NY SYSTEMS Typical N-tier application architecture





.NET and Java: application pla

.NET

- The .NET Framework
- Java
 - Java application servers
 - Products include:
 - IBM WebSphere Application Server
 - BEA WebLogic Application Server
 - Sun iPlanet Application Server
 - Oracle Application Server
 - Many others



.NET vs. Java: standard

• .NET Framework class library

- Defined by Microsoft
- Somewhat Windows-oriented
- Organized into a hierarchy of namespaces

J2SE, J2EE

- Defined by Sun and the Java Community Process
- Not bound to any operating system
- Defined as packages and interfaces



Class Libraries



NY SYSTEMS

.NET Class Library

I0

- GUI Programming
- System Information
- Collections
- Components
- Application Configuration
- Connecting to Databases (ADO.NET)
- Tracing and Logging
- Manipulating Images/Graphics

NY SYSTEMS Class Library

- Interoperability with COM
- Globalization and Internationalization
- Network Programming with Sockets
- Remoting
- Serialization
- > XML
- Security and Cryptography
 - Threading
 - Web Services

NY SYSTEMS

The .NET Framework

What is Microsoft .NET?

- A programming model: CLR + Classes
- XML Web services
- Server and Client software and tools





Common Language Runtime

- It's a VM (Java-like) on which any (supported) language can run.
- Why a VM?
 - Memory Protection
 - Cross-language
 - Support for strong-typing *across languages* (the data are typed)
 - Thread support

```
JIT compilation in the VM
```

NY SYSTEMS Languages in CLR

Language of choice is C# ("C-sharp") a Java-like language

- No inner classes
- Better type checking

Other languages will run on CLR, but only within the CLR constraints

- Visual Basic, JScript are full fledged CLR languages
- For example, only C++ that is VM-safe will run
- That subset looks much like C#
- Under CLR, all languages get object features
 - Inheritance used extensively
 - Every language gets constructors



Languages compile to MSIL

- Languages compile to MSIL (Microsoft Intermediate Language)
 - Can you say "bytecodes"?
- MSIL is shipped in *portable executable* (PE) units
 - Can you say .class files or applets?
- An application is made up of *assemblies*



In general, a static assembly can consist of four elements:

- The <u>assembly manifest</u>, which contains assembly metadata.
- Type metadata.
- Microsoft intermediate language (MSIL) code that implements the types.
- A set of resources.

MyAssembly.dll

Assembly metadata

Type metadata

MSIL code

Resources



Assemblies can be spread acros



NY SYSTEMS Assemblies are the security un

Each assembly has a set of corresponding grants

- Each grant allows certain permissions
 - DnsPermission, Environment, FileDialog, FileIO, IsolatedStorage, Reflection, Registry, Security, UI, WebPermission, SocketPermission
- The set of grants establishes a security policy



Class Library

- Data classes support persistent data management and include SQL classes.
 - XML classes enable XML data manipulation and XML searching and translations.
- Windows Forms support development of Windows GUI applications across CLR
- Web Forms include classes that enable you to rapidly develop web GUI applications.

NY SYSTEMS System.Object

- Public methods:
 - Equals
 - GetHashCode
 - GetType
 - ToString
 - Overriding inherited behaviors is common



NY SYSTEMS Web, Windows, Whatever

- Part of the idea is to smooth transitions between Windows and Web
- Web interfaces become easier for Windows developers
- Windows apps become .NET Web-based apps

NY SYSTEMS

Data <-> XML, Everywhere

- All CLR data can be serialized to XML
- All XML can be expanded into CLR data
- Thus, anything can be shipped around on the Web
- Typing through XML Schema





<rpre><xsd:complexType name="Person">

<xsd:sequence>

<xsd:choice>

<rpre><xsd:element name="name" type="xsd:string"</pre>

xsi:nillable="true" />

<xsd:element name="id" type="xsd:string" />

</xsd:choice>

<xsd:any processContents="lax"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="AgedPerson">

<xsd:complexContent mixed="false">

<xsd:extension base="target:Person">

<xsd:choice>

<xsd:element name="age" type="xsd:double" />

<xsd:element name="timeOnEarth" type="xsd:double" />

</xsd:choice>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:element name="don" type="target:Person" />



Example Instance

<ns:don

>

xmlns:ns="uuid:048b2fa1-d557-473f-ba4cacee78fe3f7d"

<name>Don Box</name>

<niceStuffForDon/>

</ns:don>



Second Example Instance

<ns:don

>

xmlns:ns="uuid:048b2fa1-d557-473f-ba4cacee78fe3f7d"

xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"

xsi:type="ns:AgedPerson"

<name>Don Box</name>

<niceStuffForDon/>

<age>26</age>

</ns:don>