



IBM IT Education Services

E11

Wilhelm Mild

IBM Boeblingen Laboratory

VSEESA@de.ibm.com

**Web Services, the modern
Technology of future computing**

VSE Technical Conference

November 10 - 12, 2003 | Hilton, Las Vegas, NV

© 2003 IBM Corporation



Web services

What are Web Services?

Web Services Architecture

Web Services Protocols

What is the value of Web Services?



IBM @server. For the next generation of e-business.

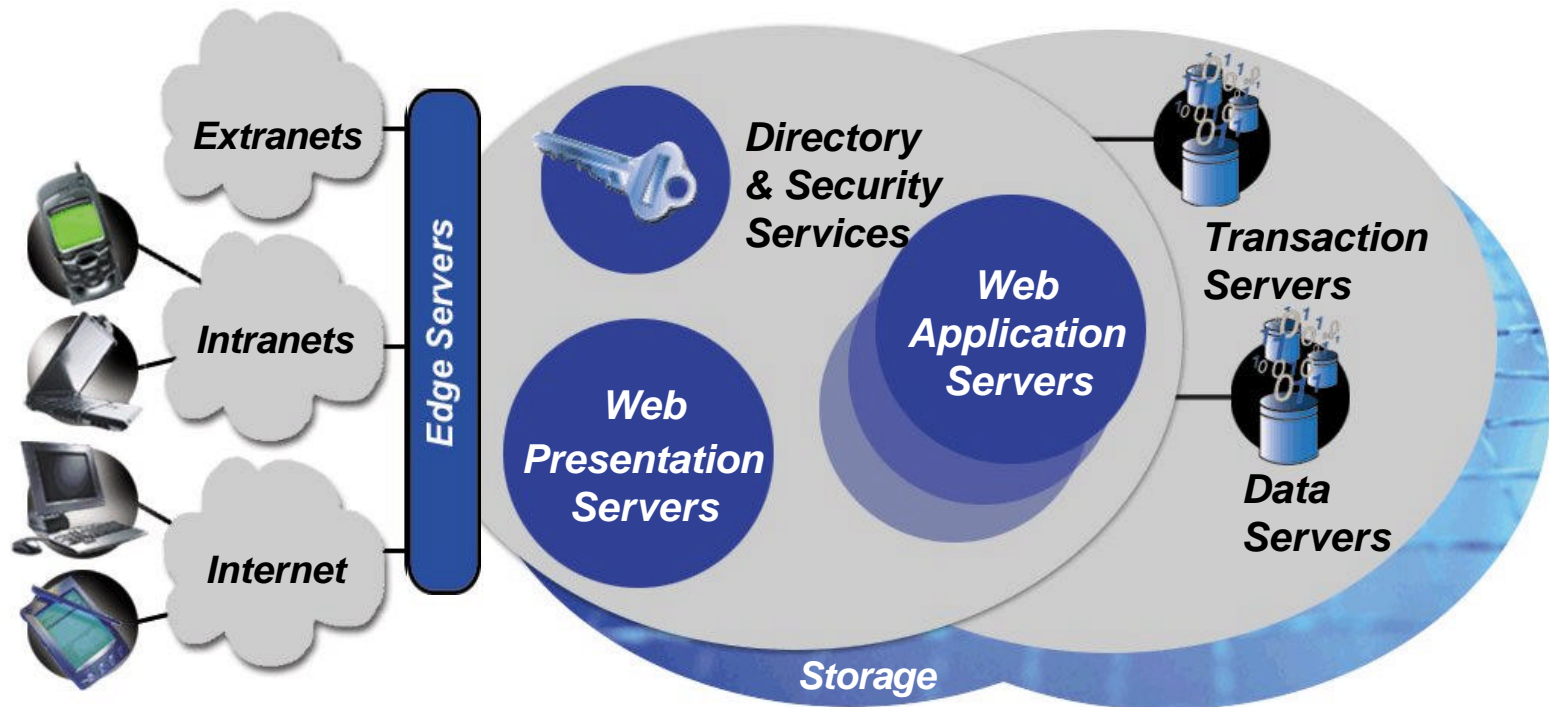
Infrastructure



End User Services

Web - Business Appl. Services

Data / Transaction Services



Responsive

Dynamic & Flexible

Reliable & Robust

IBM @server. For the next generation of e-business.



Dynamic e-business

The Networked economy is driving an evolution of e-business and integration is the key

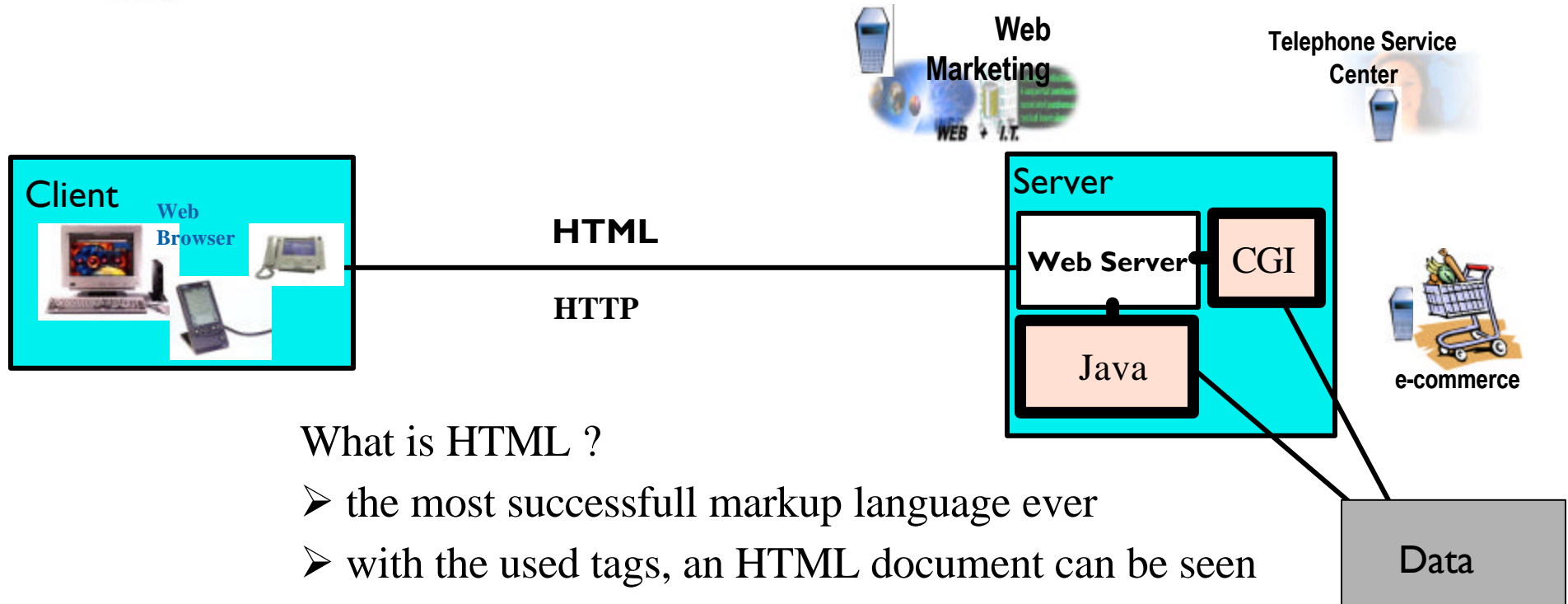
- Business-to-business
- Across towers
- With 3rd party software and services
- Across tiers



IBM @server. For the next generation of e-business.



HTML - Traditional Internet Technology



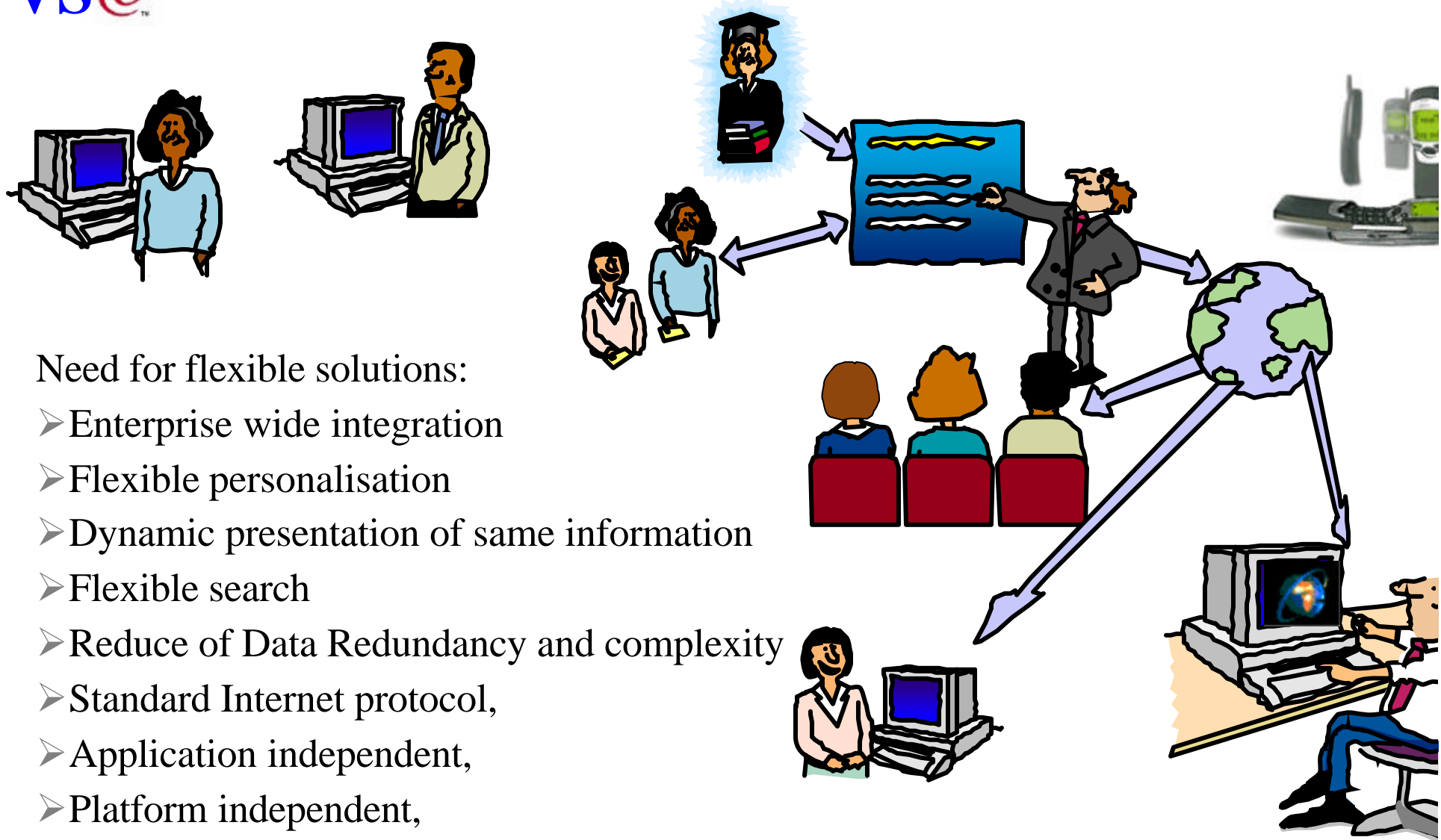
What is HTML ?

