

Introduction to .NET Framework

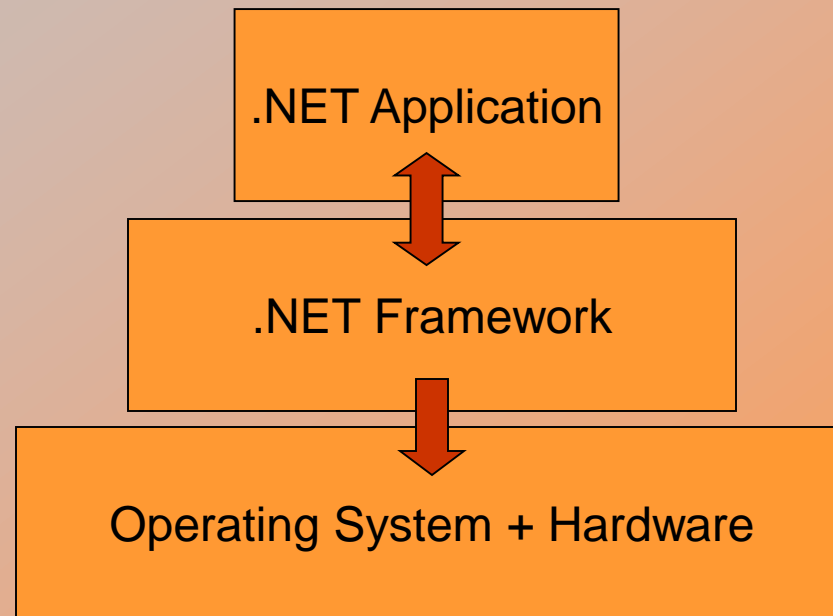
.NET – What Is It?

- Software platform
- Language neutral
- In other words:
 - .NET is not a language (Runtime and a library for writing and executing written programs in any compliant language)

