

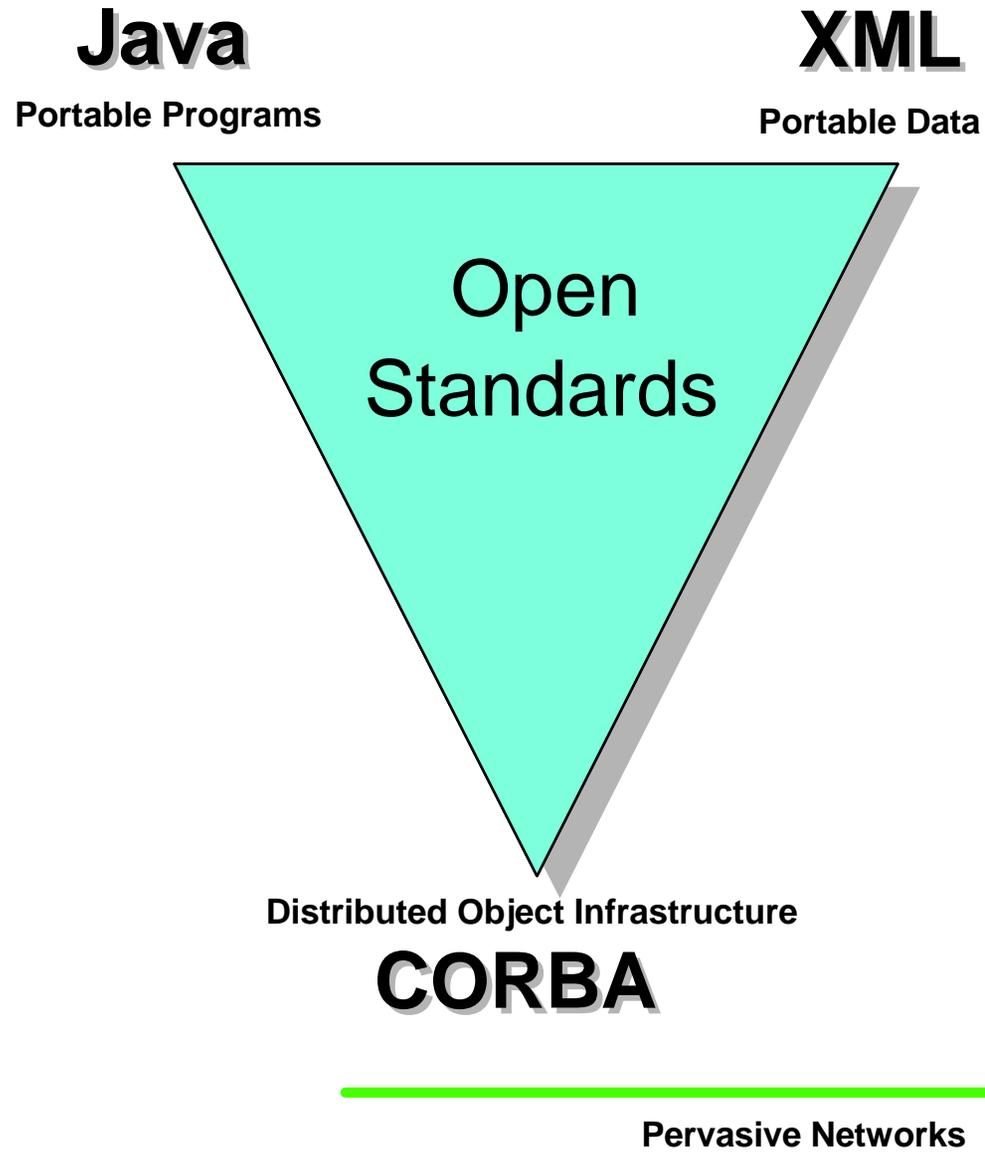
# **XML and Java: Lessons Learned in Building Application**

**Ted Leung  
Technical Lead, XML4J Parser**

# Agenda

- **Technology background**
  - ▶ XML
  - ▶ Java
- **Architectures**
- **Guidelines**
- **XML Tools**
- **Summary**

# Distributed OO Middlewar



# XML - eXtensible Markup Language

- XML was derived from SGML
  - ▶ 80% of function, 20% of complexity
- XML is key for e-business
  - ▶ standard way to share structured data
- XML and Java work well together
  - ▶ Java = portable code
  - ▶ XML = portable data
- XML says nothing about presentation

# XML example - Address Book

- Address Book Markup Language sample

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE addressBook SYSTEM "abml.dtd">
```

```
<addressBook>
```

```
  <person salary="26350.00" band="D">
```

```
    <name>
```

```
      <family>Wallace</family> <given>Bob</given>
```

```
    </name>
```

```
    <email>bwallace@megacorp.com</email>
```

```
  </person>
```

```
</addressBook>
```

# Document Type Description

## ■ DTD Example for Address Book

