Software Architecture course

Service-Oriented Architecture (SOA)

By Võ Văn Hi
Faculty of Information Technologies
Industrial University of Ho Chi Minh City

Session objectives

SOA Introduction
Web Services Introduction
XML
Simple Object Access Protocol
WSDL
Web Service Registry
REST

Problems Addressed by a Service Oriented Architecture

I want to make sense out of this. How do I do that?
What is a Service Oriented Architecture (SOA)?

- SOA is a method of design, deployment, and management of both applications and the software infrastructure where:
  - All software is organized into business services that are network accessible and executable.
  - Service interfaces are based on public standards for interoperability.

Key Characteristics of SOA

- Quality of service, security and performance are specified.
- Software infrastructure is responsible for managing.
- Services are cataloged and discoverable.
- Data are cataloged and discoverable.
- Protocols use only industry standards.
What is a “Service”?

- A Service is a reusable component.
- A Service changes business data from one state to another.
- A Service is the only way how data is accessed.
- If you can describe a component in WSDL, it is a Service.

How to View Organizing for SOA

SOA Must Reflect Timing

Organization of Infrastructure Services
Data Concepts

- Data Element Definition
  - “Text associated with a unique data element within a data dictionary that describes the data element, give it a specific meaning and differentiates it from other data elements. Definition is precise, concise, non-circular, and unambiguous.” (ISO/IEC 11179 Metadata Registry specification)

- Data Element Registry
  - “A label kept by a registration authority that describes a unique meaning and representation of data elements, including registration identifiers, definitions, names, value domains, syntax, ontology and metadata attributes.” (ISO 11179-1)
Security Services = Information Assurance

- Conduct Attack/Event Response
  - Ensure timely detection and appropriate response to attacks.
  - Manage measures required to minimize the network's vulnerability.
- Secure Information Exchanges
  - Secure information exchanges that occur on the network with a level of protection that is matched to the risk of compromise.
- Provide Authorization and Non-Repudiation Services
  - Identify and confirm a user’s authorization to access the network.

Computing Services

- Provide Adaptable Hosting Environments
  - Global facilities for hosting to the "edge".
  - Virtual environments for data centers.
- Distributed Computing Infrastructure
  - Data storage, and shared spaces for information sharing.
- Shared Computing Infrastructure Resources
  - Access shared resources regardless of access device.

Organization of Computing Services

- Computing Services
- Resource Planning
- Control & Quality
- Configuration Services
- Financial Management

Organization of Communication Services

- Communication Services
- Interoperability Services
- Spectrum Management
- Connectivity Arrangements
- Continuity of Services
- Resource Management
Network Services Implementation

- From point-to-point communications (push communications) to network-centric processes (pull communications).
- Data posted to shared space for retrieval.
- Network controls assure data synchronization and access security.

Communication Services

- Provide Information Transport
  - Transport information, data and services anywhere.
  - Ensures transport between end-user devices and servers.
  - Expand the infrastructure for on-demand capacity.

Organization of Application Services

A Few Key SOA Protocols

- Universal Description, Discovery, and Integration, **UDDI**. Defines the publication and discovery of web service implementations.
- The Web Services Description Language, **WSDL**, is an XML-based language that defines Web Services.
- **SOAP** is the Service Oriented Architecture Protocol. It is a key SOA in which a network node (the client) sends a request to another node (the server).
- The Lightweight Directory Access Protocol, or **LDAP** is protocol for querying and modifying directory services.
- Extract, Transform, and Load, **ETL**, is a process of moving data from a legacy system and loading it into a SOA application.
Components of SOA

SOA was implemented by Web Service

- **Service Provider (Create WS)**
- **Service Requestor**
- **Service Broker**
- **Service Registry (UDDI)**

**Publish**

**Bind/Communication**

**Find/Look up**

SOA implementation by WS

- WS, similar to the SOA, enable location and implementation transparency by using XML for communication

Web service definition

**src: wikipedia**

- A Web service is a method of communication between two electronic devices over a network. It is a software function provided at a network address over the Web with the service always on as in the concept of utility computing. The W3C defines a web service generally as:
  - a software system designed to support interoperable machine-to-machine interaction over a network.[11]
  - The W3C Web Services Architecture Working Group [1] defined a Web Services Architecture, requiring a specific implementation of a "Web service." In this:
    - A Web service has an interface described in a machine-processable format (specifically, WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP (Simple Object Access Protocol) messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.[12]
    - The W3C also states:
      - We can identify two major classes of Web services:
        - RESTful Web services, in which the primary purpose of the service is to manipulate representations of Web resources using a uniform set of stateless operations.
        - Simple Web services, in which the service may expose an arbitrary set of operations.[13]
  
Web service definition (cont.)

