

Software Architecture course

# Service-Oriented Architecture (SOA)



By Võ Văn Hải  
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



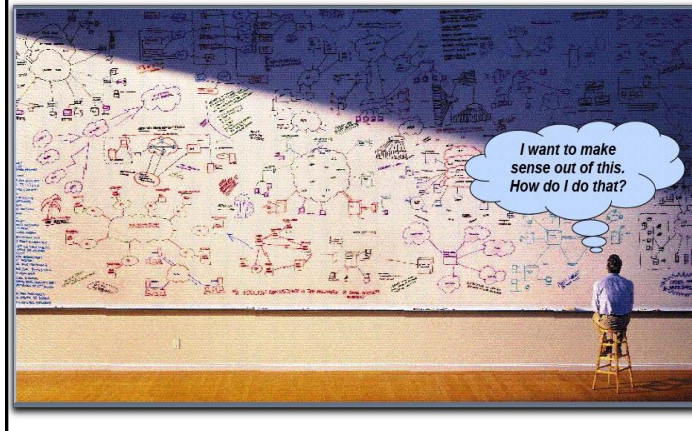
2

## Introduction



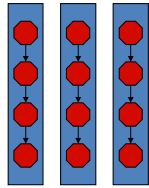
3

## Problems Addressed by a Service Oriented Architecture



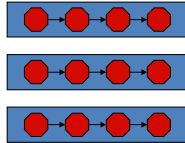
## Directions of System Architecture

1960 - 1980



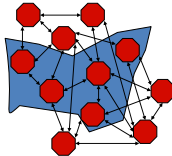
- Organization Focus
- Mainframe Centric
- Internal Use
- Unique Data

1990 - 2000



- Process Focus
- Client Server
- Partial Connectivity
- EDI File Transfer

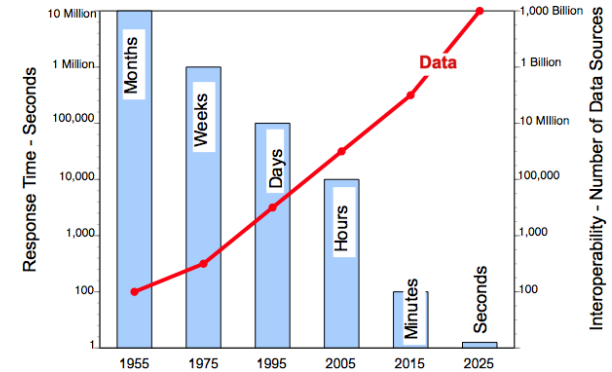
2010 - 2050



- Distributed Functions
- Data Centric
- Universal Interoperability
- Real-time Connectivity

