.NET Standard

Reuse All the Code

Jonathan "J." Tower

Hi, I'm J.

Jonathan "J." Tower
Principal Consultant & Partner
Trailhead Technology Partners

🏆 Microsoft MVP in ASP.NET

🏆 Telerik/Progress Developer Expert

🍺 Organizer of Beer City Code

TRAILHEAD
TECHNOLOGY PARTNERS

trailheadtechnology.com

✉ jtower@trailheadtechnology.com

🌐 trailheadtechnology.com/blog

🐦 jtowermi

github.com/jonathantower/dotnet-standard

If You Give $245, So Will I!

bit.ly/gang-conf

"charity:water is a non-profit organization that provides clean and safe drinking water to people in developing nations. The organization was founded in 2006 and has helped fund 22,936 projects in 24 countries, benefiting over 4.6 million people." - Wikipedia

"4/4 Stars"
- CharityNavigator.org

charity: water

Overview

What is the .NET Standard?

Why should I care?
  DEMO: How to read Nuget listing
  DEMO: Creating a .NET Standard library from scratch

Compare/Contrast with Portable Class Libraries (PCLs)

DEMO: Converting a PCL to .NET Standard

.NET Standard v2.0 & v2.1

What is .NET Standard?

The .NET Standard is a <u>formal specification</u> of .NET APIs that are intended to be available on all .NET runtimes

.NET Standard is a specification, not an implementation

Specification in code

Goals of .NET Standard

Uniform set of APIs on all .NET platforms

Portable libraries across platforms

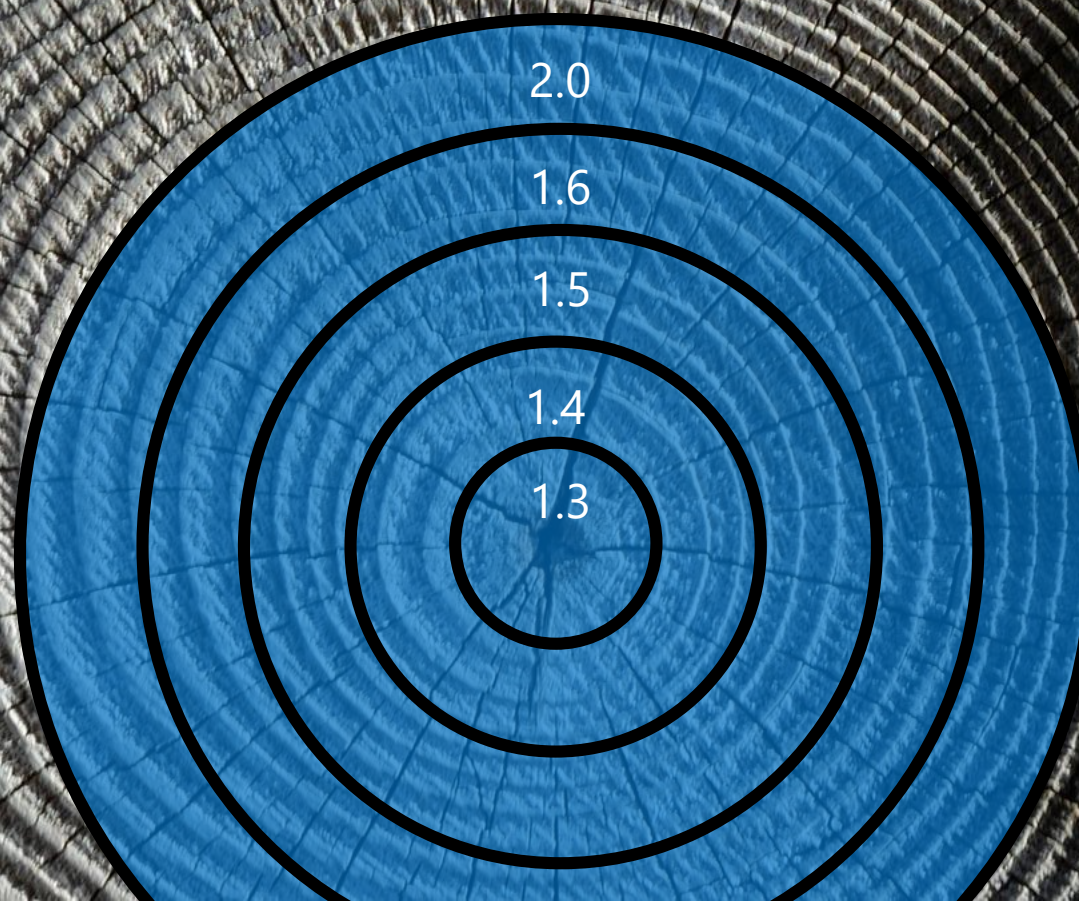Reduce/eliminate conditional compilation

The Standards Gap

A standard exists for .NET runtime implementations (ECMA 335)

No standard existed for .NET Base Class Library (BCL) implementations

.NET Standard Versioning:
Additive

2.0
1.6
1.5
1.4
1.3

.NET Standard Versioning: Immutable

.NET Standard	1.0	1.1	1.2	1.3	1.4	1.5	1.6	2.0	2.1
.NET Core	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	
.NET Framework	4.5	4.5	4.5.1	4.6	4.6.1	4.6.1	4.6.1	4.6.1	
Mono	4.6	4.6	4.6	4.6	4.6	4.6	4.6	5.4	
Xamarin.iOS	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.14	
Xamarin.Mac	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.8	
Xamarin.Android	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	
Unity	2018.1	2018.1	2018.1	2018.1	2018.1	2018.1	2018.1	2018.1	
Universal Windows Platform	10.0	10.0	10.0	10.0	10.0	10.0.16299	10.0.16299	10.0.16299	
Windows	8.0	8.0	8.1						
Windows Phone	8.1	8.1	8.1						
Windows Phone Silverlight	8.0								

https://github.com/dotnet/standard/blob/master/docs/versions.md

What's Included?

