

*Features, Strengths and Weaknesses  
of Microsoft .NET*

by

*Prabhu Sunderaraman*

*prabhus@durasoftindia.com*

**DuraSoft**


*http://www.durasoftindia.com*

at

*Satyam*

*April 03 , 2003*

---

- Full permission to copy and distribute this material is granted to any one, provided the original contents of this document is distributed in full and the source and ownership of the document is acknowledged.
  - This document was produced at and belongs to DuraSoft (<http://www.durasoftindia.com>)
  - Pages with a  have detailed examples
-

**Abstract:**

Microsoft's .NET technology is gaining a lot of momentum. It has brought the next wave of excitement in the software development community and is challenging the state of the current practices. A number of technologies that have been developed in the recent past have made significantly contribution in the .NET technology. What are the main features and goals of the .NET environment? The Common Type System, the managed code and the Common Language Runtime makes the programming language almost a non-issue in .NET. It truly provides great language interoperability. The common class library/framework makes developing graphical and multithreaded applications a breeze in both VB.NET and C# and any of the other supported languages for that matter. What about the distributed object computing? XML technology serves as the marshaling wire protocol in .NET, making web services truly platform independent. In this presentation, we will look at the goals and features of .NET, we will discuss some of the strengths of this framework and take a look at some of the weaknesses as well. Code examples will be presented to illustrate the concepts.

**Targeted Audience**

Software developers, programmers and managers who are involved in day to day technical activities of software development will benefit from this presentation. This session is mainly intended for programmers or people with fairly good programming experience.

---

## Overview of the Presentation

- What is .NET?
  - .NET Architecture
  - CLR
  - Languages of .NET
  - ASP.NET
  - WinForms
  - Web Services
  - Strengths and Weakness
  - Conclusion
-

## Technologies of the Past

- Software Systems
  - Client-Server Computing
  - Object-Oriented Programming
  - Distributed Object Computing
  - Web-based technology
-

## Distributed Object Computing

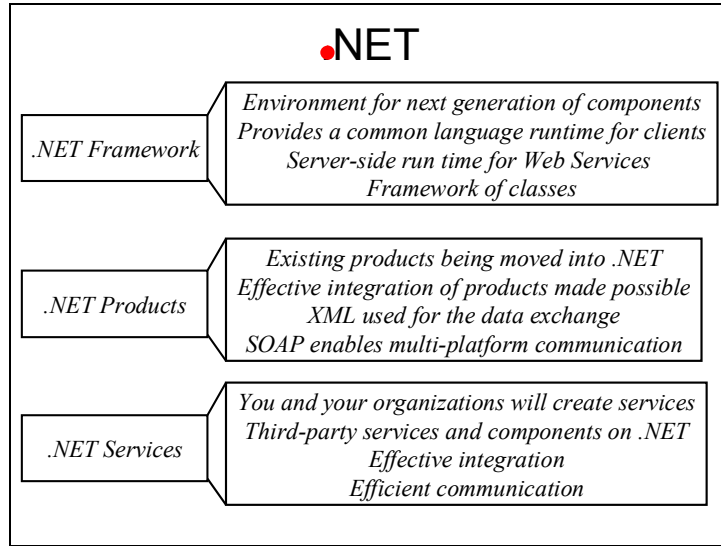
- Component based software development
  - A Server or Component provides some service
    - Objects define behavior on services
  - Clients want to access these services
  - CORBA, DCOM technologies allow these
  - Power and limitations?
-

