

# **.NET and J2EE Strategies for Interoperability**

**Simon Guest**

Program Manager

.NET Enterprise Architecture Team

Microsoft Corporation

**Chris Stewart**

HP ISS Microsoft Infrastructure  
Solutions



# If I had a \$...

- Which of my systems are interoperable?
  - What can .NET interoperate with today?
  - What are the challenges and the pitfalls?
  - How do I connect my 'boxes' together?
  - Interoperability is Web services, right?
  - I've heard Web services are slow / insecure
  - But, how do I ensure that I don't take a proprietary route?
  - Oh, did I forget to mention? We're a J2EE shop...
  - Oh, did I forget to mention? We're a MQ Series shop...
  - Oh, did I forget to mention? We're a IBM/VMS/SunOS\* shop...
- \* Replace as applicable

# Agenda

Interoperability Fundamentals  
Legacy Integration Strategies  
Web Service Interactions  
Dealing with Complex Data Types  
Products and Technology  
Web Services Architectures  
Conclusion, More Info and Q&A



**PROLIANT SERVERS**



Java 2 Enterprise  
Edition

# Agenda

## Interoperability Fundamentals

Legacy Integration Strategies  
Web Service Interactions  
Dealing with Complex Data Types  
Products and Technology  
Web Services Architectures  
Conclusion, More Info and Q&A



**PROLIANT SERVERS**



Java 2 Enterprise  
Edition

# Interoperability

## ■ Formal Definition

- The capability to communicate, execute programs, or transfer data among various functional units **in a manner that requires the user to have little or no knowledge of the unique characteristics of those units.**

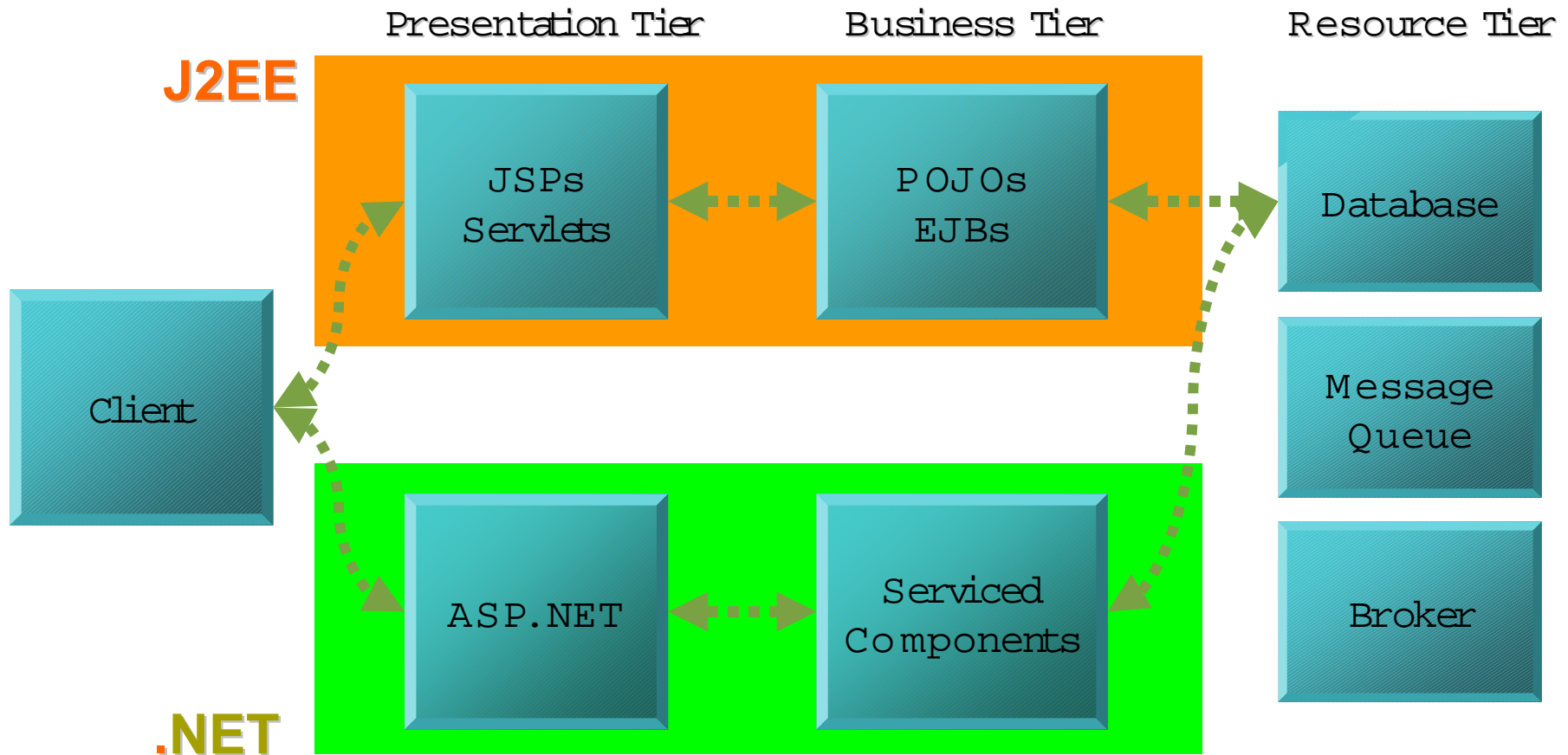
[ISO/IEC 2382 Information Technology Vocabulary]

# Interoperability

- Interoperability Enables
  - Re-use of Existing Systems
    - Integration of legacy systems
    - Extends life of current systems and knowledge*
  - Proof of Concepts
    - Rip and replace is no longer an option
    - Increases technical agility*
  - Migration
    - Migration is not an 'overnight' process
    - Allows for planned, correctly executed migrations*
  - Potentially Lower Project Costs
    - Development time and resource cost savings
    - Faster Go-to-Market (GTM)