- To summarize, a complete web service is, therefore, any service that:
  - Is available over the Internet or private (intranet) networks
  - Uses a standardized XML messaging system
  - Is not tied to any one operating system or programming language
  - Is self-describing via a common XML grammar
  - Is discoverable via a simple find mechanism
How Does a Web Service Work?

- A web service enables communication among various applications by using open standards such as HTML, XML, WSDL, and SOAP. A web service takes the help of:
  - XML to tag the data
  - SOAP to transfer a message
  - WSDL to describe the availability of service.
- You can build a Java-based web service on Solaris that is accessible from your Visual Basic program that runs on Windows.
- You can also use C# to build new web services on Windows that can be invoked from your web application that is based on JavaServer Pages (JSP) and runs on Linux.

Characteristics of Web services

- XML-Based
- Loosely Coupled
- Coarse-Grained
- Ability to be Synchronous or Asynchronous
- Supports Remote Procedure Calls (RPCs)
- Supports Document Exchange

Uses of Web Services

- Web services are basically used in Application-to-Application (A2A) Integration. A2A integration is also known as Enterprise Application Integration (EAI)
- When different applications belonging to multiple organizations, typically business partners, exchange data using Web Services it is called Business-to-Business (B2B) communication

Architecture of a web services

- Server Request
- Server Response
- Listener
- Web Service Facade
- Web Service Logic
- Data Access
- Data
- Web Service
Web Service Roles

There are three major roles within the web service architecture:

Service Provider
This is the provider of the web service. The service provider implements the service and makes it available on the Internet.

Service Requestor
This is any consumer of the web service. The requestor utilizes an existing web service by opening a network connection and sending an XML request.

Service Registry
This is a logically centralized directory of services. The registry provides a central place where developers can publish new services or find existing ones. It therefore serves as a centralized clearing house for companies and their services.

Web Service standards

• Web services are a set of specifications formulated and accepted by the leading enterprises that provide or avail Web services.
• Various Web services standards are:
  o XML: Represents data in a standard format
  o SOAP: Common, extensible, message format
  o WSDL: Common, extensible, service description language
  o UDDI: Maintains registries storing information about service providers and their services

Life cycle of web services

Create Web Service
Define Service Interface & Invocation Methods
Publish Web Service on Internet or Intranet
Web users should find Web Service to use
Invoke Web Service to be used by web users
Unpublish Web Service if not needed
Information exchange approach(1)

• Electronic Data Interchange (EDI)
  o Is a technique used by business partners to exchange business documents that included purchase orders, invoices, shipping notification, financial payments, and so on.
  o To send an EDI document
    • Install translation software on your system that is used to convert business documents into X12 format.
    • Next, you set up a private wide area network to send and receive the documents.
    • The same process is repeated at the receiver's end.
  • Drawbacks:
    – Cost involved in setting up private wide area networks was too high.
    – Bus partners had to buy proprietary software for transmission of messages from their system to private network.
    – Each business partner had to buy proprietary software to translate business document to X12 format.

Information exchange approach(2)

• Remote Procedure Call (RPC)
  o Involved invoking remote methods that allowed information exchange in the form of parameters and returned values
  o Approach standardized the communicate protocol and eliminated the need of private networks.
  o Drawbacks:
    • Several vendors came up with RPC-based technologies, such as CORBA, RMI, and DCOM.
    • Communication between distributed systems required relaxation of security features.
SOAP

- Provides interoperability between applications by using XML & HTTP
- Uses HTTP protocol to transfer messages between applications. HTTP is recognized by all browsers, proxies, firewalls and servers. Furthermore, it uses port 80 to transmit data.
- Advantages:
  - Vendor Neutral.
  - Transport Protocol Independent.
  - Platform Independent.
  - Language Independent.
  - Interoperability
  - Simple

SOAP (con’t)

- SOAP message is an XML document
- It is used to exchange data between applications.
- Structure:

SOAP Message

- SOAP message is an XML document
- It is used to exchange data between applications.
- Structure:
HTTP Header

HTTP method used: Post /iemfamily/CreditCard.asmx HTTP/1.1
Host: www.iemfamily.com
Content-Type: text/xml; charset=utf-8
Content-Length: length

This is a placeholder and holds the length of the document.