5

## Data Interoperability Expands as Response Time Shrinks



6

## 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.

7

## 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.

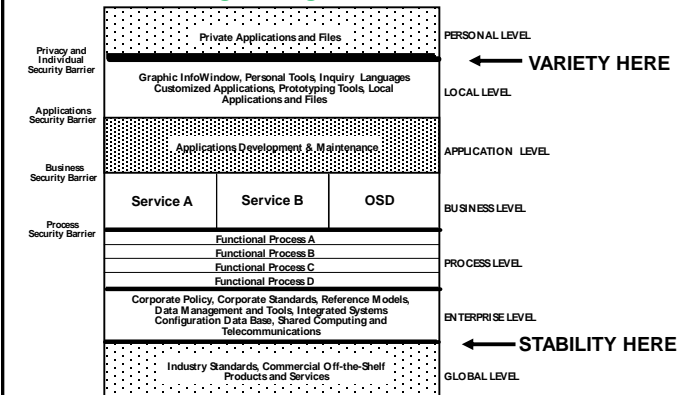
8

## 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.

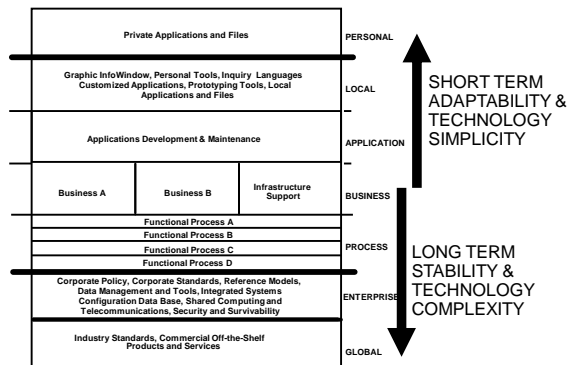
9

## How to View Organizing for SOA



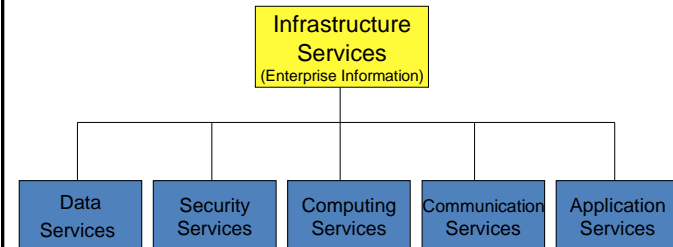
10

## SOA Must Reflect Timing



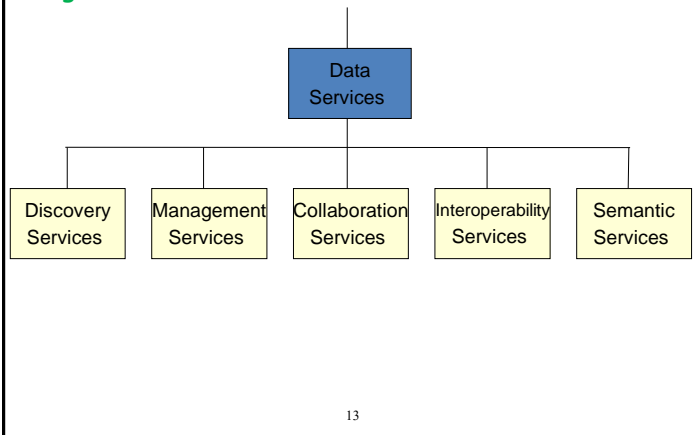
11

## Organization of Infrastructure Services



12

## Organization of Data Services



## What Data?

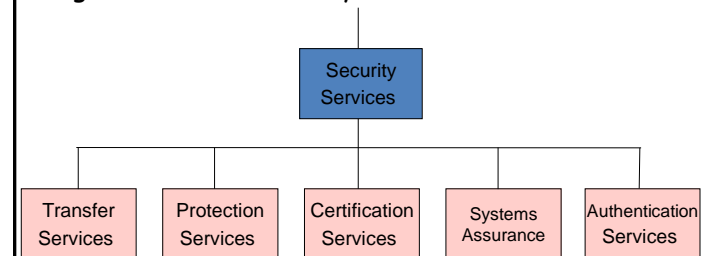
UNSTRUCTURED	SEMI-STRUCTURED	XML
Microsoft Word	XML	LegalXML
Microsoft Excel	HL7	IFX
PowerPoint	HIPAA	cXML
PDF	ASTM	ebXML
Star Office	EDI-X12	HL7 V3.0
Word Perfect	EDI-Fact	ACORD (AL3,
ASCII reports	FIX	GJXDM
HTML	Cargo IMP	TWPDES
EBCDIC	MVR	
Undocumented	AFP	
Flat files	Post Script	
RPG	DJDE	
ANSI		

## 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).

15

## Organization of Security Services

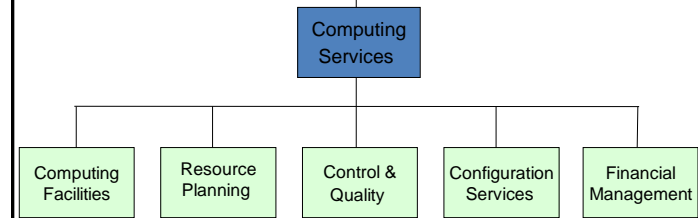


## Security Services = Information Assurance

- 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.