## Web based technology

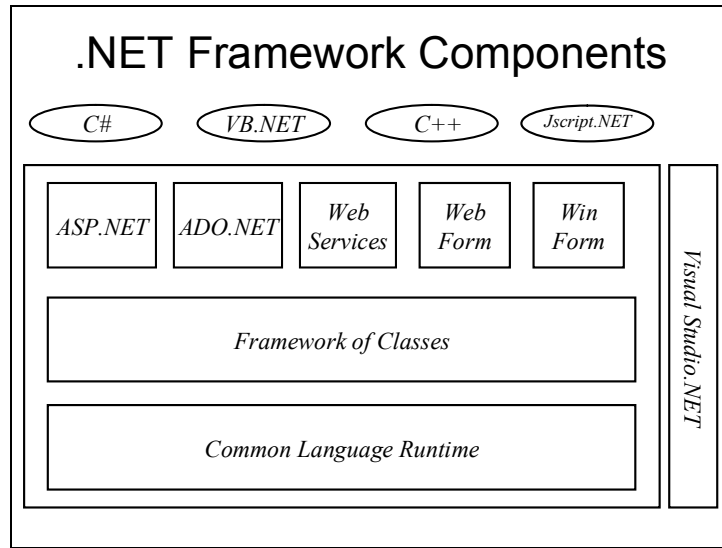
- The world full of clients
  - Any platform, any where
  - Browser based
  - Power and limitations?
  - How much can you do with a browser front end?
-

## What is needed?

- How about a framework where
    - a software component is available on any server
    - any client can access it irrespective of platform
    - the client may be just about any thing
      - thin, fat
      - any language of your choice
  - The Software simply becomes a service
-



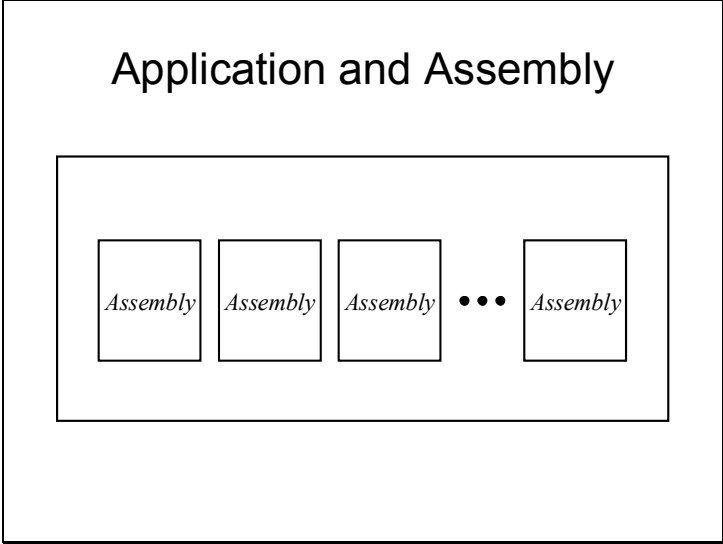
---



---

## Overview of the Presentation

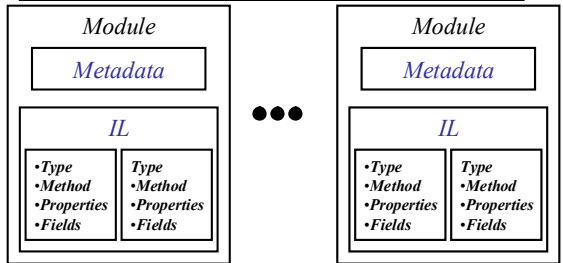
- What is .NET?
  - .NET Architecture
  - CLR
  - Languages of .NET
  - ASP.NET
  - WinForms
  - Web Services
  - Strengths and Weakness
  - Conclusion
-



---

# Assemblies

<i>Is a collection of files</i>		<i>Manifest</i>
•dlls	•bitmaps	•Identity of assembly
•binary data	•resource files	•Security declarations
•data files	•any thing else	•Dependencies
		•Exposed types/resources

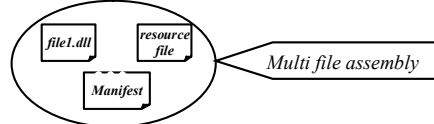
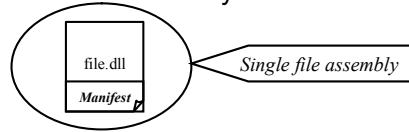



---

## Manifest

- Describes details about the assembly

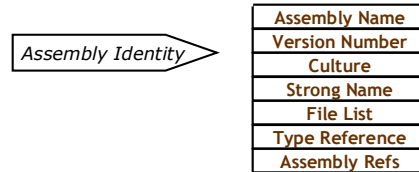
- version
- security
- scope
- resources
- classes
- types
- dependencies



- Stored either
  - as a Portable Executable (PE) file along with MISL
- or as stand alone PE file with only the manifest info

## Manifest Info

- Provides enumeration of all files in assembly
- This makes the assembly self-describing
- Maps references to types, resources & files with their declarations/implementations
- Provides details of assemblies that containing assembly depends on



- Additional information may be provided in manifest  
– company, version, copyright, trademark, etc.



