

# **B336 Advanced Internet Computing**

# **Enterprise XML Environments**

# Learning Objectives

- Look at the key enterprise-level XML development environments today.
- Learn the advantages of using the tools in these environments, and the implications of using them.
- Get some ideas on what XML tools can be useful in an Enterprise environment.

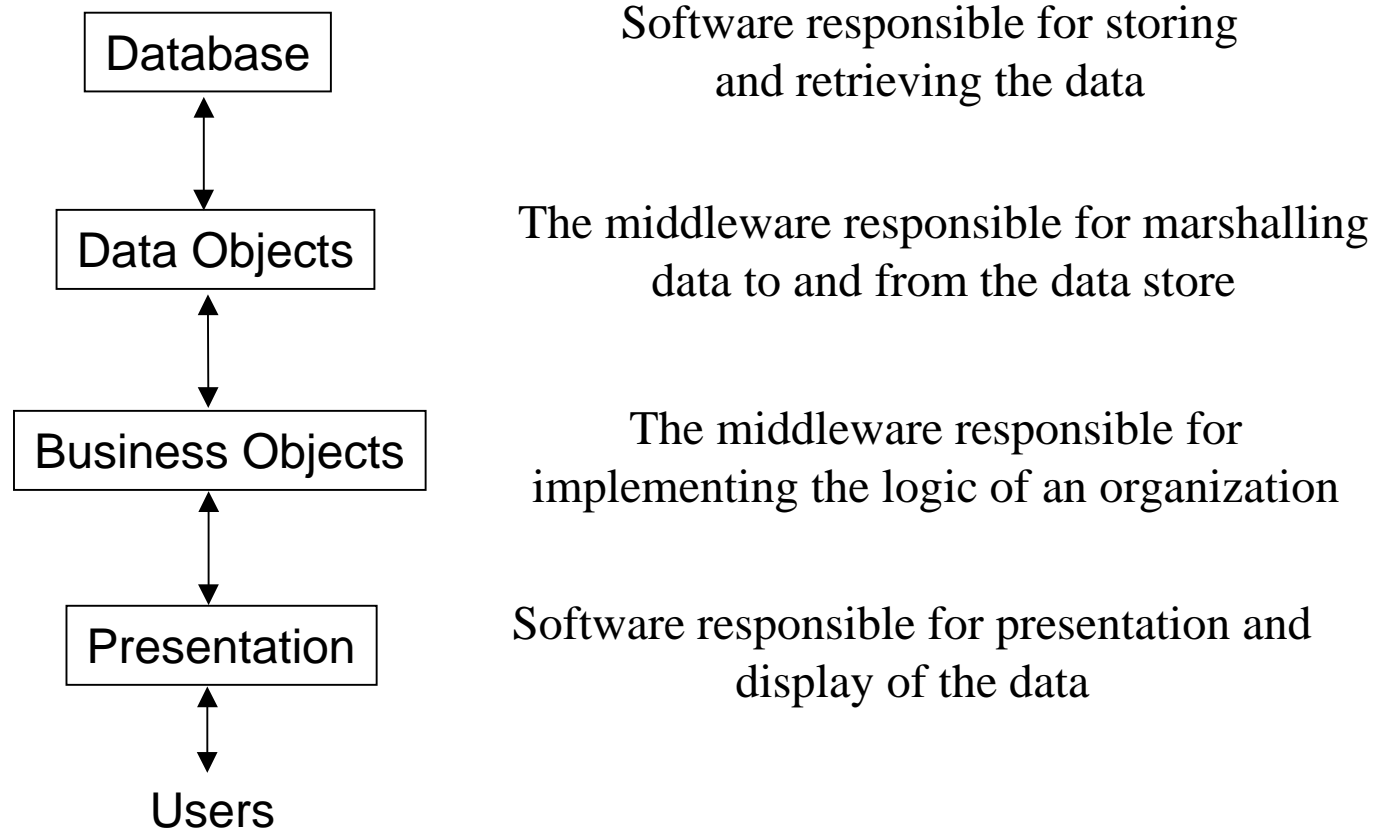
# Learning Objectives

- In the scheme of what we are doing in this unit:
  - We are studying how to use XML as an important set of Internet technologies to use as solutions in different areas (for example, the problem defined in Assignment 2).
  - It is important to know what kind of development environments are available to you to develop enterprise-level XML software and solutions in.