BCLs but **NOT** FCLs

David Fowler's "Developer Analogy"

```csharp
// .NET Standard
interface INetStandard10
{
    void Primitives();
    void Reflection();
    void Tasks();
    void Collections();
    void Linq();
}
```

```csharp
interface INetStandard11 :
        INetStandard10
{
    void ConcurrentCollections();
    void InteropServices();
}

interface INetStandard12 :
        INetStandard11
{
    void ThreadingTimer();
}

...
```

David Fowler's "Developer Analogy"

```csharp
// .NET Framework
interface INetFramework45 : INetStandard11
{
    void FileSystem();
    void Console();
    void ThreadPool();
    void Crypto();
    void WebSockets();
    void Process();
    void Sockets();
    void AppDomain();
    void Xml();
    void Drawing();
    void SystemWeb();
    void WPF();
    void WindowsForms();
    void WCF();
}

// Mono
interface IMono43 : INetFramework46
{
    void MonoSpecificApi();
}
```

```csharp
// Windows Universal Platform
interface IWindowsUniversalPlatform : INetStandard14
{
    void GPS();
    void Xaml();
}

// Xamarin
interface IXamarinIOS : INetStandard15
{
    void AppleAPIs();
}

interface IXamarinAndroid : INetStandard15
{
    void GoogleAPIs();
}

// .NET Core
interface INetCoreApp10 : INetStandard15 { }
```

David Fowler's "Developer Analogy"

```
class Example
{
    public void FuturePlatformApplication(ISomeFuturePlatform platform)
    {
        JsonNet(platform); //JSON.NET supports .NET Standard 1.0
    }

    public void JsonNet(INetStandard10 platform)
    {
        platform.Linq();
        platform.Reflection();
        platform.Collections();
    }
}
```

David Fowler's "Developer Analogy"

MORE: http://bit.ly/dev-analogy

Why Should I Care?

