Web Services Overview
Agenda

- Evolution of network computing
- What is Web Services?
- Why Web Services?
- Where is Web Services?
- Web Services Architecture
- Web Services Standards
- Java™ APIs for Web Services
- J2EE as platform of choice for Web Services
- Web Services Tools
- Roadmap and Summary
Evolution of Network Computing
The New Software

- **Developer**
  - X1
  - X10^6

- **Shrink Wrap**

- **Software-as-a-Service**

- **New Service**
  - **Payment**
    - X10^6
  - **User’s Device**
  - **Locator**
  - **Calendar**
  - **Authentication**
History of Distributed Computing
## Platform Evolution

<table>
<thead>
<tr>
<th>Catch Phrase</th>
<th>The Network Is the Computer</th>
<th>Objects</th>
<th>Legacy to the Web</th>
<th>The Computer Is the Network</th>
<th>Network of Embedded Things</th>
<th>Network of Things</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>100s</td>
<td>1,000s</td>
<td>1,000,000s</td>
<td>10,000,000s</td>
<td>100,000,000s</td>
<td>100,000,000s</td>
</tr>
<tr>
<td>Leaf Protocol(s)</td>
<td>X</td>
<td>X</td>
<td>+HTTP (+JVM)</td>
<td>+XML Portal</td>
<td>+RM</td>
<td>Unknown</td>
</tr>
<tr>
<td>Directory(s)</td>
<td>NS, NS+</td>
<td>+CDS</td>
<td>+LDAP(*)</td>
<td>+UDDI</td>
<td></td>
<td>+Jini</td>
</tr>
<tr>
<td>Session</td>
<td>RPC, XDR</td>
<td>+CORBA</td>
<td>+CORBA, RM</td>
<td>+SOAP, XML</td>
<td>+RM/Jini</td>
<td>+?</td>
</tr>
</tbody>
</table>

### Schematic

- [Image of schematic diagram]

*Note: The table includes various protocols and directory servers used in different phases of the platform evolution.*
Communication Patterns

- Client-Server
- 3-Tier
- Web Application
- Web Services
- Hybrid P2P
- Fractal
Communication Patterns: Sun ONE

Context and Identity (LDAP, Policy, Liberty)

XML (UDDI, SOAP)

Web Service

J2EE

J2SE/J2ME

Browser

Web

App

DB

Sys.

Bus.
What is a Web Service?
Web Services Definition by W3C

- A Web service is a **software application**
- identified by a **URI**, 
- whose **interfaces and binding** are capable of being defined, described and discovered by **XML artifacts** and
- supports direct **interactions** with other software applications
- using **XML based messages**
- via **internet-based protocols**
Distributed Computing Evolution

- **Client-Server (C/S) silos**
- **Web-based computing**
- **Web Services/Peer-to-Peer**
Traditional C/S vs. Web Services

<table>
<thead>
<tr>
<th>Traditional C/S</th>
<th>Web Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Within enterprise</td>
<td>• Between enterprises</td>
</tr>
<tr>
<td>• Tied to a set of programming languages</td>
<td>• Program language independent</td>
</tr>
<tr>
<td>• Procedural</td>
<td>• Message-driven</td>
</tr>
<tr>
<td>• Usually bound to a particular transport</td>
<td>• Easily bound to different transports</td>
</tr>
<tr>
<td>• Tightly-coupled</td>
<td>• Loosely-coupled</td>
</tr>
<tr>
<td>• Efficient processing (space/time)</td>
<td>• Relatively not efficient processing</td>
</tr>
</tbody>
</table>
# Web Application vs. Web Services

<table>
<thead>
<tr>
<th>Web Application</th>
<th>Web Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>● User-to-program interaction</td>
<td>● Program-to-program interaction</td>
</tr>
<tr>
<td>● Static integration of components</td>
<td>● Possibility of dynamic integration of components (in the future)</td>
</tr>
<tr>
<td>● Monolithic service</td>
<td>● Possibility of service aggregation (in the future)</td>
</tr>
</tbody>
</table>
Characteristics of Web Services

