Dot NET Framework Introduction

- All the capabilities of .NET Framework are exposed via a vast set of managed types.
- These types are organized into hierarchical namespaces and packaged into a set of assemblies, which together with the CLR (Common Language Run- time) comprise the .NET platform.
- Some of the .NET types are used directly by the CLR and are essential for the managed hosting environment.
- These types reside in an assembly called **mscorlib.dll**.

CONTD…

- The .NET Framework consists of the common language runtime and the .NET Framework class library.
- Types include C#'s built-in types, as well as the basic collection classes, types for stream processing, serialization, reflection, threading, and native interoperability ("mscorlib" is an abbreviation for "Multi-language Standard Common Object Runtime Library").

CONTD…

- .NET Framework is a technology that supports building and running the applications.
- The .NET Framework is designed to fulfill the following objectives:
  - Provide a runtime environment that minimizes software deployment and versioning conflicts.
  - Enable the safe execution of code.
  - Provide a consistent developer experience across all types of applications in a way that is language- and platform-independent.
  - Provide a runtime environment that minimizes or eliminates the performance problems of scripted or interpreted environment.

PLATFORM SUPPORT

- C# is typically used for writing code that runs on Windows platforms.
- Xamarin allows cross platform C# development for mobile applications
- Microsoft's ASP.NET Core is a cross-platform lightweight web hosting framework that can run either on the .NET Framework or on .NET Core
- It is an open source cross-platform runtime.

C# AND THE CLR

- C# is an object-oriented, ***component-oriented*** programming language.
- C# depends on a runtime equipped with a host of features such as security, automatic memory management and exception handling.
- Common Language Runtime (CLR) is the core of the Microsoft .NET Framework which provides these runtime features (.NET Core and Xamarin frameworks provide similar runtimes).
- The CLR is language-neutral, allowing developers to build applications in multiple languages (e.g., C#, F#, Visual Basic .NET, and Managed C++).
- C# is one of several managed languages that get compiled into managed code.
- Managed code is represented in Intermediate Language or IL.

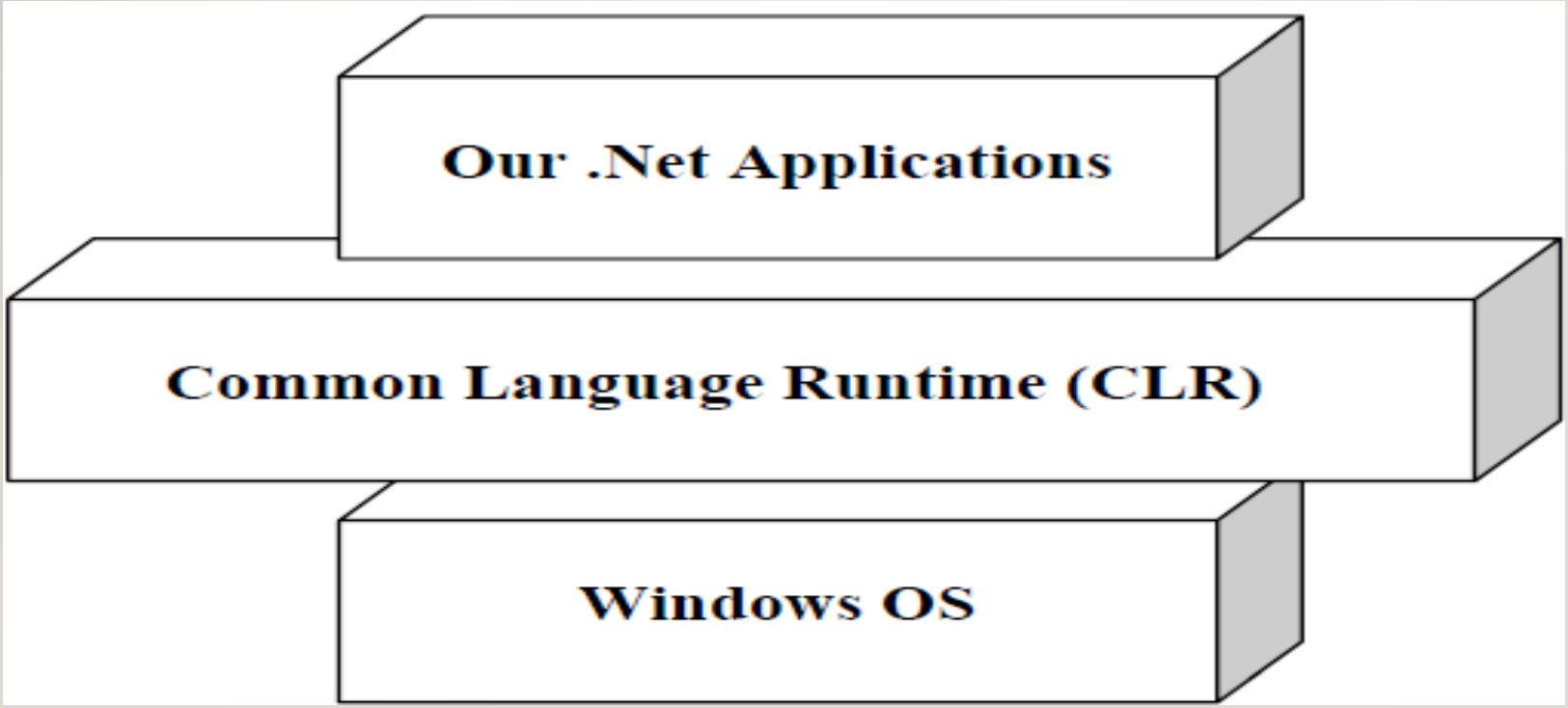
COMMON LANGUAGE RUNTIME(CLR)