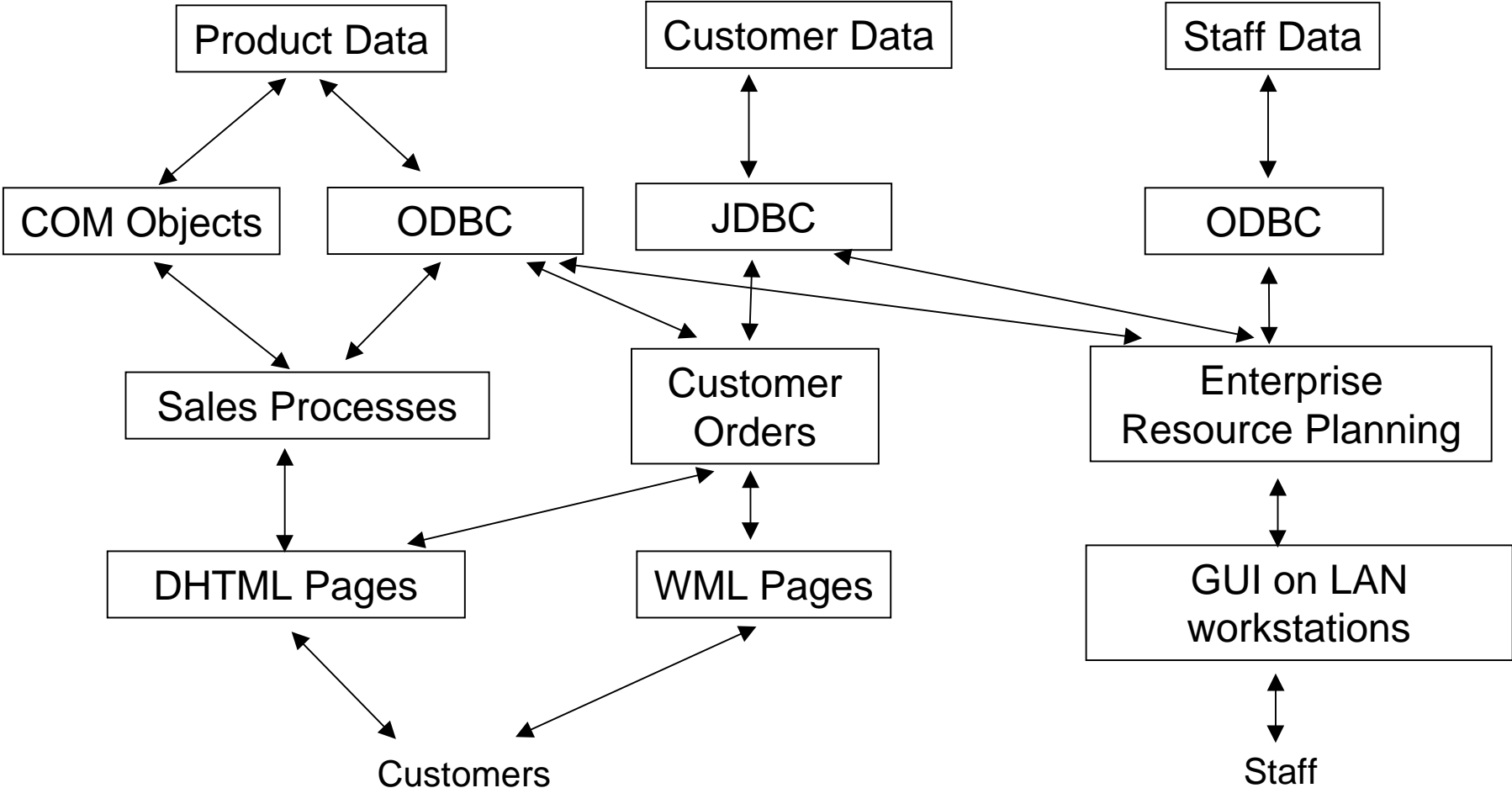
# Lecture Outline

- Information in enterprise systems
- XML and Databases
- Key XML Strategies
  - Microsoft's .NET
  - SUN's ONE
  - IBM, Oracle, Apache

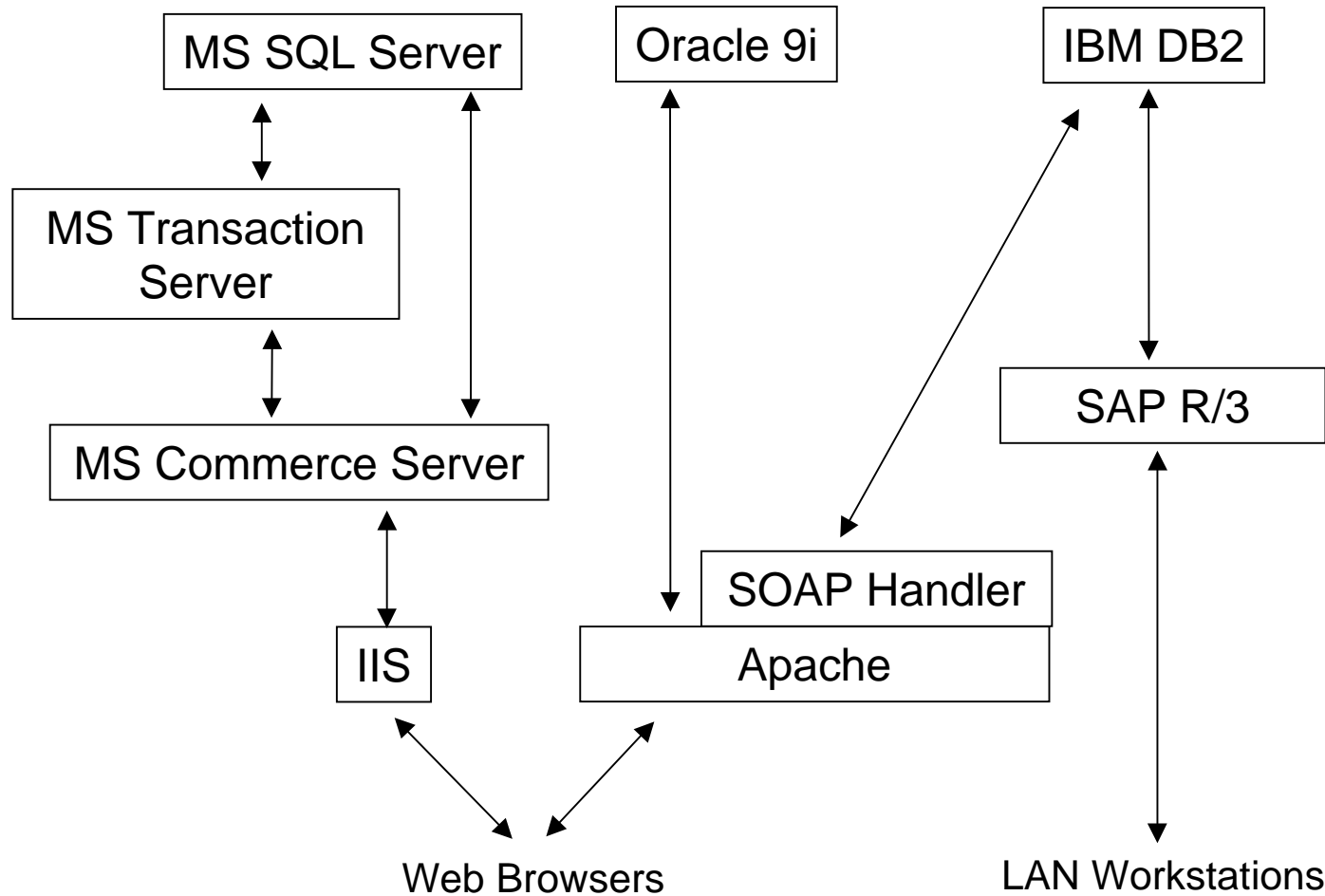
# An N-Tier Architecture for Most Enterprise Systems



# An Example N-Tier Architecture



# A Different Perspective: An Example “Physical” Architecture



The logical components from the last overhead exist in different boxes in this diagram, some overlapping in the same box.

# The Crying Game

- Why the tiers?
  - Why not a simply have database servers talking directly to users?
- Answer: to handle complexity
  - Advantages of an N-tier architecture:
    - » Easy to change and replace parts of the system without touching others (flexibility)
    - » Able to seamless grow the system (scalability)
    - » Reusability of middleware in different parts of the architecture, and different sections of the enterprise.
    - » Security by not allowing direct access to database.
  - For these reasons, some enterprises have even more tiers, to give more of the advantages above.

# Where does XML fit in?

- The XML technologies we have seen in this unit can be implemented and deployed in any one of the middle-tiers, and at the presentation layer.
  - Justifications for doing so are the same justifications for having XML software (see past lectures)
- XML at the presentation layer will depend on developments of display tools for the different XML applications.
  - See last lecture.

# XML as a Data Store

- What about the database?
  - Should we use XML as the format for storing the data as well?
- Some advantages of native database management systems (DBMS) as a data store:
  - Very efficient in retrieving and searching
  - Able to support very complex queries
  - DBMS servers can handle much higher data query/search loads than current servers (eg web servers) can serve XML documents.
  - Able to have different data types for different fields (as oppose to just plain text in XML)

# XML as a Data Store

- Some advantages of XML documents over native database formats as a data store:
  - Better storage fields which are large and complex. Eg.
    - » fields where dataset size is highly variable (pages in a book, comments about a subject)
    - » number of fields varies
    - » fields contained in another field - a comparison between the **table** format of relational databases, and the XML **tree** format. Note that this does not apply to hierarchical databases, which represents data in trees exactly like XML.
  - Platform independence
  - Stores very small data sets more efficiently
  - Expected long archive times
    - » XML as a public open standard less likely to require upgrade migration to new formats.