- XML based everywhere
- Message-based
- Programming language independent
- Could be dynamically located
- Could be dynamically assembled or aggregated
- Accessed over the internet
- Loosely coupled
- Based on industry standards
Web Services

Service Registry

Service discovery

Service registration

Service invocation

Service delivery

The Service Grid
Service Assembly

- Macro Service
- Business Process Management
- Micro Service
- Micro Service
- Micro Service
- Micro Service
Service Aggregation

User

Stock Service Portal
Input: Symbol
Output: Price, News, Trade

Nasdaq
Input: Symbol
Output: Price

News feed 1
Input: Symbol
Output: News links

News feed n
Input: Symbol
Output: News links

Brokerage 1
Input: Symbol, Price, Qty

Brokerage n
Input: Symbol, Price, Qty
Why Web Services?
Why Web Services?

Web Services:
- Are platform neutral
- Are accessible in a standard way
- Are accessible in an interoperable way
- Use simple and ubiquitous plumbing
- Are relatively cheap
- Simplify enterprise integration
Why Web Services?

• **Interoperable** – Connect across heterogeneous networks using ubiquitous web-based standards

• **Economical** – Recycle components, no installation and tight integration of software

• **Automatic** – No human intervention required even for highly complex transactions

• **Accessible** – Legacy assets & internal apps are exposed and accessible on the web

• **Available** – Services on any device, anywhere, anytime

• **Scalable** – No limits on scope of applications and amount of heterogeneous applications
Web Services Usage Example

“Growing need for a standard lightweight infrastructure for data exchange in e-business applications.”
Impact of Web Services on Software: “Application Dis-Integration”

Monolithic Software

Application
System Software
A Computer

Web Services

System Service
App Service
System Service
App Service
System Service
App Service

The Network
“Portfolio” can be an application, a portal channel, or a web service itself

A web service is accessed programmatically by applications or other web services.

Macro web services – Virtual Systems

Web Services
- Bank Balance
- Stock Position
- Insurance Cash Value

Portfolio
- Net worth
- Stock ticker

News
- Biz News
- World News
“Word processing” can be an application, a capability, or a web service itself.

A web service is accessed programmatically by applications or other web services.
Three Laws of Computing

• Moore's Law
  – Computing power doubles every 18 months

• Gilder's Law
  – Network bandwidth capacity doubles every 12 months

• Metcalfe's Law (Net Effect)
  – Value of network increases exponentially as number of participants increases
Impact on Integration: Trigger the Network Effect

Metcalfe’s Law: The value of the network is proportional to the square of the number of
### Myth: Web Services is a New Concept

- Web services is distributed computing all over again – only now it is based on the web

<table>
<thead>
<tr>
<th>Concept</th>
<th>Distributed Computing ala CORBA / Java</th>
<th>Basic Web Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Description</td>
<td>CORBA IDL, Java interface</td>
<td>WSDL</td>
</tr>
<tr>
<td>RPC support</td>
<td>ORBs, Idl2java compilers, rmic</td>
<td>SOAP, compilers for WSDL</td>
</tr>
<tr>
<td>Service Registry</td>
<td>CORBA naming service, JNDI</td>
<td>UDDI</td>
</tr>
<tr>
<td>Messaging support</td>
<td>CORBA Event/Notification service, JMS</td>
<td>?</td>
</tr>
<tr>
<td>Transaction support</td>
<td>CORBA Transaction service, JTS</td>
<td>?</td>
</tr>
<tr>
<td>Security support</td>
<td>CORBA Security service, Java security</td>
<td>?</td>
</tr>
</tbody>
</table>
Other Popular Myths Surrounding Web Services

- Web services require only SOAP, WSDL, UDDI: We need more high-level semantics
- Web services are based on the RPC paradigm: Document-driven model would be more popular communication model
- Web services must be based on HTTP: Other transports such as SMTP can be also used
Where is & Where is Web Services going?
Myths about Web Services

- Web Services cure cancer: Not for a very very very long time!
- Web Services are something completely new: Not True!
- You have to write Web Services from scratch: Not True!
- J2EE Platform does not support web services: Not True!
State of Web Services

