



E11

Web Services, modern technology of future computing

Wilhelm Mild

VSEESA@de.ibm.com

zSeries Expo

Nov. 1 - 5, 2004

Miami, FL

© IBM Corporation 2004

Trademarks

References in this publication to IBM products or services do not imply that IBM intends to make them available in every country in which IBM operates. Consult your local IBM business contact for information on the products, features, and services available in your area.

AIX*, APPN*, CICS*, CICS/VSE*, CICS, DB2*, DB2 Connect, DB2 Universal Database, DFSORT, DRDA*, e-business logo*, Enterprise Storage Server, FlashCopy, HiperSockets, IBM*, IBM logo*, IBM eServer, iSeries, Language Environment*, MQSeries*, Multiprise*, pSeries, S/390*, S/390, Parallel Enterprise Server, TotalStorage, VSE/ESA, VTAM*, WebSphere*, xSeries, z/OS, z/VM, zSeries and Distributed Relational Database Architecture are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds.

Java and all Java-related trademarks and logos are trademarks or registered trademark of Sun Microsystems, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through The Open Group.

Microsoft, Windows, Windows NT, Visual Basic and the Windows flat logo are Trademarks of Microsoft Corporation.

Other trademarks and registered trademarks are the properties of their respective companies.

IBM hardware products are manufactured from new parts, or new and used parts. Regardless, our warranty terms apply. This equipment is subject to all applicable FCC rules and will comply with them upon delivery. Information concerning non-IBM products was obtained from the suppliers of those products. Questions concerning those products should be directed to those suppliers.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

(C) Copyright IBM Corporation 2004
All Rights Reserved.

Web services

What are Web Services?

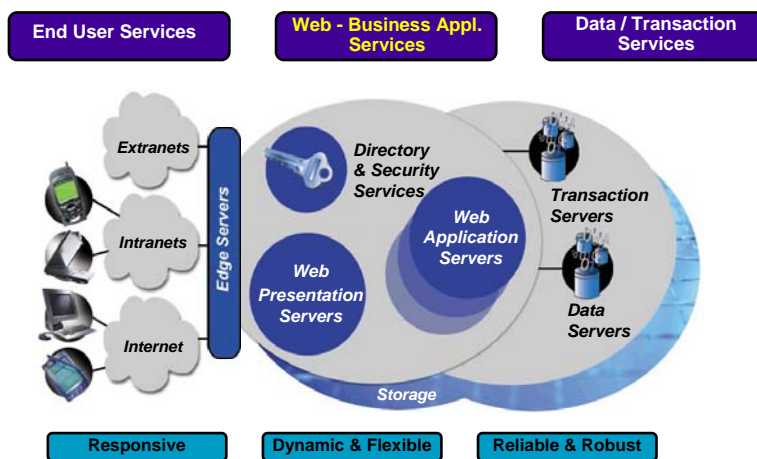
Web Services Architecture

Web Services Protocols

What is the value of Web Services?