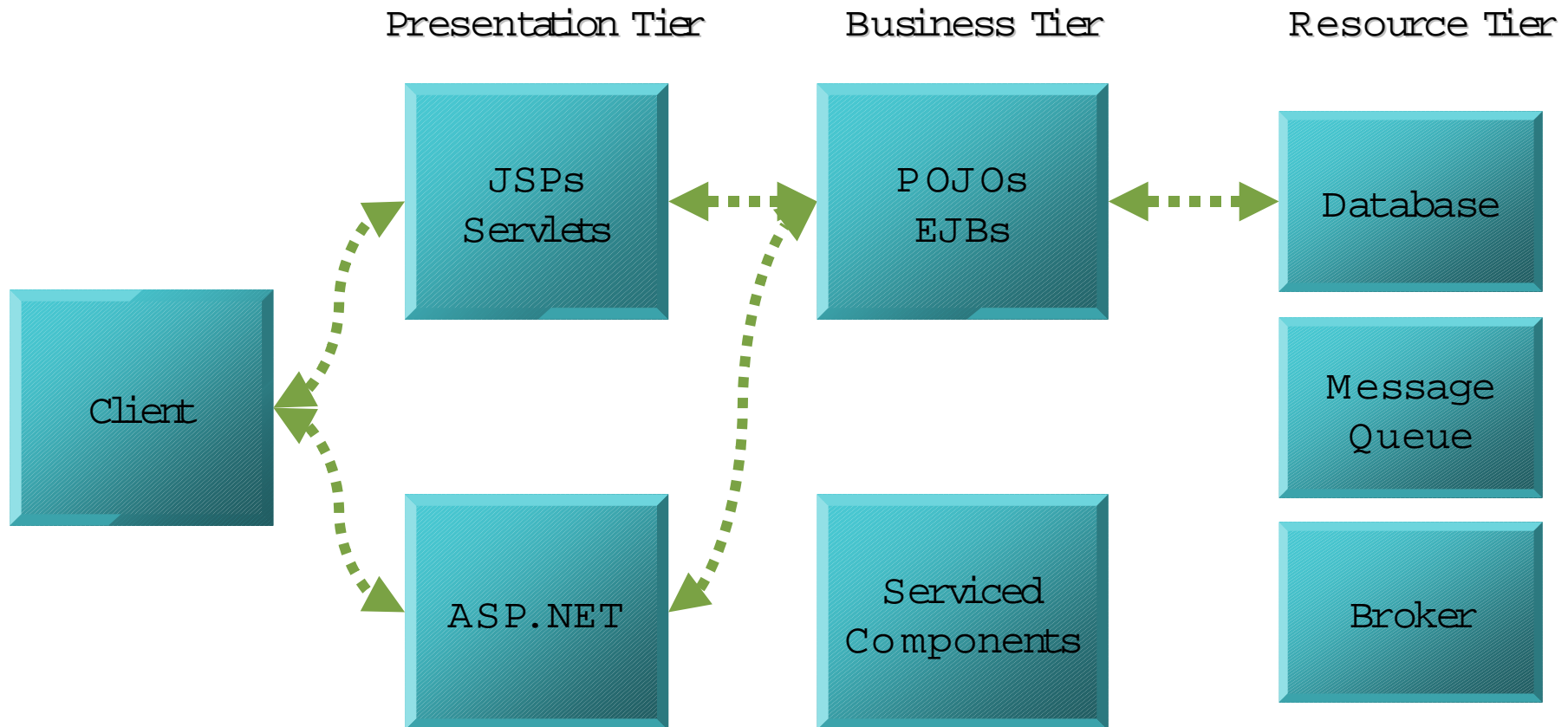
# Interoperability Requirements

## Technology Aligned Development



# Interoperability Requirements

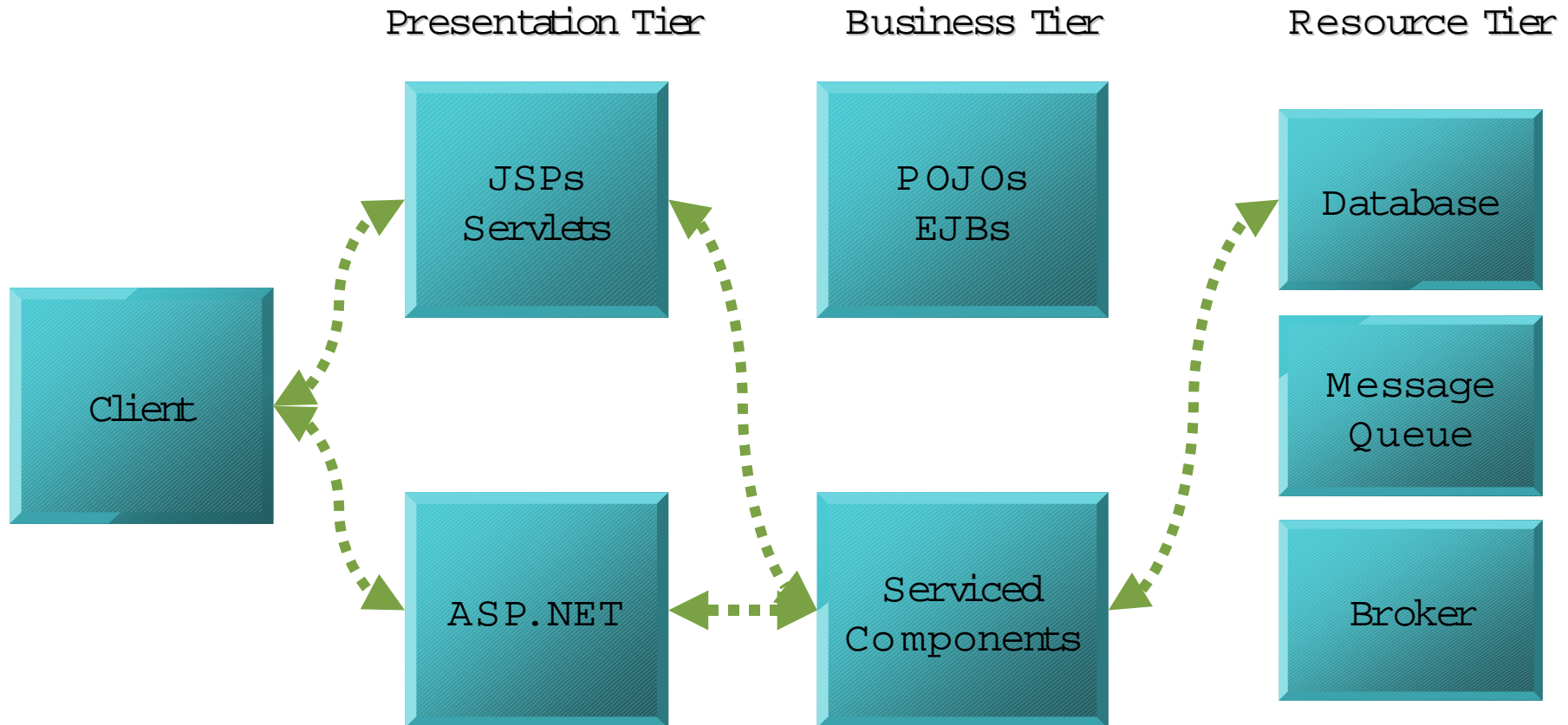
## Scenario 1: Interoperability at the Presentation Tier





