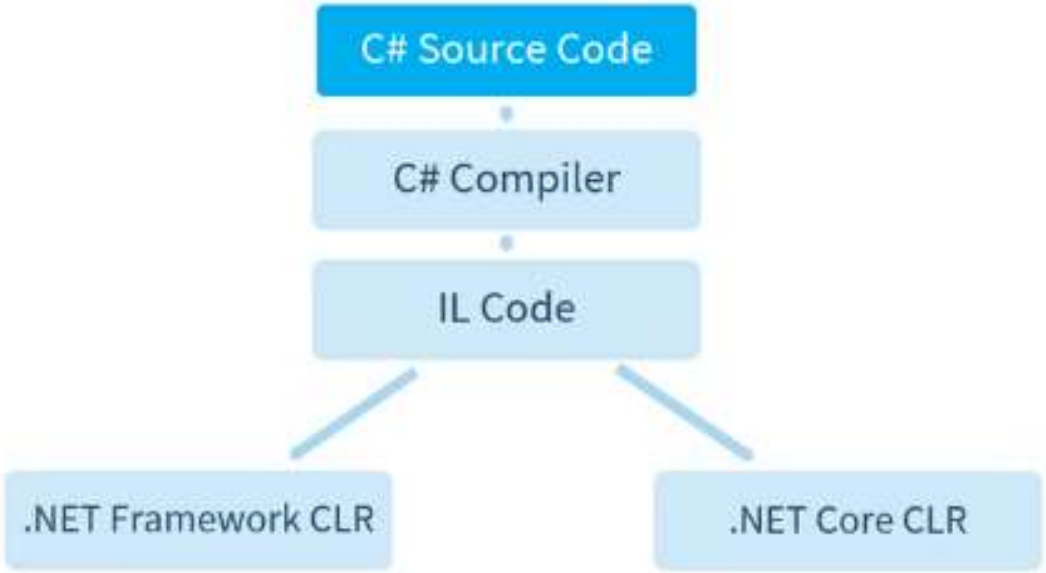
WHY DID .NET CORE NEED TO EXIST?

1. The .NET Framework was full of patches and workarounds that slowed it down.
2. The .NET Framework was tied directly into Windows.
3. The .NET Framework was designed to house everything you needed in one package.
4. The .NET Framework was designed originally over 20 years ago when the Internet was a very different place.

THE COMMON LANGUAGE RUNTIME (CLR)



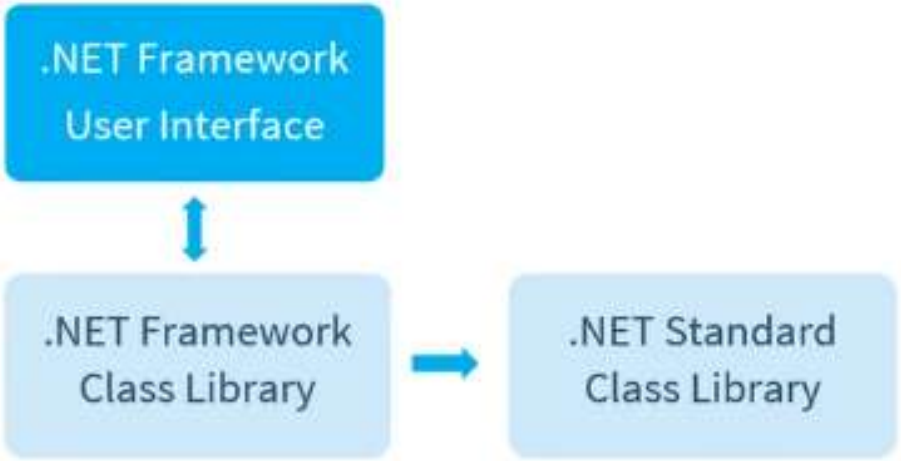
BENEFITS OF .NET CORE OVER THE .NET FRAMEWORK

- Works on most platforms – Windows, Mac, Linux*, Android, iOS, Xbox, IoT
- Faster
- Modular
- More features
- Follows best practices
- Open-source
- Modern
- Compliant with industry standards
- Easy to upgrade

* Linux desktop available using third-party, open-source tools like Uno and Avalonia