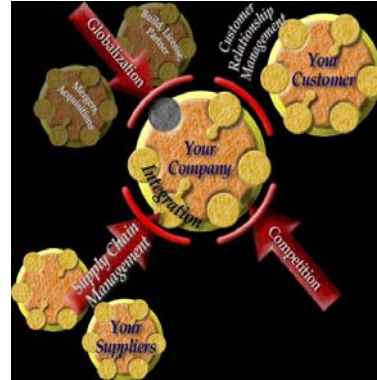


Infrastructure



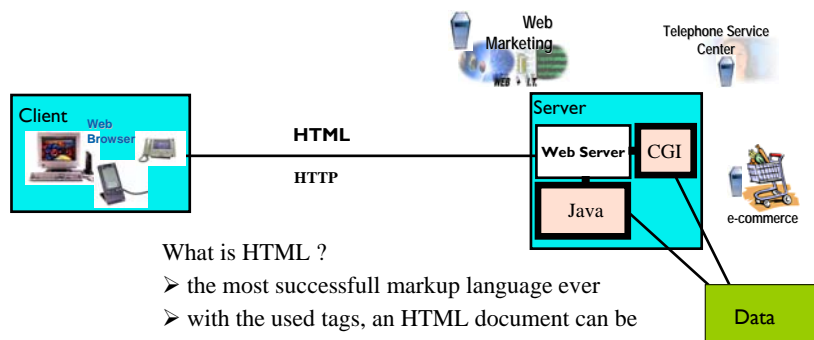
Dynamic e-business

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

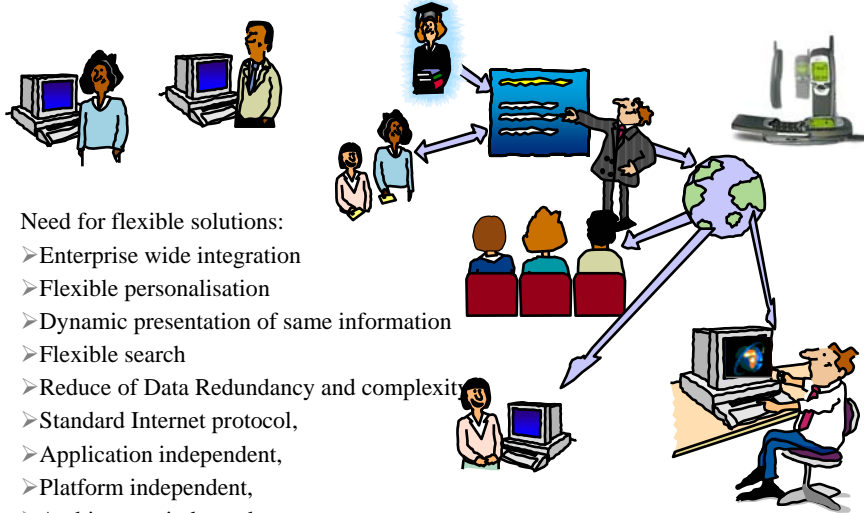
HTML - Traditional Internet Technology



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**

HTML - wasn't designed for today's requirements



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

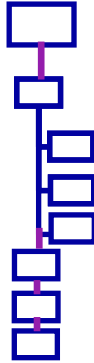
```
<address>  
  <title>Mrs.</title>  
  <firstname>Mary</firstname>  
  <lastname>Brown</lastname>  
  <street>1401 MainStreet</street>  
  <city state="WN">Winston Salem</city>  
  <postalcode type="int">34123</postalcode>  
</address>
```

XML - extendet Markup Language

Addresses.DTD – Document Type Definitions

Addresses.DTD is an XML Schema Definition that defines the tags, attributes and their valid values. It contains the definitions how the document should be displayed.

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



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>
    <name>
      <title>Mrs.</title>
      <first-name>
        Mary
      </first-name>
      <last-name>Brown</last-name>
    </name>
    <street>
      1401 MainStreet
    </street>
    <city state="WN">Winston Salem</city>
    <postal-code type="int">34123</postal-code>
  </address>
</AddressCollection>
```

XML – Namespaces and Schemas

- **Example:**
 - <state>xxxxxxx</state> - is it a country state, a state for a material, ...
 - <code type="int">34123</code> - is it a code for a piece, postal code ...

The XML language has No naming restrictions – therefore the needs for definitions

- `xmlns:xsd="http://www.w3.org/2001/XMLSchema"`
 - Defines the data types and their underlying functions with it, i.e. "xsd:string"
- `xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`
 - defines the namespace for the "type" – Schema instance, i.e. "xsi:type"
 - If "type" is specified only, the program doesn't know if a schema with this name is meant or a data definition. Therefore the specification of the Schema-Instance Namespace for the "type" attribute can explain that a type definition from this schema is meant.

XML – Namespaces and Schemas

Schema definition for STRING:

