



# **ERP Analysts (India) Pvt. Ltd.**

**Lead with Vision**

**705, Swapnalok Complex, Secunderabad - 500 003**

# Agenda

- .NET Framework
- Common language runtime
- Building user interfaces
- Data and ADO.NET
- Summary

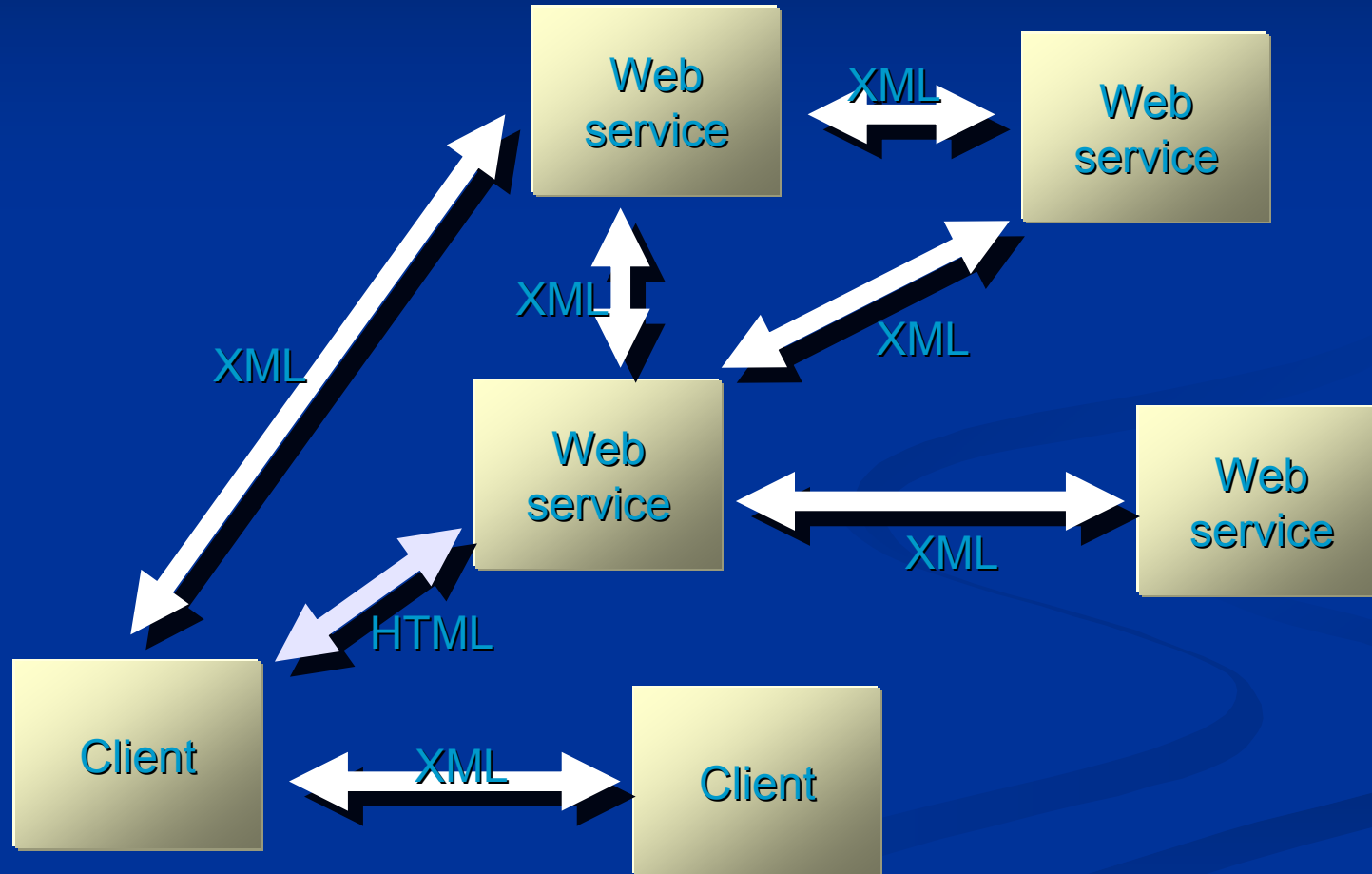
# The .NET Framework

## XML Web Services

- The center of .NET architecture
- Technical definition:
  - “a programmable application component accessible via standard Web protocols”
- Expose functionality from Web sites
  - almost like component programming over the Web

