



ANDROID, IOS AND HYBRID
APPLICATIONS

Mobile-App Development

- ▶ Marco von Ballmoos
- ▶ marco@encodo.ch
- ▶ Bachelor
- ▶ Encodo Systems AG, Winterthur
- ▶ Mobile Development in Banking, Pharma, User Assistance etc.

QUESTIONS FOR YOU

- ▶ What's your name?
- ▶ Where do you work?
- ▶ What technologies do you have experience with?
- ▶ Do you have any mobile-app experience?
- ▶ What do you want to learn from this course?
- ▶ Which OS and device would you like to use?

GENERAL

- ▶ After 21:00 will be for questions, problems etc.
- ▶ There is one written exam at the end of the semester
- ▶ The app that you create during this course will be graded

GENERAL

- ▶ Documentation and materials

https://github.com/mvonballmo/APE2021_Docs

- ▶ Sample app

https://github.com/mvonballmo/APE2021_App

- ▶ Microsoft documentation

<https://docs.microsoft.com/xamarin/xamarin-forms/>



COURSE AGENDA

DAY 1

- ▶ Overview Mobile Development
- ▶ Xamarin Forms basics
- ▶ Setup Xamarin Sample-App
- ▶ Cross-platform Basics
- ▶ Navigation Basics
- ▶ Document your app/idea and get it approved

DAY 2

- ▶ MVVM
- ▶ XAML for Forms
- ▶ Controls and differences from WPF
- ▶ Bindings
- ▶ Commands

DAY 3

- ▶ Dialogs
- ▶ Styling
- ▶ Inversion of Control (IOC)
- ▶ Testing

DAY 4

- ▶ Notifications
 - ▶ Local
 - ▶ PUSH/Remote

DAY 5

- ▶ Hybrid Applications
- ▶ Interoperability with the native part
 - ▶ Design a possible interface
 - ▶ Present your approach
- ▶ Create a small working sample

DAY 6

- ▶ Security Basics (Block Cyphers, PK-Infrastructure)
- ▶ Mobile Security
- ▶ Biometrics

DAY 7

- ▶ Local Databases (SQLite)
- ▶ Logging & Crashes
- ▶ Continue working on your app

DAY 8

- ▶ Written exam
- ▶ Continue working on your app
 - ▶ Short review with each one

DAY 9

- ▶ Present your apps from the project week



MOBILE DEVELOPMENT

Overview and IDE Setup

OVERVIEW MOBILE DEVELOPMENT

Mobile Development				
	IDE	Languages	Frameworks	Build/Deployment
Android	Android Studio Eclipse NetBeans	Java Kotlin C++	Dagger data-bind Crashlytics Google Play Service Support Library gson jdeferred	PlayStore Gradle Maven
iOS	xCode AppCode	Swift Objective-C	Alamofire CryptoSwift SwiftlyJSON SwiftlyRSA PromiseKit	Testflight AppStore CocoaPods
Cross	Visual Studio JB Rider WebStorm Visual Studio Code	C#/F# TypeScript/JS Dart etc. JavaScript HTML/CSS Less/Sass React/Redux Angular	React Native Ionic PhoneGap (Cordova) NativeScript	fastlane HockeyApp Artifactory Jenkins Teamcity

SETUP

- ▶ Choose an OS and target device
 - ▶ iOS/macOS: Big Sur/Xcode
- ▶ Install Visual Studio Xamarin Tools
- ▶ Set up GitHub account and share your GitHub username
 - ▶ I will add access to the course repositories
- ▶ Set up an emulator (Android) or simulator (iOS)
 - ▶ Optional: [Set up your device to allow debugging](#)
- ▶ [Install SmartGit (<https://www.syntevo.com/smartgit/>)]

QUESTIONS?