```
<xs:simpleType name="string" id="string">
  <xs:annotation>
    <xs:appinfo>
      <hfp:hasFacet name="length" />
      <hfp:hasFacet name="minLength" />
      <hfp:hasFacet name="maxLength" />
      <hfp:hasFacet name="pattern" />
      <hfp:hasFacet name="enumeration" />
      <hfp:hasFacet name="whiteSpace" />
      <hfp:hasProperty name="ordered" value="false" />
      <hfp:hasProperty name="bounded" value="false" />
      <hfp:hasProperty name="cardinality" value="accountably infinite" />
      <hfp:hasProperty name="numeric" value="false" />
    </xs:appinfo>
    <xs:documentation source="http://www.w3.org/TR/xmlschema-2/#string" />
  </xs:annotation>
  <xs:restriction base="xs:anySimpleType">
    <xs:whiteSpace value="preserve" id="string.preserve" />
  </xs:restriction>
</xs:simpleType>
```

XML – Namespaces and Schemas

A Schema defines a namespace (xmlns) for a group of data types.
The XML file can then use the defined types:

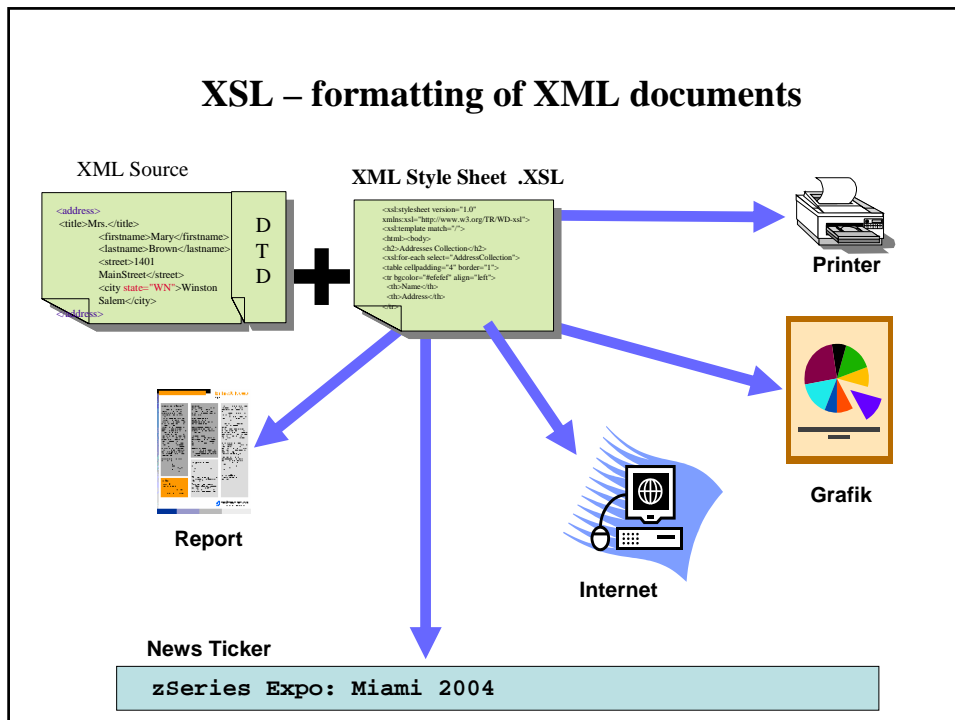
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MYRECORD SYSTEM "myrecord.dtd">
<record
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <address>
    <lastname xsi:type="xsd:string">Mueller</lastname>
    <firstname xsi:type="xsd:string">Klaus</firstname>
    <location xsi:type="xsd:string" zip="12345">Munich</location>
  </address>
  <employee>
    <empno xsi:type="xsd:int">1234567</empno>
    <salary xsi:type="xsd:float">1234.56</salary>
  </employee>
</record>
```

With a DTD and a schema, a document is self-described

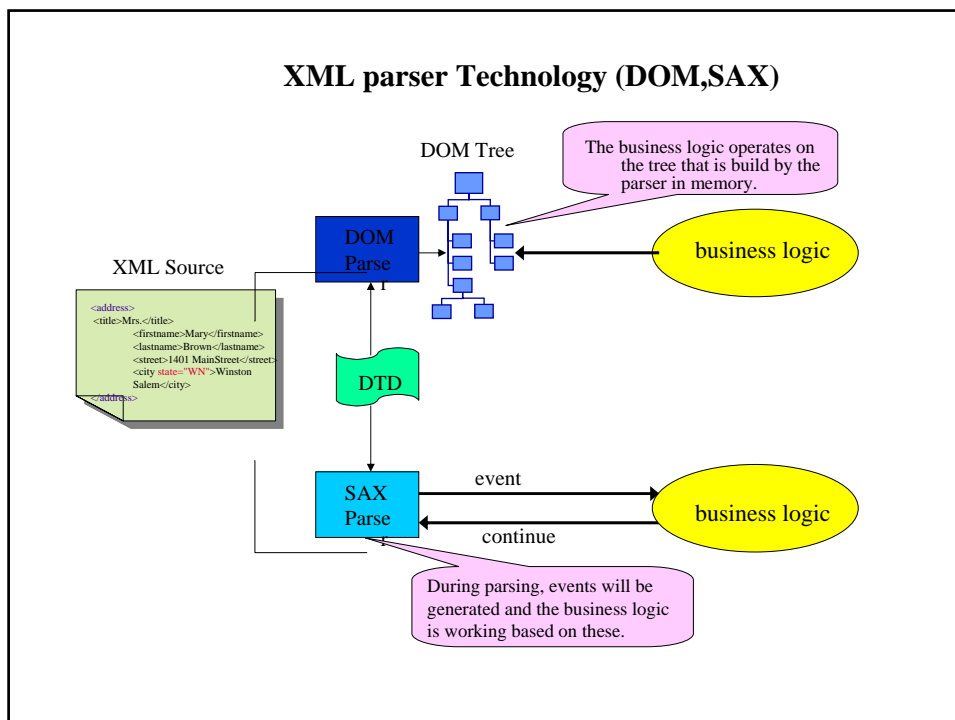
Without a DTD and a schema (namespace) the program must know:

- structure of the XML document
- meaning of the tags
- what data types are used in that document

XSL – formatting of XML documents





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

Simple Object Access Protocol (SOAP)

Call:

```
<soap:Envelope>
  <soap:Body xmlns:m="http://www.stock.org/stock" />
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

Answer / Result:

```
<soap:Envelope>
  <soap:Body xmlns:m="http://www.stock.org/stock" />
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>
</soap:Envelope>
```


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

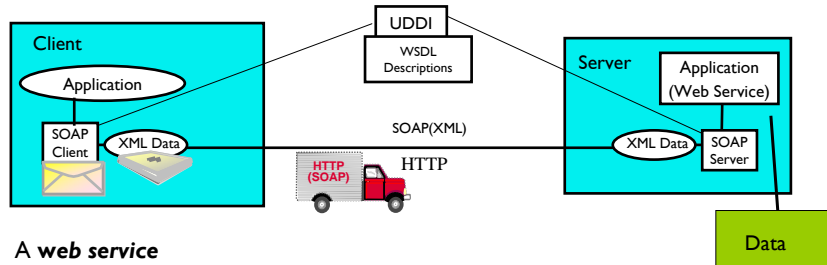


- **There are three public registry nodes on the web**
 - ▶ sponsored by IBM, Microsoft and HP
 - ▶ in UDDI Business Registries, businesses register information about themselves and their services
 - ▶ registration is free
 - ▶ registration entries are replicated to other nodes on daily basis
 - ▶ businesses 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

Web Services

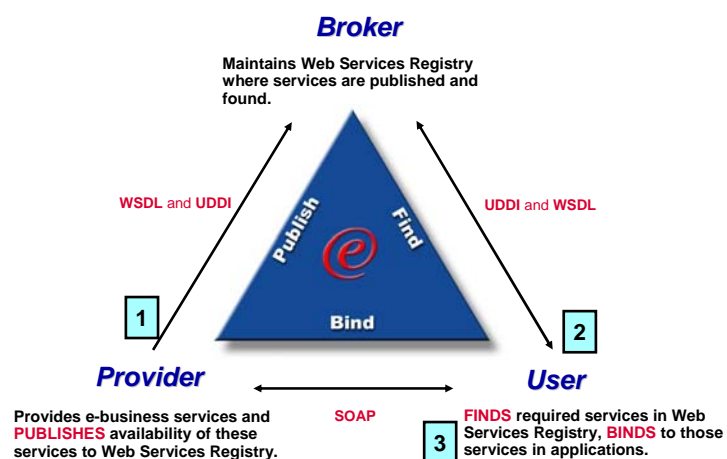