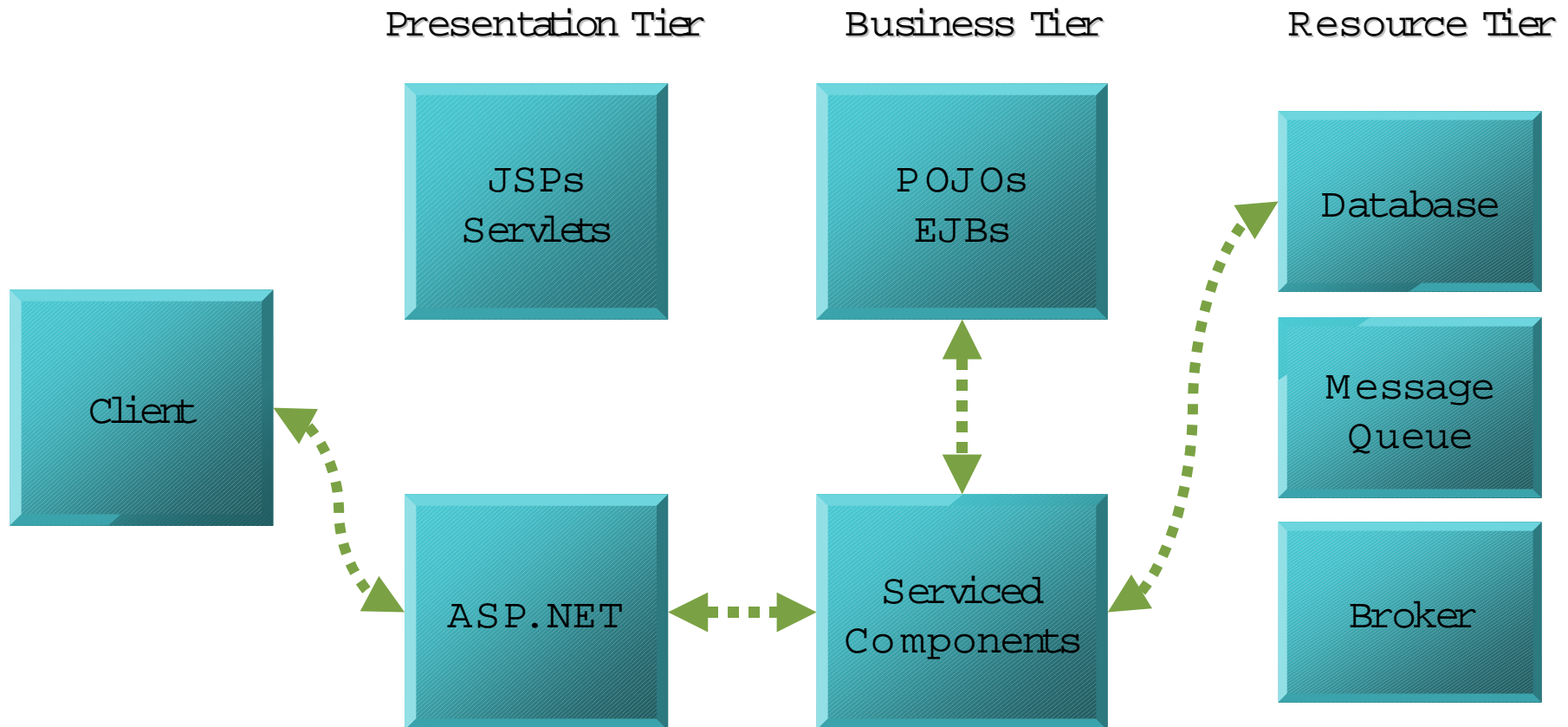
# Interoperability Requirements

## Scenario 1: Interoperability at the Presentation Tier



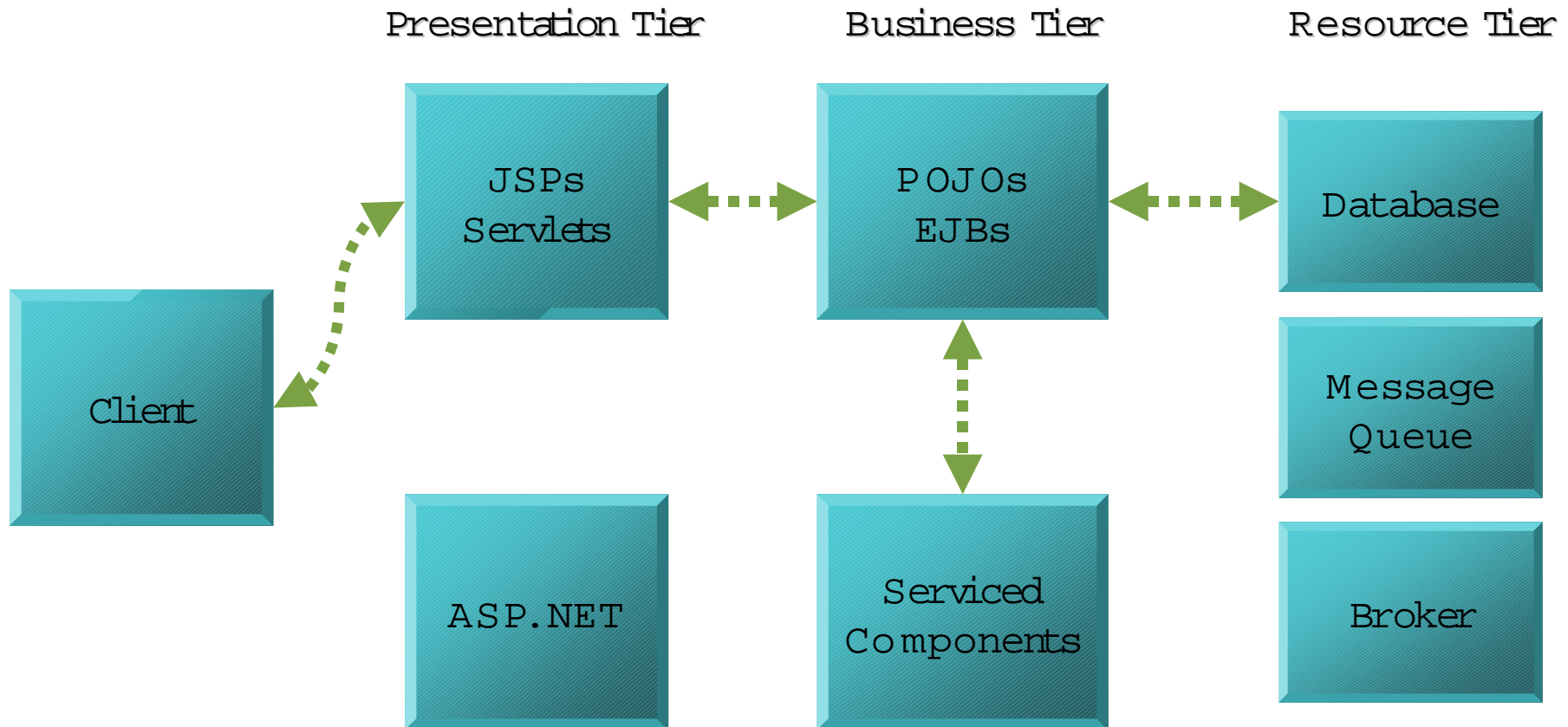
# Interoperability Requirements

## Scenario 2: Re-use of Business Tier Components



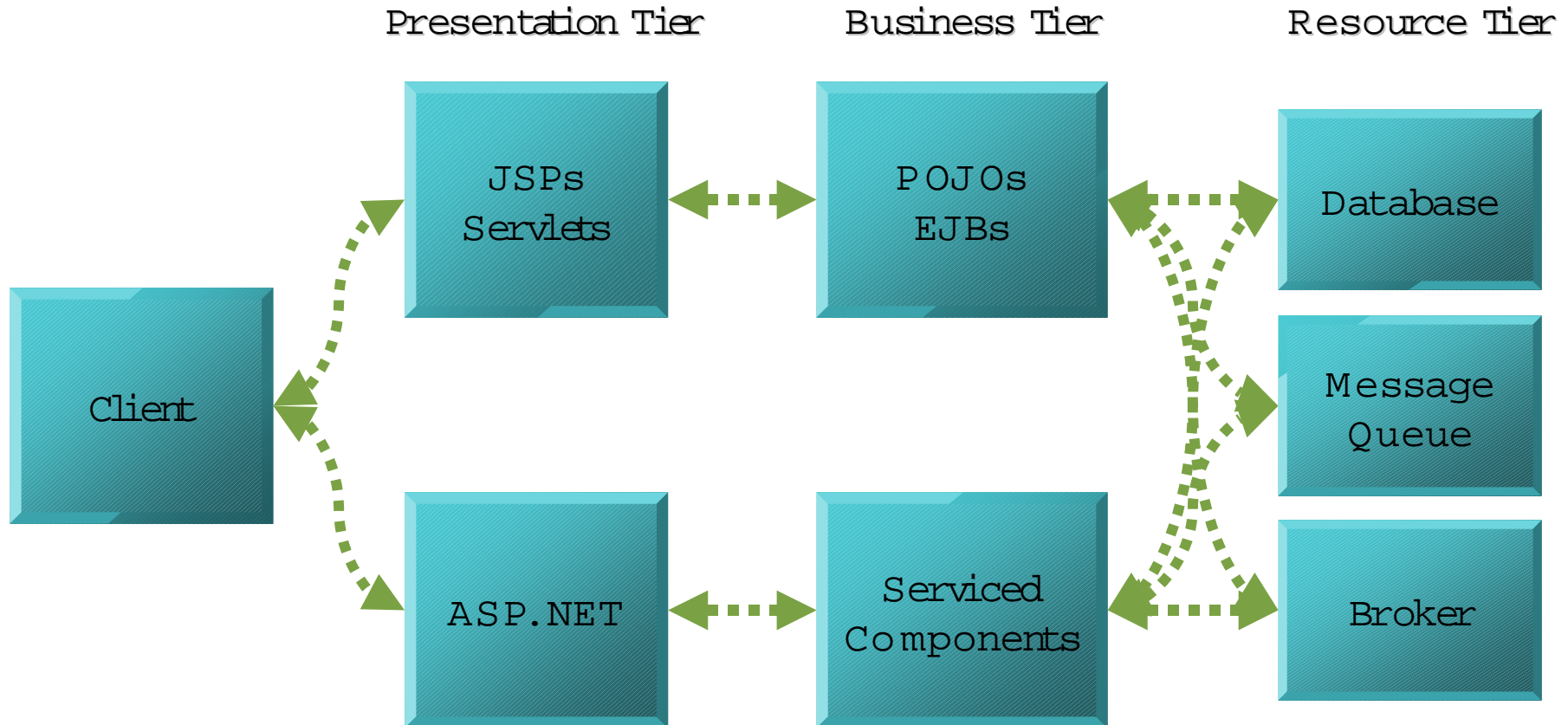
# Interoperability Requirements

## Scenario 2: Re-use of Business Tier Components



# Interoperability Requirements

## Scenario 3: Re-use of Shared Resources



# Agenda

Interoperability Fundamentals  
Legacy Integration Strategies  
Web Service Interactions



Dealing with Complex Data Types

PROLIANT SERVERS

Products and Technology  
Web Services Architectures  
Conclusion, More Info and Q&A



Java 2 Enterprise  
Edition

# Complex Data Types

## Beyond the 'Hello World' Example

Presentation Tier

Business Tier

Resource Tier

Client	Full Name	John Doe	se
	Company Name	Microsoft	
	Address	(1 Main St., Seattle. WA.)	
	Work Tel	425 703 5839	ge e
	Cell Tel	425 395 4082	
	Email	johndoe@microsoft.com	
	Last Updated	4 Jan 2003, 11:23.37am	er

# Complex Data Types

## Beyond the 'Hello World' Example

- .NET and J2EE Data Types
  - Basic data types do not necessarily match
    - `java.lang.String == System.String?`
  - Data types in one do not exist in the other
    - `System.Data.DataSet`
    - `java.sql.ResultSet`
- Two ways of converting the Data
  - Binary Serialization
  - XML (Parsing or Serialization with XML Schema)

# Complex Data Types

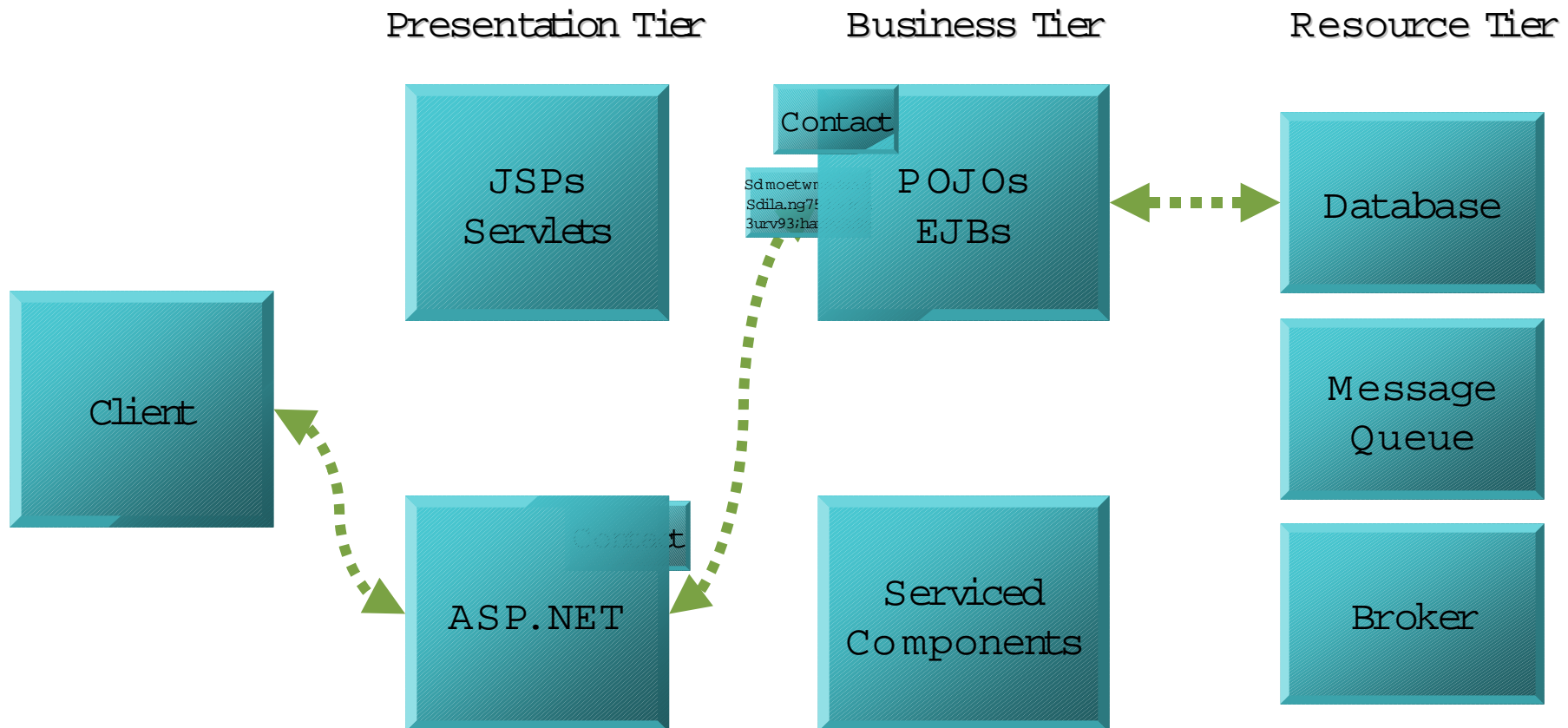
## Beyond the 'Hello World' Example

- Binary Serialization
  - Mechanism where Complex Data Types are converted to a stream of bytes
  - De-serialization must occur to same type
  - Serialization relies on Formatters
    - Both parties must agree on the formatter
    - .NET Fx 1.0 and J2SE 1.4 binary formatters are incompatible
    - .NET Fx 1.0 binary formatter can be licensed



# Complex Data Types

## Beyond the 'Hello World' Example



# Complex Data Types

## Beyond the 'Hello World' Example

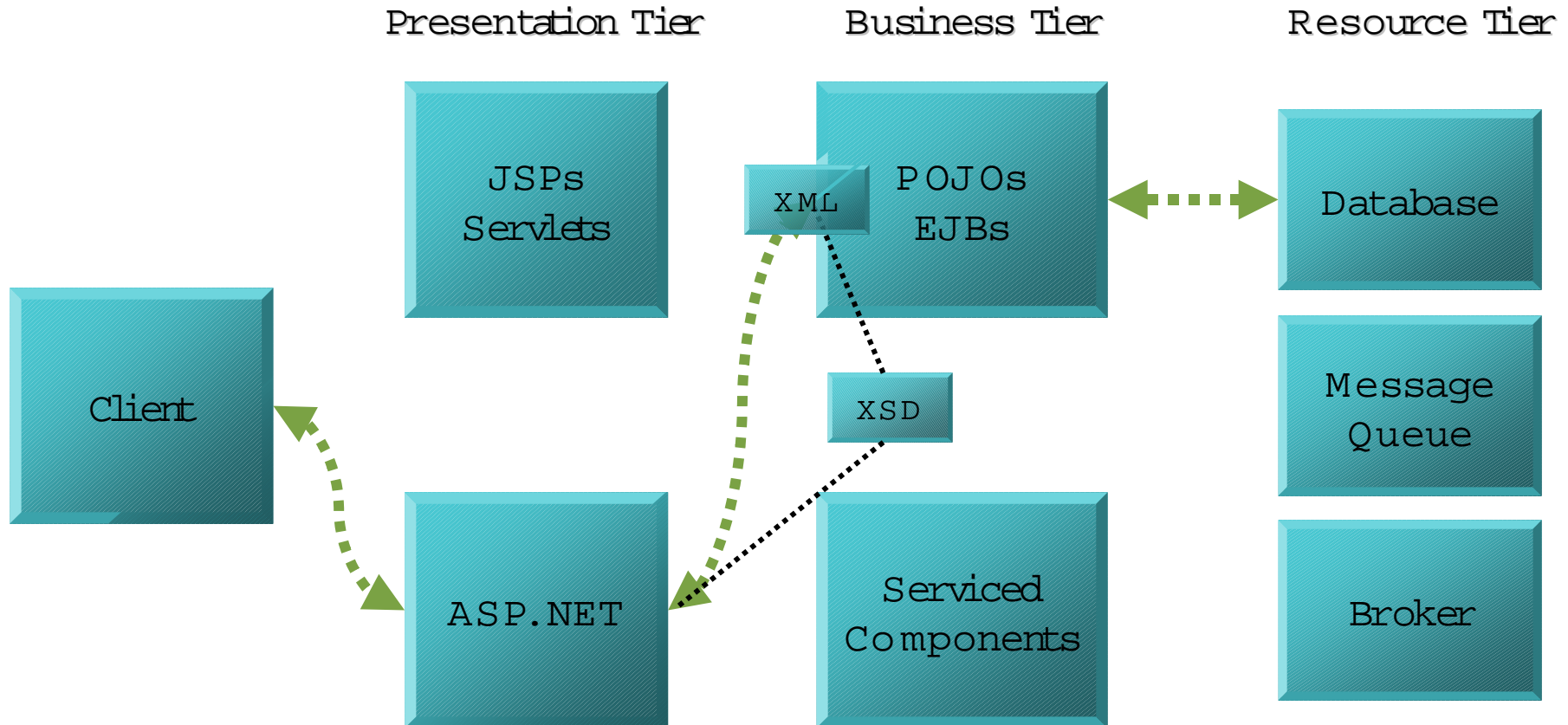


### ■ XML Schema

- Defines a structure and common datatypes of an XML Document
  - Stored in an XSD, which can be remote and shared
  - Provides arbitration for type mapping
    - `java.lang.String == xs:String == System.String`
- Cross-type possible, providing XML document is still valid