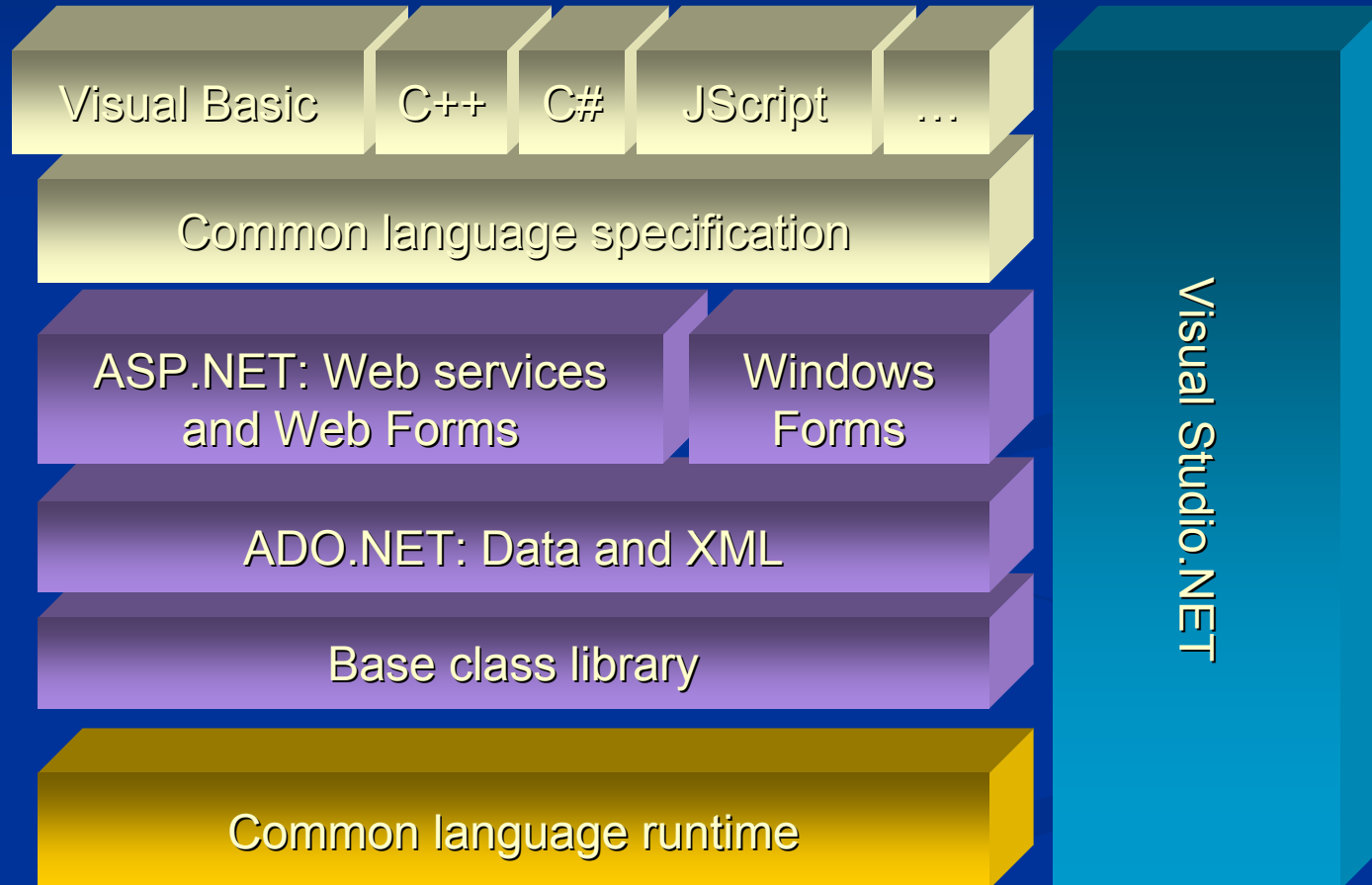
# The .NET Framework

## XML Web Services



# The .NET Framework

## The .NET Framework and Visual Studio .NET



# The .NET Framework

## Common Language Runtime

- simplified development
- XCOPY deployment
- scalability
- rich Web clients and safe Web hosting
- potentially multi-platform
- multiple languages (cross-inheritance)
- increases productivity