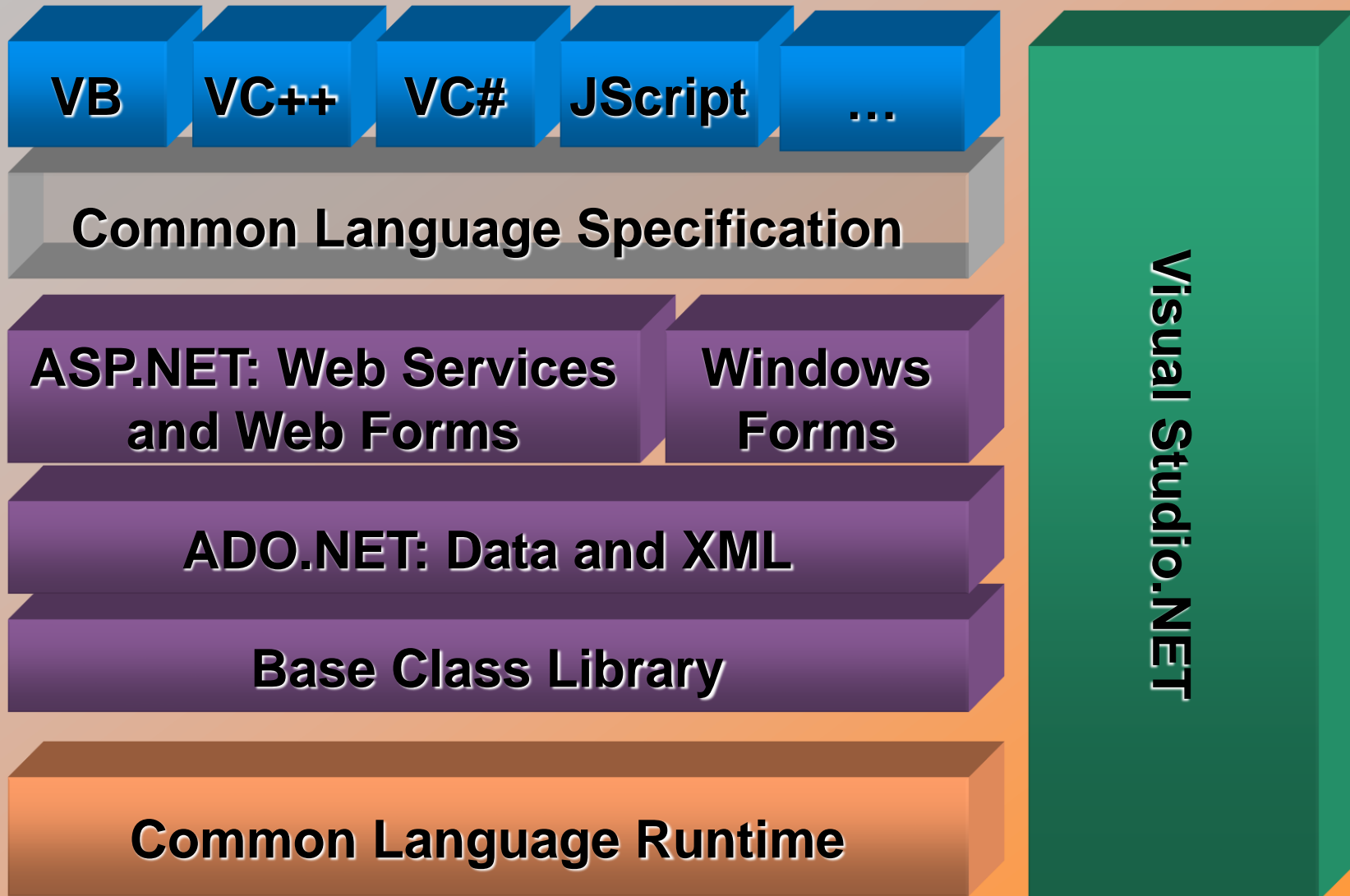
What Is .NET FRAMEWORK

- .Net framework is a platform for developing, deploying, running web-based and windows-based applications within the Microsoft environment.
- The framework offers a fundamental shift in Microsoft strategy: it moves application development from client-centric to server-centric.
- Two Core Components Of .Net Framework are
 1. CLR (Common Language Runtime)
 2. .Net Framework Base Class Libraries

.NET – What Is It?



Framework, Languages, And Tools



The .NET Framework

.NET Framework Services

- Common Language Runtime
- Windows[®] Forms
- ASP.NET
 - Web Forms
 - Web Services
- ADO.NET, evolution of ADO
- Visual Studio.NET

Common Language Runtime (CLR)

- CLR is the core component of .Net Framework-It forms the heart of the .Net Framework.
- Java has JVM (Java Virtual Machine) etc. Similarly .Net has CLR.
- CLR works like a virtual machine in executing all languages.
- All .NET languages must obey the rules and standards imposed by CLR. Examples:
 - Object declaration, creation and use
 - Data types, language libraries
 - Error and exception handling
 - Interactive Development Environment (IDE)

