A Lap Around the .NET Ecosystem

Barry Luijbregts
SOFTWARE ARCHITECT & DEVELOPER

@AzureBarry www.azurebarry.com

Find me on Pluralsight

- Introduction to Azure App Services
- Building a Global App with Azure PaaS
- Continuous Integration and Continuous Delivery: The Big Picture
- Cloud Design Patterns for Azure: Design and Implementation
- Cloud Design Patterns for Azure: Availability and Resilience
- Cloud Design Patterns for Azure: Data Management and Performance
- The .NET Ecosystem: The Big Picture
- Microsoft Azure for Developers: What to Use When?
- Microsoft Azure Cognitive Services: The Big Picture

Find me on Pluralsight

- Introduction to Azure App Services
- Building a Global App with Azure PaaS
- Continuous Integration and Continuous Delivery: The Big Picture
- Cloud Design Patterns for Azure: Design and Implementation
- Cloud Design Patterns for Azure: Availability and Resilience
- Cloud Design Patterns for Azure: Data Management and Performance

- **The .NET Ecosystem: The Big Picture**

- Microsoft Azure for Developers: What to Use When?
- Microsoft Azure Cognitive Services: The Big Picture

Survey!

.NET Framework?

.NET Core?

Mono for Xamarin?


.NET Standard?

Win- or Web Forms?

The .NET Ecosystem is Confusing

Portable Class Libraries

.NET Framework

.NET

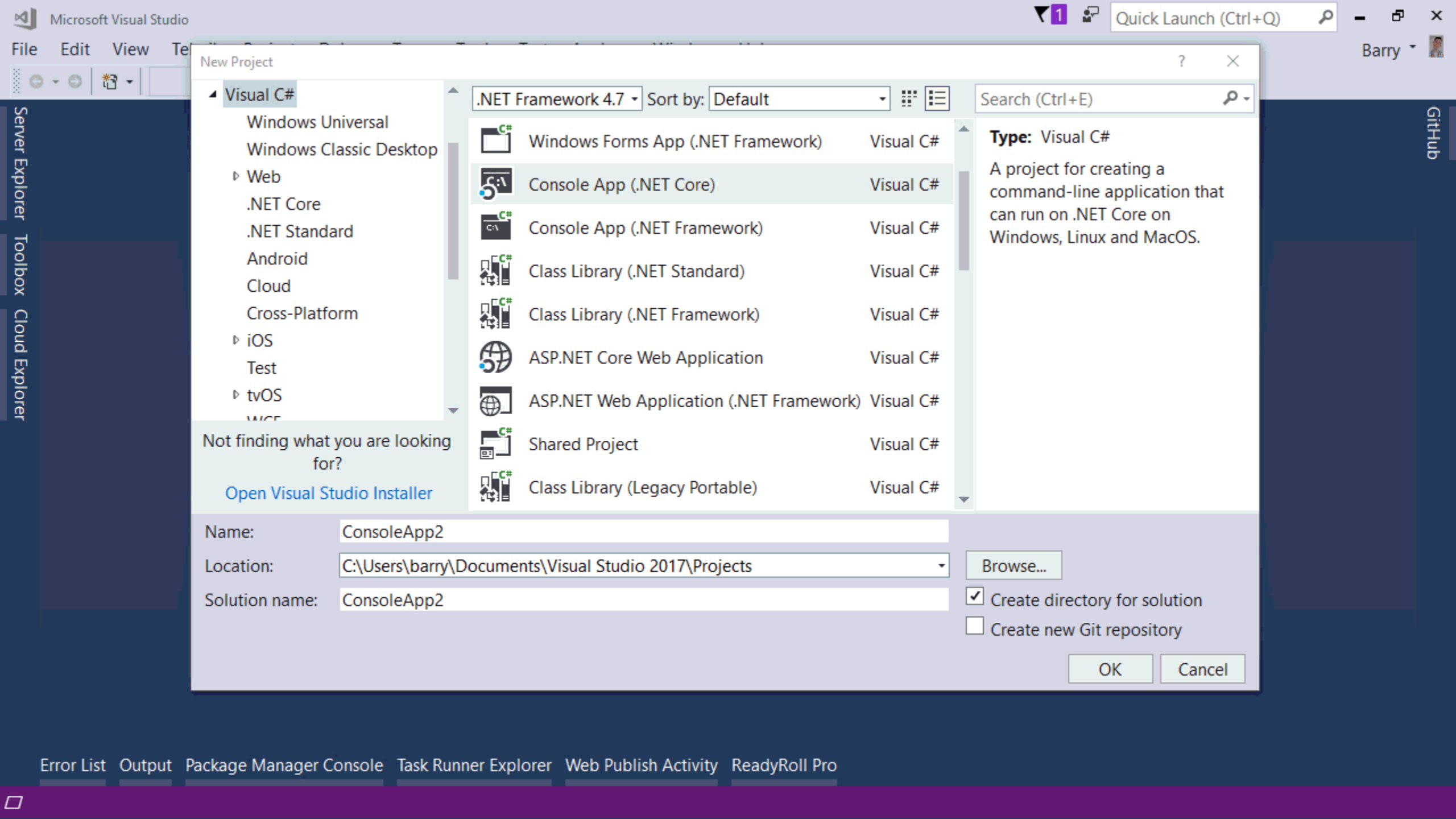
.NET Core

Roslyn Compiler

February 13, 2002

.NET Standard

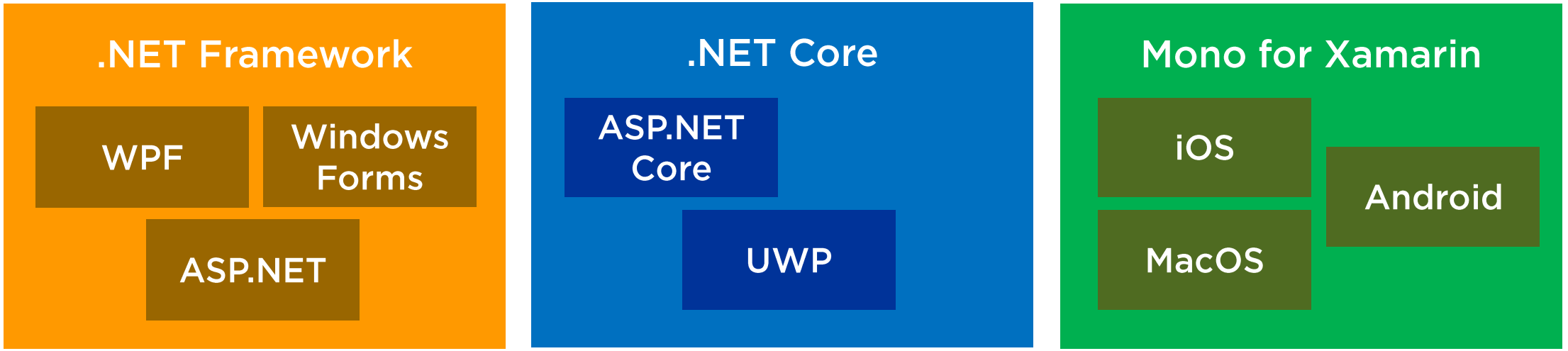
Base Class Library


Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View
Barry
Server Explorer
Toolbox
Cloud Explorer
GitHub
New Project
Visual C#
Windows Universal
Windows Classic Desktop
Web
.NET Core
.NET Standard
Android
Cloud
Cross-Platform
iOS
Test
tvOS
.NET Framework 4.7
Sort by: Default
Search (Ctrl+E)
Windows Forms App (.NET Framework) Visual C#
Console App (.NET Core) Visual C#
Console App (.NET Framework) Visual C#
Class Library (.NET Standard) Visual C#
Class Library (.NET Framework) Visual C#
ASP.NET Core Web Application Visual C#
ASP.NET Web Application (.NET Framework) Visual C#
Shared Project Visual C#
Class Library (Legacy Portable) Visual C#
Type: Visual C#
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
Not finding what you are looking for?
Open Visual Studio Installer
Name: ConsoleApp2
Location: C:\Users\barry\Documents\Visual Studio 2017\Projects
Browse...
Solution name: ConsoleApp2
Create directory for solution
Create new Git repository
OK
Cancel
Error List
Output
Package Manager Console
Task Runner Explorer
Web Publish Activity
ReadyRoll Pro


Dr. Phil

An Overview of the .NET Ecosystem

The .NET Ecosystem

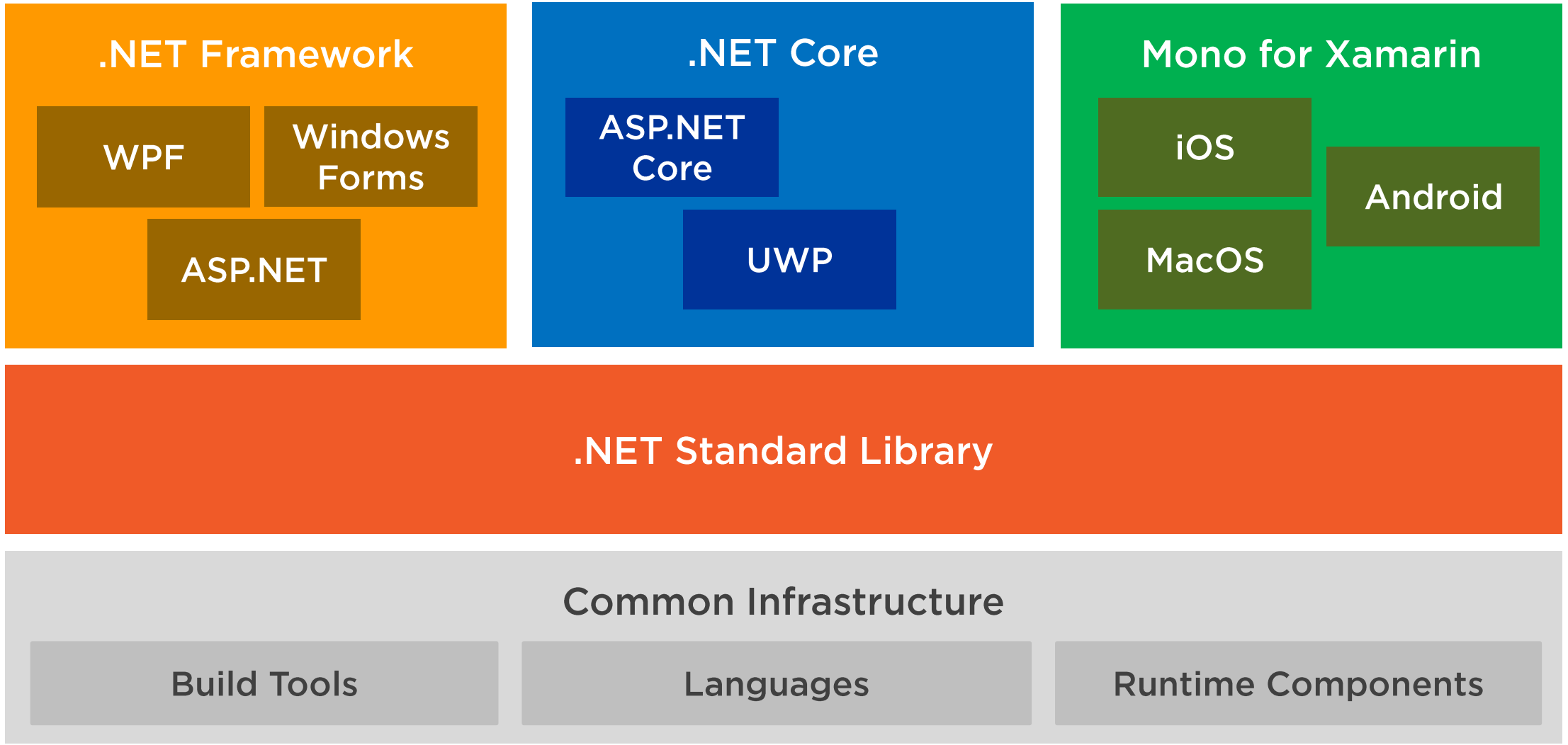


- Windows centric
- Windows specific APIs

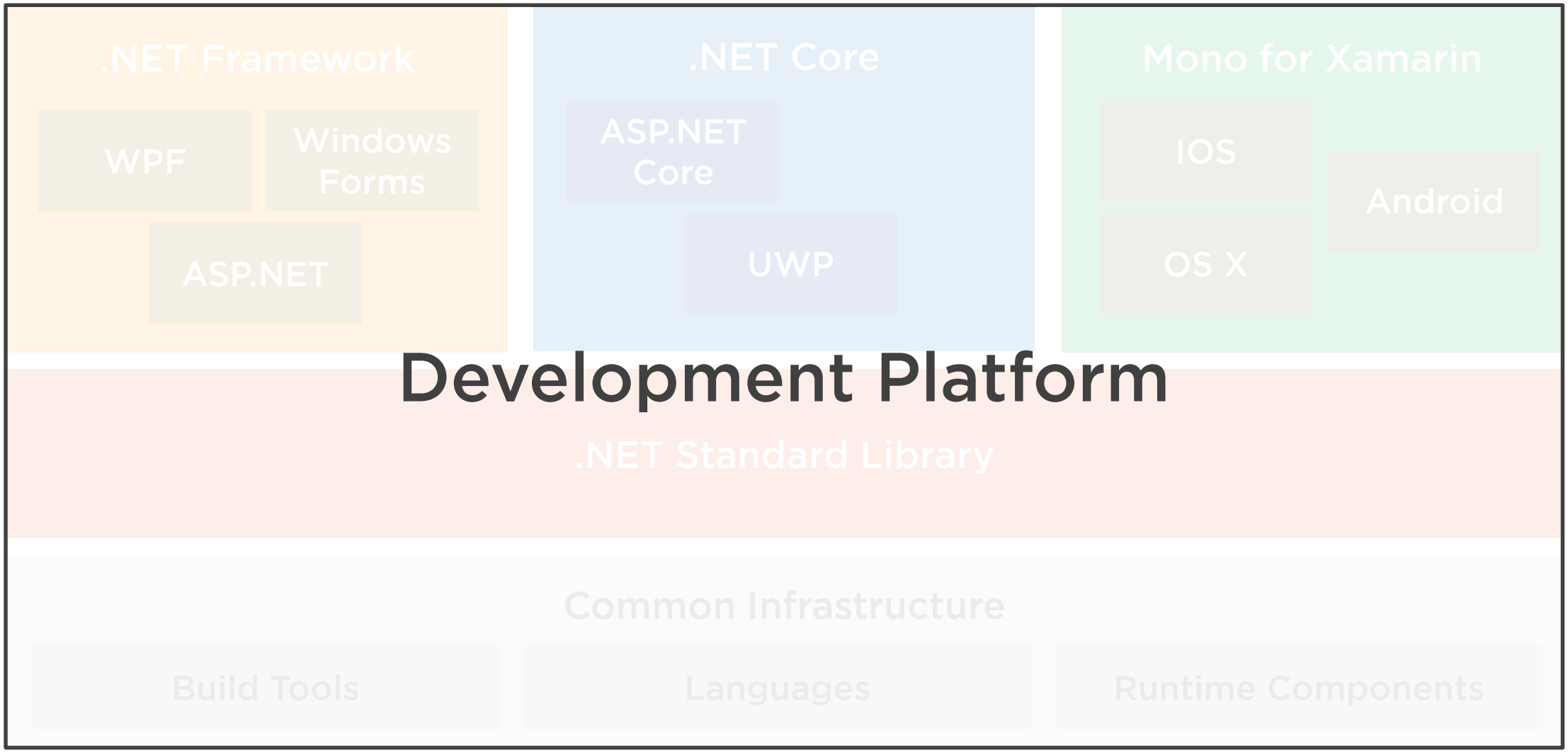
- Cross-platform
- Run side-by-side
- Performance

