What is .NET

Microsoft started development on the .NET Framework in the late 1990s originally under the name of "Next Generation Windows Services" (NGWS).

So why did Microsoft choose the name .NET?

What is .NET

- New programming methodology
  - Multiple Languages (VB.Net, C#, J#, Cobol.Net, etc.)
  - JIT Compiler
- Primary Parts:
  - .Net Framework
  - Common Language Runtime (CLR)

Framework

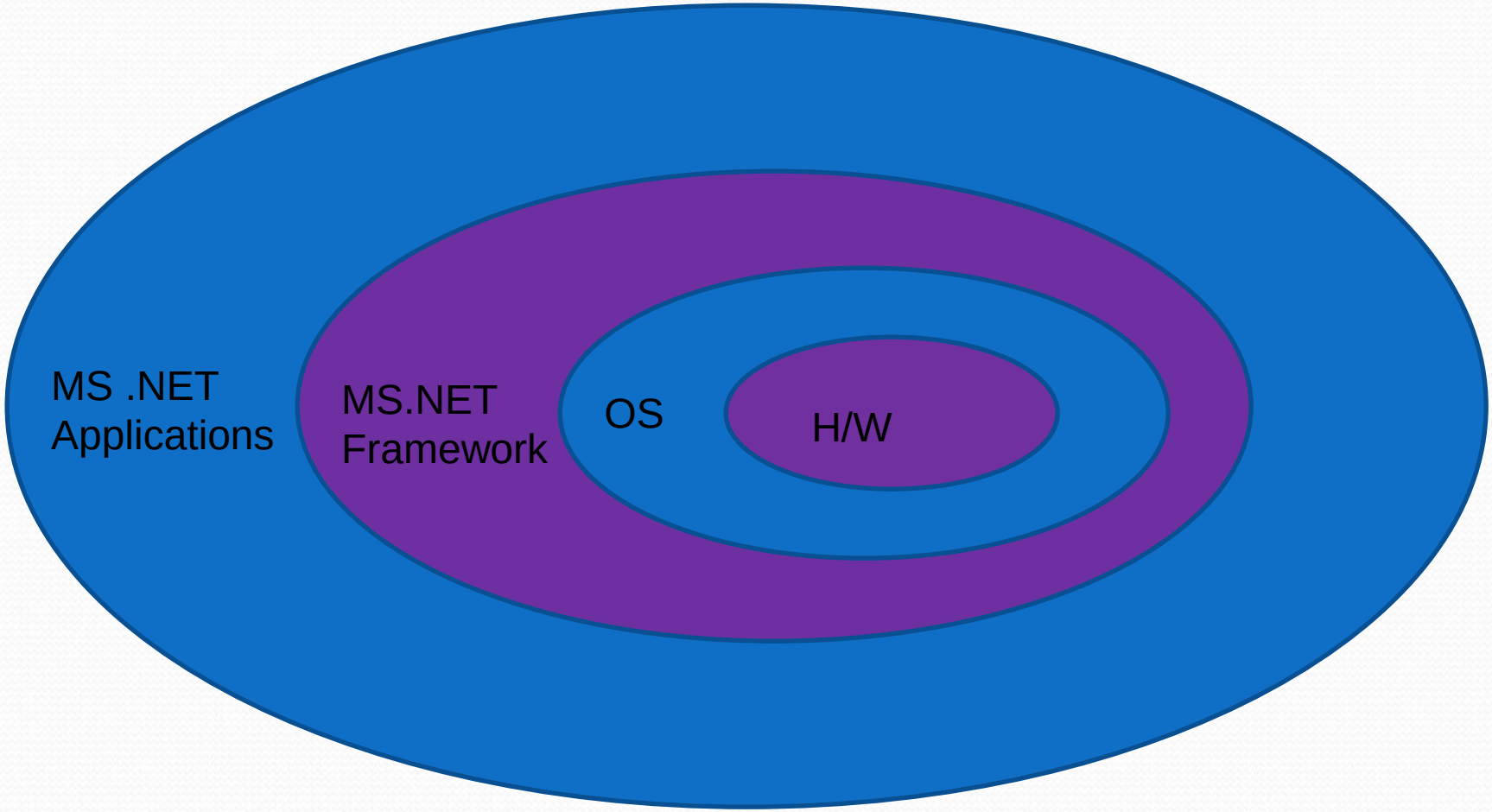
a structure supporting or containing something

A software framework is an abstraction in which software providing generic functionality can be selectively changed by user code, thus providing application specific software.

framework is ready to use collection of classes and interfaces used for developing a particular type of application.