17

## Organization of Computing Services



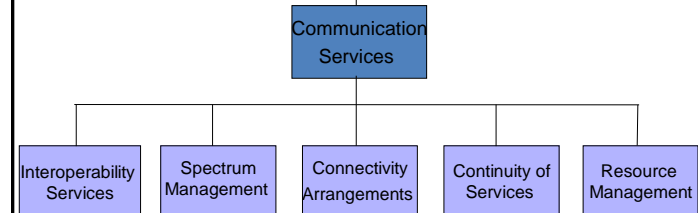
18

## 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.

19

## Organization of Communication Services



20

## 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.

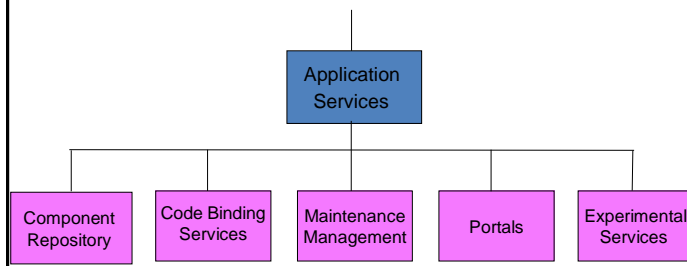
21

## 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.

22

## Organization of Application Services



23

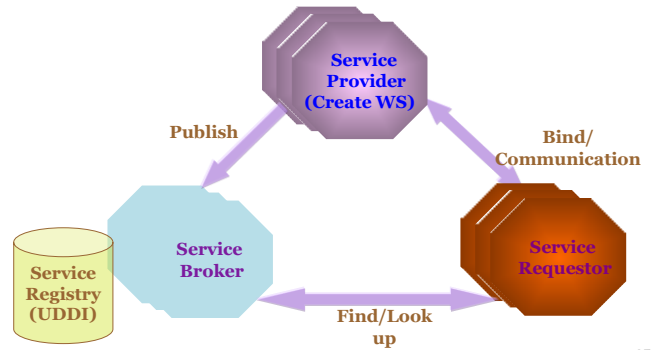
## A Few Key SOA Protocols

- 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.

24

## Components of SOA

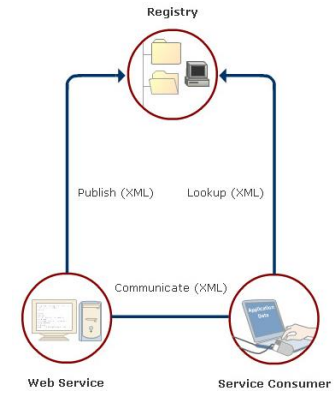
SOA was implemented by Web Service



25

## SOA implementation by WS

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



26

## Web service definition

src: wikipedia

### Web service

From Wikipedia, the free encyclopedia

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](#).<sup>[1]</sup>

The W3C Web Services Architecture Working Group [\[2\]](#) 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.<sup>[1]</sup>

The W3C also states:

We can identify two major classes of Web services:

- *REST-compliant Web services*, in which the primary purpose of the service is to manipulate representations of Web resources using a uniform set of *stateless* operations.
- *Arbitrary Web services*, in which the service may expose an arbitrary set of operations.<sup>[2]</sup>