- the most successful markup language ever
- with the used tags, an HTML document can be seen on all platforms from Palmtops to mainframe
- extension interfaces like CGI (Common Gateway Interface) and Java (applets, servlets, EJB) allow access to backend systems
- **HTML was designed with humans in mind**

IBM @server. For the next generation of e-business.



HTML - wasn't designed for today's requirements



Need for flexible solutions:

- Enterprise wide integration
- Flexible personalisation
- Dynamic presentation of same information
- Flexible search
- Reduce of Data Redundancy and complexity
- Standard Internet protocol,
- Application independent,
- Platform independent,
- Architecture independent,

IBM @server. For the next generation of e-business.

VS@

HTML vs. XML - extended Markup Language

- HTML - contains tags to tell a browser how to display information
- but not WHAT that information is
- XML was designed with applications in mind (distributed application)
- XML has information about structure and content of information
- XML supports **attributes** that hold additional information about a **tag**
- HTML can be written within XML

An address in HTML

```
<p>  
<b>Mrs. Mary Brown</b>  
<br>  
1401 MainStreet  
<br>  
Winston Salem, WN 34123  
</p>
```

An address in XML

```
<address>  
  <title>Mrs.</title>  
  <firstname>Mary</firstname>  
  <lastname>Brown</lastname>  
  <street>1401 MainStreet</street>  
  <city state="WN">Winston Salem</city>  
  <postalcode type="int">34123</postalcode>  
</address>
```

IBM @server. For the next generation of e-business.

XML - extended Markup Language



addresses DTD - Document Type Definitions

addresses XSL - XML Style Sheet

addresses contains the descriptions of the tags, attributes and their valid values

addresses describes how the document should be displayed

addresses the XML document - validations

addresses stylesheet version="1.0" xmlns:xsl="http://www.w3.org/TR/WD-xsl">

addresses <!--ELEMENT AddressCollection (address+)-->

addresses <!--ELEMENT address (title?,firstname,lastname,street,city,postalcode)-->

addresses <!--ELEMENT title (#PCDATA)-->

addresses <!--ELEMENT first-name (#PCDATA)-->

addresses <!--ELEMENT lastname (#PCDATA)-->

addresses <!--ELEMENT street (#PCDATA)-->

addresses <!--ELEMENT city (#PCDATA)-->

addresses <!--ELEMENT postalcode (#PCDATA)-->

addresses <thead>

addresses <tr>

addresses <th>Name</th>

addresses <th>Address</th>

addresses </tr>

addresses <xsl:for-each select="address">

addresses <td>

addresses

addresses <xsl:value-of select="title"/>

addresses <xsl:value-of select="first-name"/>

addresses <xsl:value-of select="lastname"/></td>

addresses <td><xsl:value-of select="street"/>,</td>

addresses <xsl:value-of select="postalcode"/>

addresses <xsl:value-of select="city"/></td>

addresses </td>

addresses </xsl:for-each>

addresses </body></html>

addresses </xsl:stylesheet>

addresses </xsl:stylesheet>

addresses </xsl:stylesheet>

addresses </xsl:stylesheet>

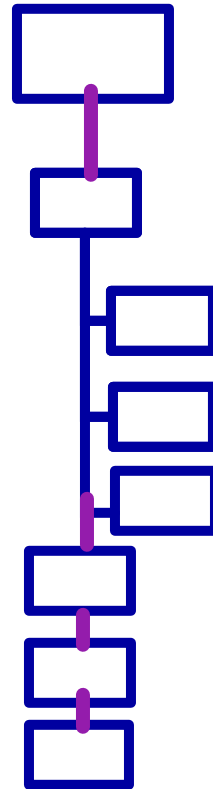
addresses </xsl:stylesheet>

addresses </xsl:stylesheet>

addresses </xsl:stylesheet>

addresses </xsl:stylesheet>

addresses </xsl:stylesheet>



An address in XML

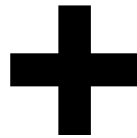
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE addresses SYSTEM "addresses.dtd">
<?xml-stylesheet type='text/xsl' href='addresses.xsl'?>
<AddressCollection>
  <address>
    <title>Mrs.</title>
    <first-name>
      Mary
    </first-name>
    <last-name>Brown</last-name>
    <street>
      1401 MainStreet
    </street>
    <city state="WN">Winston Salem</city>
    <postal-code type="int">34123</postal-code>
  </address>
</AddressCollection>
```


XSL – formatting of XML documents



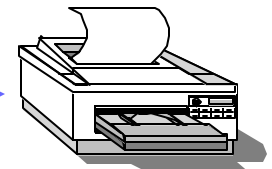
XML Source

```
<address>
<title>Mrs.</title>
<firstname>Mary</firstname>
<lastname>Brown</lastname>
<street>1401 MainStreet</street>
<city state="WN">Winston
Salem</city>
</address>
```

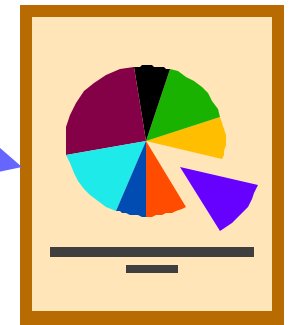


XML Style Sheet .XSL

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<html><body>
<h2>Addresses Collection</h2>
<xsl:for-each select="AddressCollection">
<table cellpadding="4" border="1">
<tr bgcolor="#efefef" align="left">
<th>Name</th>
<th>Address</th>
</tr>
```



Printer



Grafik



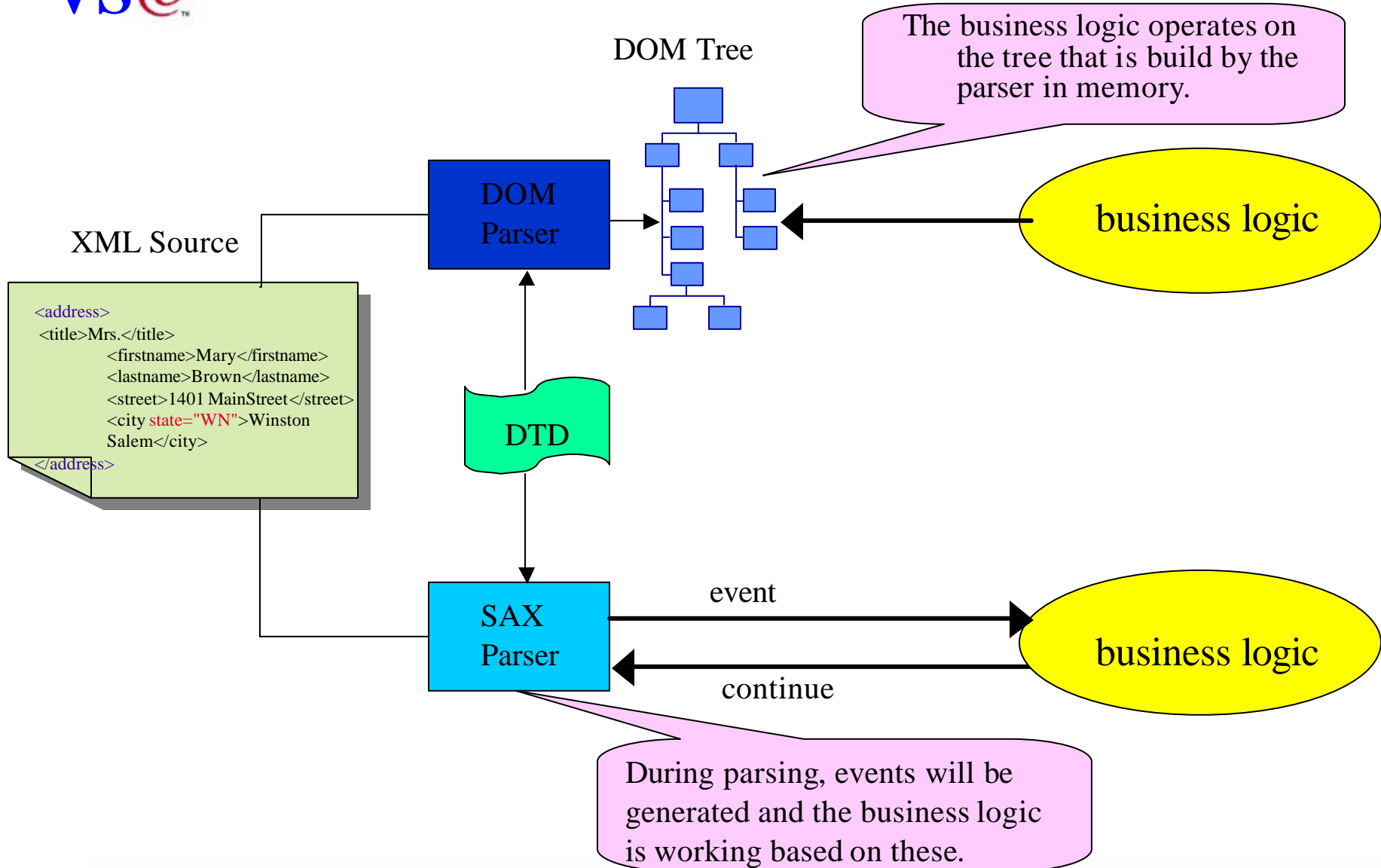
Report



Internet


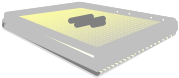
News Ticker

XML parser Technology (DOM,SAX)



What is SOAP?