Platform

Platform is an environment for developing and executing applications.



Framework contain key distinguishing features that separate them from normal libraries:

1. Default behavior
2. Extensibility
3. Non-modifiable framework code

Example

1. Application Framework
2. Web application framework

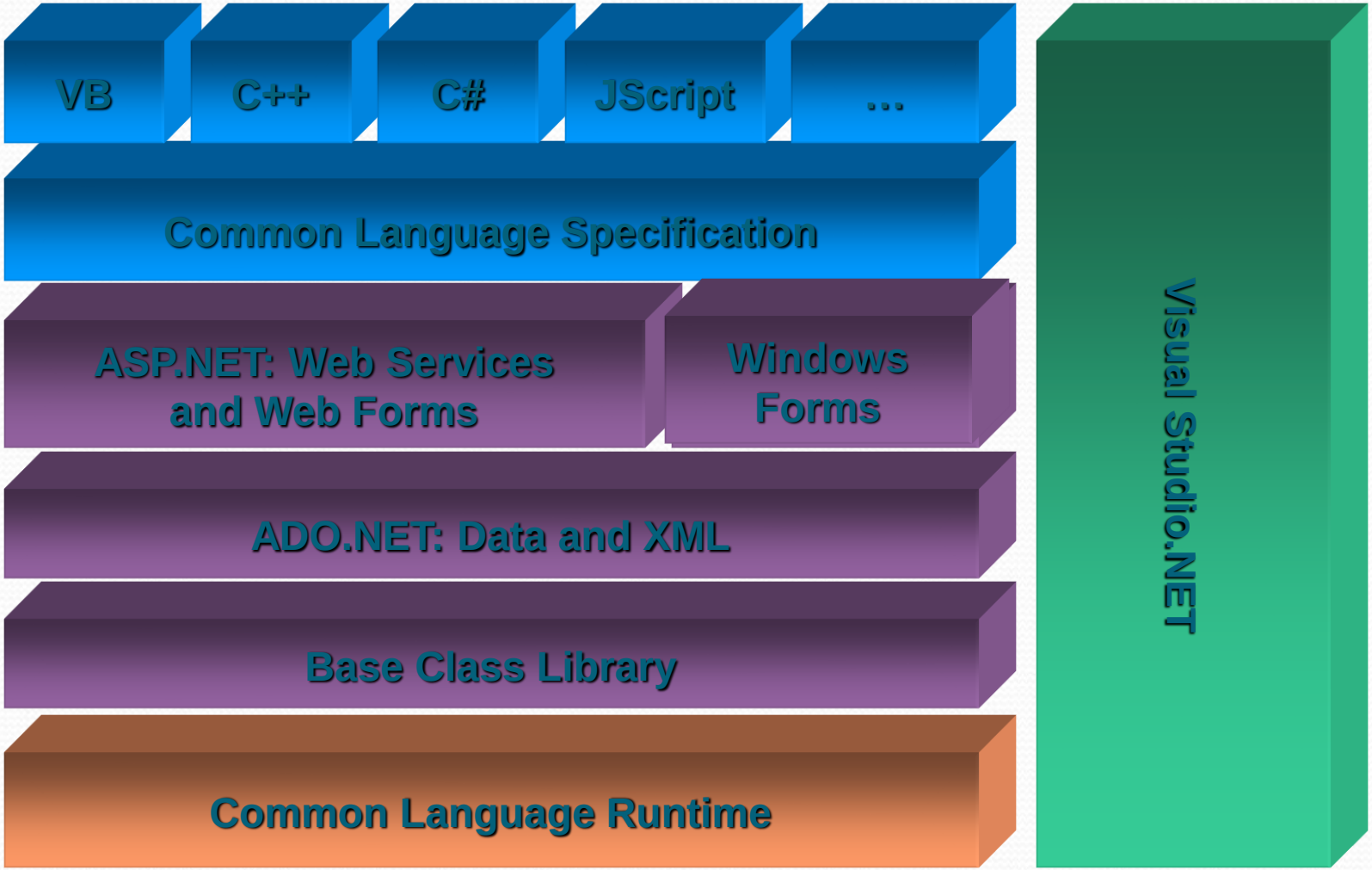
.NET Framework

The .NET Framework is an integral windows component that supports building and running the next generation of application and XML web services.

.NET Framework

- A set of approximately 3500 classes.
- Classes are divided into namespaces grouping similar classes.
- For organization, each class belongs to only one namespace.
- Most classes are lumped into a name space called **System**
  - System.Data: DB access
  - System.XML: reading/writing XML
  - System.Windows.Forms: Forms manipulation
  - System.Net: network communication.