27

## 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

28

## 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.

29

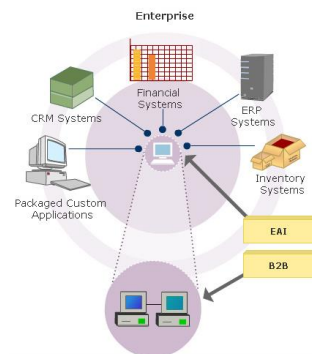
## Characteristics of Web services

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

30

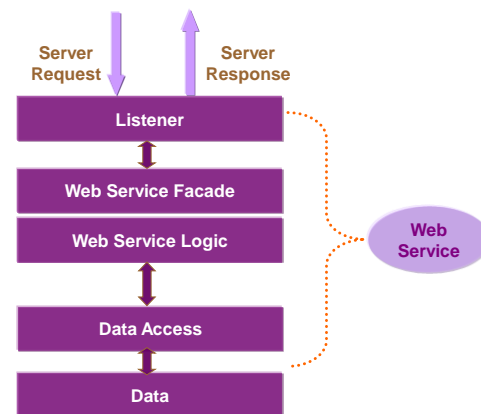
## Uses of Web Services

- 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



31

## Architecture of a web services



32



## 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

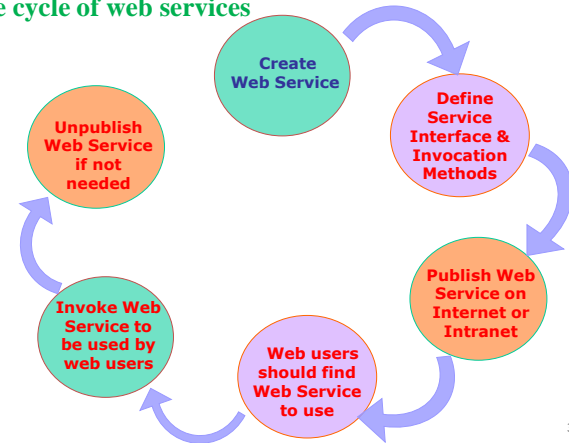
This is any consumer of the web service. The requestor utilizes an existing web service by opening a network connection and sending an XML request.

### 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.

33

## Life cycle of web services



34

## 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:
  - XML: Represents data in a standard format
  - SOAP: Common, extensible, message format
  - WSDL: Common, extensible, service description language
  - UDDI: Maintains registries storing information about service providers and their services

35



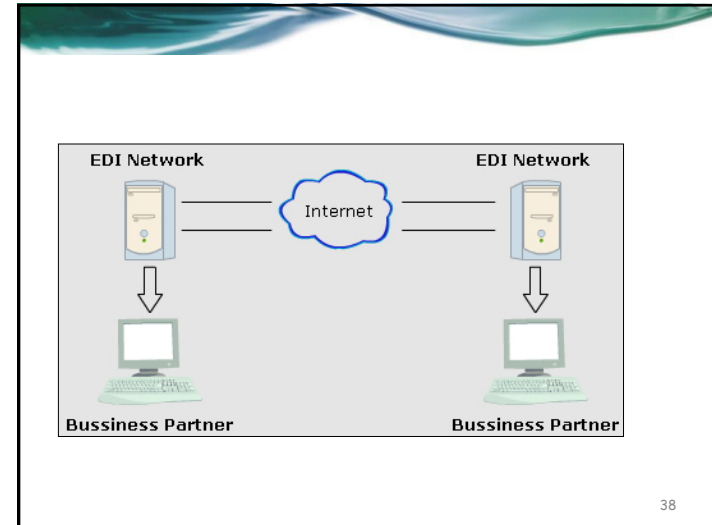
36

## Information exchange approach(1)

### • Electronic Data Interchange (EDI)

- Is a technique used by business partners to exchange business documents that included purchase orders, invoices, shipping notification, financial payments, and so on.
- 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 propriety software to translate business document to X12 format.