```
<?xml encoding="UTF-8"?>
<!ELEMENT addressBook (person)+>
<!ELEMENT person (name,email*)>

<!ATTLIST person salary CDATA #REQUIRED >
<!ATTLIST person band (A|B|C|D|E|F) #REQUIRED >
<!ATTLIST person active (true|false) "true"
#IMPLIED >

<!ELEMENT name (family, given)>
<!ELEMENT family (#PCDATA)>
<!ELEMENT given (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

# A Sampling of DTDs

- **Open Financial Exchange (OFX)**
- **Online Trading Protocol (OTP)**
- **Information and Content Exchange (ICE)**
- **XML Bookmark Exchange Language (XBEL)**
- **Channel Definition Format (CDF)**
- **XML Remote Procedure Call (XML-RPC)**
- **Wireless Markup Language (WML)**
- **Resource Description**
- **Bean Markup Language (BML)**
- **Translation Memory eXchange (TMX)**
- **Mathematical Markup Language (MathML)**
- **Scalable Vector Graphics (SVG)**
- **Astronomical Markup Language (AML)**
- **Biopolymer Markup Language (BIOML)**
- **Common Business Library (CBL)**
- **Extensible Logfile Format (XLF)**

# XSLT / XSL

- **XSL = eXtesnbile Stylesheet Language**
- **XSL is an XML to XML transformation system**
  - ▶ **There are two parts**
  - ▶ **XSLT is the tree transformation part of the language**
  - ▶ **Formatting Objects**
- **The transformation is declaratively specified in XML**
- **A big use of XSL is to convert XML to HTML**
- **Still in working draft stage**
  - ▶ **Being combined with XPointer**

# XSLT Example

```
<?xml version="1.0" encoding="US-ASCII"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0">
<xsl:template match="person">
  <html><body>
    <xsl:apply-templates/>
  </body></html>
</xsl:template>
<xsl:template match="name"> <!-- reverse given & family name -->
  <xsl:value-of select='given' />
  <xsl:text> </xsl:text>
  <xsl:value-of select='family' />
</xsl:template>
<xsl:template match="email">
  <a>
    <xsl:attribute name="href"> <!-- add an href attribute -->
      <xsl:text>mailto:</xsl:text>
      <xsl:apply-templates/>
    </xsl:attribute>
    <xsl:apply-templates/>
  </a>
</xsl:template>
</xsl:stylesheet>
```

# XSLT Result

```
<html>  
  <body>  
    Bob Wallace  
    <a href="mailto:bwallace@megacorp.com">  
      bwallace@megacorp.com  
    </a>  
  </body>  
</html>
```

# XML Schema

- **A richer language for constraining XML content**
- **Syntax is regular XML syntax, not DTD syntax**
- **Support for data typing, inheritance, namespaces**
- **Still in early Working Draft form**