Common Language Runtime

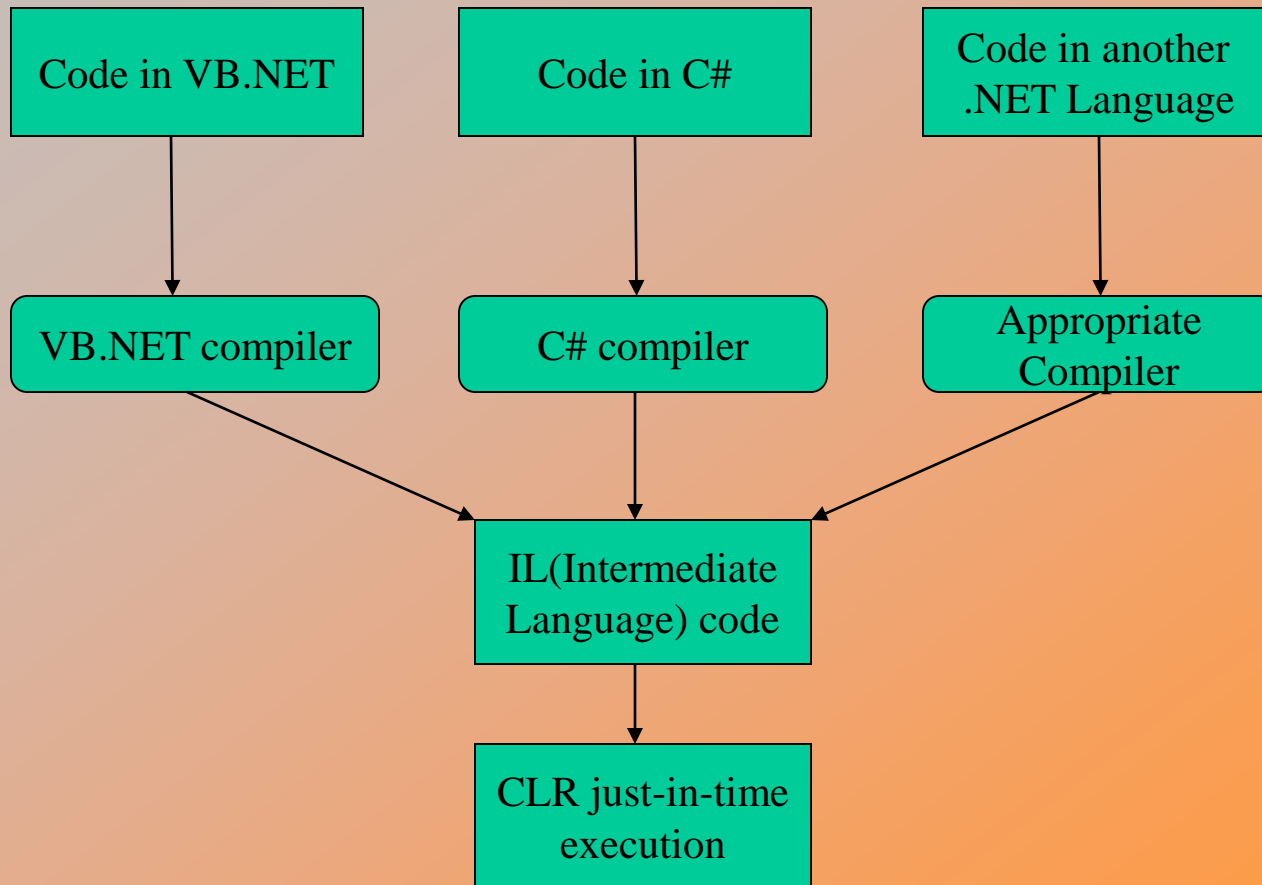
- Development
 - Mixed language applications
 - Common Language Specification (CLS)
 - Common Type System (CTS)
 - Standard class framework
 - Automatic memory management
 - Consistent error handling and safer execution
 - Potentially multi-platform
- Deployment
 - Removal of registration dependency
 - Safety – fewer versioning problems

Common Language Runtime

Multiple Language Support

- CTS is a rich type system built into the CLR
 - Implements various types (int, double, etc)
 - And operations on those types
- CLS is a subset of CTS which all .Net languages are expected to support.
- CLS is a set of specifications that language and library designers need to follow
 - This will ensure interoperability between languages

Compilation in .NET



Intermediate Language (IL)

- .NET languages are not compiled to machine code. They are compiled to an Intermediate Language (IL).
- CLR accepts the IL code and recompiles it to machine code. The recompilation is just-in-time (JIT) meaning it is done as soon as a function or subroutine is called.
- The JIT code stays in memory for subsequent calls. In cases where there is not enough memory it is discarded thus making JIT process interpretive.

Languages

- Languages provided by MS
 - VB, C++, C#, J#, JScript
- Third-parties are building
 - APL, COBOL, Pascal, Eiffel, Haskell, ML, Oberon, Perl, Python, Scheme, Smalltalk...

Windows Forms

- Framework for Building Rich Clients
 - RAD (Rapid Application Development)
 - Rich set of controls
 - Data aware
 - ActiveX[®] Support
 - Licensing
 - Accessibility
 - Printing support
 - Unicode support
 - UI inheritance

ASP.NET

- ASP.NET, the platform services that allow to program Web Applications and Web Services in any .NET language
- ASP.NET Uses .NET languages to generate HTML pages. HTML page is targeted to the capabilities of the requesting Browser
- ASP.NET “Program” is compiled into a .NET class and cached the first time it is called. All subsequent calls use the cached version.

ASP.NET

- Logical Evolution of ASP
 - Supports multiple languages
 - Improved performance
 - Control-based, event-driven execution model
 - More productive
 - Cleanly encapsulated functionality

ASP.NET Web Forms

- Allows clean cut code
 - Code-behind Web Forms
- Easier for tools to generate
- Code within is compiled then executed
- Improved handling of state information
- Support for ASP.NET server controls
 - Data validation
 - Data bound grids

ASP.NET Web Services

- A technical definition
 - “A programmable application component accessible via standard Web protocols”

Web Services

- It is just an application...
- ...that exposes its features and capabilities over the network...
- ...using XML...
- ...to allow for the creation of powerful new applications that are more than the sum of their parts...

