INTRODUCTION TO .NET FRAMEWORK AND ASP.NET

BEFORE .NET FRAMEWORK

- Windows OS applications developers use of the COM programming model.
- COM (Component Object Model).

–Allow to build libraries of code that could be shared across diverse programming languages.

–Plagued by complicated

–Possible only on the Windows operating system

.NET PLATFORM

- .NET Framework

Is a software platform for building systems on the Windows family of operating systems, as well as on numerous non-Microsoft operating systems such as Mac OS X and various Unix/Linux distributions

SOME KEY BENEFITS OF THE .NET PLATFORM

- Interoperability with existing code:

  –Existing COM software can commingle with newer .NET software, and vice versa.

- Support for numerous programming languages:

  –C#, Visual Basic, F#, and so on.

- A common runtime engine shared by all .NET-aware languages:

  –One aspect of this engine is a well-defined set of types that each .NET-aware language understands(Int32 , int64 etc).

SOME KEY BENEFITS OF THE .NET PLATFORM

- Language integration:

  – .NET supports cross-language inheritance, cross-language exception handling, and cross-language debugging of code.

  –For example, you can define a base class in C# and extend this type in Visual Basic.

- A comprehensive base class library:

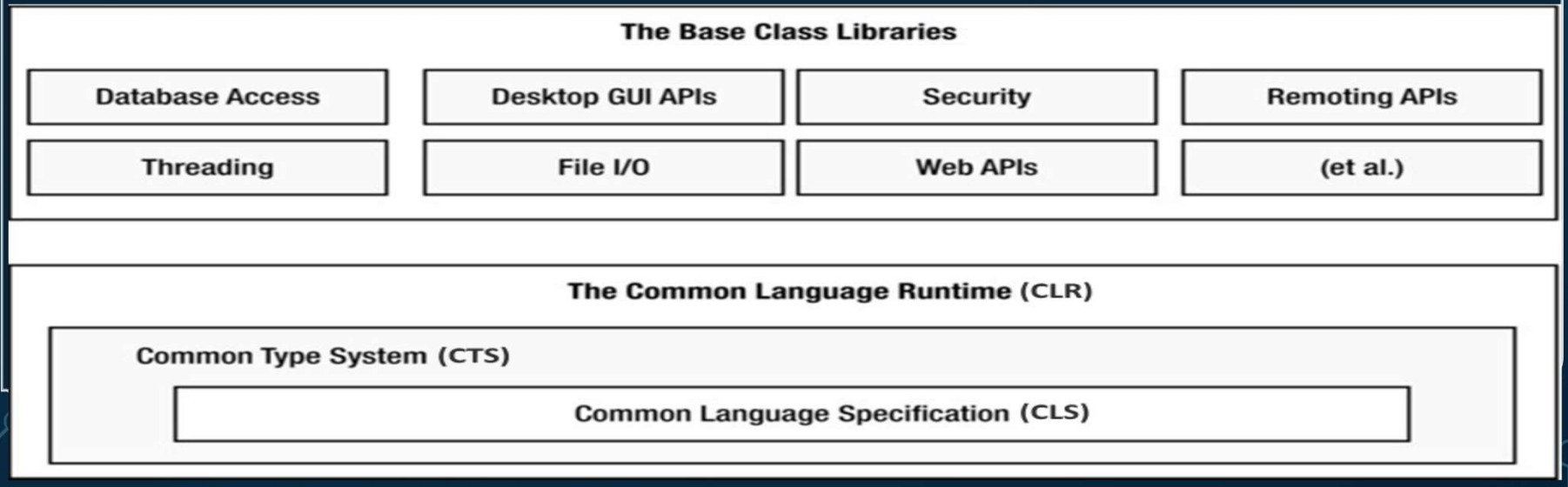
  –This library provides thousands of predefined types that allow you to build code libraries, simple terminal applications, graphical desktop application, and enterprise-level web sites.

SOME KEY BENEFITS OF THE .NET PLATFORM

- A simplified deployment model:
  - –Unlike COM, the .NET platform allows multiple versions of the same *.dllto exist in harmony on a single machine.

