



# Overview of .NET Framework

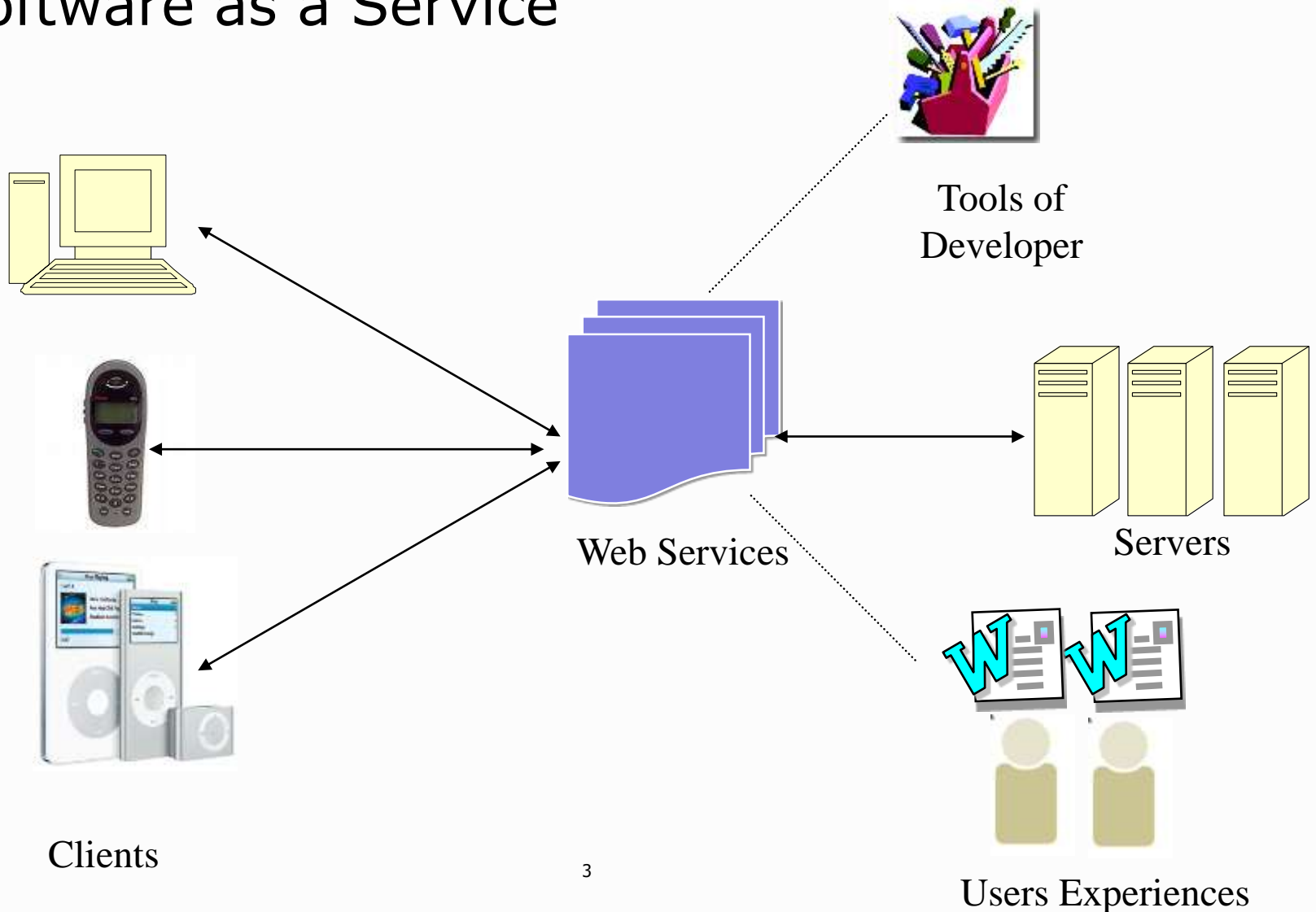
# Objectives

---

- At the end of this session, you will be able to:
  - ◆ List the features of .NET framework.
  - ◆ Describe the components of .NET framework.
  - ◆ Describe flow of execution for .NET application.
  - ◆ Define an assembly and list its features.
  - ◆ Name the types of Assembly.
  - ◆ List the new features in framework 3.0
  - ◆ Use Visual Studio.NET as IDE tool.