# XML as a Data Store

- Some DB management systems exist today which use XML as native storage formats.
  - Eg eXcelon ([www.exceloncorp.com](http://www.exceloncorp.com))
- Current database systems (relational, object-oriented, hierarchical, etc) will likely remain as the preferred choice for data storage
  - Especially in large enterprise systems
  - It is unlikely that XML storage and retrieval will reach the stage of efficiency and flexibility current database systems have.

# XML Support in Database Systems

- Due to XML's importance, most databases have now included support for XML.
- These can include
  - ability to read, and export XML documents
  - ability to generate DTDs based on relational tables
  - ability to process SQL queries on XML documents
    - » SQL, Structured Query Language, is a language use to define queries to relational databases - the most popular type of databases today.
- A look at two examples:
  - Oracle
  - Microsoft

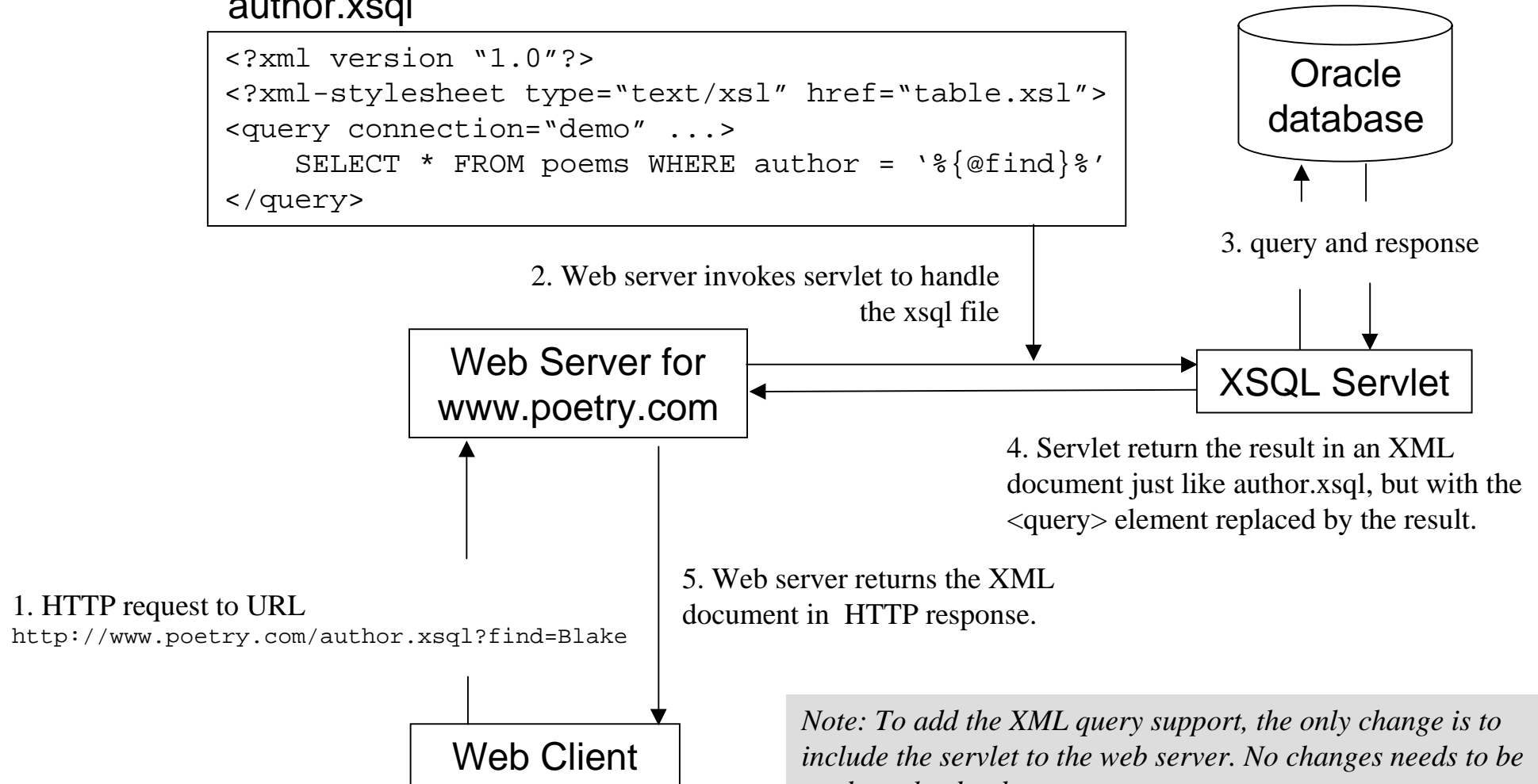
# XML Support from Oracle

- Oracle provides a number of tools in their XML Developer's Kit (XDK) for Java, C/C++, and PL/SQL:
  - XML Parser
  - Code generator
    - » generate classes from DTD
  - XML SQL Utility for Java
    - » generates an XML document in text or DOM from an SQL query
  - XSQL Servlet
    - » Takes an XML document containing an SQL query, and replace the query with a result.

