# .NET CORE DESKTOP REVOLUTION

CREATE BY: SHALOM LEVI

## WHO AM I

## AGENDA DAY 1

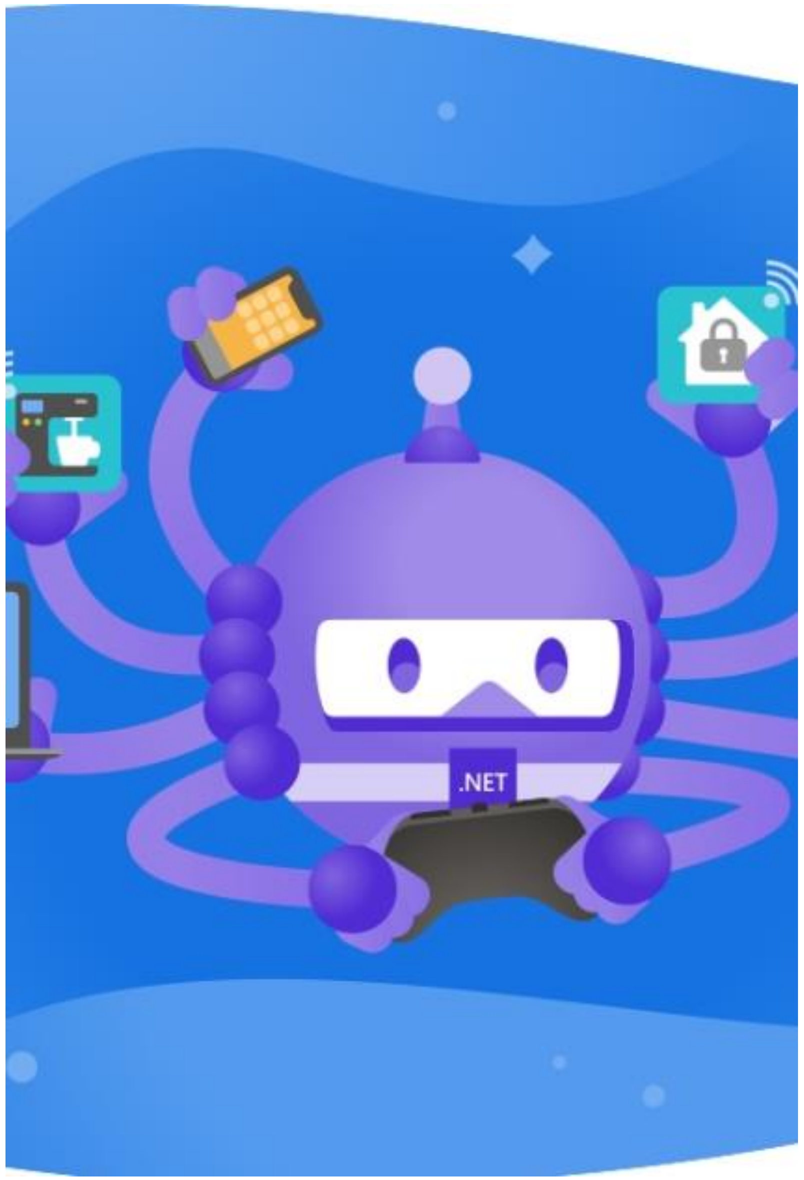
**Part 1:**

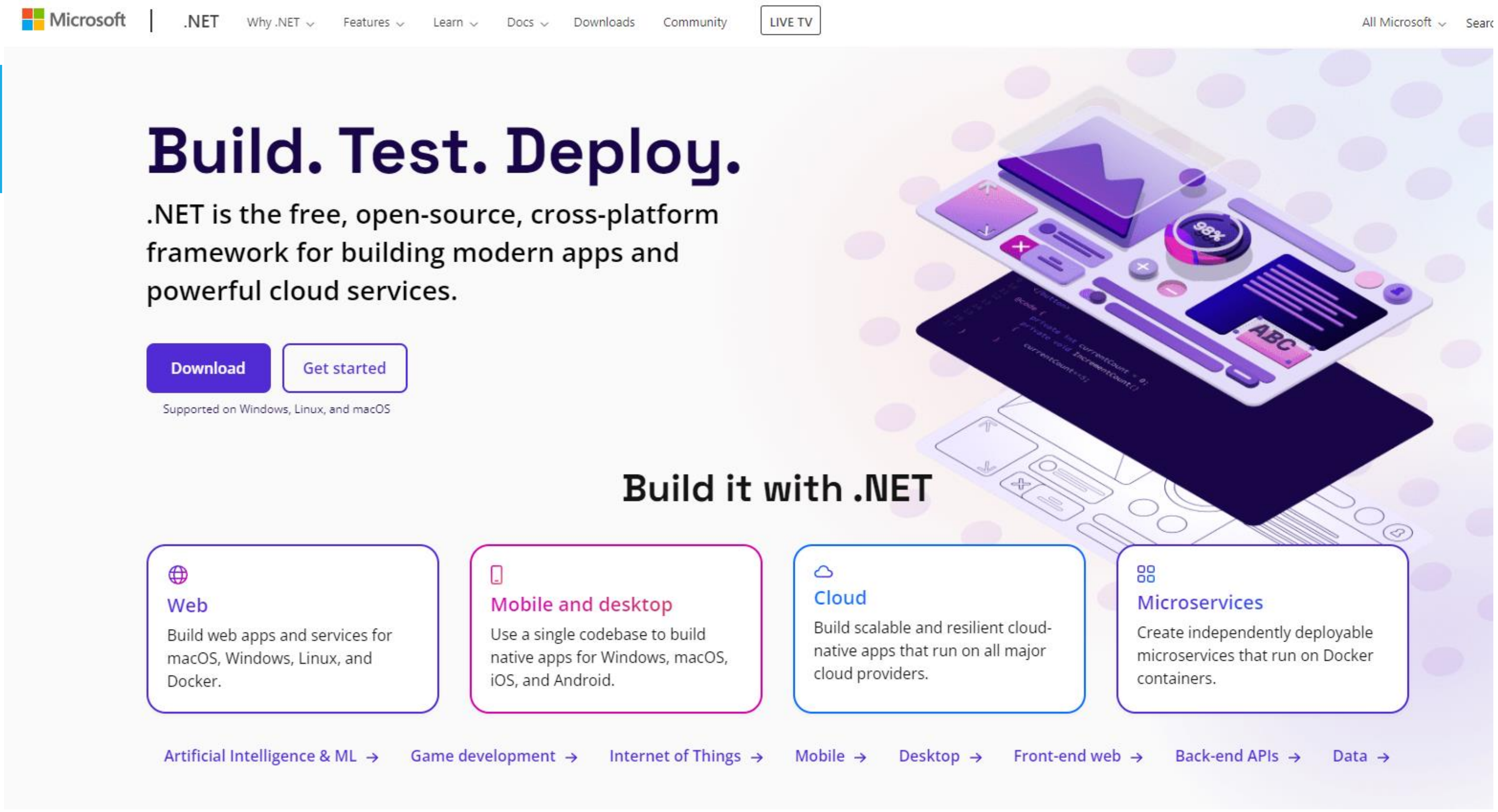
- Overview of .NET Core
- Building Console Applications in .NET Core
- Introduction to .NET Core Command-Line Interface (dotnet CLI)
- Managing Packages with NuGet in .NET Core

**Part 2:**

- Implementing Dependency Injection in .NET Core
- Advance Dependency Injection in .NET Core

## .NET CORE





# Build. Test. Deploy.

.NET is the free, open-source, cross-platform framework for building modern apps and powerful cloud services.

Supported on Windows, Linux, and macOS

## Build it with .NET

**Web**
Build web apps and services for macOS, Windows, Linux, and Docker.

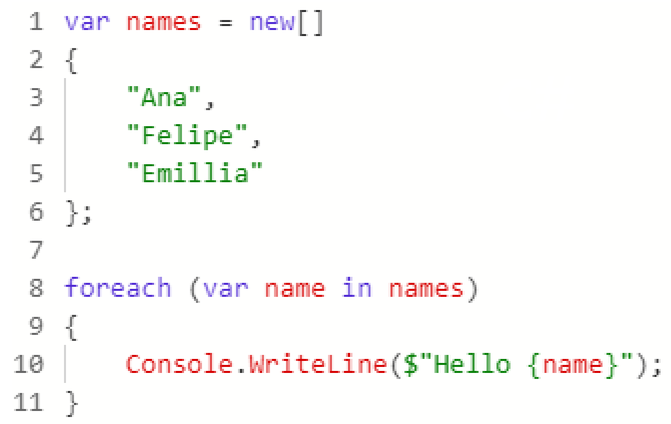
**Mobile and desktop**
Use a single codebase to build native apps for Windows, macOS, iOS, and Android.