# XML Schema Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE schema PUBLIC "-//W3C/DTD XML Schema Version 1.0//EN"
"http://www.w3.org/1999/05/06-xmlschema-1/structures.dtd">
<schema>
  <elementType name="addressBook">
    <elementTypeRef name="person" minOccurs="1"></elementTypeRef></elementType>
  <elementType name="person">
    <sequence occurs="1">
      <elementTypeRef name="name" minOccurs="1" maxOccurs="1"></elementTypeRef>
      <elementTypeRef name="email" minOccurs="0"></elementTypeRef></sequence>
      <attrDecl name="band" required="true">
        <datatypeRef name="NMTOKEN">
          <enumeration>
            <literal>A</literal>
            <literal>B</literal>
            <literal>C</literal></enumeration></datatypeRef></attrDecl>
        <attrDecl name="active">
          <datatypeRef name="NMTOKEN">
            <enumeration>
              <literal>>true</literal>
              <literal>>false</literal></enumeration>
            <default>true</default></datatypeRef></attrDecl></elementType>
      <elementType name="name">
        <sequence occurs="1">
          <elementTypeRef name="family" minOccurs="1" maxOccurs="1"></elementTypeRef>
          <elementTypeRef name="given" minOccurs="1"
            maxOccurs="1"></elementTypeRef></sequence></elementType>
        </elementType></schema>
```

# Must I use Java to use XML?

- **NO!!!**