- Simple Object Access Protocol
- SOAP is an **XML** based protocol for communication between two remote applications:
 - ▶ is based on RPC messaging
 - ▶ is language independent (de-couples interface from implementation)
 - ▶ represents remote procedure calls and responses
- A SOAP message consists of:
 - ▶  envelope
 - wraps the message itself
 - defines rules for decoding the message
 - ▶  message
 - request
 - method to invoke on a remote object and parameters
 - response
 - result of running the method and exceptions



SOAP XML data stream sample

SOAP Call

```
POST /cics/CWBA/SOAPSERV HTTP/1.0
Host: 9.164.143.23
Content-Type: text/xml; charset=utf-8
Content-Length: 505
SOAPAction: "urn:mytestservice"
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:testMethod
      xmlns:ns1="urn:mytestservice"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <filename xsi:type="xsd:string">
        FILEA
      </filename>
      <key xsi:type="xsd:string">
        00000001
      </key>
      <data xsi:type="xsd:string">
        Record1
      </data>
    </ns1:testMethod>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Response

```
HTTP/1.0 200 OK
Content-type: text/xml; charset="utf-8"
Content-Length: 00000391

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:testMethodResponse
      xmlns:ns1="urn:mytestservice"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <Result>
        OK
      </Result>
    </ns1:testMethodResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

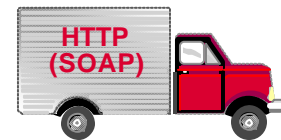
Parameters

What is WSDL?

VS[®]

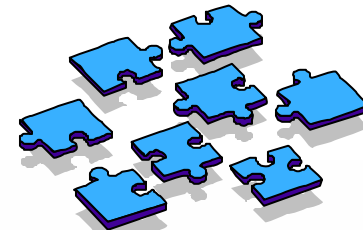


- Web Services Description Language
- WSDL is an **XML** based vocabulary for defining a Web Service:
 - ▶ interfaces
 - operation types (i.e. one-way, request-response, notification)
 - messages defining a Web Service interface
 - definition of data types (XML Schema)
 - ▶ access protocol (i.e. SOAP over HTTP)
 - ▶ contact endpoints (i.e. Web Service URL and URNs¹)



(URNs are location independent pointers to a file, or to different representations of the same content. In most ways they can be used like URLs)

- A Web Service URL returning WSDL makes Web Services self-describing
- Similar in purpose to IDL (Interface Definition Language)
 - From a WSDL file, wizards can generate:
 - proxy classes for calling Web Service
 - skeleton classes to implement a Web Service



IBM [®]server. For the next generation of e-business.



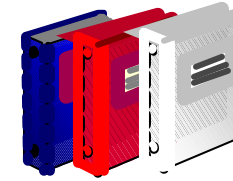
What is UDDI?



- Universal Description, Discovery and Integration
- UDDI is a specification for publishing and discovery of businesses and the services they provide
- UDDI specifications define how to construct UDDI Business Registries
- UDDI specifications are based on **XML** and **SOAP**:
 - ▶ API to communicate with a UDDI Registry are **SOAP** based
 - UDDI4J (UDDI for Java) - Open Source implementation in Java
 - JAXR (Java API for XML Registries) - Sun
 - ▶ data structures that define Web Service in UDDI Registry are **XML** based



UDDI Business Registry



- There are three public registry nodes on the web
 - ▶ sponsored by IBM, Microsoft and HP
 - ▶ in UDDI Business Registries, businesses register information about themselves and their services
 - ▶ registration is free
 - ▶ registration entries are replicated to other nodes on daily basis
 - ▶ businesses can discover each other's services:
 - automated searches (UDDI APIs)
 - manual searches (search engines in UDDI Registry web sites)
- UDDI Registries are organized into three areas:

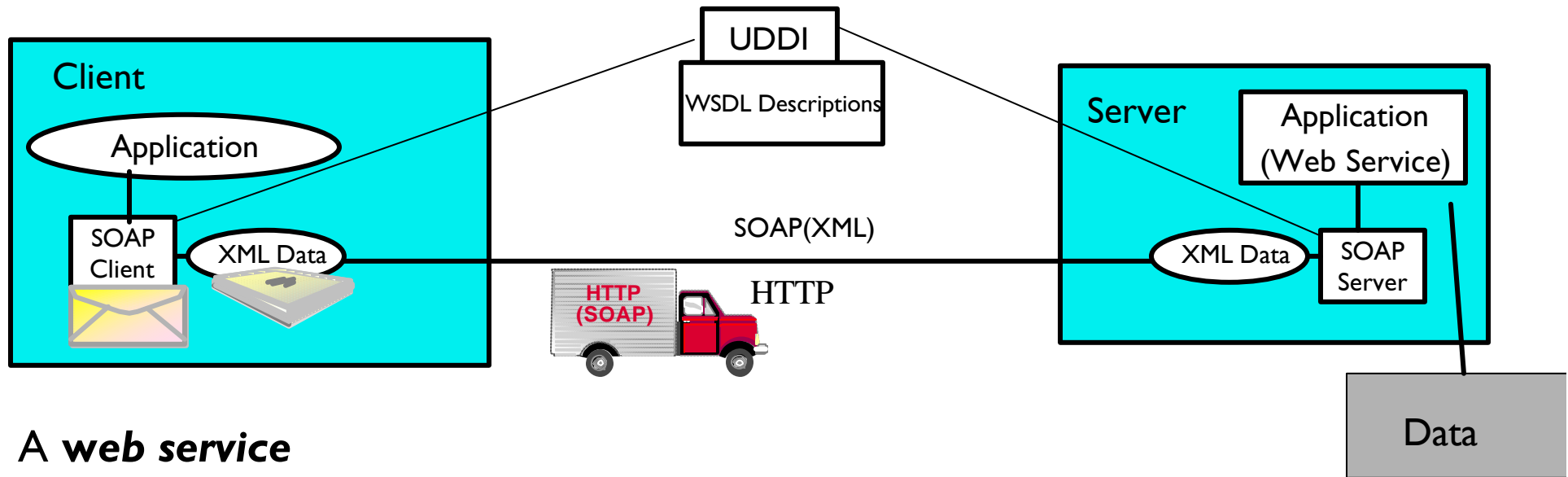
White Pages	business information	business name, description, contact information
Yellow Pages	service information	industrial categorizations based on standard taxonomies
Green Pages	binding information	the technical information about services that are exposed by the business

IBM @server. For the next generation of e-business.



Web Services

XML Document + SOAP Protocol = Web Services

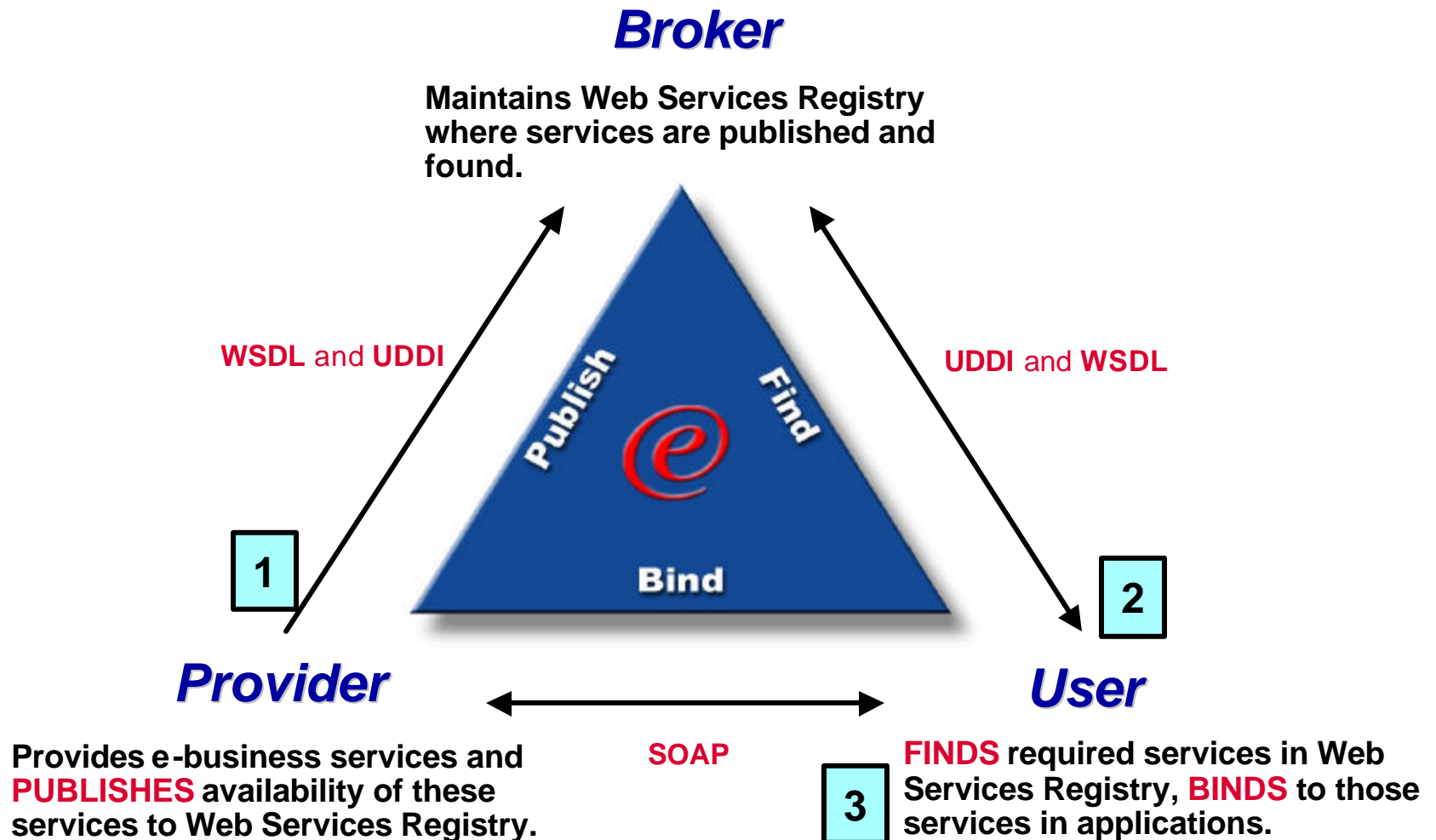


A *web service*

- ☞ implements a business, application or system functionality
- ☞ is intended for application communication
- ☞ is useable in internet, intranet, extranet
- ☞ is useable for browser-based solutions up to the B2B integration between companies
- ☞ uses only standard internet technologies

IBM @server. For the next generation of e-business.

