Introduction of .NET

.NET Architecture

.NET

designed and developed by Microsoft and the first beta version released in 2000.

The first version of the .Net framework was released in the year 2002

Goals and Applications

- Web Applications

- Windows Applications

- phone Applications

Salient Features

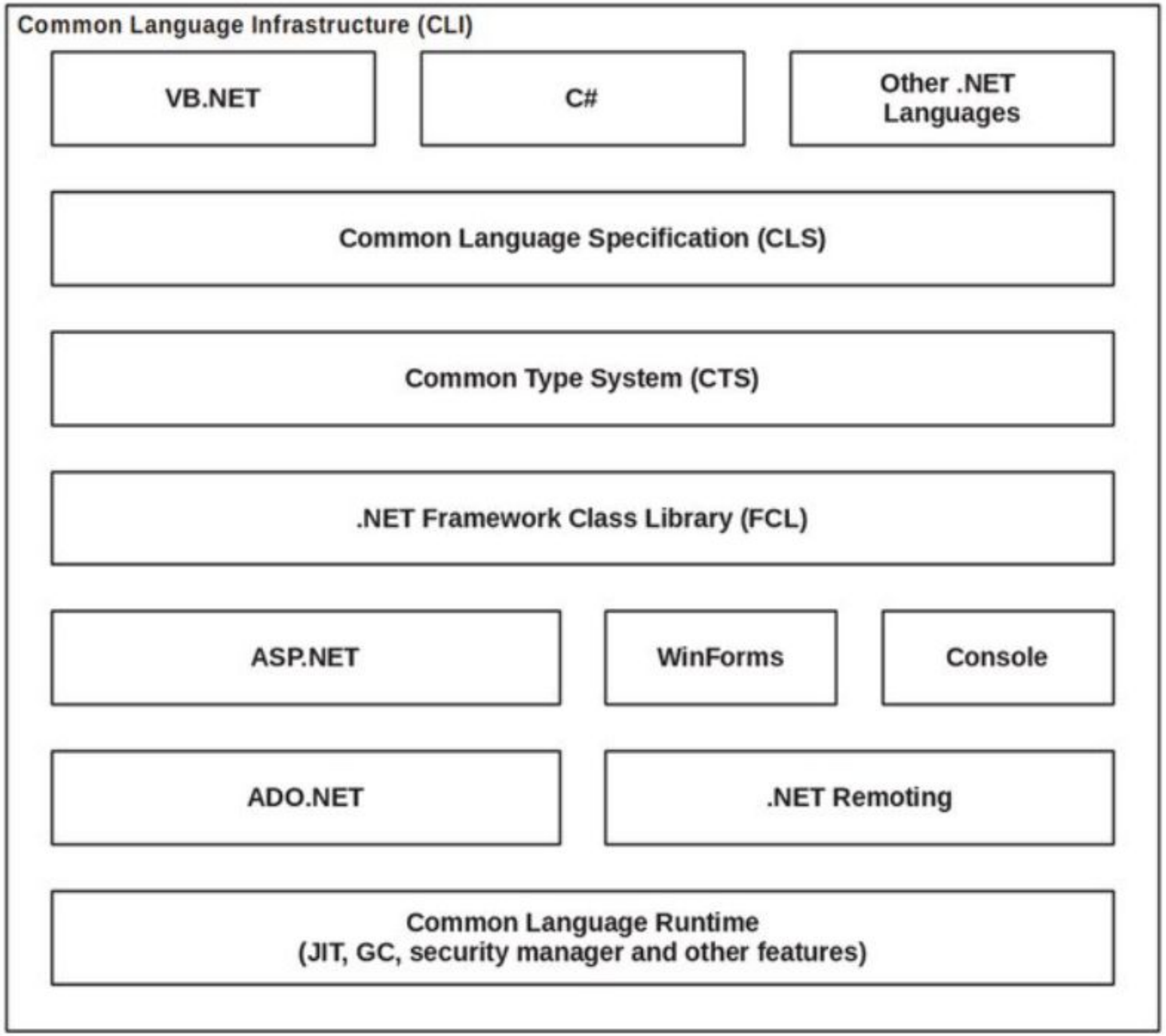
- Less Coding and Increased Reuse of Code
- Reliability
- Security
- Language Interoperability
- Deployment