# An Example XSQL Query

author.xsql

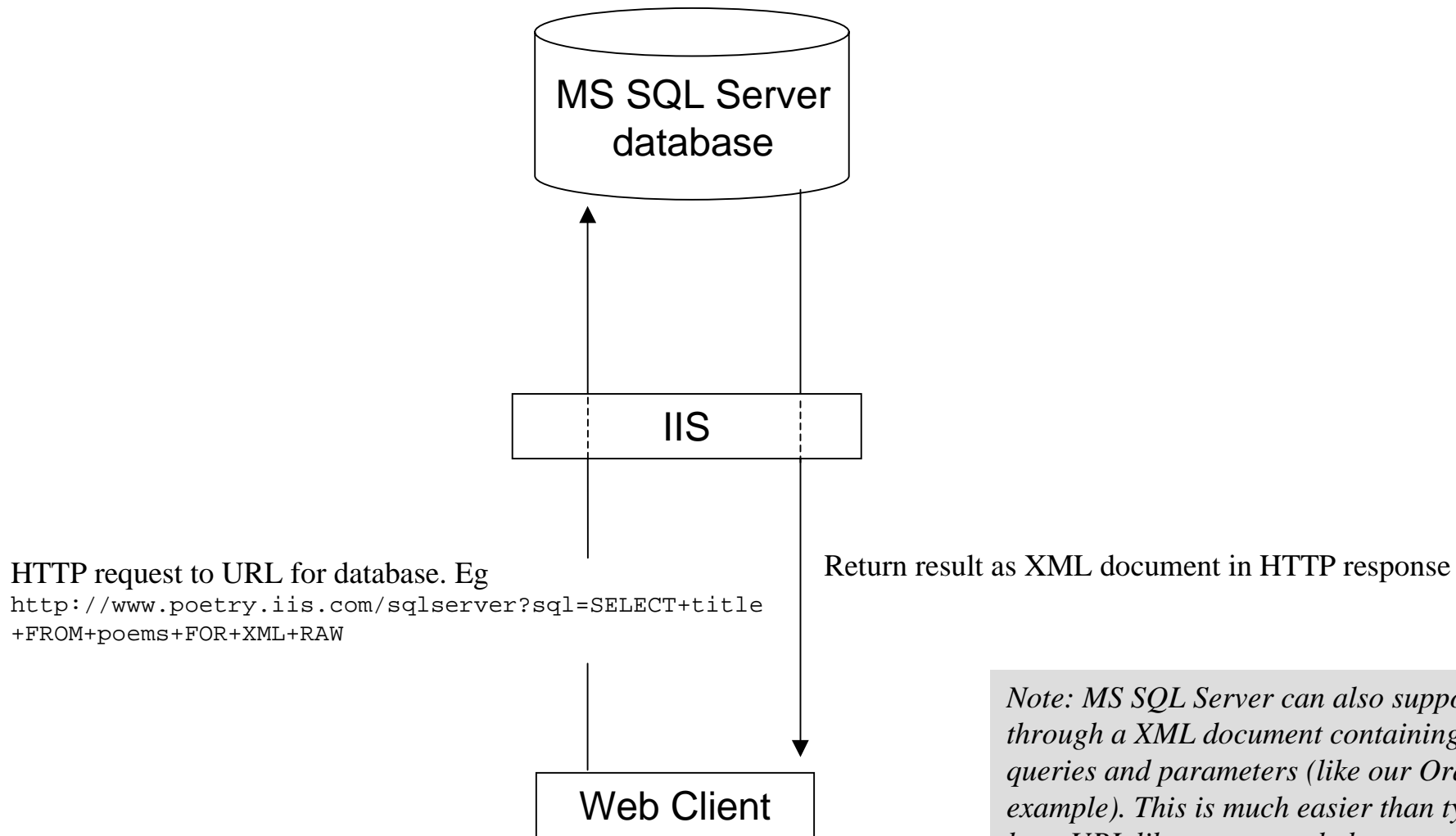
```
<?xml version "1.0"?>
<?xml-stylesheet type="text/xsl" href="table.xsl">
<query connection="demo" ...>
    SELECT * FROM poems WHERE author = '%{@find}%'
</query>
```



# Microsoft's XML Support for Databases

- XML database support from Microsoft
  - MSXML parser
    - » DOM interface with XPath
    - » SAX2 interface, beta version
    - » XSLT transformation (though not standard)
  - Visual Basic code generator
    - » read XML Schema and generate VB classes.
  - Built-in support in SQL Server
    - » Server can receive an SQL query through a HTTP request (filtered through IIS), and respond either in XML, or transform to HTML using XSL.

# Example SQL Query to MS SQL Server



*Note: MS SQL Server can also support queries through a XML document containing the queries and parameters (like our Oracle XSQL example). This is much easier than typing a long URL like our example here.*

# Microsoft's .NET Framework

- We can't speak of Microsoft's XML directions without talking about their .NET framework.
- .NET is a model targeted towards migrating Microsoft's focus on desktop-based software towards Internet-based software.
- Based on XML as the basis for data exchange.

# Components of .NET

- Most of the information on the framework is still only promotional material, as a catch-cry for the company's advertising.
- More and more products will be released in the future tied to .NET.

# Components of .NET

- The plans for different parts of the .NET initiative:
  - Tools for software developers
    - » Visual Studio .NET
    - » the .NET Framework (consisting of ADO, ASP, C#, etc)
    - » Microsoft Windows .NET (the operating system).
  - .NET Enterprise Servers
    - » where all the data will reside and be served from
  - Services.
    - » Eg. Passport .NET for user authentication
  - User experience on multiple devices
    - » especially wireless

# Current .NET Software

- Most of the .NET software are still in their early stages of release:
  - Visual Basic .NET
  - Visual Studio .NET Beta 1
  - BizTalk Server
  - SQL Server
  - etc
- All .NET products currently released are still MS-platform dependent.
  - The industry is waiting for future products that demonstrate the current claim that .NET will be an open **platform-independent** framework.

# The XML Factor

- The role of XML in the .NET strategy is as the communication format.
- The development for XML is towards:
  - Having XML API support in their Visual Studio suite of software IDEs.
  - Releasing (database, web, application) servers which can serve XML efficiently
  - Having their client-side products (eg the Windows OS, and the upcoming MS-Office XP) be able to import, export, and manipulate XML.
  - Developing services available over the Internet which uses XML as the communication mechanism, using their servers and products.