**Cloud**
Build scalable and resilient cloud-native apps that run on all major cloud providers.

**Microservices**
Create independently deployable microservices that run on Docker containers.

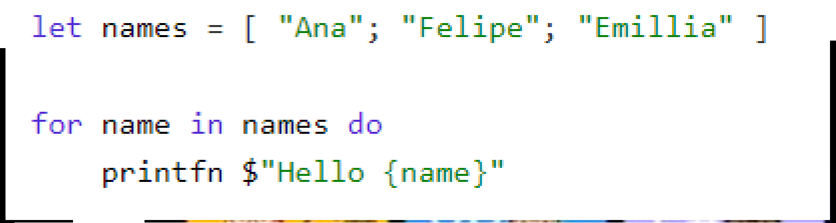
.NET | Free. Cross-platform. Open Source. (microsoft.com)

## DOTNET CORE

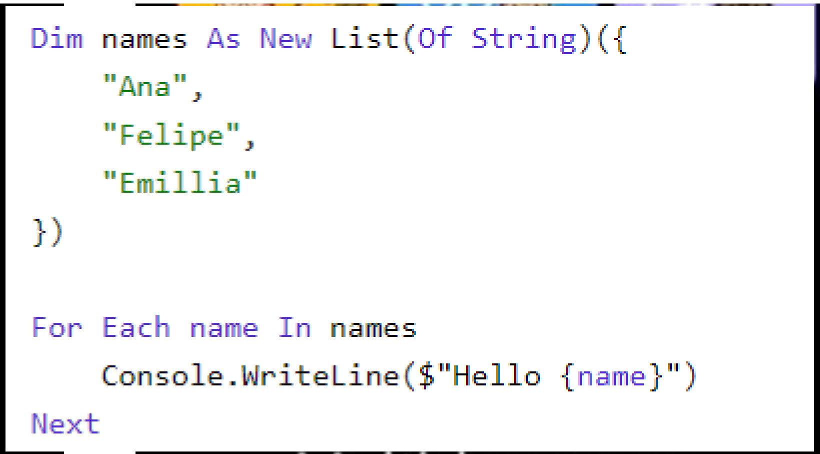


```csharp
var names = new[]
{
    "Ana",
    "Felipe",
    "Emillia"
};

foreach (var name in names)
{
    Console.WriteLine($"Hello {name}");
}
```

Compatible



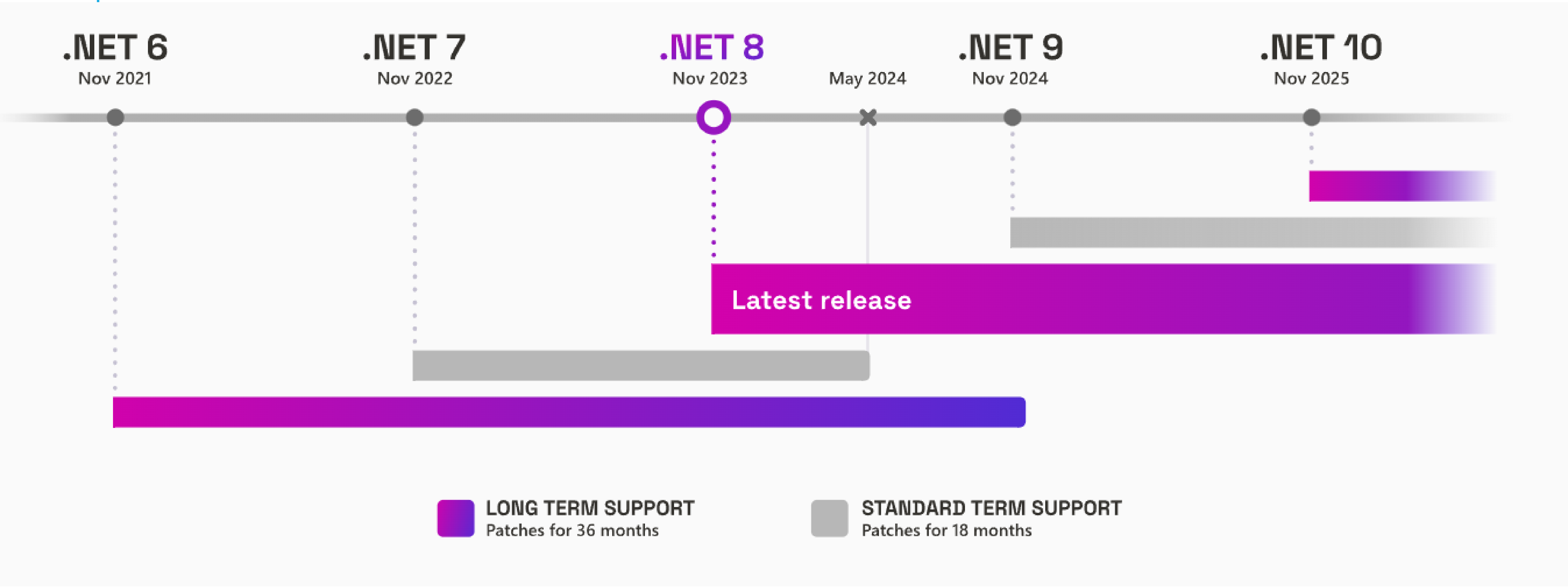
```fsharp
let names = [ "Ana"; "Felipe"; "Emillia" ]

for name in names do
    printfn $"Hello {name}"
```



```vb
Dim names As New List(Of String)({
    "Ana",
    "Felipe",
    "Emillia"
})

For Each name In names
    Console.WriteLine($"Hello {name}")
Next
```