- Cross-platform
- Specific APIs for
  - iOS
  - Android
  - Xamarin.Mac

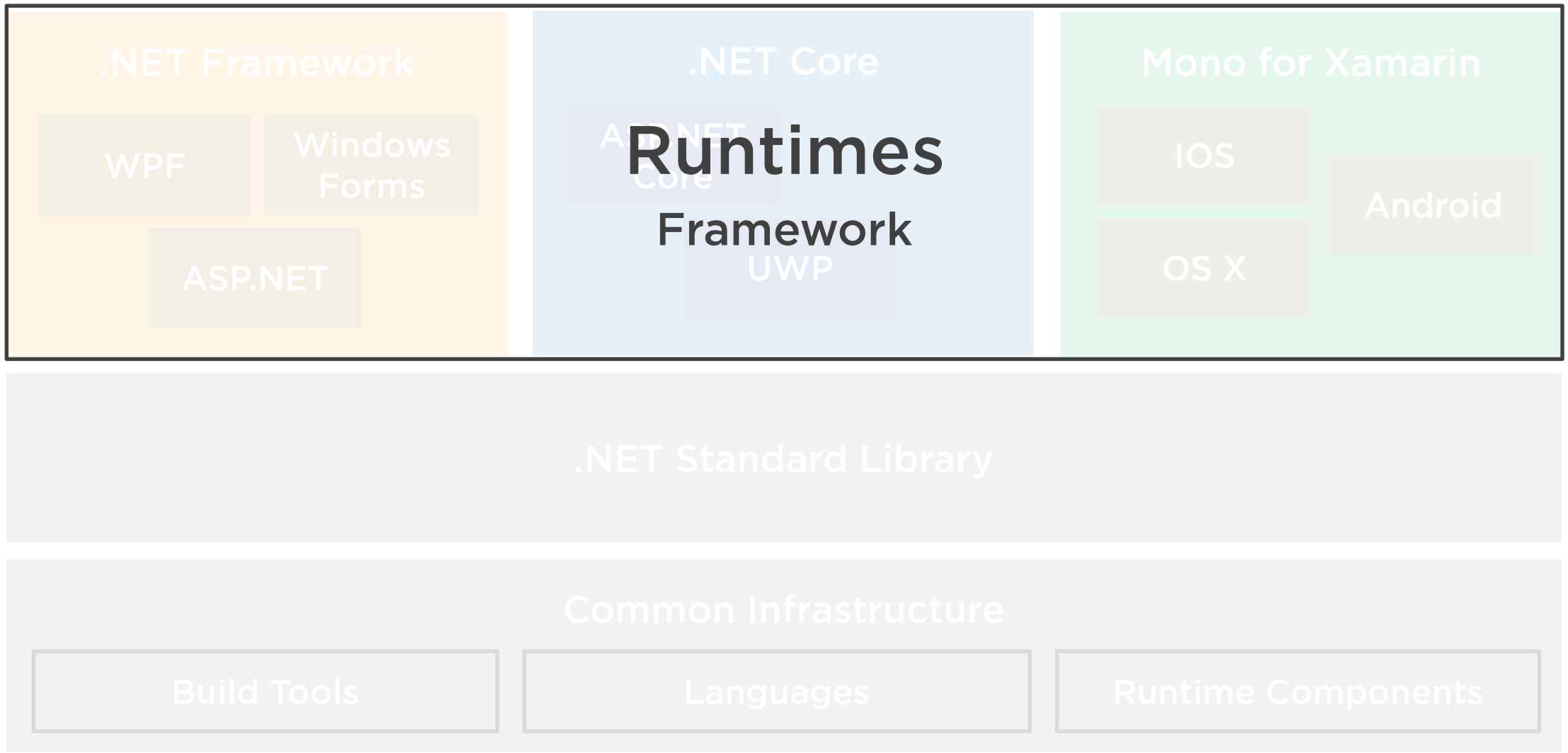
The .NET Ecosystem



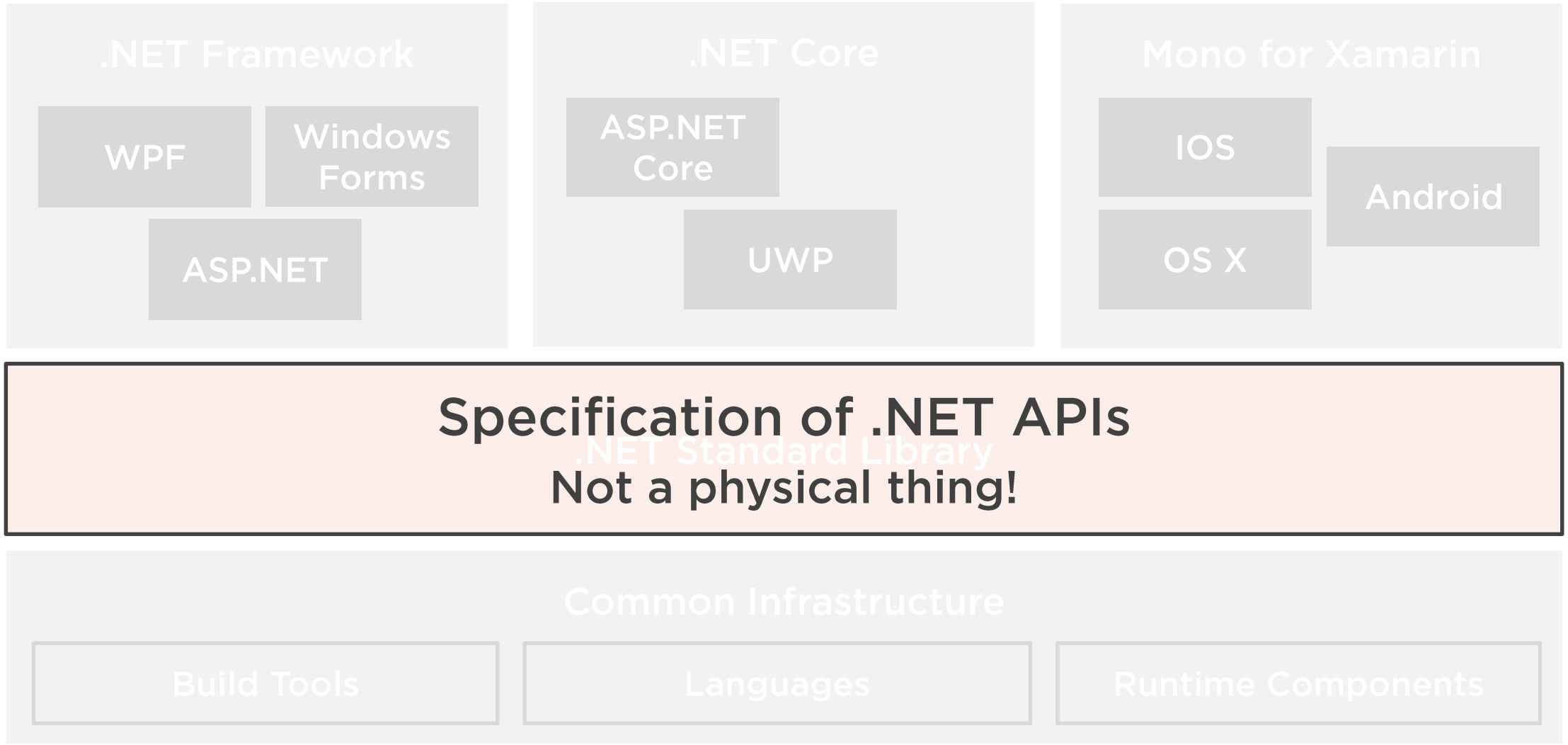
The .NET Ecosystem



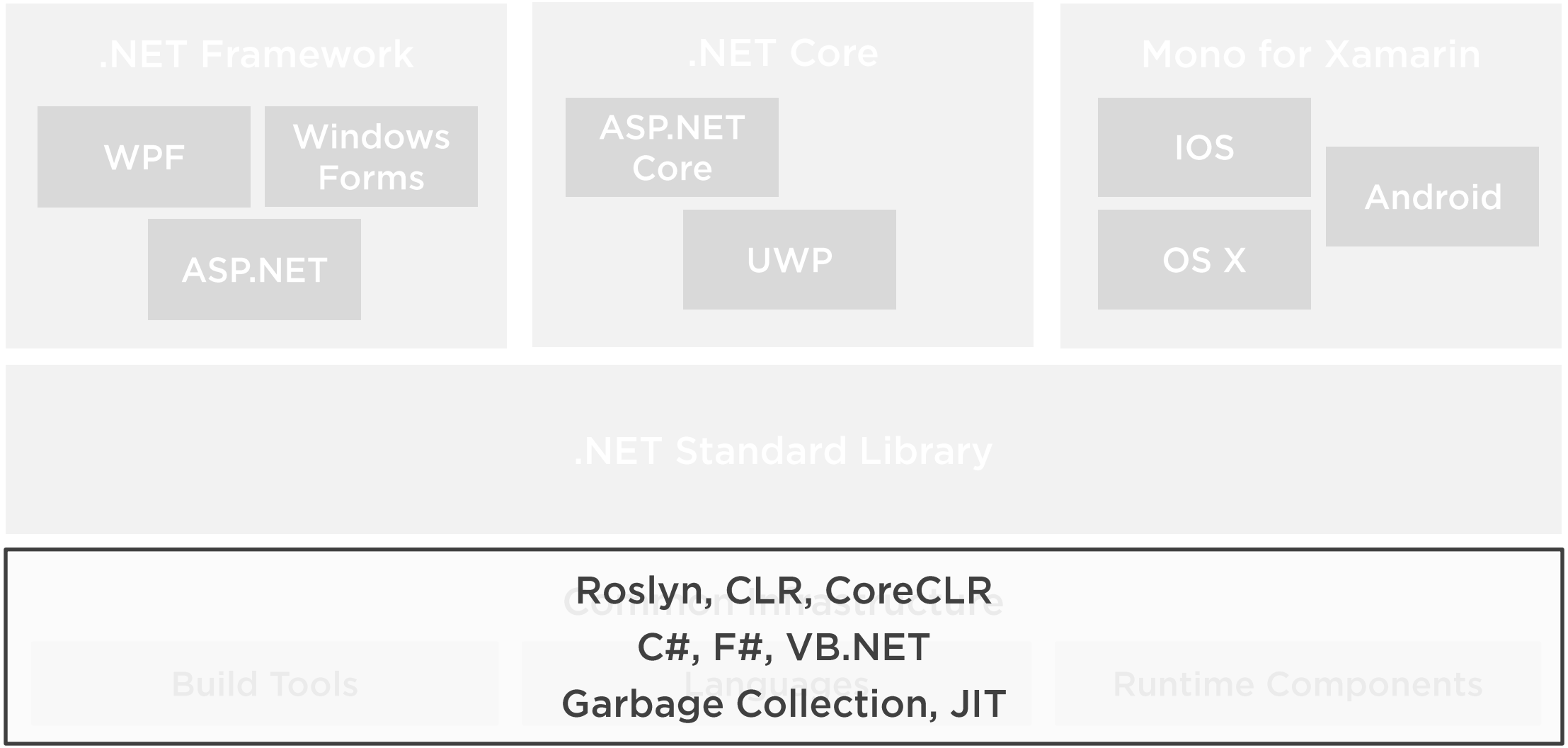
The .NET Ecosystem



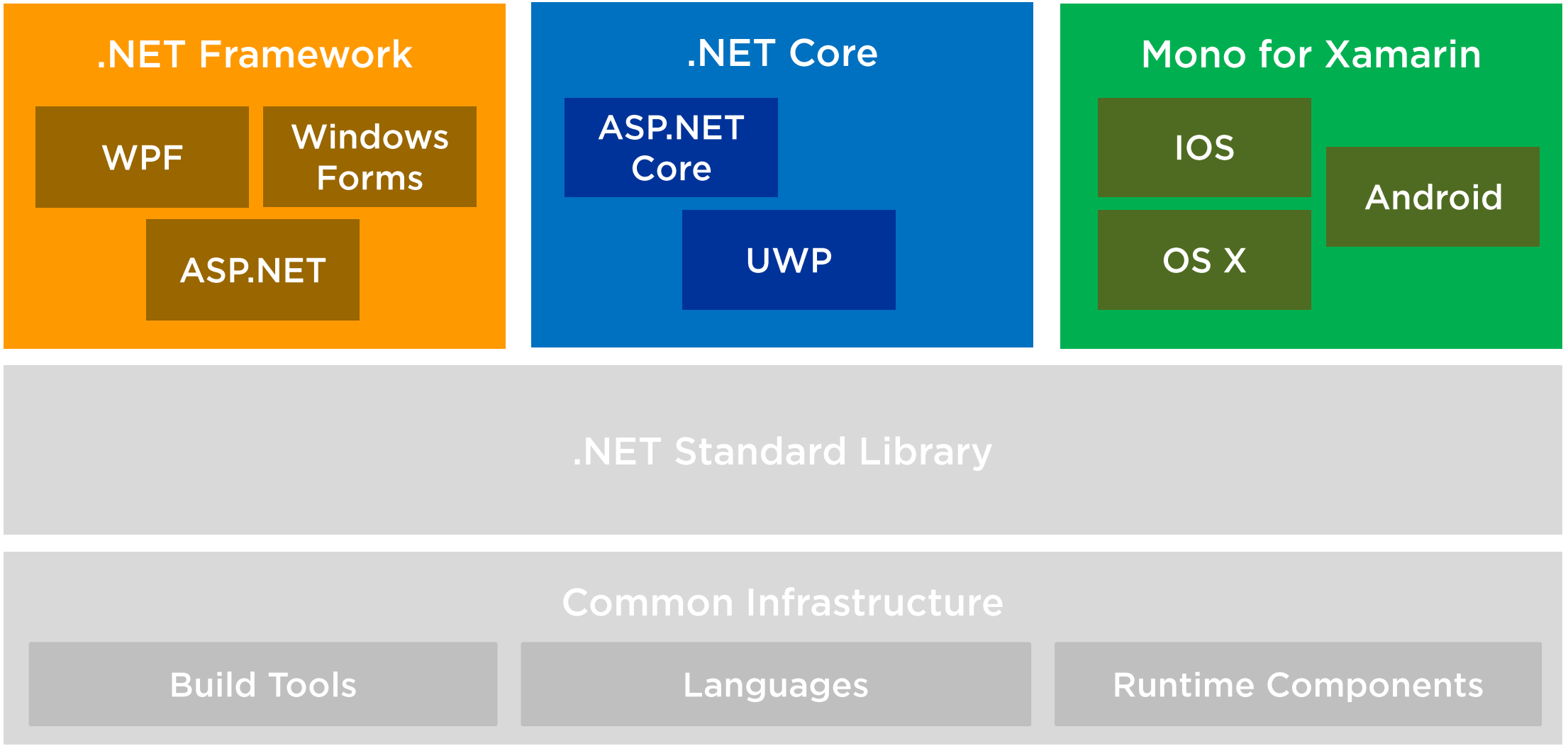
The .NET Ecosystem



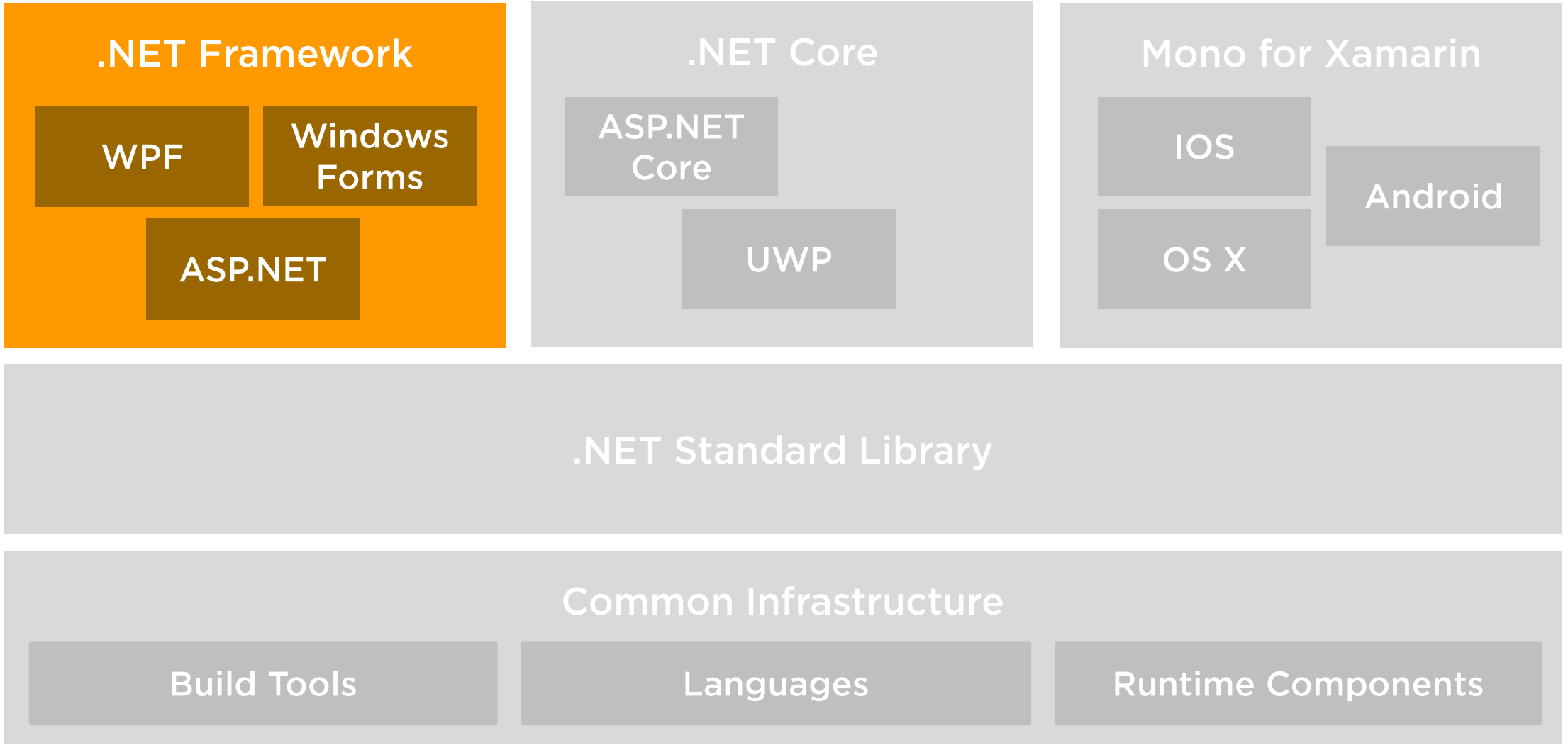
The .NET Ecosystem



The .NET Ecosystem

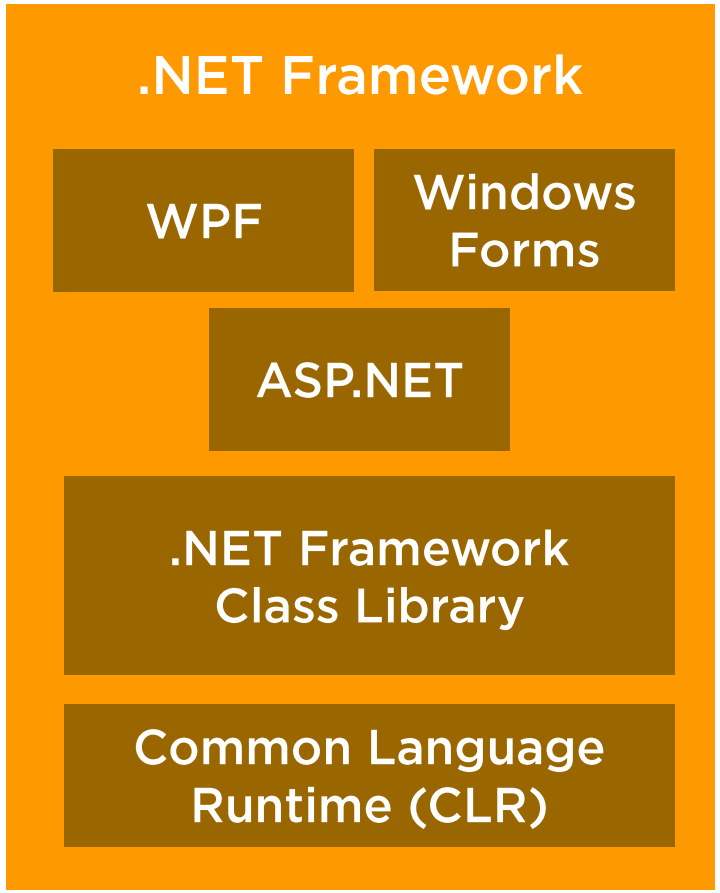


The .NET Ecosystem

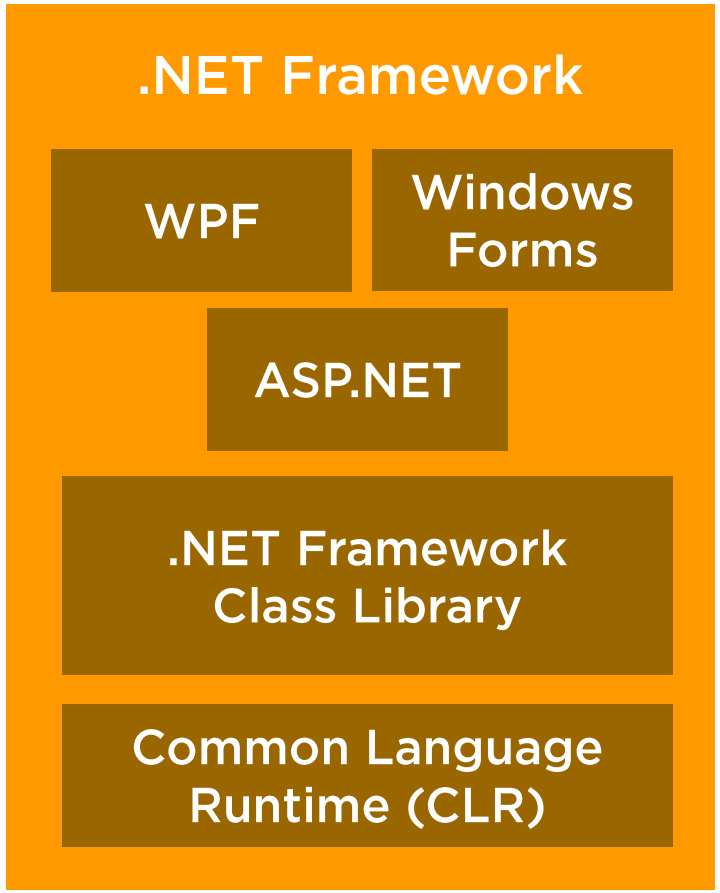


The .NET Framework

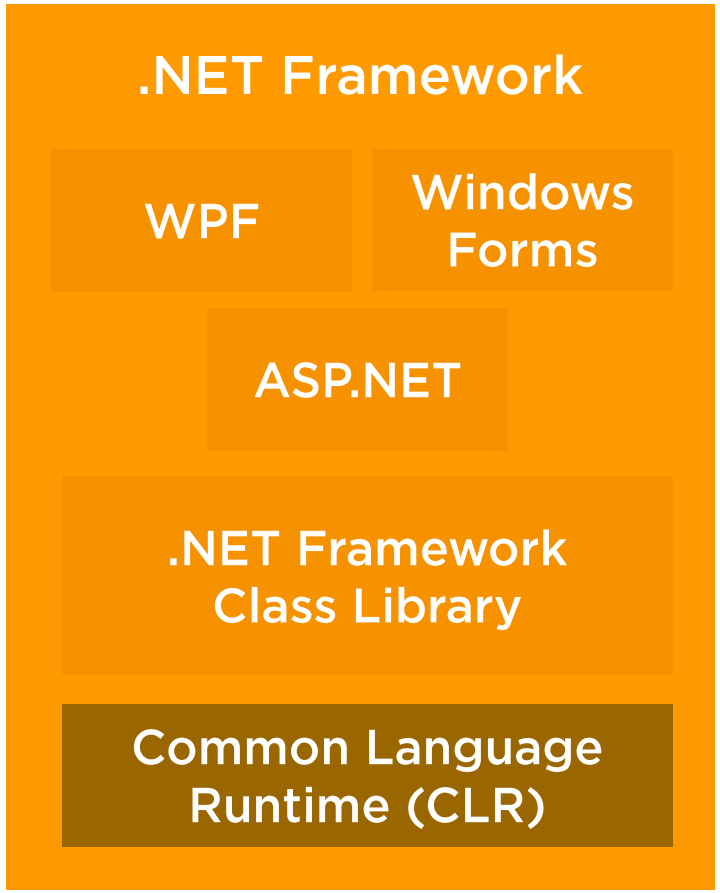
- Released in 2002
- Partly open-source
  - github.com/microsoft/referencesource