ADO.NET

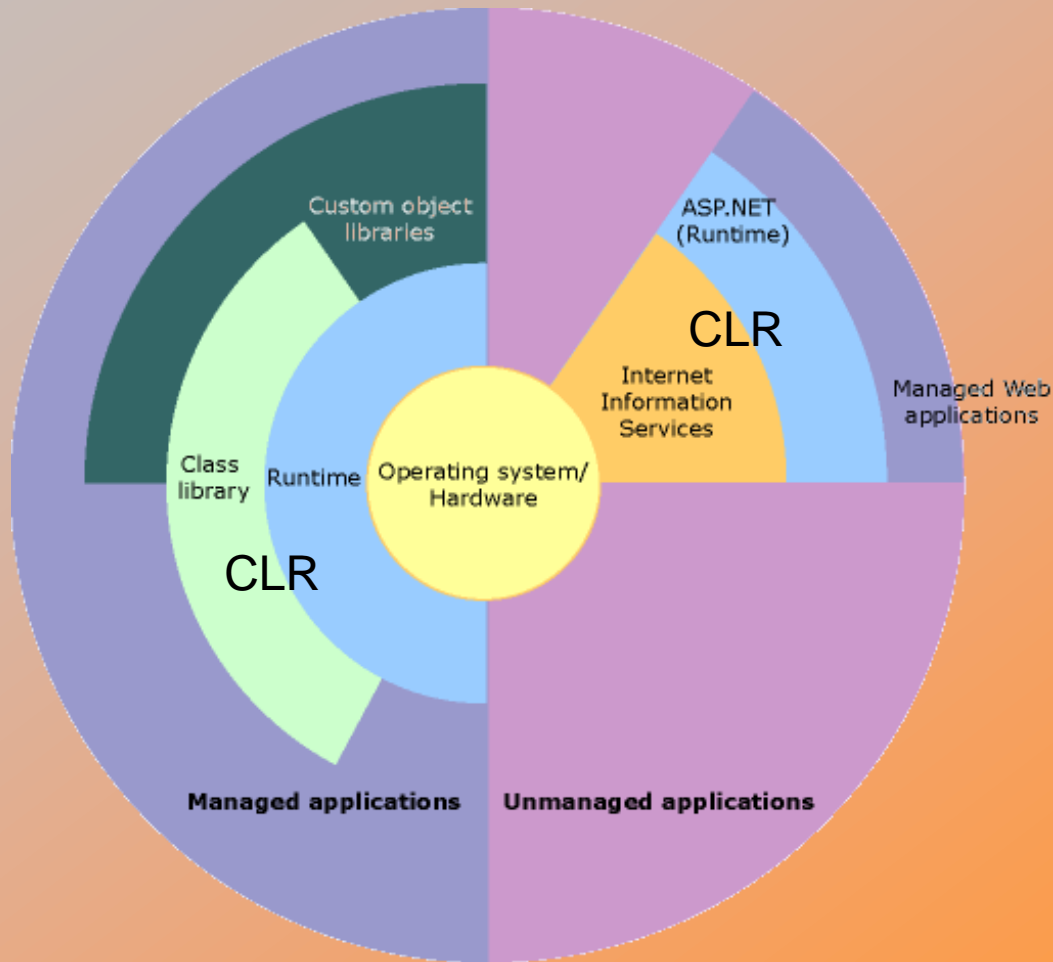
(Data and XML)

- New objects (e.g., DataSets)
- Separates connected / disconnected issues
- Language neutral data access
- Uses same types as CLR
- Great support for XML

Visual Studio.NET

- Development tool that contains a rich set of productivity and debugging features