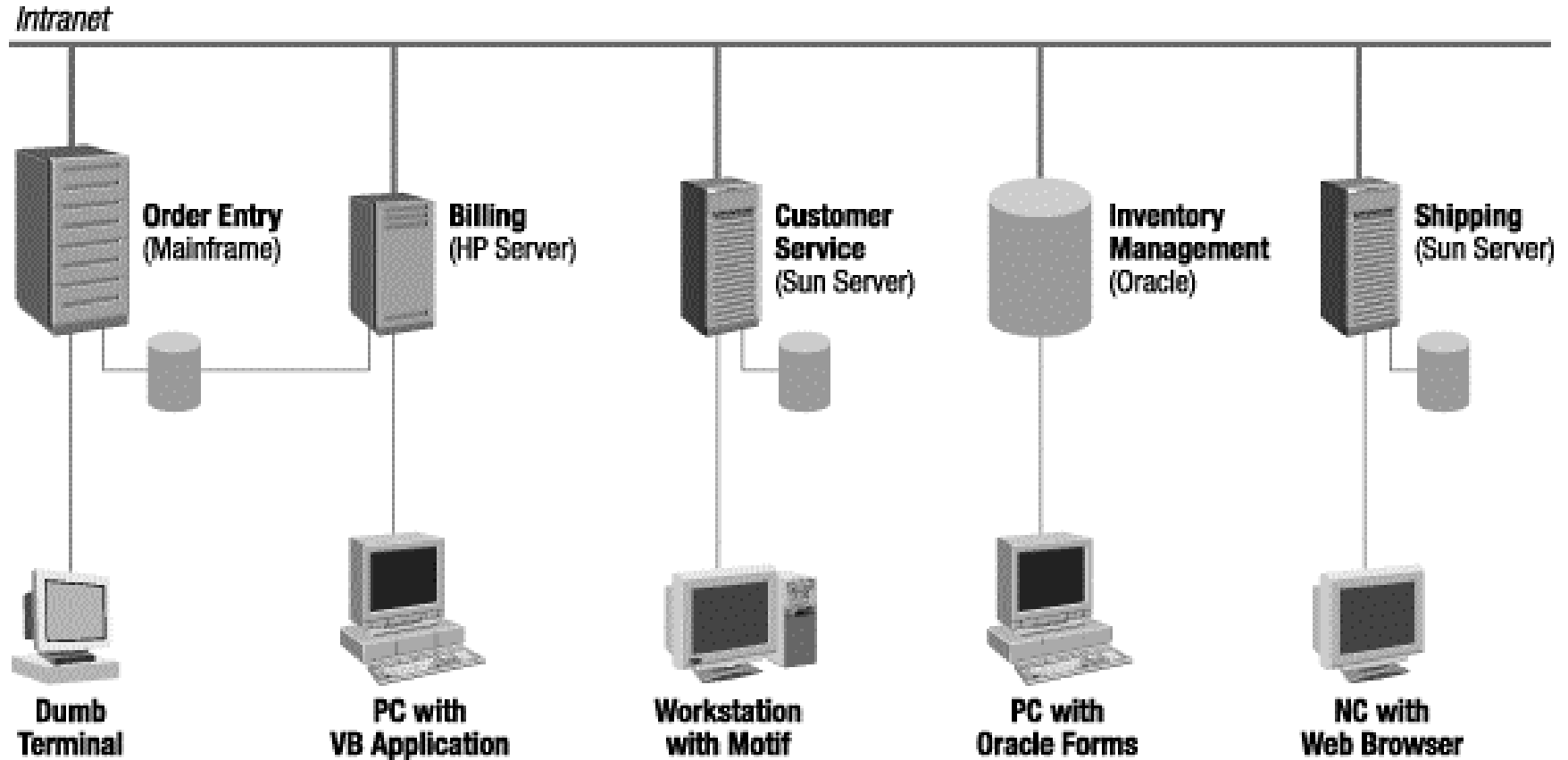
# XML at SUN Microsystems

- Emphasis on open standards.
  - Less developments tied to their Solaris OS and proprietary servers.
- Heavy promotion of developments tying XML with Java technologies
  - SUN has many of its products based on Java
  - Also, more customers moving to Java-based technologies means less customers using Microsoft-based products (which do not run on SUN hardware)

# Sun's ONE Integration

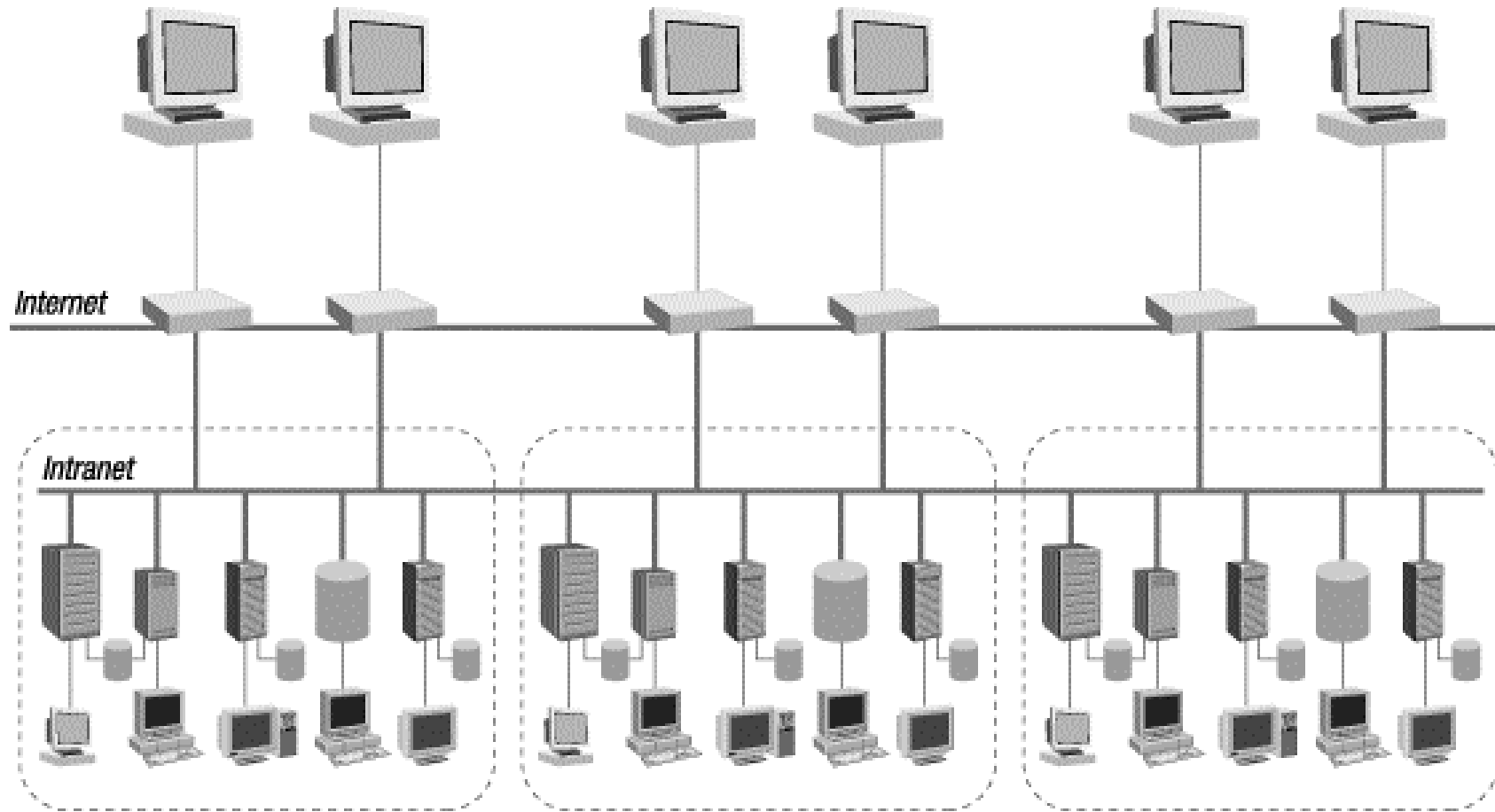
- Sun's future vision of web services in the future is based on its Open Net Environment (ONE) initiative.
- The ONE approach is based on communicating information using XML.
  - Application interfaces, implemented as either servlets or JavaServer Pages (JSP), contain code that manipulates XML messages and converts them into Java to give to back-end applications that actually implement the organizational logic.