Web Services Participants and their Roles



IBM @server. For the next generation of e-business.



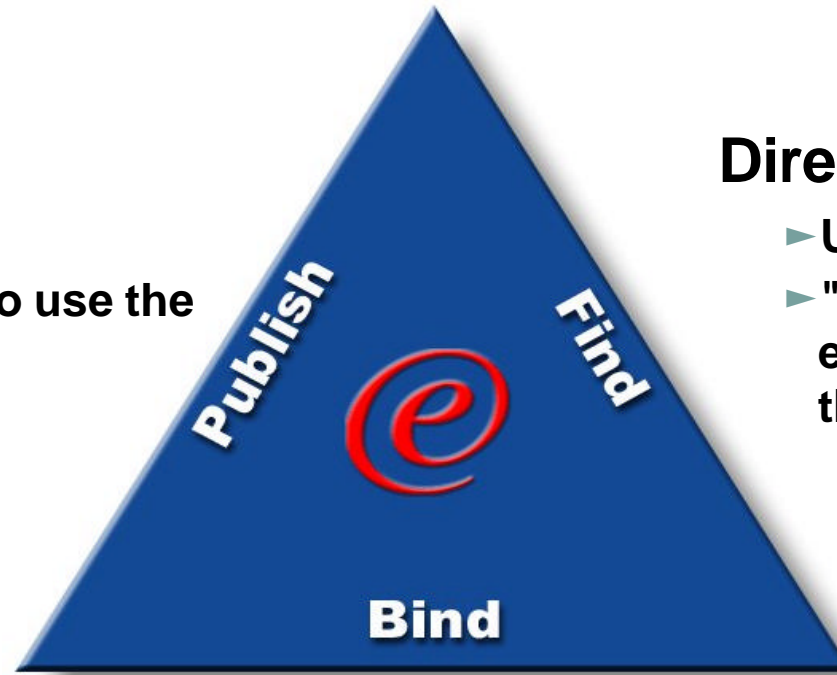
Web Services Architecture

Interface

- ▶ WSDL
- ▶ Defines how to use the service.

Directory

- ▶ UDDI
- ▶ "Yellow pages" that enable users to locate the services



Transport

- ▶ SOAP
- ▶ Mechanism for connecting with applications and data.

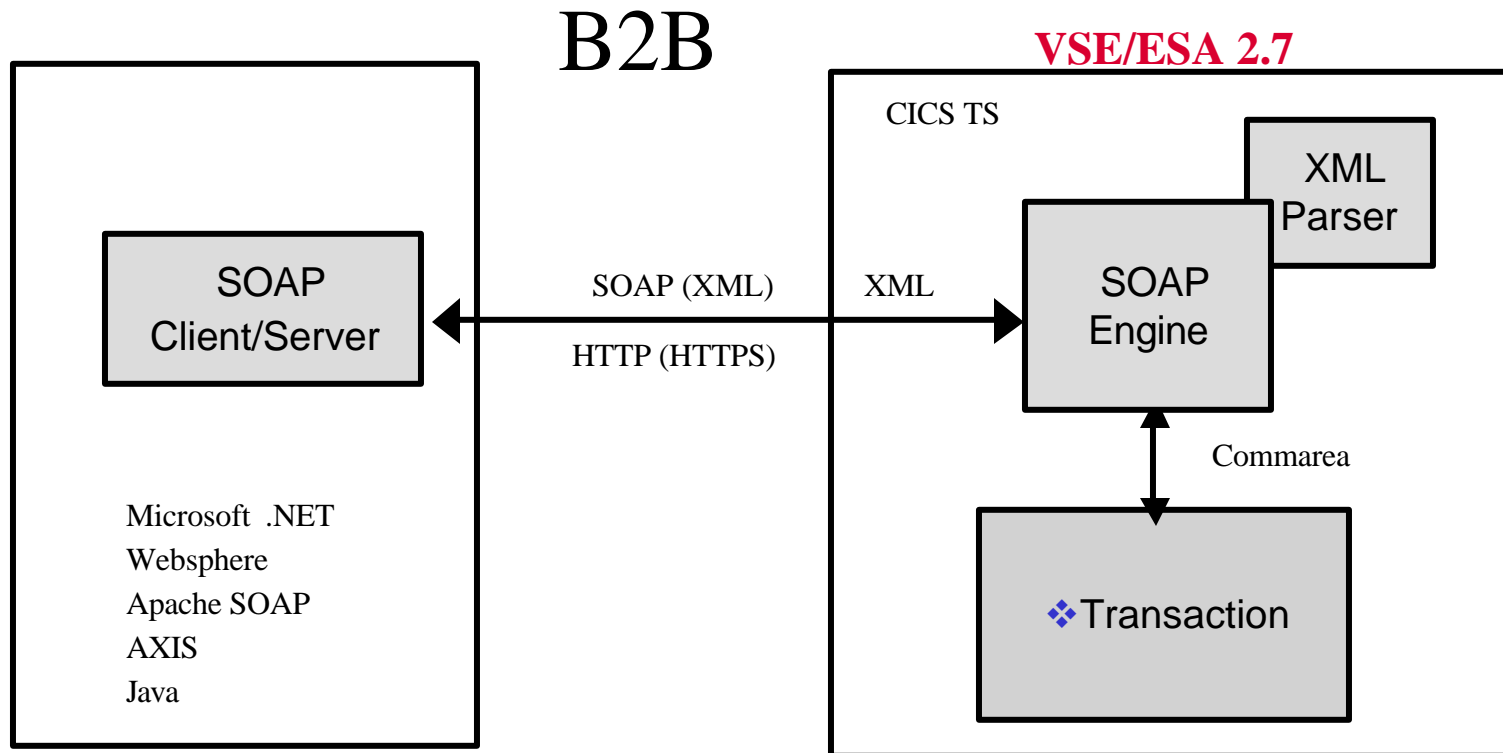
Set of industry-standard approaches to enable simplified connection of applications.

IBM @server. For the next generation of e-business.

Web services with VSE/ESA 2.7



Integration of CICS Transactions as Web Services (XML data interchange with SOAP)



❖ VSE/ESA Transactions as Web Service

IBM @server. For the next generation of e-business.



VSE XML Parser

- The VSE XML Parser can be invoked by user programs
- There are 2 calling interfaces:
 - SAX like interface to parse a XML document
 - Callable from C programs
 - Can be invoked directly
 - Uses Callback functions
 - DOM like interface to parse and generate XML documents
 - Callable via EXEC CICS LINK (or direct call from programs)
 - Builds a DOM like tree of Control blocks inn memory
 - XML generator converts the tree in a XML document



Overview

- VSE can act as
 - SOAP server
 - Driven through CICS Web Support
 - Allows to invoke a CICS program from remote
 - Transport protocol is HTTP (and HTTPS)
 - SOAP client
 - A CICS program can invoke a WebService
 - Transport protocol is HTTP
 - Connection possible through firewalls
 - HTTP Proxy
 - Socks V4/V5



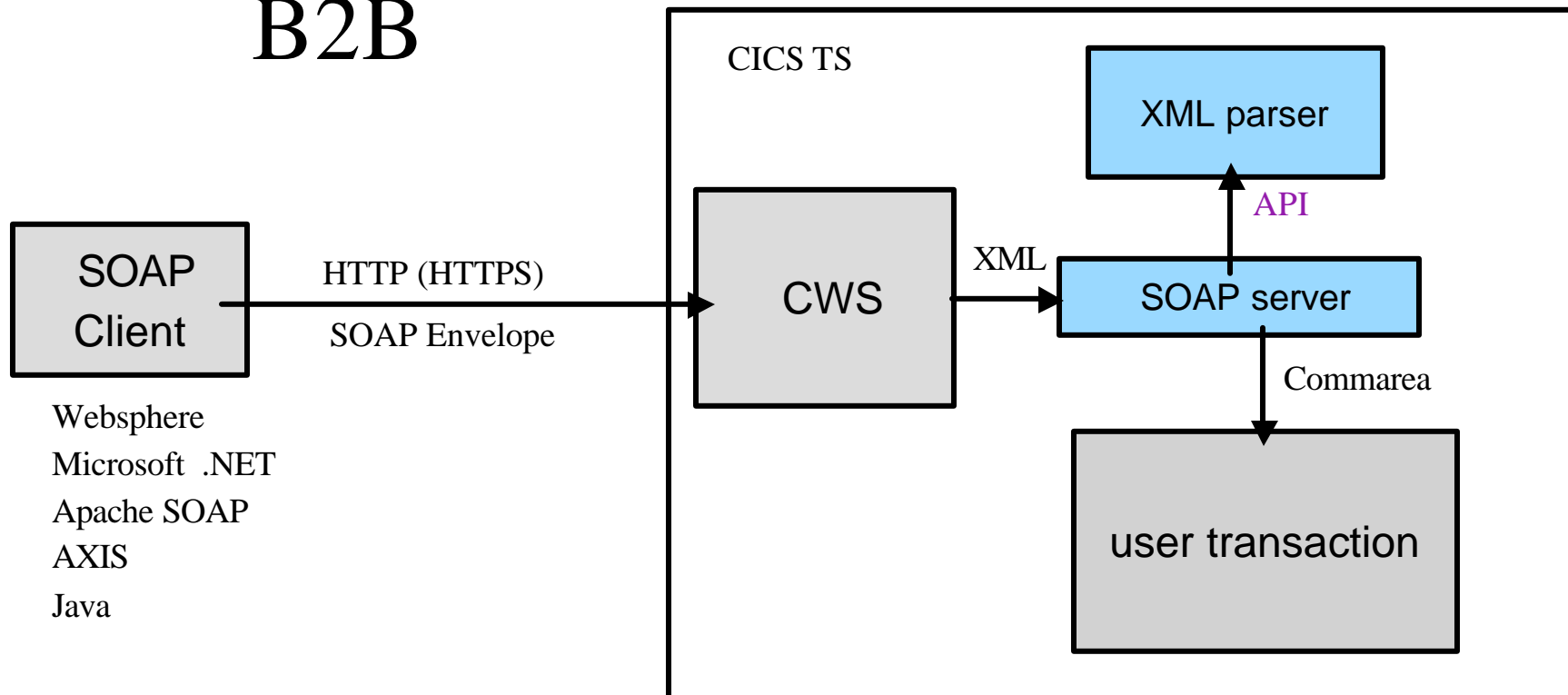
VSE/ESA as SOAP server

Web Services (SOAP)

SOAP - Simple Object Access Protocol
(platform independent remote procedure call)

VSE/ESA 2.7

B2B



IBM @server. For the next generation of e-business.