# Enterprise Application

- Software as a Service



# Building an Enterprise Application

---

- ♦ Would my application be able to support a set of thousand more users?
- ♦ Would the service be available for 24x7x365 ?
- ♦ Would the response time suffer if user load increase?
- ♦ Would the data integrity be maintained?
- ♦ How easy it would be to fiddle around the components of a running system?
- ♦ Would the hardware resource requirements increase day by day with increasing user load?
- ♦ Would my application be able to get integrated with external applications smoothly?
- ♦ Will anybody be able to hack , steal or corrupt my confidential data?
- ♦ And many more ...

# Quality of Service Requirements

---

- Scalability
- Availability
- Performance
- Flexibility
- Security
- Re-usability

# **.Net Infrastructure**

---

Consist of all technologies that helps in creating and running Robust,Scallable,and Distributed Application

# .Net Framework

---

It is a collection of service and classes it exist as a layer between the .net application and the underlying OS it fully supports

**Class, Inheritance, Methods, Properties, Events, Polymorphism, Constructor** and other OOP Constructs

# Why .Net ?

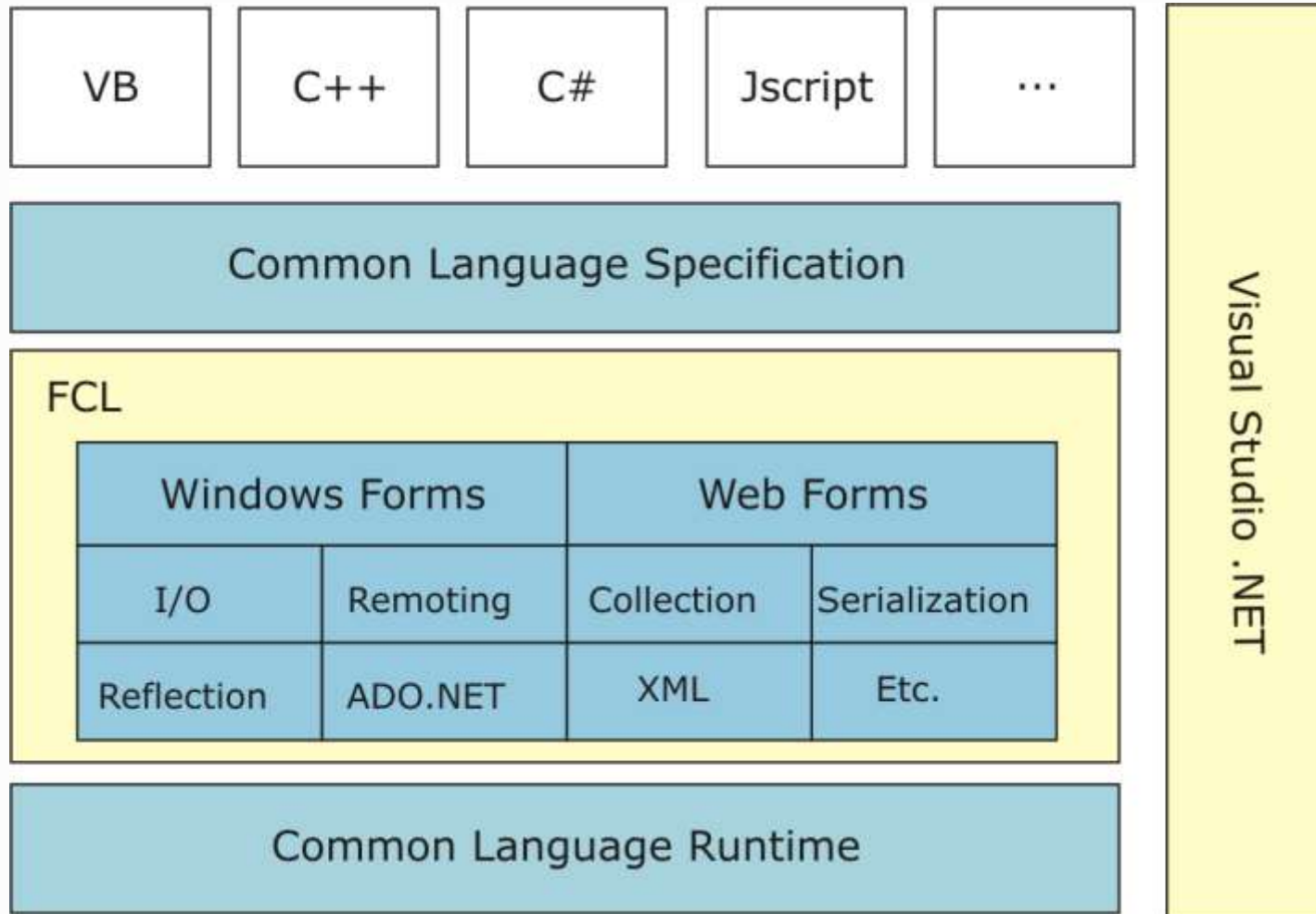
---

1. Fully based on Object Oriented programming Principles.
2. Good Design-a base Class Library.
3. Language independence.
4. Better support for Dynamic Web Pages.
5. Efficient Data Access
6. Code Sharing Via Assemblies traditionally Known as DLL.
7. Improved Security Features.
8. Zero Impact Installation-Shared & Private Assemblies.

