Common Language Runtime

.NET

Common Language Runtime (CLR) *is a managed execution environment that is part of Microsoft's .NET framework.* ***CLR*** *manages the execution of programs written in different supported languages.* ***CLR*** *transforms source code into a form of bytecode known as* ***CIL (Common Intermediate Language)****.*

HTTPS://WWW.TECHOPEDIA.COM/DEFINITION/5225/COMMON-LANGUAGE-RUNTIME-CLR#:~:TEXT=

CLR (Common Language Runtime)

https://docs.microsoft.com/en-us/dotnet/standard/clr
https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview

---

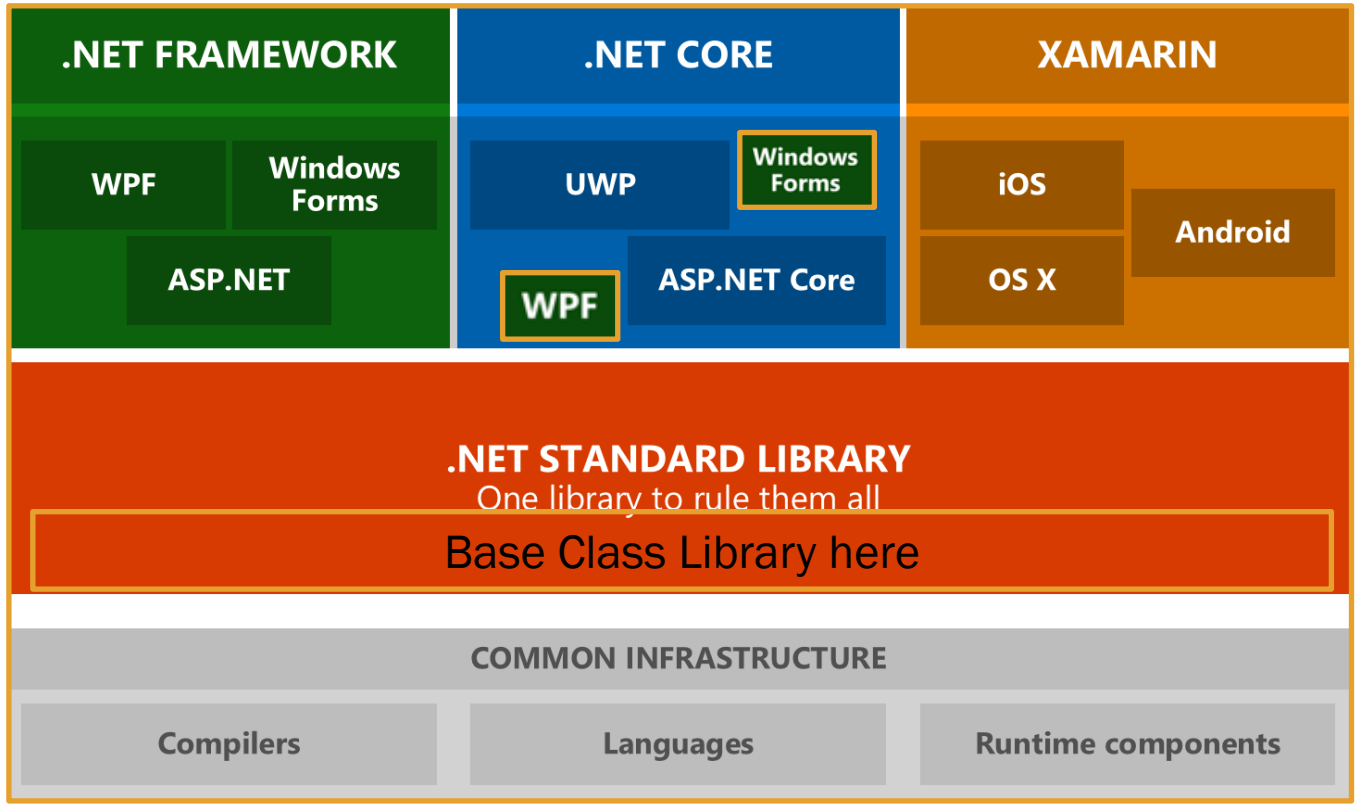
- The .NET Framework consists of the ***Common Language Runtime*** (CLR) and the ***.NET Framework base class library***.
- The ***CLR*** is the foundation for .NET Framework. It manages and runs the code and provides services like memory management, remoting, type enforcement (through the ***CTS***), and security.