The .NET Framework

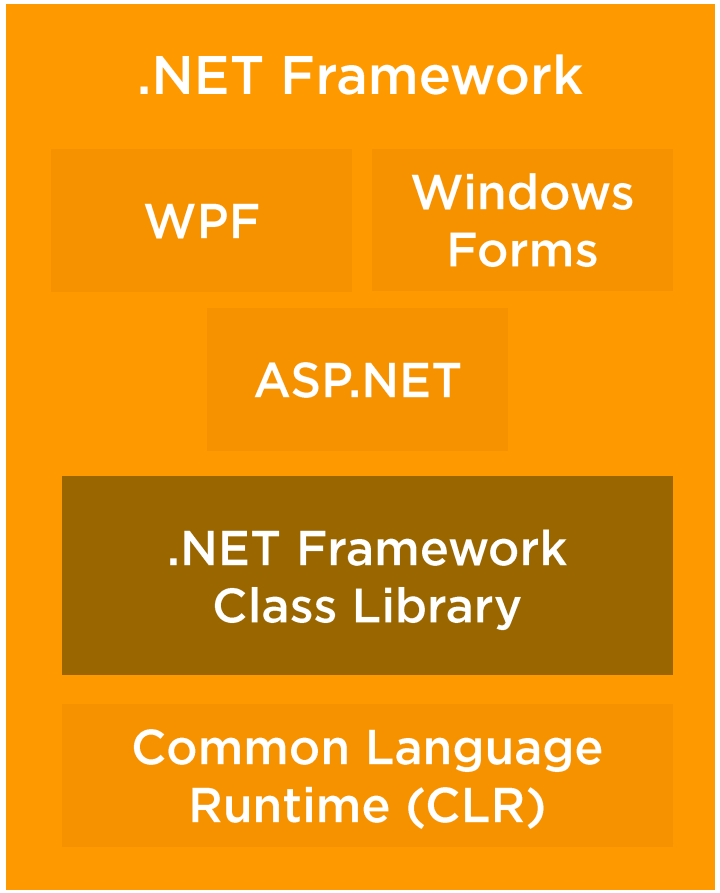


The .NET Framework



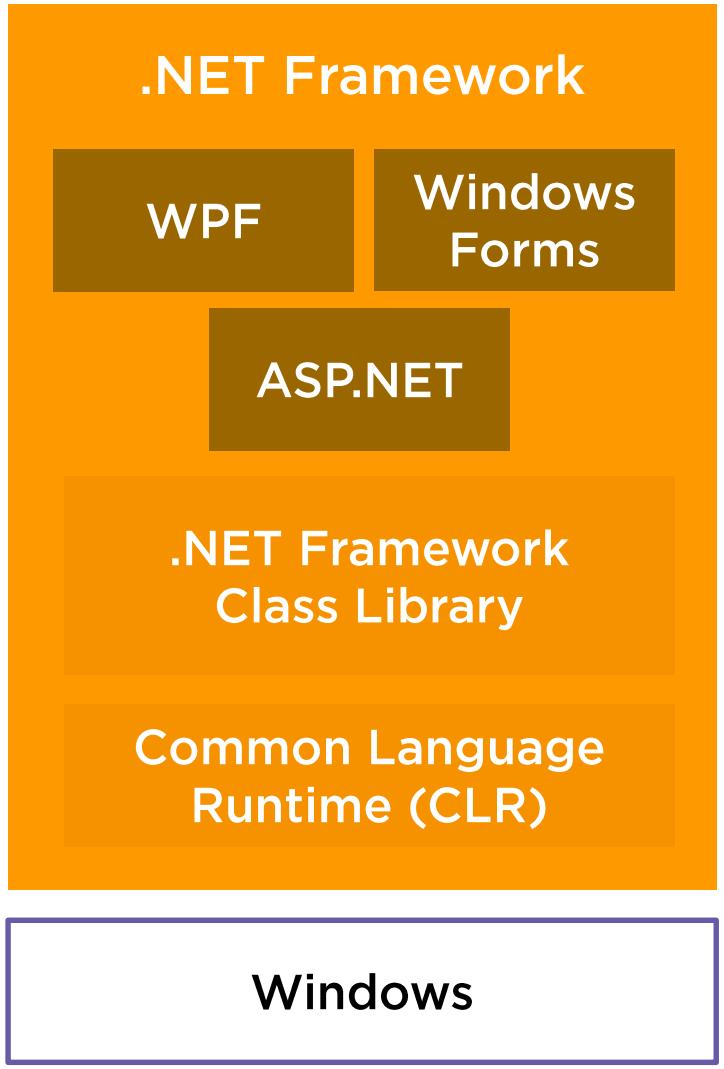
- Run code, Garbage Collection
- C#, VB.NET, F#

The .NET Framework



- Classes, interfaces and value types that provide capabilities

The .NET Framework

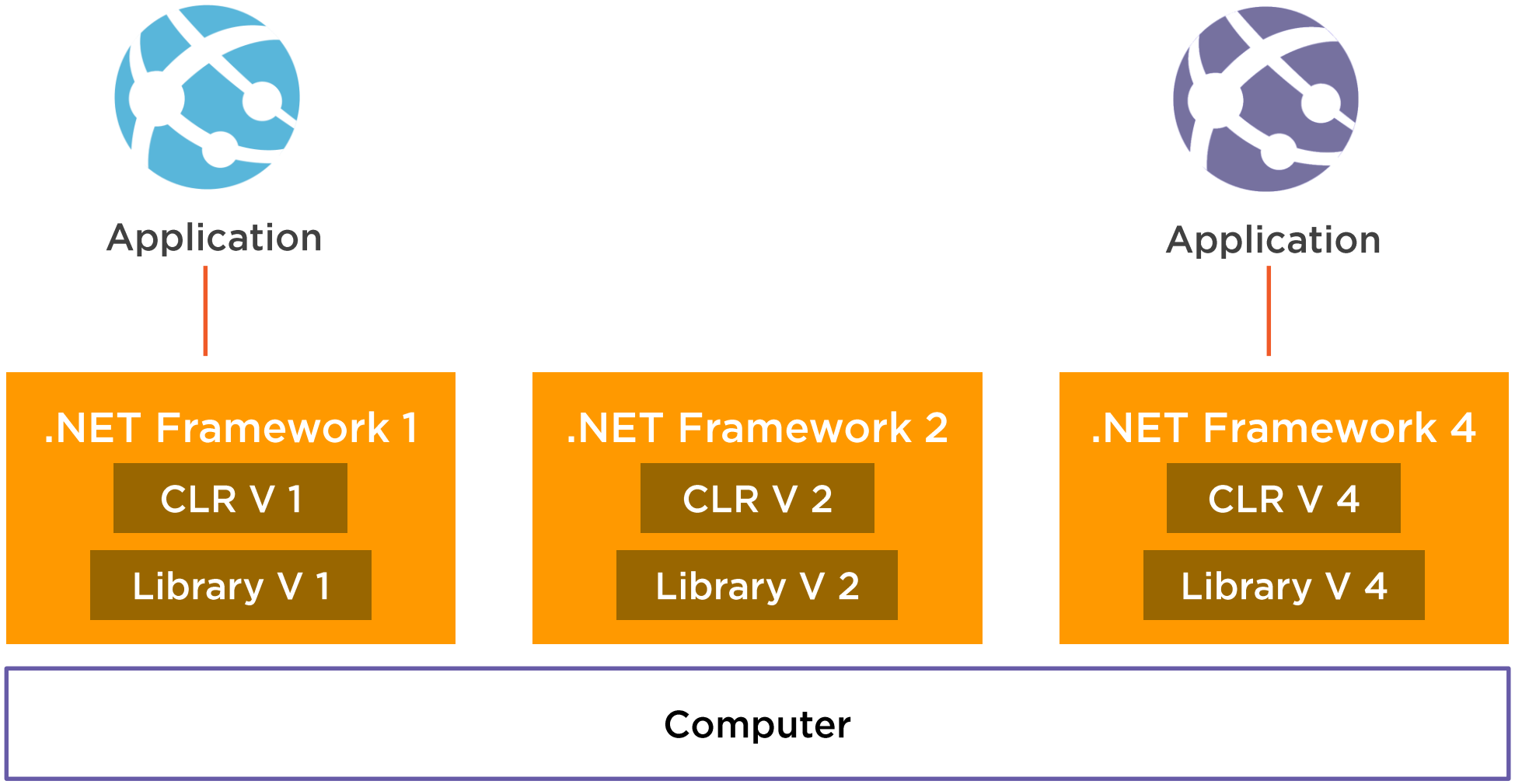


Workloads (application types)

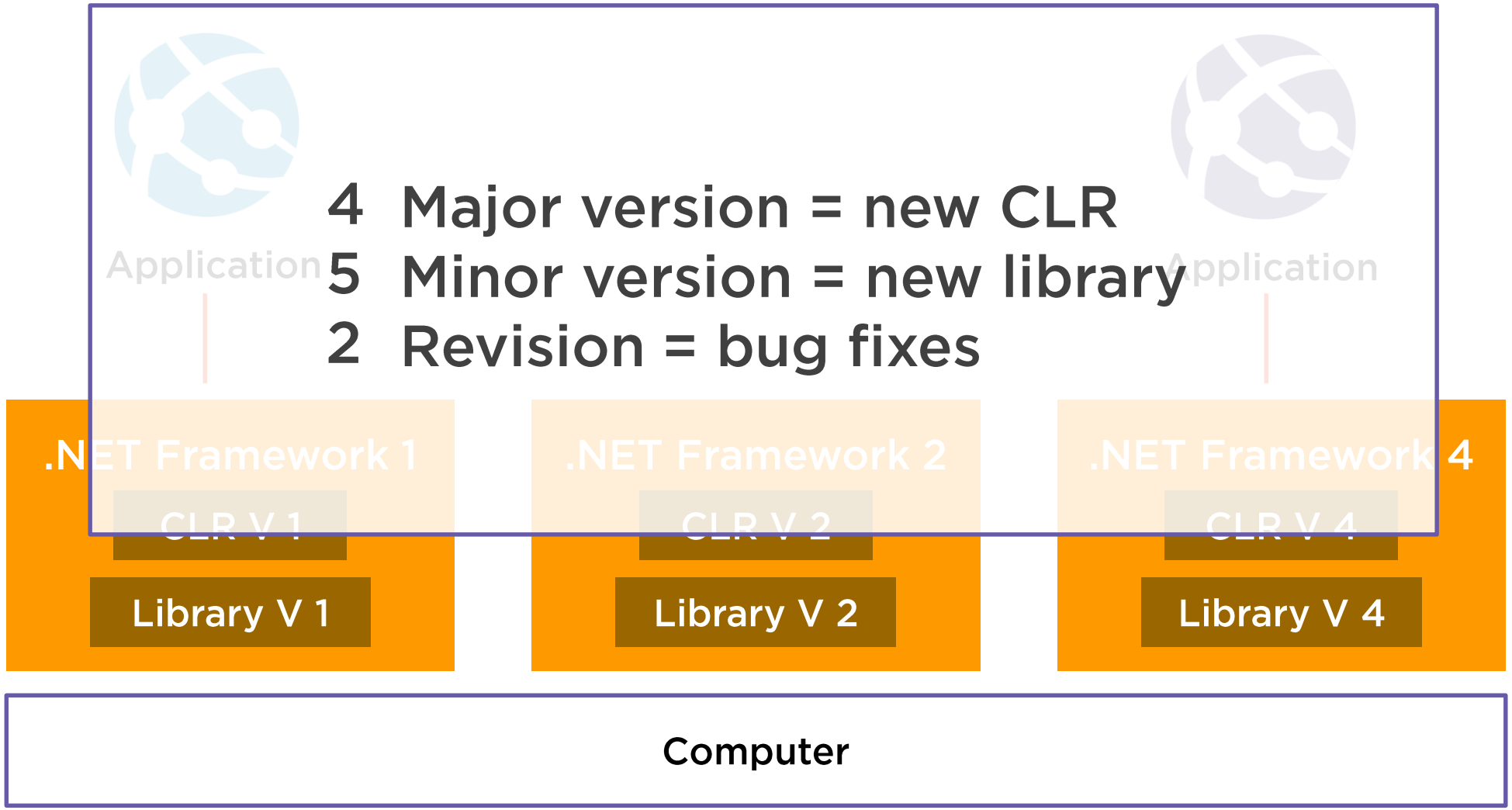
- Console applications
- Windows Communication Foundation (WCF)
- Windows Workflow Foundation (WF)
- Windows Presentation Foundation (WPF)
- Windows Forms
- ASP.NET
  - Forms, MVC, Web API
- Azure (WebJobs, Cloud Services)
- ...