XML Document + SOAP Protocol = Web Services



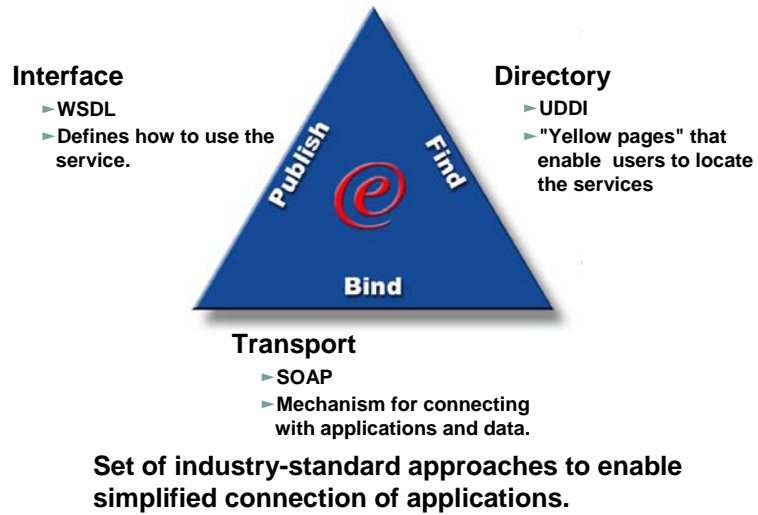
A web service

- ☞ 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

Web Services Participants and their Roles

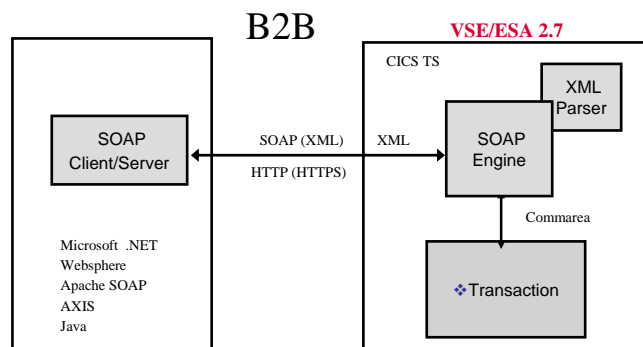


Web Services Architecture



Web services with VSE/ESA 2.7

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

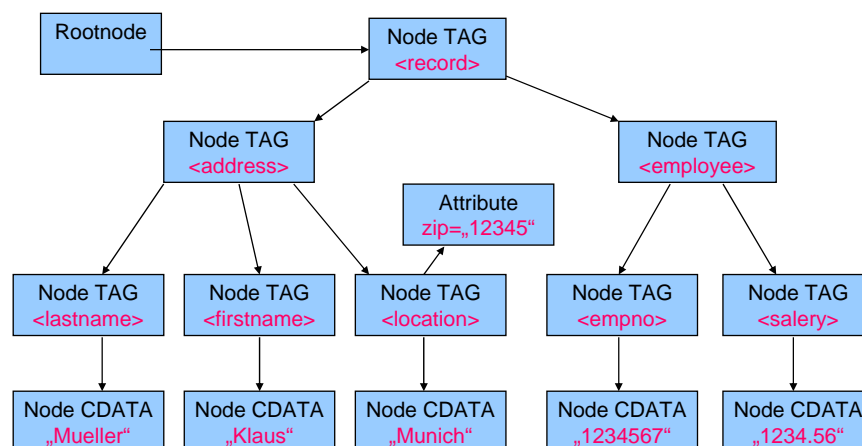


❖ VSE/ESA Transactions as Web Service

VSE XML Parser

- The VSE XML Parser can be invoked by user programs
- Entire XML documents only can be parsed
- 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 in memory
 - XML generator converts the tree in a XML document

The VSE XML Parser (DOM) – XML tree



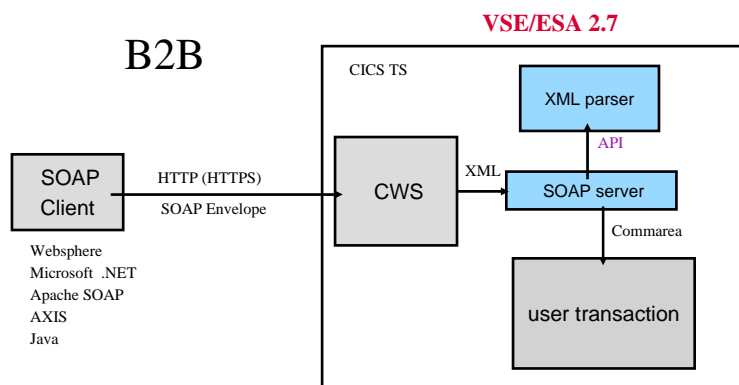
Overview

- 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)

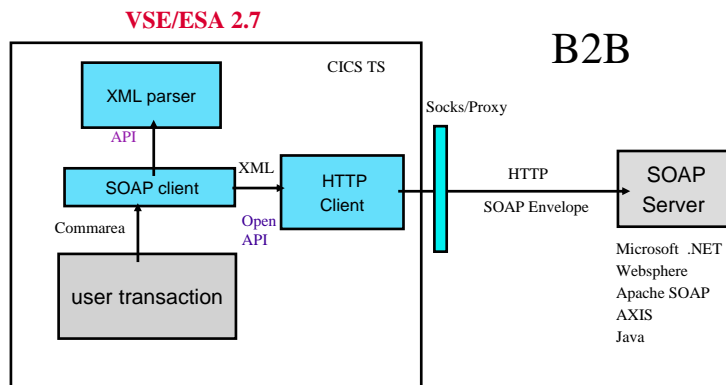
SOAP - Simple Object Access Protocol
(platform independent remote procedure call)



VSE/ESA 2.7 as SOAP client

Web Services (SOAP)

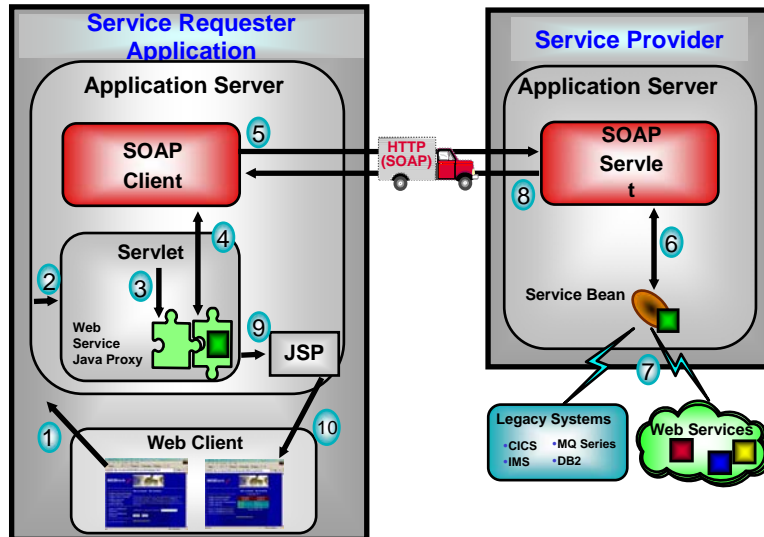
SOAP - Simple Object Access Protocol