Benefits of CLR:		
cross-language integration	cross-language exception handling	enhanced security
versioning and deployment support	a simplified model for component interaction	debugging and profiling services.

.NET Class Libraries

https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview

A ***class library*** is an object-oriented collection of reusable ***types*** that you can use to develop apps ranging from traditional command-line or graphical user interface (GUI) apps to apps based on the latest innovations provided by ASP.NET, such XML Web services.

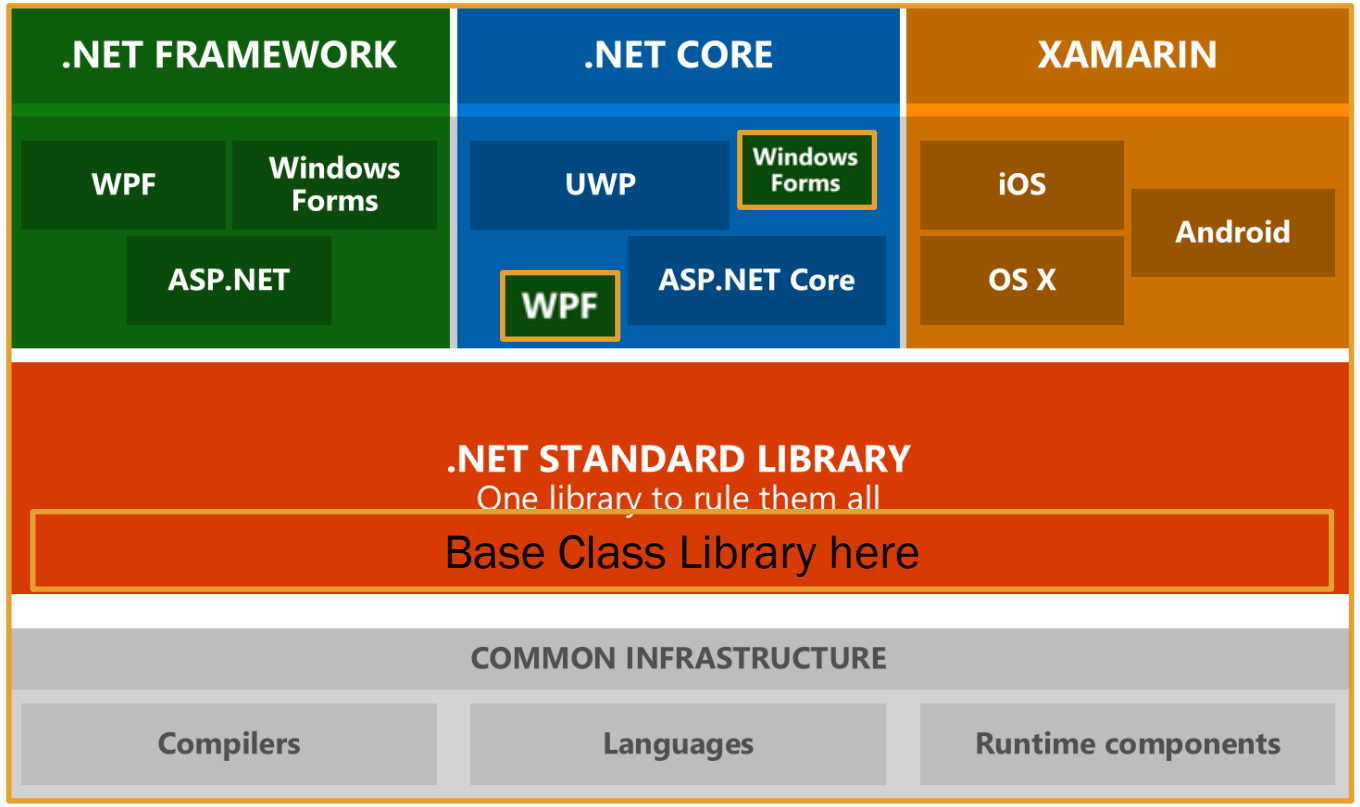


BCL (Base Class Library)

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

BCL stands for ***Base Class Library*** (AKA, ***Class library (CL)***). A .NET Framework library, ***BCL*** is the foundation for the C# runtime library and one of the ***Common Language Infrastructure (CLI)*** standard libraries.