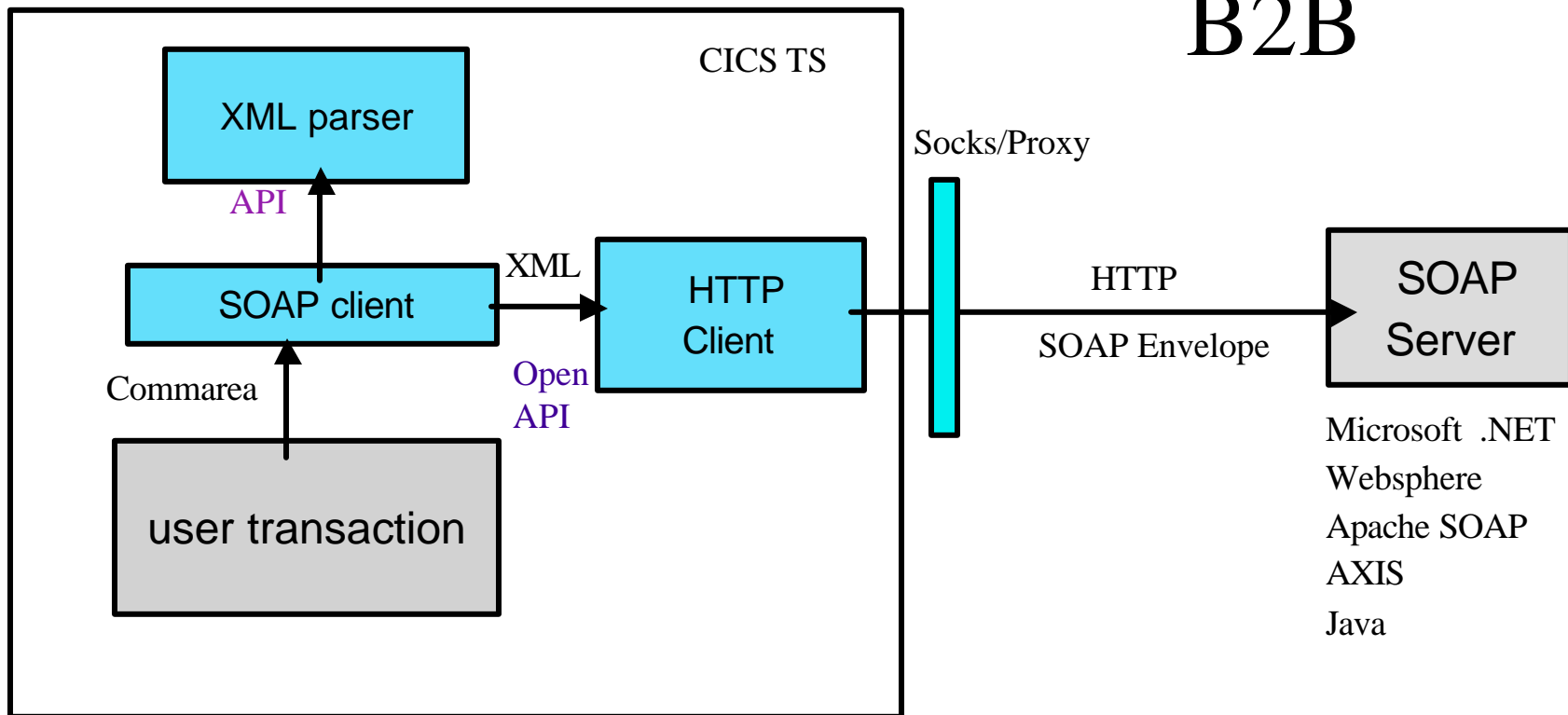
VSE/ESA 2.7 as SOAP client

Web Services (SOAP)

SOAP - Simple Object Access Protocol
(platform independent remote procedure call)

VSE/ESA 2.7

B2B



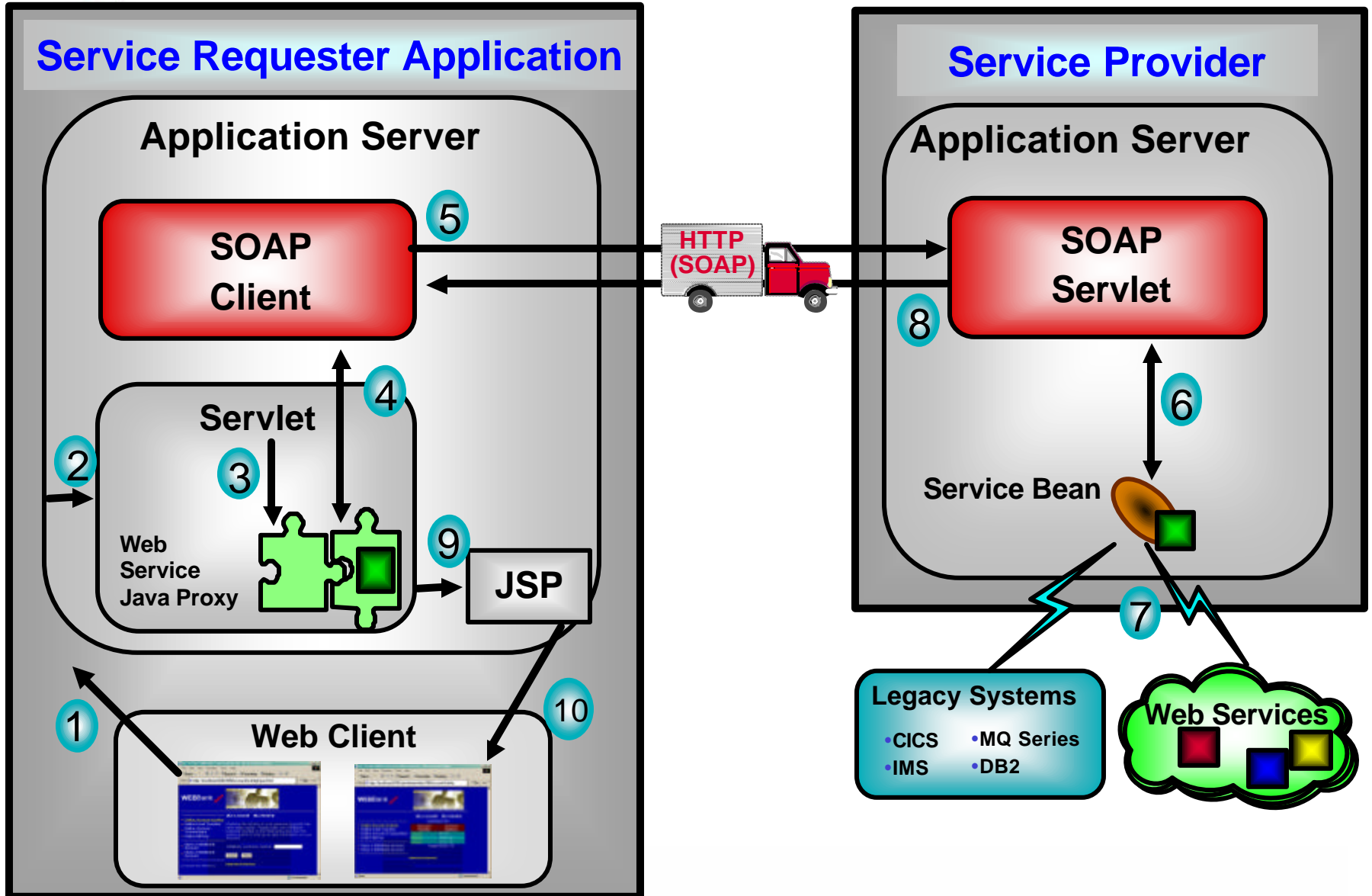
IBM @server. For the next generation of e-business.



VSE provided HTTP client

- The VSE provided HTTP client can be used by user written programs
- EXEC CICS LINK interface (or direct call)
- Supports connections through firewalls
 - HTTP proxy
 - Socks V4 and V5
- HTTP Methods supported
 - GET
 - POST
- Data to send/receive can be passed via
 - Buffers in memory
 - Callback functions/programs

Web Services Runtime View



IBM @server. For the next generation of e-business.

Web Services Value - Technical Perspective



1 Technology is backed by key software vendors (IBM, Microsoft, Oracle, etc.) and Open Source organizations (Apache)

- ▶ all cooperate to develop Web Services standards
 - Web Services protocols continue to evolve rapidly
 - Web Services are based on open protocols
- ▶ software vendors compete in the tooling and runtimes arena
 - high quality Web Services tooling is available
 - developers can concentrate only on writing the business logic
 - developers do not need to write the Web Services "plumbing" code
 - ◆ i.e. IBM's WSAD wizards generate (Java Proxy to consume Web Service, Java Bean Skeleton to create Web Service, etc...)
 - most runtimes (Web Application Servers) support consumption and provision of Web Services

2 Implementation is de-coupled from interface

- ▶ Web Services developers (producers and consumers) are not restricted to any particular language, operating system or object model
 - a Web Service built with .net technology can be called from J2EE application

IBM @server. For the next generation of e-business.

Web Services Value - Business Perspective



- Web Services technology will enable businesses to:

- 1 deliver new IT solutions **faster** and at **lower cost**

- development can focus on the code related to core business, and
- use Web Services application for non-core business programming

- 2 protect their investment in IT **legacy systems**

- use Web Services to wrap legacy software systems for integration with modern IT systems

- 3 **externalize** their business processes and **integrate** them with business processes of their customers and partners at a much lower cost

- Web Services make this integration feasible by allowing to share business processes without sharing technology
 - with lowered entry costs even small business will be able to participate in B2B integration

- 4 enter **new markets** and **widen customer base**

- Web Services listed in UDDI Registries can be "discovered" and thus are "visible" to the entire web community

IBM @server. For the next generation of e-business.

How Can Web Services Be Used?