(platform independent remote procedure call)



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



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

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

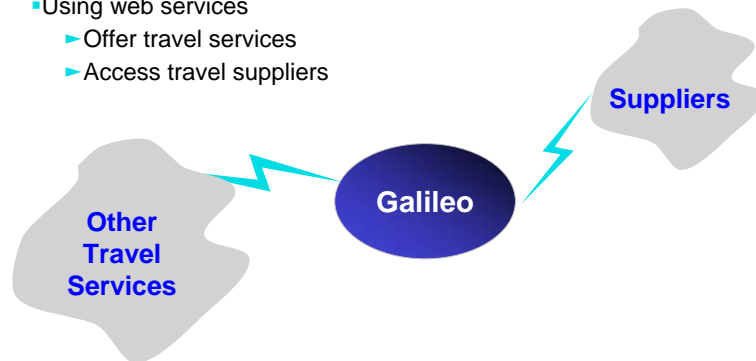
How Can Web Services Be Used?

- Between businesses
 - ▶ Providing service to your customers
 - ▶ Integrating business processes with your partners and suppliers
- Within a business
 - ▶ Accelerate and reduce the cost of integrati
 - ▶ 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



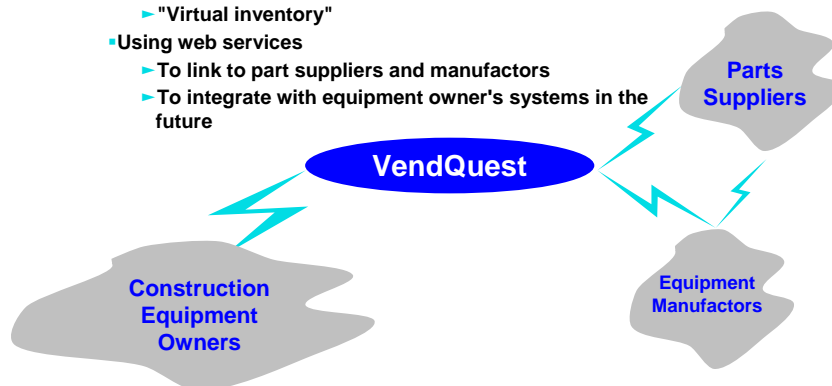
Travel services

- 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



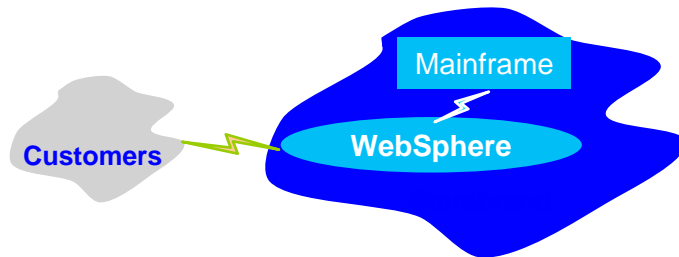
e-Market

- 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

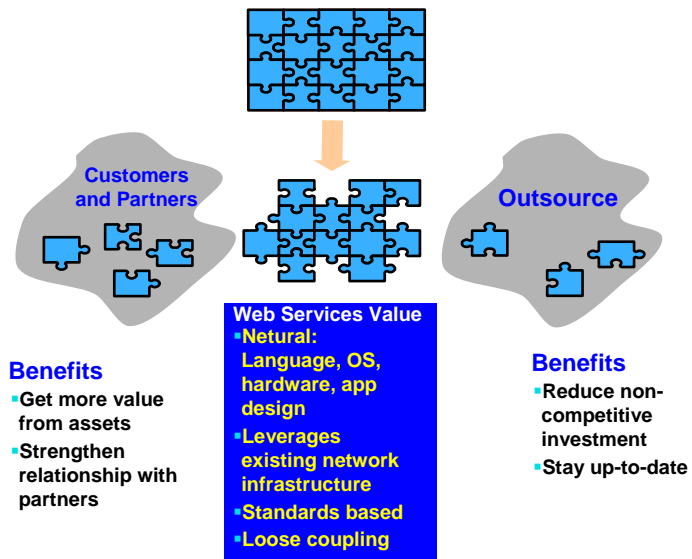


Financial

- 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

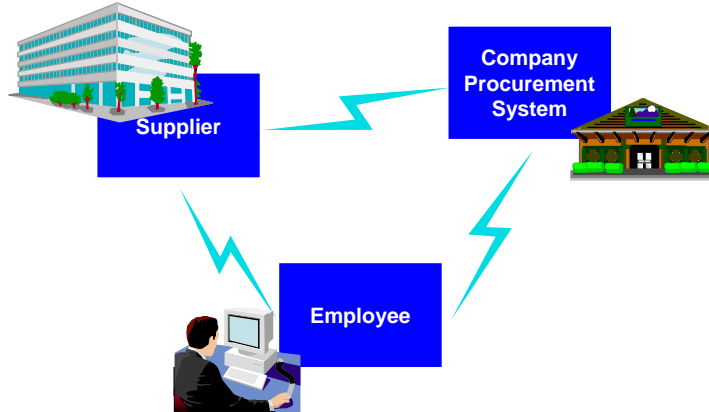


Unbundling IT



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



Complete IBM Software Portfolio for Web Services