## Assembly Types

- Private Assemblies
  - the default
  - only intended for use by one application
  - if several applications use, each will have a copy
    - (not the intent)
- Shared Assemblies
  - this is intended for use by multiple applications
  - name must be unique so it does not collide
  - place in the global assembly cache



---

## Configuration Files

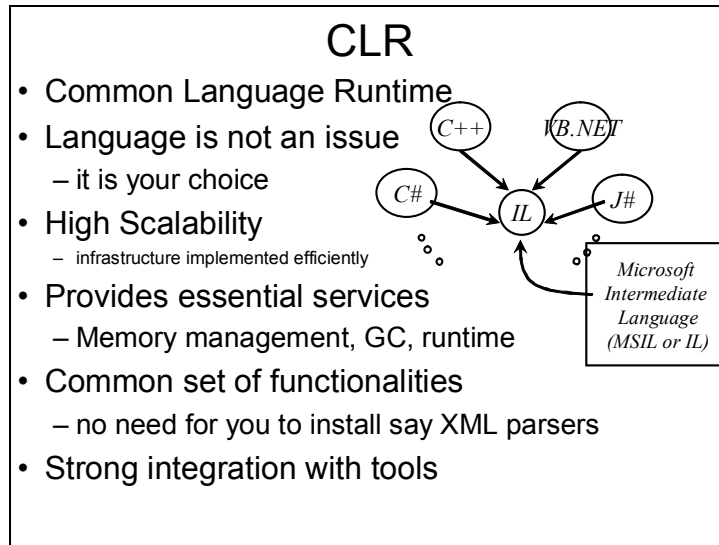
- Configuration files are used to specify various information for run time to use
  - API's can be used to read (not alter) these files
  - This makes it possible to make variations without having to change and recompile code
  - You may use **.NET Framework Configuration Tool** (Mscorcfg.msc)
-

## Quiz Time

- What is a common limitation of COM & CORBA? firewall
  - What technologies are part of .NET framework?  
ASP.NET, ADO.NET, WinForms, WebForms, Web Services
  - What is a unit of deployment in .NET? Assembly
  - What are the two types of assemblies in .NET? private, shared
  - Contents, dependencies of assembly are described in? Manifest
  - Identity of an assembly is defined by? Name, version, culture, SN
  - Process of locating an assembly is called? Probing
  - CLR looks for private assembly in Application Directory ?
  - To be placed in GAC, an assembly must have? Strong Name
  - CLR to look for different assembly version, what's modified? Config. File
-

## Overview of the Presentation

- What is .NET?
  - .NET Architecture
  - CLR
  - Languages of .NET
  - ASP.NET
  - WinForms
  - Web Services
  - Strengths and Weakness
  - Conclusion
-

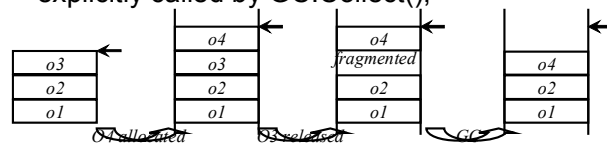


## Features of CLR

- Provides essential services
    - Memory management, GC, runtime
  - Common set of functionalities in place
    - no need for you to install things like XML parsers, compilers
  - Strong integration with tools
  - High Scalability
    - common infrastructure implemented efficiently
-

## Memory Management and GC

- CLR administers area of Heap - Managed Heap
- Takes full control of object space allocation
- As objects created, memory allocated linearly
  - this allows for fast allocation
  - results in fragmentation though
- GC takes care of this
  - invoked automatically when needed
  - explicitly called by `GC.Collect();`