- ▶ Short break
- ▶ Everyone send me their GitHub username



INTRO TO GIT AND SETUP

GIT

- ▶ Pull vs Fetch
- ▶ Push vs Commit
- ▶ Local vs Remote
- ▶ Merge vs Rebase
- ▶ Working Tree vs. Stage vs. Stash

- ▶ Demo
- ▶ Clone the ***APE2021_Docs*** & ***APE2021_App*** repositories
- ▶ ***mvonballmo*** is my account



XAMARIN & CROSS-PLATFORM

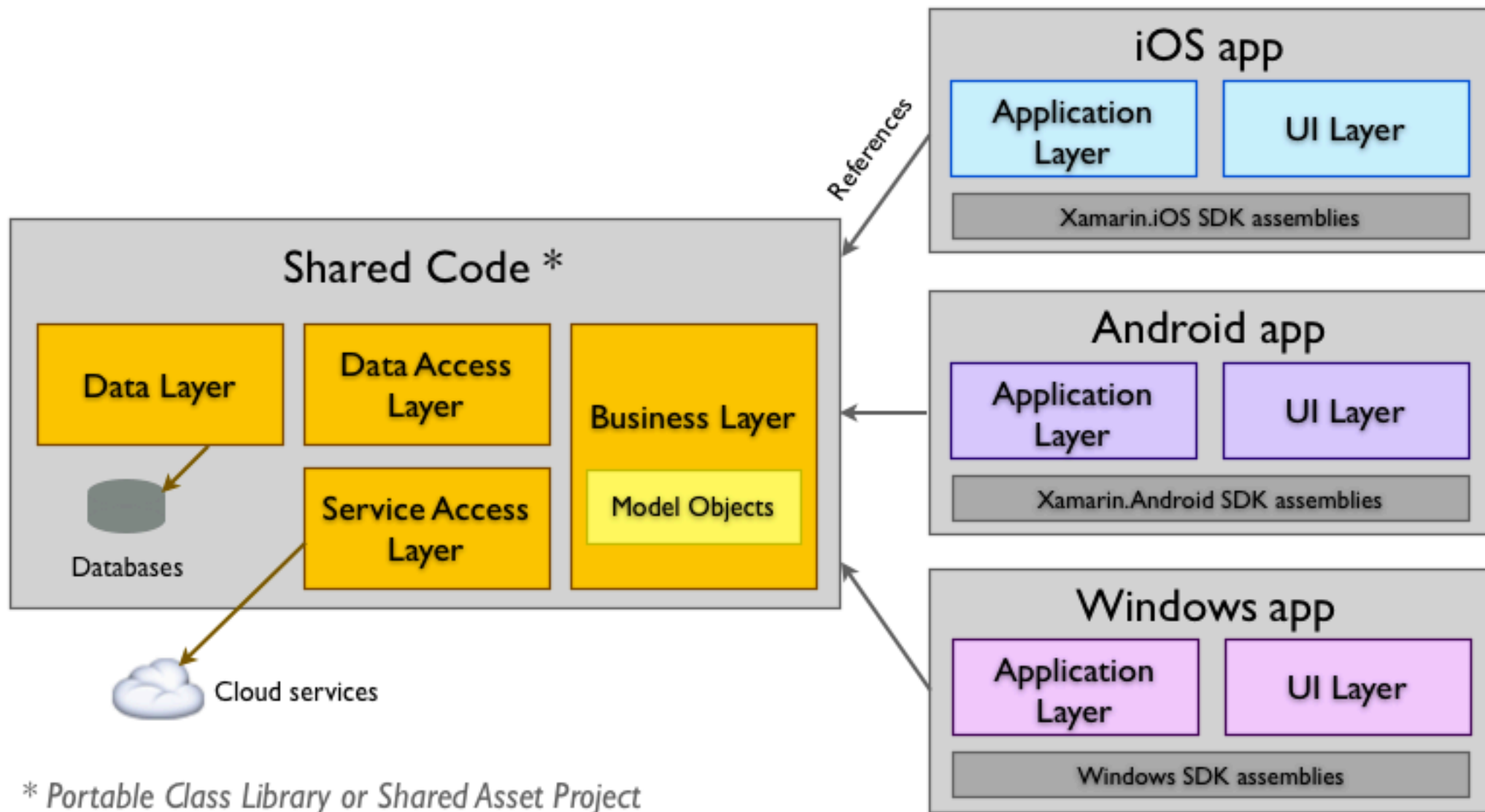
OVERVIEW

- ▶ Xamarin was created by Mono
- ▶ Enables Android and iOS development with .NET/C#
- ▶ Acquired and integrated by Microsoft in 2016