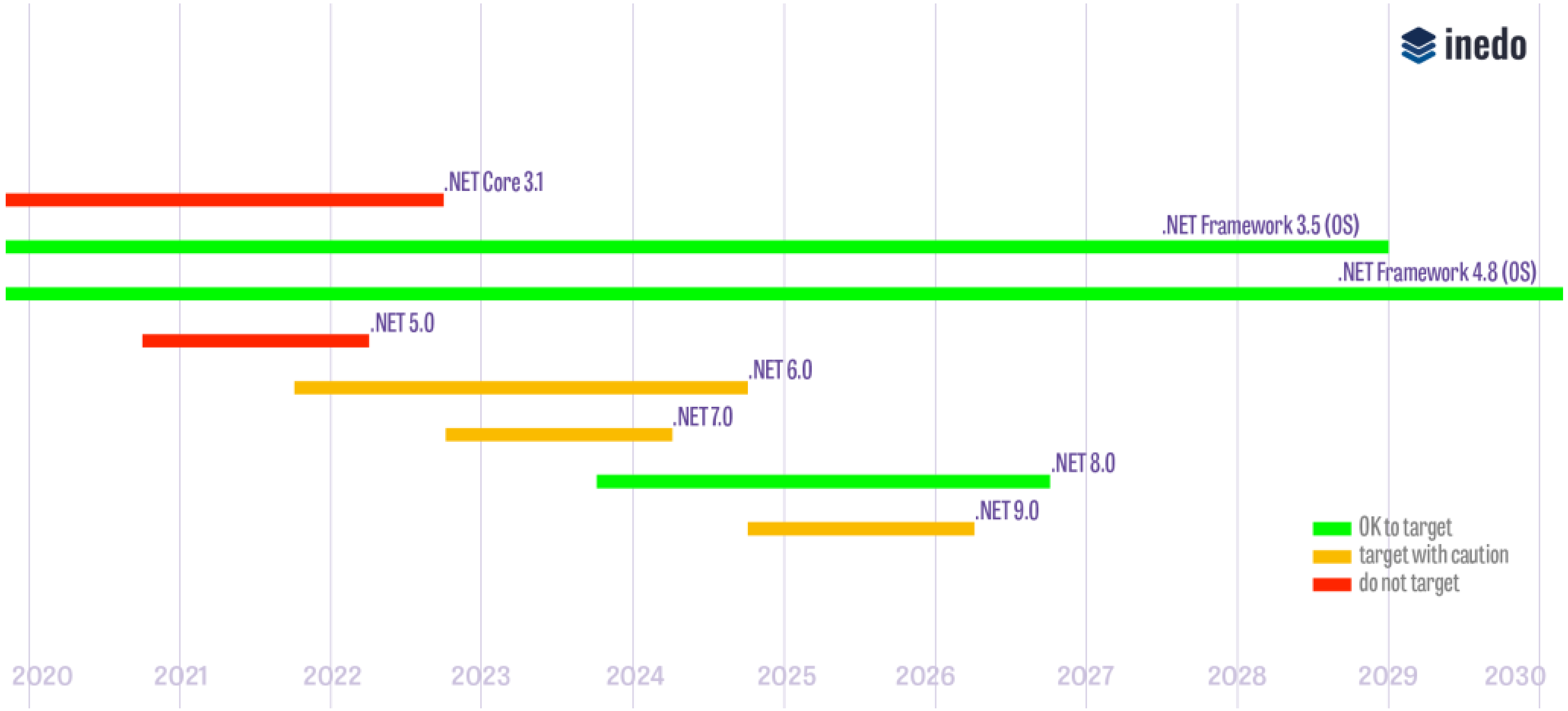
https://github.com/dotnet

# .NET - END OF LIFE



https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core

# .NET FRAMEWORK - END OF LIFE



https://learn.microsoft.com/en-us/lifecycle/products/microsoft-net-framework

# TOOLS

Single Platform

Cross Platform

SaaS

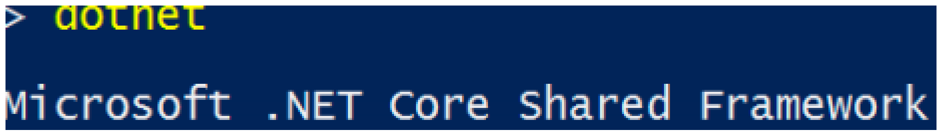
Visual Studio

CODE
Visual Studio Code

Github Codespaces

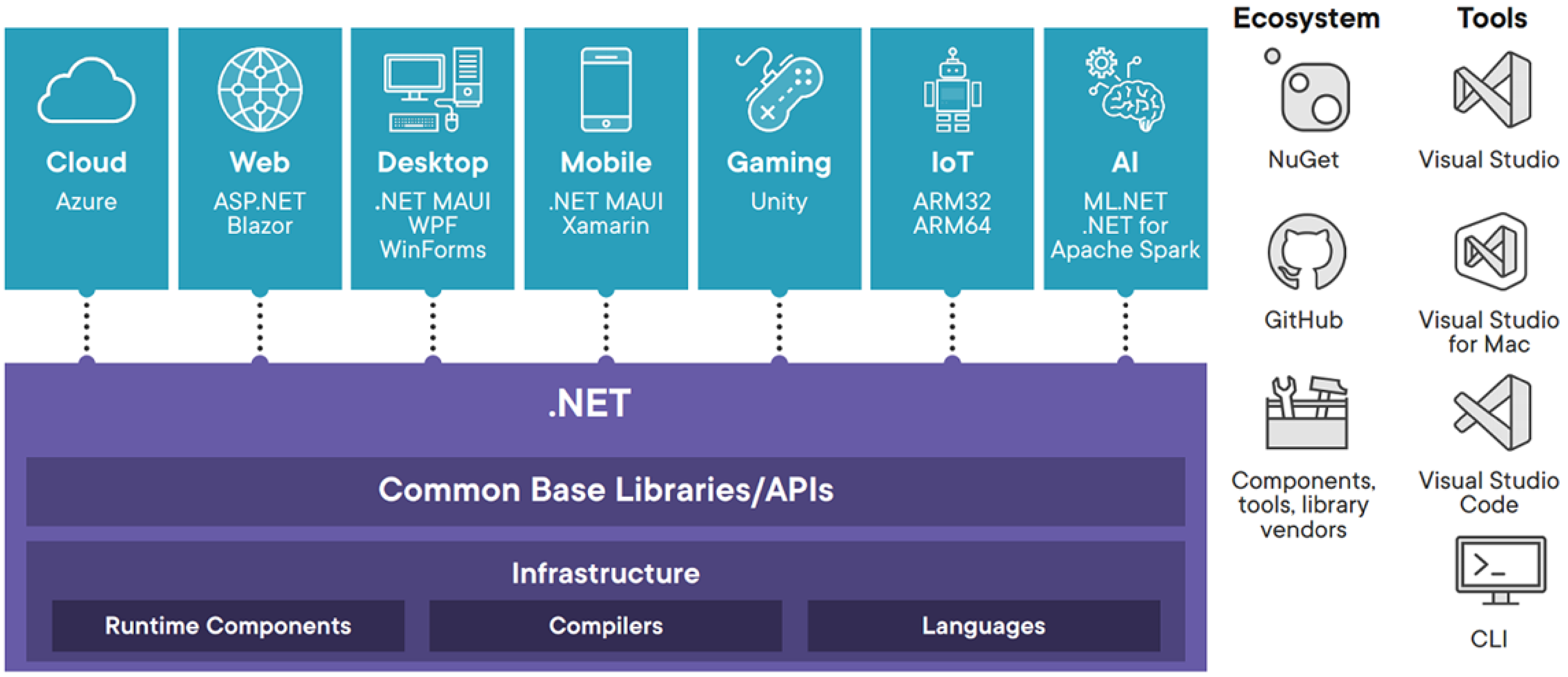
Rider

Visual Studio for Mac
August 2024 ,31



```
> dotnet

Microsoft .NET Core Shared Framework
```