## Managed Vs Unmanaged Code

- Managed Code
    - Code built to depend on CLR is managed code
  - Unmanaged Code
    - This is not executed under the tight supervision of CLR
    - No garbage collection
    - Limited debugging capabilities
    - Useful to call Platform Specific functions
  - Example
    - Interaction with COM components
-

## Unsafe vs. Unmanaged Code

- Unmanaged Code:
  - this is not executed under the tight supervision of CLR
    - no garbage collection
    - limited debugging capabilities
  - Useful to call Platform Specific functions
  
- Unsafe Code:
  - this is managed code!
  - it simply uses some constructs (like pointer usage) that C# does not encourage

---

## Unsafe Code

- CLR manages memory
  - Java does not allow pointer manipulation
  - C# derived from C++, wants to allow it, however with caution
  - Code that manipulates pointers may lead to memory leaks, etc.
  - C# declares the section of code that manipulates pointers as unsafe!
    - you take care of memory management when within this block of code – allows you to use pointers
  - To prevent GC use the fixed keyword to *pin*
-

## Quiz Time

- What does CLR provide?  
Memory MGMT, Garbage Coll., Security
  - What is the source code compiled into in .NET? MSIL
  - Code executed in CLR is called  
Managed Code ?
  - To use pointers in C#, you mark the code as  
unsafe ?
  - Is unsafe code classified as unmanaged code as well?  
No
-

## Overview of the Presentation

- What is .NET?
  - .NET Architecture
  - CLR
  - Languages of .NET
  - ASP.NET
  - WinForms
  - Web Services
  - Strengths and Weakness
  - Conclusion
-

## Common Type System

- Each language has its own set of types
  - Type conversion in inter language communication is a pain
  - Errors usually go unnoticed during compilation
    - Results in runtime errors
  - CTS addresses this
  - One set of types for all .NET languages
  - System.Object is the base type
    - Offers common set of methods for all types
-

## Primitive Types

- Boolean
  - Byte
  - Char - unicode
  - DateTime
  - Decimal – signed 28 significant digits
  - Double – 64 bit double precision
  - GUID
  - Int16, Int32, Int64
  - Sbyte – 8 bit signed integer
  - Single – 4 byte single precision
  - TimeSpan
-

## Value Type vs. Reference Type

- Value type instances created on stack
  - Reference types instances created on heap
  - These are very similar to the concept in Java
  - Evaluation of Equals
    - on value types compares value
    - on reference types compares identity
  - Reference types are initialized to null
  - *Unboxed* value types initialized to a default value
-

## Custom Types

- User defined types may be created
  - may be of value types
  - may also be of reference type
- Available to all .NET languages!

---

## Value type: Boxing and Unboxing

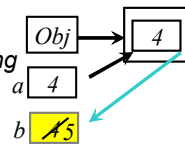
- Boxing is the conversion or copying of a value type to a reference type
  - The value type is wrapped into a reference instance
- Unboxing is the conversion or copying of a boxed reference to a value type

```
Int32 a = 4;
```

```
Object obj = a; // boxing
```

```
Int32 b = (Int32)obj; // unboxing
```

```
b = 5;
```





### C++, Java, VB and C#...

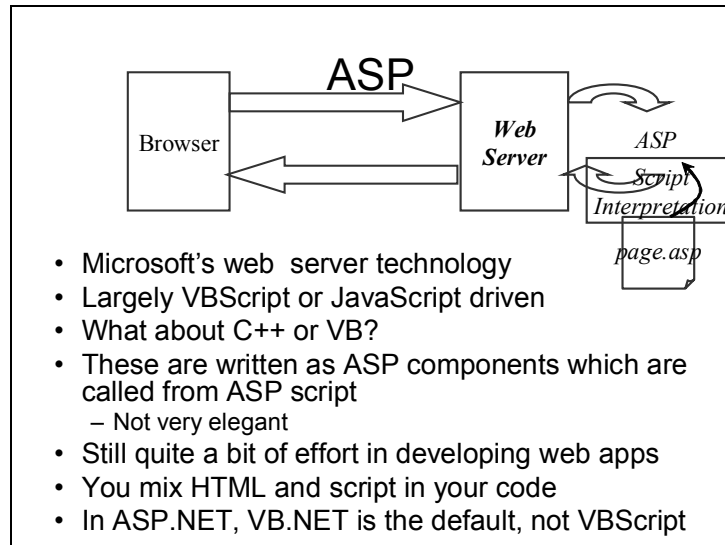
- C# made to be simpler than C++
    - eliminates a number of troublesome constructs
    - you still have the power to use C++ with in C#!
      - mark the code as unsafe
  