Two Reasons You Will Care

Creating x-platform libraries

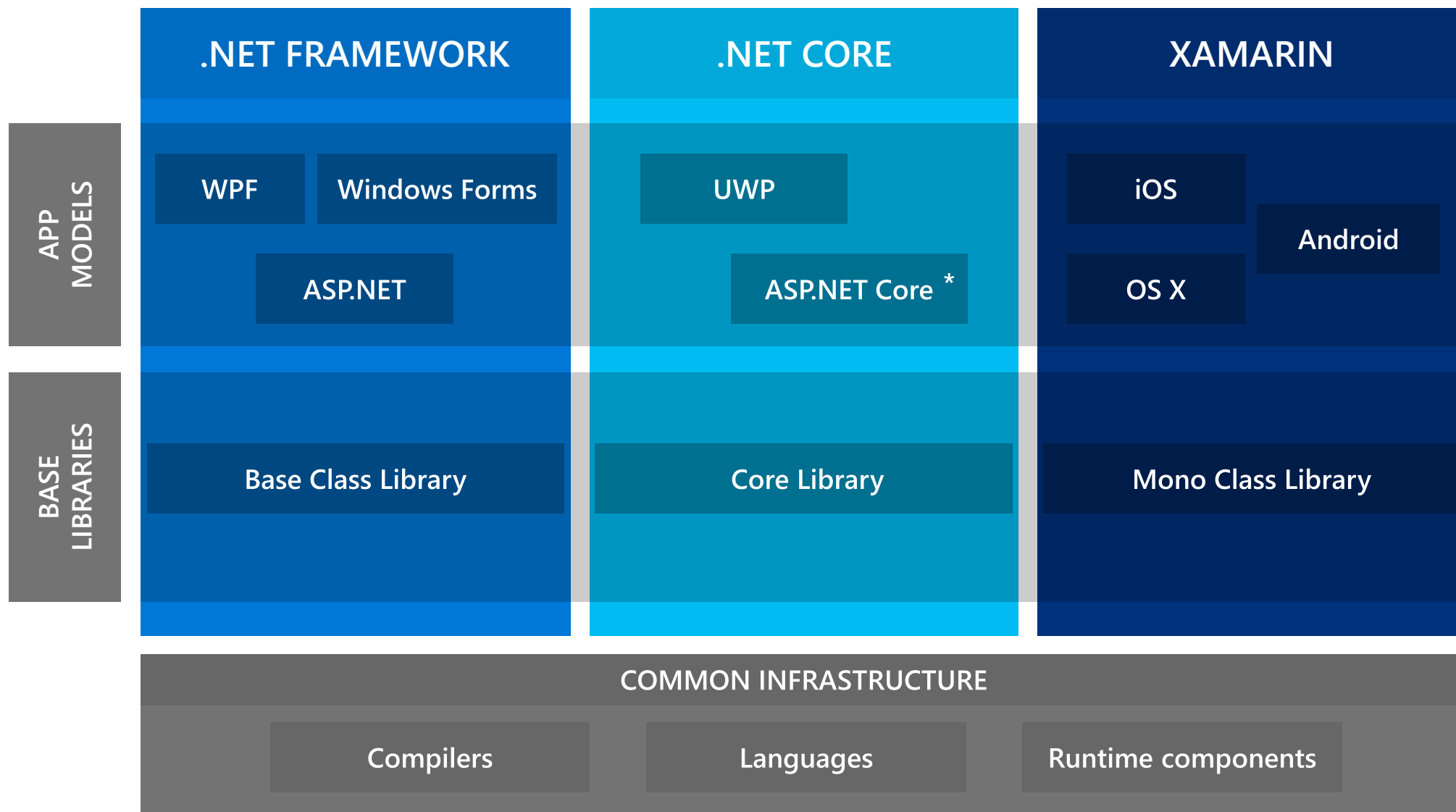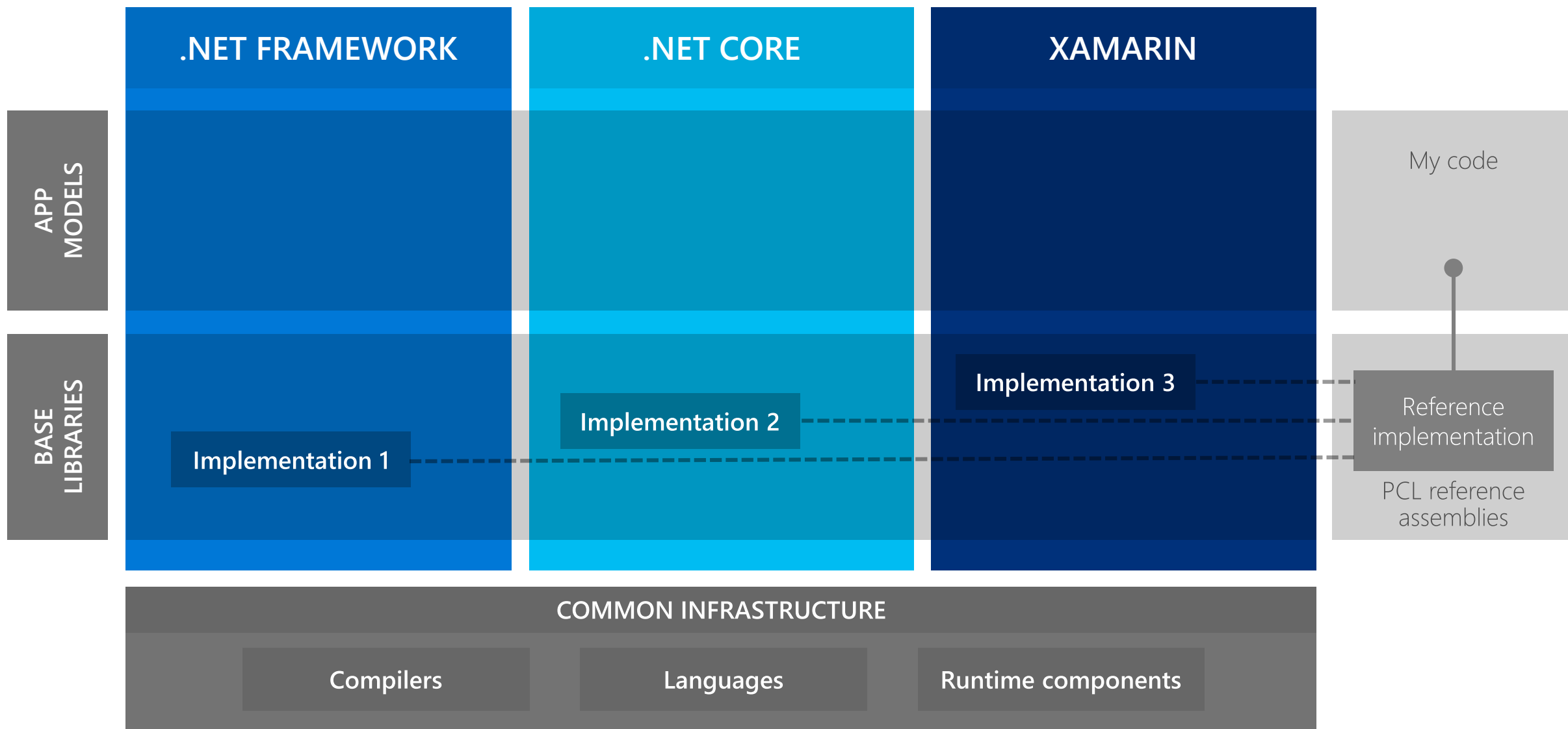*Consuming* others' x-platform libraries

DEMO

DEMO: Let's Try It

Create a .NET Standard library

.NET Standard vs Other Ways to Share

Portable Class Libraries (PCLs)

APP MODELS	.NET FRAMEWORK	.NET CORE	XAMARIN
	WPF Windows Forms	UWP	iOS Android
	ASP.NET	ASP.NET Core *	OS X
BASE LIBRARIES	Base Class Library	Core Library	Mono Class Library