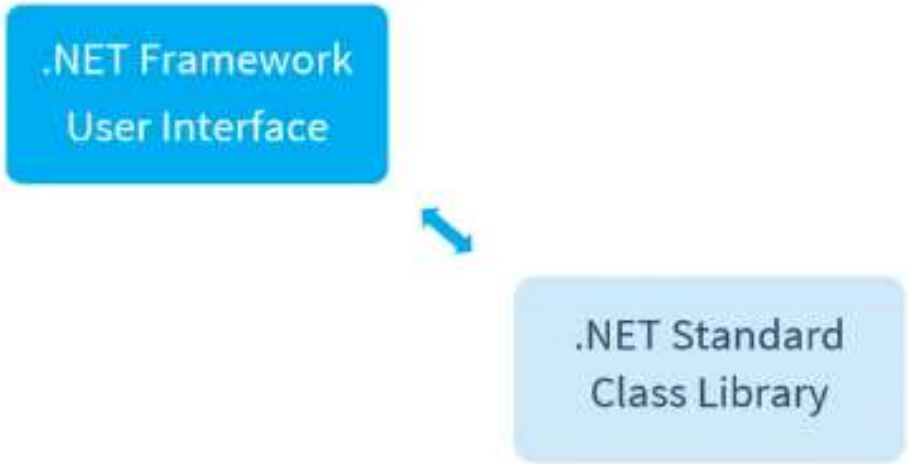
.NET STANDARD EXPLAINED

A common interface of APIs for any .NET implementation



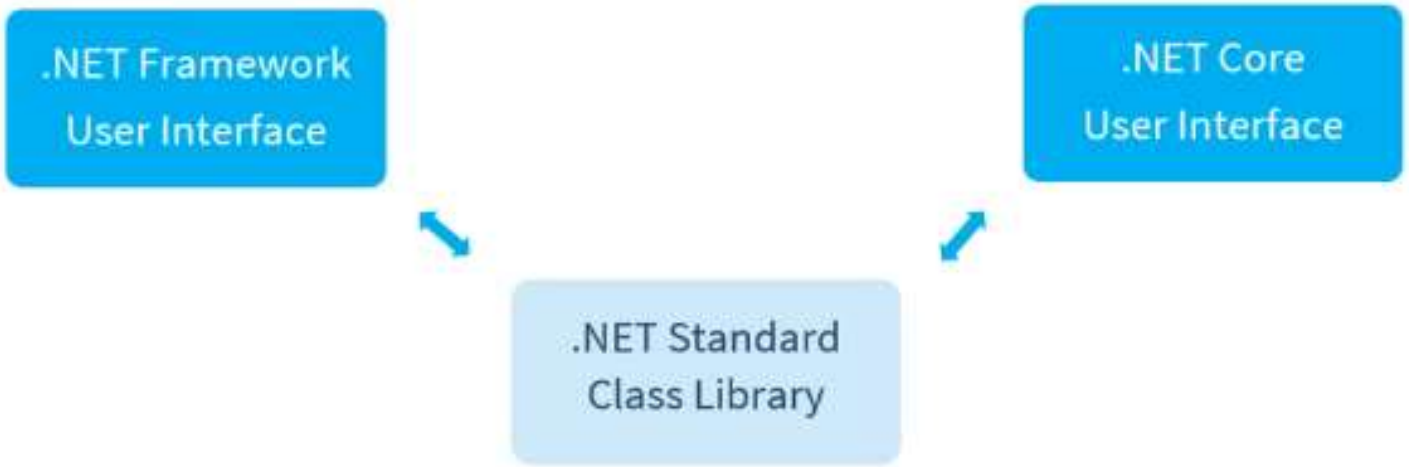
.NET STANDARD EXPLAINED

A common interface of APIs for any .NET implementation



.NET STANDARD EXPLAINED

A common interface of APIs for any .NET implementation

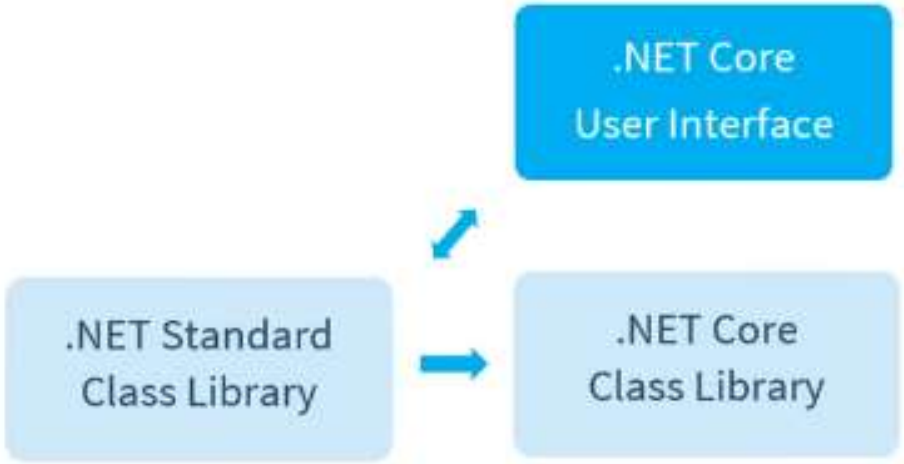


.NET STANDARD EXPLAINED

A common interface of APIs for any .NET implementation

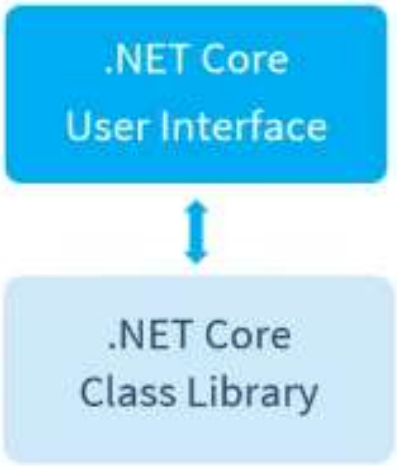