# Complex Data Types

## Beyond the 'Hello World' Example



# Complex Data Types

## Beyond the 'Hello World' Example

- Any Data Type, 100% Guaranteed ?
  - Some types are just difficult to send
    - Hashed types cannot be serialized (both runtimes)
    - .NET Datasets are dynamic
  - #1 Challenge in Achieving Interoperability
  - Interoperability Recommendation:
  - Create Unit Tests
    - Model the data in UML
    - Create repeatable test harnesses
    - Then code the application
    - (Also useful for performance testing)

*demo*

# **Code Walkthrough Serialization / De-serialization of XML as String with Validation**



# Complex Walkthrough J2EE

```
/* Class Interface Source File */  
  
import electric.xml.*;  
  
public interface IHelloXml  
{  
    Document getXml ();  
    Document getXml (  
        String strInFriend,  
        String strInMessage );  
}
```

# Java Class Implementation

```
import electric.xml.*;

public class HelloXml
    implements IHelloXml
{
    protected String strFriend;
    protected String strMessage;
    protected String strSchema;
    protected String strSchemaUrn;
    protected Document xmlDoc;
```

# Java Class Constructors

```
public HelloXml ()
{
    strFriend = "my friend";
    strMessage = "Hello";
    strSchema
        = "http://localhost/HelloXmlSchema.xsd";
    strSchemaUrn = "urn:HelloXml";
    InitXml ();
}

public HelloXml ( String strInFriend,
                  String strInMessage )
{
    strFriend = strInFriend;
    strMessage = strInMessage;
    InitXml ();
}
```



# Java Class Methods

```

protected void InitXml()    {
    String strXml = "<Hello><DocInfo><Schema>"
        + strSchema
        + "</Schema><Urn>" + strSchemaUrn
        + "</Urn></DocInfo><Friend>"
        + strFriend
        + "</Friend><Message>" + strMessage
        + "</Message></Hello>";

    try {
        xmlDoc = new Document( strXml );
    }
    catch( ParseException e ) {
        System.out.println( "Exception" );
    }
}

public String toString()    {
    return strMessage + ", " + strFriend + "!!!";
}

```

# Java Get / Set Pairs

```
public String getFriend()  
{  
    return strFriend;  
}
```

```
public void setFriend( String strIn )  
{  
    strFriend = strIn;  
  
    InitXml ();  
}
```

# Java Get / Set Pairs

```
public String getMessage()  
{  
    return strMessage;  
}
```

```
public void setMessage( String strIn )  
{  
    strMessage = strIn;  
  
    InitXml();  
}
```

# Get XML from Java

```
public Document getXml ()
{
    return xmlDoc;
}

public Document getXml( String strInFriend,
    String strInMessage )
{
    strFriend = strInFriend;
    strMessage = strInMessage;

    InitXml ();
    return xmlDoc;
}
} // Class terminator brace
```

# GLUE Web Service Wrapper

```
import electric.registry.Registry;
import electric.server.http.HTTP;

public class PublishHelloXml
{
    public static void main( String[] args )
        throws Exception
    {
        HTTP.startup(
            "http://localhost:8004/glue" );
        Registry.publish( "helloXml",
            new HelloXml() );
    }
}
```

# ASP.NET Consumer of Java Web Service



- Move into ASP.NET implementation
- Web Service to consume Java Web Service
- Includes instantiation of Java Web Service types in C#

# Complex Walkthrough: ASP.NET

```
/* Validation code adapted from
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfsystemxmlxmlvalidatingreaderclasssschemastopic.asp */
[WebMethod]
public System.Xml.XmlDocument ConsumeXmlReturnXmlValidate(
string strFriend, string strMessage )
{
    System.Xml.XmlDocument xDoc
        = new System.Xml.XmlDocument();
    string strResult = ConsumeXml( strFriend,
                                   strMessage );
    xDoc.LoadXml( strResult );

    string strSchema = GetSchema( xDoc );
    string strSchemaUrn = GetUrn( xDoc );
}
```

# ASP.NET / Java Interaction

```
[WebMethod]
```

```
public string ConsumeXml( string strFriend, string  
strMessage )  
{  
    // Change Names to Indicate What's Java & What's C#  
    HelloJavaXmlWs.HelloXml h  
        = new HelloJavaXmlWs.HelloXml ();  
    h.setFriend( strFriend );  
    h.setMessage( strMessage );  
    HelloJavaXmlWs.document doc;  
    doc = h.getXml ();  
    return ByteToString( doc.document1 );  
}
```



# XML Document Convention

- This Approach Presumes

- The organization has adopted a XML document convention of

```
<RootElement>
```

```
  <DocInfo>
```

```
    <Schema><!-- URL of Schema --></Schema>
```

```
    <Urn><!-- urn value --></Urn>
```

```
  </DocInfo>
```

```
  <!-- Rest of document -->
```

```
</RootElement>
```

- **<RootElement>** can be any value

**<Hello>**, as in the example

Anything else

- Fixed values

```
<DocInfo>
```

```
<Schema>
```

```
<Urn>
```

# Retrieve Schema and Urn

```
protected string GetSchema( System.Xml.XmlDocument
                           xDoc )
{
    System.Xml.XmlNode xNode;
    xNode = xDoc.SelectSingleNode(
        xDoc.DocumentElement.Name
        + "/DocInfo/Schema" );
    return xNode.InnerText;
}

protected string GetUrn( System.Xml.XmlDocument
                        xDoc )
{
    System.Xml.XmlNode xNode;
    xNode = xDoc.SelectSingleNode(
        xDoc.DocumentElement.Name
        + "/DocInfo/Urn" );
    return xNode.InnerText;
}
```

# XML Validation

```
m_success = true;
System.IO.StringReader sReader = new
    System.IO.StringReader( strResult );
System.Xml.XmlTextReader xReader = new
    System.Xml.XmlTextReader( sReader );
System.Xml.XmlValidatingReader vReader = new
    System.Xml.XmlValidatingReader( xReader );
System.Xml.Schema.XmlSchemaCollection schema = new
    System.Xml.Schema.XmlSchemaCollection();

schema.Add( strSchemaUrn, strSchema );
vReader.Schemas.Add( schema );

//Set the validation event handler
vReader.ValidationEventHandler += new
    System.Xml.Schema.ValidationEventHandler
    (ValidationCallback);
```

# Callback on Invalid Document

```
public void ValidationCallBack(  
    object sender,  
    System.Xml.Schema.ValidationEventArgs args )  
{  
    m_success = false;  
}
```

# Complete and Return XML Data

```
//Read and validate the XML data.
while (vReader.Read()) {}
if( m_success )
{
    xDoc.LoadXml( strResult );
    return xDoc;
}
else
{
    xDoc.LoadXml( "<error><errorCode>Document failed"
        + "schema validation</errorCode></error>" );
    return xDoc;
}
```

# Agenda

Interoperability Fundamentals  
Legacy Integration Strategies  
Web Service Interactions  
Dealing with Complex Data Types

## Products and Technology

Web Services Architectures  
Conclusion, More Info and Q&A



PROLIANT SERVERS



Java 2 Enterprise  
Edition

# Interoperability Technologies

- Point to Point Interoperability
  - Direct, between two points, calls tend to be synchronous
  - Two technology options for .NET and J2EE Interoperability
    - .NET Remoting
    - XML Web Services

# Interoperability Technologies

## Point to Point Interoperability

- .NET Remoting
  - Specification created by Microsoft
    - 3<sup>rd</sup> Party Implementations for Java 1.2+
      - Intrinsyc Ja.NET (<http://www.intrinsyc.com>)
      - JNBridge Pro (<http://www.jnbridge.com>)
  - Interoperability Benefits
    - Binary communication over TCP
    - Pass by reference
    - .NET Framework support
    - Component Services support (via SOAP Activation)



# Interoperability Technologies

## Point to Point Interoperability



- XML Web Services
  - At the core of the .NET Framework
  - Multiple Java Implementations
  - Interoperability Benefits
    - Standards based (XML, SOAP, WSDL, UDDI)
    - Firewall (and inter-organization) friendly
    - Extensible
    - Massive industry momentum

# Interoperability Technologies

## Point to Point Interoperability



- XML Web Services
  - 3<sup>rd</sup> Party implementations for Java
    - Apache AXIS 1.0
    - BEA WebLogic Platform
    - GLUE 4.0.1 (The Mind Electric)
    - IBM ETTK 3.3 and Websphere Support
  - Interoperability Recommendation: Abstract Vendor
    - No **java.webservices.\*** package
    - Have to use **com.ibm.\*** (IBM) or **com.electric.\*** (GLUE)
    - Create abstract layer (**com.myorg.webservices**) then call
    - Good practice for .NET as well