BUILDING BLOCKS OF THE .NET PLATFORM (THE CLR, CTS, AND CLS)

- .NET can be understood as a runtime environment and a comprehensive base class library



COMMON LANGUAGE RUNTIME (CLR)

- The primary role of the CLR is to locate, load, and manage .NET objects
- Takes care of a number of low-level details such as :
  - –memory management
  - –application hosting
  - –coordinating threads
  - –performing basic security checks

COMMON TYPE SYSTEM (CTS)

- The CTS specification fully describes all possible data types and all programming constructs supported by the runtime.

- Specifies how these entities can interact with each other, and details how they are represented in the .NET metadata format.

COMMON LANGUAGE SPECIFICATION (CLS)

- .NET-aware language might not support every feature defined by the CTS.

- CLS, is a related specification that defines a subset of common types and programming constructs that all .NET programming languages can agree on.

THE ROLE OF THE BASE CLASS LIBRARIES

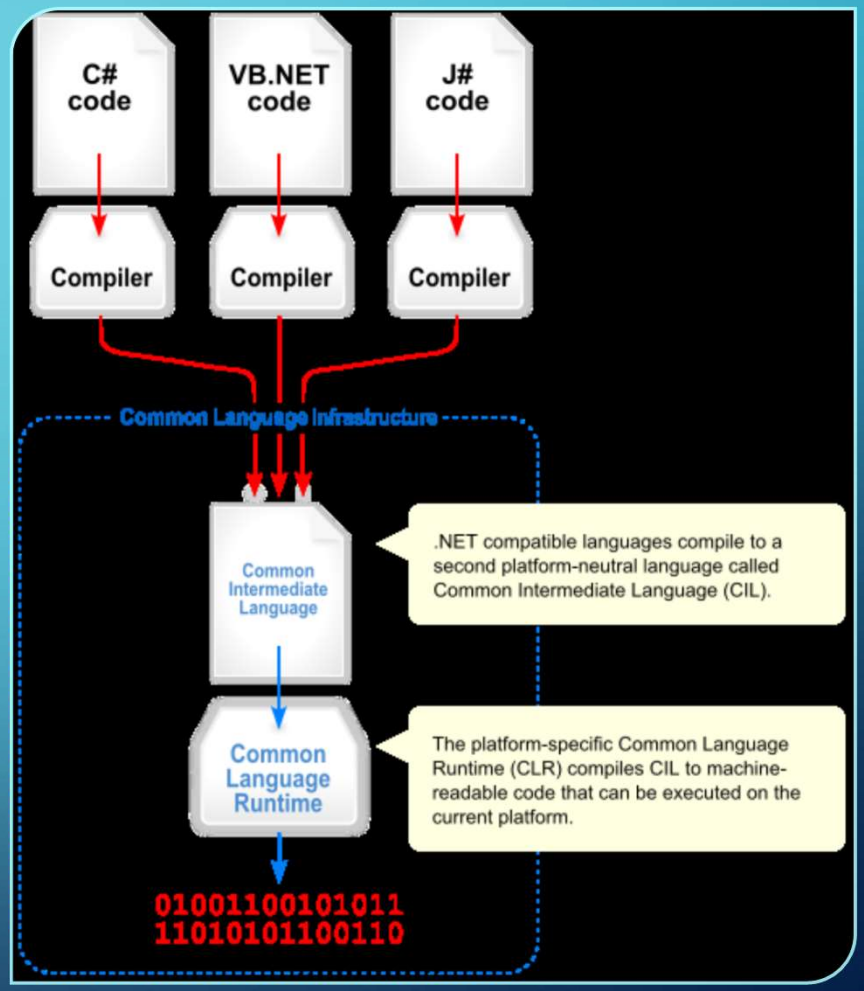
- **encapsulate various primitives such as:**
  - –Threads
  - –file input/output (I/O)
  - –graphical rendering systems
  - –interaction with various external hardware devices
- **provide types to interact with:**
  - –XML documents
  - –the directory and file system on a given computer
  - –communicate with a relational databases (via ADO.NET), and so on.