.NET STANDARD EXPLAINED

A common interface of APIs for any .NET implementation



.NET STANDARD EXPLAINED

A common interface of APIs for any .NET implementation

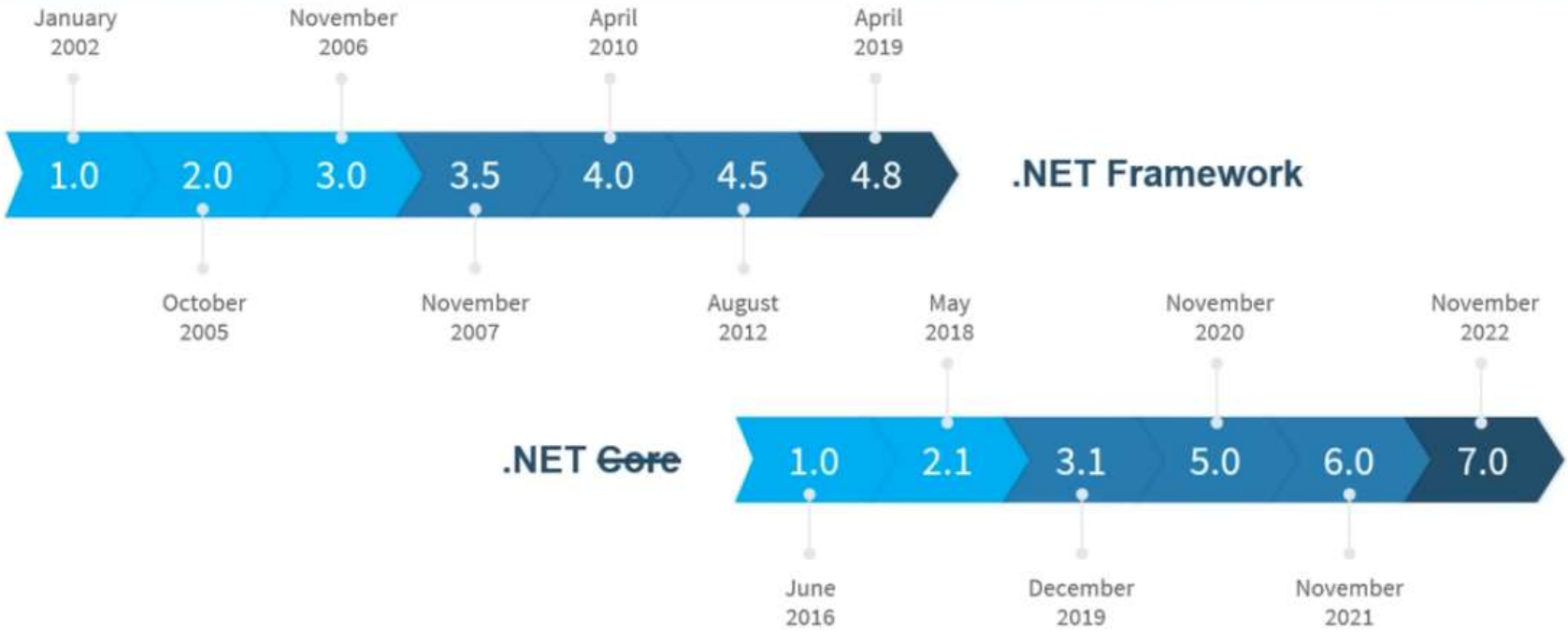


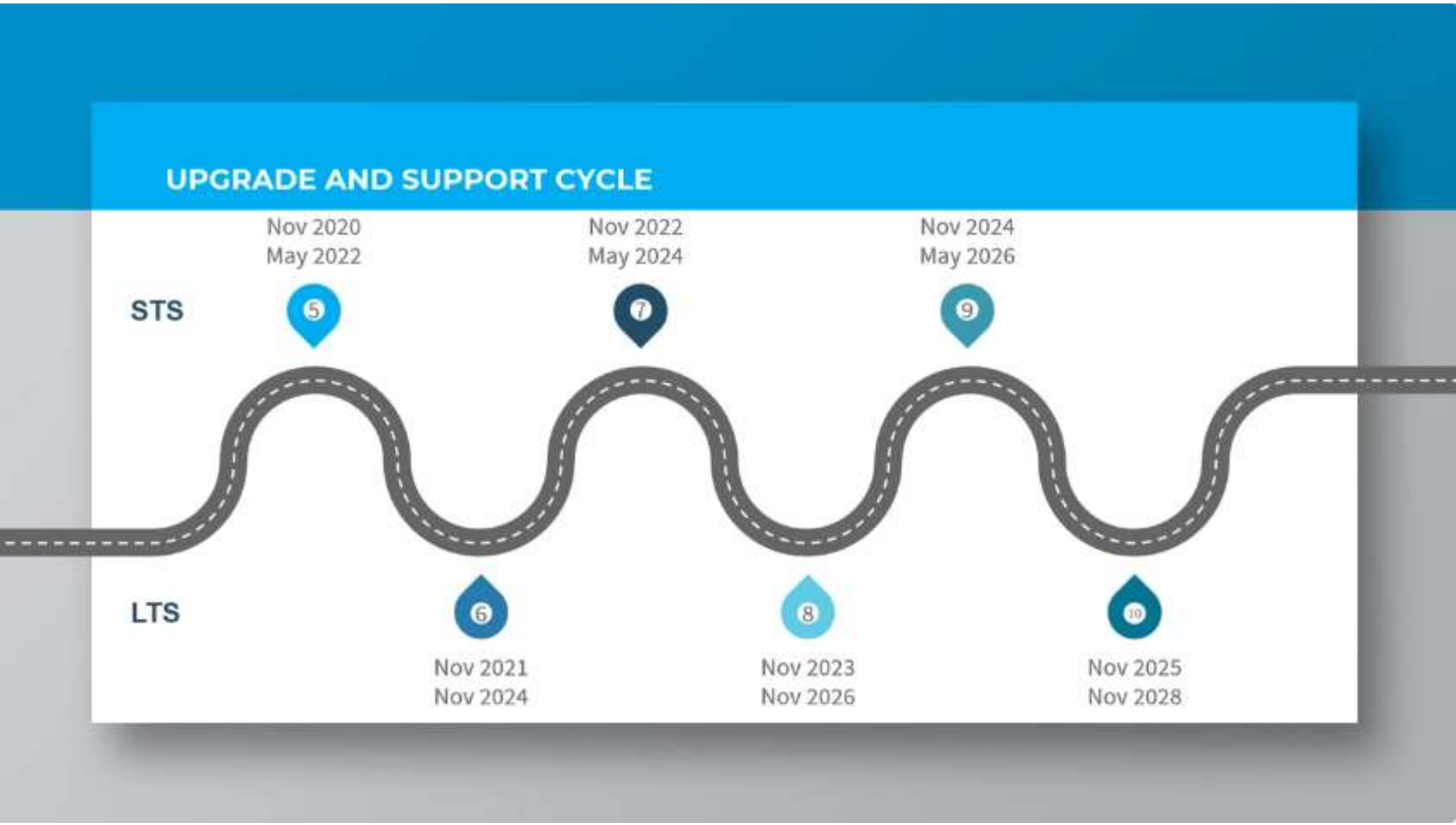
.NET STANDARD VERSIONS

- V1.0 – 7,949 out of 37,118 APIs
  - .NET Framework 4.5 & higher
  - .NET Core 1 & higher
- V2.0 – 32,638 out of 37,118 APIs
  - .NET Framework 4.7.2 & higher
  - .NET Core 2 & higher
- V2.1 – 37,118 out of 37,118 APIs
  - No .NET Framework support
  - .NET Core 3.0 & higher

* .NET Standard also supports Xamarin, Mono, UWP, Unity, and more

.NET TIMELINES




UPGRADE AND SUPPORT CYCLE
Nov 2020
May 2022
Nov 2022
May 2024
Nov 2024
May 2026
STS
5
7
9
LTS
6
8
10
Nov 2021
Nov 2024
Nov 2023
Nov 2026
Nov 2025
Nov 2028

WHAT ABOUT ASP.NET AND ASP.NET CORE?

ASP.NET

- WebForms
- MVC
- WebAPI

ASP.NET Core*

- Razor Pages
- MVC
- API
- Blazor Server
- Blazor WebAssembly

* ASP.NET Core will not drop the "Core" in order to not be confused with ASP.NET