# Interoperability Technologies

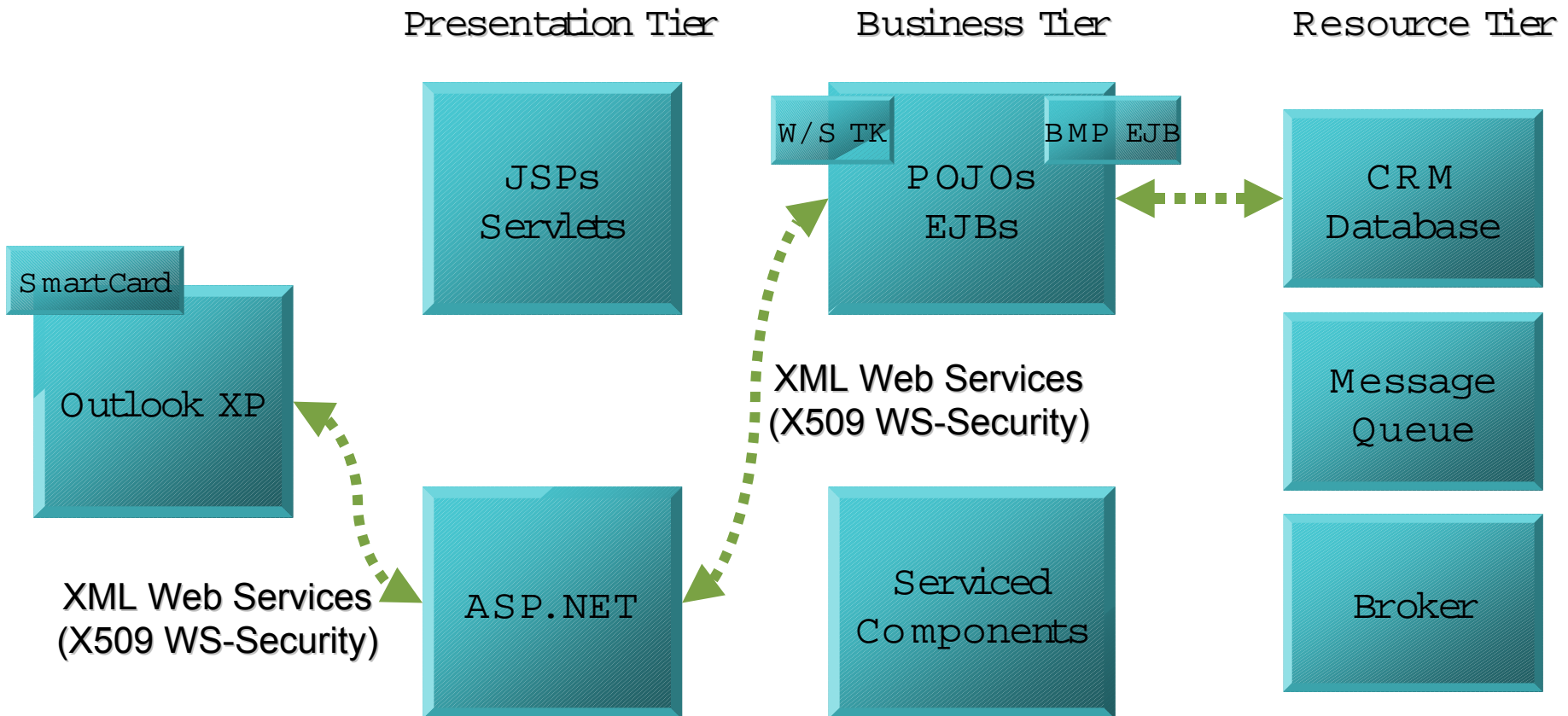
## Point to Point Interoperability



- XML Web Services Interoperability Today
  - Web Services Interoperability Org. (WS-I)
  - Vendors, System Integrators, Enterprise Customers
  - Goal to deliver resources, samples and tools
  - WS-I Basic Profile 1.0 Draft
    - (SOAP 1.1, WSDL 1.1, UDDI 2.0)
- Moving Forward
  - Latest Web Service Specifications
    - Moving from connectivity to application specific Interoperability
    - Microsoft WSE 1.0 (<http://msdn.microsoft.com/webservices>)
    - WS-Timestamp, WS-Security, WS-Routing, DIME

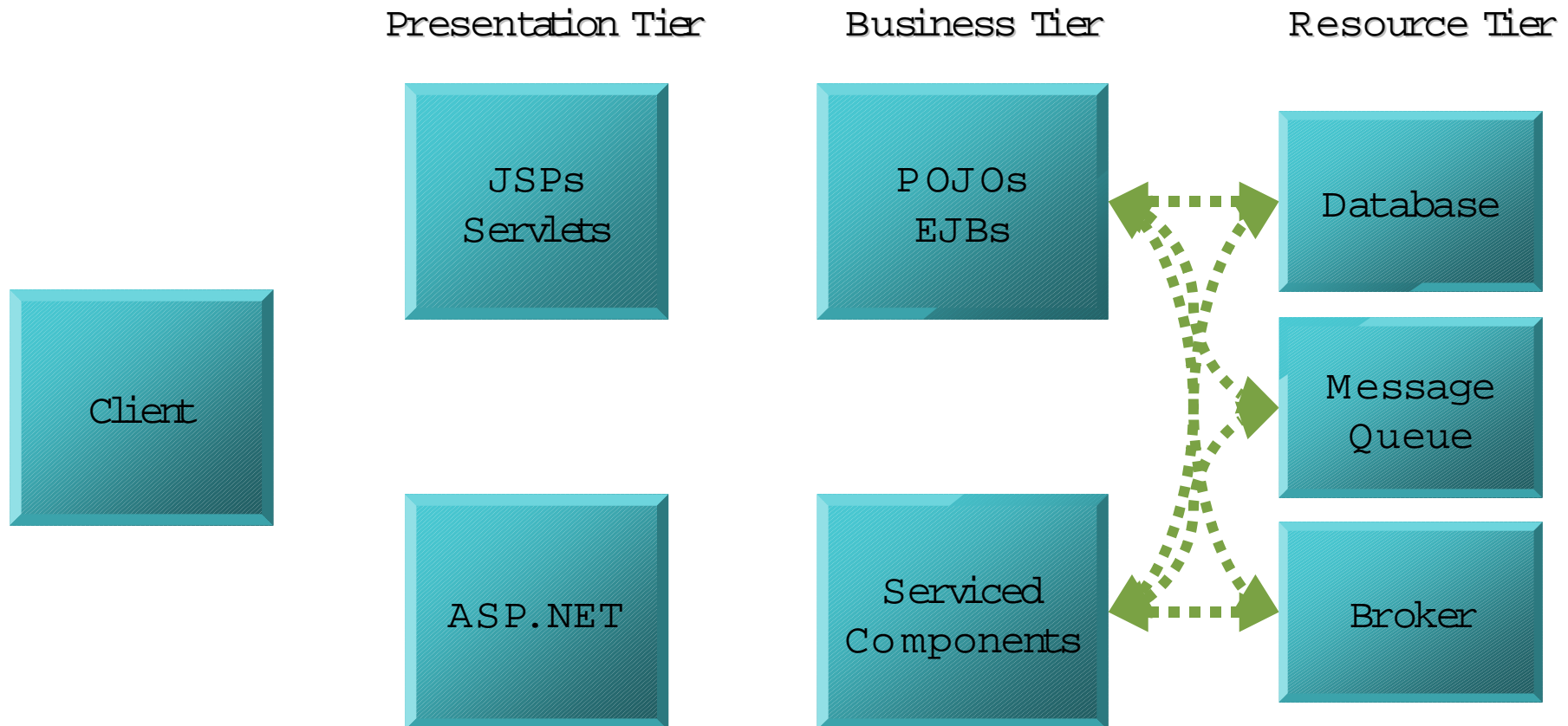
# Interoperability Solutions

## Solution: Outlook Client to existing, secure J2EE CRM Database



# Interoperability Technologies

## Interoperability at the Resource Tier

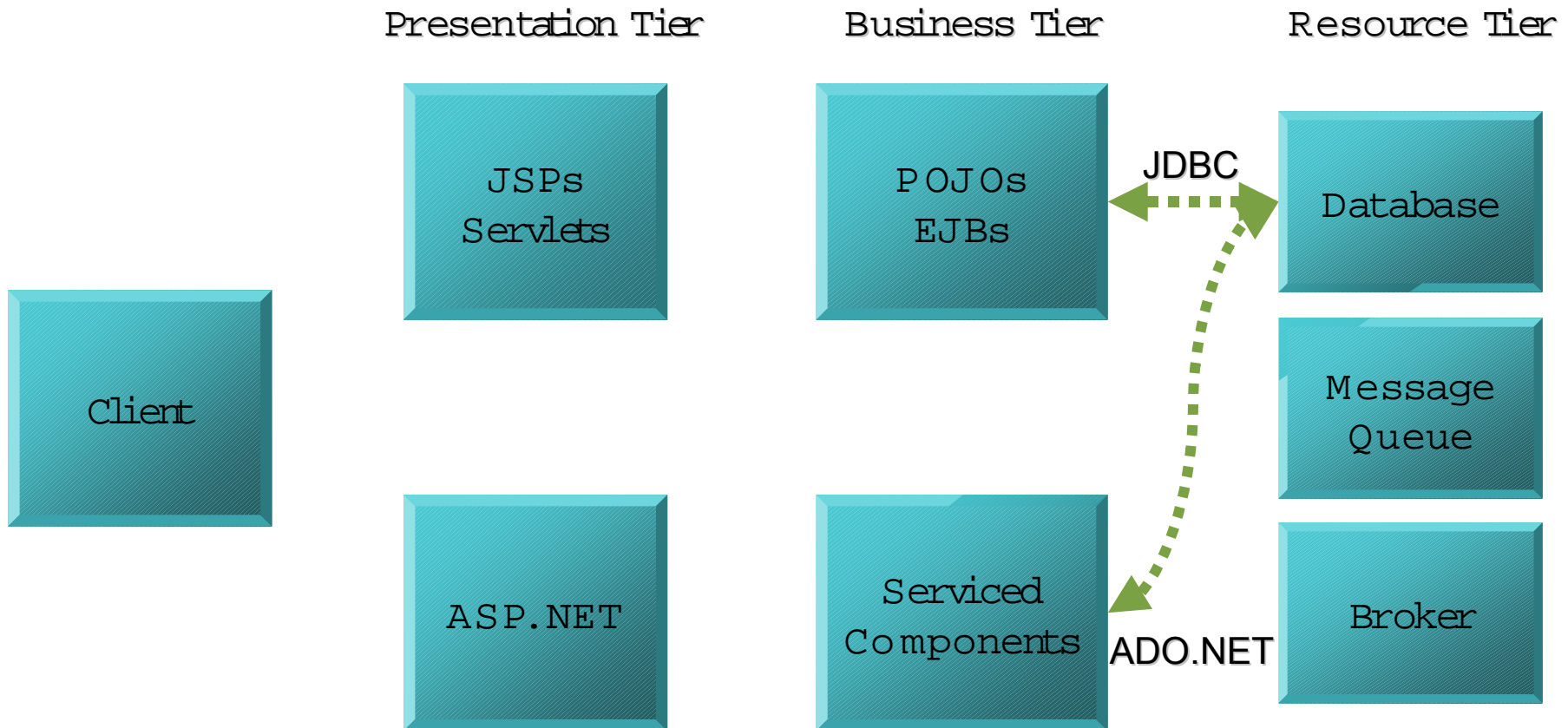


# Interoperability Technologies

- Interoperability at the Resource Tier
  - Arbitrated, mostly asynchronous, between two or more points
  - Can utilize point to point technology for connectivity
  - For .NET and J2EE Interoperability
    - Shared Database
    - Message Queue
    - Broker