## .NET Architecture

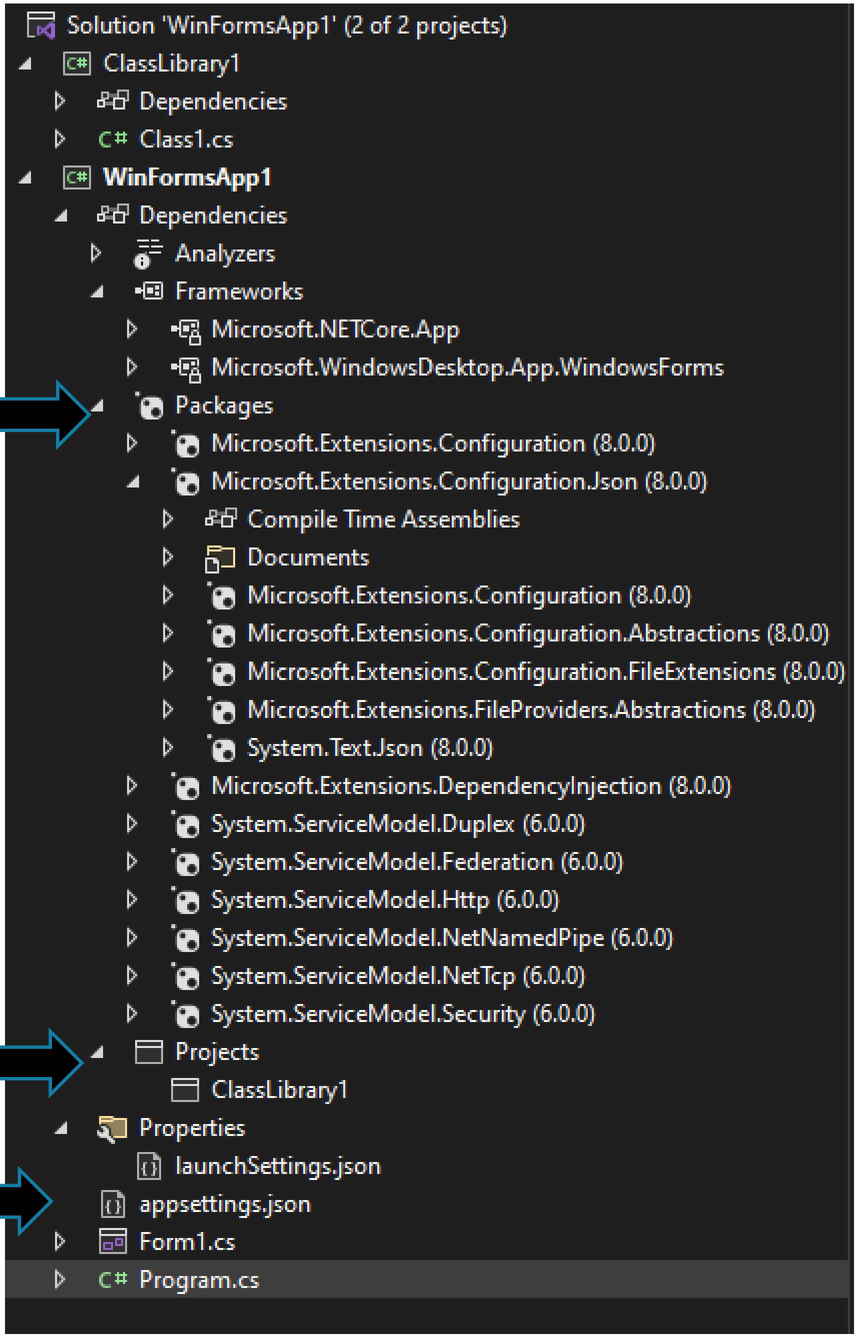


## EXPLORING THE PROJECT STRUCTURE

**NuGet/ project ref / npm /dll’s / other**

**Debug profile running**

**settings**



```
Solution 'WinFormsApp1' (2 of 2 projects)
◢ ClassLibrary1
    ▷ Dependencies
    ▷ Class1.cs
◢ WinFormsApp1
    ◢ Dependencies
        ▷ Analyzers
        ◢ Frameworks
            ▷ Microsoft.NETCore.App
            ▷ Microsoft.WindowsDesktop.App.WindowsForms
        ◢ Packages
            ▷ Microsoft.Extensions.Configuration (8.0.0)
            ◢ Microsoft.Extensions.Configuration.Json (8.0.0)
                ▷ Compile Time Assemblies
                ▷ Documents
                ▷ Microsoft.Extensions.Configuration (8.0.0)
                ▷ Microsoft.Extensions.Configuration.Abstractions (8.0.0)
                ▷ Microsoft.Extensions.Configuration.FileExtensions (8.0.0)
                ▷ Microsoft.Extensions.FileProviders.Abstractions (8.0.0)
                ▷ System.Text.Json (8.0.0)
            ▷ Microsoft.Extensions.DependencyInjection (8.0.0)
            ▷ System.ServiceModel.Duplex (6.0.0)
            ▷ System.ServiceModel.Federation (6.0.0)
            ▷ System.ServiceModel.Http (6.0.0)
            ▷ System.ServiceModel.NetNamedPipe (6.0.0)
            ▷ System.ServiceModel.NetTcp (6.0.0)
            ▷ System.ServiceModel.Security (6.0.0)
        ◢ Projects
            ClassLibrary1
    ◢ Properties
        launchSettings.json
    appsettings.json
    ▷ Form1.cs
    ▷ Program.cs
```

## GETTING STARTED WITH .NET APPLICATIONS

Runtime

.NET SDK

**In short**

The SDK is what you use to build and run your application.

The Runtime is to run the application

## SDK Components

- ✓ .NET CLI
- ✓ .NET driver
- ✓ Roslyn and F# compilers
- ✓ MSBuild build engine
- ✓ .NET runtime
- ✓ Runtime libraries
- ✓ ASP.NET Core runtime
- ✓ Desktop runtime

## CLI

Dotnet

dotnet --version

dotnet --list-sdks

Dotnet --help

Dotnet new

Dotnet new  console –o console_demo

Dotnet new wpf –o wpf_demo

Dotnet new wepapp –o web_app

Dotnet build

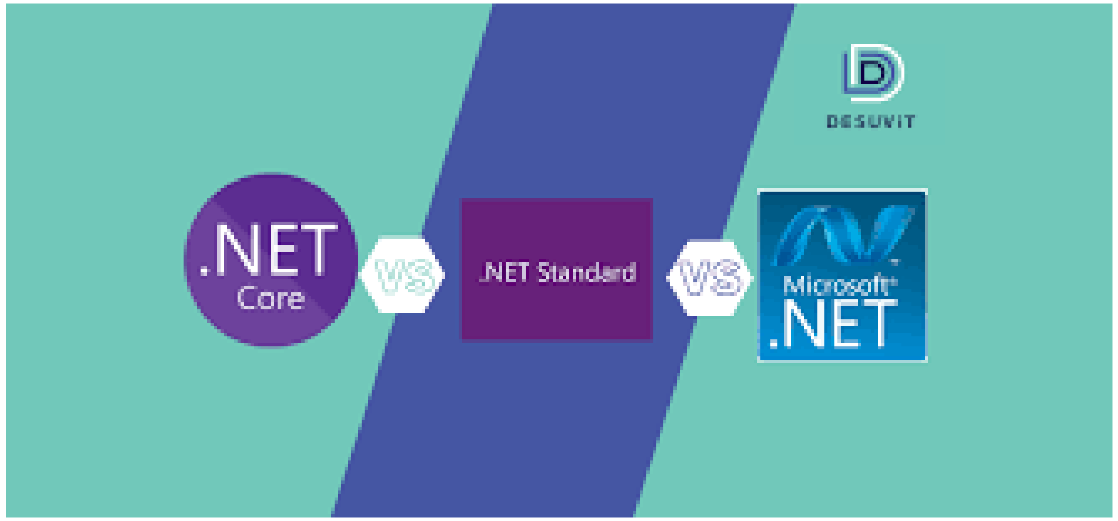
Dotnet publish

## INSTALL ON DOCKER

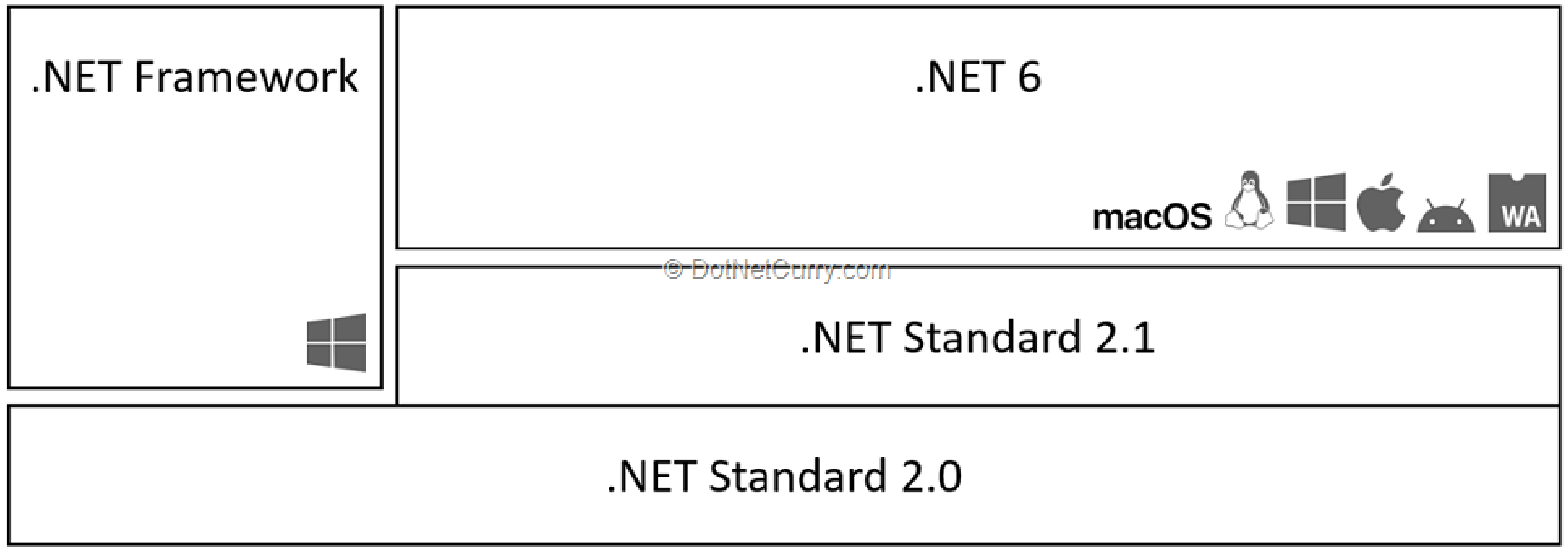
```
docker run --rm mcr.microsoft.com/dotnet/core/samples
```

```
docker run -it --rm -p 8000:80 --name aspnetcore_sample mcr.microsoft.com/dotnet/core/samples:aspnetapp
```