11 Programming Languages which are designed and developed by Microsoft are:

- C#.NET
- VB.NET
- C++.NET
- J#.NET
- F#.NET
- JSCRIPT.NET
- WINDOWS POWERSHELL
- IRON RUBY
- IRON PYTHON
- C OMEGA
- ASML(Abstract State Machine Language)

IDE

Visual Studio is a very powerful Integrated Development Environment (IDE) where we actually write our C# or .NET programs. It is popular because it supports code editing, interface design, server management, debugging, and performance analysis. We can download Visual Studio for free.



Common Language Infrastructure (CLI)

VB.NET

C#

Other .NET Languages

Common Language Specification (CLS)

Common Type System (CTS)

.NET Framework Class Library (FCL)

ASP.NET

WinForms

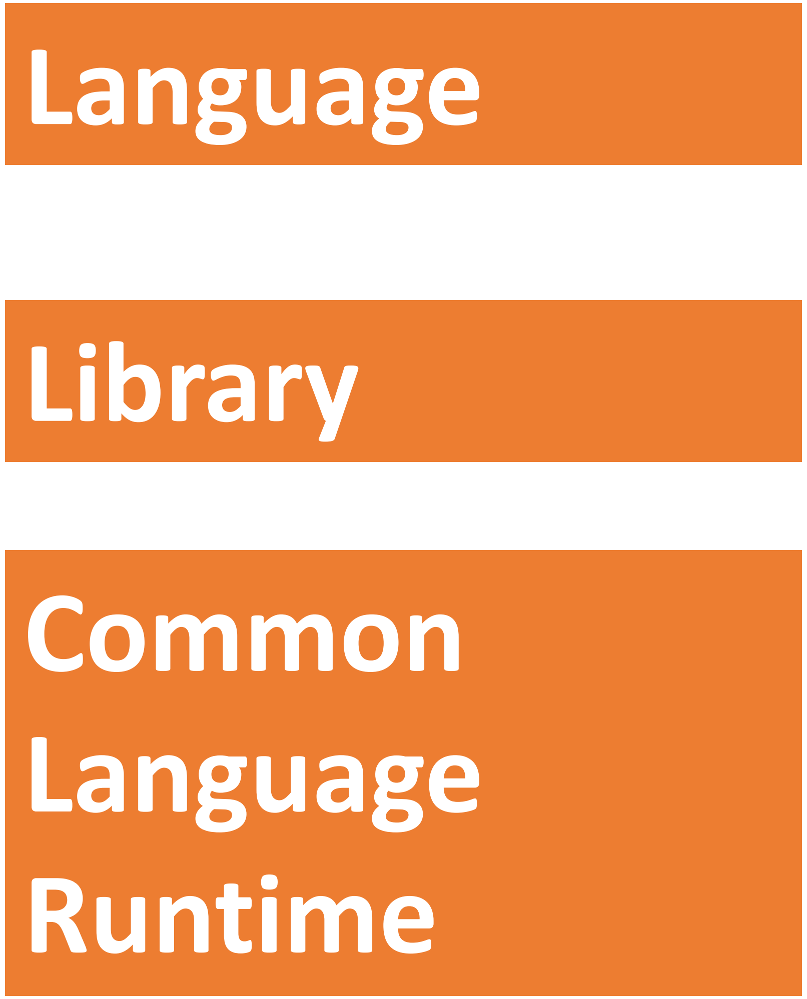
Console

ADO.NET

.NET Remoting

Common Language Runtime
(JIT, GC, security manager and other features)

.NET Architecture



Language

WinForms, ADO.NET, ASP.NET,

Library

Framework Class Libraries (FCL)

Common Language Runtime

Common Language Runtime (Execution Engine)

Language

The types of applications that can be built in the .NET framework is classified broadly into the following categories :