- **While many of the best programming tools for XML are currently Java-based, XML is completely language neutral**
- **XML is about system-to-system interaction and component-to-component collaboration, regardless of the underlying programming technology**

# Servlets

**Servlet is a Java class that can be used to dynamically extend your server's function. Servlets, as defined by Sun, are:**

"... objects which conform to a specific interface that can be plugged into a Java-based server. Servlets are similar to applets in that they are object byte codes that can be dynamically loaded off the net....  
They serve as platform independent, dynamically loadable, pluggable helper byte-code objects on the server side...."

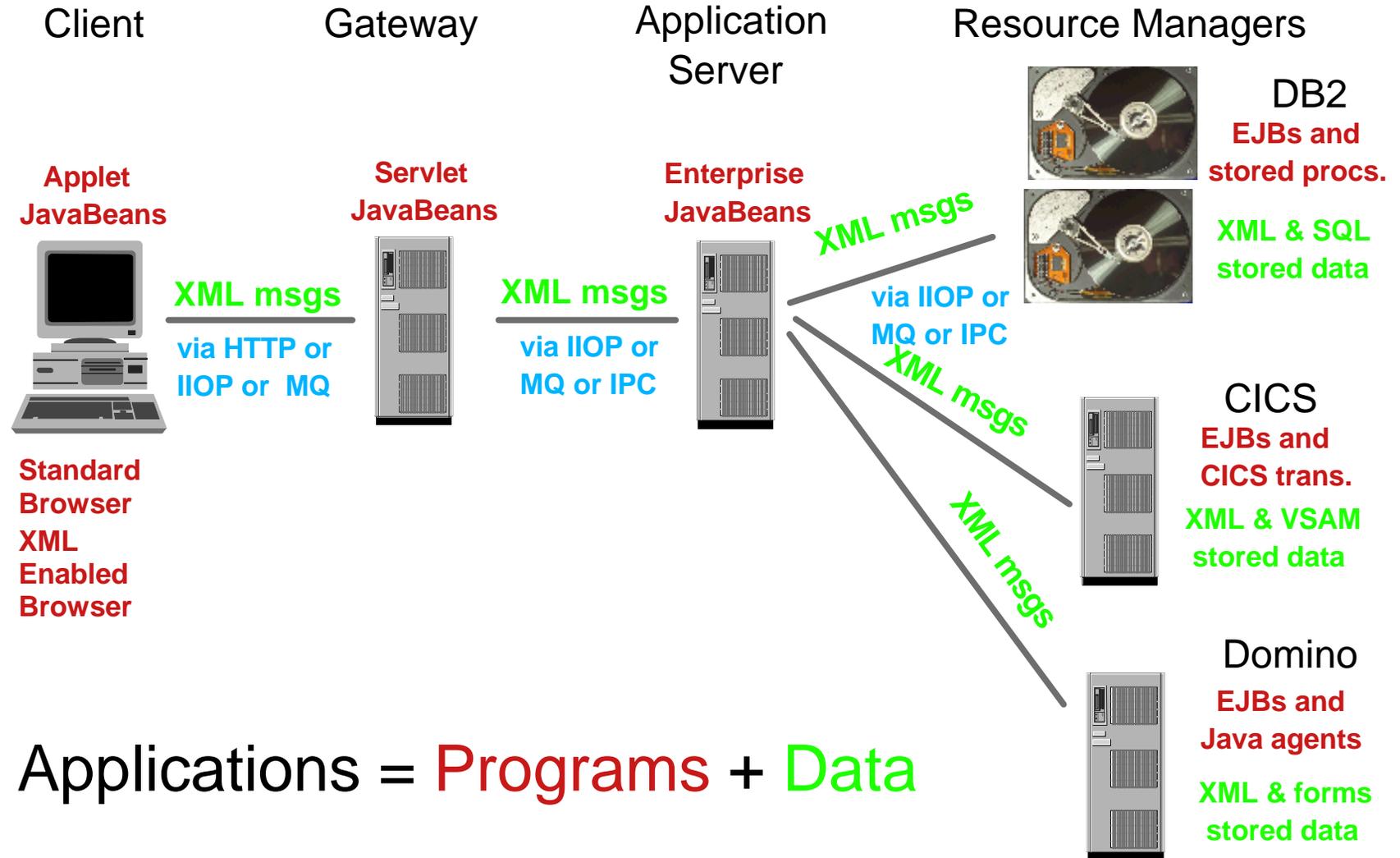
**In short, a servlet is to the server what the applet is to the client browser.**

**Servlets have a standard interface, which is defined in the package `javax.servlet`.**

# What XML and Java are Not

- **A silver bullet**
  - ▶ still have to design, code, and test
- **Guaranteed communication**
  - ▶ agreement between vendors and users is still required
- **Not an Object-Oriented Modeling Language**
  - ▶ use UML/XMI for that
- **Not middleware**
  - ▶ used to develop robust middleware
- **A replacement for HTML**

# Java + XML - The Winning Team



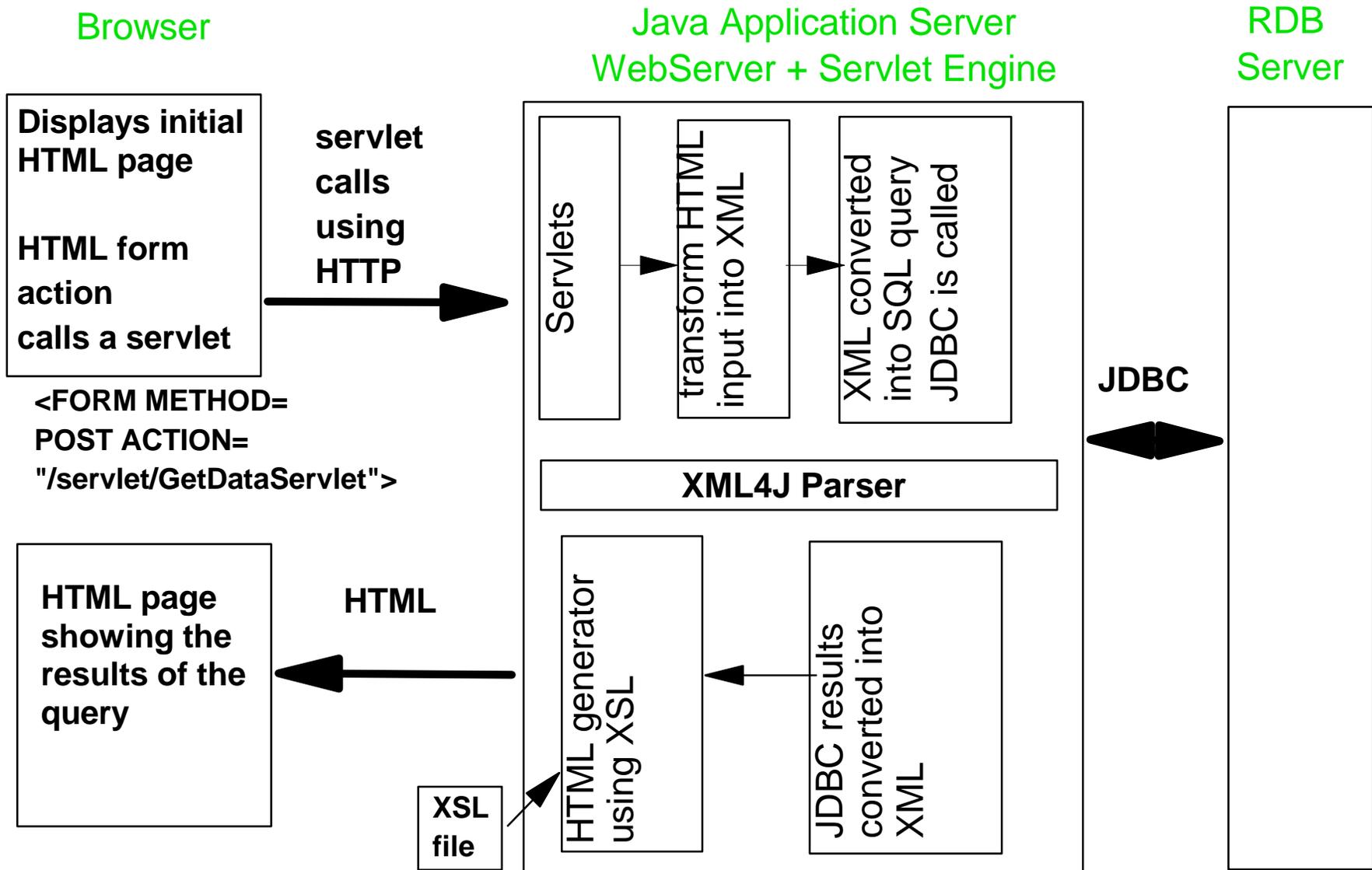