9. Support for Web services (can be developed



# .NET Framework



# .NET Languages

---

- Key features of .NET Languages
  - ◆ Standardized by ECMA
  - ◆ Component based
  - ◆ Object-oriented
  - ◆ Robust and durable code
- New languages created for .NET i.e. C#, VB.NET, J#, .....etc. (more than 25)

# Common Language Specification (CLS)

---

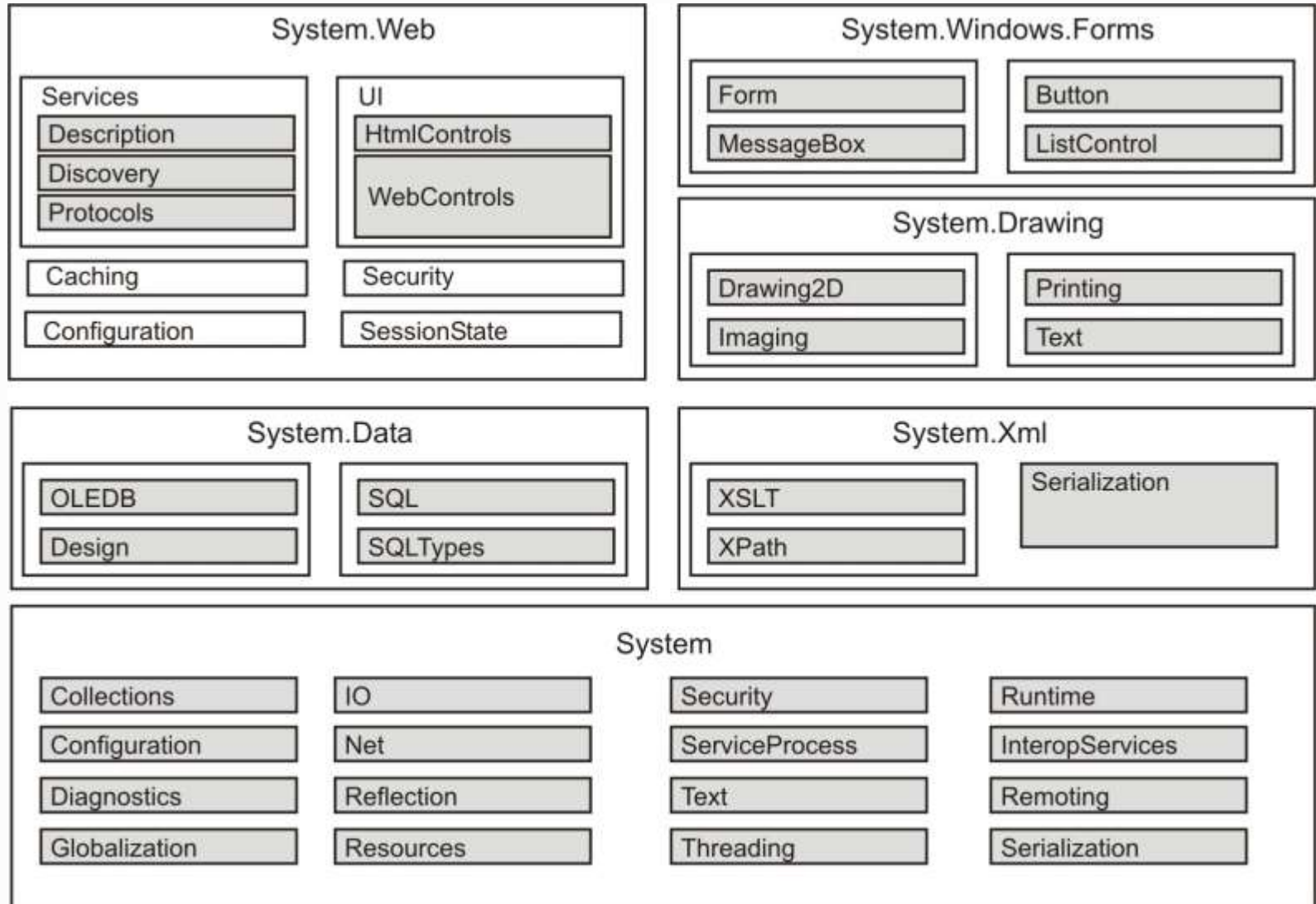
- Ensures source code compiled by a .NET compiler interoperates with .NET Framework.
- CLS describes a set of features that different languages have in common.
- The CLS includes a subset of the Common Type System (CTS).
- The CTS defines the rules concerning data types.
- Any program which uses CLS-compliant types can interoperate with any .NET program written in any language.
- .NET languages support both CLS and CTS.