- WinForms This is used for developing Forms-based applications, which would run on an end user machine
- ASP.NET This is used for developing web-based applications, which are made to run on any browser such as Internet Explorer, Chrome or Firefox. (Active Server Page)
- ADO.NET This technology is used to develop applications to interact with Databases such as Oracle or Microsoft SQL Server (ActiveX Data Object)

Library

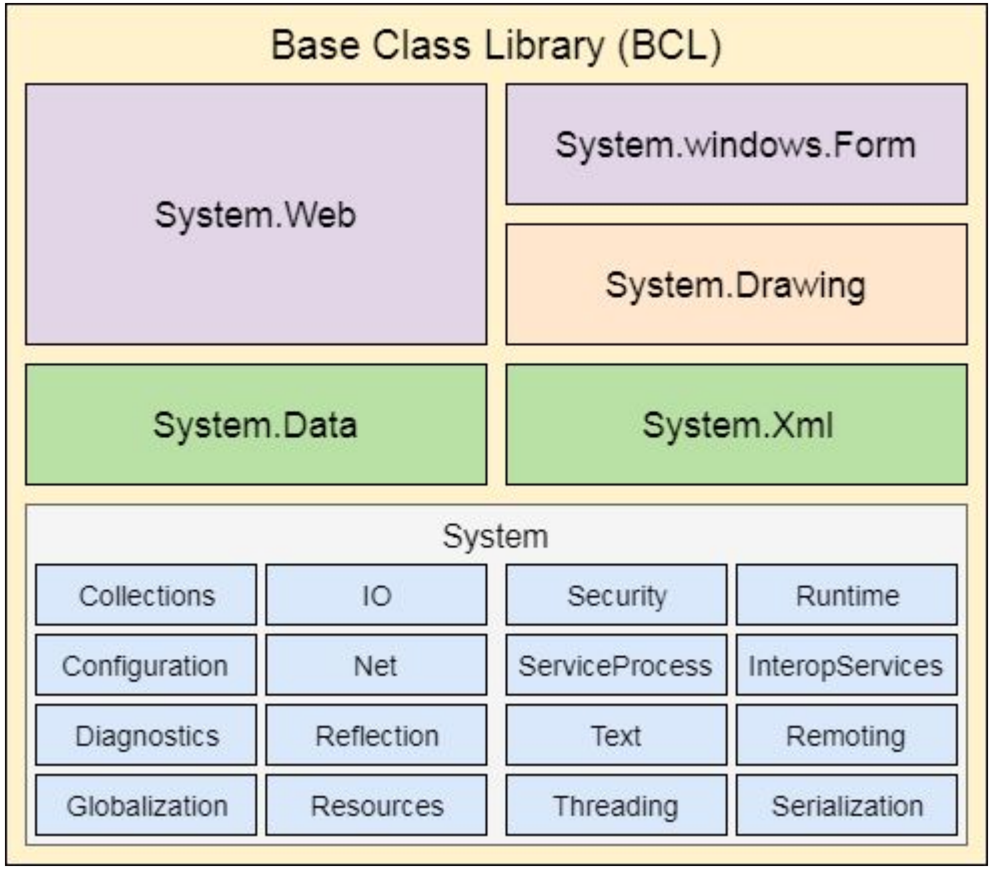
A class library is a collection of methods and functions that can be used for the core purpose.

e.g. -

class library with methods to handle all file-level operations. So there is a method which can be used to read the text from a file. Similarly, there is a method to write text to a file.

Library

FCL (Framework Class Library)



Base Class Library (BCL)