- It is the foundation of the .NET framework.
- It is the responsibility of the runtime to take care the code execution of the program.
- It is a framework layer that resides above the OS and handles the execution of all the .Net applications.
- Programs don't directly communicate with the OS but go through the CLR.
- It manages memory, thread execution, code execution, code safety verification, compilation, and other system services.

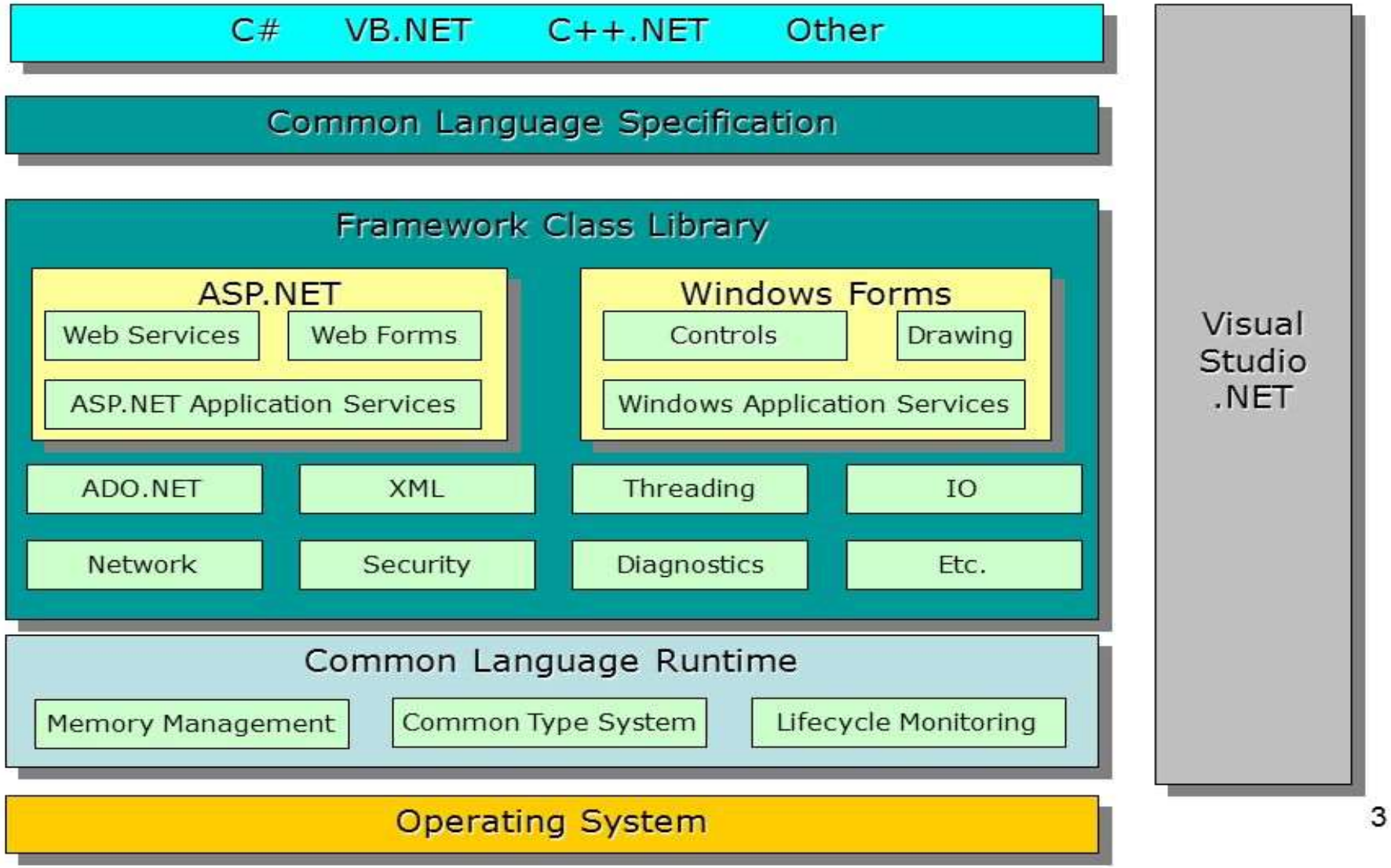
CONT....