Windows APIs

The .NET Framework



The .NET Framework



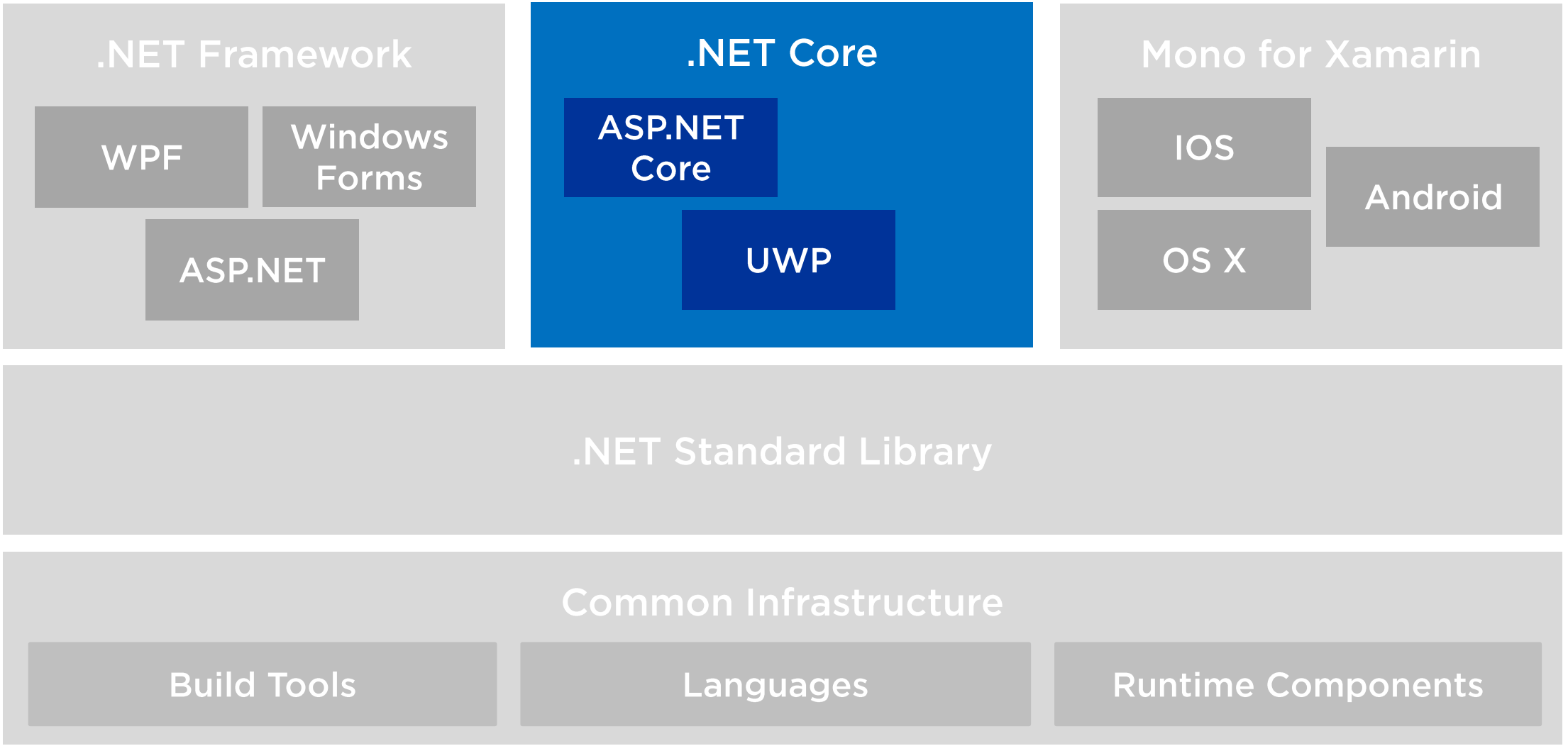
Demo

Windows Forms project

- Windows specific

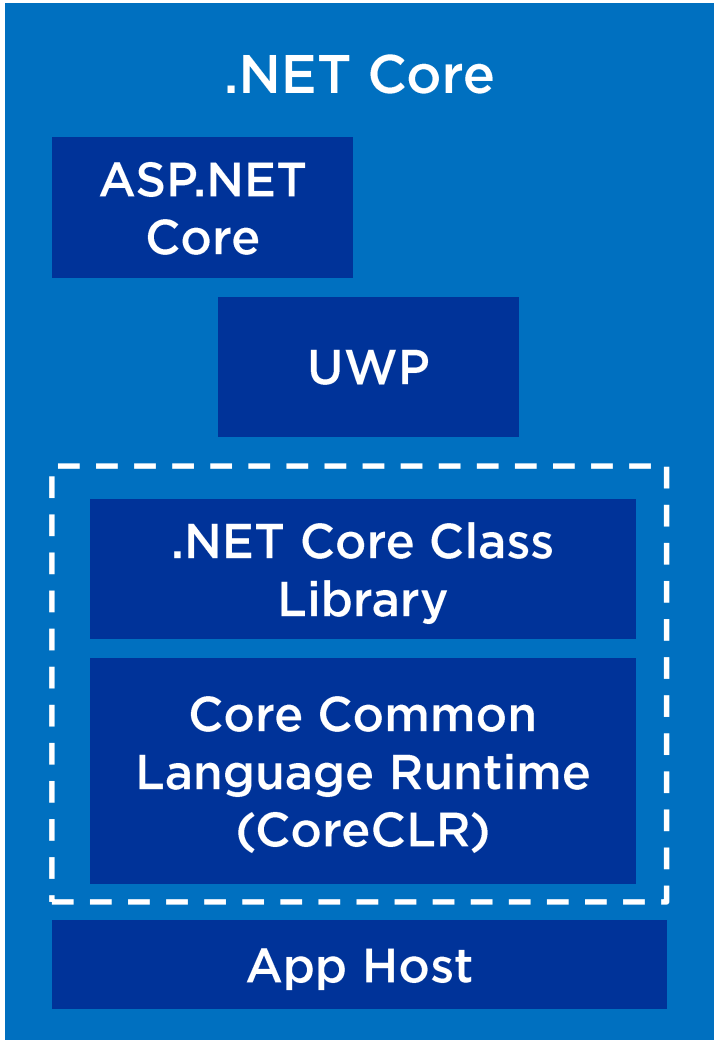
Where the .NET Framework is installed

The .NET Ecosystem

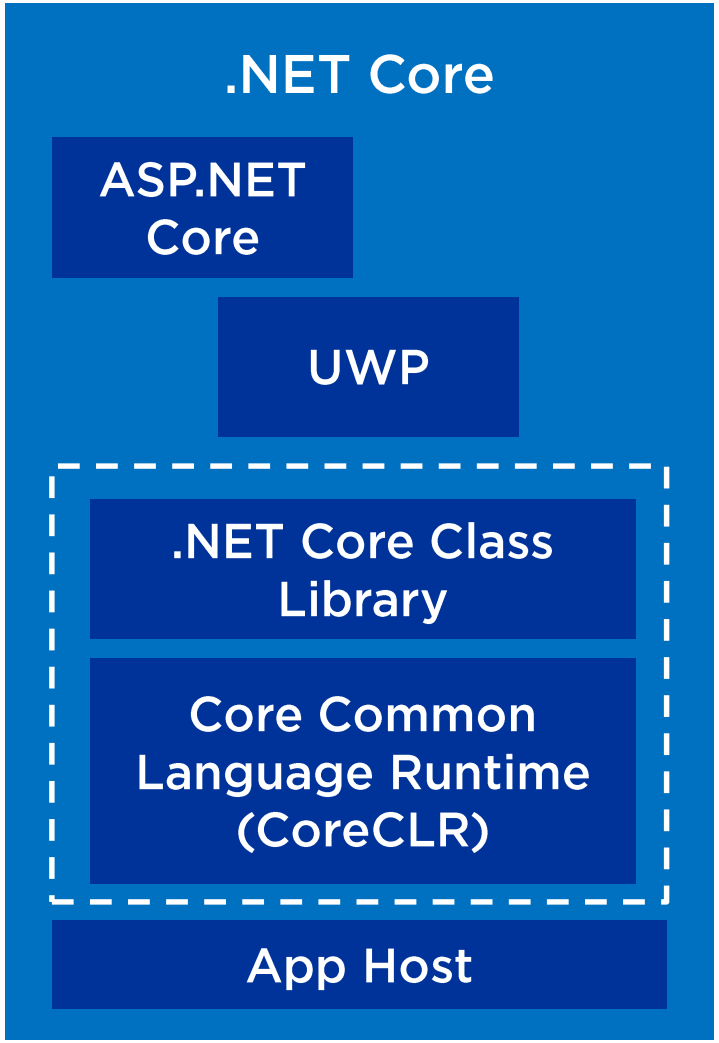


.NET Core

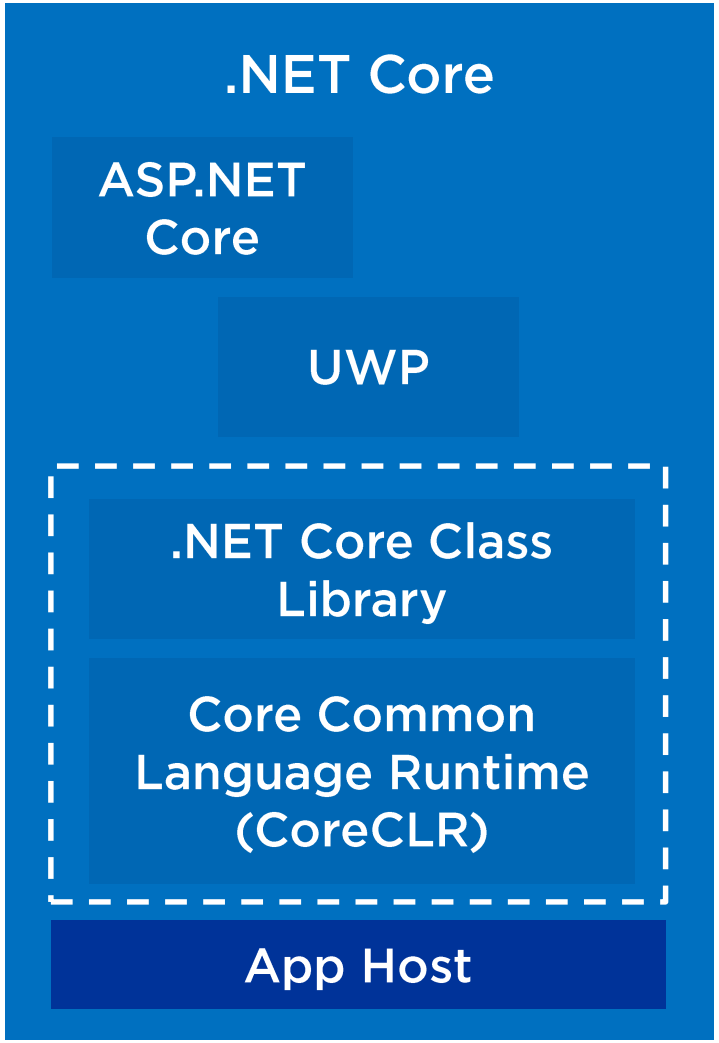
- Released in 2016
- Open source
  - https://github.com/dotnet/core



.NET Core

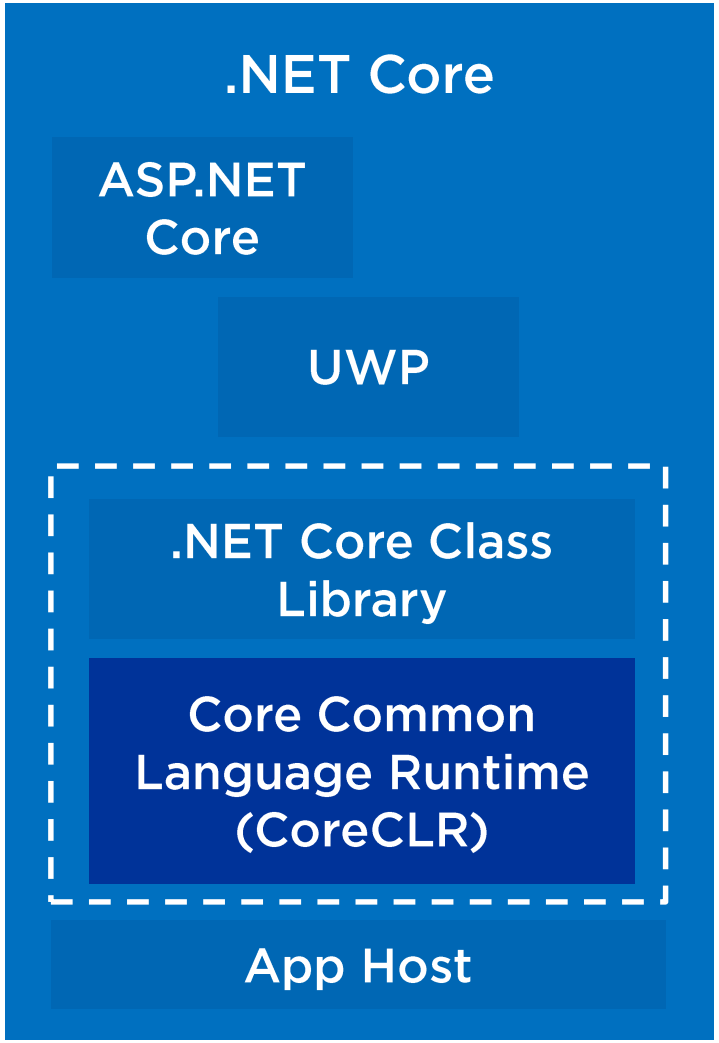

.NET Core
ASP.NET Core
UWP
.NET Core Class Library
Core Common Language Runtime (CoreCLR)
App Host

.NET Core



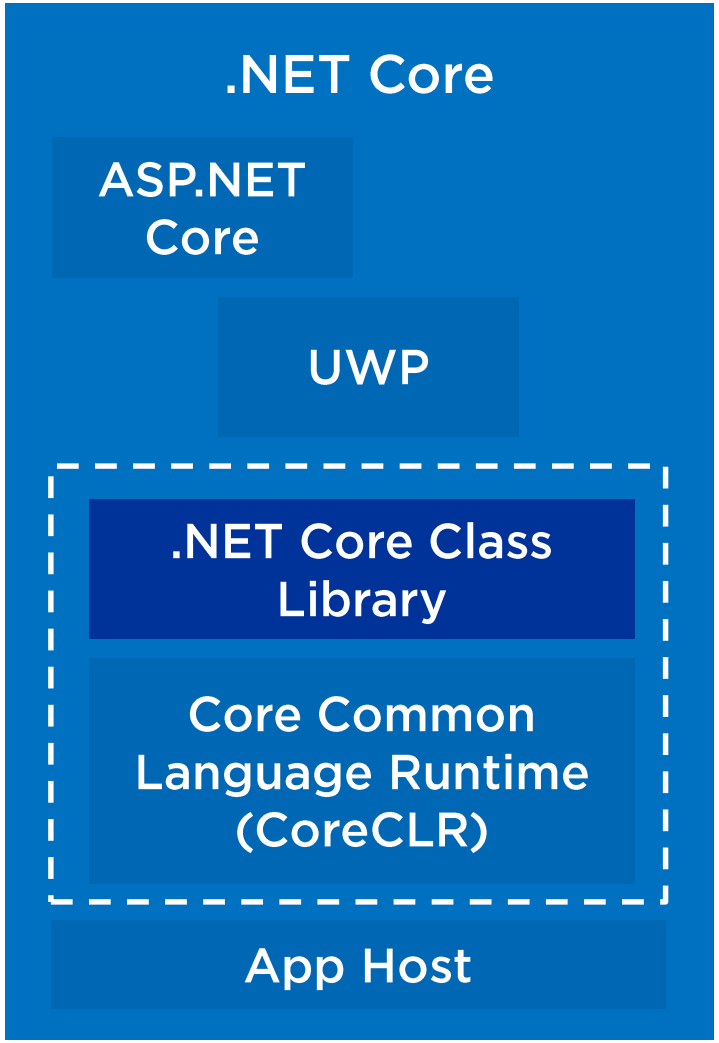
- Hosts the CoreCLR and launches the app

.NET Core



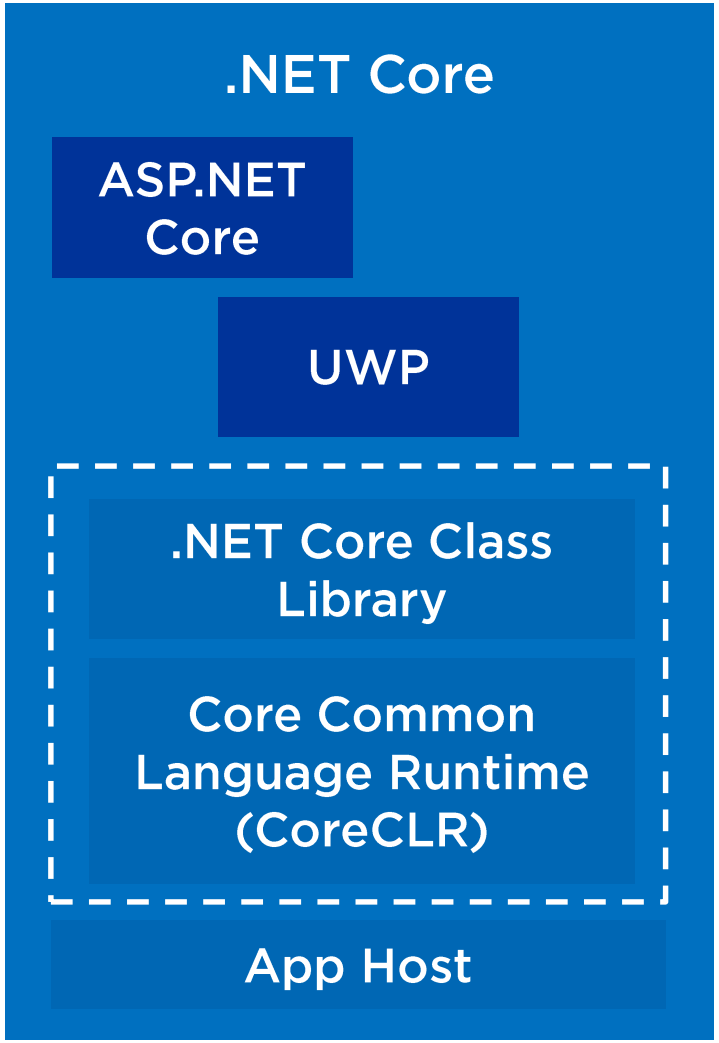
- Assembly loading, Garbage Collection
- C#, VB.NET, F#