COMMON INFRASTRUCTURE
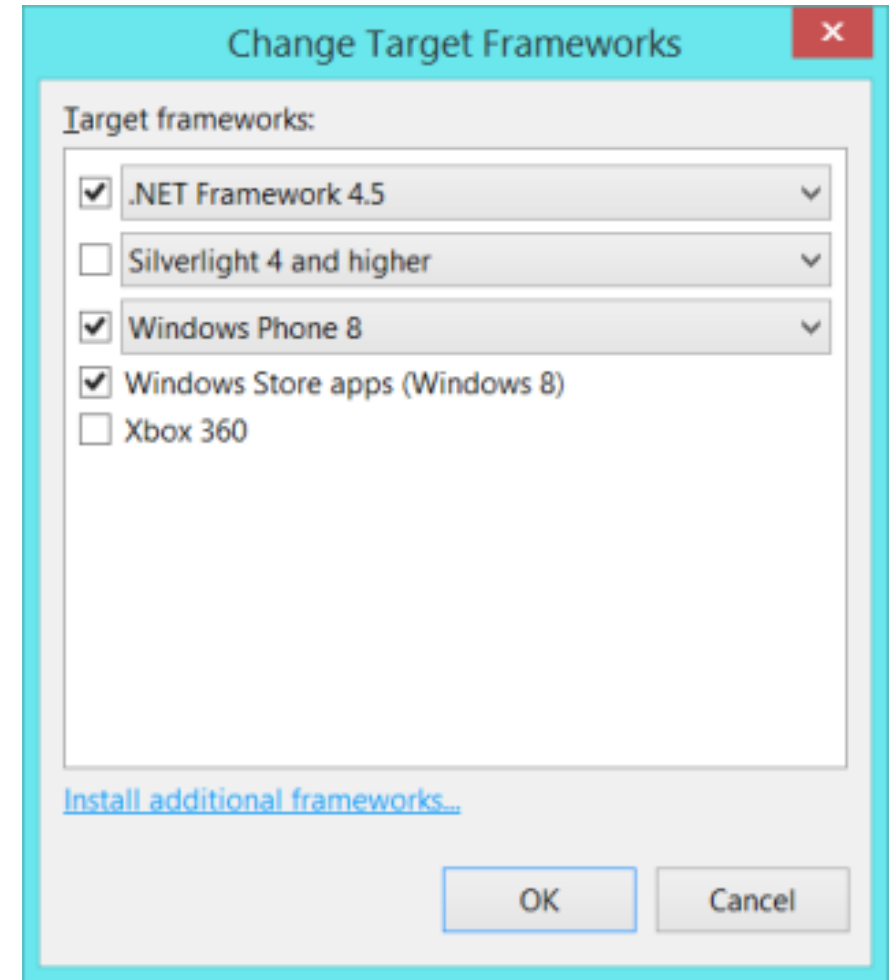
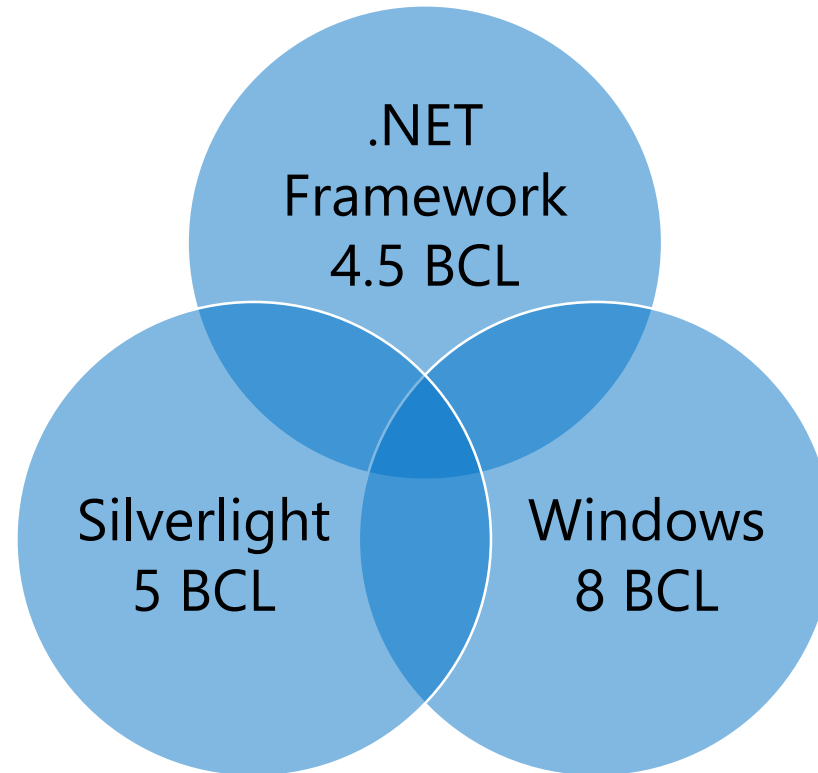| Compilers | Languages | Runtime components |

