* **What does the .NET ecosystem provide?**

The .NET ecosystem is a collection of tools, libraries, and frameworks that are used to develop, deploy, and run applications on the .NET platform. The ecosystem includes the following components:

1. **.NET Framework**: A software framework developed by Microsoft that runs primarily on Microsoft Windows operating systems. It includes a large class library known as Framework Class Library (FCL) and provides language interoperability across several programming languages.
2. **.NET Core**: A free, open-source, cross-platform framework for building modern cloud-based applications. It is designed to be modular and lightweight and can be used to build applications for Windows, Linux, and macOS.
3. **ASP.NET**: A web application framework developed by Microsoft that allows developers to build dynamic web sites, web applications, and web services.
4. **Entity Framework**: An object-relational mapping (ORM) framework that allows developers to work with relational data using domain-specific objects.
5. **Visual Studio**: An integrated development environment (IDE) used to develop computer programs, websites, web apps, mobile apps, and cloud services.
6. **NuGet**: A package manager for the Microsoft development platform including .NET. It is used to install, update, and manage packages in a project.
7. **Roslyn Compiler**: A set of open-source compilers and code analysis APIs for C# and Visual Basic .NET languages.
8. **Xamarin**: A cross-platform app development platform that allows developers to create native iOS, Android, and Windows apps using a single codebase.
9. **Azure**: A cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers.

* **What are .NET implementations? Which ones are used nowadays?**
* **.NET Framework**: A software framework developed by Microsoft that runs primarily on Microsoft Windows operating systems. It includes a large class library known as Framework Class Library (FCL) and provides language interoperability across several programming languages.
* **.NET Core**: A free, open-source, cross-platform framework for building modern cloud-based applications. It is designed to be modular and lightweight and can be used to build applications for Windows, Linux, and macOS.
* **Mono**: A cross-platform implementation of the .NET Framework that is mainly used when a small runtime is required. It supports all of the currently published .NET Standard versions .

Microsoft supports four .NET implementations: .NET 5 (and .NET Core) and later versions, .NET Framework, Mono, and Universal Windows Platform (UWP). The latest version of .NET is .NET 7, which is built on a single code base that supports multiple platforms and many workloads such as Windows desktop apps and cross-platform console apps, cloud services, and websites.

Each implementation of .NET includes one or more runtimes, a class library, and optionally one or more application frameworks. The choice of implementation depends on the specific requirements of the application being developed.

* **What is CLR?**

The **Common Language Runtime (CLR)** is a component of the Microsoft .NET Framework that manages the execution of .NET applications. It is responsible for loading and executing the code written in various .NET programming languages, including C#, VB.NET, F#, and others. When a C# program is compiled, the resulting executable code is in an intermediate language called Common Intermediate Language (CIL) or Microsoft Intermediate Language (MSIL). This code is not machine-specific, and it can run on any platform that has the CLR installed. When the CIL code is executed, the CLR compiles it into machine code that can be executed by the processor. The CLR provides many services to .NET applications, including memory management, type safety, security, and exception handling. It also provides Just-In-Time (JIT) compilation, which compiles the CIL code into machine code on the fly as the program runs, optimizing performance. Additionally, the CLR provides a framework for developing and deploying .NET applications, including a set of libraries, called the .NET Framework Class Library, which provides access to a wide range of functionality, such as input/output operations, networking, database connectivity, and user interface design.

* **Why is .NET 5 a bit of a special version?**

.NET 5 is a bit of a special version for several reasons. Firstly, it is the **first release** of .NET that does not include the word “Core” in its name. Secondly, it is the **first release** of .NET that supports more types of apps and more platforms than .NET Core or .NET Framework. Thirdly, it is the **first release** of .NET that combines and replaces the netcoreapp and netstandard Target Framework Monikers (TFMs). Fourthly, it is the **first release** of .NET that includes many new features and improvements compared to .NET Core 3.1. Lastly, it is the **first release** of .NET that is built on a single code base that supports multiple platforms and many workloads such as Windows desktop apps and cross-platform console apps, cloud services, and websites.

* **Which technologies are supported by the .NET framework?**

The .NET framework is a collection of tools, libraries, and frameworks that are used to develop, deploy, and run applications on the .NET platform. The following are some of the technologies supported by the .NET framework:

1. **ASP.NET**: A web application framework developed by Microsoft that allows developers to build dynamic web sites, web applications, and web services.
2. **Windows Communication Foundation (WCF)**: A framework for building service-oriented applications.
3. **Windows Presentation Foundation (WPF)**: A graphical subsystem for rendering user interfaces in Windows-based applications.
4. **Windows Workflow Foundation (WF)**: A framework for building workflow-enabled applications.
5. **Windows CardSpace**: A software component that provides a consistent way for users to manage their digital identities.
6. **ADO.NET**: A set of classes that expose data access services for .NET Framework programmers.
7. **LINQ**: A set of extensions to the .NET Framework that provides a standard way to query data from different data sources.
8. **Parallel Extensions**: A managed concurrency library that simplifies the development of concurrent and parallel applications.
9. **Managed Extensibility Framework (MEF)**: A framework for building extensible applications.
10. **Windows Azure**: A cloud computing platform developed by Microsoft for building, deploying, and managing applications and services through a global network of Microsoft-managed data centers.

* **Does cross-platform development is possible in .NET? What about cross-platform UI?**

Yes, cross-platform development is possible in .NET. **.NET Core** is a free, open-source, cross-platform framework for building modern cloud-based applications that can be used to build applications for Windows, Linux, and macOS. Additionally, .NET Multi-platform App UI (**MAUI**) is a cross-platform framework for creating native mobile and desktop apps with C# and XAML that can run on Android, iOS, macOS, and Windows from a single shared code-base.

* **What does the multitargeting term mean?**

In the context of .NET, multitargeting refers to the ability to compile applications that can run on different versions of the .NET runtime. It allows developers to target multiple versions of the .NET Framework or .NET Core with a single codebase. Multitargeting ensures that your application does not use any types or methods that were introduced in newer versions of the runtime than the one you are targeting.