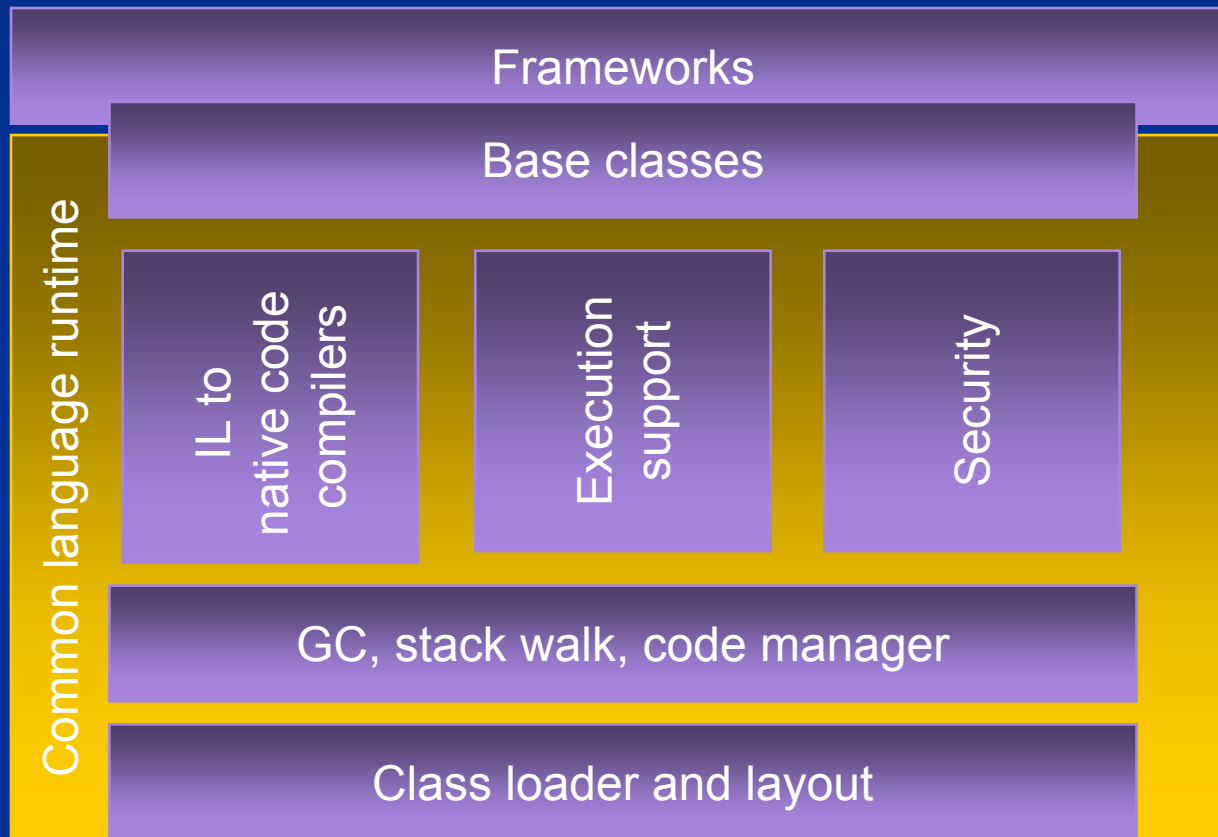
# The .NET Framework

## .NET Framework Services

- ASP.NET
  - logical evolution of ASP (compiled)
- Web Forms
  - manageable code (not “spaghetti” code)
- Windows Forms
  - .NET Framework for building rich clients
- ADO.NET, evolution of ADO
  - new objects (e.g., DataSets)
- XML support throughout

# Common language runtime

# Common Language Runtime Architecture



# Common Language Runtime Goals

- Development
  - standard class framework
  - automatic memory management
  - consistent error handling
  - mixed-language applications
  - multiple platforms
  - safer execution
- Deployment
  - removal on registration dependency
  - safety: fewer versioning problems
  - the end of “DLL Hell”