Problems with PCLs
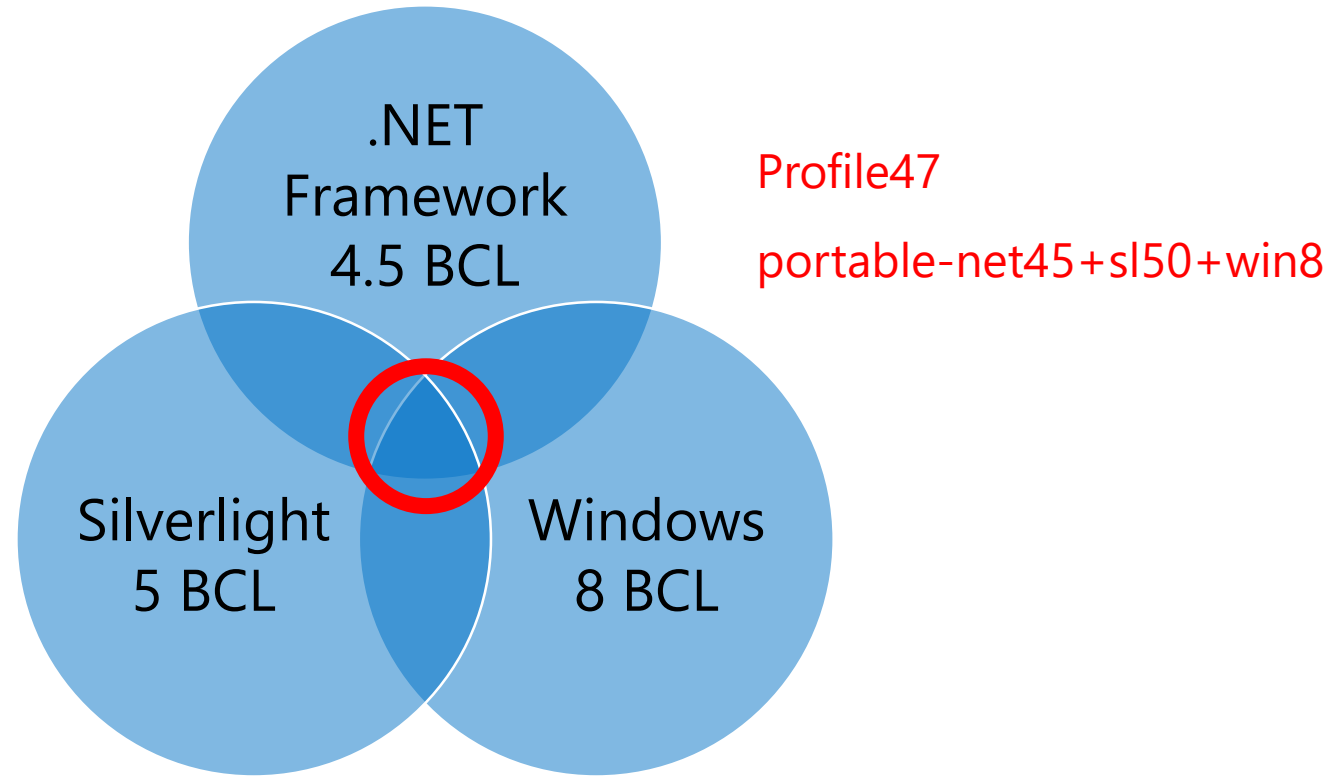
Sheer number of permutations

Intersection of APIs, not union

PCLs: Intersection of APIs

PCLs: Intersection of APIs

Other Ways to Share Code

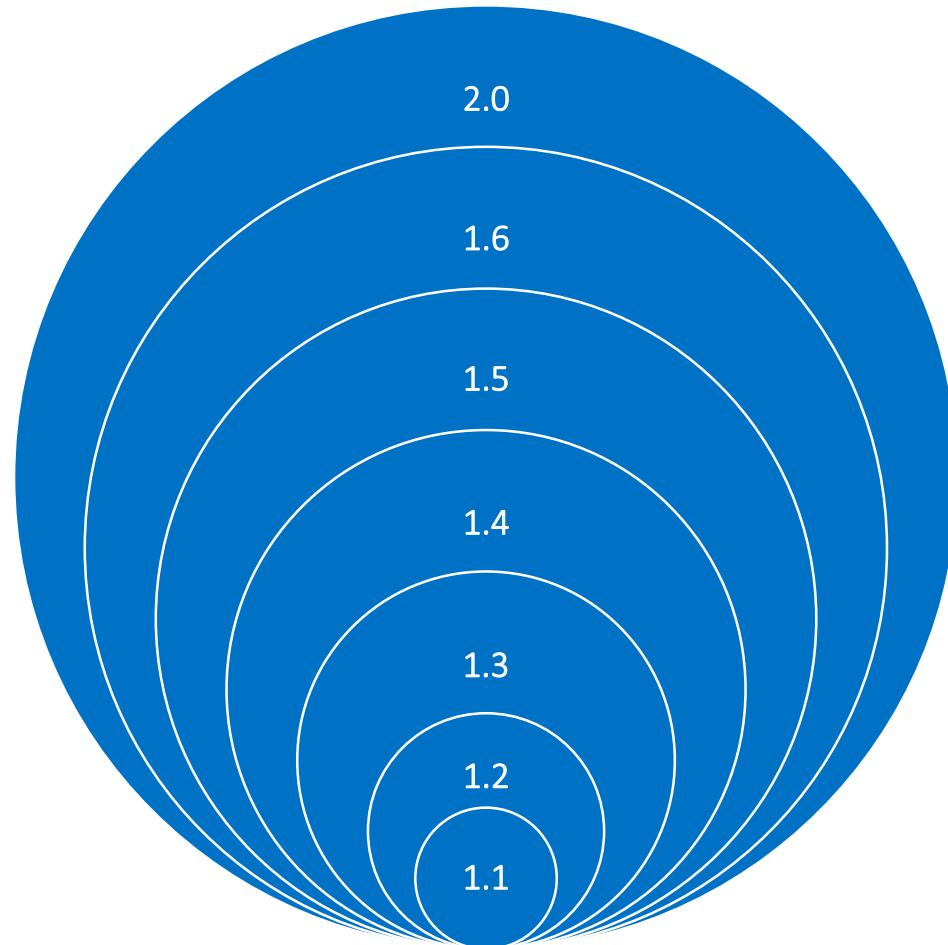Completely different implementation per target