BCL provides the ***types*** that represent built-in ***CLI*** data types, basic file access, collections, custom attributes, formatting, security attributes, I/O streams, string manipulation, etc.

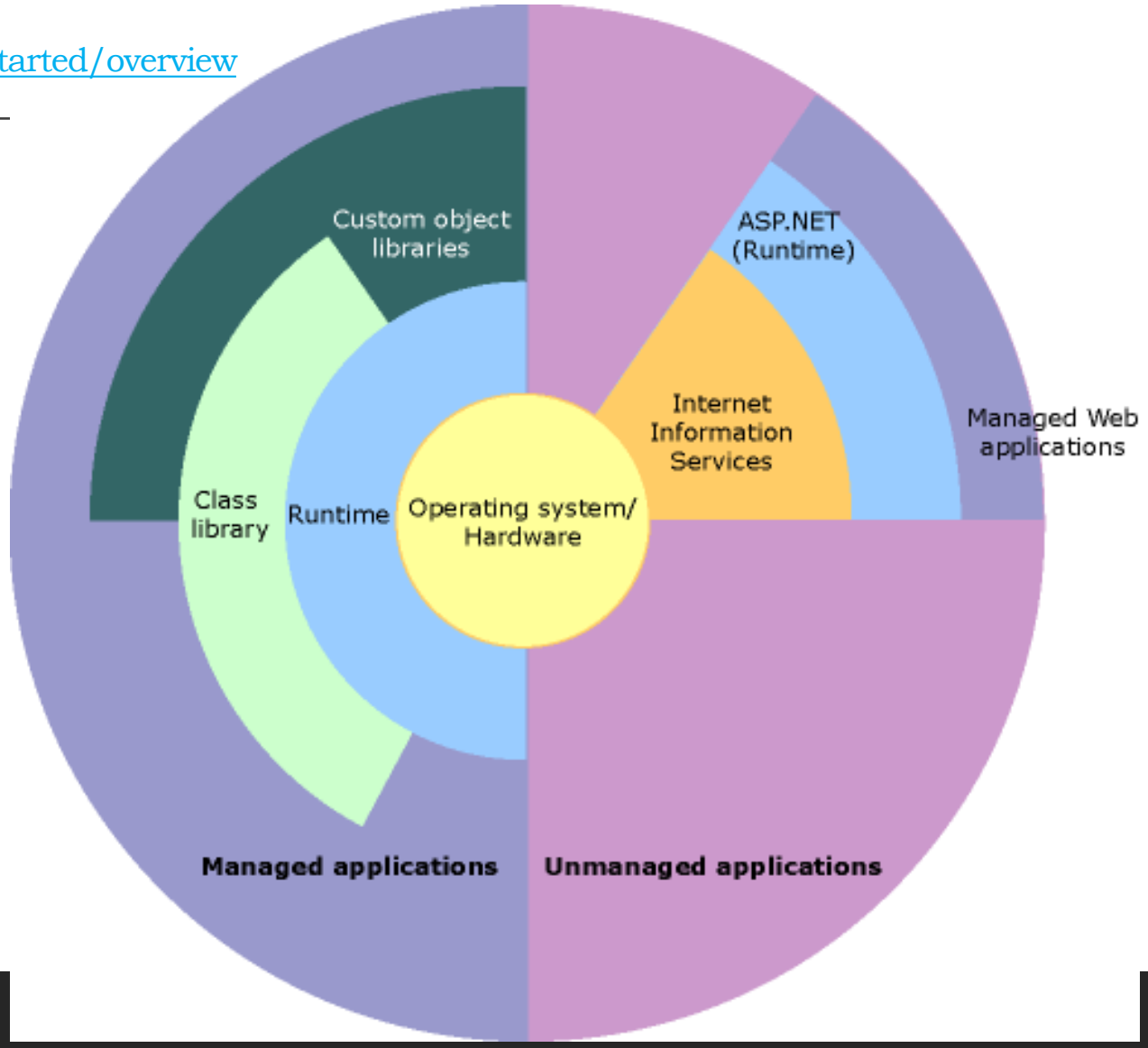


.NET CLR and Class Library Relationship

https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview

This illustration shows the relationship of the ***Common Language Runtime*** and the class library to your apps and to the overall system.