- Technology/Standards are still evolving
  - SOAP, WSDL, UDDI are not enough
- Business web services is the next big thing, but more works are needed in
  - Quality of Service, management
  - Security, transaction, state and user context
  - Work flow, Identity management,
  - Provisioning, Accounting
- Will be adopted in phases
Web Services Adoption Phases

• 1<sup>st</sup> phase (current state)
  – Concerted deployment internally within an organization mainly for interoperability
  – SOAP over HTTP/S

• 2<sup>nd</sup> phase (1 to 2 years)
  – Selective and non-aggregate deployment with trusted outside business partners
  – Private registry deployment

• 3<sup>rd</sup> phase (at least 3 to 4 years away)
  – Wider, more dynamic and aggregate deployment with outside business partners
  – Public registry deployment
Web Services Adoption Phases

- **1\(^{st}\) Phase – Simple Web Services (Now)**
  - Consumer-focused, stateless, SOAP over HTTP/S

- **2\(^{nd}\) Phase – EAI Web Services (Begun)**
  - Deployed within organization boundaries to enable internal integration

- **3\(^{rd}\) Phase – Business Web Services (2007?)**
  - Deployed on extranets to enable business transactions with trading partners, suppliers,
Business Web Services

- J2EE™
  - Service implementation platform standard
- ebXML and UBL
  - Business web services standards
  - More than 16 vendors and several open source projects support ebXML
  - ex) Australian gas industry uses ebXML NOW!
- Liberty Project
  - Identity system standard
Business Web Services (B2B)
Architectural Components (ebXML)

• B2B collaboration
• Secure and reliable message delivery
• Non-repudiation
• Partner profile
• Repository for business data objects
## Simple Web Services (WUS) vs. B2B Collaboration (ebXML)

**Simple Web Services**
- Simple interaction
- Consumer oriented
- Short-living process
- No business collaboration
- No partner profile
- Not secure, not reliable
- Does not support non-repudiation
- No repository support
- No legal binding

**B2B Collaboration**
- Complex interaction
- Business oriented
- Long-running process
- Supports business collaboration
- Supports partner profile
- Secure and reliable and non-repudiation
- Supports non-repudiation
- Registry and repository support
- Supports legal binding
EAI vs. B2B Collaboration (ebXML)

**EAI**
- Within a business organization
- Centralized control
- Implicit contract
- Small number of business processes and participants

**B2B Collaboration**
- Between business organizations
- Distributed control
- Explicit contract
- Potentially large number of business processes and participants
Trends Towards Service Orientation

- Evolution of EAI to web service standards
- XML RPC => Asynchronous XML Messaging
- Towards de-centralization
- Componentized services
  - Composable and composite services
  - Data encapsulated within component
  - Data ownership follows component ownership
- Brokered web services
- Flexible relationships => Adaptive businesses
Simple Web Services Architectural Components (WUS)

- Service Description
- Service Registration (Publication) and Discovery
- Service Invocation
Core Web Services
Standards
(Simplified) Web Service Architecture

1. Service Registers
   PUBLISH

2. Client Request Service Location
   FIND

3. Client calls Service
   BIND

Registry

Web Service

Service Client
SOAP
(Simple Object Access Protocol)
**SOAP**

- Simple Object Access Protocol
- Wire protocol similar to
  - IIOP for CORBA
  - JRMP for RMI
- **XML** is used for **data encoding**
  - “text” based protocol vs. “binary” protocol
- Supports XML-based RPC
What SOAP is Not

- **Not** a component model
  - So it will **not** replace objects and components, i.e. EJB, JavaBeans

- **Not** a programming language
  - So it will **not** replace Java

- **Not** a solution for **all**
  - So it will **not** replace other distributed computing technologies such as RMI
What does SOAP Define?