System.Web

System.windows.Form

System.Drawing

System.Data

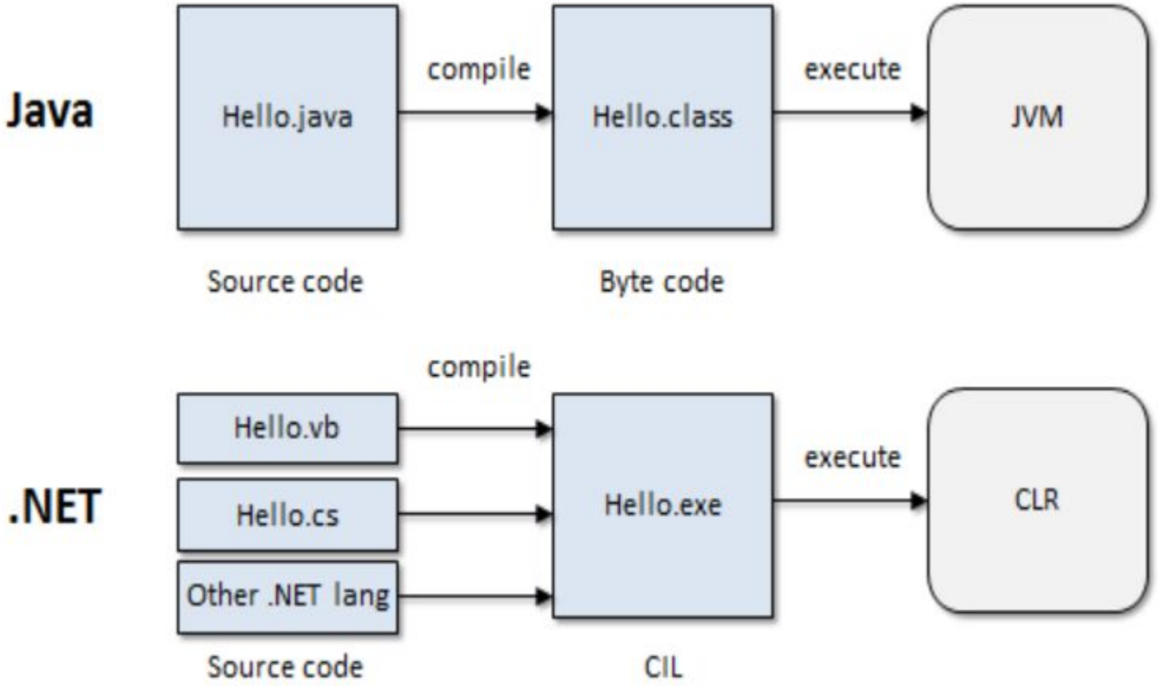
System.Xml

System

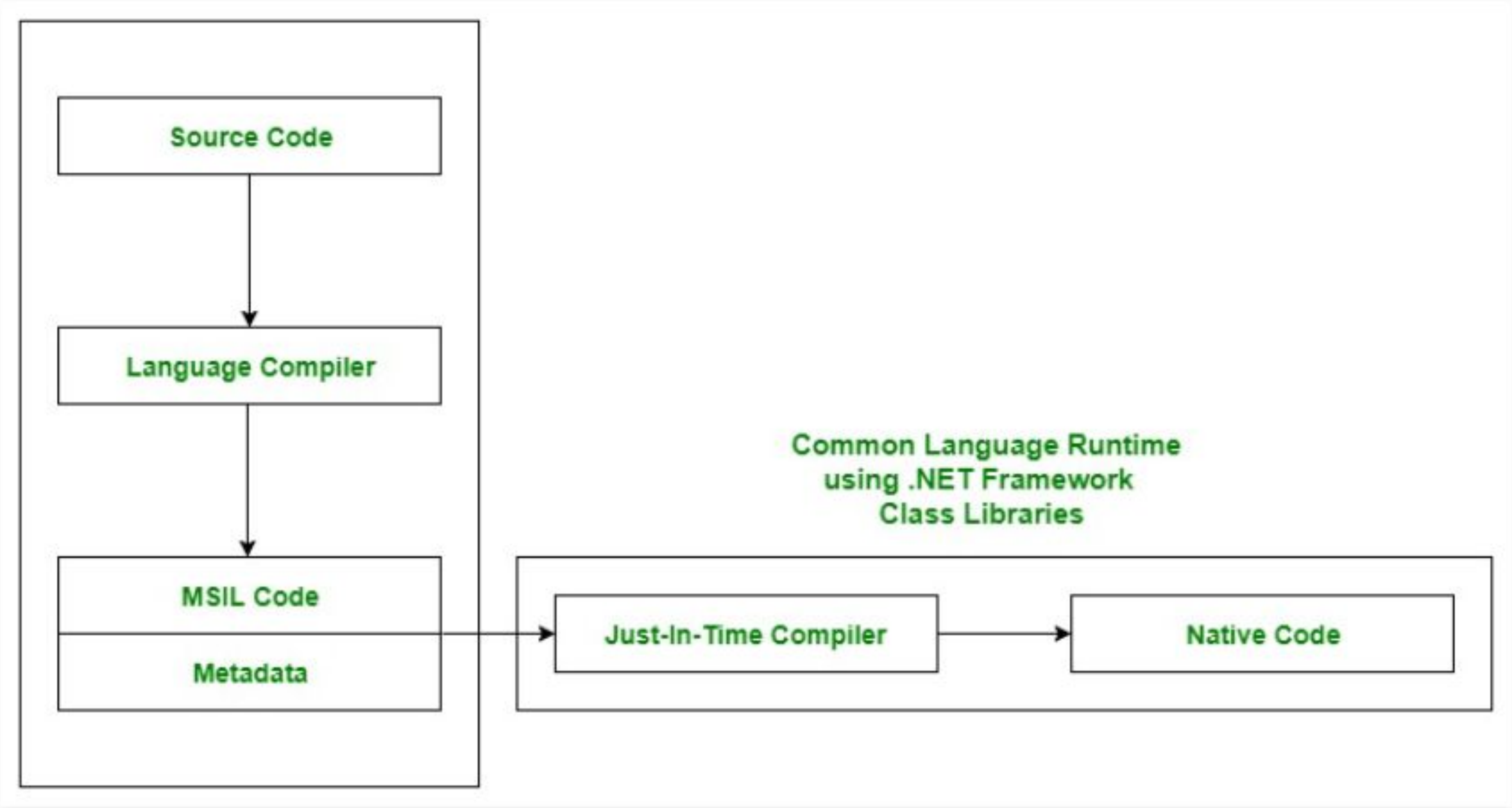
Collections	IO	Security	Runtime
Configuration	Net	ServiceProcess	InteropServices
Diagnostics	Reflection	Text	Remoting
Globalization	Resources	Threading	Serialization

The BCL (Base Class Library) is the core of the FCL and provides basic functionalities.

Common Language Runtime (CLR)



It is a program execution engine that loads and executes the program. It converts the program into native code. It acts as an interface between the framework and operating system.