# Interoperability Technologies

## Resource Tier – Shared Database



(XWS with  
SQL Server 2000)

# Interoperability Technologies

## Interoperability at the Resource Tier

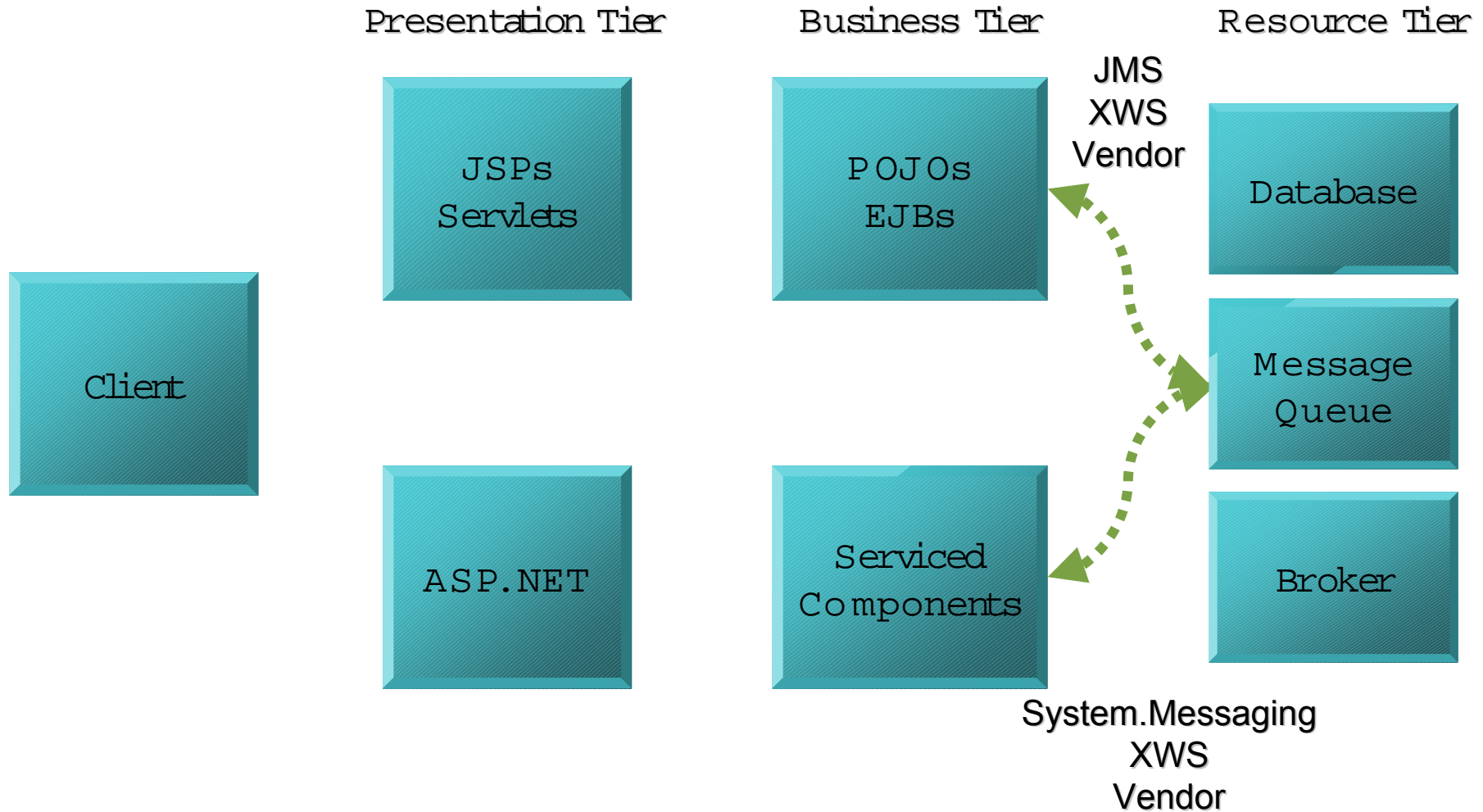


- Shared Database
  - Enabled by use of Open Access Database Drivers
    - Oracle OCI Driver for ADO.NET
    - SQL Server 2000 Driver for JDBC
    - Many vendor and 3<sup>rd</sup> Party Implementations
  - Can be used with Serialization for Complex Data Types
  - Interoperability Benefits
    - Universally recognized way to store data
    - Stored Procedures allow logic to be written once
    - Database locking and transaction support



# Interoperability Technologies

## Resource Tier – Message Queue



# Interoperability Technologies

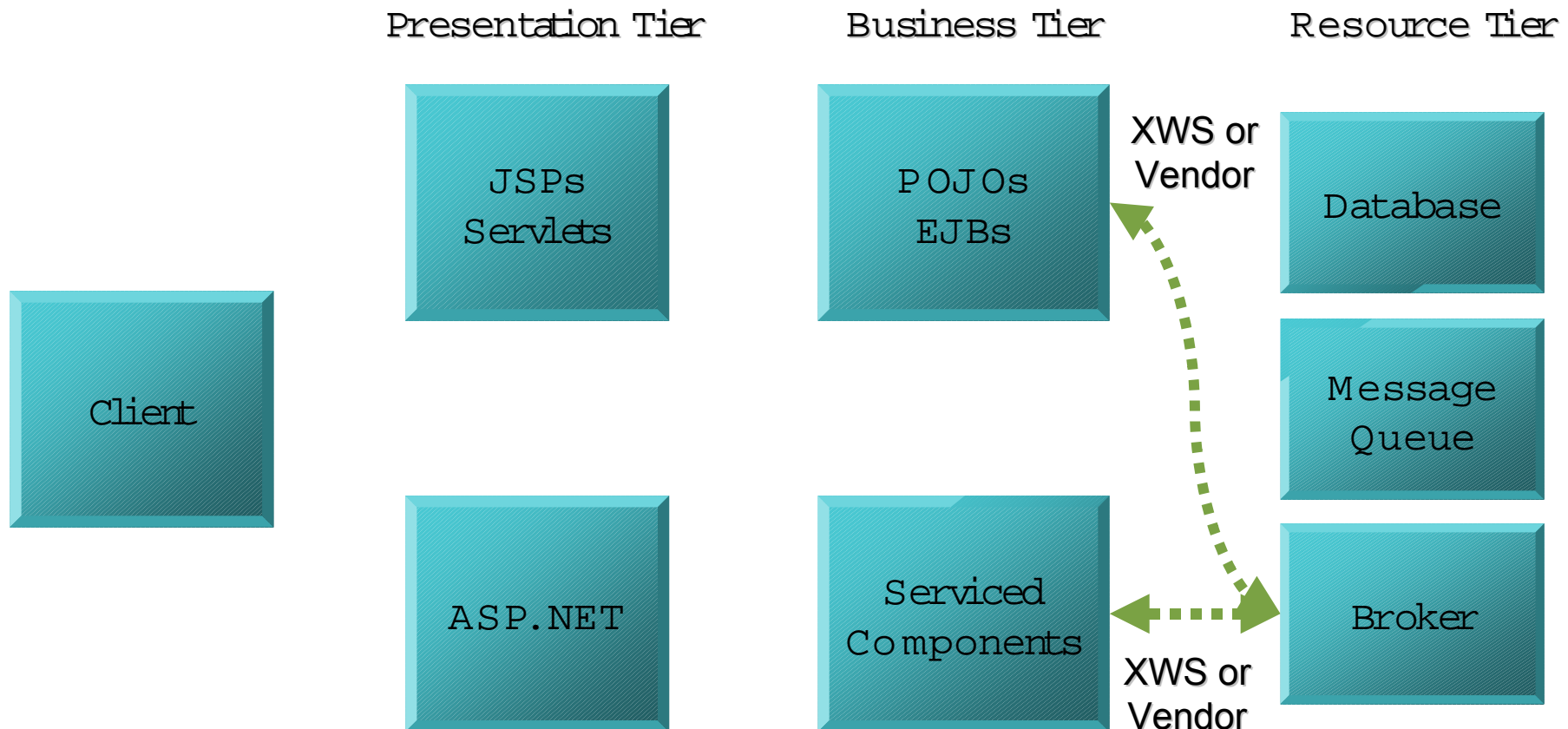
## Interoperability at the Resource Tier



- Message Queue
  - Vendor-rich Selection
    - MSMQ 3.0 (HTTP SRMP endpoint)
    - IBM WebSphere MQ (.NET Adapter available via SupportPac)
    - J2EE Application Server Vendor via JMS API
    - Fiorano, Silverstream, Sonic, Tibco etc.
  - Interoperability Benefits
    - Ideal for N-N Interoperability Scenarios
    - Transaction Support
    - Reliable Messaging Support
    - Publish / Subscribe

# Interoperability Technologies

## Resource Tier – Broker



# Interoperability Technologies

## Interoperability at the Resource Tier



- Broker
  - Technologies
    - Microsoft BizTalk Server 2002
    - Microsoft BizTalk Server 2004 Beta 1
  - Interoperability Benefits
    - Ideal for N-N Interoperability Scenarios
    - Transaction Support
    - Multiple Adapter Support
      - Connection to legacy and disparate systems
    - Message Transformations
    - Orchestration

# Agenda

Interoperability Fundamentals  
Legacy Integration Strategies  
Web Service Interactions  
Dealing with Complex Data Types  
Products and Technology