- **Between businesses**
 - ▶ Providing service to your customers
 - ▶ Integrating business processes with your partners and suppliers
- **Within a business**
 - ▶ Accelerate and reduce the cost of integration
 - ▶ Save on infrastructure deployment and management costs
 - ▶ Reduce skill requirements
 - ▶ Improve reuse
- **Between a business and end-users**
 - ▶ Deliver a better user experience
 - ▶ Integrate diverse content
 - ▶ Reduce the cost of content delivery

- **Standards and common infrastructure reduce the barriers**
- **Simplicity accelerates deployment**
- **Dynamics opens new business opportunities**

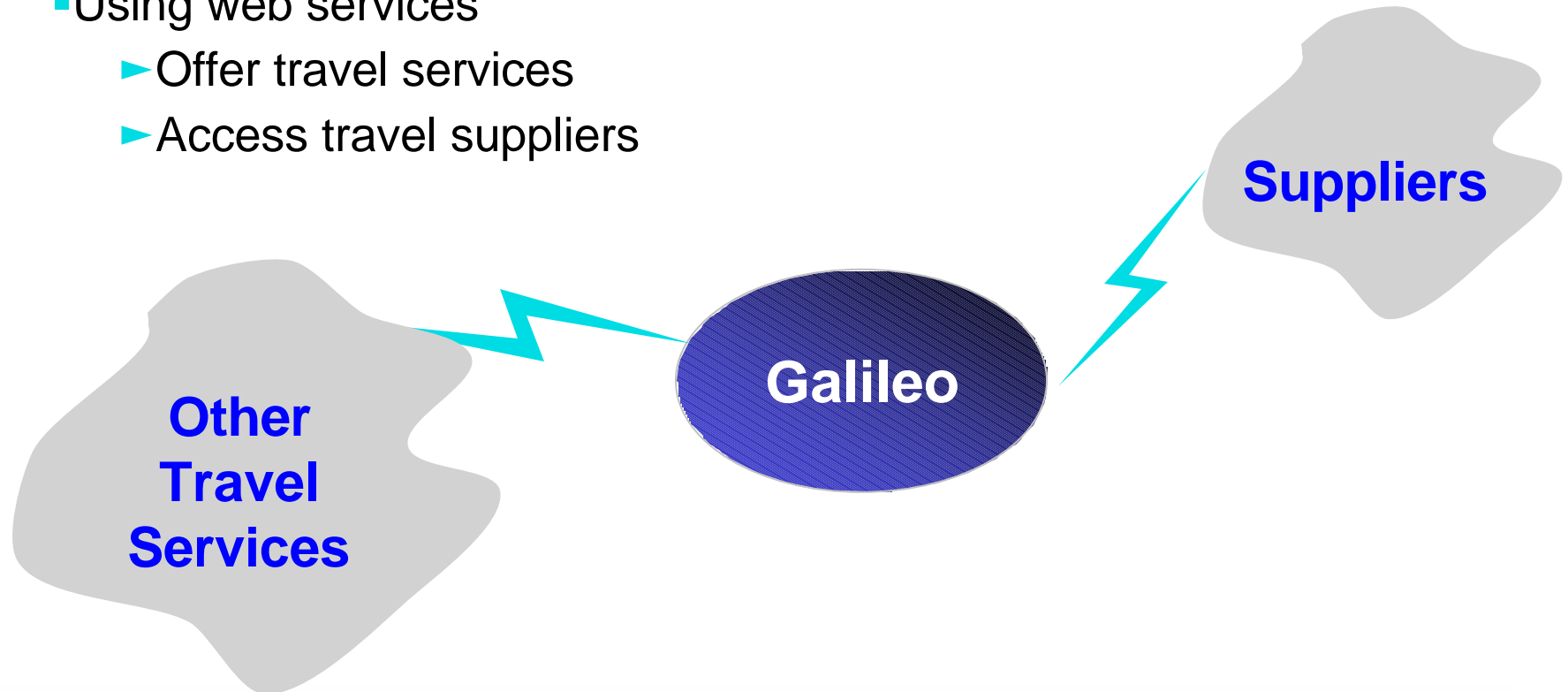


IBM @server. For the next generation of e-business.

Travel services

VS[®]

- One of the world's leading providers of electronic global distribution services (GDS) -- connecting more than 42,000 travel agency locations to 511 airlines, 37 car rental companies, 47,000 hotel properties, 350 tour operators and all major cruise lines in an electronic marketplace.
- Using web services
 - ▶ Offer travel services
 - ▶ Access travel suppliers

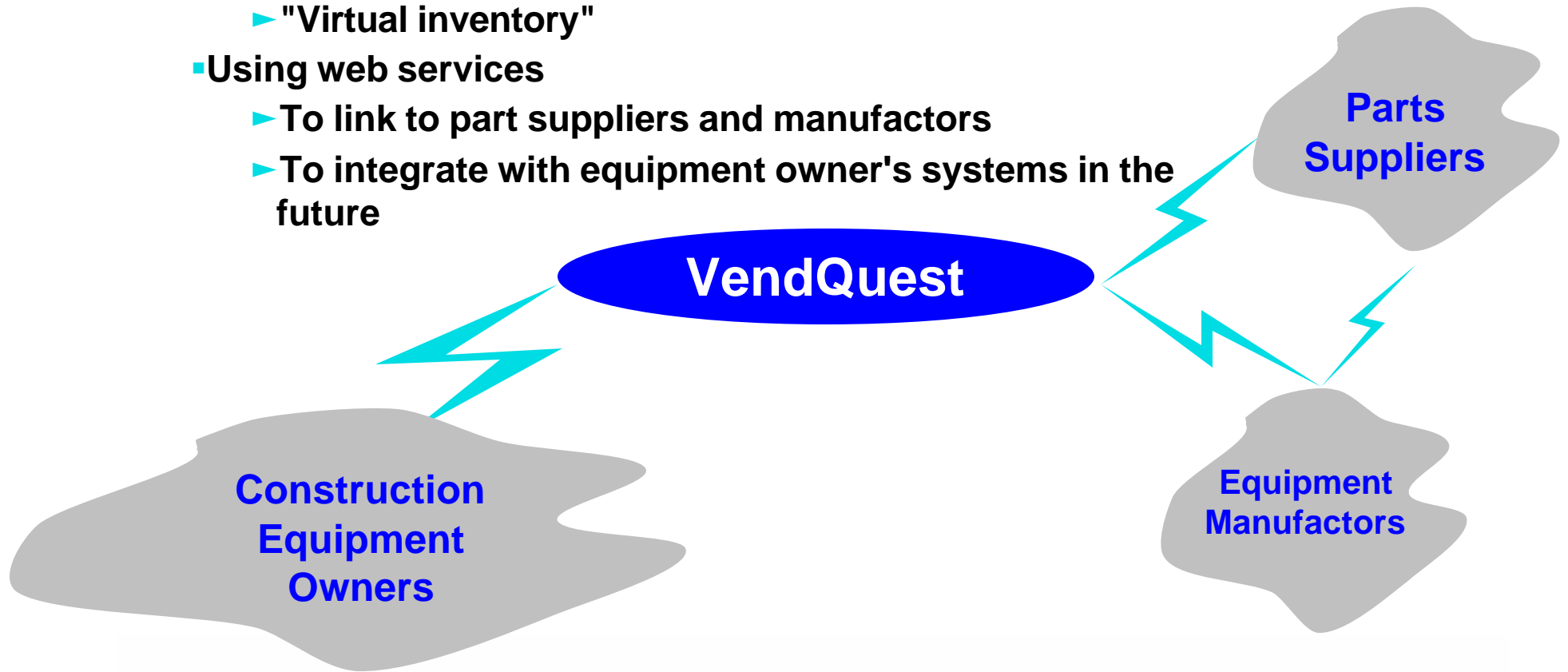


IBM [®]server. For the next generation of e-business.



e-Market

- Provide e-commerce support to construction equipment part suppliers
 - ▶ Integrate with parts suppliers systems
 - ▶ Immediate availability data
 - ▶ Immediate part reservation
 - ▶ "Virtual inventory"
- Using web services
 - ▶ To link to part suppliers and manufacturers
 - ▶ To integrate with equipment owner's systems in the future

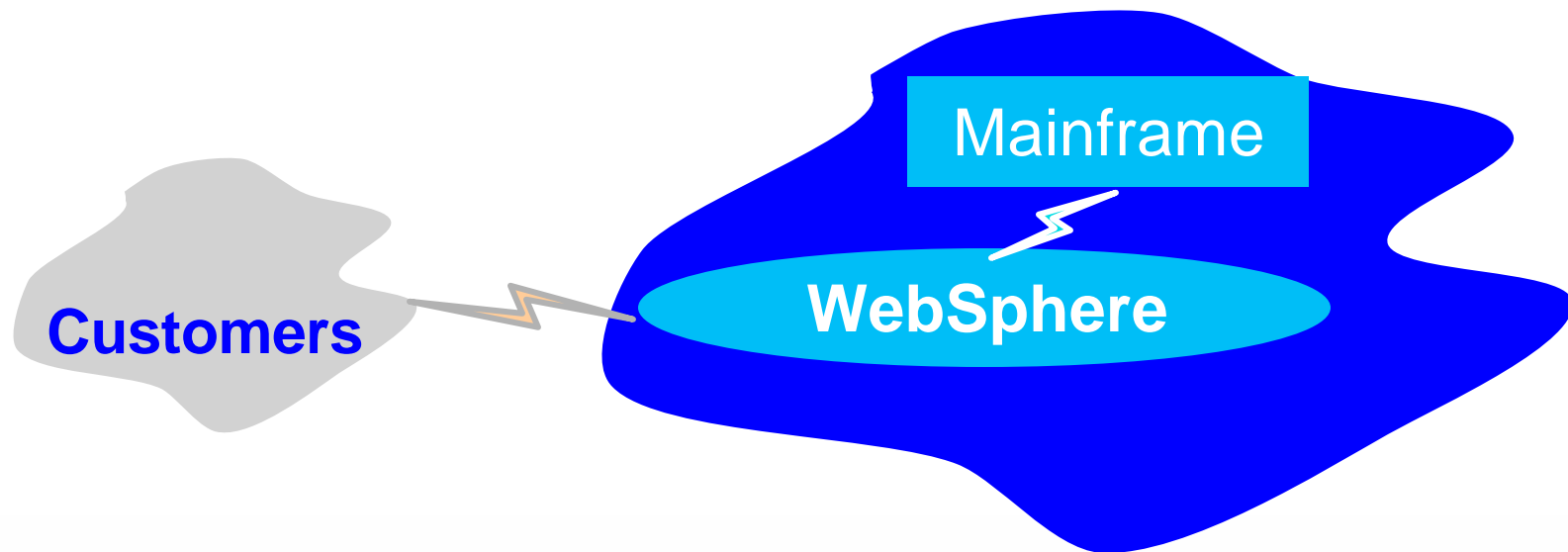


IBM @server. For the next generation of e-business.