# Common Type System (CTS)

---

- Enables cross-language integration, type safety, and high performance code execution.
- Provides an object-oriented model where all types are derived from a common class, `System.Object`
- Supports two main categories :
  - ♦ Value types
  - ♦ Reference types

# Framework Class Library (FCL)



# Windows Forms

---

- Framework for building rich clients
- Rich set of controls
- Extensible controls
- Data-aware
- ActiveX support
- Printing support
- Advanced graphics (GDI+)

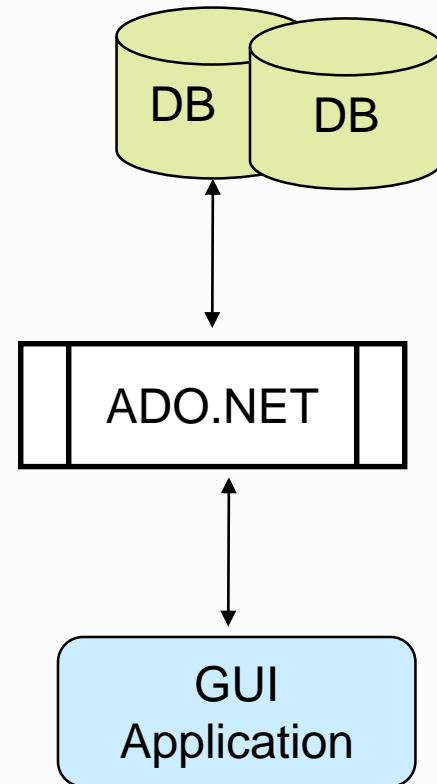
# Web Forms

- Built with ASP.NET
  - ◆ Logical evolution of Active Server Page
  - ◆ Similar development model: edit the page and go
  - ◆ Requires less code
- New programming model
  - ◆ Event-driven/server-side controls
  - ◆ Rich controls (e.g. data grid, validation)
  - ◆ Data binding
  - ◆ Simplified handling of page state



# ADO.NET

- Object model for data access
- Supports two styles of data access
  - ◆ Connected
  - ◆ Disconnected
- Supports data binding
- Can view and process data relationally (tables) or hierarchically (XML)