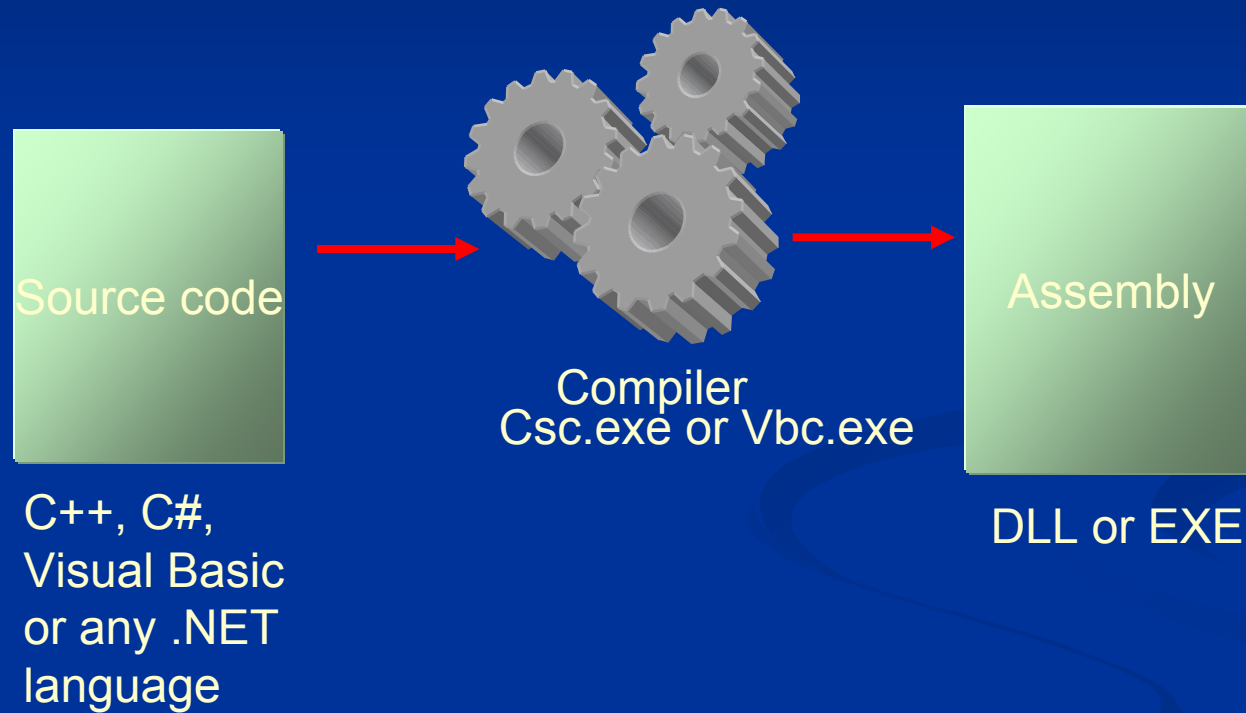
# Common Language Runtime Goals

- Development
  - standard class framework
  - automatic memory management
  - consistent error handling
  - mixed-language applications
  - multiple platforms
  - safer execution
- Deployment
  - removal of registration dependency
  - safety: fewer versioning problems
  - the end of “DLL Hell”

# Common Language Runtime Multiple Language Support

- What about types?
  - Common Type System (CTS)
- Other languages and compilers
  - Common Language Specification (CLS)

# Common Language Runtime Compilation



# Common Language Runtime Assemblies



# Common Language Runtime Metadata

- Type information
  - more complete than IDL / TLB
  - automatically bound into assembly
    - inseparable
    - stored in binary format
  - describes every class type
  - used by Microsoft IntelliSense® in Visual Studio .NET