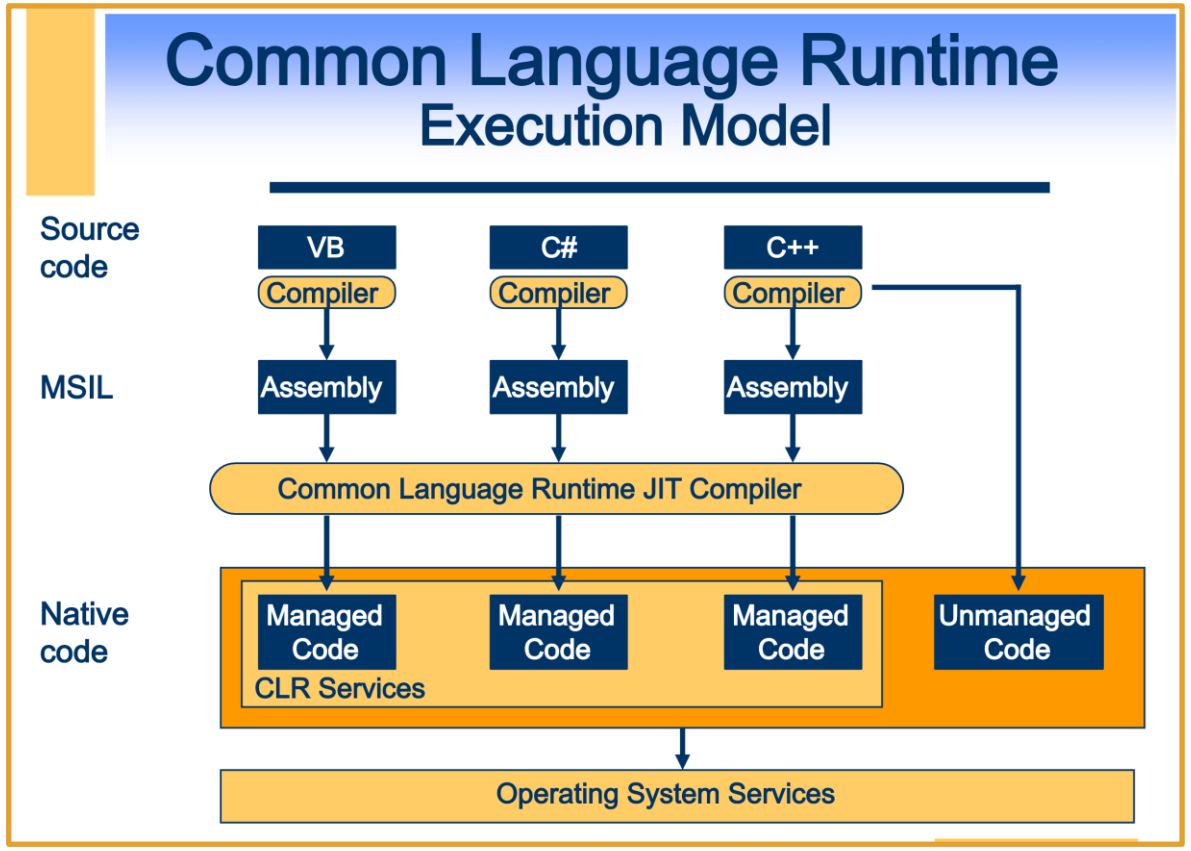
The illustration also shows how managed code operates within a larger architecture.



Managed Code

https://docs.microsoft.com/en-us/dotnet/standard/managed-code
https://docs.microsoft.com/en-us/dotnet/standard/managed-execution-process

- Managed code is managed by the **Common Language Runtime** (***CLR***) at runtime.
- The ***CLR*** knows what your code is doing and can *manage* it.
- The ***CLR*** provides memory management (***GC***), security boundaries, type safety, etc.
- Managed code is written in a high-level language that can be run on top of .NET.
- Code is compiled into **Intermediate Language** code, which the ***CLR*** compiles and executes.
- The ***CLR*** manages the **Just-In-Time** compiling of code from ***IL*** to machine code that can be run on a ***CPU***.



Unmanaged Code

https://docs.microsoft.com/en-us/dotnet/framework/interop/

Code that runs outside the ***CLR*** is called Unmanaged Code.

The .NET Framework promotes interaction with COM components, COM+ services, external type libraries, and many operating system services.

Examples of Unmanaged Code:

- COM components,
- ActiveX interfaces,
- Windows API functions.