?' is followed by a brief description: 'NuGet is the package manager for .NET. The NuGet client tools provide the ability to produce and consume packages. The NuGet Gallery is the central package repository used by all package authors and consumers.' Three circular icons below the description represent 'Learn', 'Explore', and 'Create'. To the right of these icons is the Visual Studio 25th anniversary logo.

The screenshot shows a GitHub repository page for 'xamarin / XamarinComponents'. The repository has 665 commits, 10 branches, 0 releases, 23 contributors, and follows the MIT license. The commit history lists several changes, including fixes for addin versions, project URLs, and build scripts. Below the commit history is a 'README.md' file containing sections for 'Open Source Components for Xamarin' and 'Xamarin Supported Open Source Components'. To the right of the repository content is a large graphic featuring a central 'X' icon connected to various mobile device icons (iOS, Android, Windows) and other developer-related icons (camera, location, user).

Visual Studio Live! San Diego 2018

Screenshot of the Microsoft Docs page for Xamarin.Essentials.

The page title is "Xamarin.Essentials". It shows a "Pre-release NuGet" badge. The main content area discusses the features of Xamarin.Essentials, mentioning cross-platform APIs for mobile applications. It includes sections for "Get Started with Xamarin.Essentials" and "Feature Guides". A sidebar on the left lists various features like User Interface, Platform Features, and Sensor-related components. A sidebar on the right provides links to Get Started, Feature Guides, Troubleshooting, and API Documentation. A central diagram illustrates the cross-platform nature of the library, showing a central "X" icon connected to three mobile device icons (iOS, Android, Windows) and various application icons (camera, location, clipboard, etc.).

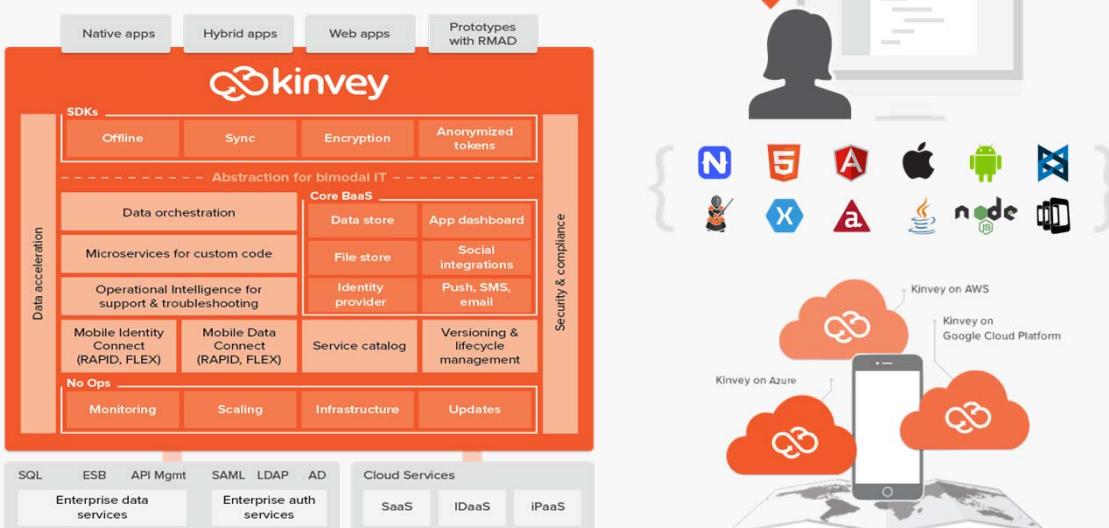
Azure Services for your Mobile Apps

The Azure Services for your Mobile Apps section highlights several key services:

- Web Apps**: Web apps that scale.
- Mobile Apps**: Build mobile apps for any device.
- API Apps**: Easily build and consume APIs.
- Logic Apps**: Automate business processes.
- Functions**: Process events with serverless code.
- SQL Database**: Managed relational SQL Database as a service.
- Storage**: Durable, highly-available, and massively-scalable cloud storage.

At the bottom right, there is a Visual Studio 25th anniversary logo.