OVERVIEW – XAMARIN “CLASSIC”

- ▶ Xamarin.iOS & Xamarin.Android (Classic)
 - ▶ They basically map the platform 1:1
 - ▶ Support new APIs in 24 hours
 - ▶ Share the “Core” libraries
 - ▶ No shared UI - unless you use a Hybrid approach

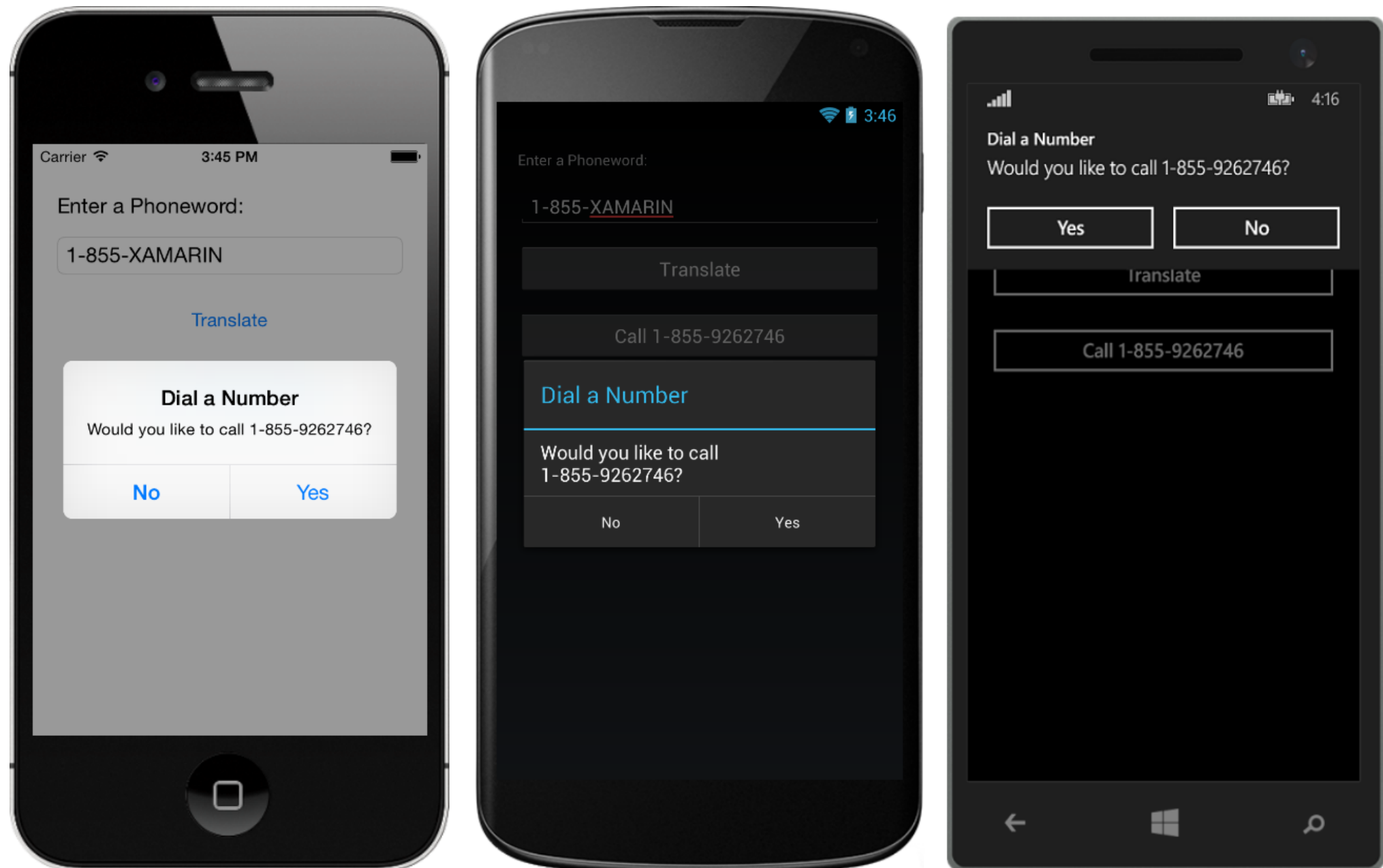
OVERVIEW – XAMARIN CLASSIC



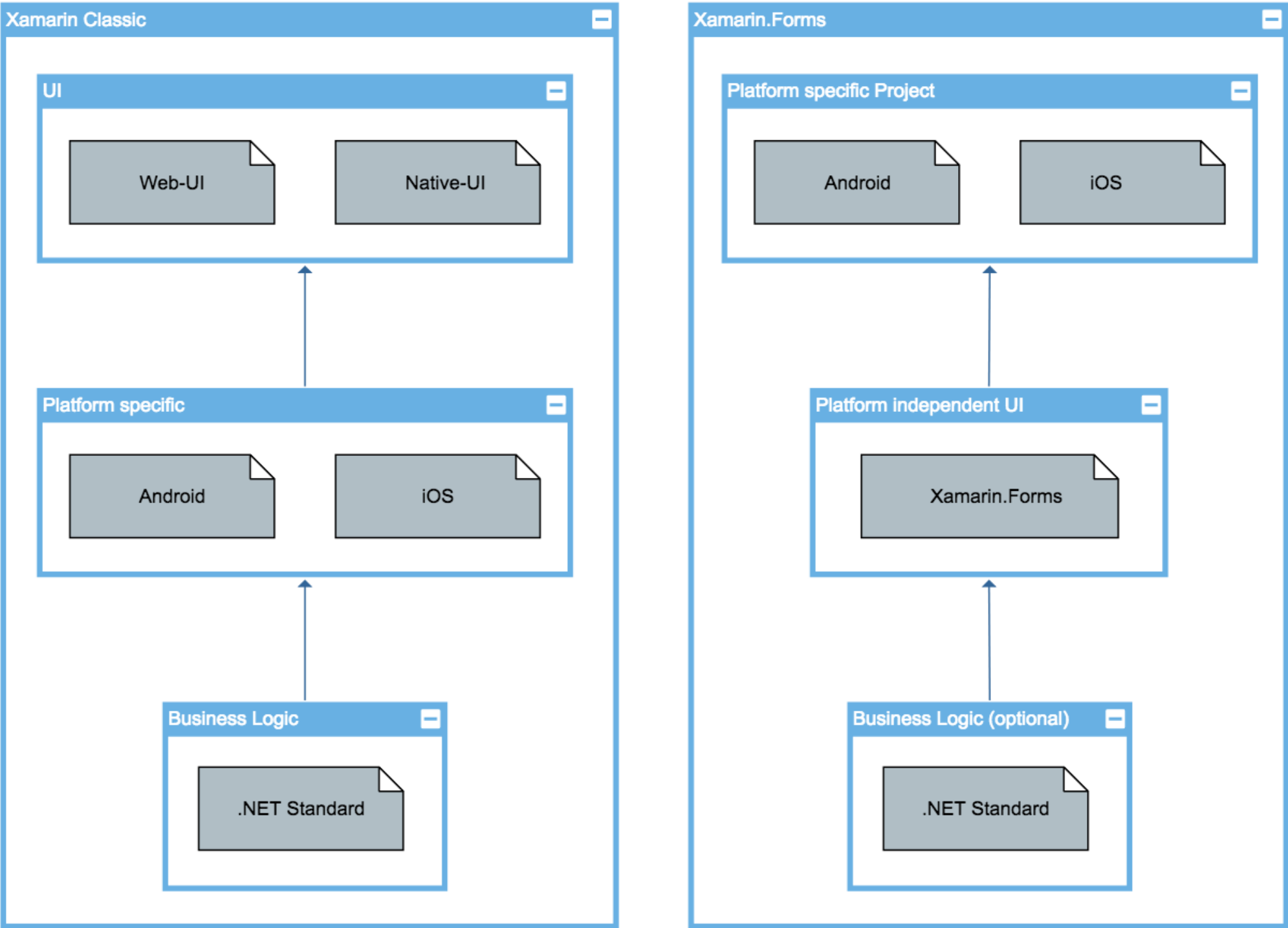
OVERVIEW – XAMARIN FORMS

- ▶ Xamarin.Forms
 - ▶ Share UI code and write it in XAML
 - ▶ Use DependencyService (IoC) to access platform features
 - ▶ Abstraction of features (Dialogs, Notifications, etc.)
 - ▶ Built-in support for navigation

OVERVIEW – XAMARIN FORMS



OVERVIEW - ARCHITECTURE



.NET STANDARD

- ▶ Common API (independent of target runtime)
- ▶ Version 2.1 is the latest
 - ▶ The higher the version the more APIs are available
- ▶ Think of it as an "interface" it does not contain actual code
 - ▶ Mono implements .NET Standard for iOS/Android
 - ▶ .NET Framework implements it for Windows
- ▶ You can find the definitions on [GitHub](https://github.com/dotnet/standard/blob/master/docs/versions/netstandard2.1.md)