.NET Core



- Classes that provide capabilities
- Subset of the .NET Framework library

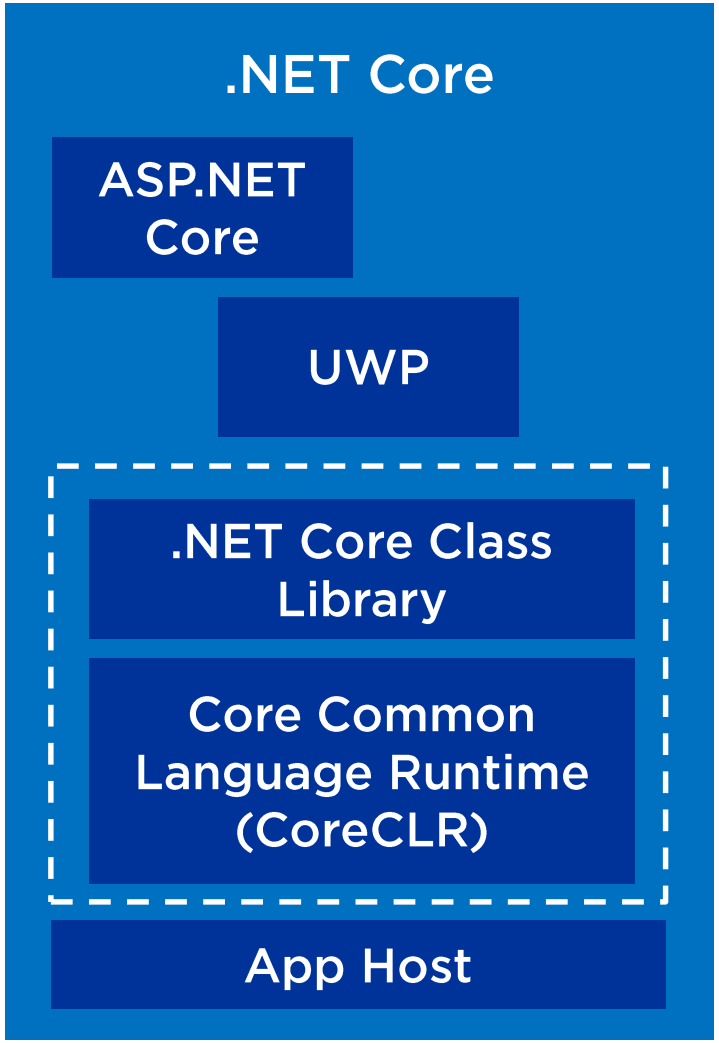
.NET Core



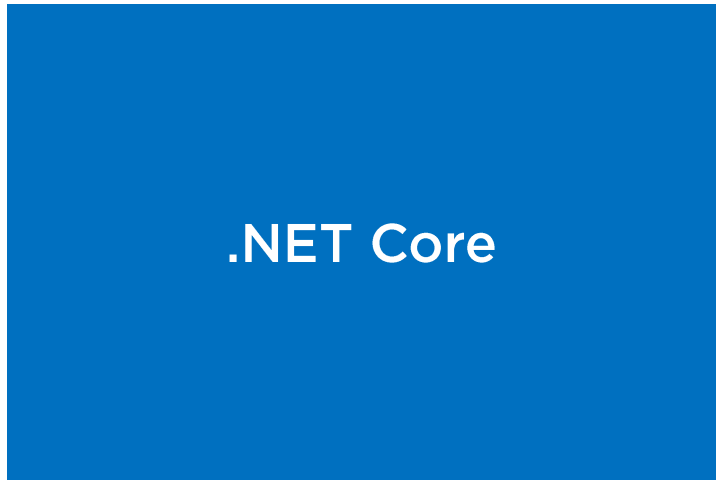
Workloads (application types)

- Console applications
- ASP.NET Core
  - MVC
  - API
- Universal Windows Platform Apps
- ...

.NET Core


.NET Core
ASP.NET Core
UWP
.NET Core Class Library
Core Common Language Runtime (CoreCLR)
App Host

.NET Core



Windows Client

Windows IoT

Ubuntu

FreeBSD

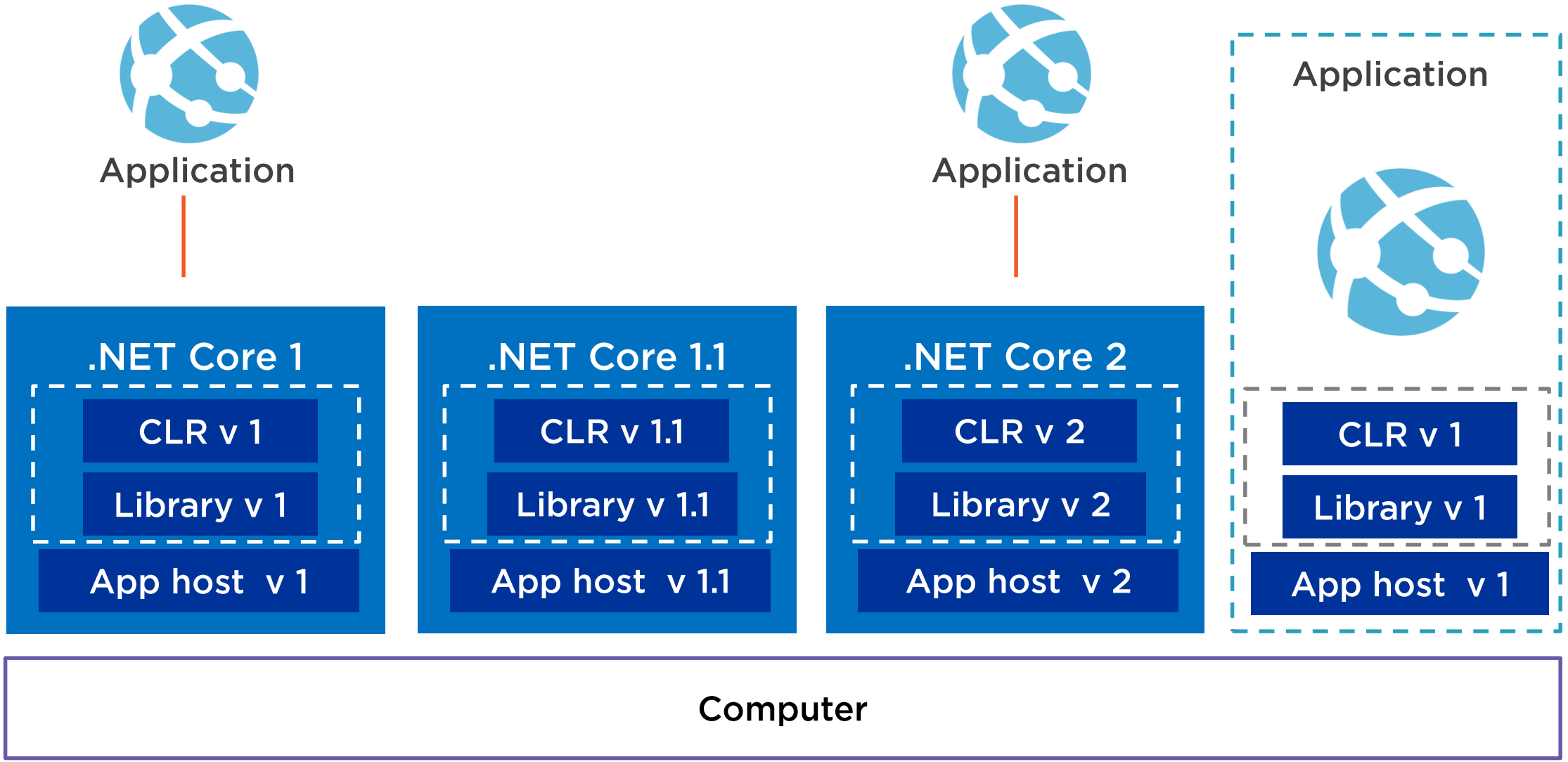
Windows Server

(Red Hat) Linux

Tizen

Mac OS X

.NET Core


Application
Application
Application
.NET Core 1
CLR v 1
Library v 1
App host v 1
.NET Core 1.1
CLR v 1.1
Library v 1.1
App host v 1.1
.NET Core 2
CLR v 2
Library v 2
App host v 2
CLR v 1
Library v 1
App host v 1
Computer

Demo

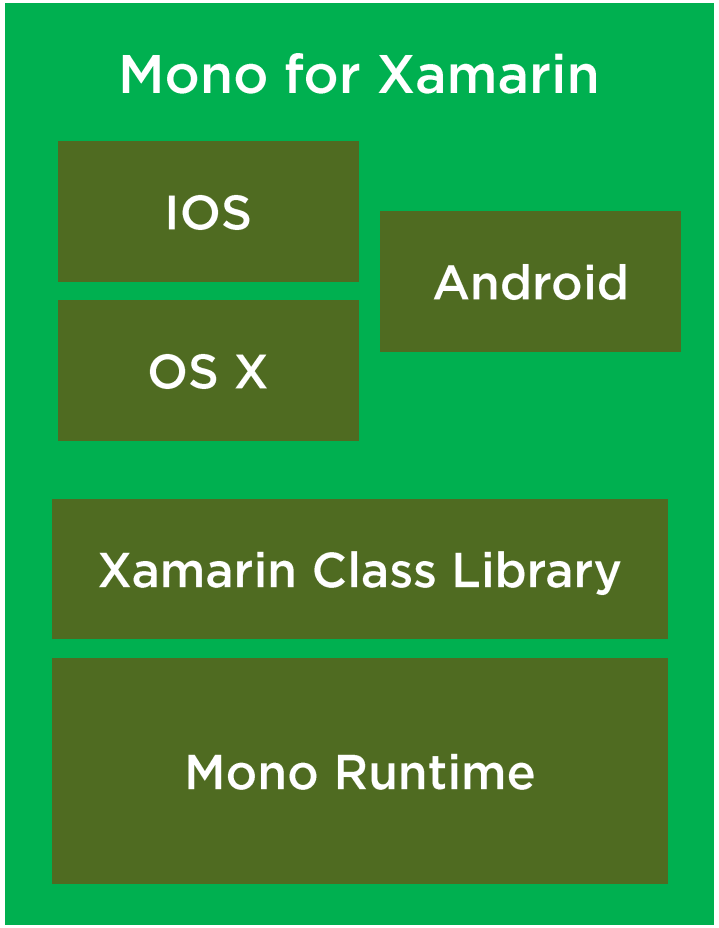
Simple .NET Core application

- Framework dependent application
- Self-contained application

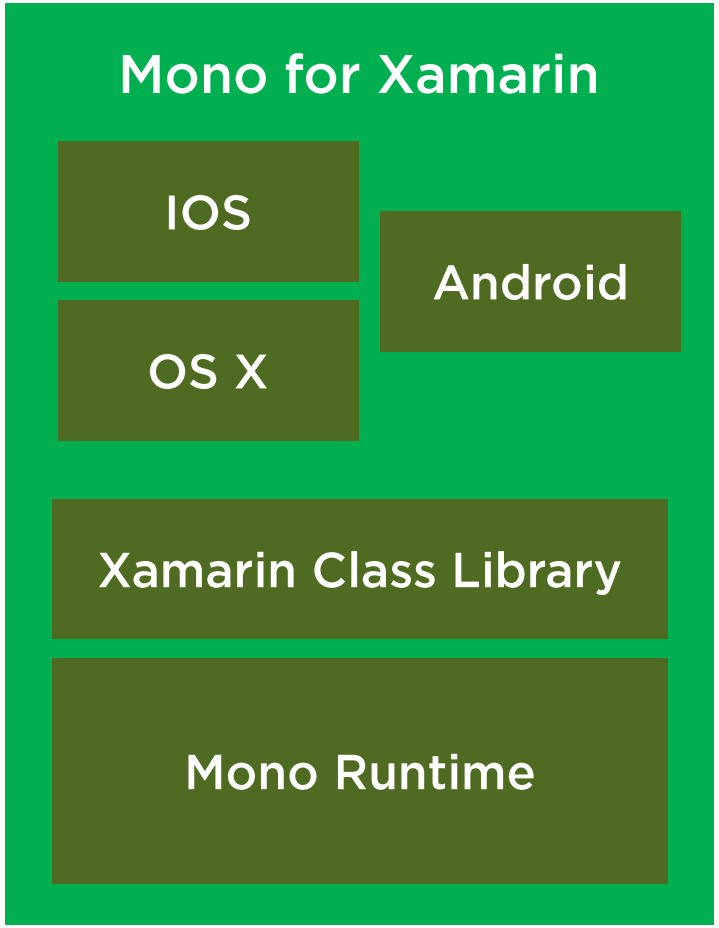
Cross-platform

Mono for Xamarin

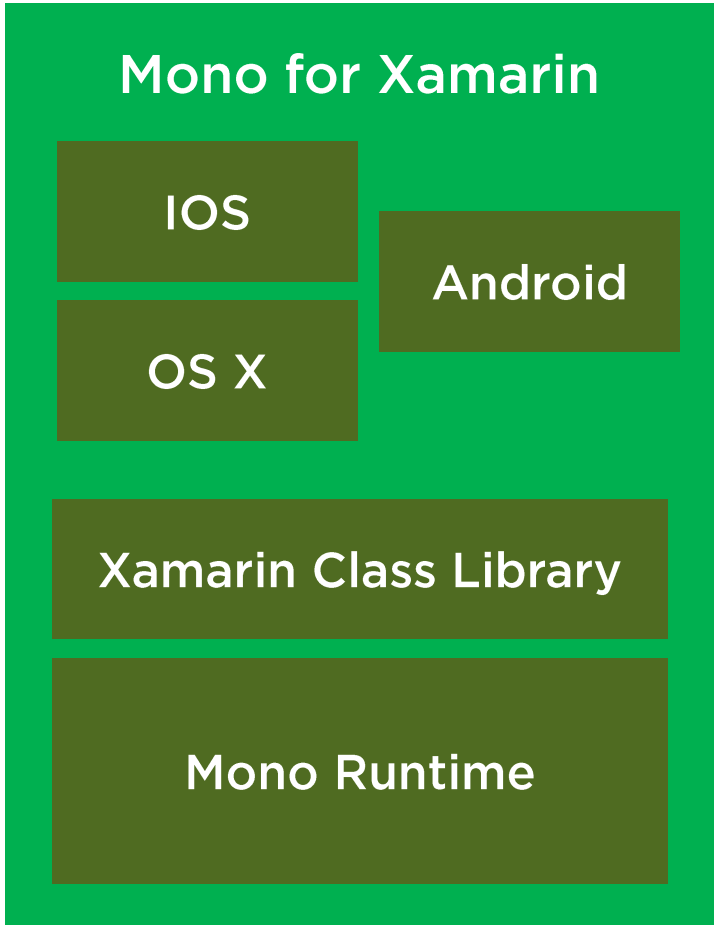
- Released in 2011
- Mono is open source
  - https://github.com/mono/mono
- www.mono-project.com