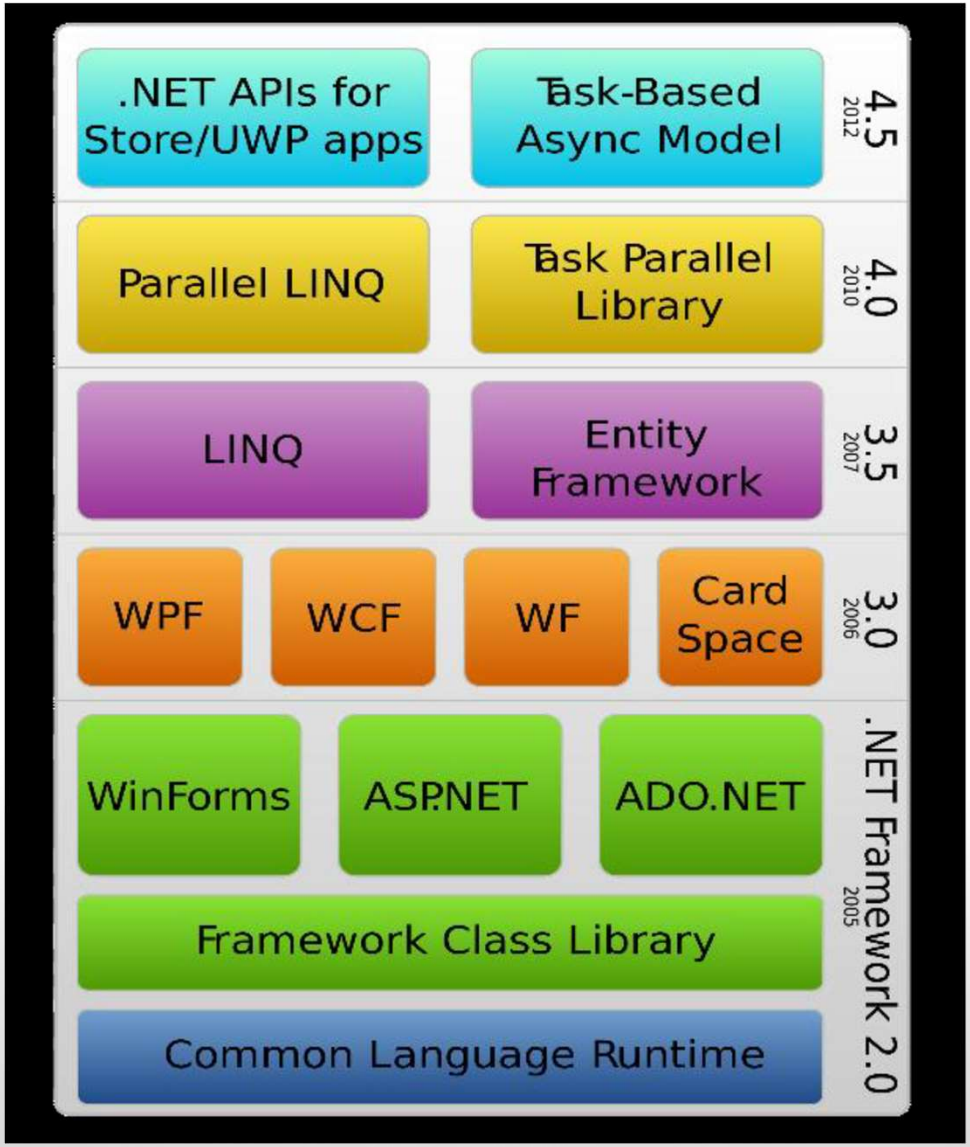
THE ROLE OF THE BASE CLASS LIBRARIES

- provides support for a number of services required by most real-world applications.
  - –ASP.NET to build web sites
  - –WCF to build distributed systems
  - –WPF to build desktop GUI applications

AN OVERVIEW OF .NET ASSEMBLIES

- Regardless of which .NET language you choose to program with, understand that despite that .NET binaries take the same file extension as unmanaged Windows binaries (*.dllor *.exe)