- ▶ .NET 5 is the current LTS target

.NET STANDARD

- ▶ Some popular APIs:
 - ▶ File (System.IO)
 - ▶ Collections & LINQ
 - ▶ Task & async await
 - ▶ Http (Client) (System.NET)

CROSS PLATFORM

QUESTIONS?



SAMPLE APP

Create & Run

CREATE A PROJECT

- ▶ Pick Xamarin.Forms (with .NET Standard)
- ▶ You'll see three projects (Shared, iOS, Android)
- ▶ Shared one is a .NET Standard project
- ▶ Walkthrough
- ▶ Everyone get it running!

XAMARIN FORMS: NAVIGATION

► Push other pages onto the stack

```
// App.xaml.cs  
MainPage = new MainPage();
```

```
// ItemsPage.xaml.cs  
await Navigation.PushAsync(new ItemDetailPage(new ItemDetailViewModel(item)));
```

```
// NewItemPage.xaml.cs  
await Navigation.PopModalAsync();
```

NAVIGATION

- ▶ Example
- ▶ See source code in generated sample app

XAMARIN FORMS TABS

- ▶ See MainPage.xaml.cs
- ▶ Populate pages as tabs in the "TabbedPage.Children"

```
<TabbedPage.Children>
  <NavigationPage Title="Browse">
    <x:Arguments>
      <views:ItemsPage />
    </x:Arguments>
    <NavigationPage.Icon>
      <OnPlatform x:TypeArguments="FileImageSource">
