

# Introduction to Web Services

SWE 642, Spring 2008  
Nick Duan

April 9, 2008

1

## Overview

- What are Web Services?
  - A brief history of WS
  - Basic components of WS
  - Advantages of using WS in Web application development
- Simple Object Access Protocol (SOAP)
  - Concept of SOAP
  - SOAP structures and usages
- Web Services Definition Language (WSDL)
  - Concept of WSDL
  - WSDL structures and usages
  - WSDL-driven WS development
- Summary

April 9, 2008

Nick Duan

2

# What are Web Services?

**W3C Consortium:** A Web service is a software application identified by a URI, whose interface and bindings are capable of being identified, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols

**In General:** Web Services are Web applications based on Web Services standards on top of the common Web infrastructure

**History:** originated from Distributed Object Computing

8/27/2008

Nick Duan

3

# Basic Web Services Standards

- XML Standards
  - XML schema, XML namespaces, XML Security, etc.
  - W3C standards
- Simple Object Access Protocol (SOAP)
  - Application-level protocol defined on top of HTTP or other text-based protocols
  - A W3C standard
- Web Services Definition Language (WSDL)
  - An XML derivative for defining web services interfaces, including message/service types, and end points
  - A W3C standard
- Universal Discovery, Description, and Integration (UDDI)
  - A set of XML schema definitions for defining a registry service structure along with a set of WSDL interfaces for interacting with the registry service
  - An OASIS standard

April 9, 2008

Nick Duan

4

## Additional Standards

- Additional WS Standards for extending application-specific functions using WS
  - WS-Addressing (W3C)
  - WS-Security (OASIS)
  - WS-Policy (W3C)
  - WS-Notification (OASIS)
  - WS-Eventing (W3C)
  - WS-\*
- Programming Standards
  - Java – JAX-WS, JAX-RPC
  - Microsoft - .Net

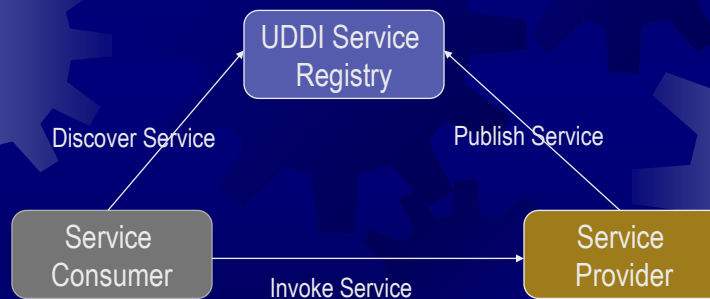
8/27/2008

Nick Duan

5

## Web Service Architecture

### ■ Web service interactions



8/27/2008

Nick Duan

6

# What is SOAP?

SOAP (Simple Object Access Protocol) – a light weight distributed computing protocol that allows information to be exchanged in a decentralized, distributed environment.

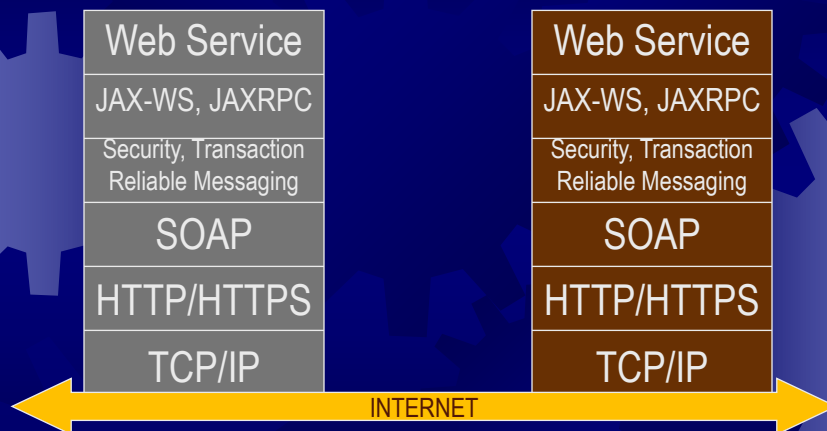
- Platform, programming language independent (HTTP, XML)
- Lightweight
- Text-based XML protocol
- The conventions for representing errors

8/27/2008

Nick Duan

7

# SOAP and WS Communication Stack



8/27/2008

Nick Duan

8

# SOAP Message Structure

HTTP  
Message

SOAP Envelope(required)

SOAP Header(optional)

SOAP Body(required)

Fault(optional)

8/27/2008

Nick Duan

9

# SOAP Envelope

- Wraps around the message content within a HTTP message body
- Root element of a SOAP message
- Must contain the namespace declaration for the SOAP message
- Must contain the encodingStyle attribute (with a URL as the attribute value) to specify encoding of the message. There is no default encoding style
- Contains an optional SOAP header element and a mandatory SOAP body element as sub-elements