Linked files in source control

#if defs

.NET Standard

.NET Standard: Curated Set of APIs

More APIs / Fewer Platforms

.NET FRAMEWORK

WPF | Windows Forms

ASP.NET

.NET CORE

UWP

ASP.NET Core*

XAMARIN

iOS

OS X | Android

.NET STANDARD LIBRARY
One library to rule them all

COMMON INFRASTRUCTURE

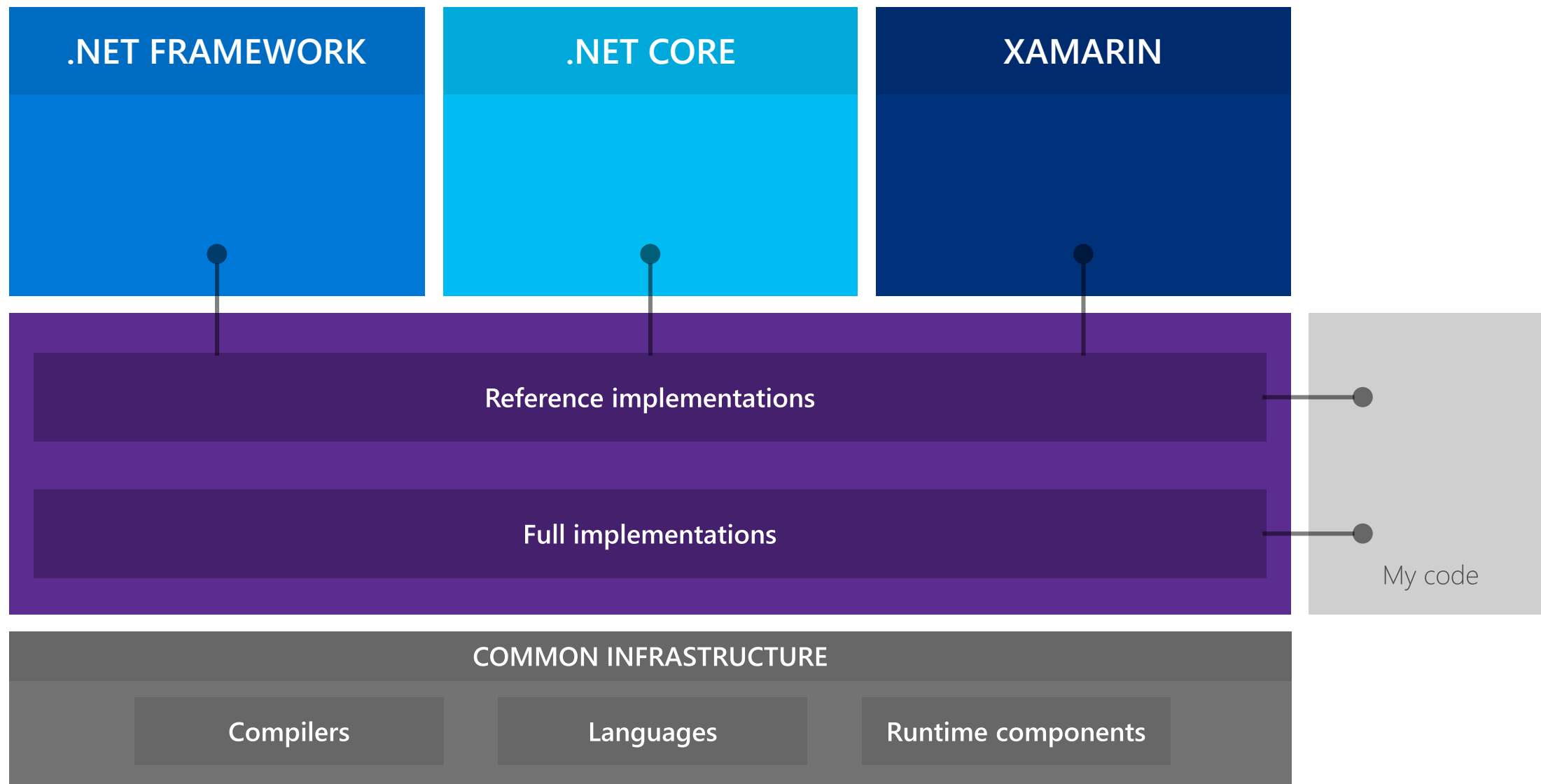Compilers | Languages | Runtime components
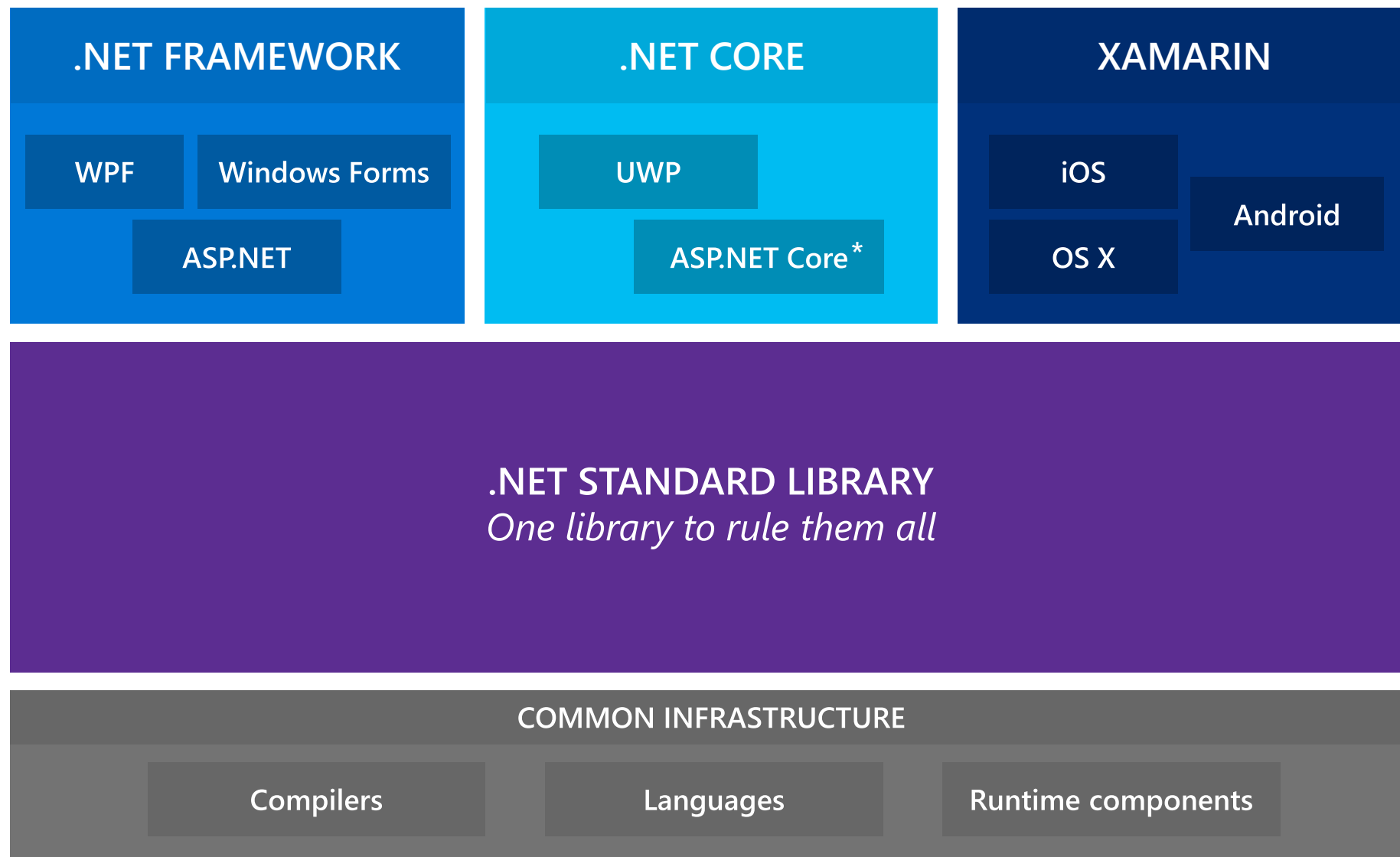
Advantages of .NET Standard