  - VB.NET has evolved to features offered by C#
  
  - Java and C# have similarities
    - reflection, no global functions
    - However, C# carries features of C++ that Java does not
      - operator overloading, enumeration
-

## Quiz Time

- Types used by .NET languages are from? CTS
  - What types of objects are created on stack? value type
  - What keyword is used to make it a value type? struct
  - What keyword is used to make it reference type? class
  - Conversion of value type to reference type is called boxing ?
  - What executes the MSIL code in .NET CLR ?
  - What are the int types in CTS Int16, Int32, Int64 ?
  - What languages does C# resemble Java, C++ ?
  - Which language has gone through major revision? VB.NET
  - What features does C# have that Java does not?  
Operator overloading, enumeration
-

## Overview of the Presentation

- What is .NET?
  - .NET Architecture
  - CLR
  - Languages of .NET
  - ASP.NET
  - WinForms
  - Web Services
  - Strengths and Weakness
  - Conclusion
-



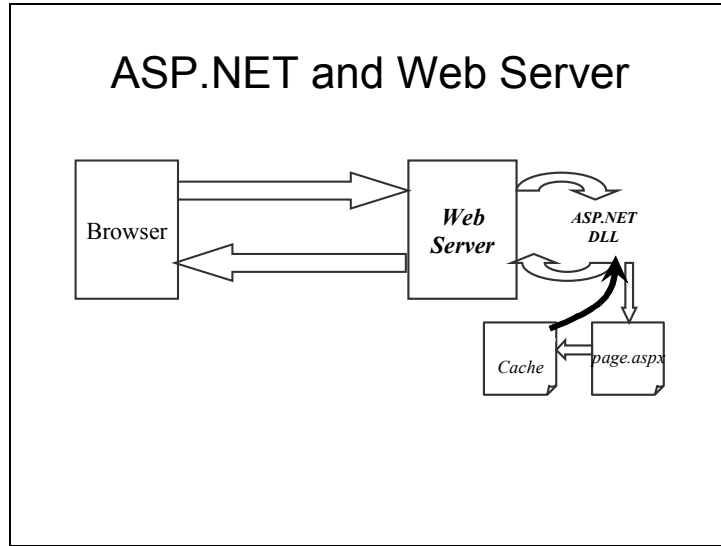
## Writing an ASP Page



---

## What does ASP.NET solve?

- Less code
    - You rely on controls that manage session state with almost no code to be written
    - You get better control towards code reuse
    - Not interpretive – code compiled
      - .NET class created on first visit to a page
      - Similar to how JSP works!
      - Any changes are automatically recompiled
      - Cached class gives better performance
    - Much better control over components development/reuse
      - Easy to upgrade and redeploy
      - Strong typed languages make it easy to use components
    - Good separation of HTML and Script code
    - Better control on client variations
      - Server side components sense type of browser and can tailor response
  - Strong XML support
    - Probably most driving feature SOAP leading to web services
-



---

## Code-Behind Page

- Place code in a separate file, called a **code-behind** class
  - Provides cleaner separation between display code and application code
  - Gives exceptional reuse and extensibility
-