# Architectures using Servlets and XML

- We will discuss some architectures which combine use of servlet and XML.
- The application server which the servlet interacts with could be an EJB server; a database with or without Java stored procedures; MQSeries; CICS;etc.
- We will present design considerations based on work we have done with customers when designing servlets and XML.
  - ▶ special focus on supporting business objects

# HTML/XML/DB Architecture

HTML -> servlet -> XML -> Query (SQL) -> DB

HTML <- XSL <- XML <- JDBC result set <- DB

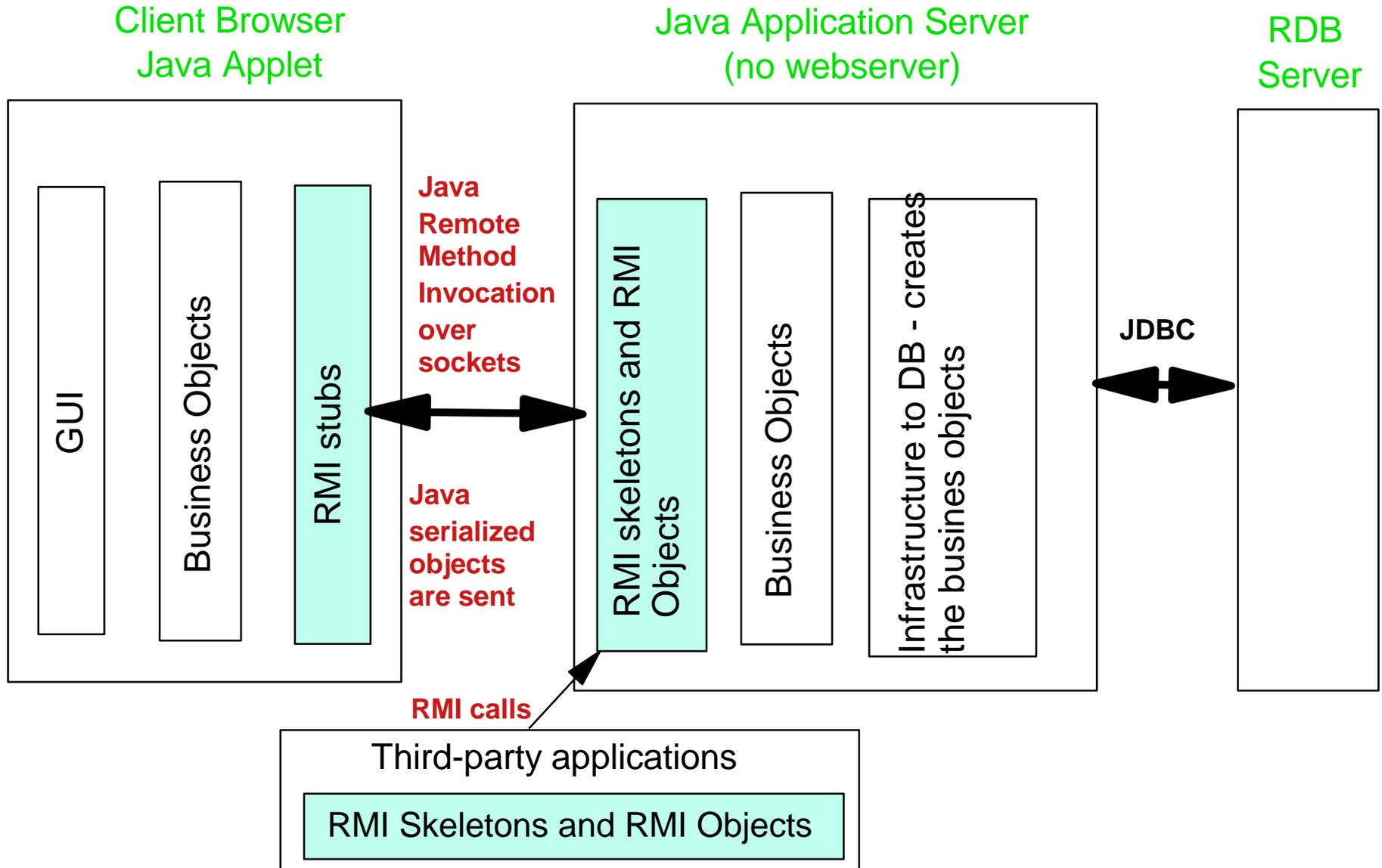


# An Example Application Problem

- **Client Requirements**
  - ▶ support an applet client for Intranet users
  - ▶ support a thin HTML client for Internet users
  - ▶ support interfaces to third-party applications
- **Complex business logic**
  - ▶ business objects to be used on server and in applet
- **Security**
  - ▶ authentication and single logon
  - ▶ SSL encryption
- **Scalability using multiple servers and tiers**
- **Server to access data in a relational database**
  - ▶ Note: other types of backends could be use instead of RDB

# Applet to Java Server using RMI (no XML)

Applet <-> RMI <-> servlet



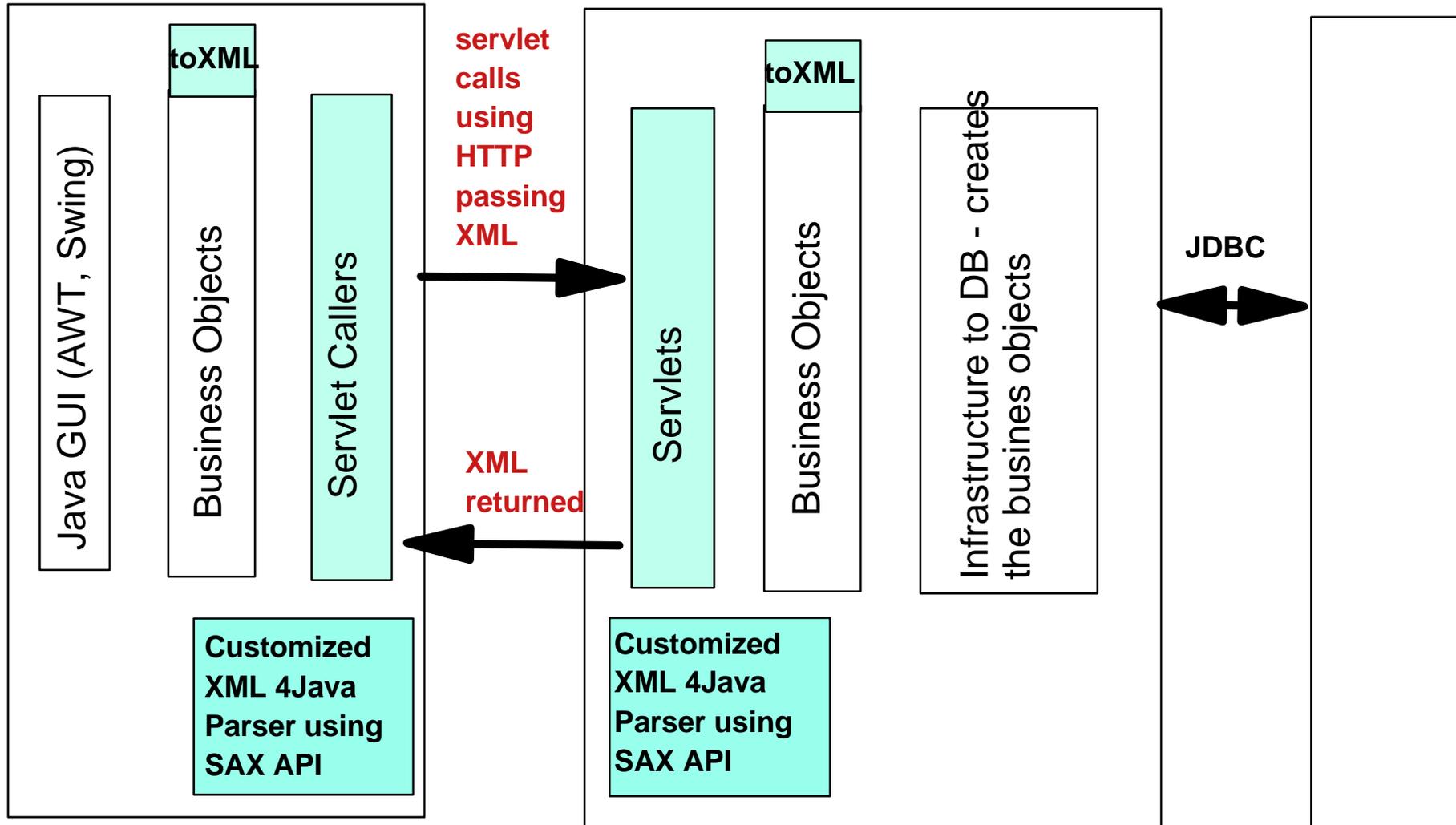
# Applet to Servlet Architecture using XML

applet <-> bus objects <-> XML <-> servlet <-> bus objects

Client Browser  
Java Applet

