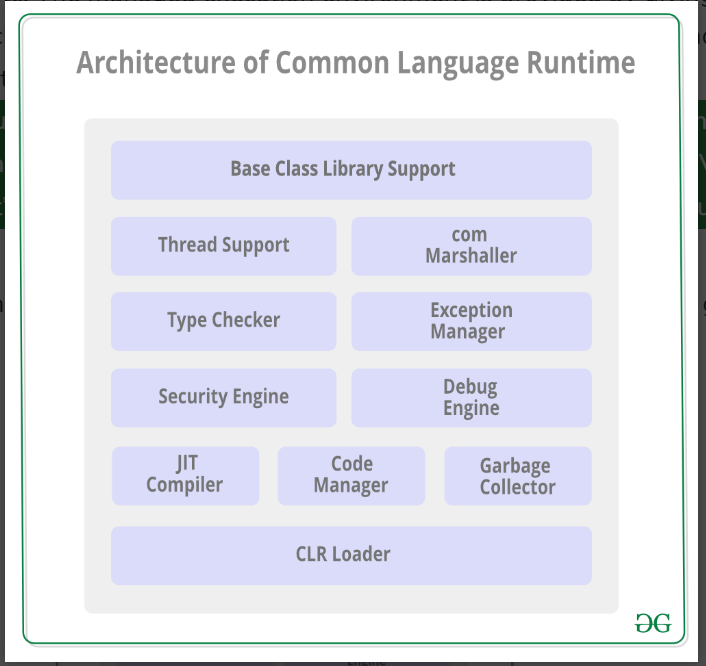
**ARCHITECTURE OF CLR :**

CLR (Common Language Runtime) is a key component of the Microsoft .NET framework. The Common Language Runtime implements the VES (Virtual Execution System) which is a run time system that provides a managed code execution environment. The VES is defined in Microsoft’s implementation of the CLI (Common Language Infrastructure).

It provides several features and services to manage and execute .NET programs. Here are some of the important features of CLR:



1. **Base Class Library Support:**

The Common Language Runtime provides support for the base class library. The BCL contains multiple libraries that provide various features such as *Collections*, *I/O*, *XML*, *DataType definitions*, etc. for the multiple *.NET* programming languages.

1. **Thread Support :**

CLR provides support for multithreading, allowing developers to create and manage threads in their applications. It includes features like thread synchronization and coordination. The *System.Threading class* is used as the base class for this.

**3 .COM Marshaller:**

Communication with the COM (Component Object Model) component in the .NET application is provided using the COM marshaller. This provides the COM interoperability support.

1. **Automatic Memory Management (Garbage Collection):**

CLR includes a garbage collector that automatically manages the memory used by .NET applications. It identifies and collects unused objects, freeing up memory and preventing memory leaks.

1. **Managed Code Execution:**. The code manager in CLR manages the code developed in the .NET framework i.e. the managed code. The managed code is converted to intermediate language by a language-specific compiler and then the intermediate language is converted into the machine code by the Just-In-Time (JIT) compiler.
2. **Security:** CLR enhances the security of .NET applications by providing a robust security model. It includes features like code access security, which restricts the permissions of managed code based on its origin and the permissions granted to it. The security engine in the CLR handles the security permissions at various levels such as the code level, folder level, and machine level.
3. **Cross-language Integration:** CLR supports multiple programming languages within the .NET framework. This enables developers to use different languages and still interoperate seamlessly. It achieves this through the Common Intermediate Language (CIL) that serves as an intermediate representation of code generated by various languages.
4. **Exception Handling:** CLR provides a unified exception-handling mechanism for .NET applications. It allows developers to write code that catches and handles exceptions consistently across different languages.. For a particular application, the catch block of the exceptions are executed in case they occur and if there is no catch block then the application is terminated.
5. **Just-In-Time Compilation (JIT):** CLR uses JIT compilation to convert the Common Intermediate Language (CIL) code into native machine code at runtime. This process helps improve the performance of .NET applications by adapting the code to the specific architecture of the machine on which it is running. The compiled MSIL is stored so that it is available for subsequent calls if required.
6. **Versioning and Side-by-Side Execution:** CLR supports versioning of assemblies, allowing multiple versions of a component to coexist on the same system. This facilitates side-by-side execution of applications that depend on different versions of the same assembly.
7. **Integration with the .NET Framework Class Library (FCL):** CLR seamlessly integrates with the .NET Framework Class Library, providing a rich set of pre-built classes and APIs that developers can use to build applications efficiently.

**11.COM Marshaller:**Communication with the COM (Component Object Model) component in the .NET application is provided using the COM marshaller. This provides the COM interoperability support.

12**.TYPE CHECKER** : Type safety is provided by the type checker by using the Common Type System (CTS) and the Common Language Specification (CLS) that are provided in the CLR to verify the types that are used in an application.

13. **DEBUG ENGINE**: An application can be debugged during the run-time using the debug engine. There are various ICorDebug interfaces that are used to track the managed code of the application that is being debugged.

**14. CLR LOADER:**Various modules, resources, assemblies, etc. are loaded by the CLR loader. Also, this loader loads the modules on demand if they are actually required so that the program initialization time is faster and the resources consumed are lesser.

These features collectively contribute to the reliability, security, and performance of .NET applications running on the CLR.

Top of Form

Bottom of Form

ChatGPT can make mistakes. Consider checking important information.