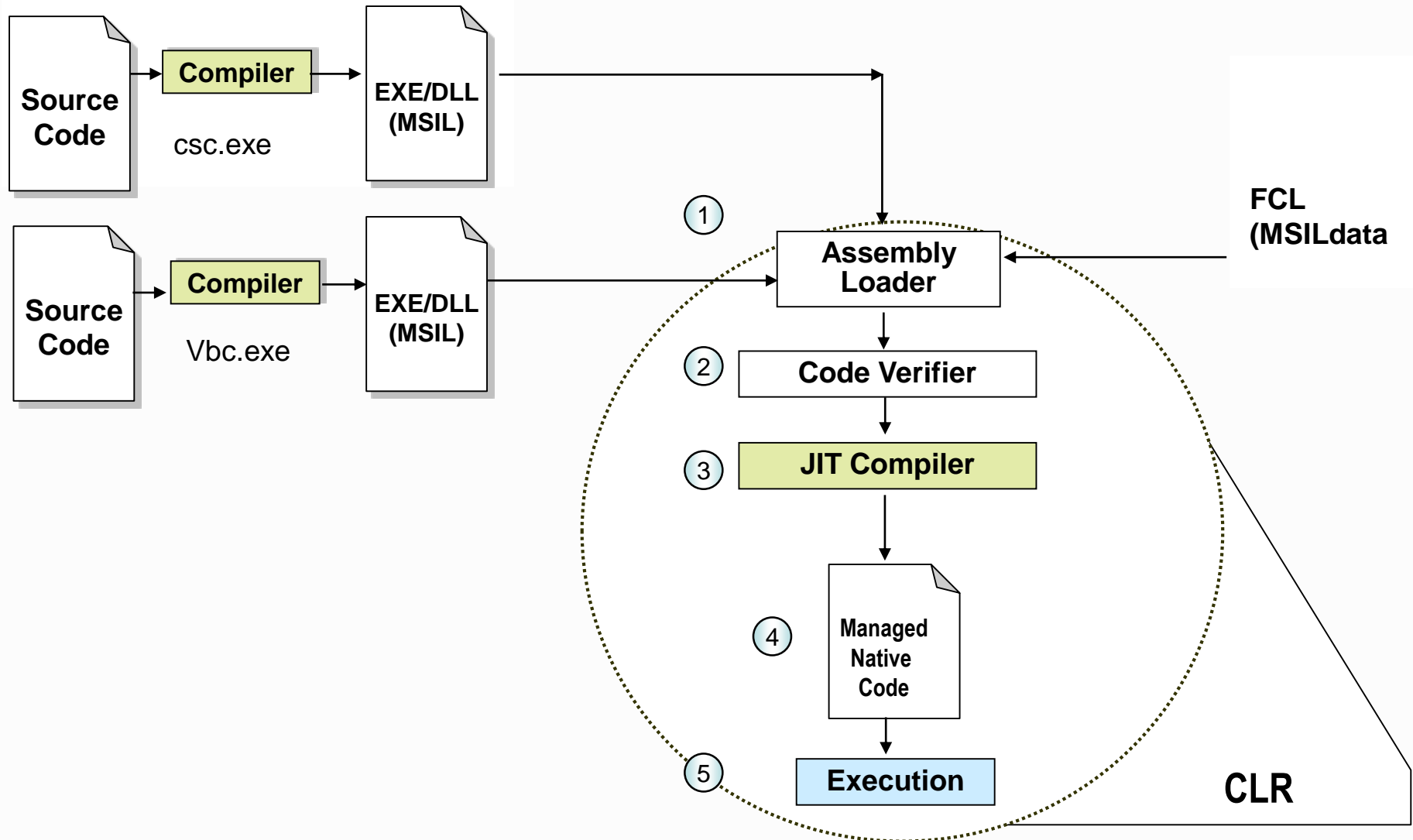


# Common Language Runtime (CLR)

---

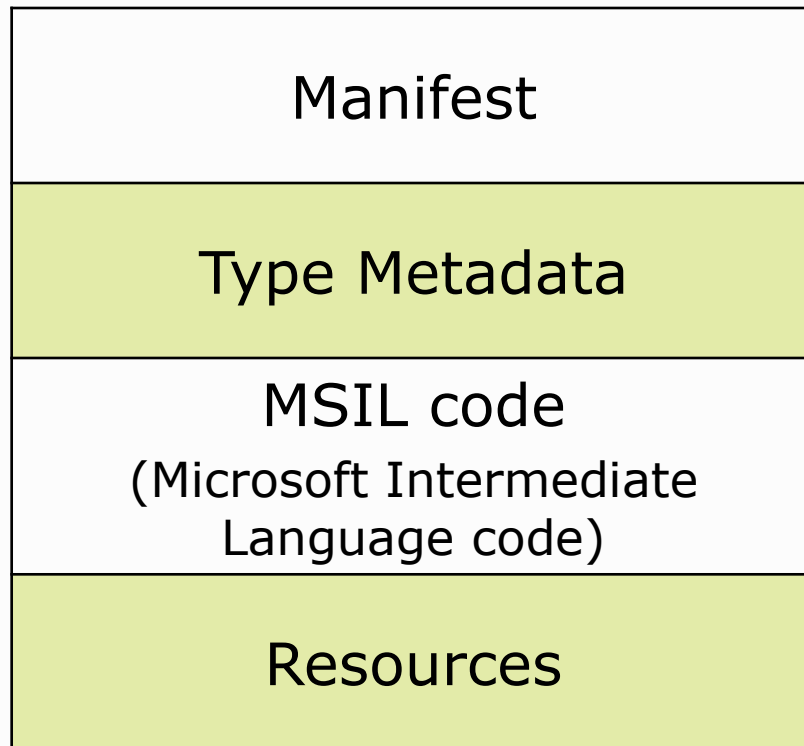
- Execution engine for managed code (MSIL code)
- Services of CLR
  - ◆ Secure Execution of code
  - ◆ Automatic Memory Management
  - ◆ Interoperability
  - ◆ Exception Handling