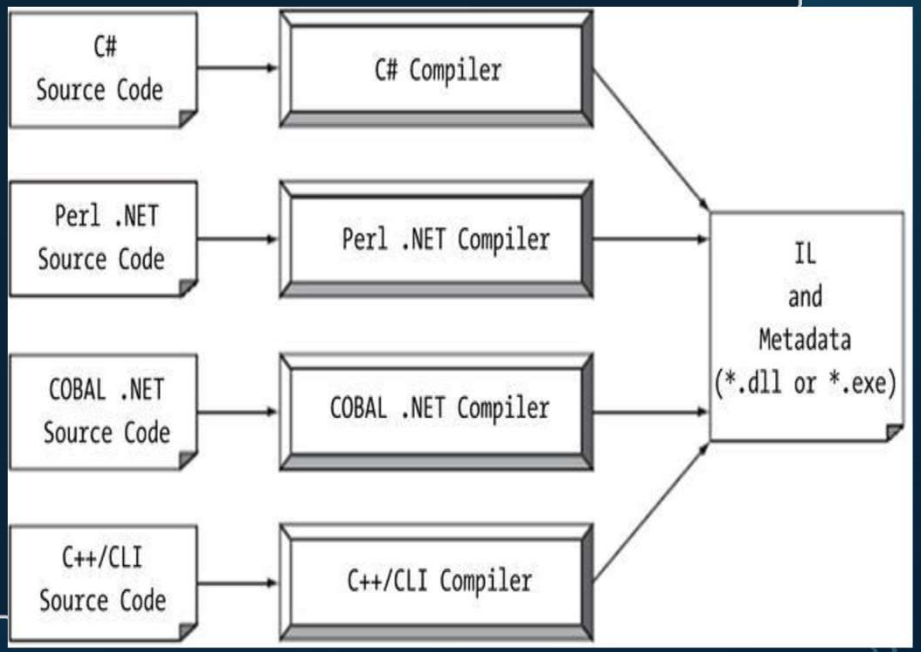


.NET 1.0 (2002)
.NET 1.1 (2002)
.NET 2.0 (2005)
.NET 3.0 (2006)
.NET 3.5 (2007)
.NET 4.0 (2010)
.NET 4.5 (2012)
.NET 4.6 (2015)

AN OVERVIEW OF .NET ASSEMBLIES

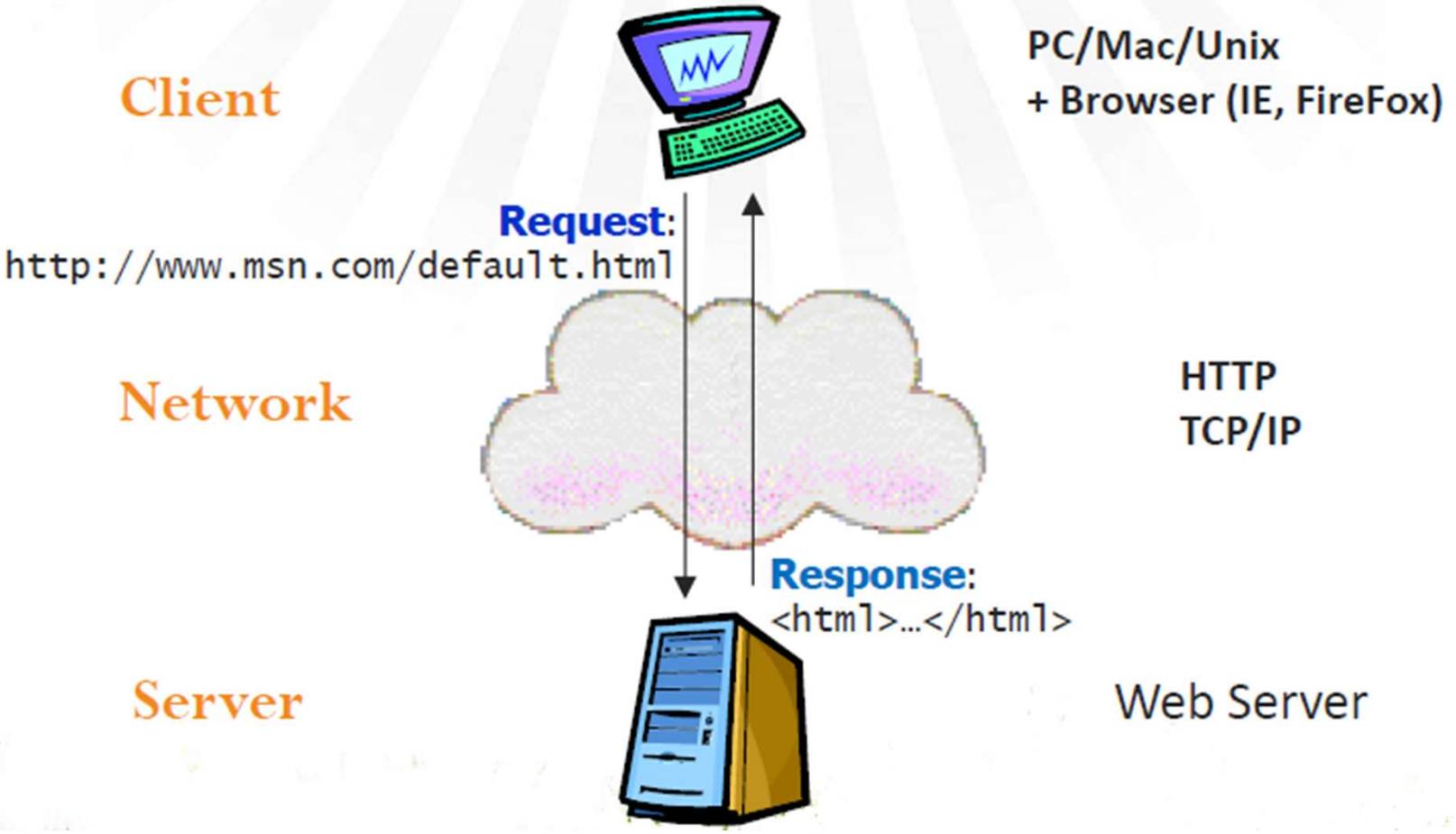
- .NET binaries do not contain platform-specific instructions but rather platform-agnostic Intermediate Language (IL) and type metadata