Mono for Xamarin

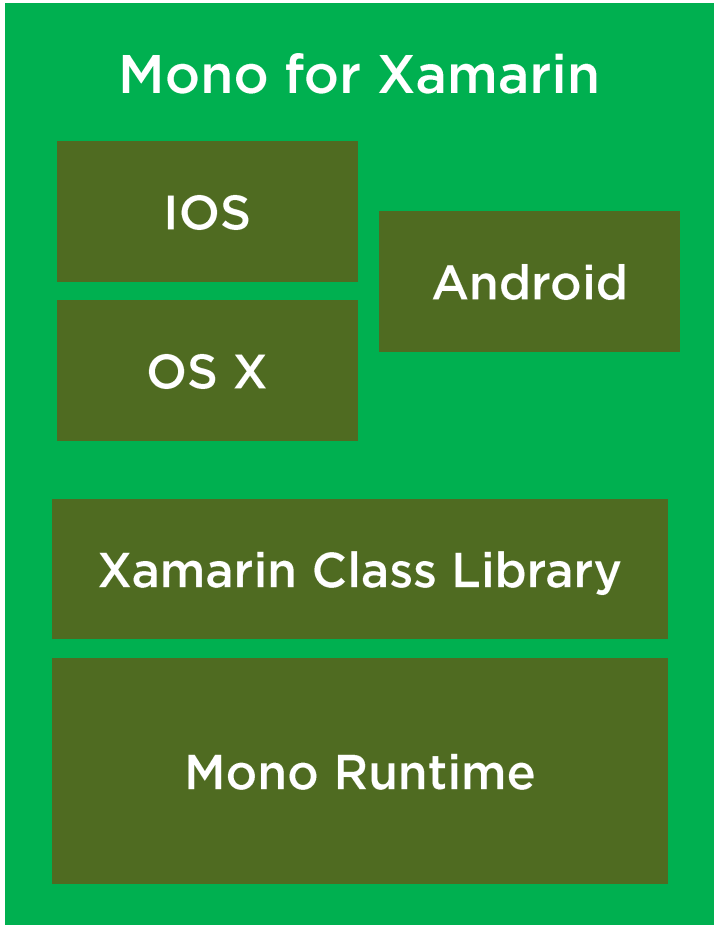


Mono for Xamarin



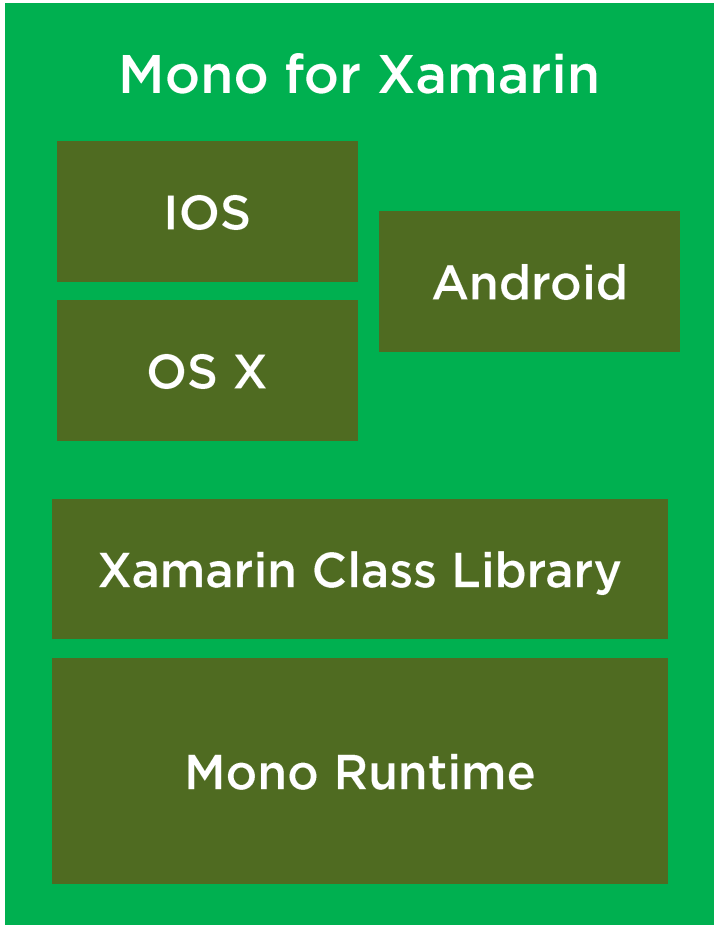
- Garbage Collection, JIT, AOT
- C#

Mono for Xamarin



- Classes that provide capabilities
- Subset of the .NET Framework library

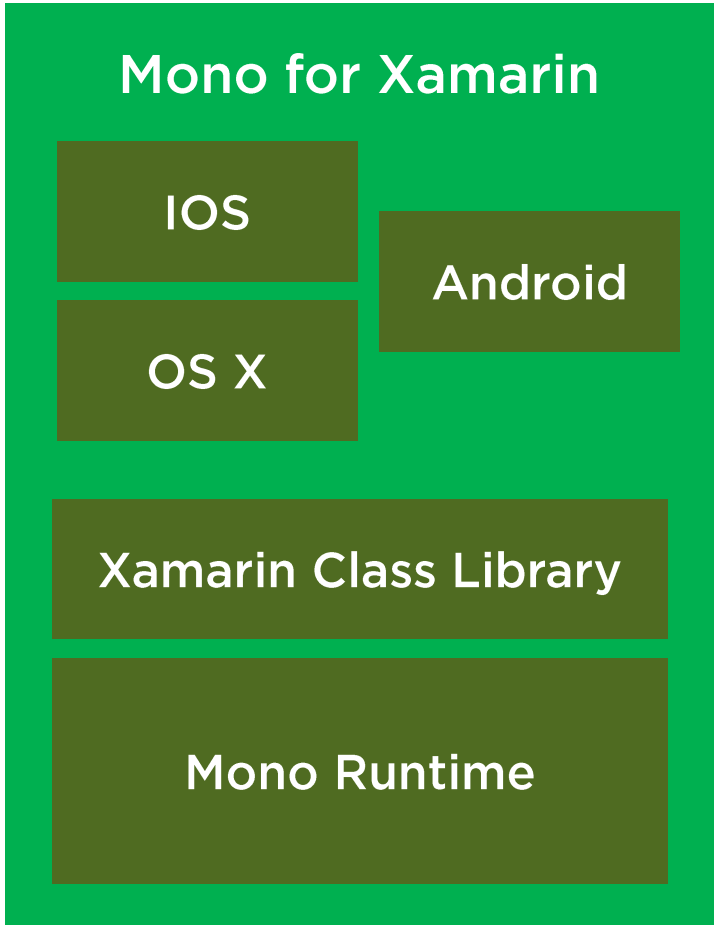
Mono for Xamarin



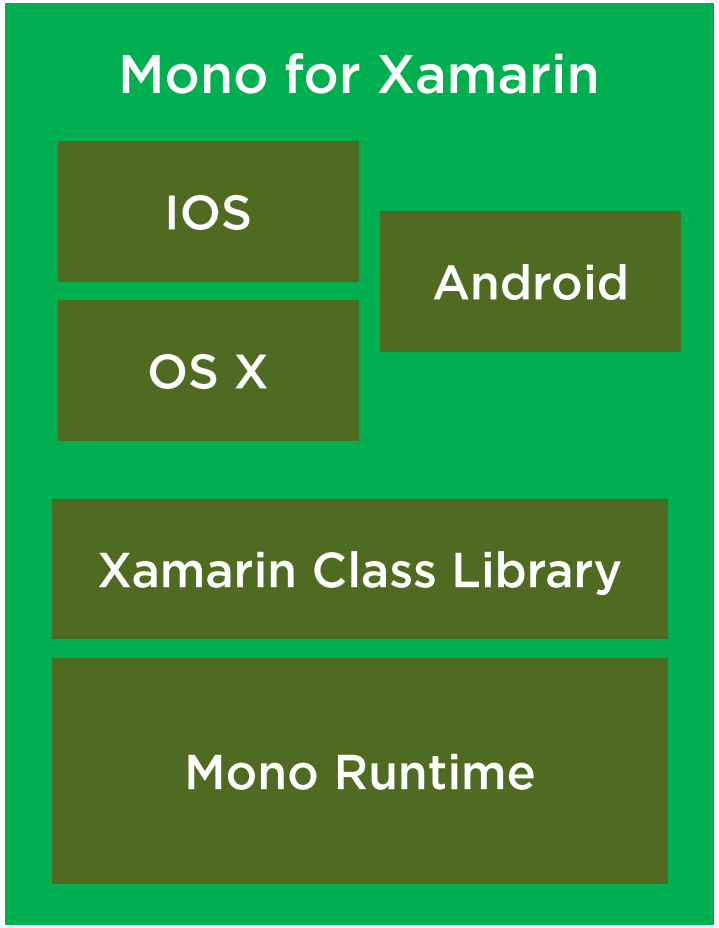
Workloads (application types)

- Apple IOS applications
- Apple Mac OS X applications
- Android applications
- ...

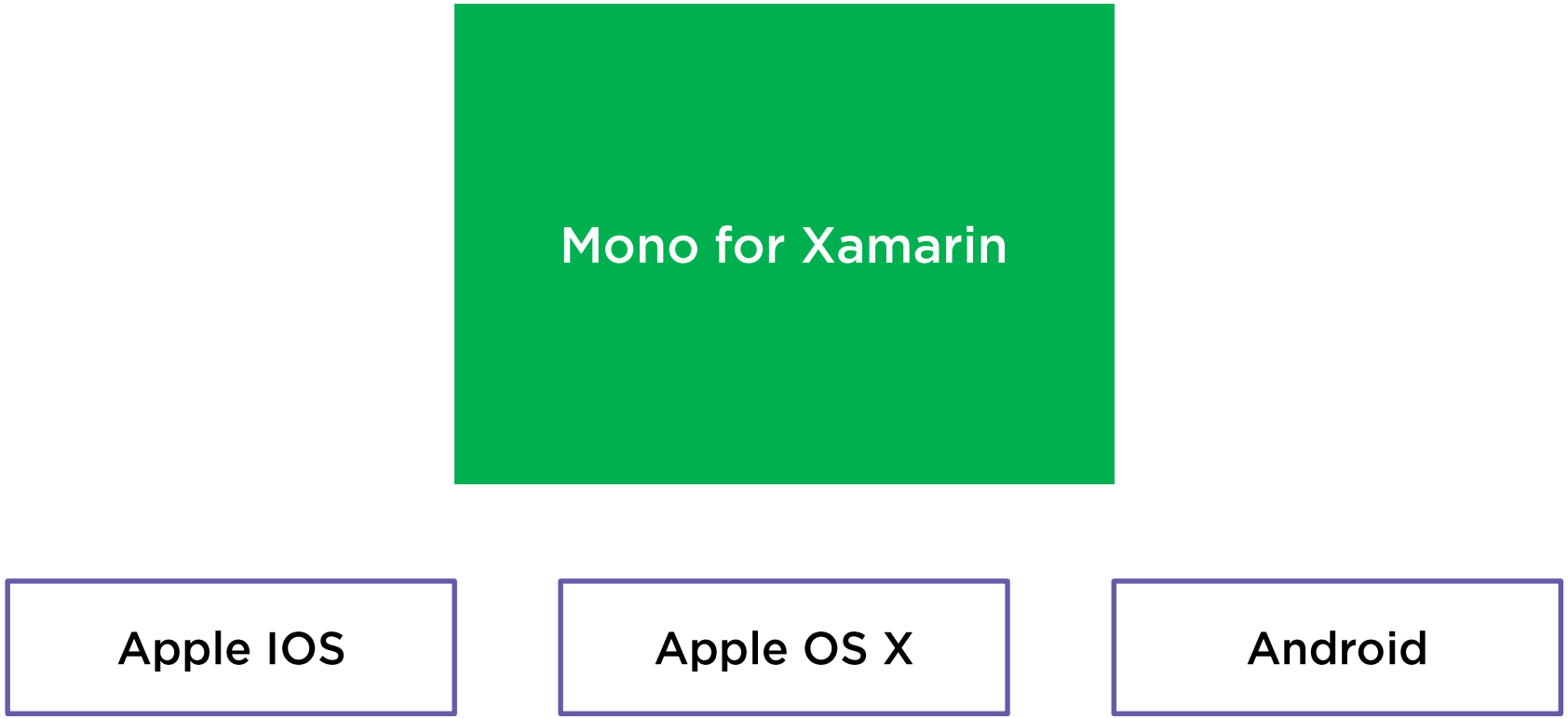
Mono for Xamarin



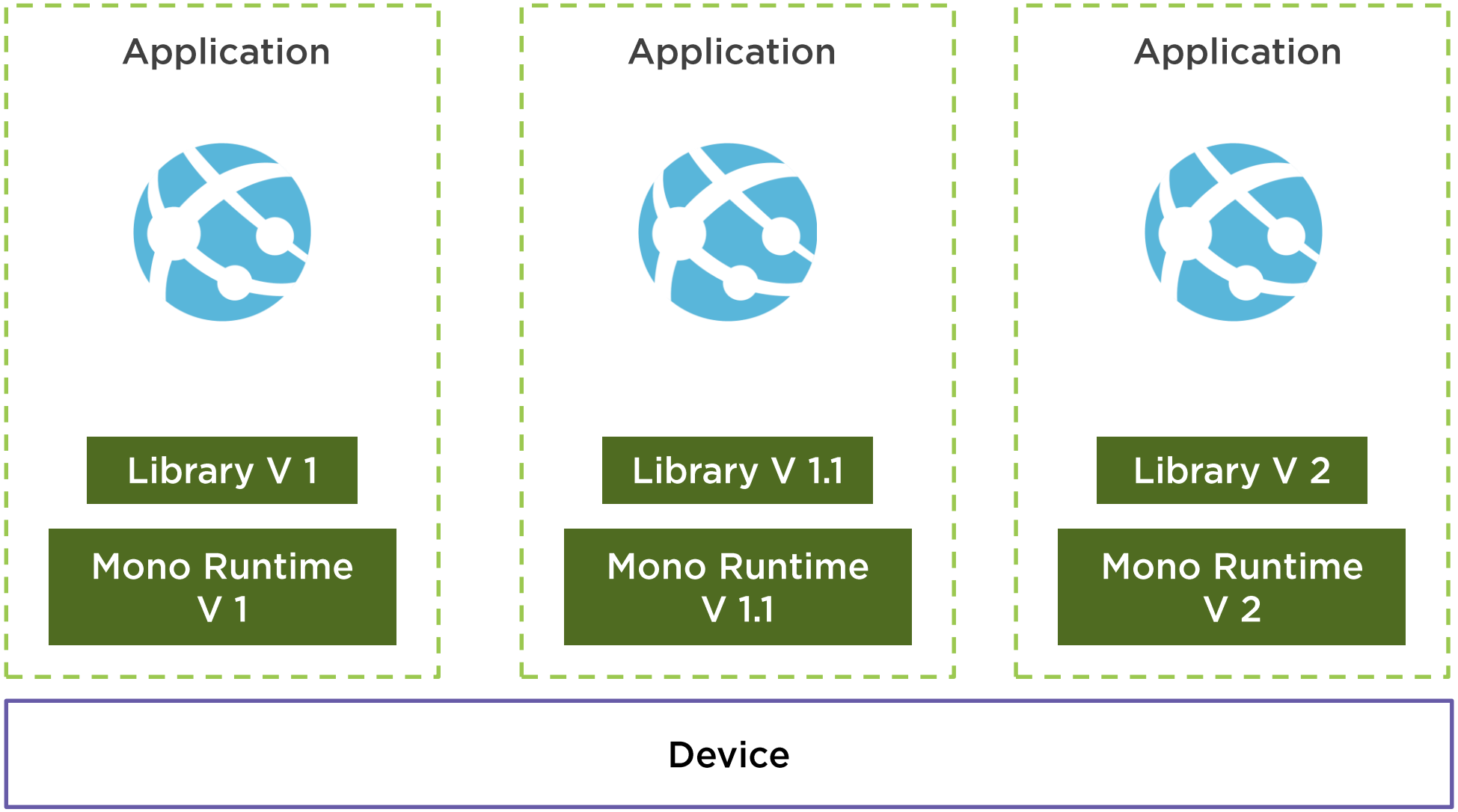
Mono for Xamarin



Mono for Xamarin



Mono for Xamarin



Demo

Simple Xamarin.Mono app

Let’s compare runtimes!

Comparison of Runtimes

	.NET Framework	.NET Core	Mono for Xamarin

Comparison of Runtimes

	.NET Framework	.NET Core	Mono for Xamarin
Workloads	WPF, Windows Forms, ASP.NET	ASP.NET Core, UWP	IOS, Mac OS X, Android

Comparison of Runtimes

	.NET Framework	.NET Core	Mono for Xamarin
Workloads	WPF, Windows Forms, ASP.NET	ASP.NET Core, UWP	IOS, Mac OS X, Android
Cross-platform		X	X

Comparison of Runtimes

	.NET Framework	.NET Core	Mono for Xamarin
Workloads	WPF, Windows Forms, ASP.NET	ASP.NET Core, UWP	IOS, Mac OS X, Android
Cross-platform		X	X
Side-by-side	Only major versions	X	X

Comparison of Runtimes

	.NET Framework	.NET Core	Mono for Xamarin
Workloads	WPF, Windows Forms, ASP.NET	ASP.NET Core, UWP	IOS, Mac OS X, Android
Cross-platform		X	X
Side-by-side	Only major versions	X	X
Self-contained		X	X

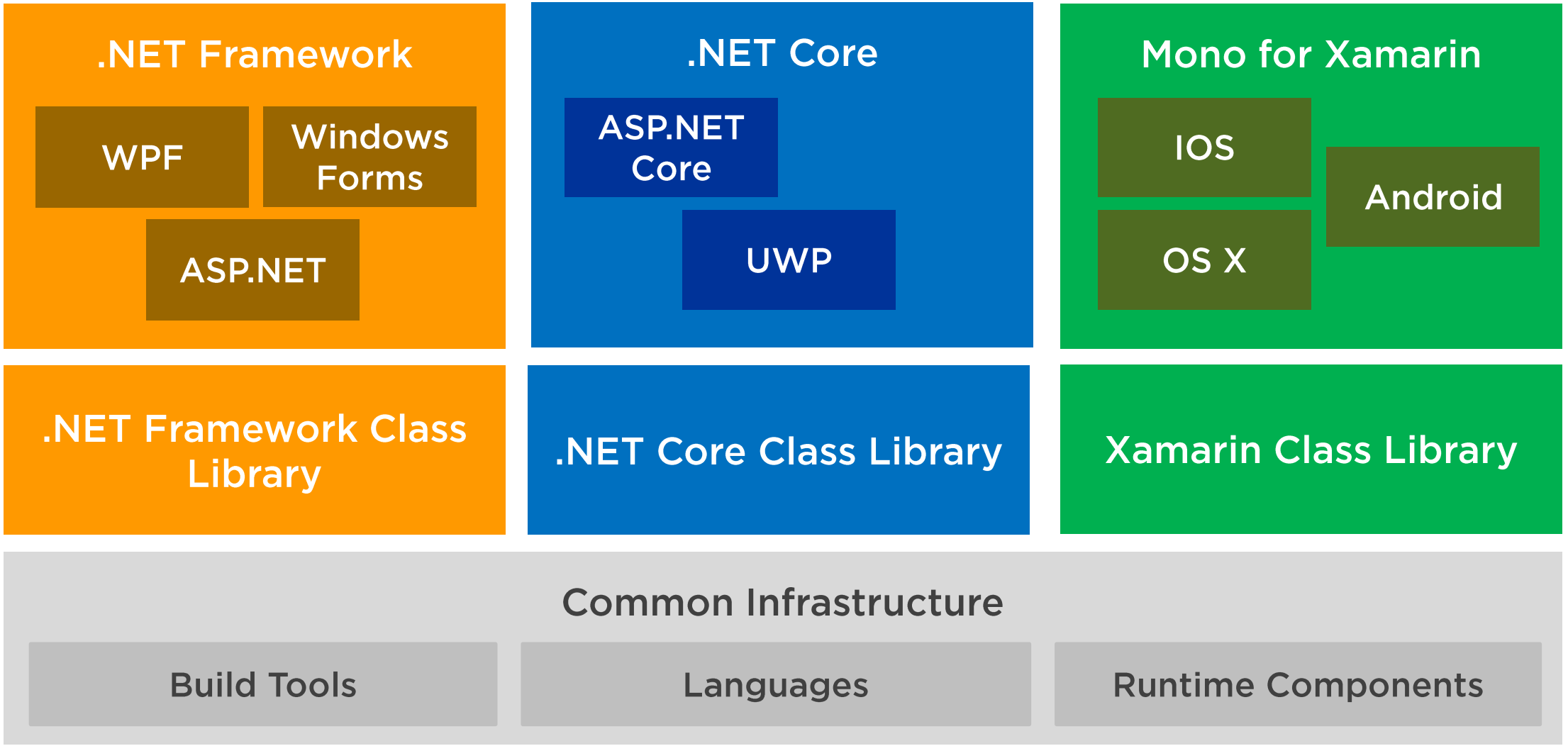
Comparison of Runtimes

	.NET Framework	.NET Core	Mono for Xamarin
Workloads	WPF, Windows Forms, ASP.NET	ASP.NET Core, UWP	IOS, Mac OS X, Android
Cross-platform		X	X
Side-by-side	Only major versions	X	X
Self-contained		X	X
Main purpose	Windows desktop apps	Cross-platform web and desktop apps	Cross-platform mobile apps

