**.NET and .NET Core**

.NET and .NET Core are both frameworks developed by Microsoft for building and running applications on the Windows platform. However, there are significant differences between the two. Let's explore the main differences between .NET and .NET Core:

**1. Cross-platform Compatibility:**

.NET: Traditionally, the .NET Framework was designed for Windows-only applications. It relies on the Windows operating system and is not compatible with other platforms.

.NET Core: On the other hand, .NET Core is cross-platform and can run on Windows, macOS, and Linux. It was developed with a focus on being lightweight and modular, making it suitable for various types of applications, including web, desktop, and cloud-based applications.

**2. Framework Architecture:**

.NET: The .NET Framework is a monolithic framework that includes a large set of libraries, APIs, and features for various purposes. It is installed on the system, and applications rely on the version installed on the user's machine.

.NET Core: In contrast, .NET Core is a modular and open-source framework. It is distributed as a set of NuGet packages and allows you to include only the components needed for your application. This results in a smaller footprint and better performance for .NET Core applications.

**3. Development and Support:**

.NET: The .NET Framework has been around for a long time and has a mature ecosystem with extensive libraries, tools, and third-party support. It is primarily used for Windows desktop applications, enterprise applications, and older ASP.NET web applications.

.NET Core: .NET Core is a more recent addition to the family of .NET frameworks. It is actively developed and has become the preferred choice for building modern cross-platform applications, especially web applications, microservices, and cloud-based applications.

**4. Dependency and Deployment:**

.NET: Applications built on the .NET Framework are tied to the version of the framework installed on the target machine. This could lead to versioning issues and requires updates to the framework separately.

.NET Core: Since .NET Core is distributed as a set of packages, applications can include the specific versions of the required components, reducing the likelihood of version conflicts. This makes deployment and maintenance easier for .NET Core applications.

**5. ASP.NET:**

.NET: ASP.NET is the web development framework built on top of the .NET Framework. It includes technologies like ASP.NET Web Forms and ASP.NET MVC.

.NET Core: ASP.NET Core is the modern web development framework built on top of .NET Core. It provides a more lightweight and modular approach to building web applications, with improved performance and cross-platform support.

In summary, while .NET Framework is well-established and suitable for Windows-specific applications, .NET Core offers a cross-platform and more lightweight alternative, making it a preferred choice for modern web, cloud, and cross-platform applications. Microsoft has also announced that .NET 5 and later versions will be unified as ".NET" and will include the best features of both .NET Framework and .NET Core.