WebSphere	Web Services Runtime and Deployment <u>Web Application Server</u> . Includes SOAP and Servlets support for access to UDDI Registries. <u>WebSphere Business Integrator</u> . New software to manage web services workflow within and between enterprises. <u>Web Services Development Environment Toolkit</u> . AlphaWorks free Toolkit for basic Web Service programming tasks: create, deploy and test.
Development	Development and Integration of Web Services Applications <u>WebSphere Studio Application/Enterprise Developer (WSAD / WSED)</u> . It is hosting a complete web application development environment. Includes major Web Services functions such as: <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.

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.

The Magic of Web Services

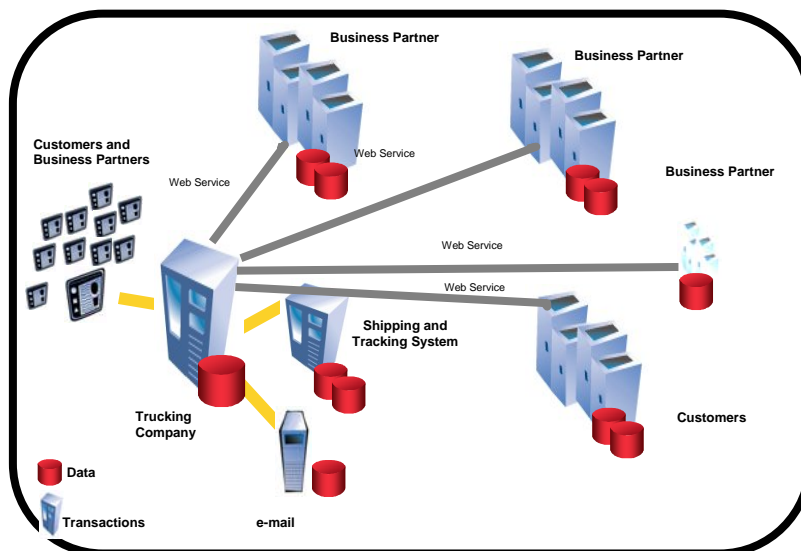
What makes web services different from previous "revolutionary" component and communication models?

- **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

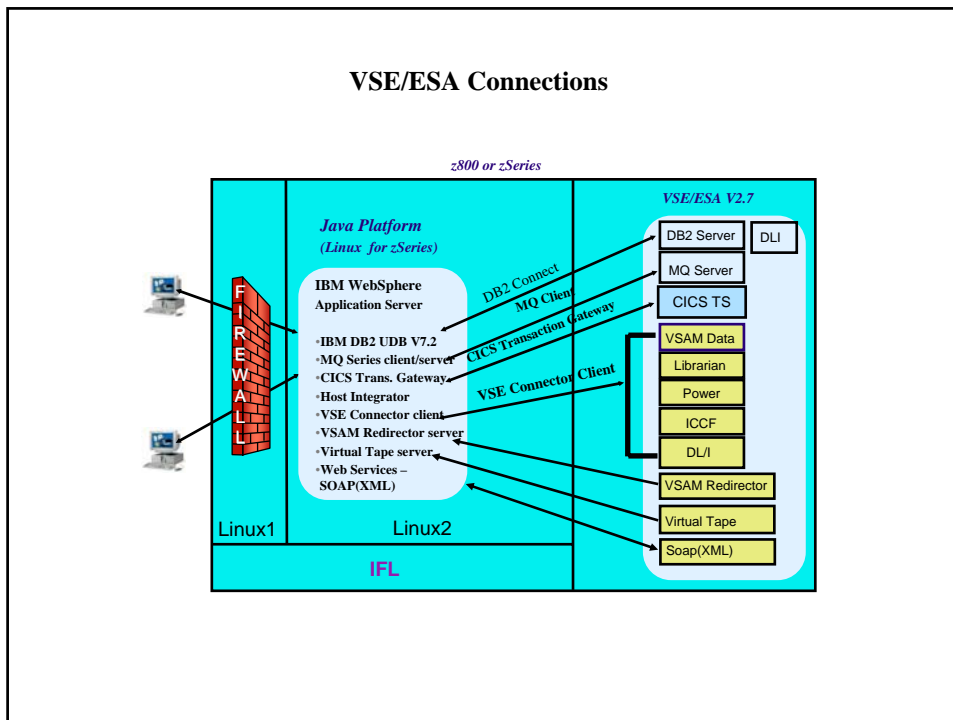
Why will Web Services Technology Succeed?

- **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

Roadmap for dynamic e-business



VSE/ESA Connections



More Information...

<p><u>developerWorks - Web Services Zone</u></p> <p>IBM sponsored site that contains articles, tutorials and latest news related to Web Services.</p>	<p>www-106.ibm.com/developerworks/webservices</p>
<p><u>AlphaWorks - Web Services Toolkit</u></p> <p>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.</p>	<p>www.alphaworks.ibm.com/tech/webservicestoolkit</p>
<p><u>XMethods</u></p> <p>An organization dedicated to promoting the development, deployment, and use of web services.</p>	<p>xmethods.com</p>
<p><u>UDDI Organization</u></p> <p>UDDI Project and Community web site contains UDDI specification, whitepapers, FAQ. IBM is a member.</p>	<p>www.uddi.org</p>
<p><u>SOAP</u></p> <p>The latest version of SOAP specification hosted by the W3C Organization web site.</p>	<p>www.w3.org/TR/SOAP</p>
<p><u>WSDL</u></p> <p>The latest version of WSDL specification hosted by the W3C Organization web site.</p>	<p>www.w3.org/TR/wsdl</p>