Encoding used

SOAP Header element

```xml
<SOAP-ENV:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header>
    <wssec:Security>
      <wssec:Token>
        <wssec:PropList>
          <wssec:Prop>
            <wssec:Name>UsernameToken</wssec:Name>
            <wssec:Value>username</wssec:Value>
          </wssec:Prop>
        </wssec:PropList>
      </wssec:Token>
    </wssec:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <![CDATA[
   ]]>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Body element

- Contains application-specific data to be exchanged between applications as parameters to a method call
- Is the mandatory element of Envelope element
- The immediate child elements of Body element must be namespace-qualified
- If the Header element is not present, then Body element should be the immediate child of Envelope element. However, in the presence of Header element, the Body element should immediately follow the Header element.

SOAP Request Envelope

```xml
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <![CDATA[
   ]]>
  </soapenv:Body>
</soapenv:Envelope>
```
SOAP Response Envelope

```xml
<S:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:a2="http://com.vvh/"
>
  <s:Body>
    <s:Response
      xmlns:a2="http://com.vvh/"
    >
      <return>7.0</return>
    </s:Response>
  </s:Body>
</S:Envelope>
```

SOAP with Attachments(1)

- Need of attachment
  - SOAP does not allow binary data such as images in the message
  - Messages that require binary data are converted to a **Multipurpose Internet Mail Extensions** (MIME) message format and then sent.
  - A MIME message can contain multiple parts and supports binary data as well.

SOAP with Attachments(2)

```
public class CreateSOAPMessage {
  public static void main(String[] args) throws Exception {
    System.out.println("Hello SOAP message from server");
    MessageFactory msgFactory = MessageFactory.newInstance();
    SOAPMessage message = msgFactory.createMessage();
    message.addSOAPHeader().addTextNode("hello header");
    SOAPBody body = message.createSOAPBody();
    SOAPElement calc = body.addChildElement("calc");
    calc.setTextValue("calc.addchildelement("a")");
    calc.addChildNode("b");
    calc.addNode("c");
    message.writeTo(System.out);
    System.out.println("XML SOAP message from server");
    SOAPConnectionFactory sfc = SOAPConnectionFactory.newInstance();
    SOAPConnection connection = sfc.createConnection();
    SOAPMessage response = connection.call(message, endpoint);
    SOAPPart part = response.getSOAPBody();
    Source src = part.getContent();
    Transformer transformer = TransformerFactory.newInstance().newTransformer();
    transformer.transform(src, new StreamResult(System.out));
    System.out.println("Finish");
  }
}
```
SOAP-HTTP binding

- HTTP is a standard protocol used worldwide to transfer data over the Web.
- SOAP was designed keeping in mind the HTTP protocol.
- SOAP messages are transmitted as a payload of an HTTP message that contains form data such as username, password, credit card number, and so on.
- HTTP is a request-response protocol.

SOAP request over HTTP

```xml
POST /orderstatus HTTP/1.1
Host: www.flings.com:80
Content-Type: text/xml; charset=utf-8
Content-Length: 482
SOAPAction: "https://www.flings.com/books/getOrderStatus"

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope

...<SOAP-ENV:Body>
<ns1:GetOrderStatus>
xmlns:ns1="http://www.flings.com/methode"></ns1:GetOrderStatus>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
SOAP response over HTTP

HTTP/1.1 200 OK
Connection: close
Content-Length: 689
Content-Type: text/xml; charset=utf-8

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   ...
   <SOAP-ENV:Body>
      <w:GetOrderStatusResponse
         xmlns:w="http://www.Ellangi.com/method">
         <w:OrderStatus>
            Shipped on 2007-08-09
         </w:OrderStatus>
      </w:GetOrderStatusResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

Fault Element

- A SOAP fault is analogous to a Java exception in that it's generated when an error occurs.

  - faultcode: Used by Web service consumer to identify the fault
  - faultstring: Used to provide description of the fault
  - faultactor: Used to identify the source of the fault through URI
  - detail: Holds application-specific error information

Web Service Description Language

- Service description language – SDL
  - Specifies functionality of service
  - Lists the parameters required by the exposed method
  - Lists parameters returned by methods
  - Documents the service contract (messages that the Web Service will generate and accept)
Web service description language

- XML file describing Web services acts as a contract between the Web service consumer and the Web service provider
- Location of Web Service
- Type of protocol to be used
- Allow specifying the way in which a service can be accessed
- Use to explain how a particular Web Service works. It provides businesses with a standardized way of describing how applications and processes can interact online.
- A WS proxy supports communication

Web Service Description Language

- Network Accessible Service Specification Language (NASSL)
- Well Defined Service Document (WDS)

XML-based Interface Definition Language used to specify operational information for a Web Service.

Specifies non-operational information for a service: service category, expiry date, company name etc.