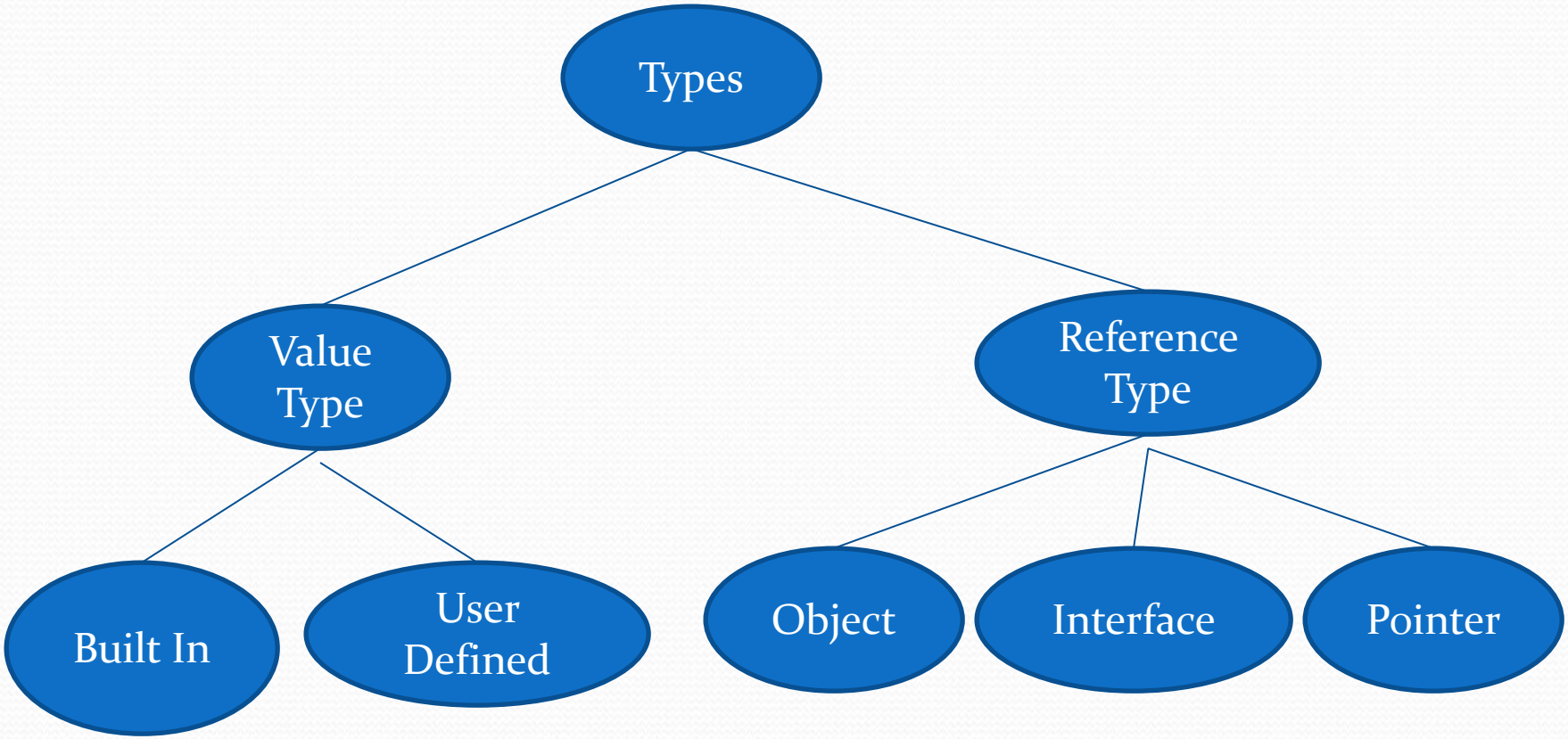
.Net Framework Architecture:-



CLR (Common Language Runtime):- A rich runtime that manages important runtime tasks for the developer including memory management and Garbage Collection (GC). Build around CTS (Common Type System) that defines a common set of data types for all .Net compliant languages.

BCL (Base Class Library):- is a comprehensive object-oriented collection of reusable types that we can use to develop applications ranging from traditional command line or GUI application to application provided by ASP.Net such as web forms and XML web services.

CLS (Common Language Specification):- Defines conventions that language must support in order to be interoperable within .Net.


Types
Value Type
Reference Type
Built In
User Defined
Object
Interface
Pointer

Reference types (Classes)

Reference types are also derived from system.object namespace and may derive directly from object or can be derived from other reference types. Most of the data types we define and use in .Net Framework application are reference types. The CLR supports three kinds of reference types:

1. Classes
2. Interfaces
3. delegates

CLS (Common Language Specification)

To fully interact with other objects regardless of the language they were implemented in, objects must expose to callers only those features that are common to all the languages they must interoperate with. For this reason, the Common Language Specification (CLS), which is a set of basic language features needed by many applications, has been defined.

- It defines rules that range from naming conventions for interface members its rules governing method overloading.
- Consistency in data types is another prerequisites in same.
- CLS is a subset of CTS.

Some standards that CLS puts forth are as under:

1. Public identifiers are case sensitive.
2. Languages must be able to resolve identifiers that are equivalent to keywords.