        <On Platform="iOS" Value="tab_feed.png"/>
      </OnPlatform>
    </NavigationPage.Icon>
  </NavigationPage>

  <NavigationPage Title="About">
    <x:Arguments>
      <views:AboutPage />
    </x:Arguments>
    <NavigationPage.Icon>
      <OnPlatform x:TypeArguments="FileImageSource">
        <On Platform="iOS" Value="tab_about.png"/>
      </OnPlatform>
    </NavigationPage.Icon>
  </NavigationPage>
</TabbedPage.Children>
```

THINK ABOUT A PROJECT

- ▶ It should include
 - ▶ Notifications
 - ▶ Some login scenario
 - ▶ Alerts (Yes/No Dialog)
 - ▶ Some input fields
 - ▶ A list or something similar
- ▶ See the TODO-App for an example

SETUP THE APPLICATION LAYERS

- ▶ Set up the necessary navigation pages
- ▶ Push to GitHub
- ▶ Invite ***mvonballmo*** to your project
- ▶ Test the navigation

ADDITIONAL TASKS

- ▶ Check out the Master-Detail pattern

<https://docs.microsoft.com/xamarin/xamarin-forms/app-fundamentals/navigation/master-detail-page>

- ▶ Check out the Carousel

<https://docs.microsoft.com/xamarin/xamarin-forms/app-fundamentals/navigation/carousel-page>