Kinvey BaaS



The diagram illustrates the Kinvey Backend as a Service (BaaS) architecture. On the left, a central orange box represents the Kinvey platform, divided into several functional sections:

- SDKs:** Offline, Sync, Encryption, Anonymized tokens.
- Data acceleration:** Data orchestration, Microservices for custom code, Operational Intelligence for support & troubleshooting.
- Core BaaS:** Data store, App dashboard, File store, Social integrations, Identity provider, Push, SMS, email.
- No Ops:** Monitoring, Scaling, Infrastructure, Updates.

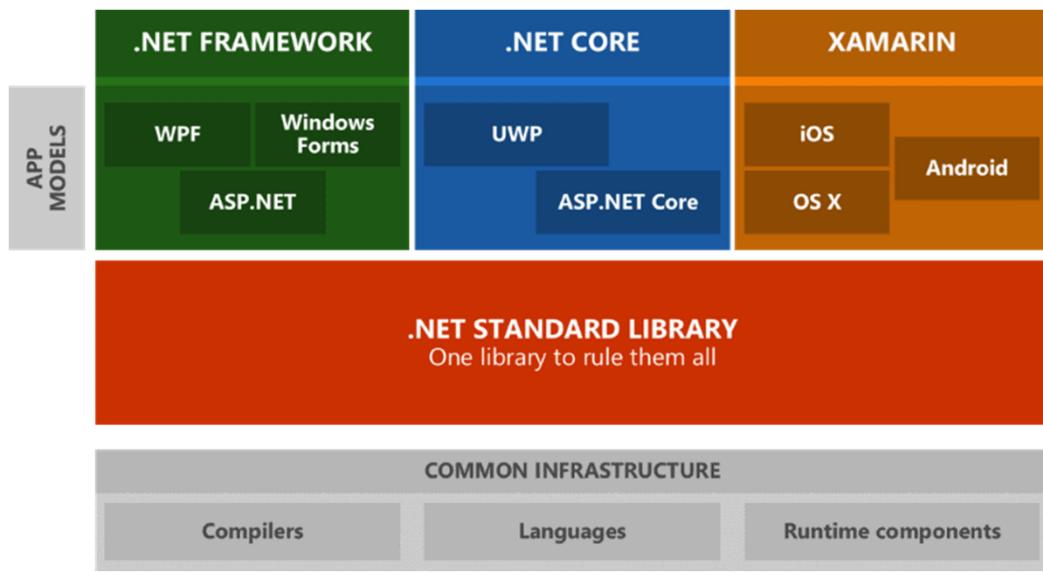
 Below these are integration layers:

- Enterprise data services: SQL, ESB, API Mgmt.
- Enterprise auth services: SAML, LDAP, AD.
- Cloud Services: SaaS, IDaaS, iPaaS.

 At the top, tabs for Native apps, Hybrid apps, Web apps, and Prototypes with RMAD are shown. To the right, a large silhouette of a person is connected to a computer screen displaying code, with three red hearts above it. Below the silhouette is a collection of developer and technology icons, including NPM, ES6, Angular, Apple, Android, Node.js, and others. Three clouds represent Kinvey deployment on AWS, Google Cloud Platform, and Azure, each with a smartphone icon indicating mobile application integration.

Let's see a **Demo**

.NET Standard



XAML Standard

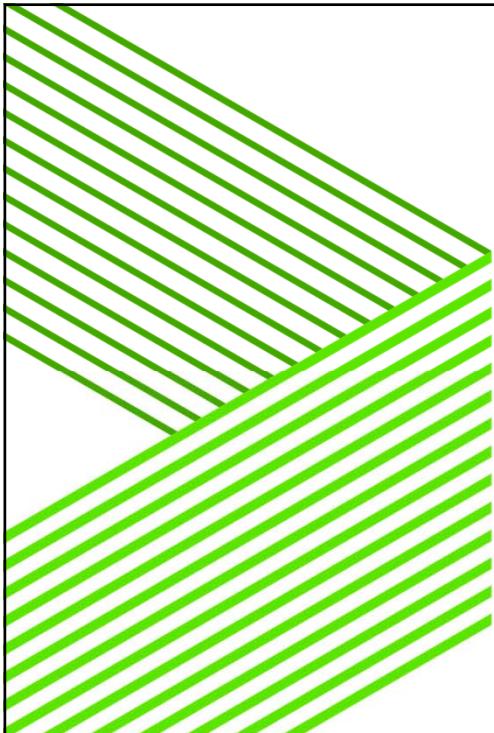
```
<!-- Xamarin.Forms XAML -->
<ContentView>
    <StackLayout Orientation="Horizontal">
        <Label Text="Work Order #"
            VerticalOptions="Center"/>
        <Entry Placeholder="Enter your work order"
            Text="{Binding WONumber, Mode=TwoWay}"/>
        <Button Text="Save"
            TextColor="White"
            BackgroundColor="#77D0E5"
            Command="{Binding SaveCommand}"/>
    </StackLayout>
</ContentView>
```

```
<!-- UWP XAML -->
<UserControl>
    <StackPanel Orientation="Horizontal">
        <TextBlock Text="Work Order #"
            VerticalAlignment="Center" />
        <TextBox PlaceholderText="Enter your work order"
            Text="{Binding WONumber, Mode=TwoWay}" />
        <Button Content="Save"
            Foreground="white"
            Background="#77D0E5"
            Command="{Binding SaveCommand}" />
    </StackPanel>
</UserControl>
```