- Message Envelope
- Encoding Rules
- RPC Convention
- Binding with underlying protocols
SOAP Message Format

- **SOAP Message**
  - Primary MIME part (text/xml)
  - Attachment
  - Attachment
  - Attachment

- **SOAP Envelope**
  - **SOAP Header**
    - Header Entry
    - Header Entry
  - **SOAP Body**
    - Body Entry
    - Body Entry
SOAP Message Envelope

- Encoding information
- Header
  - Optional
  - *Could* contain context knowledge
    - Security
    - Transaction

- Body
  - RPC methods and parameters
  - Contains application data
SOAP Encoding

• Rules of expressing application-defined data types in XML
• Based on W3C XML Schema
• Simple values
  – Built-in types from XML Schema, Part 2 (simple types, enumerations, arrays of bytes)
• Compound values
  – Structs, arrays, complex types
What is WSDL?

- XML language for describing web services
- Web service is described as
  - A set of communication endpoints (ports)
- Endpoint is made of two parts
  - Abstract definitions of operations and messages
  - Concrete binding to networking protocol (and corresponding endpoint address) and message format
- Why this separation?
  - Enhance reusability (as we will see in UDDI reference to WSDL document)
Why WSDL?

• Enables **automation** of communication details between communicating partners
  – Machines can read WSDL
  – Machines can invoke a service defined in WSDL
• Discoverable through registry
• Arbitration
  – 3rd party can verify if communication conforms to WSDL
WSDL Document Example

- Simple service providing stock quotes
- A single operation called `GetLastTradePrice`
- Deployed using SOAP 1.1 over HTTP
- Request takes a ticker symbol of type `string`
- Response returns price as a `float`
UDDI
UDDI defines a way to **publish** and **find** information about Web services.
UDDI (Universal Description, Discovery and Integration)

• “White pages”
  – address, contact, and known identifiers

• “Yellow pages”
  – industrial categorizations
    • Industry: NAICS (Industry codes - US Govt.)
    • Product/Services: UN/SPSC (ECMA)
    • Location: Geographical taxonomy

• “Green pages”
  – technical information about services
Web Services Framework for J2EE
J2EE Platform & Web Services

- B2B Applications
- B2C Applications
- Web Services
- Wireless Applications

Application Server

Existing Applications

Enterprise Information Systems
Why J2EE for Web Services?

- Web services is just **one of many service delivery channels** of J2EE
  - No architectural change is needed
  - Existing J2EE components can be easily exposed as Web services
- Many **benefits** of J2EE are preserved for Web services
  - Portability, Scalability, Reliability
  - No single-vendor lock-in
Where Are We Now?

- **Java APIs** for Web Services are being developed very rapidly
- **Tools** are available now for exposing existing J2EE components as Web services
- **J2EE community has defined overall framework** for Web Services (J2EE 1.4, JSR 109)
Design Goals J2EE Web Services Framework

- **Portability** of Web services component
  - Over different vendor platform
  - Over different operational environment
- **Leveraging** existing J2EE **programming models** for service implementation
- **Easy** to program and deploy
  - High-level Java APIs
  - Use existing deployment model
J2EE Web Services Framework

• J2EE 1.4 and Java EE 5
  – Umbrella framework for Web services
  – JSR 109, JAX-RPC, JAXR, EJB 2.1, Servlet 2.4,

• JAX-RPC (JAX-WS)
  – Defines client programming model
  – Defines Servlet-based Web services endpoint model
J2EE Web Services Framework

- **EJB 2.1**
  - Defines Stateless Session Bean-based Web services endpoint model

- **Servlet 2.4**
  - Will be aligned with JAX-RPC

- **JSR 109**
  - Defines standard Web services packaging and deployment model
Web Services Architecture over J2EE
What Is a Web Service?

• A set of endpoints (ports) operating on messages
• Ports are operating within a container
  – Container provides runtime environment
  – Contract for runtime environment are specified in JAX-RPC, EJB 2.1, JSR 109
• Service is described in WSDL document and published to a registry
  – WSDL specifies a contract between service provider and client
Web Service Component and Container

• Container vs. Component model
  – Web services components get executed within a container
  – Components are portable (under J2EE 1.4)

• Web service components
  – Web-tier (Servlet-based endpoint)
  – EJB-tier (Stateless session bean-based endpoint)
Web Service Components
Summary
Summary

- Web services provides a new paradigm for program to program communication
- Comprehensive set of Java APIs for Web Services are now available!
- J2EE is the platform of choice for Web services
Passion!