3. Strict overloading rules
4. Properties and events must follow strict naming rules.

There are about 40 conventions that CLS puts forth.

CLR (Common Language Runtime)

The CLR is the virtual execution system responsible for running all managed code. It offers a broad set of services each of which takes part in the execution of code. Code that runs under the control of CLR is called managed code. Managed code refers to the code that provides the CLR with the information it needs to run the code and which is automatically cleaned up by the GC. Unmanaged code such as COM operator may also run under the CLR via COM interoperable services which generates a .Net wrapper for the unmanaged component. Managed code may also call unmanaged resources such as file devices which works fine although programmers may need to add specific cleanup code to ensure that the resources is properly cleaned up by the GC.

Managed code:- allow the CLR to do the following

- Read metadata that describes the component interface and Types.
- Walk the code Stack
- Handle exceptions
- Retrieve security information

Design Goals of CLR/Managed Code:-

Simplify Development:-

- Define standard that permits code reuse.
- Provide a broad range of services including memory management and GC.

Simplify Deployment:-

- Components use metadata instead of registries.
- Support side by side multiple version
- Command line deployment (Xcopy) for a simple application.

Overview of CLR:-

CLR provides a number of runtime support services using VES (Virtual Execution System). VES is responsible for implementing and enforcing the CTS.

Specific components of VES are:-

1. Class loader
2. MSIL
3. Verification of type safety
4. Stack walker (Services based on Stack Heap)
5. Memory management and GC
6. Profiling and Debugging
7. Com-Interface execution
8. Unmanaged code

Compilation in .NET