# Current Environment



From <http://www.sun.com/dot-com/sunjournal/v5n1/feature2.html>

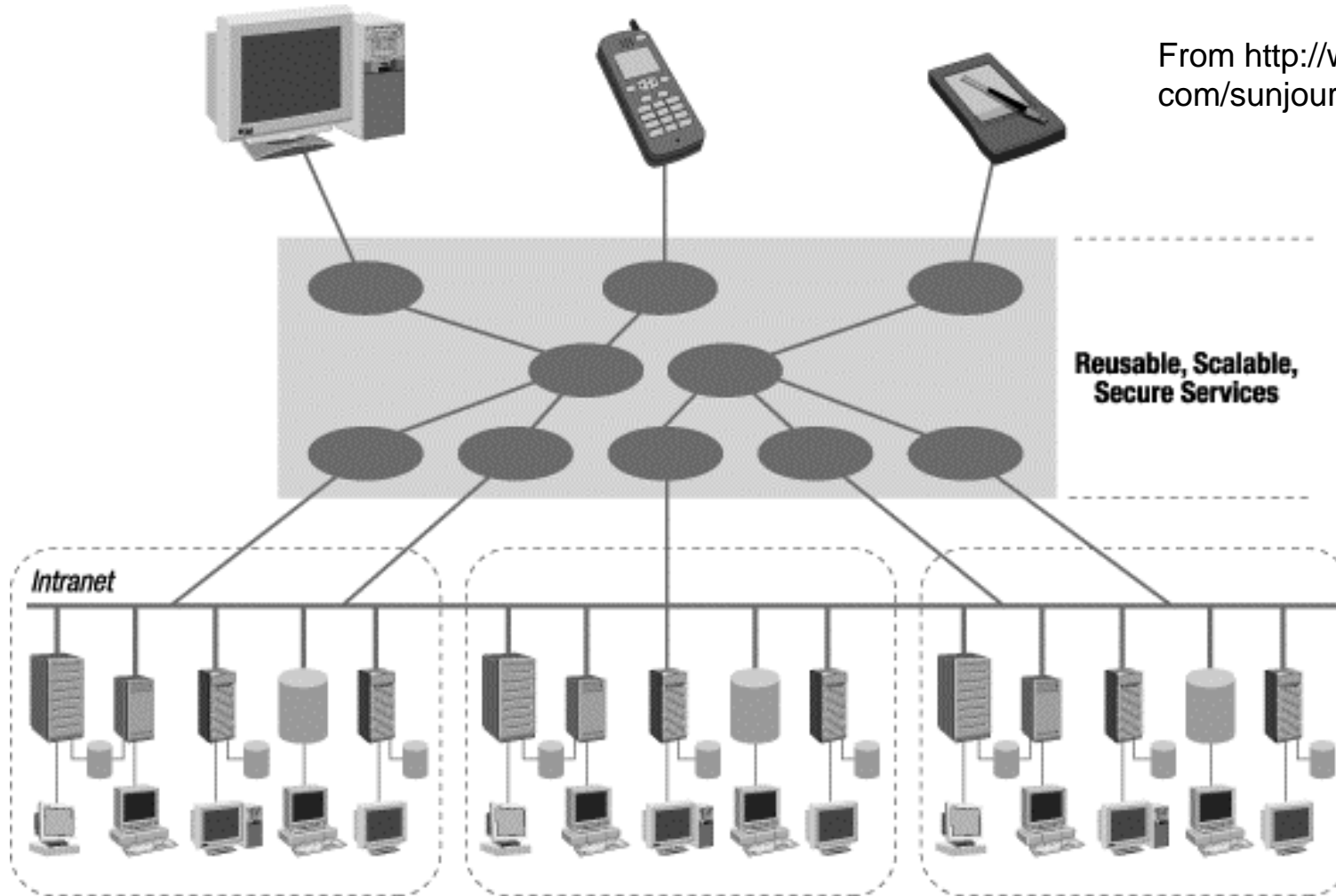
# Current Environment



From <http://www.sun.com/dot-com/sunjournal/v5n1/feature2.html>

# Sun's Vision of the Future

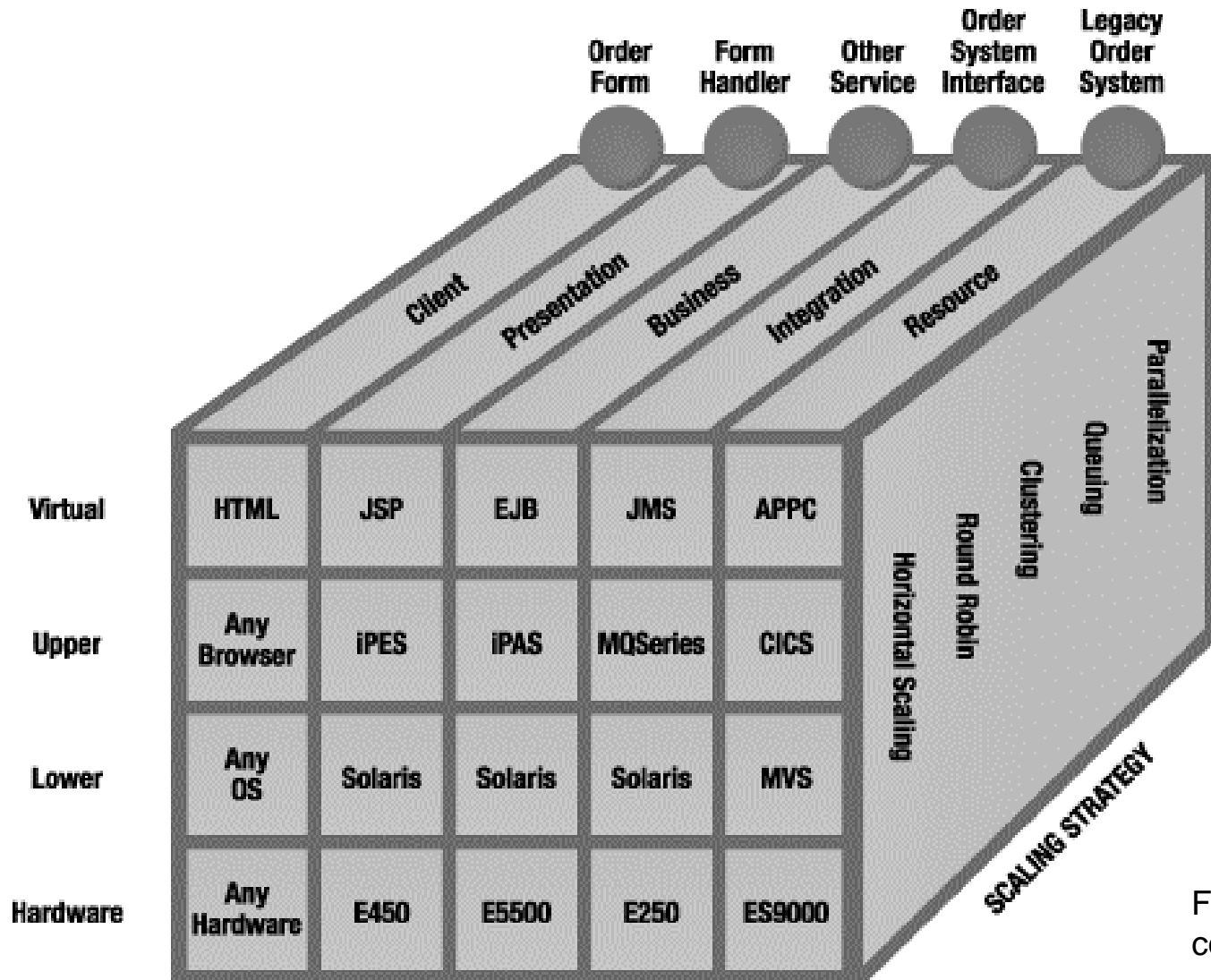
From <http://www.sun.com/dot-com/sunjournal/v5n1/feature2.html>



# Sun's Vision of the Future

- What they don't mention of course is that they want:
  - All the “reusable, scalable, secure services” to be Java-based technologies.
  - All the underlying servers to reside on Sun server hardware - their main source of income.