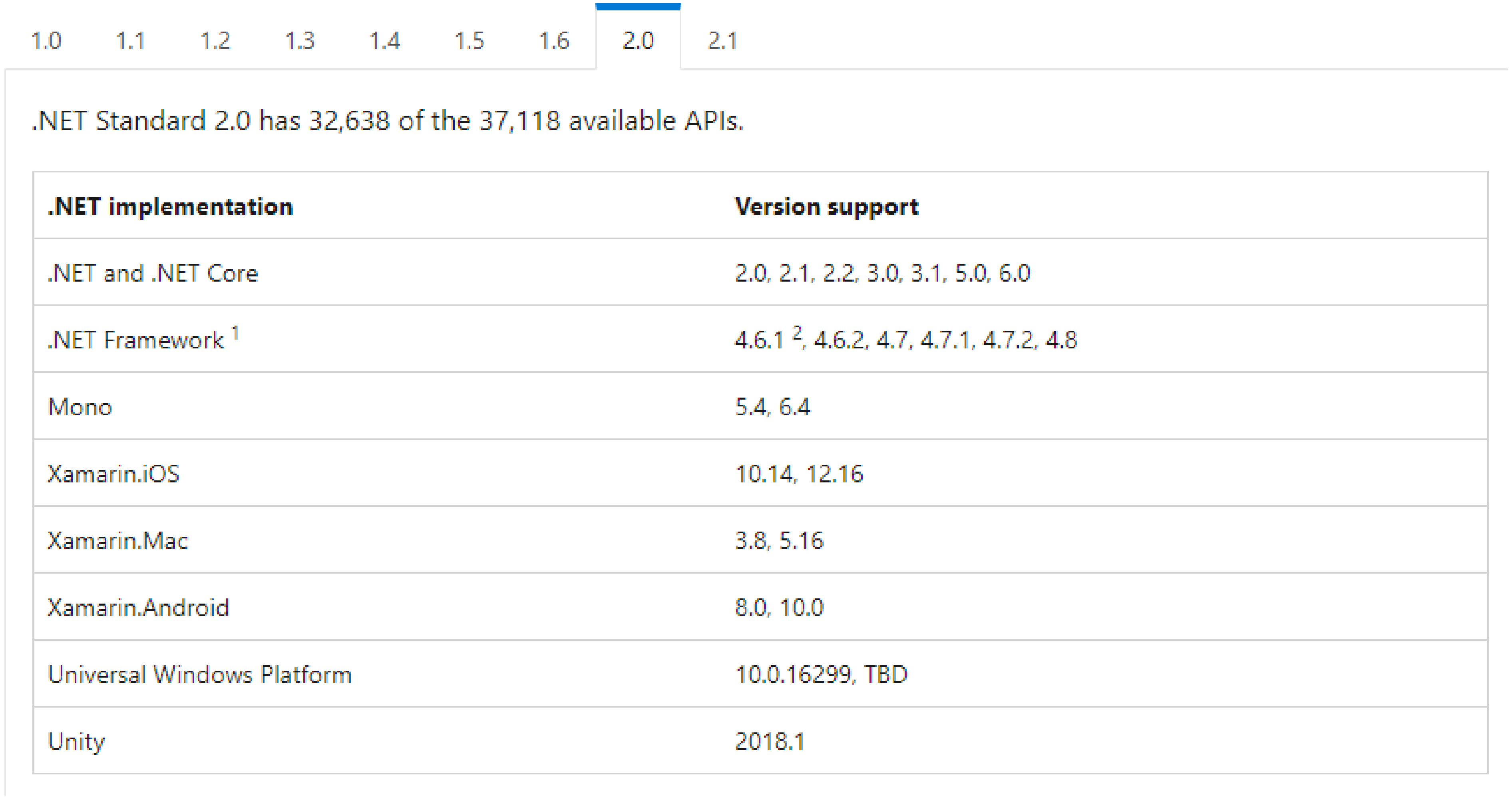
## .NET STANDARD CONTEXT



## .NET STANDARD CONTEXT



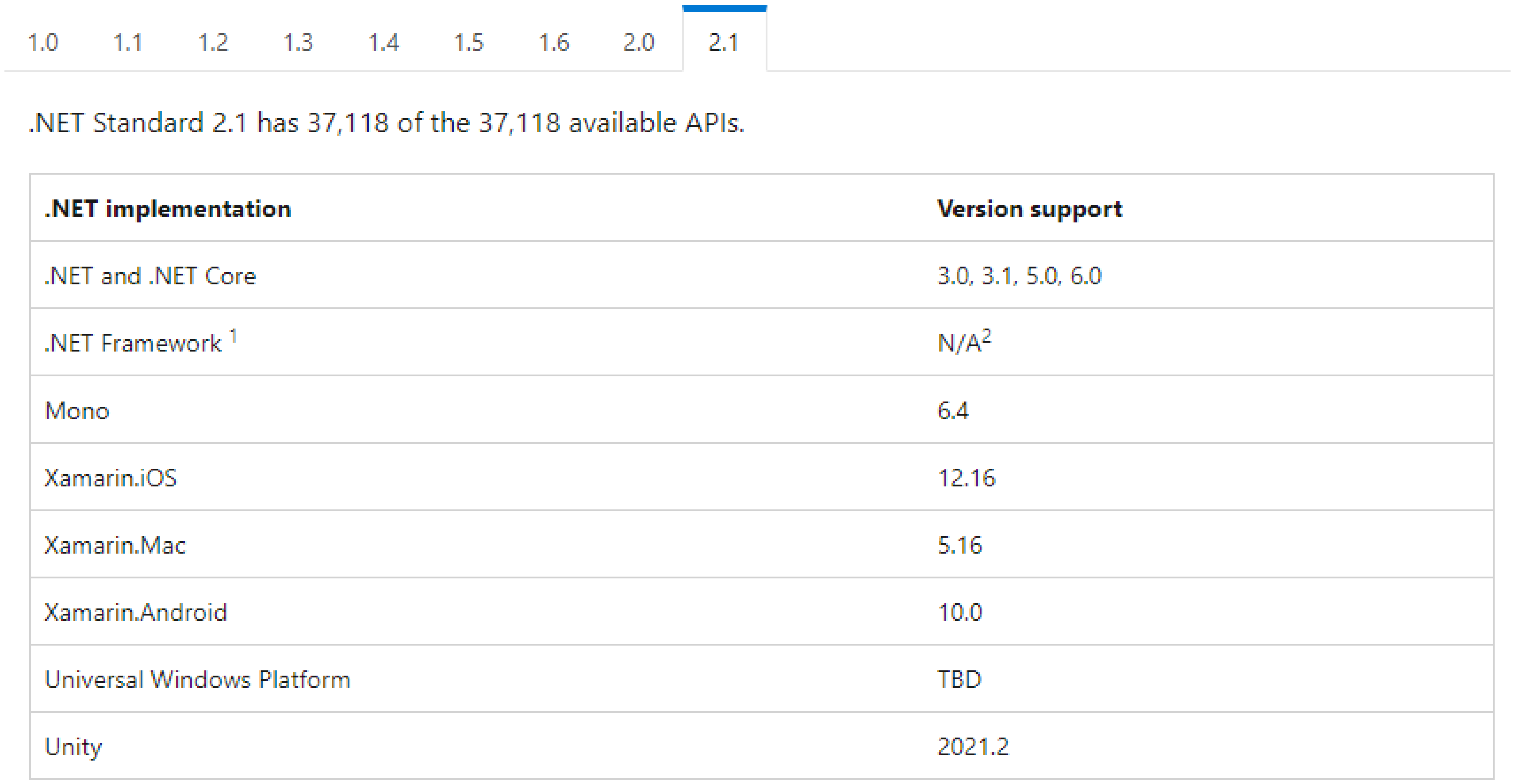
# .NET STANDARD 2.0



.NET Standard 2.0 has 32,638 of the 37,118 available APIs.