# Execution of .NET Application

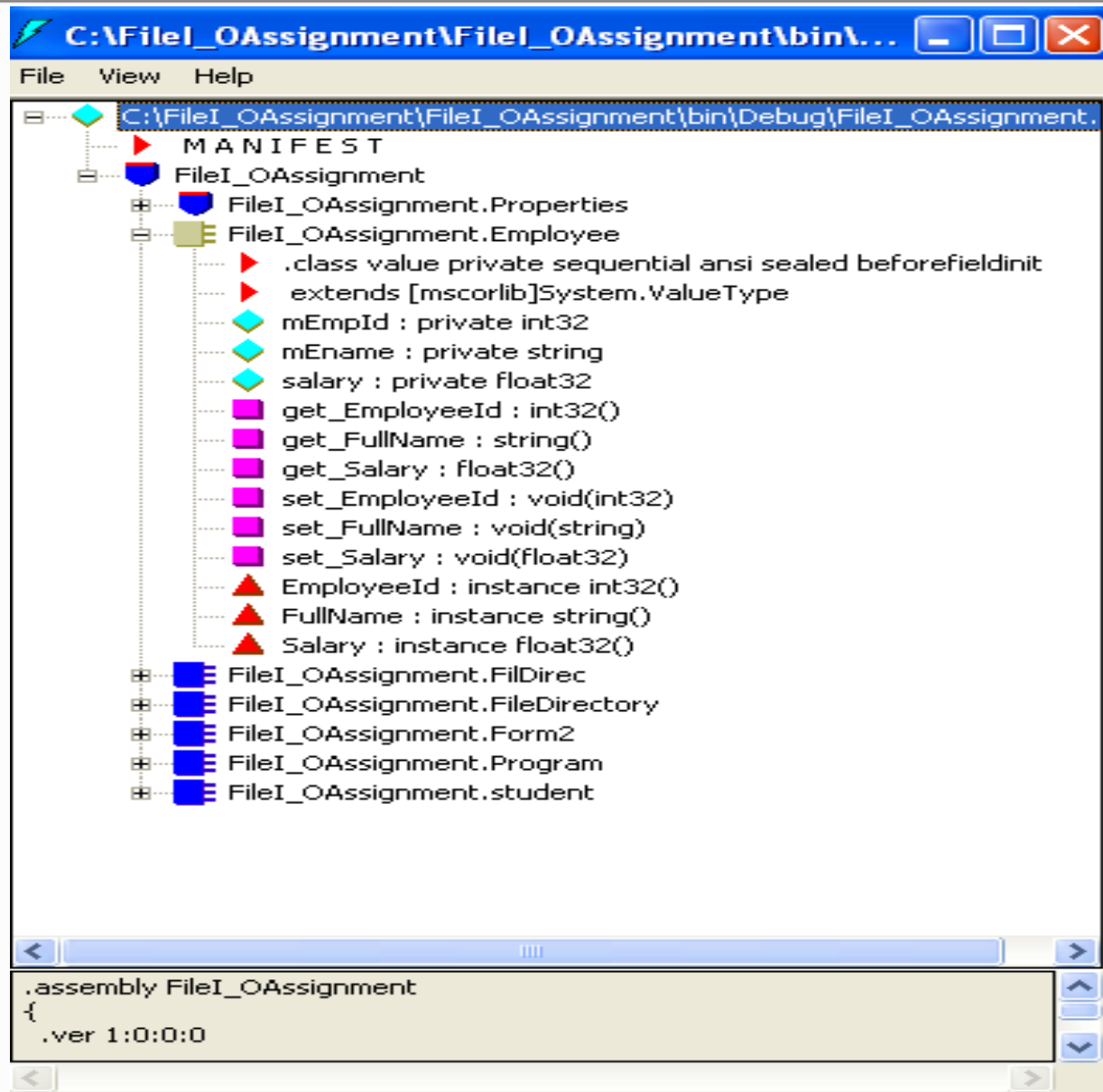


# Assembly

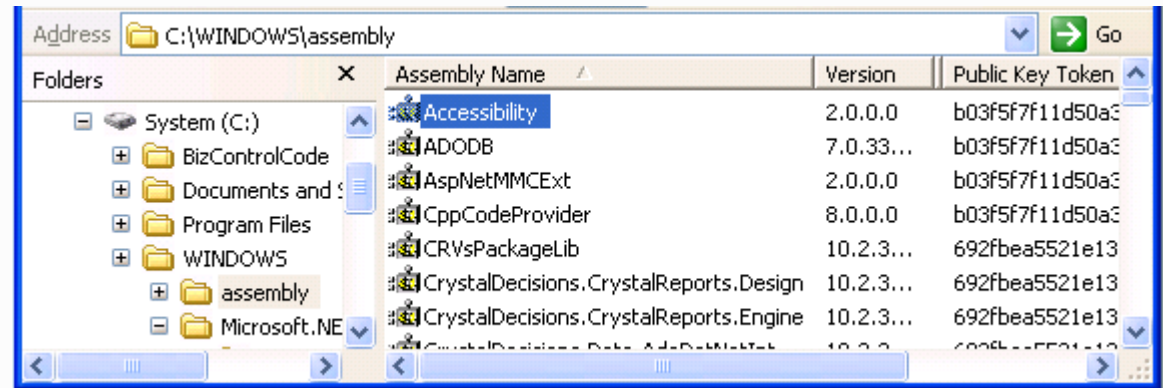
- Assembly is the logical unit of deployment.
- Main Components of Assembly