- The CLR converts the IL into the native code of the machine, such as X86 or X64, usually just prior to execution.
- This is referred to as Just-In-Time (JIT) compilation.
- Ahead-of-time compilation is also available to improve startup time with large assemblies or resource.
- The container for managed code is called an assembly or portable executable.
- An assembly can be an executable file (.exe) or a library (.dll), and contains not only IL, but type information (metadata).
- The presence of metadata allows assemblies to reference types in other assemblies without needing additional files.

.NET ARCHITECTURE



.NET Framework

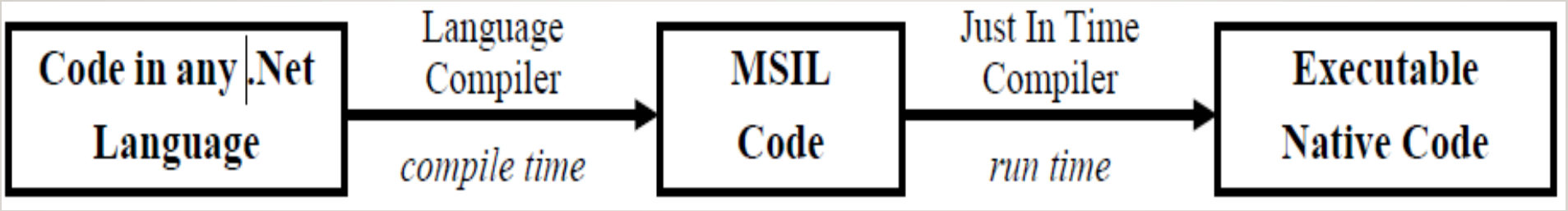


MSIL(MICROSOFT INTERMEDIATE LANGUAGE) OR CIL(COMMON IL) OR IL

- All .NET source code is converted to an intermediate code known as MSIL which is interpreted by the CLR.
- MSIL is OS and hardware independent code.
- MSIL is converted to binary executable code(native code) at the point where the software is installed.

JUST-IN-TIME(JIT) COMPILER

- It compiles the IL code to native executable code(.exe or .dll) that is designed for specific machine and OS.



Framework Class Library

- The .NET Framework consists of the CLR plus a vast set of libraries.
- The core libraries are sometimes collectively called the Base Class Library (BCL). The entire framework is called the Framework Class Library (FCL).
- The .Net Framework class library (FCL) provides the core functionality of .Net Framework architecture.

CONT…

- The .Net Framework Class Library (FCL) includes a huge collection of reusable classes, interfaces, and value types that ease and optimize the development process and provide access to system functionality.
- This library is categorized into different modules and can access to Windows application, Web development, Network programming ,IO etc.

COMMON TYPE SYSTEM(CTS)

- CTS define how types are declared, used and managed in the CLR,
- It is also an important part of the runtime's support for cross-language integration.

The common type system performs the following functions:

i. Establishes a framework that helps enable cross-language integration, type safety, and high-performance code execution.

ii. Provides an object-oriented model that supports the complete implementation of many programming languages.

CONT…

- Defines rules that languages must follow, which helps ensure that objects written in different languages can interact with each other.
- Provides a library that contains the primitive data types (such as Boolean, Byte, Char, Int32, and UInt64) used in application development.

COMMON LANGUAGE SPECIFICATION

- CLS is a set of basic language features that .Net Languages needed to develop Applications and Services.
- It is a subset of the CTS. The CLS establishes the minimum set of rules to promote language interoperability.
- When there is a situation to communicate Objects written in different .Net Complaint languages.
- Those objects must expose the features that are common to all the languages.
- It ensures complete interoperability among applications, regardless of the language used to create the application.

CONT…

- Microsoft has defined CLS, which are nothing but guidelines, that language should follow so that it can communicate with other .NET languages in a seamless manner.

COMPILATION

- The C# compiler compiles source code, specified as a set of files with the .cs extension, into an assembly.
- An assembly is the unit of packaging and deployment in .NET.
- An assembly can be either an application or a library.
- A normal console or Windows application has a Main method and is an .exe file.
- A library is a .dll and is equivalent to an .exe without an entry point.
- Its purpose is to be called upon (referenced) by an application or by other libraries.
- The name of the C# compiler is csc.exe.