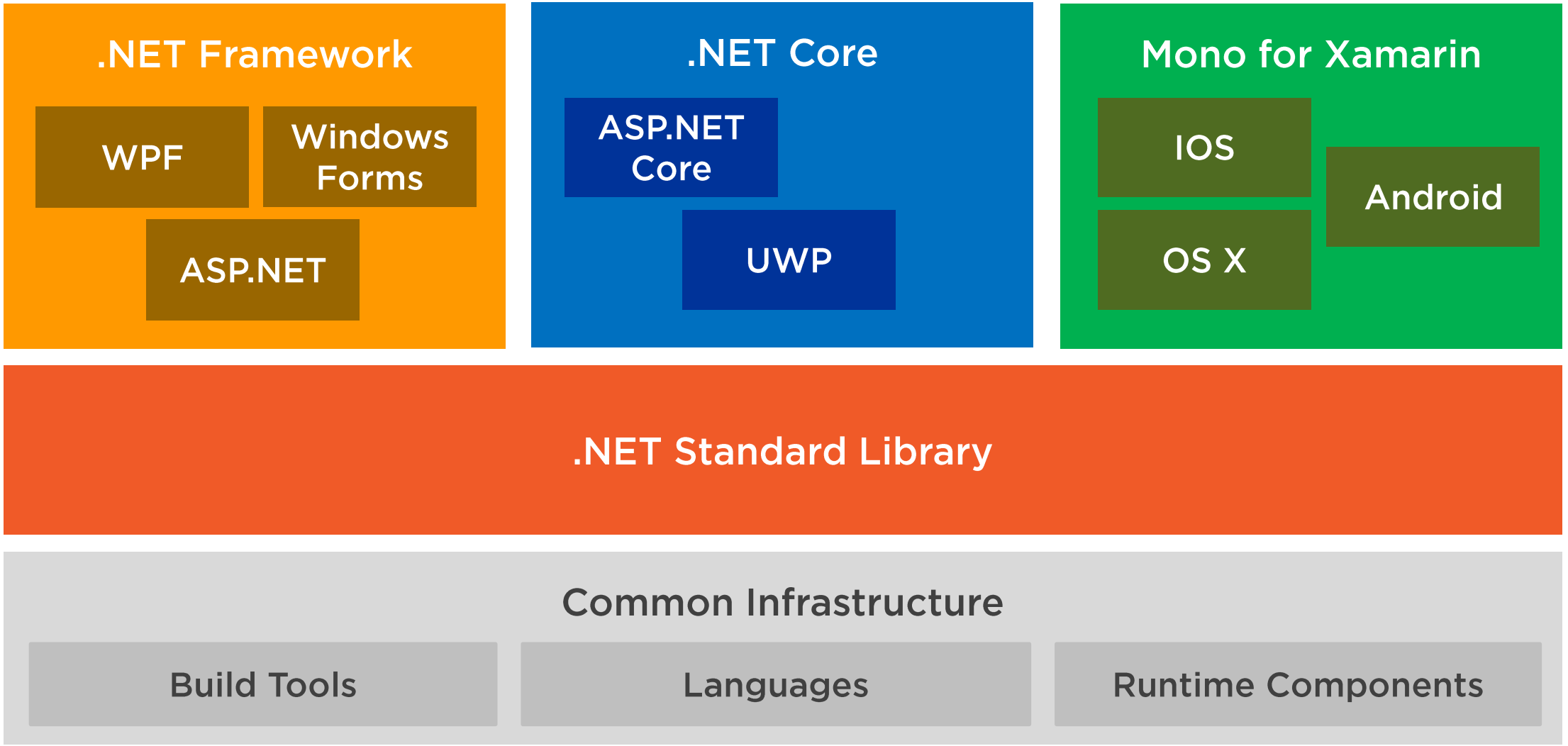
Sharing is Caring

(share code with .NET Standard)

Class Libraries in the .NET Ecosystem



.NET Standard



.NET Standard

.NET Standard

.NET Standard Library

- Not something that you install!
- Formal specification of .NET APIs
- Evolution of Portable Class Libraries (PCL)
- Runtimes implement .NET Standard
- Runtime versions implement .NET standard versions
  - .NET Framework 4.5 implements .NET standard <= 1.1

.NET Standard

.NET Standard Library

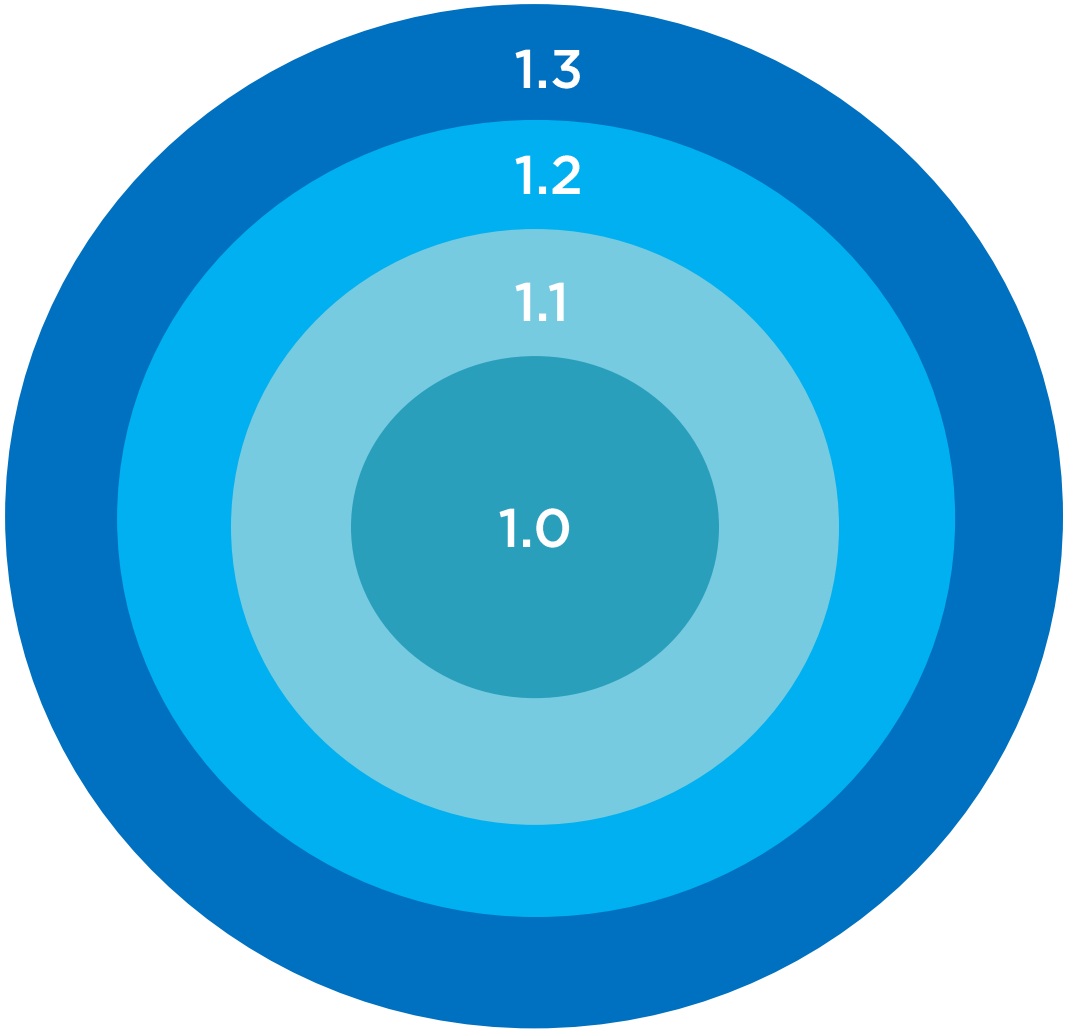
- Not something that you install!
- Formal specification of .NET APIs
- Runtimes implement .NET Standard
- Runtime versions implement .NET standard versions
  - .NET Framework 4.5 implements .NET standard <= 1.1
- Evolution of Portable Class Libraries (PCL)

Share code between runtimes

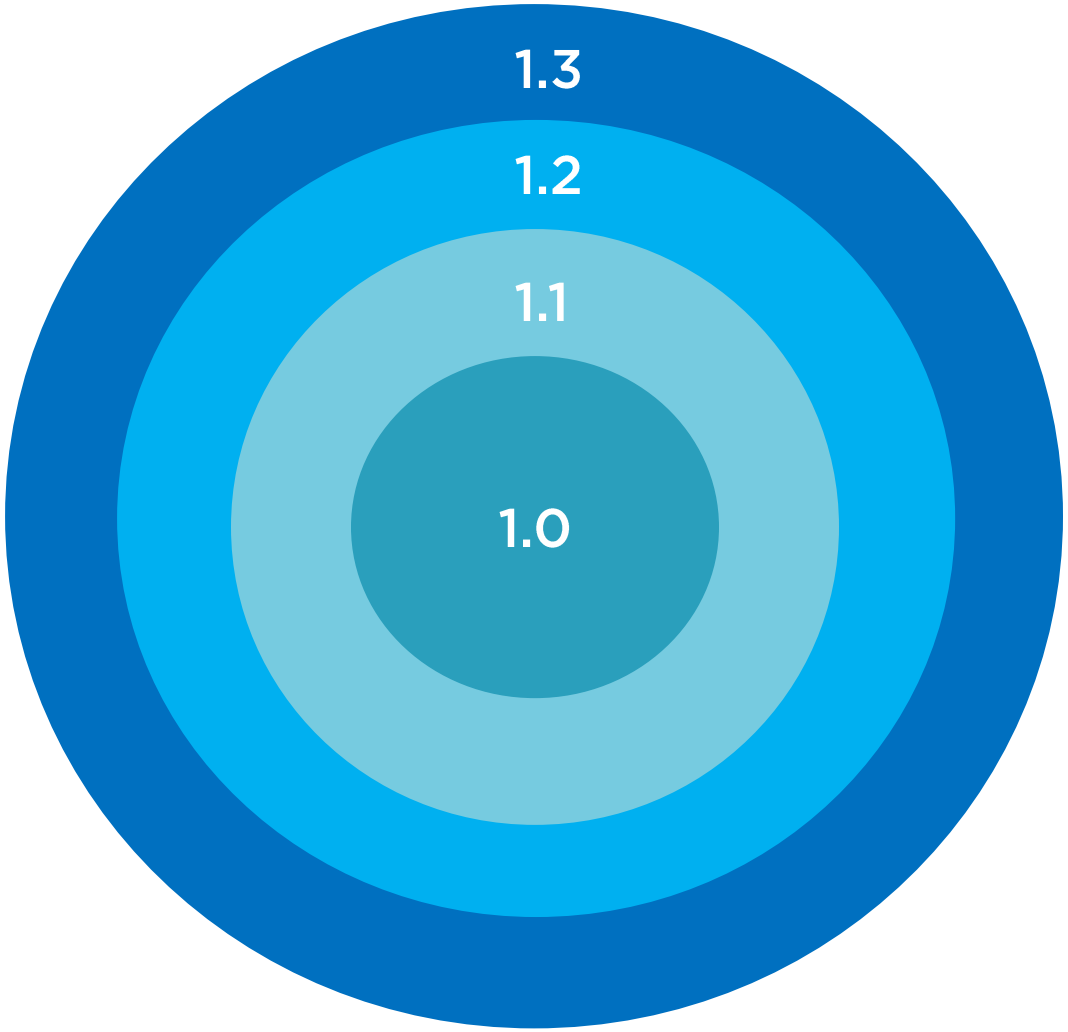
Comparison with Portable Class Libraries

.NET Standard	Portable Class Libraries (PCL)
• Curated set of APIs • Is platform agnostic	• APIs are defined by the platforms you target • Targets limited amount of platforms