8/27/2008

Nick Duan

10

# SOAP Example in HTTP

POST /InStock HTTP/1.1 Host: [www.example.org](http://www.example.org)  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: 578

```
<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>Cows</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

April 9, 2008

Nick Duan

11

# SOAP Header

- Optional element under SOAP envelop for specifying application-level requirements
- Contains one or more sub-elements. Each sub-element may contain one or more of the three standard attributes
  - *Actor* attribute  
Used to associate the header element to a specific end point defined as an URL. Example:  
`actor="http://www.service.com/myservice"`
  - *MustUnderstand* attribute  
Specify whether the recipient must to process the header element or not
    - 1, the recipient of the header must process the header, or the processing of the message must fail
    - 0, the recipient doesn't have to understand the header in order to process the message
  - *encodingStyle* attribute (same format and purpose as for envelop)
- Provide additional application information such as security to support service interaction
  - WS-Security is defined using elements in the SOAP header

8/27/2008

Nick Duan

12

# SOAP Header Example

- OASIS WS-Security standard:

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
  xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
  <S:Header> ...
    <wsse:Security mustUnderstand="1">
      <wsse:UsernameToken>
        <wsse:Username>John</wsse:Username>
        <wsse:Password>Smith</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security> ...
  </S:Header> ...
</S:Envelope>
```

April 9, 2008

Nick Duan

13

# SOAP Body

- Contains the information that must be received by the recipient of the message.
- All elements in the body must have qualified namespaces
- Example

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap=http://www.w3.org/2001/12/soap-envelope
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:Body>
    <m:GetPrice xmlns:m="http://www.b2bconnect.com/prices">
      <m:Item>Dell Precision M4300</m:Item>
    </m:GetPrice>
  </soap:Body>
</soap:Envelope>
```

8/27/2008

Nick Duan

14

# SOAP Fault

- For error handling within a SOAP application
- Contains the following sub-elements
  - `<faultCode>` – general class of errors
    - Predefined code values as follows:
      - VersionMismatch
      - MustUnderstand
      - Client
      - Server
    - `<faultString>` - fault code explanation
    - `<faultActor>` – URI of the fault source
    - `<detail>` – detailed description of the fault specific to the application

8/27/2008

Nick Duan

15

# SOAP Fault Example

```
<SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Client</faultcode>
  <faultstring>Message does not have necessary
  info
</faultstring>
  <faultactor>http://gizmos.com/order</faultactor>
  <detail> <PO:order
  xmlns:PO="http://gizmos.com/orders/"> Quantity
  element does not have a value</PO:order>
  <PO:confirmation
  xmlns:PO="http://gizmos.com/confirm"> Incomplete
  address: no zip code </PO:confirmation>
</detail>
</SOAP-ENV:Fault>
```

April 9, 2008

Nick Duan

16



# SOAP Encoding

- Adopted XML data types (<http://www.w3.org/TR/xmlschema-2>)
- Simple types
  - 44 built-in simple types
  - No element children
  - No attribute
- Compound types
  - struct
  - array

8/27/2008

Nick Duan

17

# WSDL

- Web Services Description Language (WSDL) is a specification for describing web services in XML
- A WSDL document contains the followings
  - Data type
  - Input and output parameters of a service
  - Relationship between input and output parameter
  - Logical grouping of operations
  - Protocol to be used for accessing an object's methods
  - Address of the service

8/27/2008

Nick Duan

18

# WSDL Document Structure

## <wsdl:definitions>

<types>: XML Data types used

<message>: Messages that are used by the service

<portType>: operations that are supported

<binding>: Description of network protocol for invocation

<service>: Reference to actual location of service

8/27/2008

Nick Duan

19

# WSDL Document Structure

- The <definitions> element
  - Name of the web service
  - Defining the namespaces used in the WSDL document
- The <types> element
  - Describe all the data types used between the client and server.
  - W3C XML Schema
  - Optional if all types are XML Schema primitive types

8/27/2008

Nick Duan

20

## WSDL Document Structure

- The <message> element
  - Conceptual unit of data that may be exchanged
  - Request
  - Response
  - <part> element within <message> refer to type elements

8/27/2008

Nick Duan

21

## WSDL Document Structure

- The <portType> element
  - Combine multiple <message> elements together to form different types of operations
  - <operation> elements
    - One-way
    - Request-response
    - Solicit-response
    - Notification

8/27/2008

Nick Duan

22

# WSDL Document Structure

- The <binding> element
  - Describe the mechanism that is used by a service to communicate with a client
  - SOAP binding
  - HTTP binding
  - MIME binding
- The <service> element
  - Define the address for invoking the specified service

8/27/2008

Nick Duan

23

# Online References

- SOAP
  - <http://www.w3.org/2000/xml/Group/>
- WSDL
  - <http://www.w3.org/2002/ws/desc/>
- WSDL Tutorial
  - [http://www.w3schools.com/WSDL/wSDL\\_ uddi.asp](http://www.w3schools.com/WSDL/wSDL_uddi.asp)
- Java EE Tutorial
  - <http://java.sun.com/javaee/5/docs/tutorial/doc/>
- Java Web Service Architecture, by James McGovern et. al., Morgan Kaufmann, 2004

April 9, 2008

Nick Duan

24

# Summary

- Web Services is the industry standard for implementing dynamic, **programmable** web applications with XML-based APIs
  - WS applications are client/server applications using SOAP over HTTP
  - The XML-based APIs of WS are defined using standard WSDL
  - WS development is interface-driven! (define the interfaces first before implementing the services. This will be elaborated in the next class)