# Sun's 3-D Framework for web services



From <http://www.sun.com/dot-com/sunjournal/v5n1/feature2.html>

# Some Key XML Products from SUN

- **XML support in Java 2 - Java APIs for XML (JAX)**
  - » JAXP, for the reading, manipulating, and generating of XML documents by integrating an external XML parsers into a Java application.
  - » JAXM, for the packaging, routing and transport of XML-based messages across protocols such as those based on HTTP, SMTP, and FTP protocols.
  - » JAXB, for delivering and maintaining high-performance XML-enabled applications.
- **XSLT Compiler**
  - » A Java-based tool for compiling XSL style sheets into a lightweight and portable Java class for transforming XML documents.
- **Sun XML Datatypes Library**
  - » Sun's implementation of W3C's XML Schema Part 2 Datatypes.

# ONE vs .NET

- These are the two biggest initiatives in the industry today to lock Internet service and application developers into one environment.
- At the moment, Microsoft's .NET products are still at their early stages and are relatively immature, while SUN's J2EE (which ONE is based on) have been extensively used.
  - SUN have focussed on network application development a lot more than Microsoft has.
  - This will no doubt be challenged now that Microsoft changed its focus and picks up steam in this area.

# Apache XML Project

- A project to development XML tools and services for application development
  - Emphasizing open-source development
- The goals:
  - to provide commercial-quality standards-based XML solutions that are developed in an open and cooperative fashion,
  - to provide feedback to standards bodies (such as IETF and W3C) from an implementation perspective, and
  - to be a focus for XML-related activities within Apache projects

# Apache XML Project

- Currently 7 sub-projects:
  - **Xerces** - XML parsers in Java, C++ (with wrappers for full access in Perl and COM)
  - **Xalan** - XSLT stylesheet processors, in Java and C++
  - **Cocoon** - XML-based web publishing, in Java
  - **FOP** - XSL formatting objects, in Java
  - **Xang** - Rapid development of dynamic server pages, in JavaScript
  - **Apache SOAP** - Simple Object Access Protocol
  - **Batik** - A Java based toolkit for Scalable Vector Graphics (SVG)
  - **Crimson** - A Java XML parser derived from the Sun Project X Parser.

# IBM's XML Initiatives

- IBM's focus on XML is as a development environment for web services.
- **The IBM XML and Web Services Development Environment** contains tools to:
  - » Create/Transform - Use XML editing functions to develop new Web services.
  - » Build - Wrap existing components so that they can be exchanged using object-oriented distributed technologies (such as SOAP - more later)
  - » Deploy - Deploy the Web service on the developer's machine or to a remote, production-level server for testing right away.
  - » Test - Test applications as they run locally or remotely, and get instant feedback.

# IBM's XML Initiatives

- Future versions of WSDE also plans to include tools to:
  - » Discover - Browse the UDDI Business Registry (more later) to locate existing Web services for integration.
  - » Publish - In addition to creating and deploying Web services, the development environment can also publish them to the UDDI Business Registry.

# IBM's XML Software

- The principle tool released by IBM is their XML4J development kit (mentioned in previous lectures).
- Their direction is to also add support to their principal web server, WebSphere.

# Appendix: What is UDDI?

- Universal Description, Discovery and Integration.
- The UDDI specifications define a way to publish and discover information about Web services.
  - UDDI aims to automate the process of publishing your preferred way of doing business, finding trading partners and have them find you, and inter-operate with these trading partners over the Internet.

# What is UDDI?

- UDDI specifying a framework which will enable businesses to:
  - Discover each other (manually, or using software that automatically scans these registries)
  - Define how they interact over the Internet
  - Share information in a global registry that will more rapidly accelerate the global adoption of B2B eCommerce.

# Example Entries in a UDDI Registry

The screenshot shows a Microsoft Internet Explorer browser window titled "UDDI Browser - Microsoft Internet Explorer". The address bar contains "http://demo.alphaworks.ibm.com/browser/". The page content is titled "UDDI Service Details" and shows "Service details for: GetDirectionsService".

**Service Name** | **Description**

GetDirectionsService	
----------------------	--

**Binding Templates:**

Protocol Type	Access Point	Description
http	<a href="http://demo.alphaworks.ibm.com/soap/servlet/rpcrouter">http://demo.alphaworks.ibm.com/soap/servlet/rpcrouter</a>	
http	<a href="http://demohost:4040/soap/servlet/rpcrouter">http://demohost:4040/soap/servlet/rpcrouter</a>	
http	<a href="http://localhost:80/soap/servlet/rpcrouter">http://localhost:80/soap/servlet/rpcrouter</a>	
http	<a href="http://demo.alphaworks.ibm.com/soap/servlet/rpcrouter">http://demo.alphaworks.ibm.com/soap/servlet/rpcrouter</a>	
other	<a href="jsps/directionsDesign.jsp">jsps/directionsDesign.jsp</a>	
other	<a href="jsps/directions.jsp">jsps/directions.jsp</a>	