For Developers

- Same API's available on all platforms
- Master one library not a platform
- Big surface area—no small common denominator

For Platforms

- .NET Standard can grow without updating platforms
- Fast innovation on libraries, common infrastructure, and tools

Comparing PCLs to .NET Standard

Comparison to Portable Class Libraries

.NET Standard: sort of the next generation of PCLs

"Standard-based PCL" vs "profile-based PCL"

Soft reference through a go-between vs hard-coded reference

Similarities to Portable Class Libraries

Allow sharing library across platforms

Define APIs shared across different platforms

Differences from Portable Class Libraries

.NET Standard	PCLs
A curated set of APIs	Profiles are defined by intersections of existing platforms
Linearly versions	PCL profiles do not linearly version.
Agnostic to platform	PCL profiles represents Microsoft platforms

PCL Profile	.NET Standard	PCL Platforms
Profile7	1.1	.NET Framework 4.5, Windows 8
Profile31	1.0	Windows 8.1, Windows Phone Silverlight 8.1
Profile32	1.2	Windows 8.1, Windows Phone 8.1
Profile44	1.2	.NET Framework 4.5.1, Windows 8.1
Profile49	1.0	.NET Framework 4.5, Windows Phone Silverlight 8
Profile78	1.0	.NET Framework 4.5, Windows 8, Windows Phone Silverlight 8
Profile84	1.0	Windows Phone 8.1, Windows Phone Silverlight 8.1
Profile111	1.1	.NET Framework 4.5, Windows 8, Windows Phone 8.1
Profile151	1.2	.NET Framework 4.5.1, Windows 8.1, Windows Phone 8.1
Profile157	1.0	Windows 8.1, Windows Phone 8.1, Windows Phone Silverlight 8.1
Profile259	1.0	.NET Framework 4.5, Windows 8, Windows Phone 8.1, Windows Phone Silverlight 8

PCL Compatibility

Enable .NET Standard-based PCLs to reference profile-based PCLs.

Enable profile-based PCLs to be packaged as .NET Standard-based PCLs.