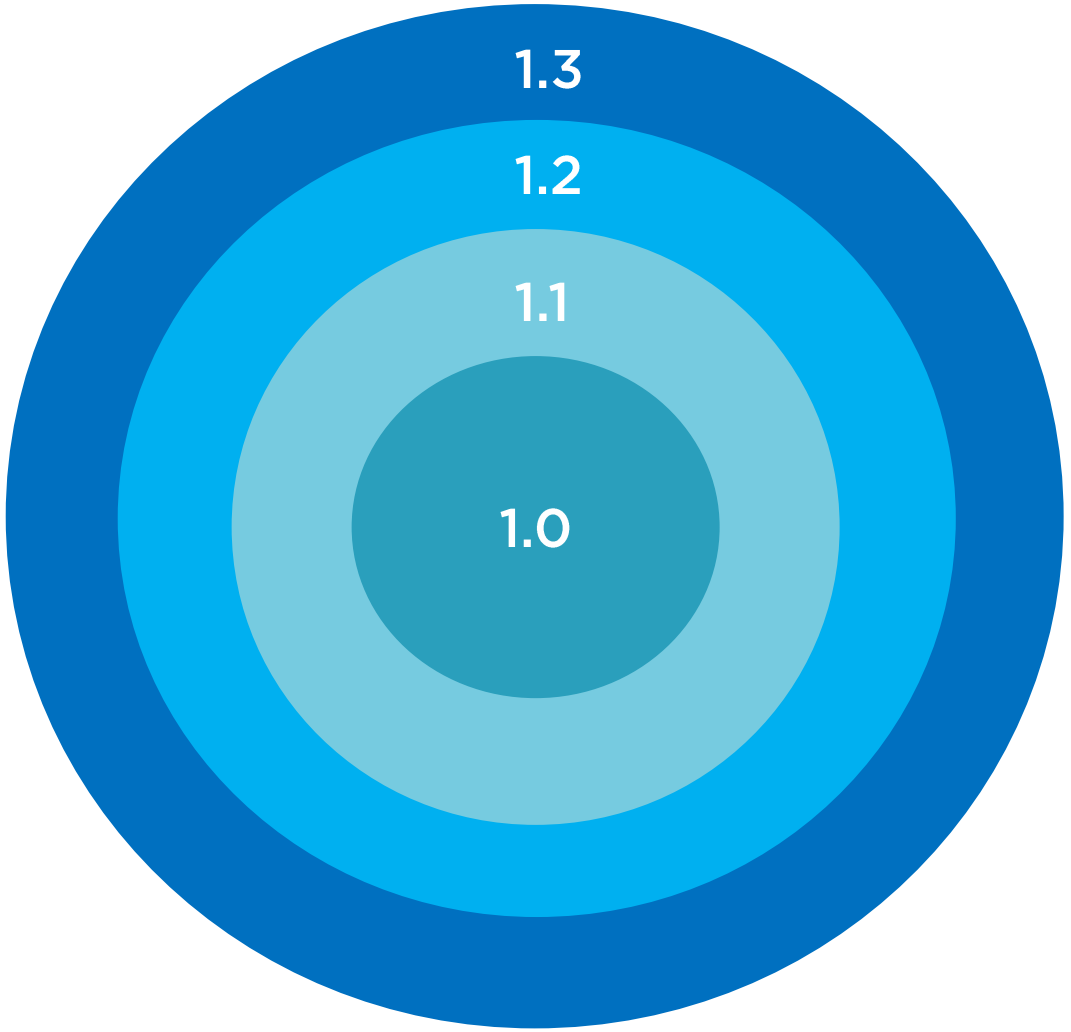
.NET Standard Versioning



.NET Standard Versioning

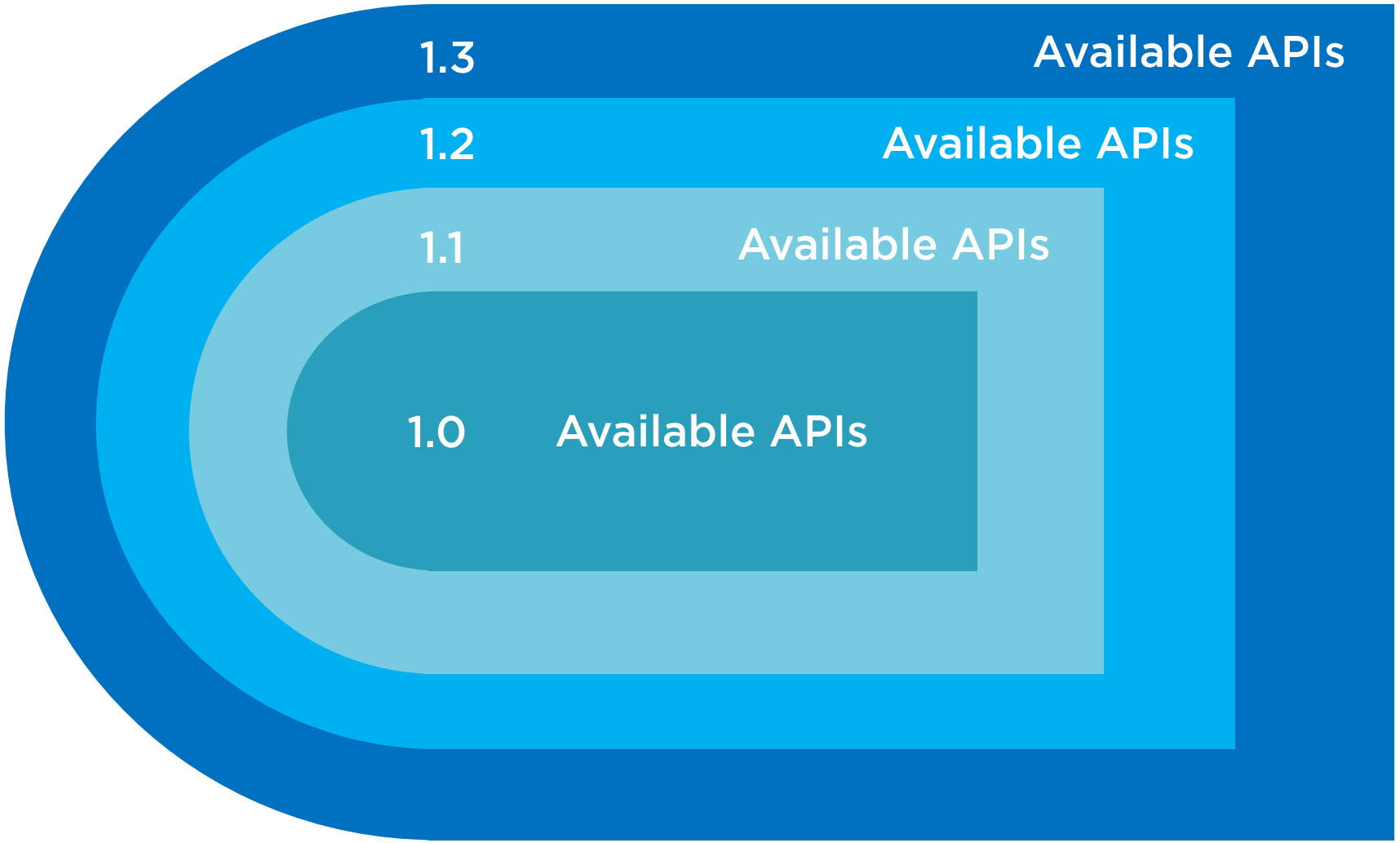


.NET Standard Versioning

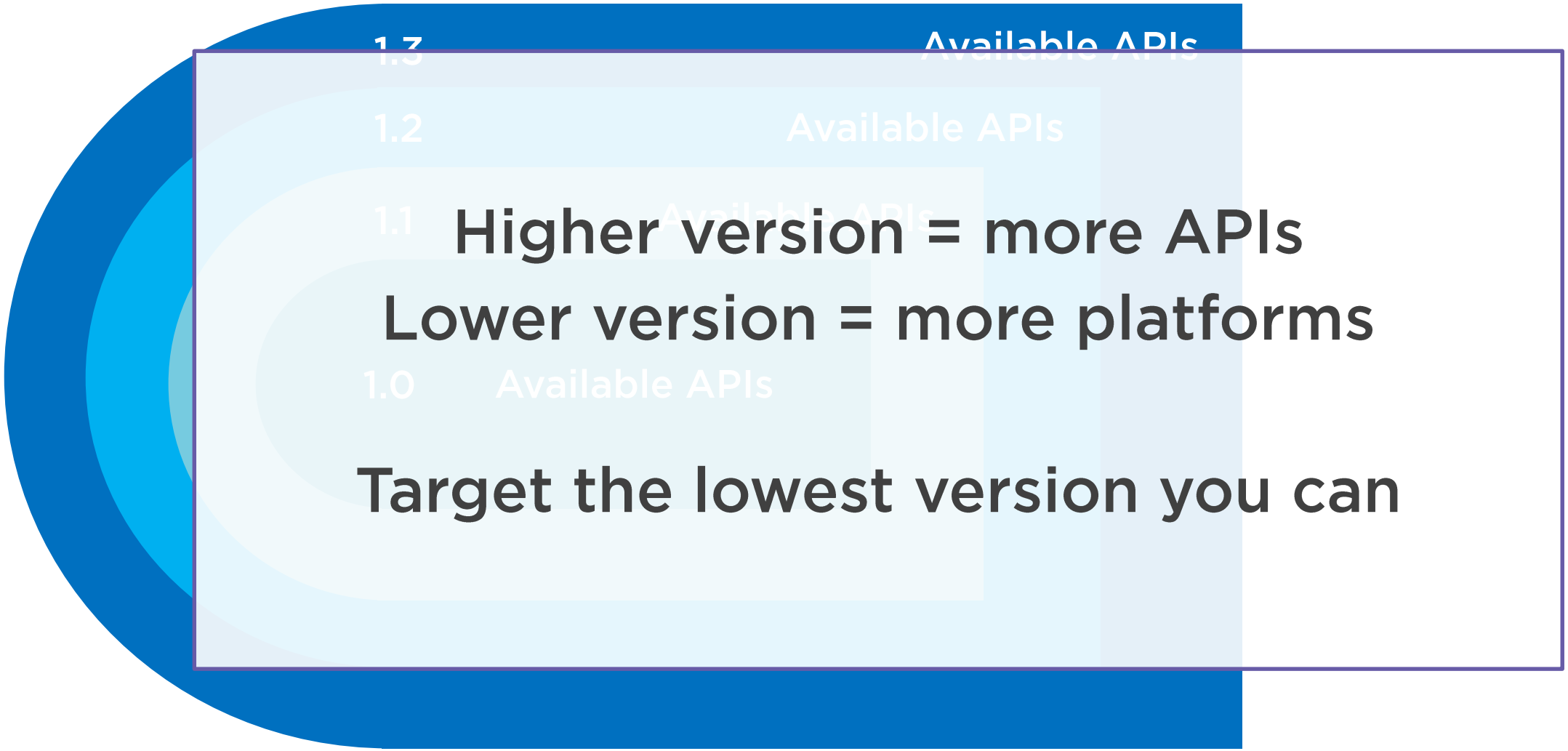


- Each version contains the APIs of the previous versions
- No breaking changes between versions
- Once shipped, versions are frozen
- Specific .NET **runtime** versions implement specific .NET **Standard** versions

.NET Standard Versioning



.NET Standard Versioning



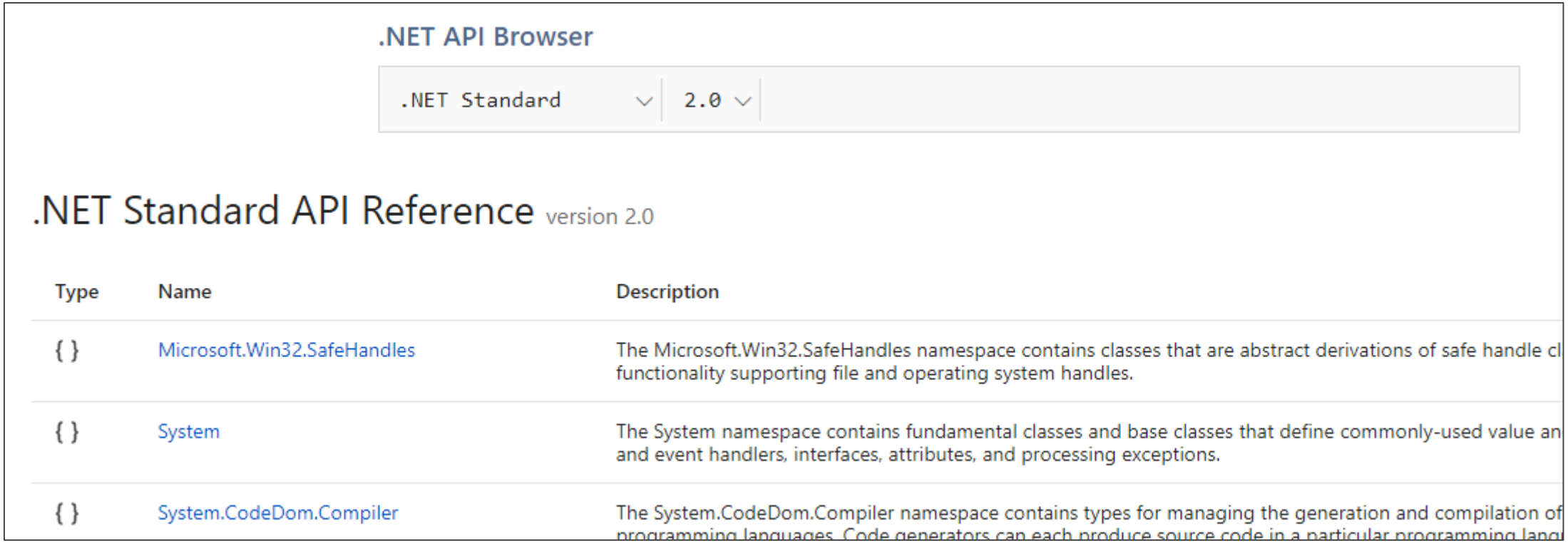
Higher version = more APIs

Lower version = more platforms

Target the lowest version you can

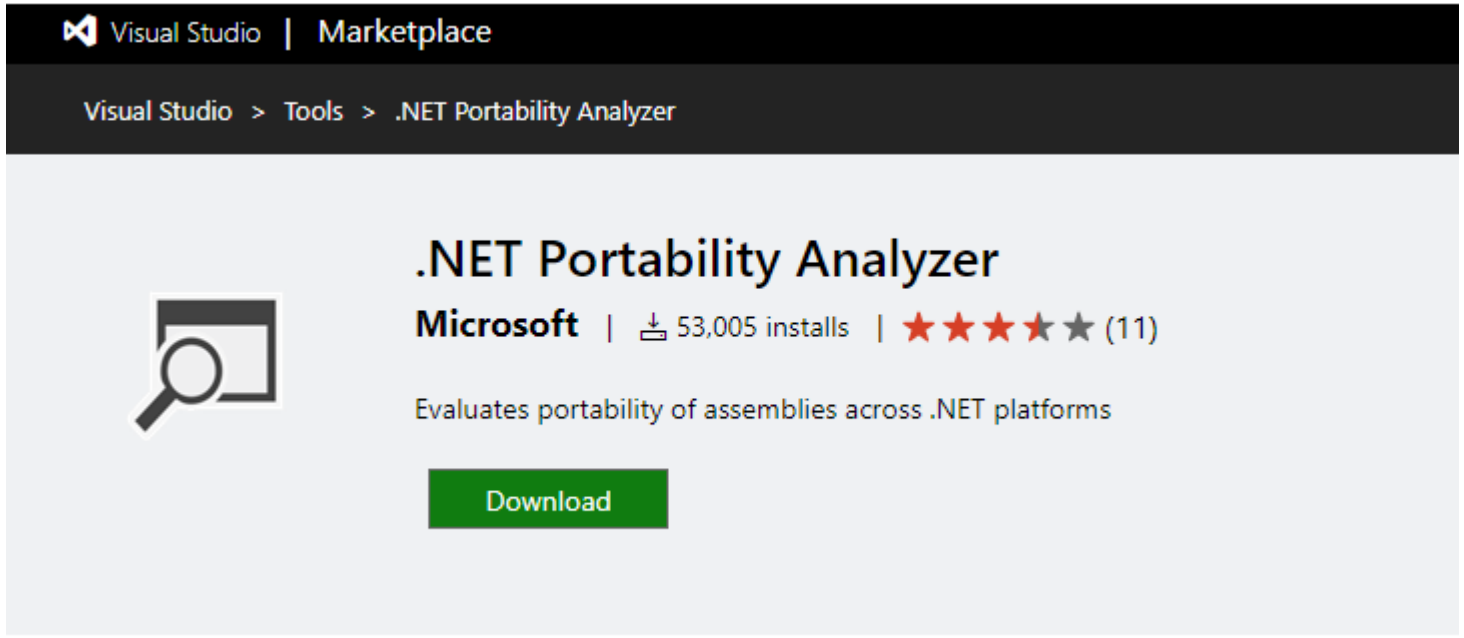
.NET API Browser

https://docs.microsoft.com/en-us/dotnet/api/

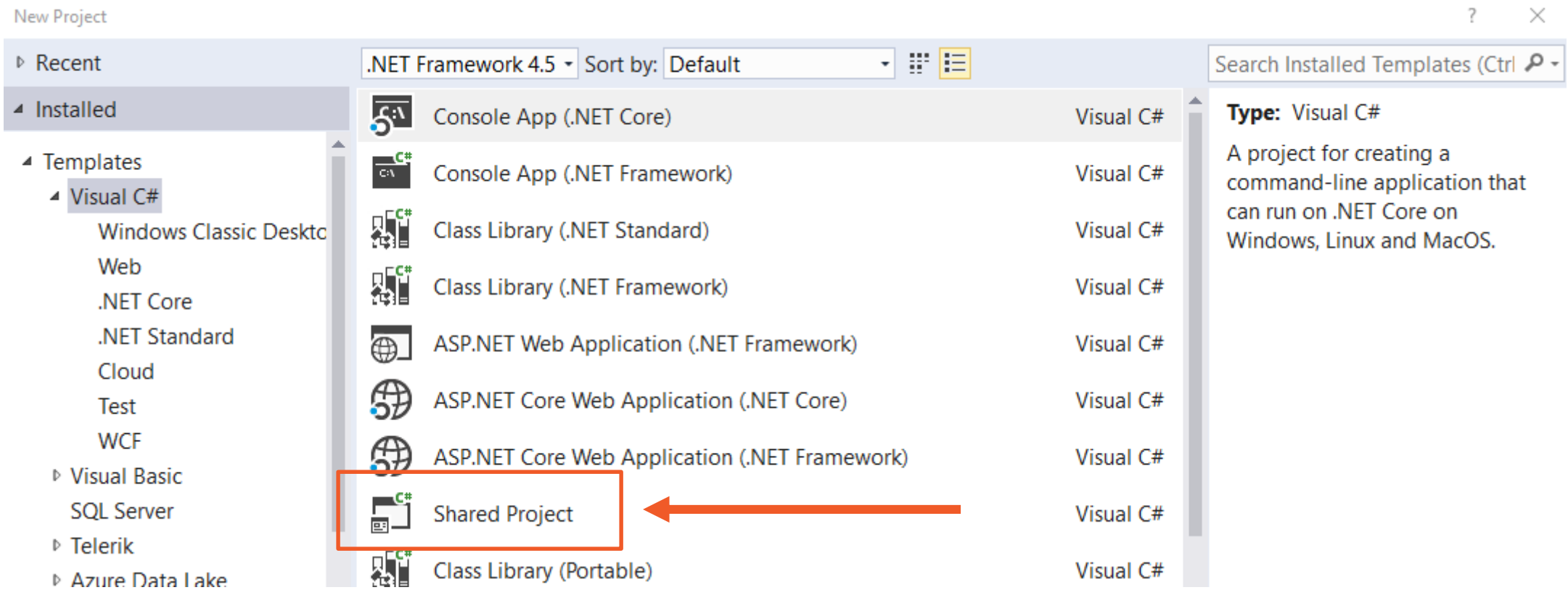


.NET Portability Analyzer

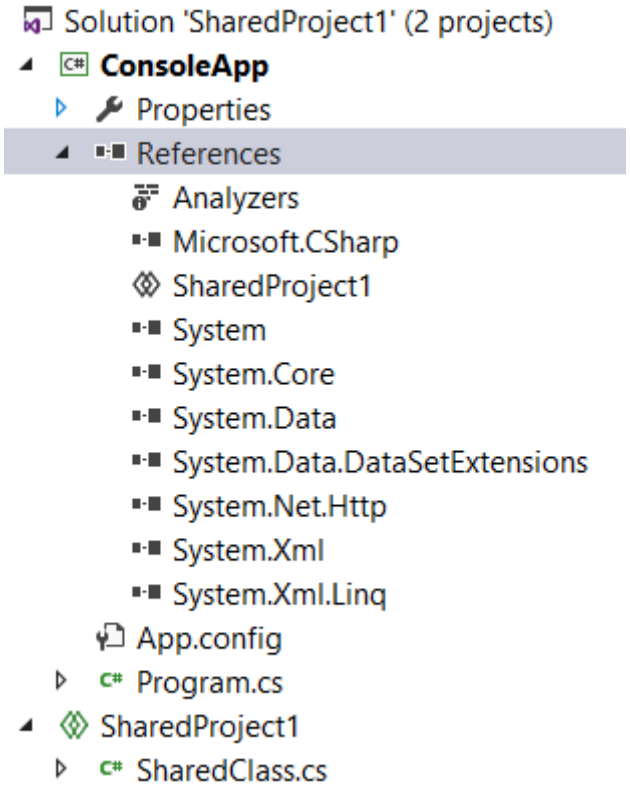
https://marketplace.visualstudio.com/items?itemName=ConnieYau.NETPortabilityAnalyzer



What About Shared Projects?



What About Shared Projects?



- Shared Projects are not compiled references
- They link files into your project
- Do not provide APIs
- Act as file-sharing mechanism between projects

Demo

.NET Standard library

Use it in the different projects

- .NET Core Console application
- Windows Forms application
- Xamarin.Android application

Choosing a direction

(what to use when?)

The .NET Ecosystem

.NET Framework

- Build ‘Forms’ applications
- That cannot run cross-platform
- Anything that you can’t do (yet) with .NET Core or Mono

.NET Core

- Build high performance apps with a small footprint
- That run cross-platform
- And run truly side-by-side

Mono for Xamarin

- Build mobile applications that utilize native features
- That run cross-platform

The .NET Ecosystem

.NET Framework

- Build ‘Forms’ applications
- That cannot run cross-platform
- Anything that you can’t do (yet) with .NET Core or Mono

.NET Core

- **Framework-dependent applications**
- **Self-contained applications**

Mono for Xamarin

- Build mobile applications that utilize native features
- That run cross-platform

Framework-dependent vs. Self-contained .NET Core Apps

Framework-dependent

+ No target OS needed
+ Small deployment
- .NET Core needs to be installed
- Different version of .NET Core might break app

Self-contained

+ Control over version, app won't break
+ .NET Core doesn't have to be installed
- Large deployment
- Need to select target OS

Choosing ASP.NET or ASP.NET Core

ASP.NET or ASP.NET Core?

ASP.NET

- Windows only
- Large set of capabilities
- ASP.NET Forms

ASP.NET Core

- Cross-platform
- High-performance
- Self-contained
- Growing set of capabilities
- No ASP.NET Forms

That’s it!


I have seen the light!

Where to Find the Demo Files

https://github.com/bmaluijb/Dot-Net-Ecosystem

Visual Studio 2017

Find me on Pluralsight

- Introduction to Azure App Services
- Building a Global App with Azure PaaS
- Continuous Integration and Continuous Delivery: The Big Picture
- Cloud Design Patterns for Azure: Design and Implementation
- Cloud Design Patterns for Azure: Availability and Resilience
- Cloud Design Patterns for Azure: Data Management and Performance
- The .NET Ecosystem: The Big Picture
- Microsoft Azure for Developers: What to Use When?
- Microsoft Azure Cognitive Services: The Big Picture