- IL is also known as Microsoft Intermediate Language (MSIL) or alternatively as the Common Intermediate Language (CIL)



ASP.NET TECHNOLOGY

INTERNET TECHNOLOGIES
WWW ARCHITECTURE



WEB TECHNOLOGIES

- HTTP / HTTPS(URL, GET/POST)
- **Client-side:**

–**HTML / XHTML (Extensible HyperText Markup Language)**

–**JavaScript / VBScript (client-side scripting)**

–**Applets / ActiveX controls**

- **Server-side:**

–**PHP**

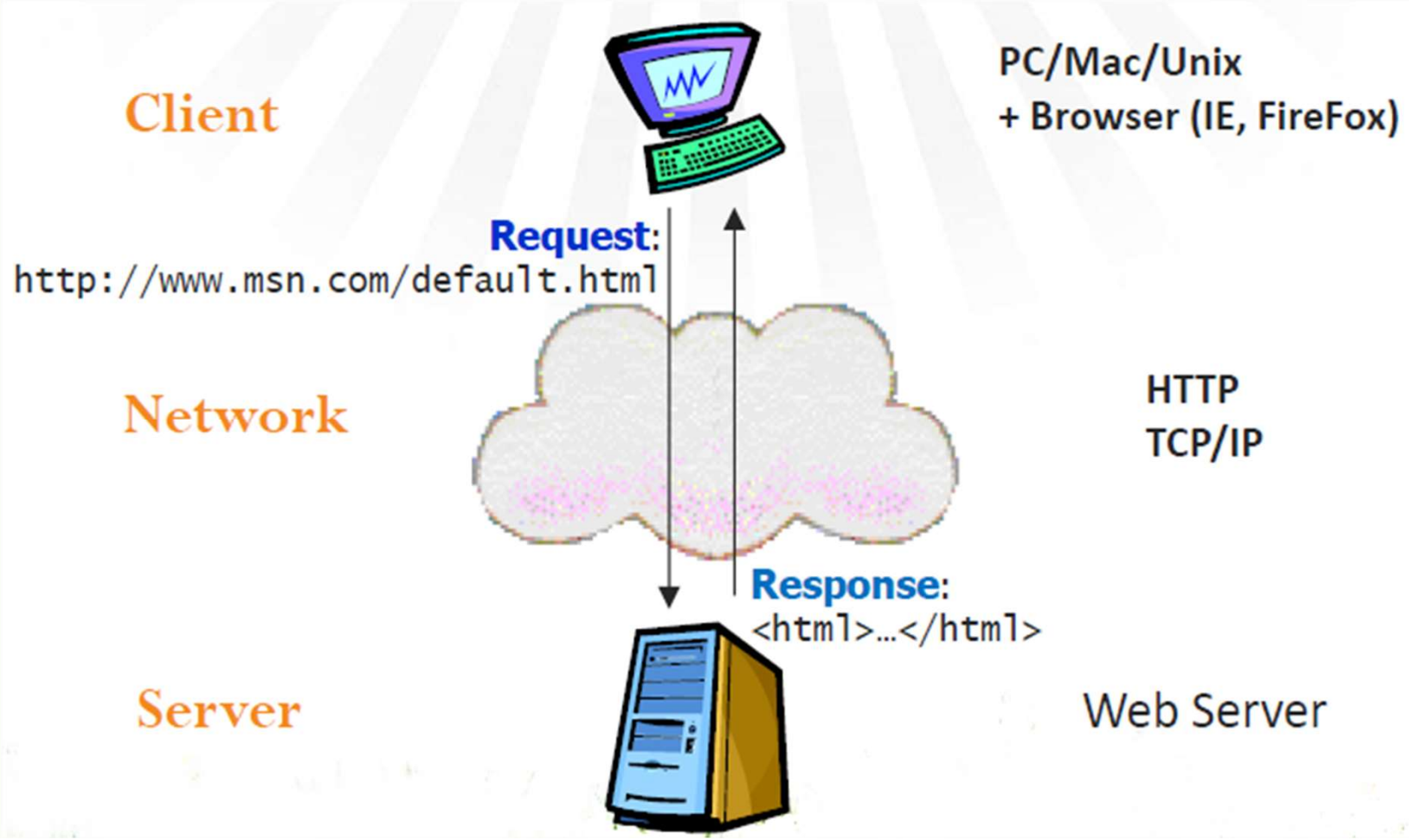
–**Python**

–**JSP(Java Server Pages)**

–**ASP (Active Server Pages)**

–**ASP.NET (next generation of ASP)**

ASP ARCHITECTURE

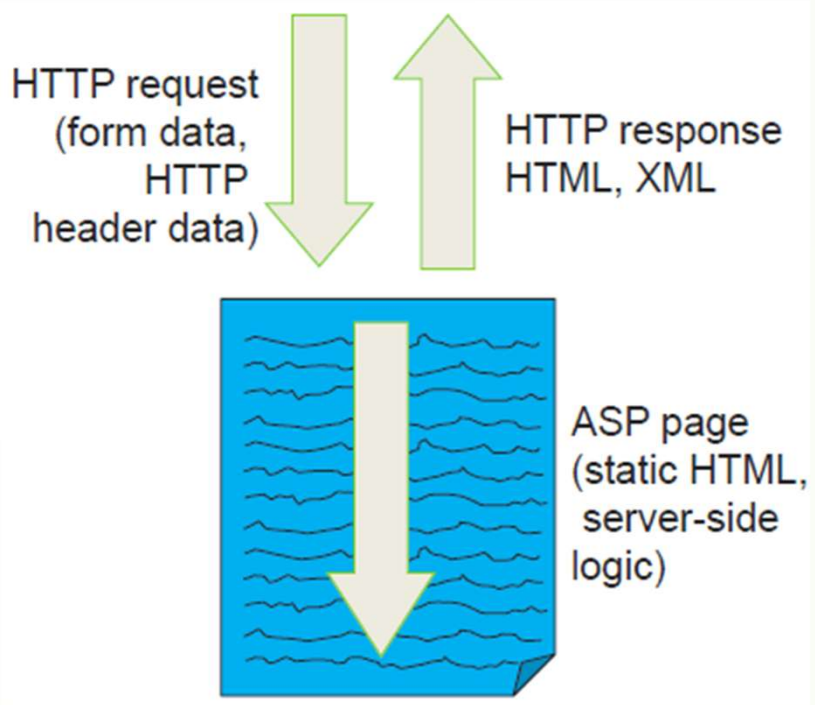


SERVER-SIDE CODE

What is server-side code?

–Software that runs on the server, not the client
–Receives input from
- **URL parameters**
- **HTML form data**

–Can Access server-side databases, e-mail servers, files, mainframes, etc.
–Dynamically builds a custom HTML response for a client



ASP.NET OVERVIEW AND FEATURES

- ASP.NET provides services to allow the creation, deployment, and execution of Web Applications and Web Services
- Web Applications are built using Web Forms
- Web Forms are designed to make building web-based applications as easy as building Visual Basic applications
- Built on .NET Framework: any .NET programming language can be used (C#, Visual Basic)
- Complete object model
- Separation of code and UI
- Maintains page state
- Session management
- Caching, Debugging, Extensibility