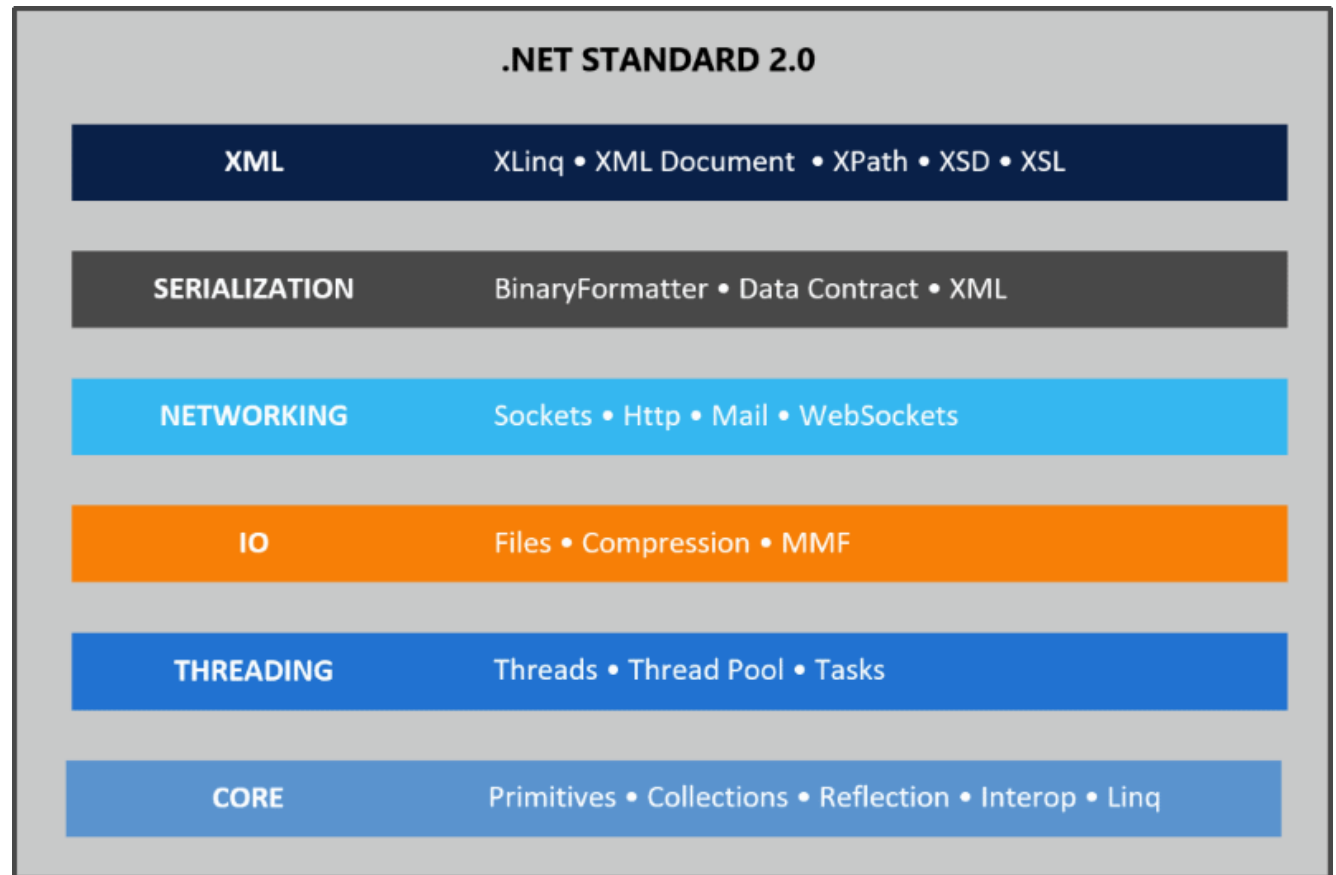# DEMO: PCL Conversion

Convert a PCL to .NET Standard

.NET Standard 2.0

.NET Standard 1.6 -> 2.0

Huge!

20k new APIs

(or 149% increase)

.NET STANDARD 2.0

XML	XLinq • XML Document • XPath • XSD • XSL
SERIALIZATION	BinaryFormatter • Data Contract • XML
NETWORKING	Sockets • Http • Mail • WebSockets
IO	Files • Compression • MMF
THREADING	Threads • Thread Pool • Tasks
CORE	Primitives • Collections • Reflection • Interop • Linq

.NET Standard 2.1

.NET Standard 2.1

3k new APIs

800 new members in .NET Core

Span<T> and foundational-APIs working with spans

Reflection emit

SIMD

ValueTask and ValueTask<T>

DbProviderFactories

System.HashCode

System.String overloads

.NET Standard 2.1

Currently 83% complete

Released with .NET Core 3.0 in 2019

.NET Standard 2.1 Support
   .NET Core 3.0
   Future versions of Xamarin, Mono, and Unity
   Note: .NET Framework 4.8 remains on .NET Standard 2.0

Recap

What is the .NET Standard?

Why should I care?
   DEMO: How to read Nuget listing
   DEMO: Creating a .NET Standard library from scratch

Compare/Contrast with Portable Class Libraries (PCLs)

DEMO: Converting a PCL to .NET Standard

.NET Standard v2.0 & v2.1

Thank You! Questions?

Jonathan "J." Tower
Principal Consultant & Partner
Trailhead Technology Partners

TRAILHEAD TECHNOLOGY PARTNERS

trailheadtechnology.com

🏆 Microsoft MVP in ASP.NET

🏆 Telerik/Progress Developer Expert

🍺 Organizer of Beer City Code

✉ jtower@trailheadtechnology.com

🌐 trailheadtechnology.com/blog

🐦 jtowermi

github.com/jonathantower/dotnet-standard

If You Give $245, So Will I!

bit.ly/gang-conf

"charity:water is a non-profit organization that provides clean and safe drinking water to people in developing nations. The organization was founded in 2006 and has helped fund 22,936 projects in 24 countries, benefiting over 4.6 million people." - Wikipedia

"4/4 Stars"
- CharityNavigator.org

charity: water