Web Services Architectures

Conclusion, More Info and Q&A

Microsoft  
.net

PROLIANT SERVERS

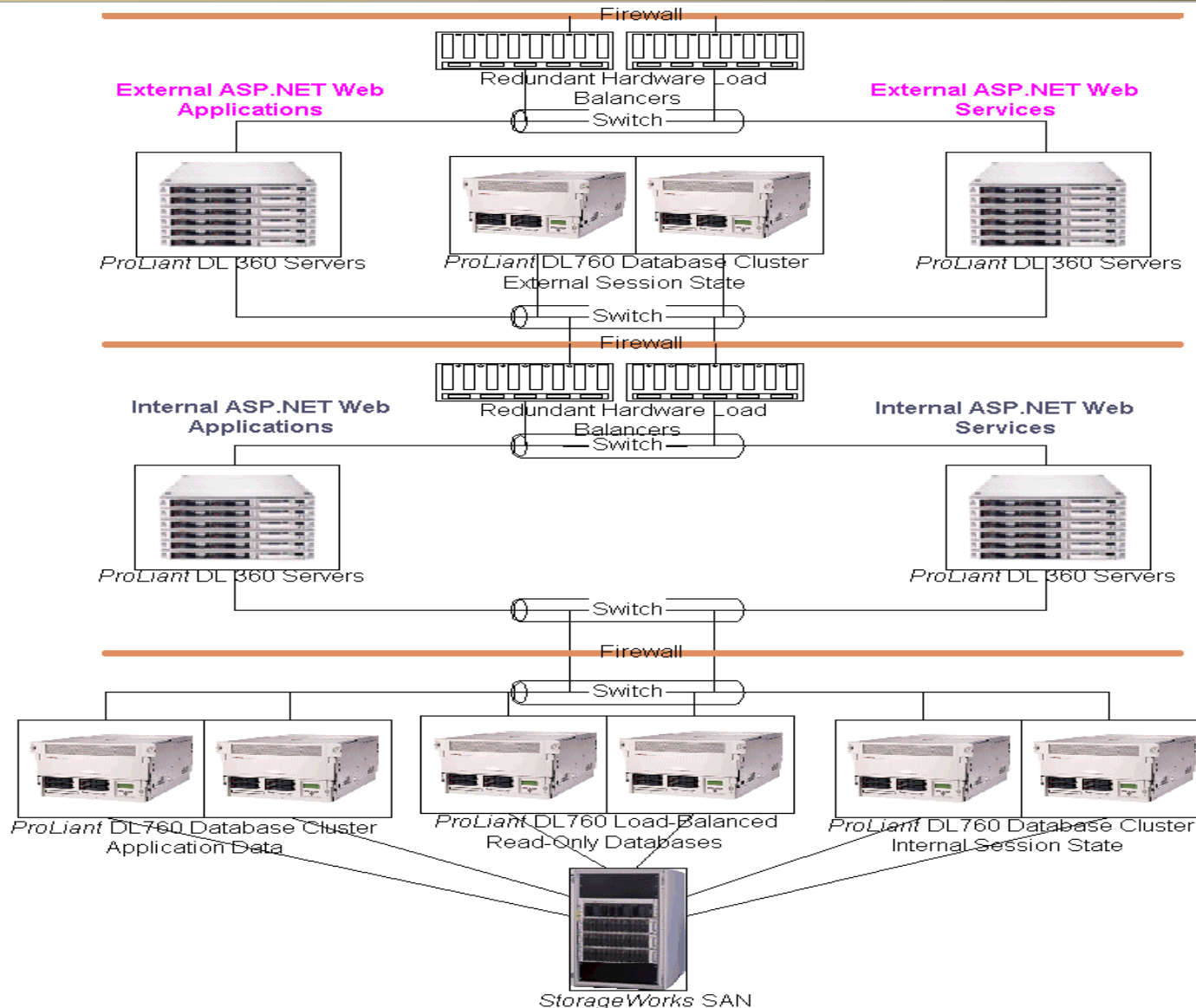


Java 2 Enterprise  
Edition

# Web Services Architectures

- Qwest Implementation
  - Case Study Available:  
[http://activeanswers.compaq.com/aa\\_downloads/6/100/225/1/54001.doc](http://activeanswers.compaq.com/aa_downloads/6/100/225/1/54001.doc)
  - Free ActiveAnswers Registration Required
  - Session 1187, HP World 2003
- Qwest divided the DISA Web and Application Resources tier into a pair of network segments, each separated by a firewall
- Two pairs of redundant hardware load balancers, one each in front of the externally facing and internally facing server farms in the Web and Application Resources tier round out the architecture.

# Web Services Architectures

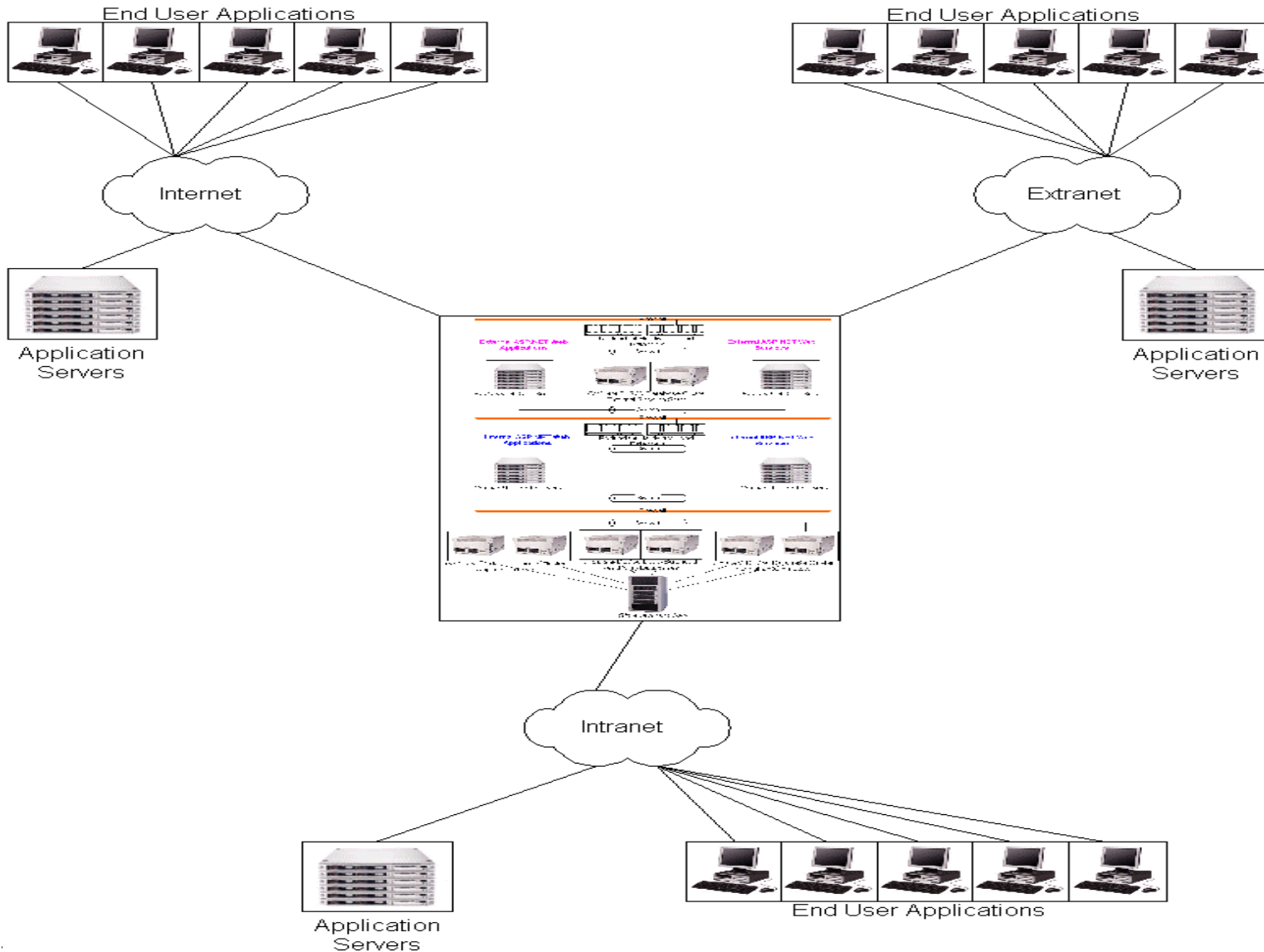


# Web Services Architectures

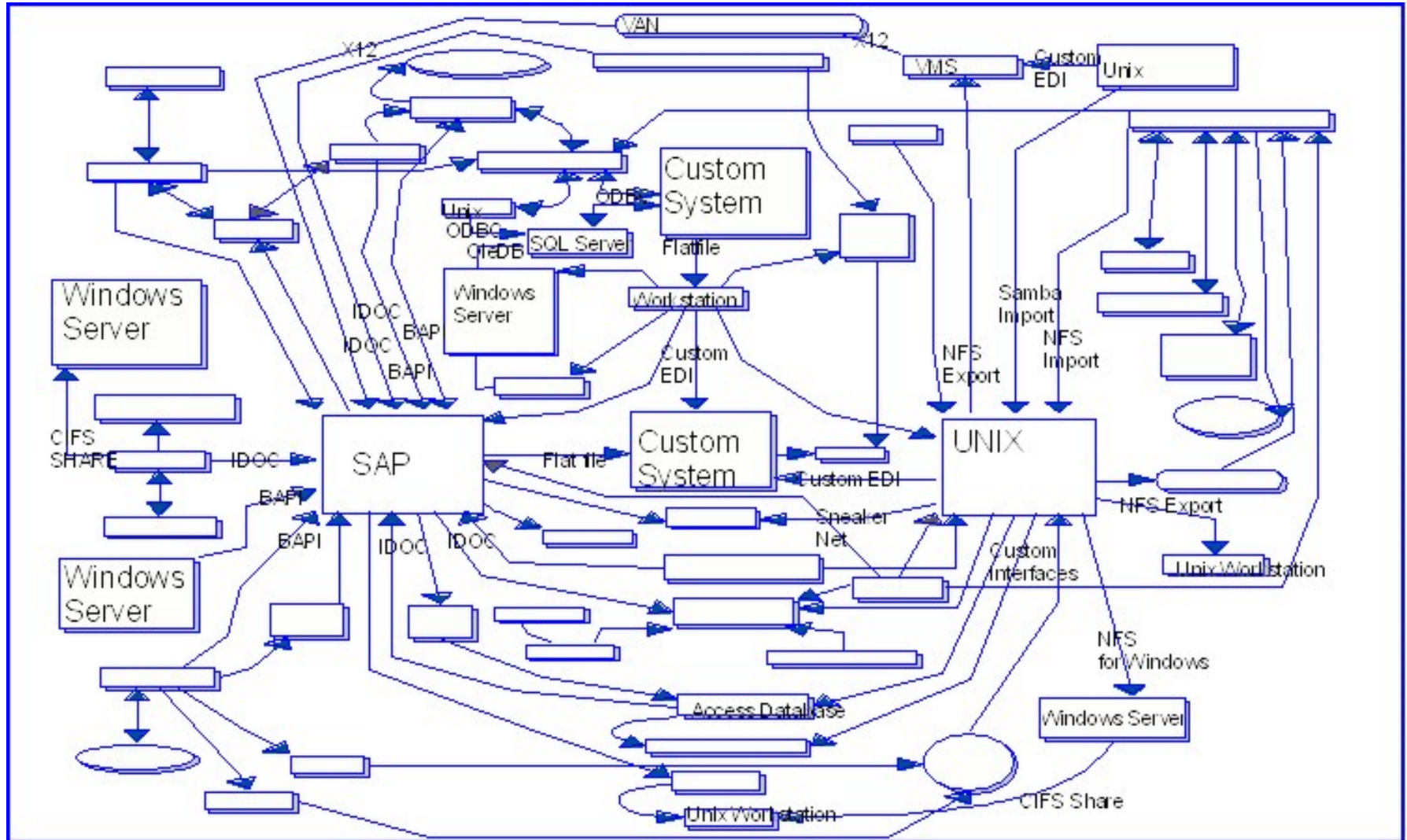
- Web Services Can Function Like Any Other HTTP Application
- Therefore Load Balancing Works for Scalability and Availability
  - Externally Facing and Internally Facing Architectural Infrastructure
  - Separate Load Balancing for Each
  - Increased Security of Important Business Logic
- Each Web Method Invocation can be made Independent of All Others
  - Helps with Application Scalability
  - Inherently Loose Coupling of Functionality



# Web Services Integration



# The Legacy Landscape



# Web Services Integration

- Unified Web Services Architecture can Host Multiple Implementations
  - ASP.NET
  - J2EE
  - Others
- Each Web Services Realm Can Be Hosted on an Independent Architecture
  - ASP.NET
  - J2EE
  - Others
- Interaction Points Across Realms Work Just Like Interactions with Other Applications
  - Standard XML, SOAP, HTTP/S, WSDL, etc.