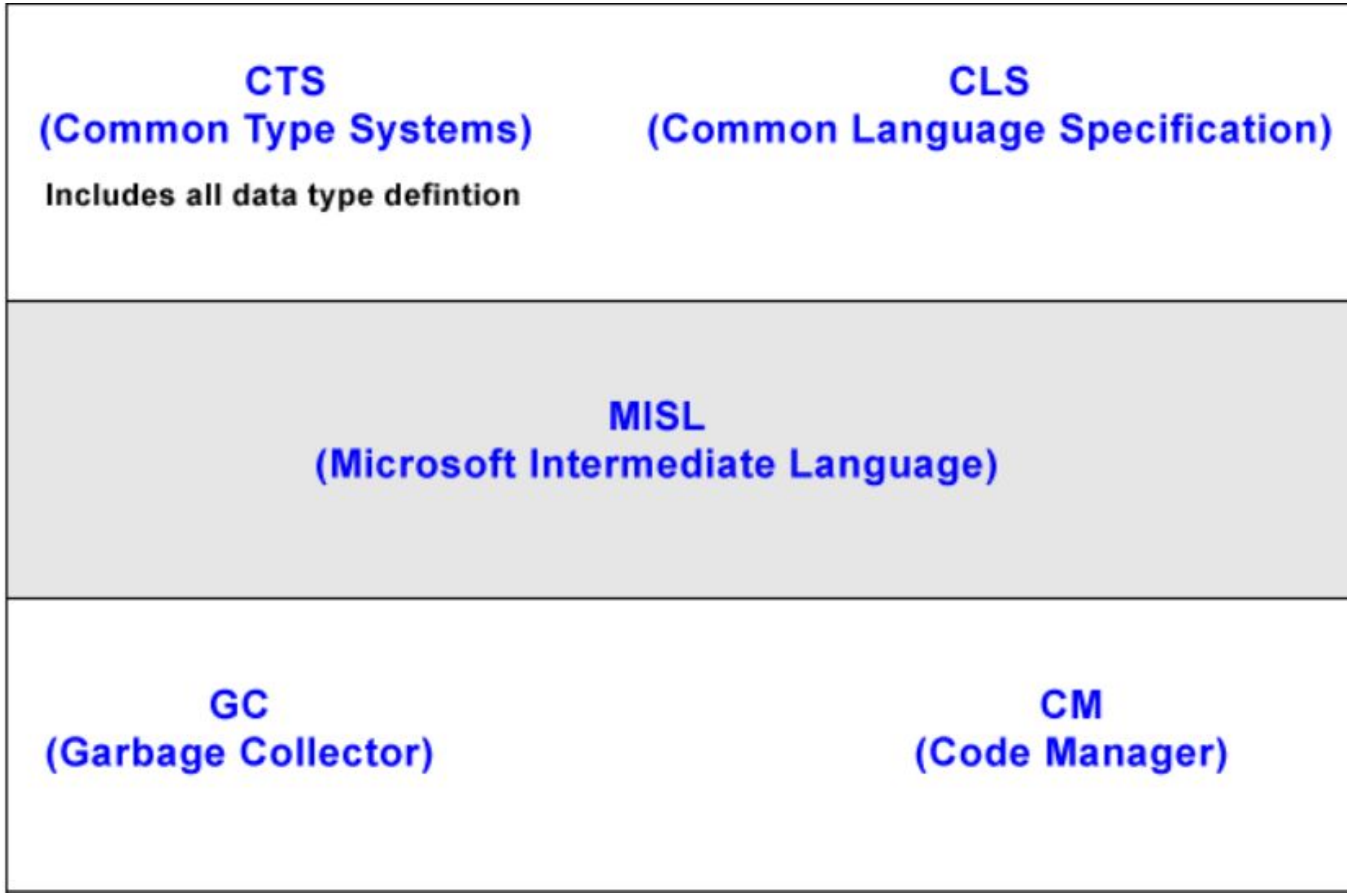

Source Code
Language Compiler
MSIL Code
Metadata
Common Language Runtime
using .NET Framework
Class Libraries
Just-In-Time Compiler
Native Code

- Suppose we have written a C# program and save it in a file.

- Language specific compiler compiles the source code into the ***MSIL(Microsoft Intermediate Language)*** which is also know as the ***CIL(Common Intermediate Language)*** or ***IL(Intermediate Language)*** along with its metadata.
- *Metadata* includes the all the types, actual implementation of each function of the program. MSIL is machine independent code.

- Now CLR comes into existence. CLR provides the services and runtime environment to the MSIL code. Internally CLR includes the JIT(Just-In-Time) compiler which converts the MSIL code to machine code which further executed by CPU.

Architecture of CLR



CTS
(Common Type Systems)

Includes all data type defintion

CLS
(Common Language Specification)

MISL
(Microsoft Intermediate Language)

GC
(Garbage Collector)

CM
(Code Manager)

Common Language Runtime (CLR)

Main components of CLR:

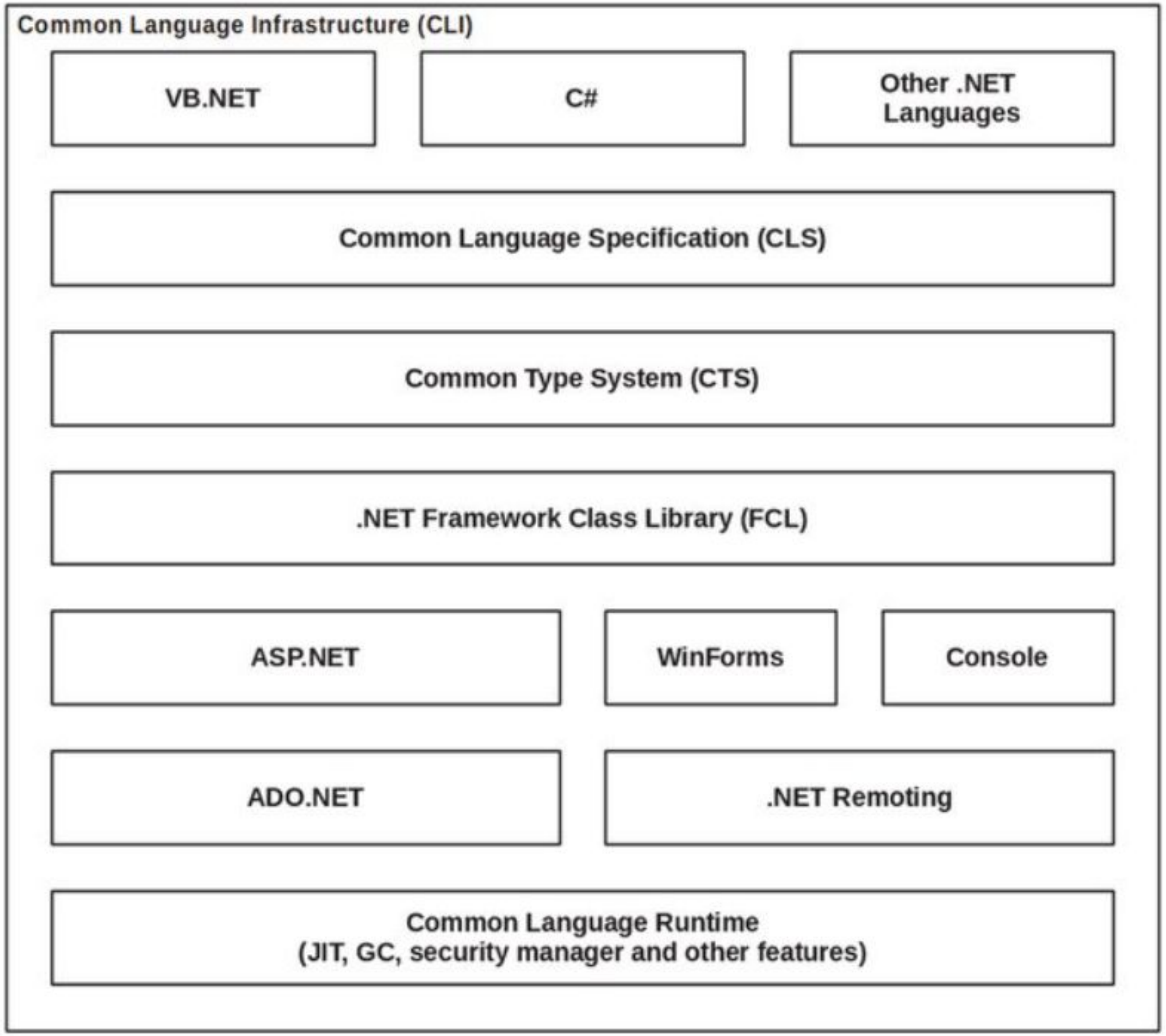
1. Common Language Specification (CLS)
2. Common Type System (CTS)
3. Garbage Collection (GC)
4. Just In – Time Compiler (JIT)

- Common Language Specification (CLS) : It is responsible for converting the different .NET programming language syntactical rules and regulations into CLR understandable format. Basically, it provides the Language Interoperability.

- Common Type System (CTS) :Every programming language has its own data type system, so CTS is responsible for understanding all the data type of system.

- Garbage Collection (GC): It is used to provide the *Automatic Memory Management* feature.

- Just In – Time Compiler (JIT) : It is responsible for converting the CIL(Common Intermediate Language) into machine code or native code using the Common Language Runtime environment.



Common Language Infrastructure (CLI)

VB.NET

C#

Other .NET Languages

Common Language Specification (CLS)

Common Type System (CTS)

.NET Framework Class Library (FCL)

ASP.NET

WinForms

Console

ADO.NET

.NET Remoting

Common Language Runtime
(JIT, GC, security manager and other features)

.Net Framework Design Principle

1. Interoperability
2. Portability
3. Security
4. Memory management
5. Simplified deployment