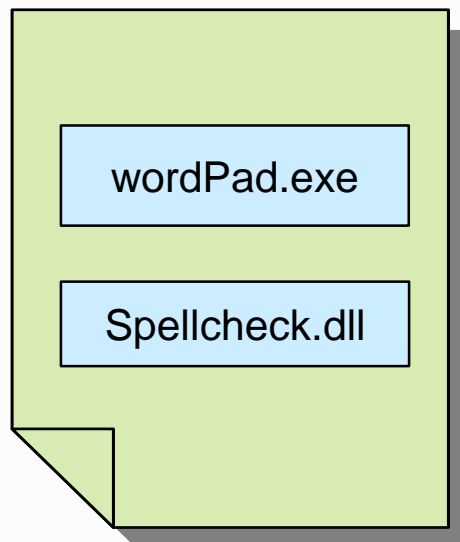
# Disassembly



# Types of Assembly

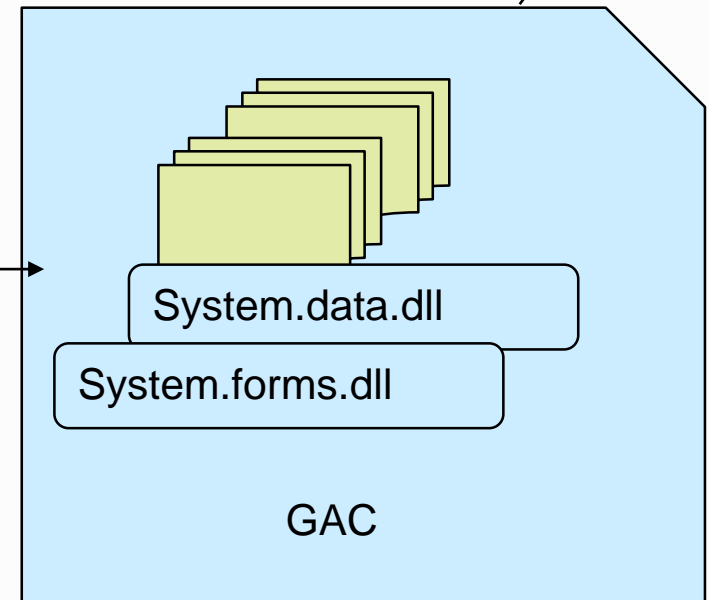


Assembly Name	Version	Public Key Token
Accessibility	2.0.0.0	b03f5f7f11d50a3e
ADODB	7.0.33...	b03f5f7f11d50a3e
AspNetMMCEnt	2.0.0.0	b03f5f7f11d50a3e
CppCodeProvider	8.0.0.0	b03f5f7f11d50a3e
CRVsPackageLib	10.2.3...	692fba5521e13
CrystalDecisions.CrystalReports.Design	10.2.3...	692fba5521e13
CrystalDecisions.CrystalReports.Engine	10.2.3...	692fba5521e13
CrystalDecisions.ReportAppServer.WebAdaptorLib	10.2.3...	692fba5521e13



Shared  
Assembly

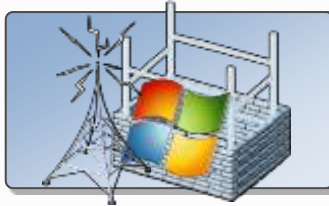
Private  
Assembly



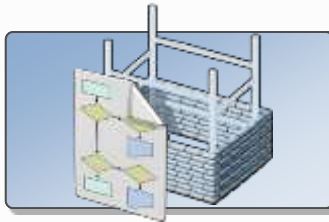
# Introduction to the .NET Framework 3.0 Technologies