Java Application Server  
WebServer + Servlet Engine

RDB  
Server



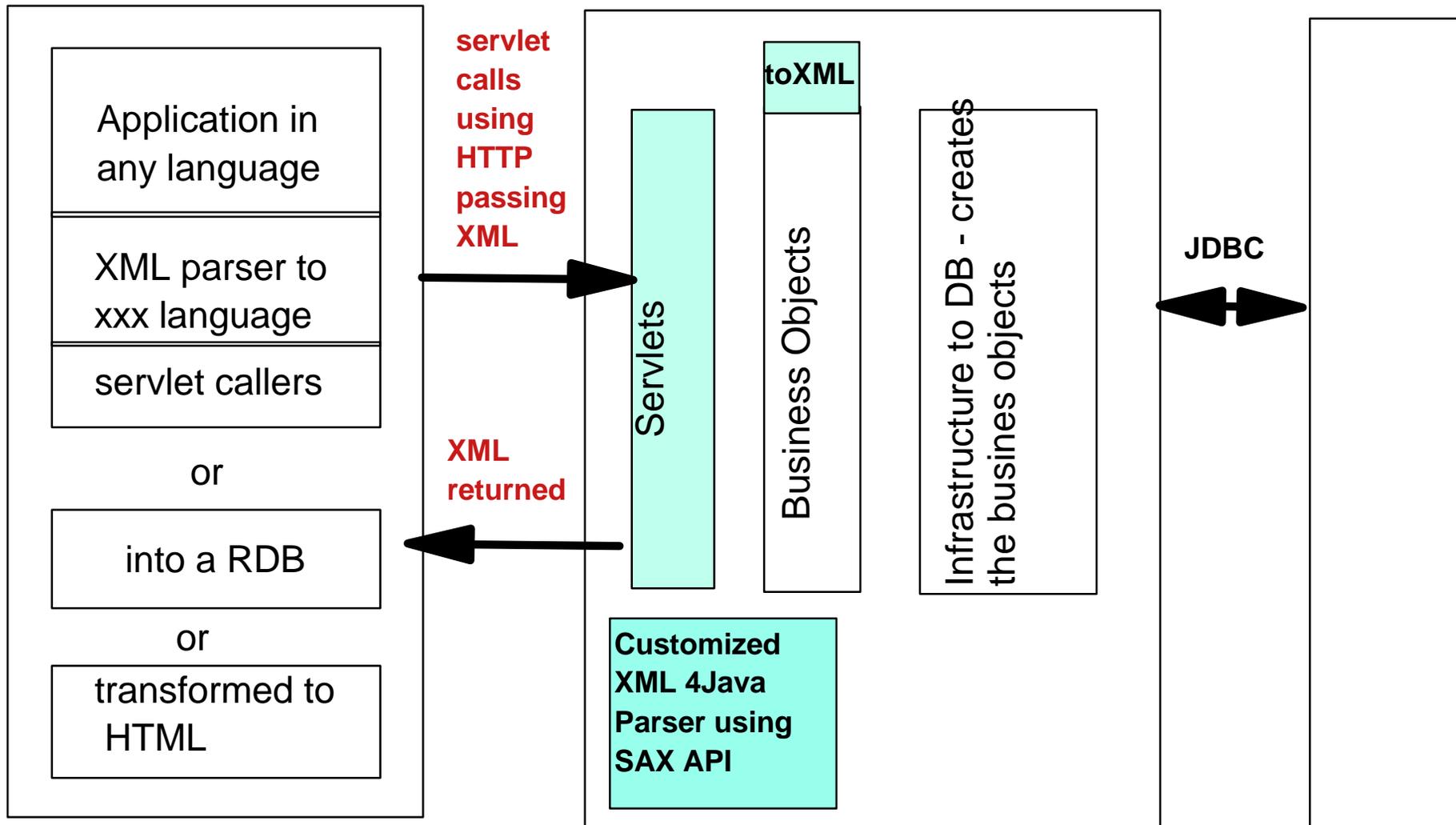
# Third-party to Servlet Architecture using XML

Third-party application <-> XML <-> servlet <-> bus objects

Third-Party Application

Java Application Server  
WebServer + Servlet Engine

RDB Server



# Potential Benefits using XML and Servlets Together

- **Leverage webserver technology for provide security**
  - ▶ **single logon**
  - ▶ **servlet security**
  - ▶ **SSL is very simple**
- **Highly scaleable using webserver and load balancing technology**
- **Integration into a large website is simplified**
- **Provide distinct, flexible interface the application servers**
  - ▶ **between application components**
  - ▶ **for 3rd party**
- **Support multiple front-ends**
  - ▶ **transform results from XML to HTML using XSL**
  - ▶ **3rd applications written in any language can call the servlets and receive XML**
  - ▶ **queuing of messages is easily supported**
  - ▶ **XML rendered directly in the browser**

# Design Choices - XML

- **Non-technical considerations**
  - ▶ cooperation in industry (suppliers, consumers)
  - ▶ competitive advantage
  - ▶ dominant players may dictate
- **When to use DTD validation**
  - ▶ better performance without validation
  - ▶ turn off when system is closed, turn on for third-party use
- **XML type definition is limited to Strings**
  - ▶ modeling work is required to map objects or tables
- **Consider XML namespace issues**
- **To transform business objects to XML requires custom code**
- **Storing XML into a RDB**
  - ▶ map from flat text structure to relational tables
  - ▶ map from business object to relational tables
  - ▶ store XML into text extenders
  - ▶ more direct support for XML is coming

# Design Choices - DOM vs SAX

## wrt XML <-> Business Objects

Both DOM and SAX being widely supported and standardized

### DOM

- Creates a Document Object Model tree.
- DOM tree which must be reparsed and converted into business objects. DOM tree is not really used.
- Need to subclass business objects from DOM superclass
- Subclassing DOM parser is more complex code.

### SAX

- API oriented mechanism which is triggered by XML tags
- Calls handler when XML tag is read
- Generates events without the DOM tree
- Code is straightforward

# Design Choices - DTD Specification

- Designed to support generalization
- Simple instance data represented as attributes instead of sub-elements
  - ▶ performance tradeoff vs richness of information
- Inheritance
  - ▶ XML's flat structure does not provide any inherent support for inheritance
  - ▶ superclass attributes included in the DTD for every concrete subclass
- Relationships represented as generalized 'Association elements'