| **.NET implementation** | **Version support** |
|---|---|
| .NET and .NET Core | 2.0, 2.1, 2.2, 3.0, 3.1, 5.0, 6.0 |
| .NET Framework [1] | 4.6.1 [2], 4.6.2, 4.7, 4.7.1, 4.7.2, 4.8 |
| Mono | 5.4, 6.4 |
| Xamarin.iOS | 10.14, 12.16 |
| Xamarin.Mac | 3.8, 5.16 |
| Xamarin.Android | 8.0, 10.0 |
| Universal Windows Platform | 10.0.16299, TBD |
| Unity | 2018.1 |

# .NET STANDARD 2.1



.NET Standard 2.1 has 37,118 of the 37,118 available APIs.

| **.NET implementation** | **Version support** |
|---|---|
| .NET and .NET Core | 3.0, 3.1, 5.0, 6.0 |
| .NET Framework [1] | N/A[2] |
| Mono | 6.4 |
| Xamarin.iOS | 12.16 |
| Xamarin.Mac | 5.16 |
| Xamarin.Android | 10.0 |
| Universal Windows Platform | TBD |
| Unity | 2021.2 |

## .NET STANDARD

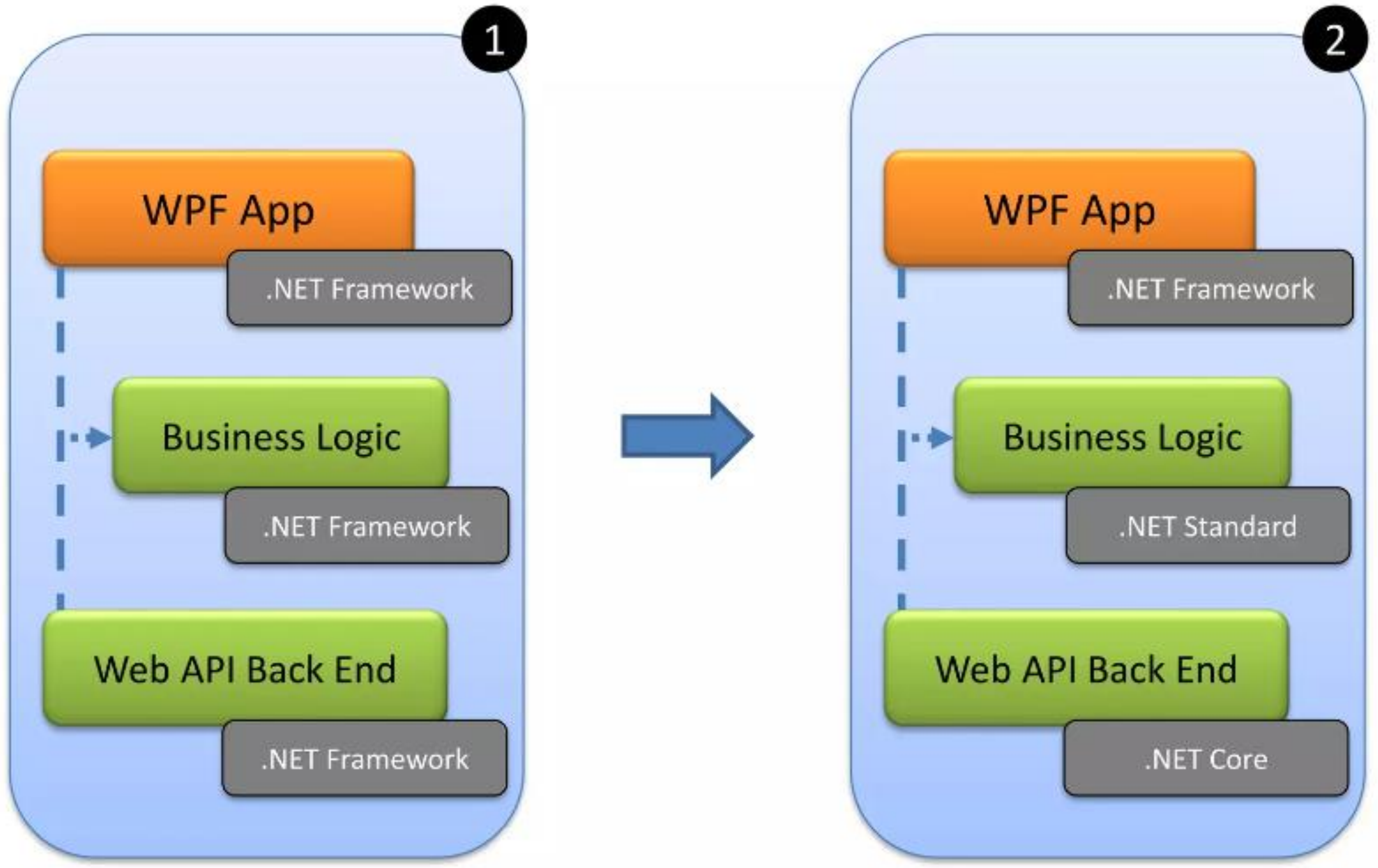
.NET Standard is also Open Source!