# Agenda

Interoperability Fundamentals  
Legacy Integration Strategies  
Web Service Interactions  
Dealing with Complex Data Types  
Products and Technology  
Web Services Architectures

Conclusion, More Info and Q&A



PROLIANT SERVERS



Java 2 Enterprise  
Edition

# Interoperability: Next Steps

- What are Microsoft and HP Doing?
  - Publishing Content
    - MSDN Interoperability Articles
      - WS-Security
      - .NET Remoting and CORBA Interop (July)
    - MSDN Chats and WebCasts (Feb 2003)
    - MSDN Foundation Demos (Mar / Apr 2003)
  - Working with Customers
    - Many projects throughout the Field
    - Customers and Projects make the Interoperability message real

# hp ActiveAnswers Solutions Website




- ❑ Recommended Configurations
- ❑ Interactive Web-based Sizers
- ❑ Solution Guides
- ❑ Performance Guides
- ❑ Installation Checklists
- ❑ System Configurator
- ❑ Performance Analyzer

UNITED STATES

>> hp home >> products & services >> support & drivers >> solutions >> how to buy

>> contact hp search:  >> [more options](#)

ActiveAnswers  all of hp US



invent

### ActiveAnswers

- home
- >> solutions
- >> tools
- >> site map

---

#### member services

- >> new user
- >> what's new
- >> update profile
- >> logoff
- >> ActiveAnswers tour

---

#### What's New

HP delivers the quality manageability and innovation you expect from a world-class market leader of enterprise solutions. [Click here to find out what's new on ActiveAnswers.](#)

#### Cross-Industry Solutions

- >> [Customer Relationship Management](#)
- >> [Database & Business Intelligence](#)
- >> [ERP & Supply Chain](#)
- >> [Infrastructure & Architecture](#)
- >> [Internet & E-Commerce](#)
- >> [Messaging/Collaboration and Portals](#)

#### Industry Solutions

- >> [Service Providers](#)

#### About ActiveAnswers

Welcome to ActiveAnswers—the online solutions authority where HP shares its experiences and best practices for selling, planning, deployment and operation of IT solutions. It includes up-to-date technical information, and the configurator and sizing tools. Why Register? Because it's simply the best way to succeed with today's enterprise solutions. Registration is free and will only take a few minutes, allowing you complete access to the ActiveAnswers portfolio of tools, information, and User Community events. While registering, you can select your username and password, sign up for the ActiveAnswers eNews letter, as well as decide how much information you would like emailed to you. [Register Now!](#)

---

[privacy statement](#) [using this site means you accept its terms](#)

© 1994-2002 Hewlett-Packard Company

[www.hp.com/solutions/activeanswers](http://www.hp.com/solutions/activeanswers)

**Free, Fast Registration**

# .NET and J2EE Interoperability

## ■ Conclusion

- Interoperability Fundamentals
  - What is interoperability? Why interoperate?
  - Common scenarios for interoperability
- Dealing with Complex Data Types
  - Sending Data between Tiers
  - Standards that are helping promote interoperability
- Products and Technology
  - Enabling Point to Point Interoperability
  - Interoperability at the Resource Tier
- Conclusion, More Info and Q&A

# demo

## ■ Resources

- Outlook XP with .NET Smart Client Add-In  
[http://msdn.microsoft.com/library/en-us/dnoxpta/html/odc\\_dnscof.asp](http://msdn.microsoft.com/library/en-us/dnoxpta/html/odc_dnscof.asp)
- WS-Security Authentication Interop using X.509 Certificate (IBM WSTK 3.3.2)  
<http://msdn.microsoft.com/webservices/building/wse/default.aspx?pull=/library/en-us/dnwebsrv/html/wsejavainterop.asp>
- WS-Security Authentication Interop using X.509 Certificate (TME GLUE 4.0.1)  
<http://msdn.microsoft.com/webservices/building/wse/default.aspx?pull=/library/en-us/dnwebsrv/html/wsejavainterop2.asp>





# HP WORLD 2003

Solutions and Technology Conference & Expo

Interex, Encompass and HP bring you a powerful new HP World.



*demo*

# Outlook XP Client, CRM and WS-Security



# Backup

# Agenda

Interoperability Fundamentals  
Legacy Integration Strategies  
Web Service Interactions  
Dealing with Complex Data Types  
Products and Technology  
Web Services for Infrastructure  
Integration

HP Dynamic Internet Solutions  
Architecture

Web Services Architectures  
Conclusion, More Info and Q&A

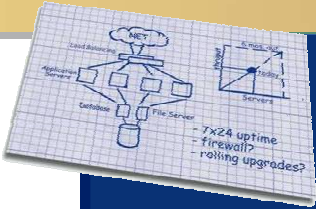
Microsoft  
.net

PROLIANT SERVERS



Java 2 Enterprise  
Edition

# hp Dynamic Internet Solutions Architecture (DISA)



Clients



Firewalls, Load Balancers, Cache, SSL, VPN



Web & Application Servers



Database, Storage  
(NAS & SAN)



Business Applications

Integration & Services

Operations & Management

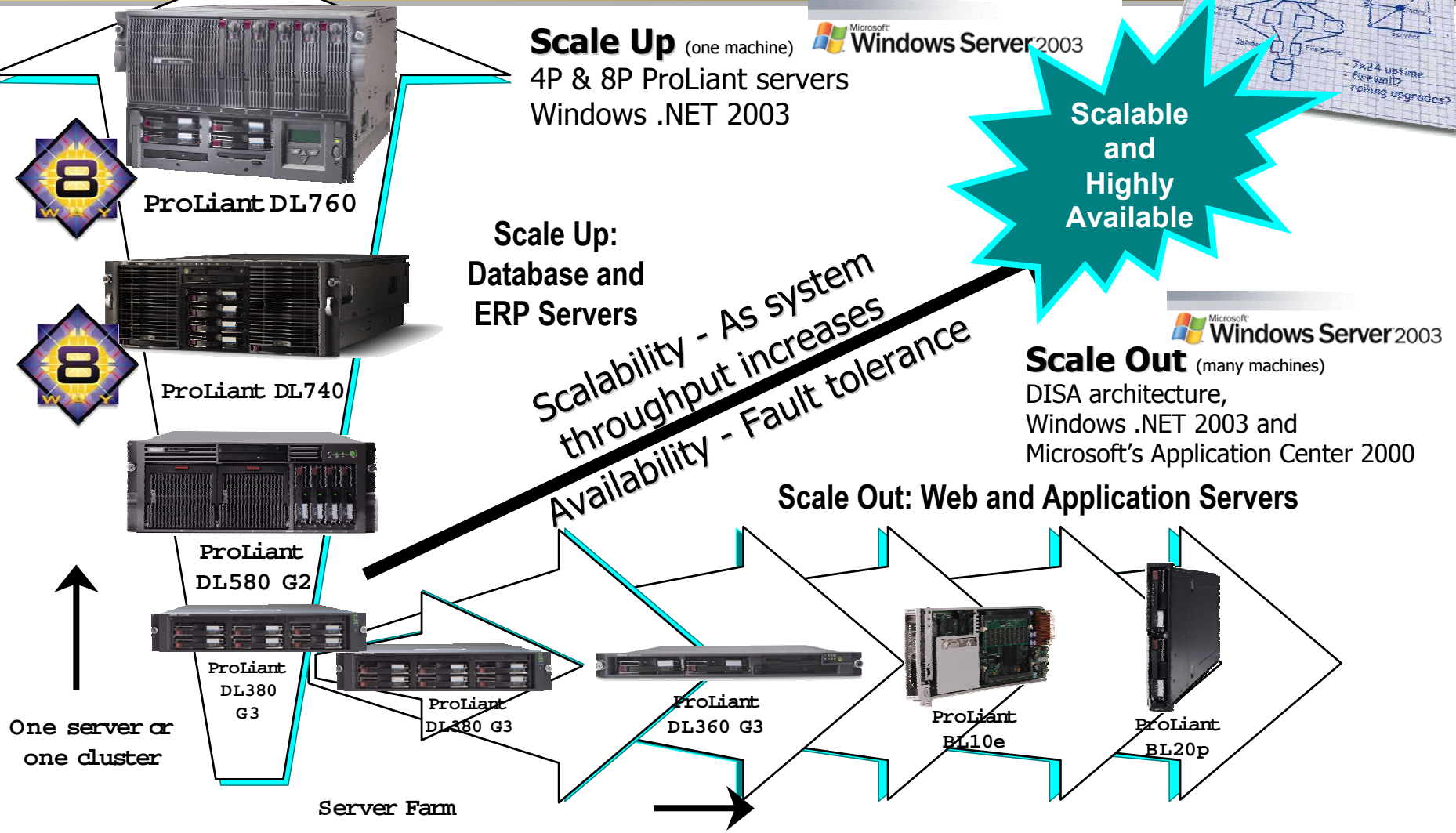
Security

Access & Acceleration Resources

Web & Application Resources

Data Resources

# A DISA Key Principle



# hp Dynamic Internet Solutions Architecture (DISA)



- Industry Standard Server Architecture for Internet Applications
- Uses Load Balancing to Provide Application Scalability and Availability
- Accommodates Different Functional Tiers within the Same Architecture
  - Access & Acceleration Resources
  - Web & Application Resources
  - Data Resources
- Each Formal Tier can be Further Segmented to Accommodate Specific Needs