Financial

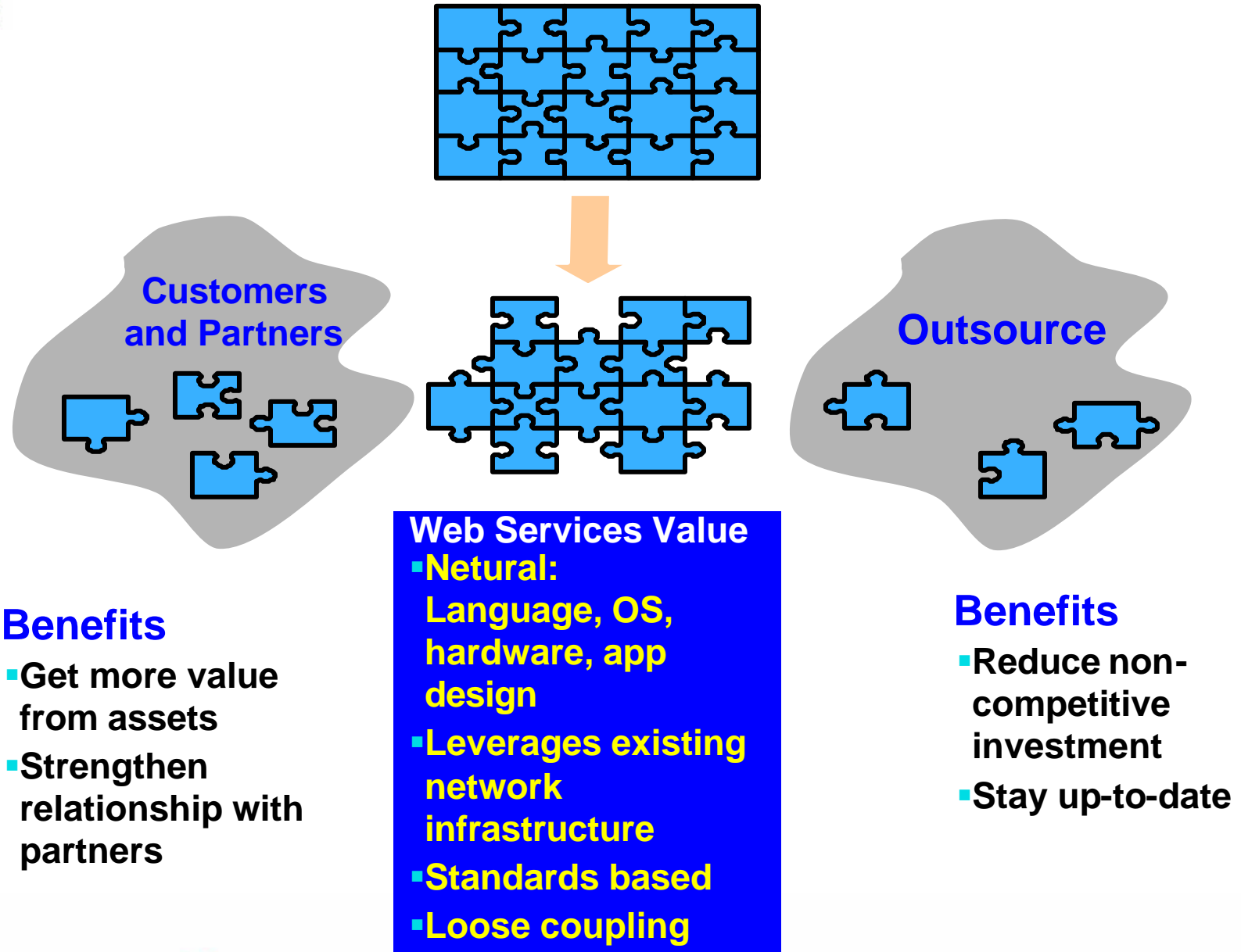
- A Financial and insurance holding company
- 6,500 companies with approximately 390,000 employees
 - ▶ Manually synchronize the records of those employees such that individual coverage could be accurately calculated for each employee under a variety of insurance schemes.
- Replace the manual process with an automated web services process
 - ▶ Extract information directly from payroll systems and transmit via web services requests to Storebrand where it is entered into Storebrand's mainframe



IBM @server. For the next generation of e-business.



Unbundling IT

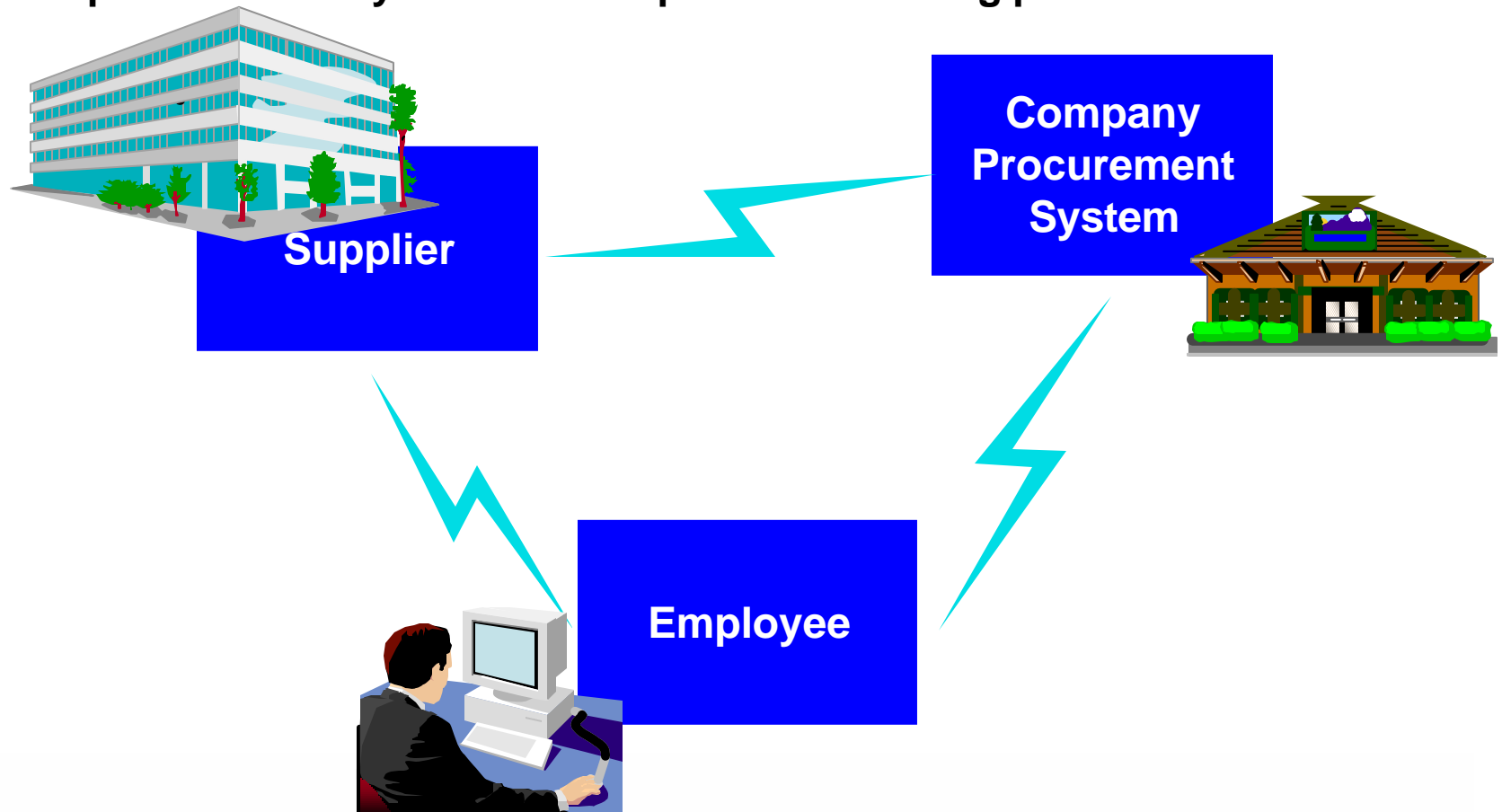


IBM @server. For the next generation of e-business.



Business Process Extension

- Supplier provides a UI for parts selection
- Company "passes-thru" UI for parts selection from its internal procurement system
- Supplier forwards the results of parts selection to the company's procurement systems to complete the ordering process



IBM @server. For the next generation of e-business.



Complete IBM Software Portfolio for Web Services

WebSphere	<p>Web Services Runtime and Deployment</p> <p><u>Web Application Server 4.0/50</u>. Includes SOAP Servlet and support for access to UDDI Registries.</p> <p><u>WebSphere Business Integrator</u>. New software to manage web services workflow within and between enterprises.</p> <p><u>Web Services Development Environment Toolkit</u>. AlphaWorks free Toolkit for basic Web Service programming tasks: create, deploy and test.</p>
Development	<p>Development and Integration of Web Services Applications</p> <p><u>VAJ V4.0/50 WebSphere Studio Application Developer</u> . IDE, hosting a complete web application development environment. Includes major Web Services functions such as:</p> <ul style="list-style-type: none">▪ Web Service creation: Java beans, DADX, DB2 XML Extender, WSDL, URLs;▪ Test Client generation for Web Services test and verification▪ Web Services consumption: generate Java Proxy from WSDL▪ Full interface to UDDI Registries to browse, discover, download and publish.

IBM @server. For the next generation of e-business.



Complete IBM Software Portfolio for Web Services

DB2 UDB 7.2/8.x	Manage Data in Web Services Environment <u>DB2 XML Extender</u> . Stored procedures and SQL statements to be exposed as invocable Web Service operations. Enables Web services applications to access data stored in DB2 as an XML structured document.
Tivoli	Manage the IBM Middleware that Runs the Web Services <u>Tivoli Web Services Manager</u> . Software for performance monitoring of Web Services. <u>Tivoli Manager for WebSphere Application Server</u> . Software providing a single point of control to manage Web Services. <u>Tivoli SecureWay Policy Director</u> . Software to build consistent security policy into Web Services.
Lotus	Web Services Enablement in Lotus Products <u>Lotus Web Services Enablement Kit</u> . Software tools to build Web Services with Lotus products. <u>Lotus Domino Server</u> . Collaboration, workflow and messaging software that can be accessed by Web Services. <u>Lotus Knowledge Discovery System</u> . Knowledge management software to access expertise from Web Services.