# Common Language Runtime Metadata in an Assembly

## Type Descriptions

Classes

Base classes

Implemented interfaces

Data members

Methods

## Assembly Manifest

Name

Version

Culture

Other assemblies

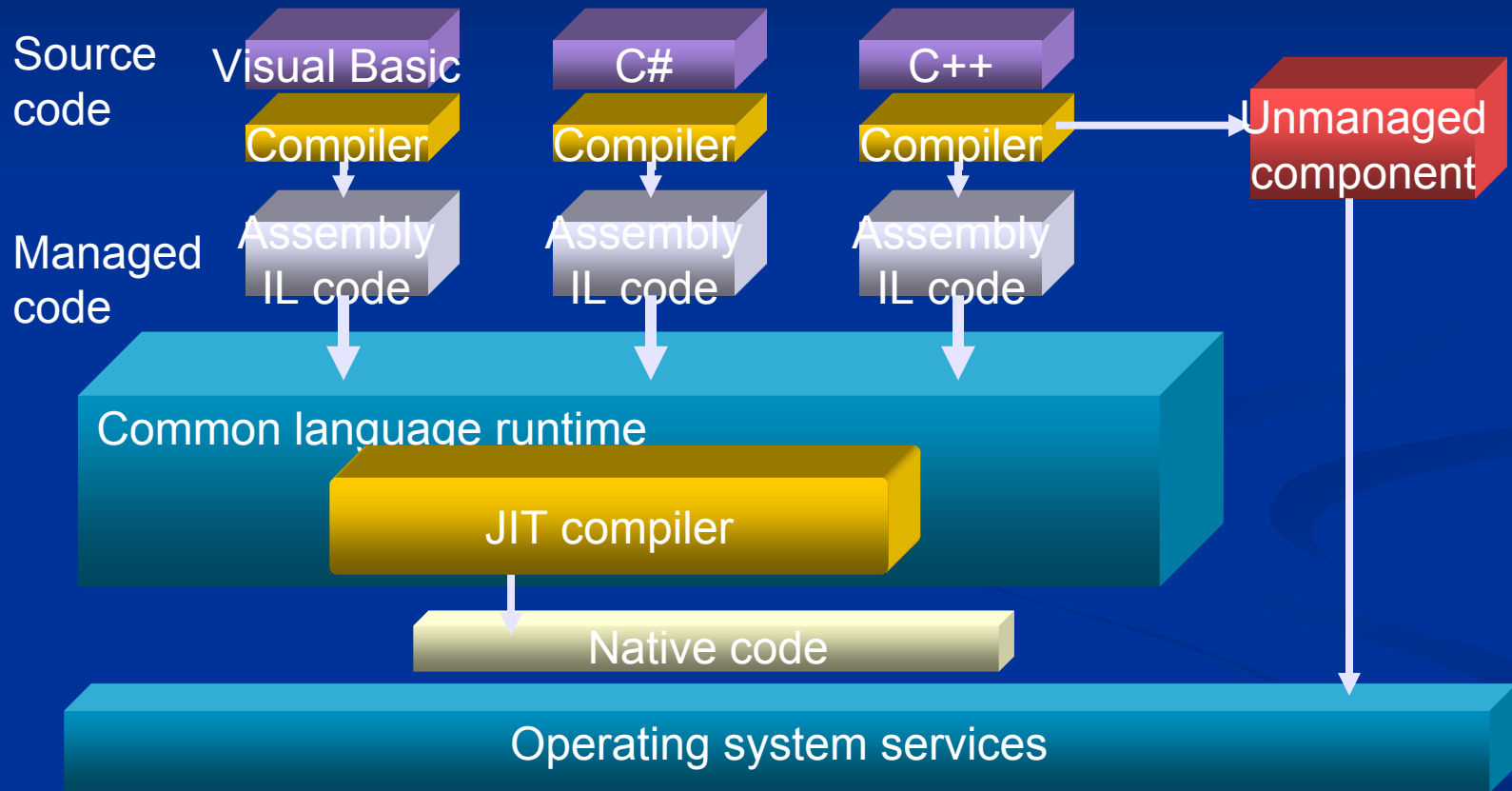
Security permissions

Exported types

# Common Language Runtime Applications

- One or more assemblies
- Assemblies resolution
  - using metadata
    - local (preferred)
    - Global Assembly Cache (GAC)
- Different applications may use different versions of an assembly
  - easier software updates
  - easier software removal

# Common Language Runtime Execution Model



# Web Forms and Web Services

# Building User Interfaces

## Windows Forms

- Framework for building rich clients
  - RAD (rapid application development)
  - rich interfaces
  - easily hooked into Web services
  - rich set of controls
  - data-aware
  - ActiveX® Support
  - licensing
  - accessibility
  - printing support