Anybody can propose API additions

The review board approves the API

https://docs.microsoft.com/en-us/dotnet/standard/net-standard

## HANDLING SHARED CODE WHEN TARGETING MULTIPLE .NET IMPLEMENTATIONS



## NUGET PACKAGE

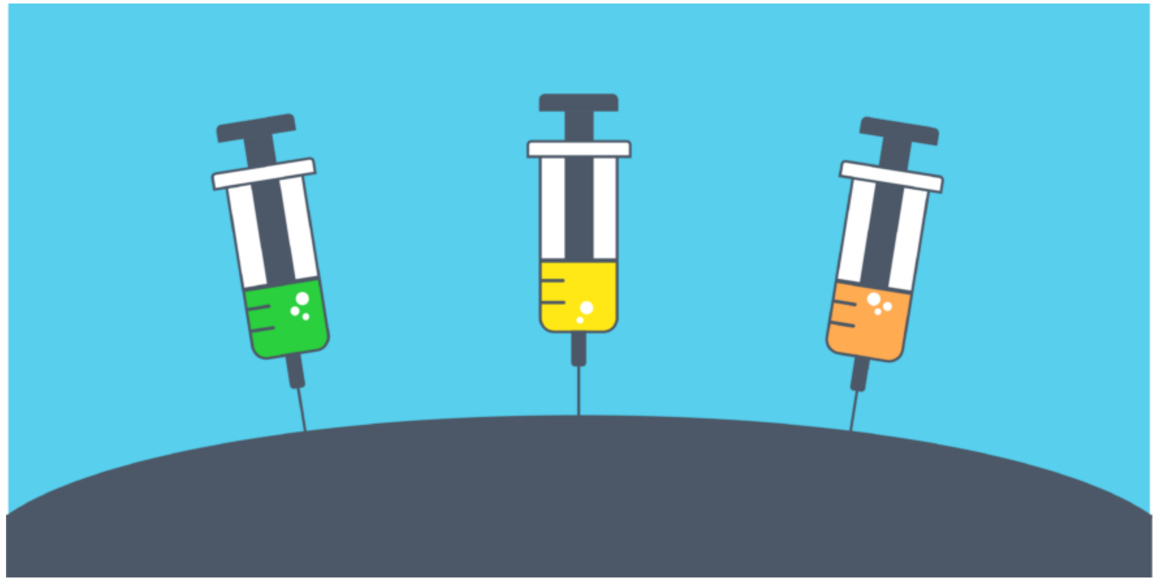
NuGet package manager

Project *.csproj file

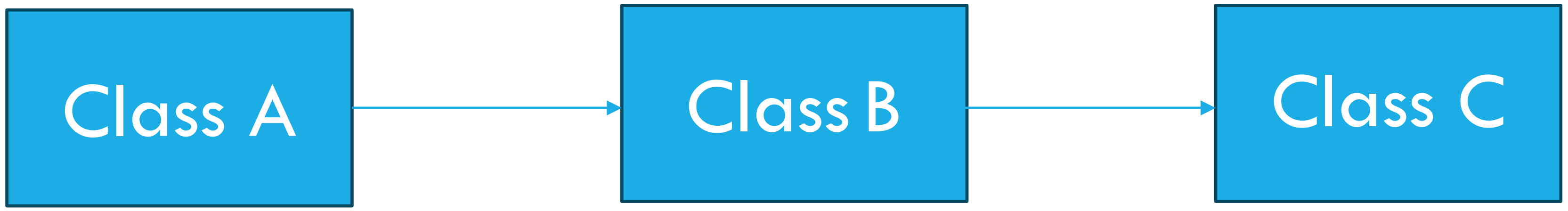
CLI: dotnet add package

PM> Install-Package

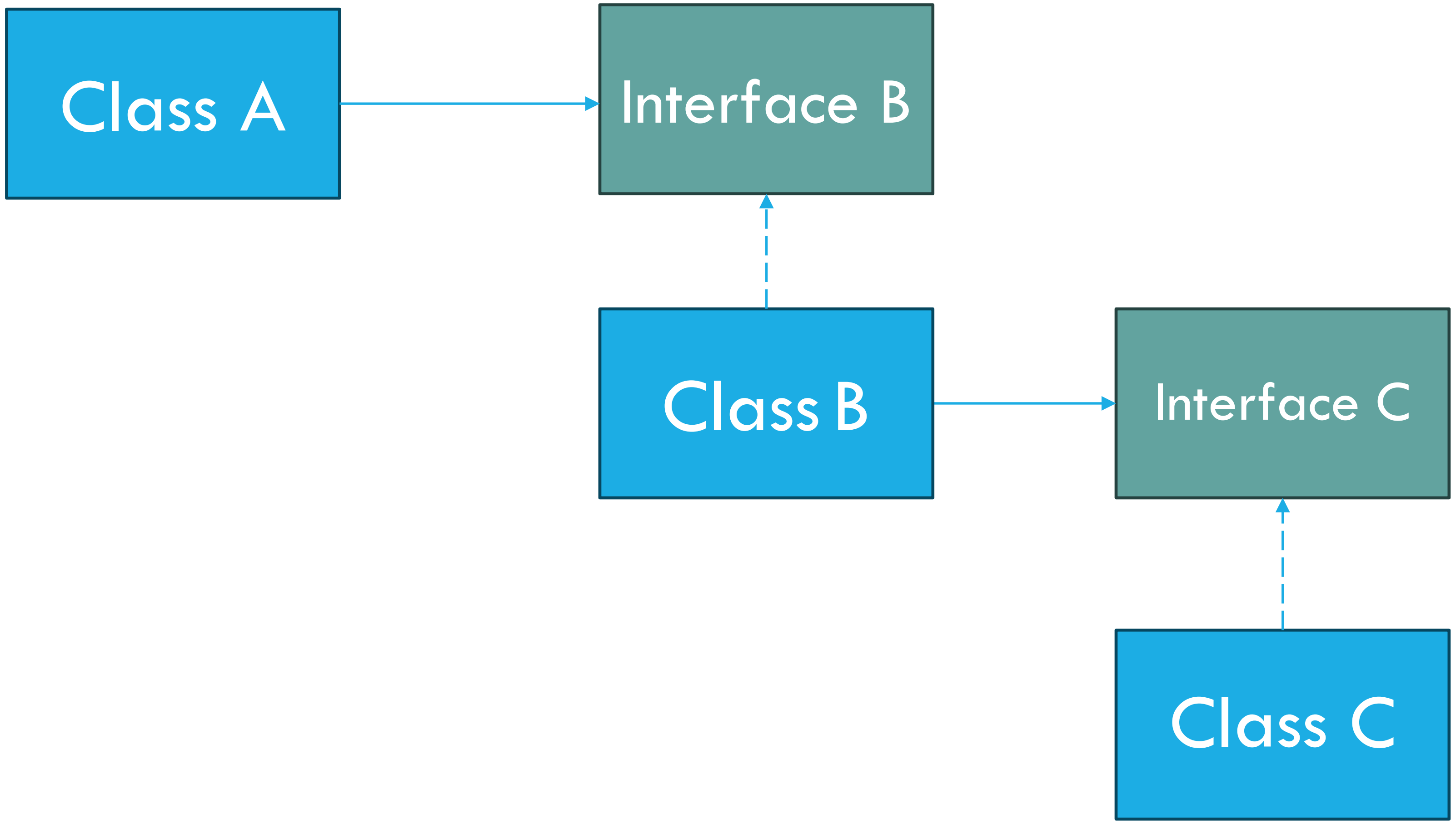
## DEPENDENCY INJECTION



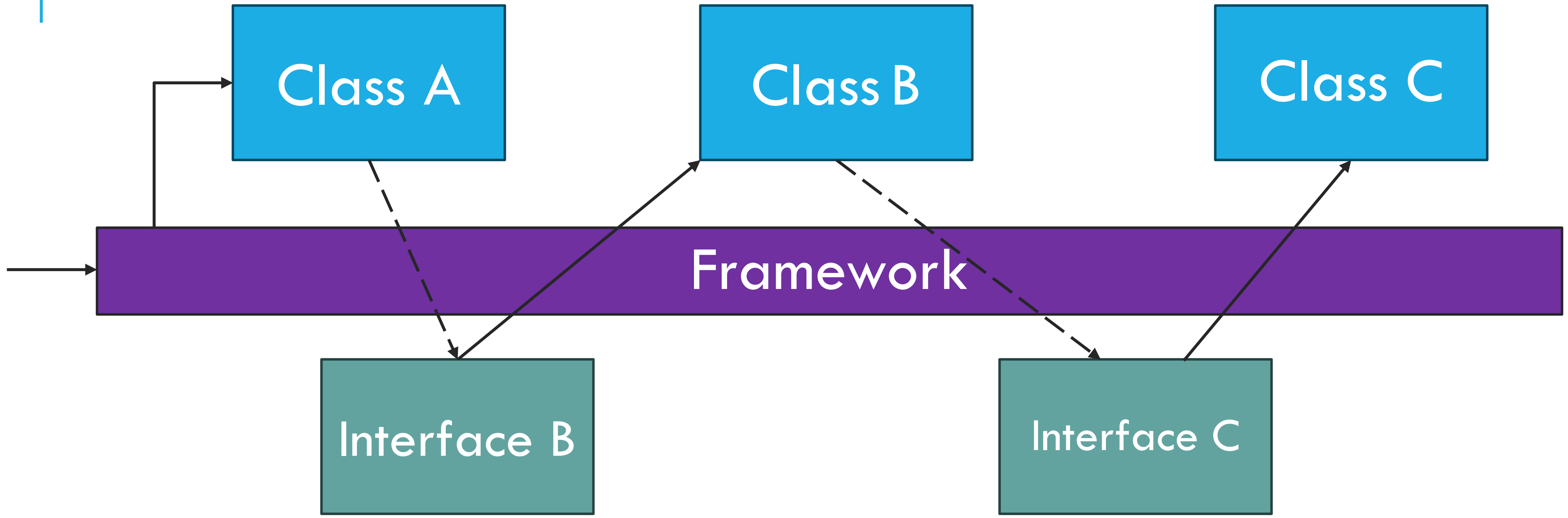
## TRADITIONAL FLOW



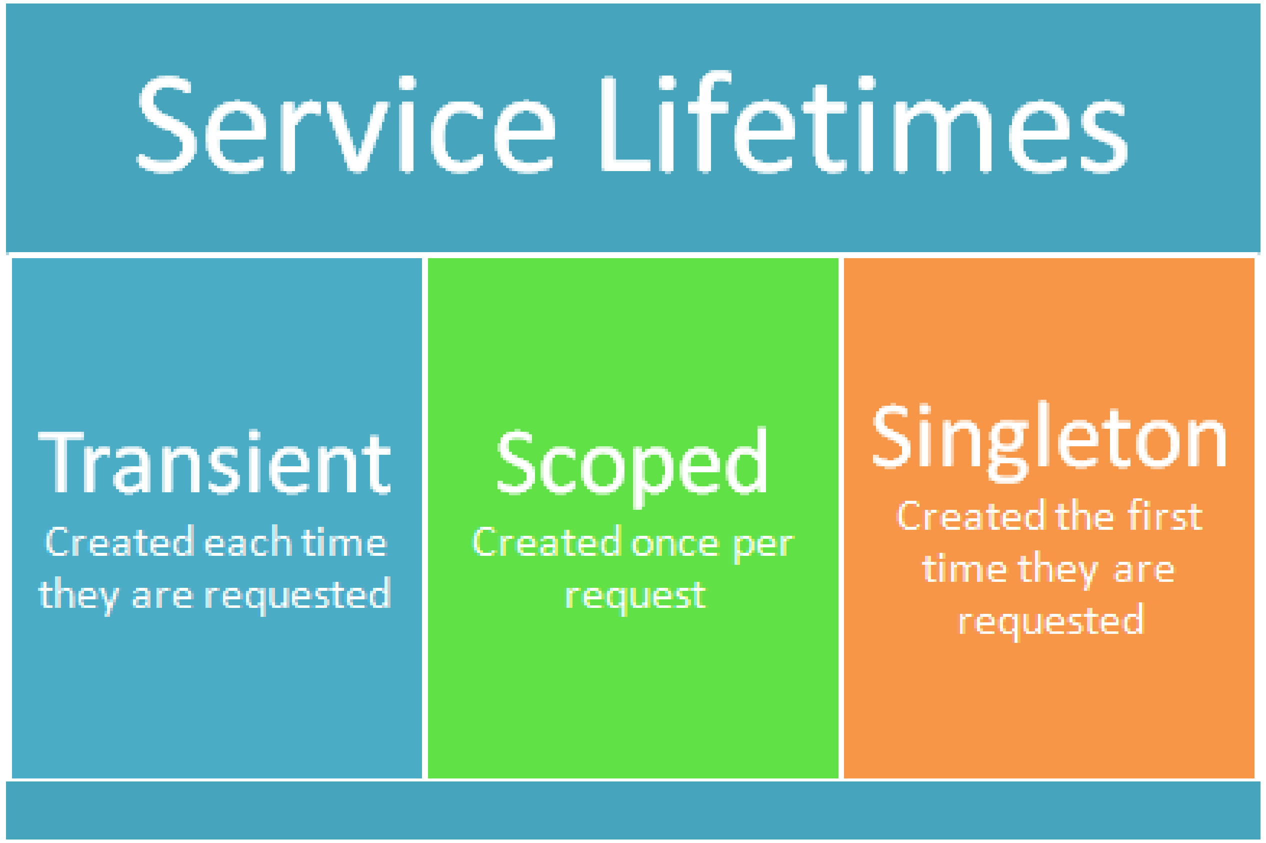
## DEPENDENCY INVERSION



## INVERSION OF CONTROL (IOC)



## LIFE TIME



Service Lifetimes

| Transient | Scoped | Singleton |
|---|---|---|
| Created each time they are requested | Created once per request | Created the first time they are requested |

# DEMO

# תרגיל 1

1. צור פרויקט Console דרך visual studio בשם Exercise1
2. צור מחלקה בשם StudentRepository
   - המחלקה תכלול פונקציה שמחזירה שמות סטודנטים לפי מספר בית ספר.
3. צור מחלקה בשם SchoolService
   - המחלקה תכלול פונקציה שמקבלת מספר בית ספר ומחזירה את שמות הסטודנטים.
4. הדפס את כל התלמידים בבית ספר מספר 1 בתוצאה.
5. השתמש ב IOC
6. הזרק את התלויות באמצעות ServiceProvider.–1ServiceCollection

| School ID | Student name |
|---|---|
| 1 | Moshe Levi |
| 1 | Avi Perez |
| 1 | Galit Mizrahi |
| 2 | Ronit Chen |
| 2 | Nivi Shemesh |

## ADD VS. TRYADD

**Add{Lifetime}**
If there is an existing registration for the type, this will overwrite it

**TryAdd{Lifetime}**
If there is an existing registration for the type, this will do nothing