IBM @server. For the next generation of e-business.



The Magic of Web Services

What makes web services different from previous "revolutionary" component and communication models?

- **Based on real Standards**
 - ▶ TCP/IP, HTTP, XML, ...

- **Leverages industry support**
 - ▶ The Internet, routers, load balancing, firewalls, web servers, ...

- **We made the right choices this time**
 - ▶ Loose coupling, simple, solution driven

IBM @server. For the next generation of e-business.



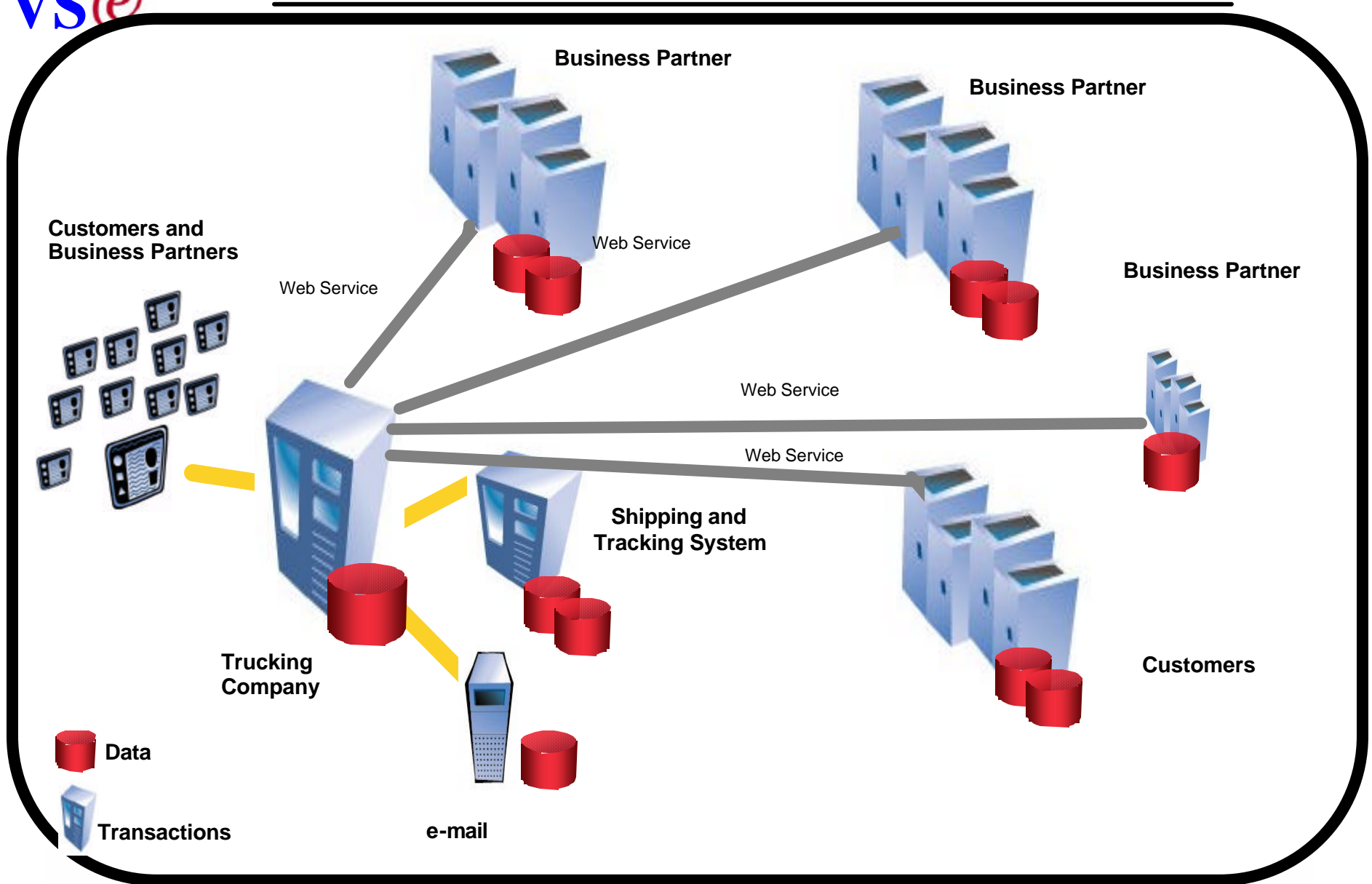
Why will Web Services Technology Succeed?

- **Other distributed technologies are not optimized for the Internet because they required strong coupling:**
 - ▶ required the use of the same transport technologies
 - ▶ required knowledge of each others implementation
- **What is wrong with other distributed technologies?**
 - ▶ DCOM - requires Windows at each endpoint
 - ▶ CORBA - requires compatible ORBs at each endpoint
 - ▶ RMI - requires Java at each endpoint
- **Why is XML/WSDL/SOAP/UDDI so much better?**
 - ▶ implementation choices are de-coupled from interfaces
 - any language can be used to write and consume Web Services
 - more than one open standard transport technology can be specified
 - operating system differences are not a factor

IBM @server. For the next generation of e-business.

Roadmap for dynamic e-business

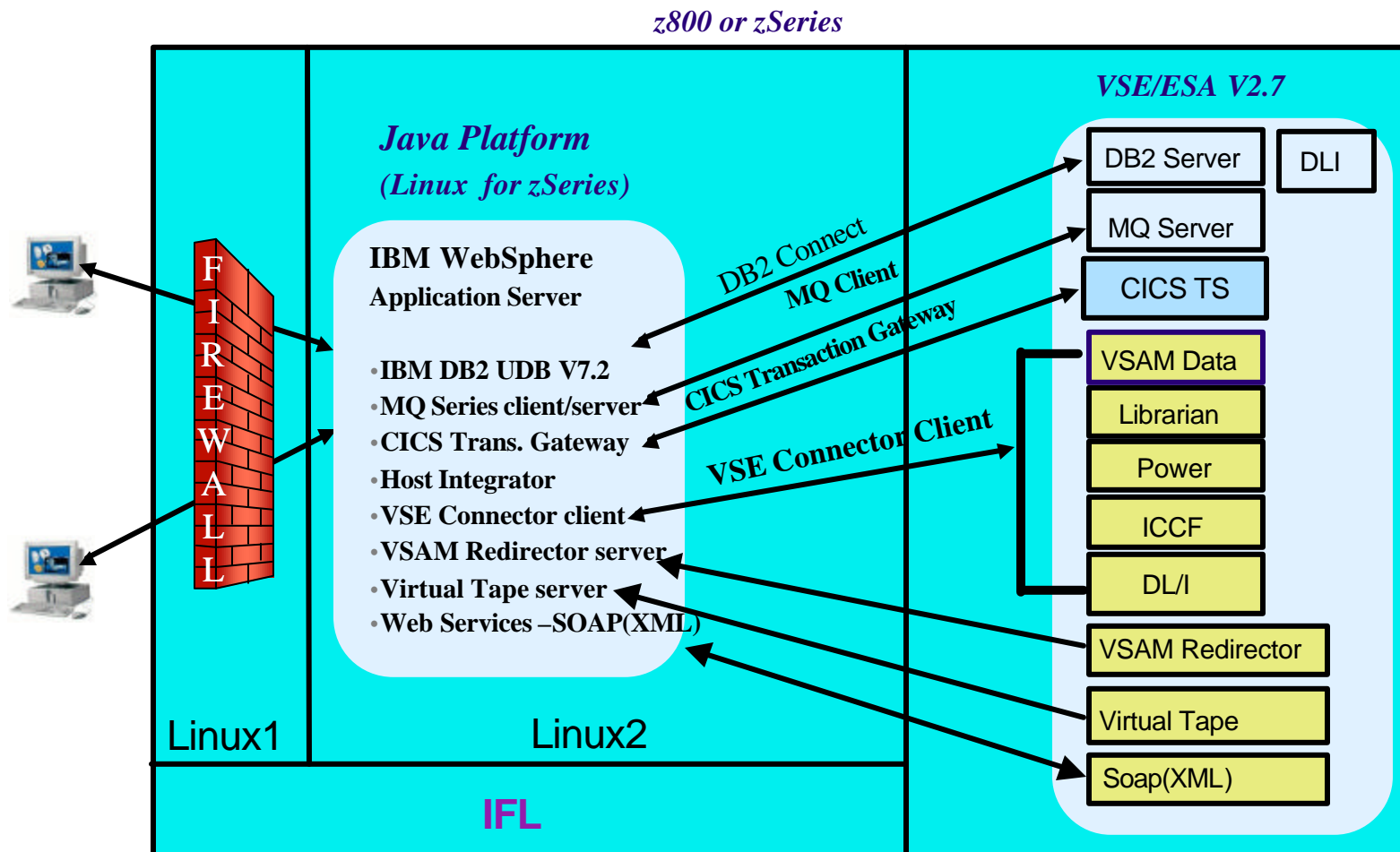
VS@



IBM @server. For the next generation of e-business.



VSE/ESA Connections



IBM @server. For the next generation of e-business.



More Information...

<u>developerWorks - Web Services Zone</u> IBM sponsored site that contains articles, tutorials and latest news related to Web Services.	www-106.ibm.com/developerworks/webservices
<u>AlphaWorks - Web Services Toolkit</u> IBM sponsored site to provide early adopters access to emerging "alpha-code" technology. The Web Services Toolkit exploits new technologies that may be adopted in future releases of IBM Web Services Tools.	www.alphaworks.ibm.com/tech/webservicestoolkit
<u>XMethods</u> An organization dedicated to promoting the development, deployment, and use of web services.	xmethods.com
<u>UDDI Organization</u> UDDI Project and Community web site contains UDDI specification, whitepapers, FAQ. IBM is a member.	www.uddi.org
<u>SOAP</u> The latest version of SOAP specification hosted by the W3C Organization web site.	www.w3.org/TR/SOAP
<u>WSDL</u> The latest version of WSDL specification hosted by the W3C Organization web site.	www.w3.org/TR/wsdl