.NET – Hierarchy, Another View

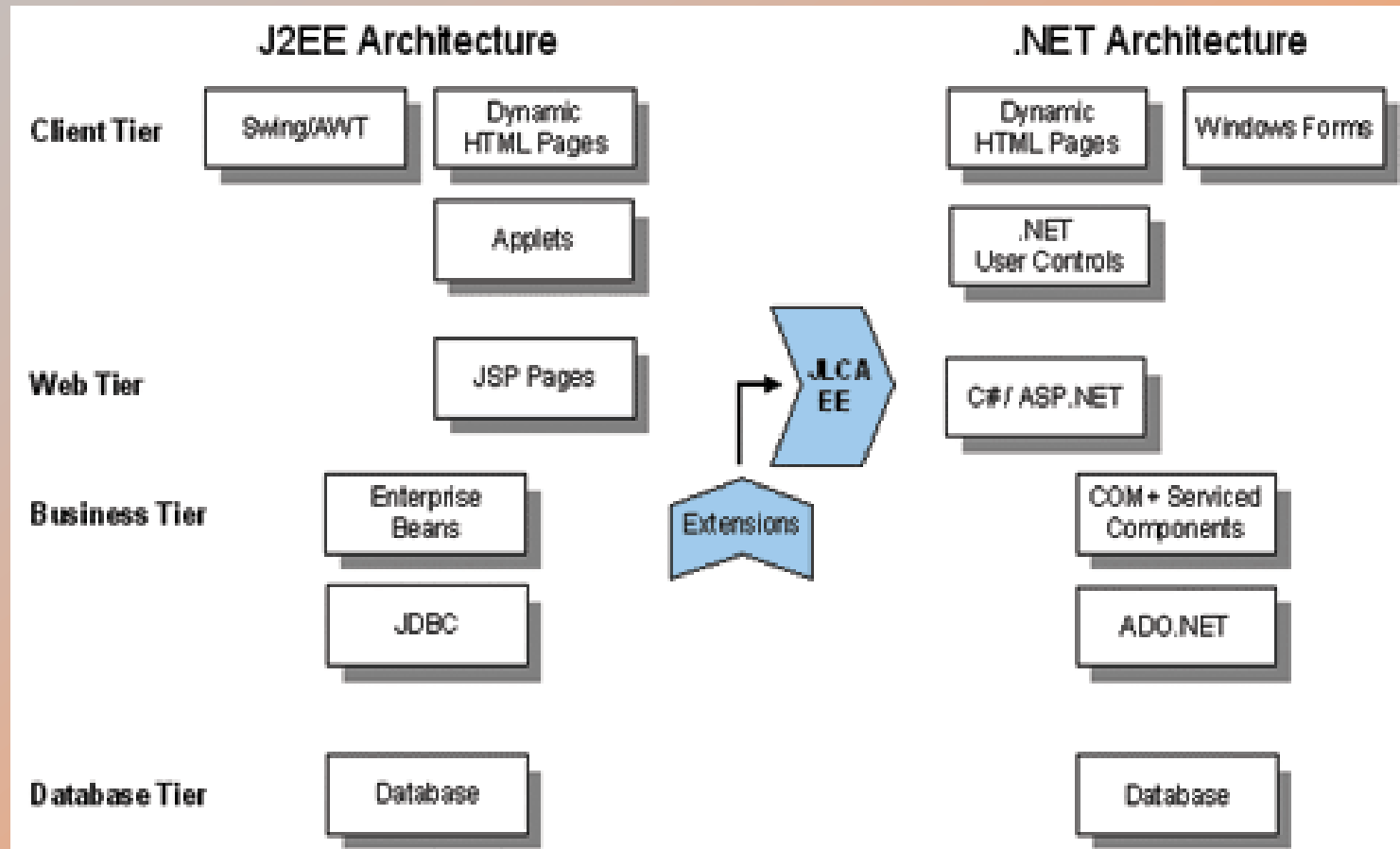


Summary

- The .NET Framework
 - Dramatically simplifies development and deployment
 - Provides robust and secure execution environment
 - Supports multiple programming languages

Comparison between J2EE and .NET

Comparison between J2EE and .NET Architectures



J2EE and .NET

Execution Engine

➤ J2EE

Java source code compiles into machine-independent byte code

Runtime Environment : JVM

➤ .NET

Any compliant language compiles into MSIL

Runtime environment : CLR

Both JVM and CLR ,support services, such as code verification, memory management via garbage collection, and code security

J2EE and .NET

Cross Platform Portability

➤ J2EE

Platform Independent

JDK should exist on target machine

➤ .NET

Supports Windows platform

CLR should exist on target machine

Can support other platforms provided it has its own JIT compiler

J2EE and .NET

Language Support

➤ J2EE

Tied to Java

Supports other languages via interface technology

➤ .NET

Language independent

Supports any language if mapping exists from that language to IL

J2EE and .NET

Tools Support

➤ J2EE

Can employ any number of tools

Pro :Developer has a great deal of choice

Con :Difficulty in choosing a right tool for a given job

➤ .NET

Visual Studio.NET, single IDE for building an application