Unified XAML Dialect • Shared UI across UWP & Xamarin.Forms • OSS Specs



MVVM Help



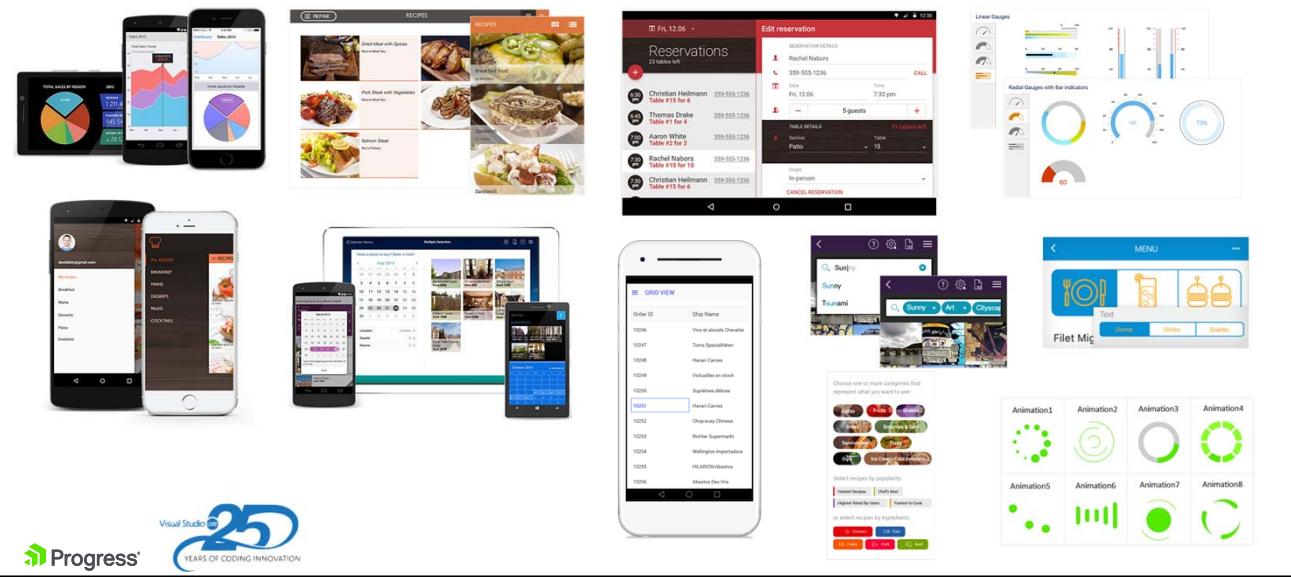
Telerik®
by  Progress

Complete Rich .NET Toolbox
Any App | Any Platform



Telerik UI for Xamarin

Sharp Performant Polished UI .. For all your Xamarin Apps!



Xamarin University

Free self-guided courses

Take free self-guided courses that cover the core Xamarin concepts and count towards certification.

Register at:
university.xamarin.com/self-guided

Get a full subscription for live classes, advanced topics, and to complete certification.

Sign up at:
xamarin.com/university

Visual Studio **LIVE!** | San Diego
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Flying High with Xamarin.Forms

Code Again for the First Time!

Thank You!
Let's keep chatting | @samidip

Visual Studio 25 YEARS OF CODING INNOVATION