## Server Controls

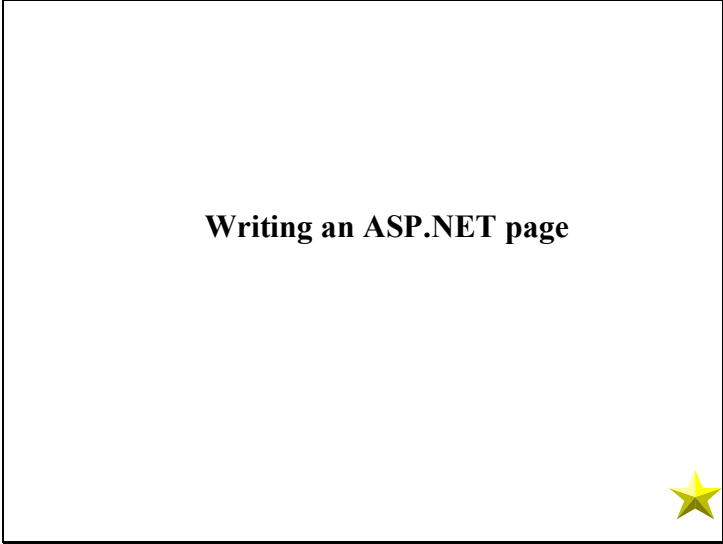
- Components that run on server
- They appear like HTML tags
  - But much shorter and replace several lines of HTML
- They expand to HTML code
- A TextBox control looks like this

```
<asp:TextBox id="TextBox1" runat="server" > </asp:TextBox>
```

- A Button control looks like this

```
<asp:Button id="Button1" Text = "Submit" runat="server" >  
</asp:Button>
```

---



---

## Overview of the Presentation

- What is .NET?
  - .NET Architecture
  - CLR
  - Languages of .NET
  - ASP.NET
  - WinForms
  - Web Services
  - Strengths and Weakness
  - Conclusion
-

## What does WinForms provide

- Windows Forms or WinForms are classes that provide User Interfaces
  - These bring rich VB UI to all .NET languages
  - Set of class library to develop efficient UI on .NET
  - Provides visual inheritance
    - If you inherit from a class, you inherit its UI as well
    - You can say inherit from a dialog and simply add a button
-

## Form Class

- Representation of any window in the application
- You can create an object of this, set it to be modal, create as a dialog, create and attach

MDI windows

```
public class Form : ContainerControl {
    public static Form ActiveForm {get; }
    public IButtonControl AcceptButton {get; set; }
    ...
    public Cursor Cursor {get; set; }
    public bool MaximizeBox {get; set; }
    ...
    public void Activate();
    public void Close();
    public void Refresh();
    ...
}
```

---

## Writing a WinForm Application

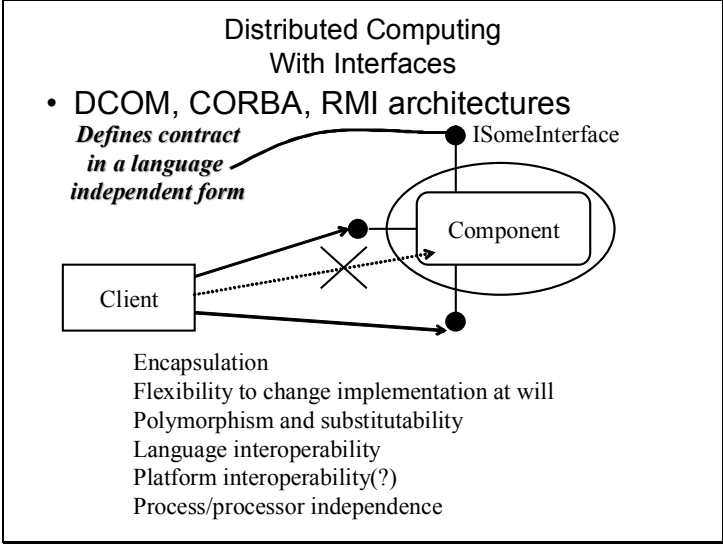
- Using Visual Studio .NET to write a windows application
- The main class
- Modifying the main class
- EXAMPLE