# Building User Interfaces

## Windows Forms

- Framework for building rich clients
  - RAD (rapid application development)
  - rich interfaces
  - easily hooked into Web services
  - rich set of controls
  - data-aware
  - ActiveX<sup>®</sup> support
  - licensing
  - accessibility
  - printing support

# Building User Interfaces

## ASP.NET

- Logical evolution of ASP
  - control-based, event-driven execution model
  - improved performance
  - supports multiple languages
  - not limited to VARIANT types
  - more productive
  - cleanly encapsulated functionality

# Building User Interfaces

## ASP.NET Web Forms

- Allows clean-cut code (non-spaghetti code)
  - code-behind Web Forms
- Easier for tools to generate
- Code within is compiled then executed
- Improved handling of state information
- Browser-compliant JavaScript generation
- Support for ASP.NET server controls
  - form validation
  - Data-bound grids
  - more

# Building User Interfaces

## Web Services

- Technical definition – “A programmable application component accessible via standard Web protocols”
  - built on SOAP/XML
- Expose functionality from Web sites
  - almost like component programming over the Web
  - functionality exposed using XML/HTML
- Standard Web services include:
  - storage service for .NET
  - calendar
  - MSN Passport

# Building User Interfaces

## Creating a Web Service

### ParcelTracker.asmx

```
<%@ WebService Language="C#" %>
using System;
using System.Web.Services;
using VBParcelTrackerNS;
public class ParcelTrackerWebService
{
    [WebMethod]
    public string GetOrderStatus (int orderNumber)
    {
        //Implementation here!
    }
}
```

# Building User Interfaces

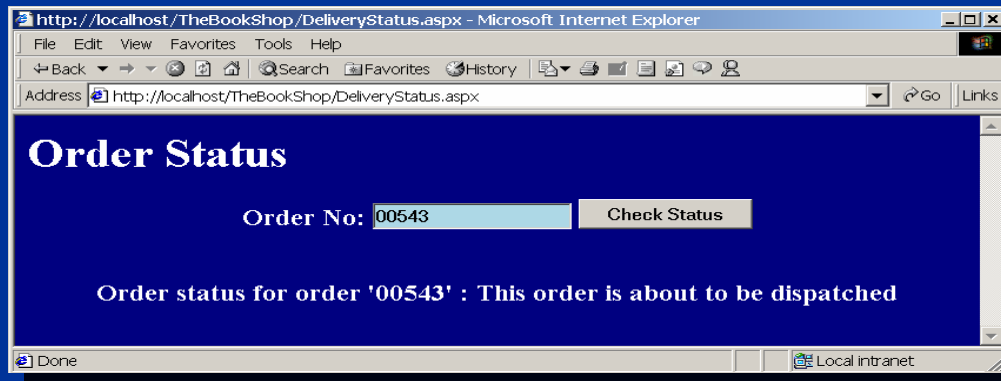
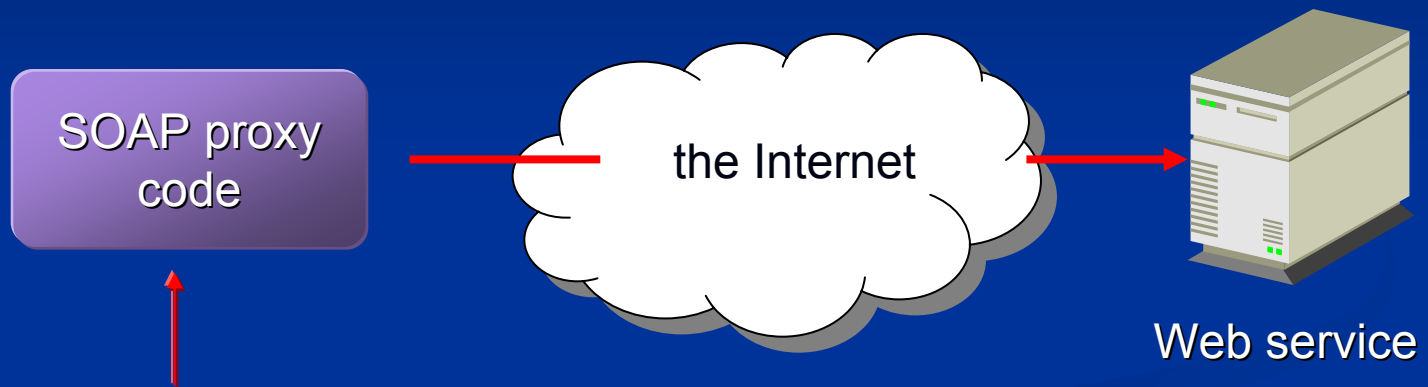
## Creating a Web Service