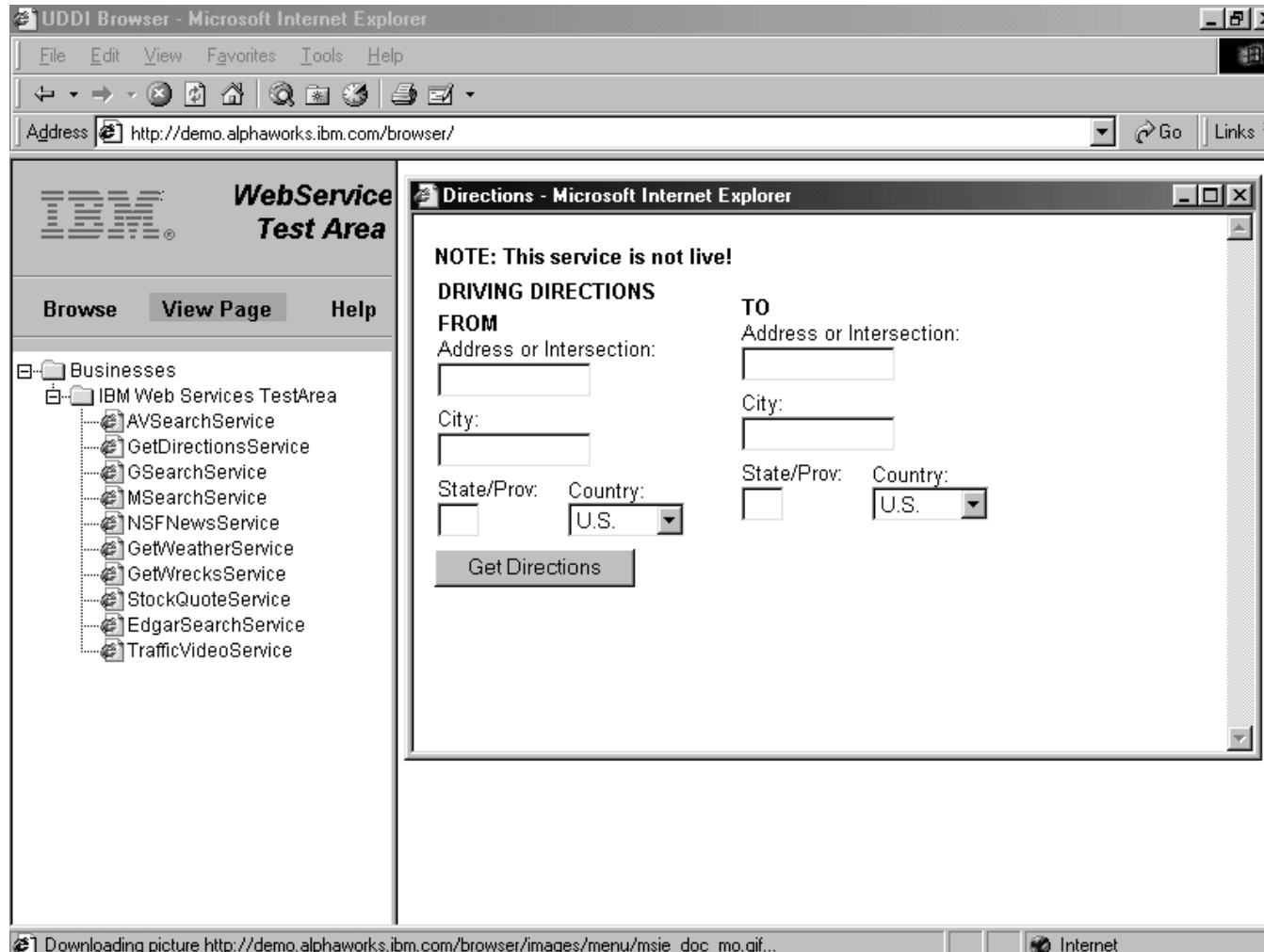
**WSDL Implementation Files**

The left sidebar shows a tree view of "Businesses" with "IBM Web Services TestArea" expanded, listing services like AVSearchService, GetDirectionsService, GSearchService, MSearchService, NSFNewsService, GetWeatherService, GetWrecksService, StockQuoteService, EdgarSearchService, and TrafficVideoService.

Opening page <http://demo.alphaworks.ibm.com/browser/displayService.jsp?sk=service2&bk=57BA1D60-102F-11D5-A4> Internet

From IBM's test registry  
<http://demo.alphaworks.ibm.com/browser/>

# Example Entries in a UDDI Registry



# Appendix: W3C's SOAP

- According to W3C's draft specifications:
  - SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment.
  - It is an XML based protocol that consists of three parts:
    - an envelope that defines a framework for describing what is in a message and how to process it,
    - a set of encoding rules for expressing instances of application-defined datatypes, and
    - a convention for representing remote procedure calls and responses.
- Different vendors (eg. Microsoft) have their own versions of SOAP, not all of them compatible.

# XML-SOAP Request using HTTP

```
POST /addpoem-request HTTP/1.1
Host: poetry.org
Content-Type: text/plain; charset="utf-8"
Content-Length: 507

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
  <SOAP-ENV:Body>
    <p:AddEntry xmlns:p="http://poetry.org/poetry-soap">
      <p:poem_headers>
        <title>Lament of the Exams</title>
        <author>Frank Ernest</author>
      </p:poem_headers>
      <p:poem_body>
        <!-- poem lines goes here -->
      </p:poem_body>
    </c:AddEntry>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

A request to  
update the data  
on the server

# XML-SOAP Response

HTTP/1.1 200 OK

Server: Apache/1.3.17 (Unix)

Date: Tue, 28 Feb 2001 04:23:03 GMT

Content-type: text/xml; charset="utf-8"

Content-length: 296

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" >
  <SOAP-ENV:Body>
    <p:AddEntryResponse xmlns:p=" http://poetry.org/poetry-soap "/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

# SOAP and other protocols

- SOAP is one of the possible transport protocols for XML messaging
  - We want a messaging transport system so that we can transport information a lot more efficiently and flexibly than using files - a lot of distributed object technologies is to satisfy this need.
  - Another popular method is using XML-RPC (remote procedure calls)
- SOAP messages can also be sent using other transports besides HTTP, such as SMTP (Internet e-mail).

# References

- Oracle and XML
  - » <http://otn.oracle.com/tech/xml/>
- Microsoft's .NET
  - » <http://msdn.microsoft.com/net/default.asp>
- XML at SUN
  - » <http://www.sun.com/xml/>
- Apache's XML Project
  - » <http://xml.apache.org/>
- IBM's Web Services Development Environment
  - » <http://www.alphaworks.ibm.com/aw.nsf/frame?ReadForm&/aw.nsf/techmain/751C2A483B9B2387852569920001AC96>
- XML and Databases (including a comprehensive list of current XML db products)
  - » <http://www.rpbouret.com/xml/XMLAndDatabases.htm>

# What have we learnt today?

- We will not be using any of the software I mentioned today in our labs exercises
  - And it is very likely that unless you dwell very deeply into XML, you will never see the software at all.
- In spite of that, it is important for you to get introduced to them:
  - As a briefing of current state-of-the-art in commercial uptake of XML.
  - As an indication of where further study of XML can lead you to.
  - As ideas to include in your solution design (eg in Assignment 2) using the basic ideas of XML, if appropriate.

# In this final overhead on XML ...

- Let me remind you that
  - I have introduced you to what I consider to be the key components of XML foundation, so you should be able to follow most reading material on XML, and further your own learning on the subject.
  - DO NOT ASSUME you know everything about XML
    - » This unit have only touched on the **beginnings** of **your first steps**.
    - » There is a whole world of wonders for you to discover!
  - Actually, while I'm at it... DO NOT ASSUME you know everything about **anything** in I.T.
    - » Even for me - I love my job and I deal with it every day, but every corner I turn I discover something new in areas which I thought I knew very well.