---

## Overview of the Presentation

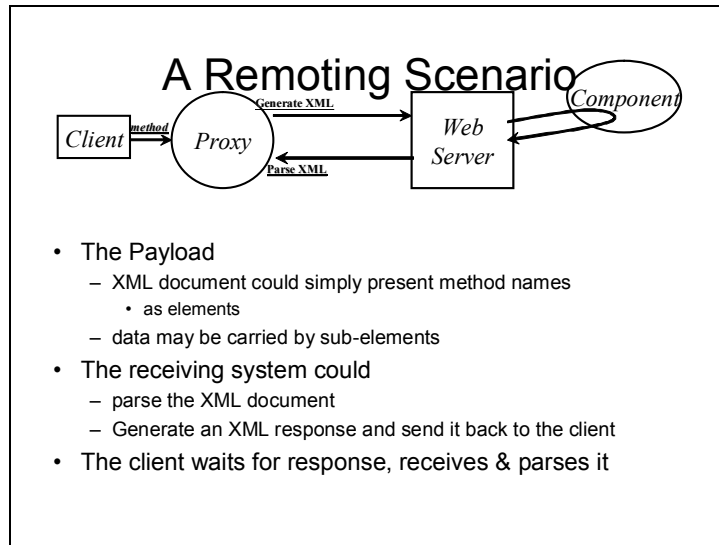
- What is .NET?
  - .NET Architecture
  - CLR
  - Languages of .NET
  - ASP.NET
  - WinForms
  - Web Services
  - Strengths and Weakness
  - Conclusion
-




---

## Distributed Computing & XML

- Problems with DCOM & CORBA
    - Interfaces based clients bind to the interface IDL
    - DCOM works only among select platforms
      - Predominantly on Windows NT, 2000, etc.
    - Client side ORB is required in case of CORBA
  
    - **These technologies are for intranet and not internet**
      - firewall will block the requests from a truly remote client!
  
  - XML
    - provides cross platform transfer mechanism
    - has multi-language / multi-vendor support
    - So why not use XML to transport method invocation rather than simply data?
-

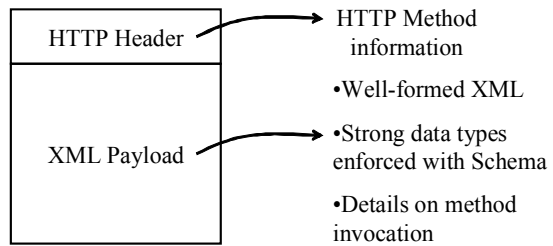


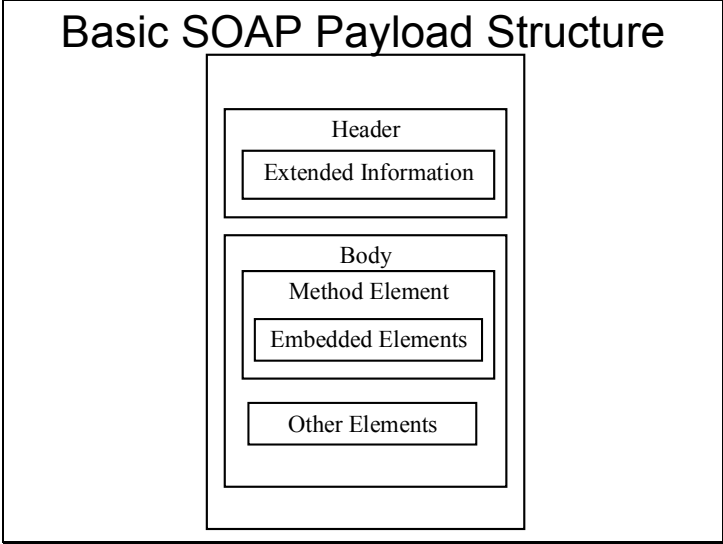
## SOAP

- A well-formed XML may be used as a wire protocol
  - XML documents can be easily transported to remote systems
    - Most convenient to use HTTP (walks through firewall)
    - Other mechanisms like SMTP
  - Mechanism is addressed by SOAP
    - Simple Object Access Protocol (SOAP)
-