```
<!ELEMENT Association EMPTY >
```

```
<!ATTLIST Association
```

```
    name          CDATA    #REQUIRED
```

```
    multiplicity  (1..1|1..n|n..m) #REQUIRED
```

```
    oids          CDATA    #REQUIRED
```

```
>
```

- ▶ No contained objects are defined in the DTD
- ▶ Linked to lazy initialization of relationships

# Design Choices - Servlets

- **Calls to servlets serve as API calls into the Java Server**
  - ▶ **'command' and parameters can be included as a parameter of the servlet**
    - ex. `VehicleServlet?command=retrieve&oid=1234567`
    - basic validation of commands
  - ▶ **represent the APIs in the XML which is passed to the servlet**
- **Servlet design**
  - ▶ **Single command processor servlet**
  - ▶ **Seperate servlet for each set of related commands such as CRUD for a business object**
  - ▶ **Seperate servlet for each command**
- **Handle required parameter information versus optional information**
  - ▶ **API should be 'forgiving'**
    - i.e. able to handle unexpected information and lack of optional information

# Guidelines for Development Process

- **Get help/input from business people**
- **Model the business objects and DTD separately**
  - ▶ **want your DTD to be application independent**
  - ▶ **DTD should be designed independently of a data or object model**
    - **but with consideration of the databases and applications to be supported**
    - **i.e. start from a purist view and compromise**
  - ▶ **map between the business objects, database, and XML as needed**
- **Validate DTD with business people and third-parties**
  - ▶ **keep it simple where possible**
- **prototype, prototype, prototype**
  - ▶ **measure performance**
- **search for existing solutions and tools**
  - ▶ **XML and Java tool space moves quickly**

# **XML Architectures**

## **(future is coming quickly)**

- **XML directly in and out of DB2**
  - ▶ **HTML -> servlet -> XML -> DB2**
  - ▶ **DB2 -> XML -> XSL processing servlet -> HTML**
- **XML transformed by MQ**
- **XML directly rendered in a browser**
  - ▶ **using XSL**
- **XML used with MQ Integrator to communicate between business or systems**
- **XML with Java Messaging Service (JMS)**
- **XML provides configuration information for systems**

# XML Tools

- **Changing drastically / quickly**
- **Standards are moving quickly, tools keeping up**
- **Java is preferred development language**
  - ▶ **Unicode support**
- **Tool Categories**
  - ▶ **XML Parsers**
  - ▶ **Database support**
  - ▶ **Messaging product support**
  - ▶ **XML Editors**
  - ▶ **XSL Processors**
  - ▶ **Java class libraries**
    - **digital signature**
    - **visual beans**

# IBM XML Tools

- Available from Alphaworks ([www.alphaworks.ibm.com](http://www.alphaworks.ibm.com))
  - ▶ XML Parser for Java (XML4J)
  - ▶ Lotus XSL
  - ▶ XML Productivity Kit for Java
  - ▶ XML Security Suite
  - ▶ Xeena
  - ▶ Bean Markup Language
  - ▶ P3P
  - ▶ XML Diff and Merge, XML TreeDiff
  - ▶ XML Parser for C++
- WebSphere
- MQ Series
- DB2

# External IBM XML Website

**IBM** [www.ibm.com/XML](http://www.ibm.com/XML)  
YOUR RESOURCE FOR XML TOOLS, NEWS, AND EDUCATION

**Welcome Developers!**  
the universal data format for e-business

**SITE HIGHLIGHTS**

- [XML Advances at OMG](#)  
1-14-99...XML was unanimously adopted by OMG's OAD task force - final approval vote in March.
- [XML & Java: The Perfect Pair](#)  
Take a close look at XML with a strict focus on code and tools written in Java and see why XML and Java are such a perfect match.
- [IBM XML and SGML Services](#)  
Ride the XML and e-business wave with help from the IBM Enterprise Document Management Solutions and Services team.

**LEARN ABOUT XML**

- [XML Tutorials for Programmers](#)  
Learn about XML technology and the key benefits of XML for business and programming. Check out the newest tutorial: [Writing XML documents.](#)
- [Learn to build an XML app!](#)  
This paper has been updated to include three steps. Learn how to write a DTD, generate XML from a data store, and then convert the XML into HTML.
- [Get the facts!](#)  
Check out some FAQs, including the W3C's classic XML FAQ, the first resource for those who want to learn XML.

**XML Parser for Java**  
100% Pure Java  
12.03.98 - Download V1.1.9 today!  
DOM 1.0

**PRODUCTS & TECHNOLOGIES**

- [WebSphere](#)  
Build and manage high-performance Web sites.
- [jCentral™](#)  
Search the Web for Java.™
- [Technology Previews](#)  
Download previews for XML from IBM's alphaWorks.

[www.alphaworks.ibm.com](http://www.alphaworks.ibm.com)  
**alphaWorks**  
IBM  
e-business

IBM Home Shop Contact IBM Search Privacy Legal

©1999 IBM Corporation

**DEMO**

# Resources

- **Useful Sites**

- ▶ [www.ibm.com/xml](http://www.ibm.com/xml)
- ▶ [www.ibm.com/java](http://www.ibm.com/java)
- ▶ [www.alphaworks.ibm.com](http://www.alphaworks.ibm.com)
- ▶ [www.xml.com](http://www.xml.com)
- ▶ [www.xml.org](http://www.xml.org)

- **Contacts**

- ▶ [xml4j@us.ibm.com](mailto:xml4j@us.ibm.com)