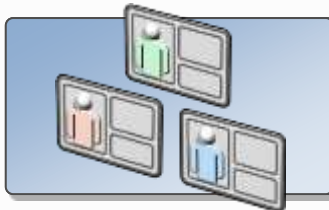
**Windows Presentation Foundation**



**Windows Communication Foundation**



**Windows Workflow Foundation**



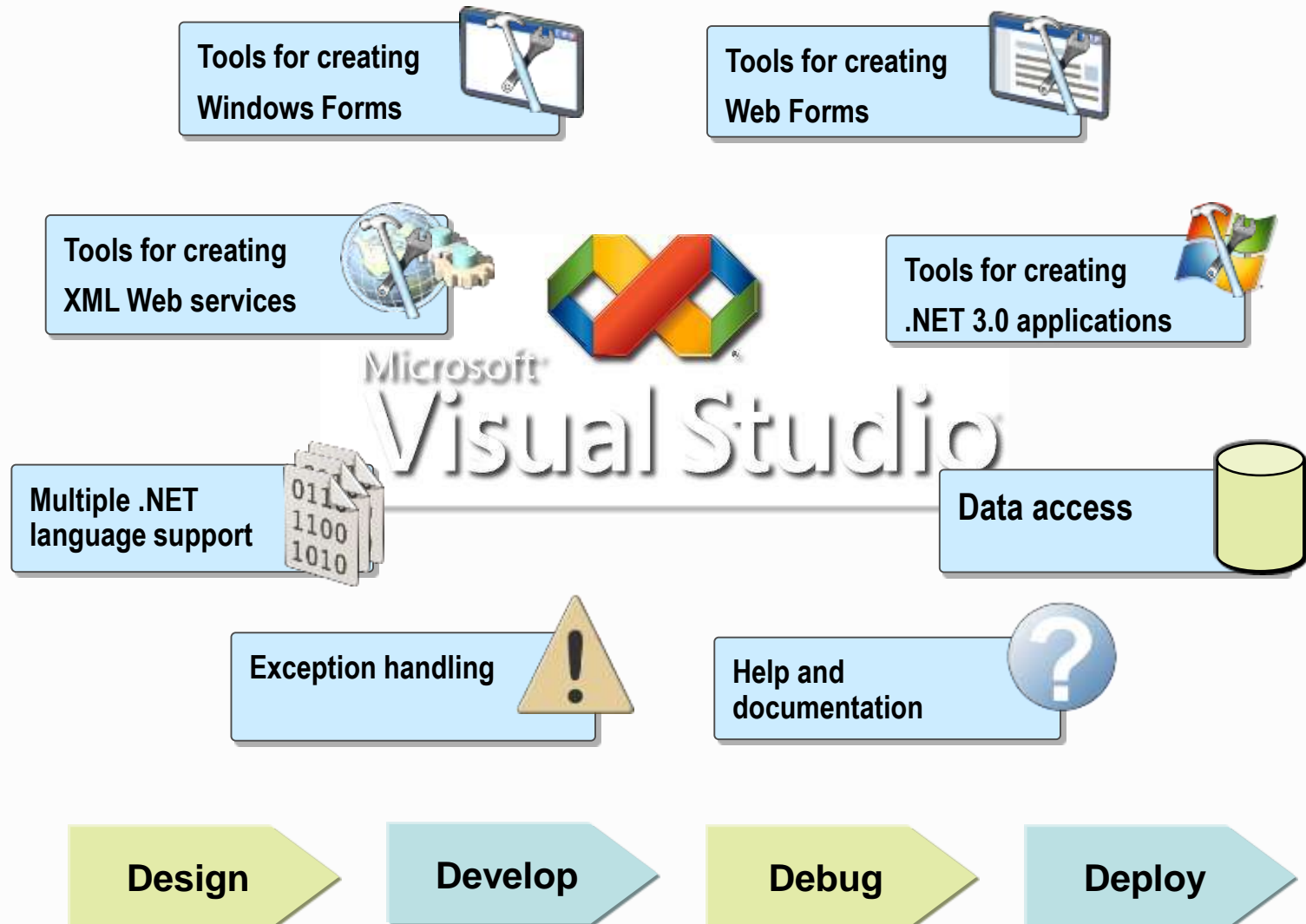
**Windows CardSpace**

# Visual Studio.NET

---

- The Design goals are:
  - ◆ Maximize Developer Productivity
  - ◆ Simplify Server based Development
  - ◆ Deliver Powerful Design Tools
- RAD (Rapid Application Development Tool) for the next generation internet
  - ◆ Forms Designer
    - For creating ASP.NET web form pages
  - ◆ Server Explorer
    - For accessing web services
- Enhanced RAD support for creating your own components.

# Visual Studio .NET as IDE





# Quick Recap ...

---

- .NET is Microsoft's new Initiative to treat software as service.
- .NET framework provides cross language Interoperability using CTS.
- CLR is an execution engine of .NET Platform.
- CLR executes managed code which is IL (intermediate Language code).
- Assembly is a logical unit of deployment.