## SOAP's approach

- To use HTTP (or other mechanisms) for transport
- Use XML for data encoding/information exchange
- Keep it simple and extensible

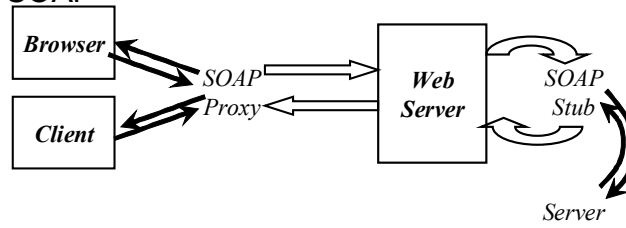




---

## SOAP Applied

- Web Services is simply an application of SOAP



---

## Web Service Infrastructure

- **Web Service Wire Format**
    - Protocol to allow open communication between various entities independent of platform
    - SOAP
  - **Web Service Description**
    - Allows for finding the details of web services, its methods, arguments, types, etc
    - Web Service Description Language (WSDL)
  - **Web Service Discovery**
    - Allows for a web service to make its presence known and for a client to find it
    - SOAP Discovery (DISCO)
-

## Writing a Web Service

- Write a web service source with .asmx

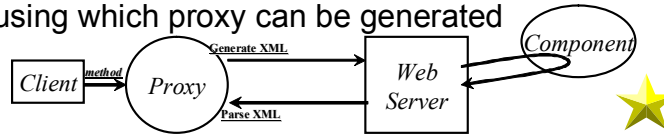
```
<%@ WebService Language="C#"
Class= "TemperatureProvider" %>
using System;
using System.Web.Services;
[WebService(Namespace="http://www.durasoftindia.com")]
public class TemperatureProvider
{
    [WebMethod]
    public double temperature(string city)
    {
        // Ideally we would fetch this information
        // from our resources.
        return new Random().NextDouble() * 100;
    }
}
```



---

### What about traditional clients?

- For non-web based client, you can create proxy
- The web client proxy receives a request, generates XML request & sends it using SOAP
- Receives response, parses it and hands it over to client!
- You can generate a web service proxy using the WSDL compiler.
- You provide a reference to service description using which proxy can be generated



## Quiz Time

- Process of packaging req. in distributed computing? marshaling
  - What is used for marshaling/wire protocol in web services? XML
  - What is SOAP? Simple Object Access Protocol
  - What is used to describe and expose a web service? WSDL
  - Attribute used to make a method a web method is WebMethod.
  - Rich client communicates with web service using proxy ?
  - By nature, as far as state goes, a web service is stateless ?
  - For each web method, how many methods are in proxy? Three
-

## Overview of the Presentation

- What is .NET?
  - .NET Architecture
  - CLR
  - Languages of .NET
  - ASP.NET
  - WinForms
  - Web Services
  - Strengths and Weakness
  - Conclusion
-

## Strengths and Weakness

- Strengths
    - Visual Studio enhances productivity
    - Shortest time to market
      - faster to develop, lower cost
    - Fairly stable (especially compared to beta 2!)
    - High interoperability
    - You code in the language you like, no force
  - Weakness
    - from Microsoft ;) (just kidding)
    - Leveraging with existing unmanaged code
    - Until future release of OS, requires special installation of framework
    - Security of Web Services major concern
-

## Overview of the Presentation

- What is .NET?
- .NET Architecture
- CLR
- Languages of .NET
- ASP.NET
- WinForms
- Web Services
- Strengths and Weakness
- Conclusion

---

## Conclusion

- Truly a very exciting technology
  - High productivity
  - Low cost of development
  - Fun to program in - my opinion :)
  
  - It is going to change the way we develop systems – I sincerely hope so
  
  - Now, lets go crack some code!
-

## References

- <http://www.w3.org>
- <http://msdn.microsoft.com>
- <http://www.durasoftindia.com/downloads.htm>

---

Thanks to Satyam  
for the opportunity to speak!



**DuraSoft**

*'helping you develop durable software'*

<http://www.durasoftindia.com/downloads.htm>

---