### ParcelTracker.asmx

```
<%@ WebService Language="C#" %>
using System;
using System.Web.Services;
using VBParcelTrackerNS;
public class ParcelTrackerWebService
{
    [WebMethod]
    public string GetOrderStatus (int orderNumber)
    {
        //Implementation here!
    }
}
```

# Building User Interfaces

## Creating a Web Service Client



# Using ADO.NET for Data Access (XML Output)

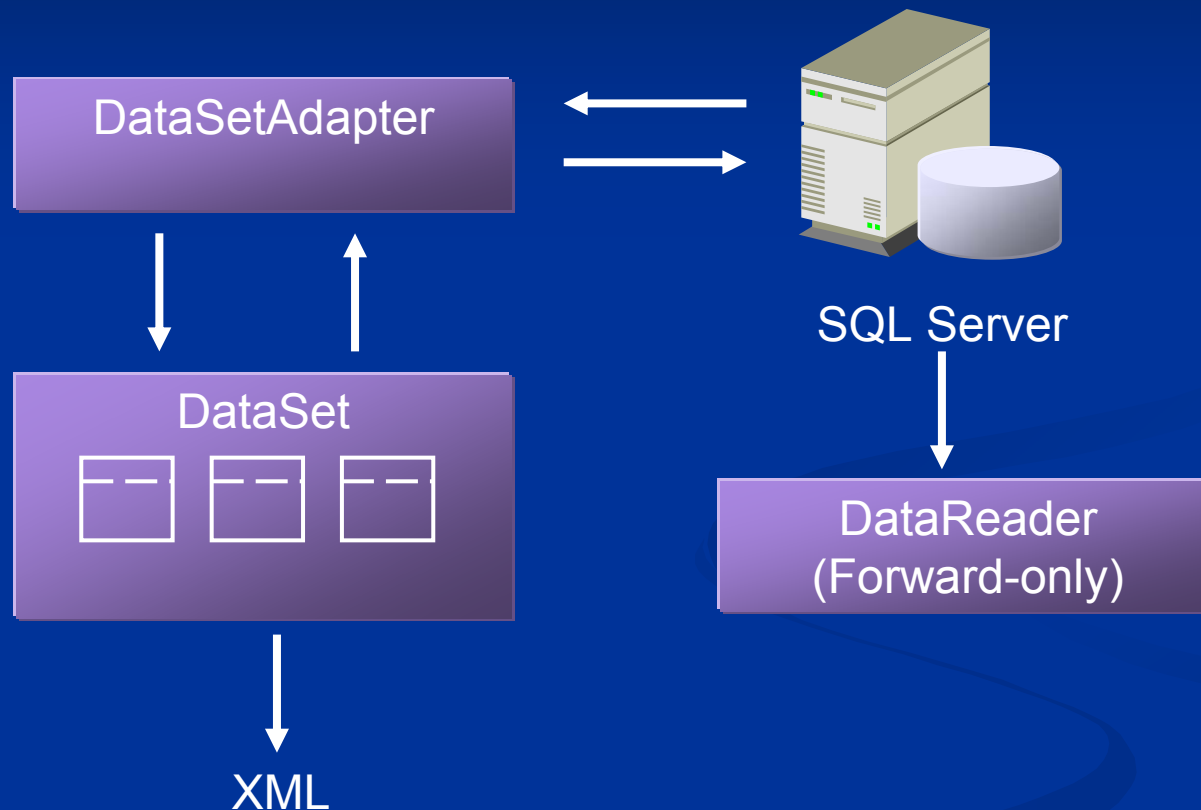
# Data and ADO.NET

## The Evolution of ADO to ADO.NET

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

# Data and ADO.NET

## ADO.NET Classes



# Data and ADO.NET

## XML Support

- Rich XML-supporting classes
  - for reading XML from files
  - writing XML from files
  - navigation
  - XSL transformations
  - DataSets

# Data and ADO.NET

## System.XML