WSDL for service providers & consumers

1. Publish Web Service on registry
2. Find a suitable service
3. Retrieve WSDL document
4. Invoke methods of Web Service

WSDL Terminologies (1)

SOAP Operation
- SOAP Request
- SOAP Response

HTTP Request
- HTTP Response
- HTTP Operation

Web Service
WSDL Terminologies (2)

- **message**: Abstract definition of the data being communicated
- **types**: Mechanism to define data type used in Web service
- **operation**: Abstract definition of a particular action supported by the Web service
- **port**: An end-point described by the combination of a network address and binding
- **portType**: Collection of operations
- **binding**: Specification that lists the communication protocol and the data format for a particular port type
- **service**: Defined as a collection of ports

Sample WSDL

http://www.webservicex.net/WeatherForecast.asmx?wsdl

WSDL Elements: definitions

- The `definitions` element is the root component of the WSDL file.
- Defines the name of the Web Service and also one or more namespaces used by its child elements.
WSDL Elements: types

• Describes data types used for exchanging messages by the service.
• Is mandatory only if the data type is other than the built-in data types of XML Schema. Example of XML schema’s built-in types are string, integer and so on.

```xml
<types>
  <s:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
    <s:element name="add">
      <s:complexType>
        <s:sequence>
          .......
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:schema>
</types>
```

WSDL Elements: message

• Describes the content of messages exchanged by applications.
• Specifies the service’s request and response mechanism.
• Not dependent upon any protocol i.e. there is no naming convention for message names.
• Divided into parts and each part is defined by the <part> element.

```xml
<message name="addSoapIn">
  <part name="parameters" element="s0:add" />
</message>
```

WSDL Elements: portType

• portType defines operations provided by the Web Service.
• Operations and input/output messages are to be defined by specifying the <operation> element.

```xml
<portType name="TestWSSoap">
  <operation name="add">
    <input message="s0:addSoapIn" />
    <output message="s0:addSoapOut" />
  </operation>
</portType>
```

WSDL Elements: binding

• Describes how the input and output messages of each operation defined in portType element will be transmitted over the Internet from one application to another.
• Represents the concrete descriptions of the operations.
• Specifies the representation of parameters for a Web method.

```xml
<binding name="TestWSSoap" type="s0:TestWSSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <operation name="add">
    <soap:operation soapAction="http://tempuri.org/add" style="document" />
    <input><soap:body use="literal" /></input>
    <output><soap:body use="literal" /></output>
  </operation>
</binding>
```

WSDL Elements: message
Web Service Registry

UDDI

- Stands for *Universal Description, Discovery and Integration*
- Helps businesses to:
  - Describe the business and the services offered by them
  - Discover other businesses, which may be of some assistance
  - Integrate with other businesses for expansion

Registering and Using Web Services

UDDI Business Registry

Aims at integrating e-commerce sites
Working of UDDI

XML File Submitted to UDDI

White Pages
- Includes address, contact etc.

Yellow Pages
- Industrial Category information

Green Pages
- Technical information about the services

Registry Architecture and Data Structures

White Pages
- businessEntity (bE)

Yellow Pages
- bindingTemplate (bT)

Green Pages
- publisherAssertion (pA)

ebXML

1. Build Legal System Implementation for Business Details
   - Exxon Corporation

2. Register implementation Details, Register Exxon Corporation Profile
   - ebXML Registry

3. Query about Exxon Corporation Profile
   - Siddo Electronics

4. Agree on Business Arrangement

5. Do Business Transactions

ebXML Registry and Repository

Core Component
- Business Process
- Registry and Repository
  - Object Manager
  - Object Query Manager
- Collaboration Protocol Profile
- Collaboration protocol Agreement
- Messaging Service
ebXML Collaboration Protocol Profiles and Collaboration Protocol Agreements

- The CPP is an XML document that contains information about a business and the way exchanges information with other businesses.
- The CPA is an intersection of two CPP documents.
- The CPA is derived from two or more CPPs.

Live Demo

- Java Web Services
- EJB with Web Services
- .Net Web Services
- Communication to each others

What is REST?

- REST stands for REpresentational State Transfer. REST is web standards based architecture and uses HTTP Protocol for data communication. It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. REST was first introduced by Roy Fielding in 2000.
- In REST architecture, a REST Server simply provides access to resources and REST client accesses and presents the resources. Here each resource is identified by URIs/ global IDs. REST uses various representations to represent a resource like text, JSON and XML. Now a days JSON is the most popular format being used in web services.
HTTP Methods

- Following well known HTTP methods are commonly used in REST based architecture.
  - **GET** - Provides a read only access to a resource.
  - **PUT** - Used to create a new resource.
  - **DELETE** - Used to remove a resource.
  - **POST** - Used to update a existing resource or create a new resource.
  - **OPTIONS** - Used to get the supported operations on a resource.

RESTful Web Services

- A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer.
- Web services based on REST Architecture are known as RESTful web services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as JSON and set of HTTP Methods.