## MULTIPLE REGISTRATIONS

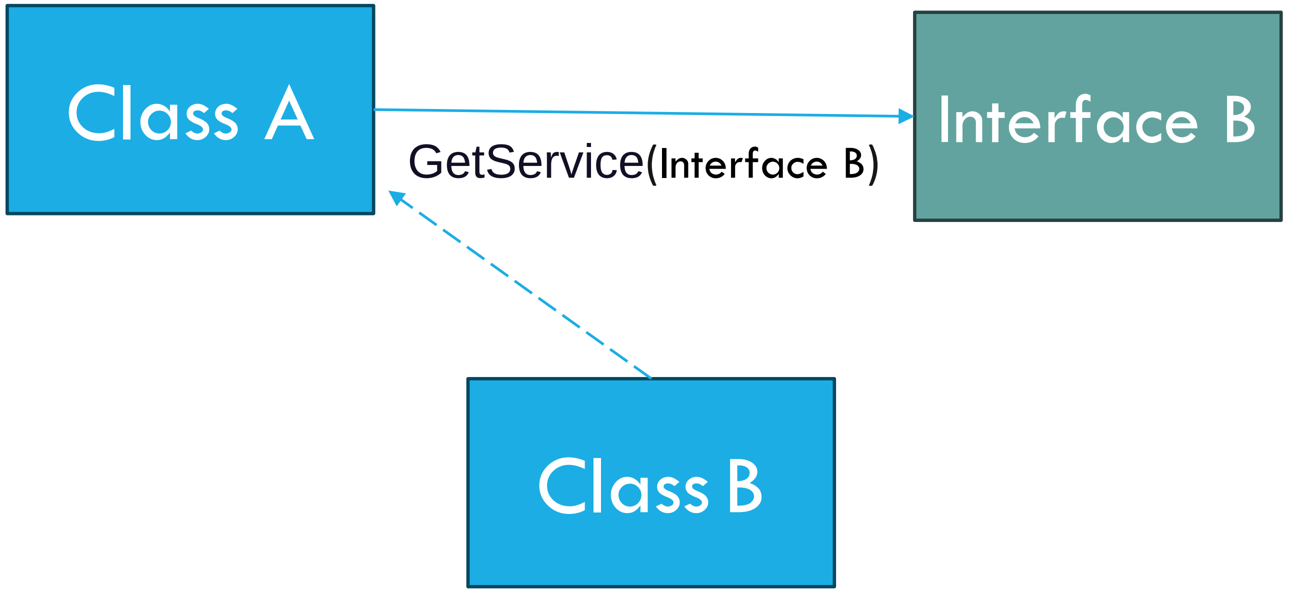
**Resolving directly**

The last registration is returned

**Resolving IEnumerable**

All registrations are returned

## SERVICE LOCATOR



```
public ClassA(IServiceProvider serviceProvider )
{
        _classB = serviceProvider.GetService<IClassB>();
}
```

## DEPENDENCY INJECTION **VS.** SERVICE LOCATOR

**Testability**

**Classes that use a Service Locator are harder to test**

**Implicit Dependencies**

**Dependencies are not clearly advertised, but implicit**

## MULTI-TENANT APPLICATION

**Problem Statement:**
In a multi-tenant application, different tenants might require different implementations of a service based on their subscription level or preferences

# BEST PRACTICES

- **Use Interfaces:** etatilicaf ot sessalc etercnoc fo daetsni secafretni esu syawlA : .gnitset dna gnippaws ycnedneped

- **Constructor Injection**.noitcejni dohtem ro ytreporp revo noitcejni rotcurtsnoc referP :

- **Keep Classes Slim:** Classes with too many dependencies may indicate too much responsibility. Try to refactor heavy classes into smaller, more focused ones.

- **Avoid Service Locator**: Minimize the use of ServiceProvider within your code. It breaks the Inversion of Control principle.

- **Service Lifetime Management:** Ensure to choose the appropriate service lifetime (Transient, Scoped, Singleton) based on the application needs.

- **Unit Testing**: Leverage dependency injection to write effective unit tests with mocks.

## תרגיל מסכם