37



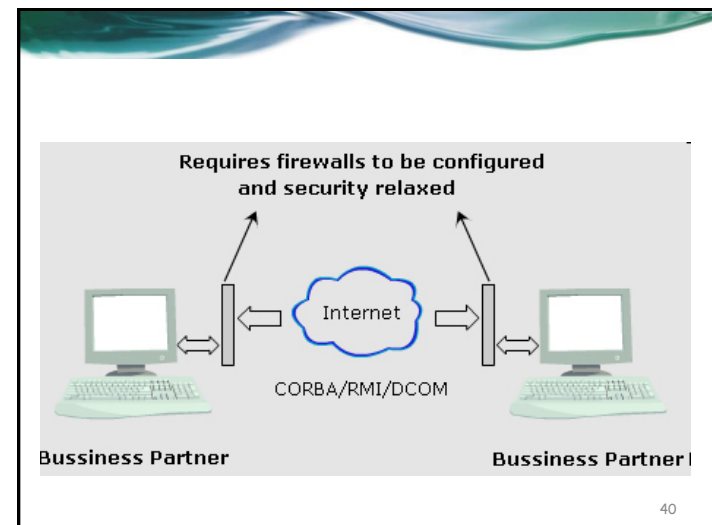
38

## Information exchange approach(2)

### • Remote Procedure Call (RPC)

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

39



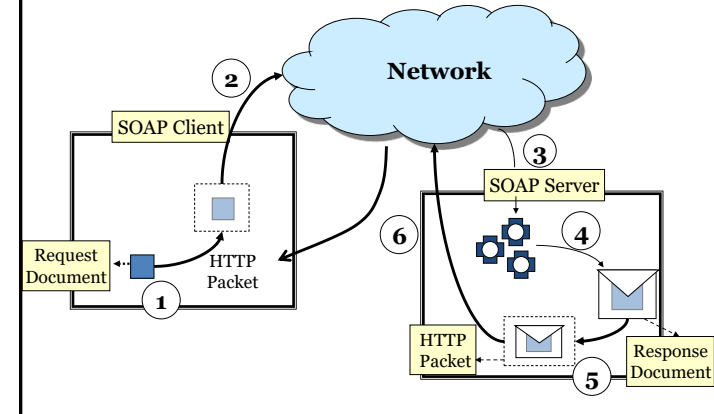
40

## SOAP

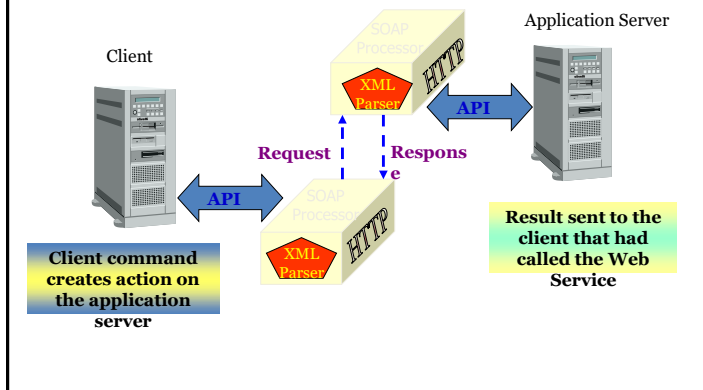
- Provides interoperability between applications by using XML & HTTP
- Uses HTTP protocol to transfer messages between applications. HTTP is recognized by all browsers, proxies, firewalls and servers. Furthermore, it uses port 80 to transmit data.
- Advantages:
  - Vendor Neutral.
  - Transport Protocol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

41

## SOAP (con't)

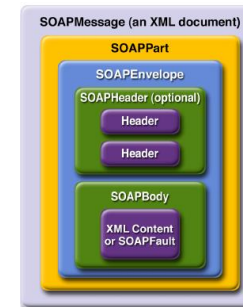


## SOAP (con't)



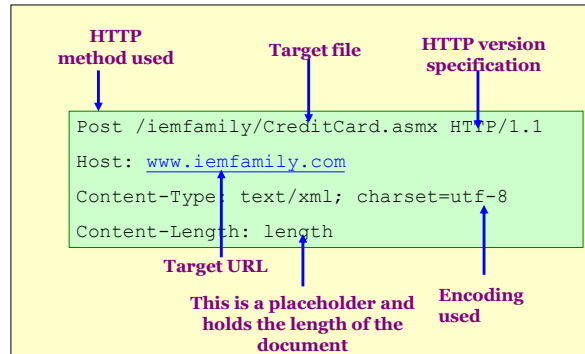
## SOAP Message

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



44

## HTTP Header



## SOAP Header element

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:mid="http://www.flamingo.com/books/message-id"
  xmlns:m="http://www.flamingo.com/books/monitored-by">
  <SOAP-ENV:Header>
    <mid:message-id
      soap:actor="http://www.flamingo.com/logger" >
      11546544ea:b134534:f3sdas5342:4354
    </mid:message-id>
    <m:monitored-by
      SOAP-ENV:actor="
        "http://schemas.xmlsoap.org/soap/actor/next">
      <node>
        <time>1078753670000</time>
        <identity>austria</identity>
      </node>
    </m:monitored-by>
  </SOAP-ENV:Header>
</SOAP-ENV:Envelope>
```

46

## 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

47

## SOAP Request Envelope

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:q0="http://com.vvh/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <q0:Cong>
      <a>3</a>
      <b>4</b>
    </q0:Cong>
  </soapenv:Body>
</soapenv:Envelope>
```

48

## SOAP Response Envelope

```
<S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <CongResponse xmlns:ns2="http://com.vvh/">
      <return>7.0</return>
    </CongResponse>
  </S:Body>
</S:Envelope>
```

49

```
public class CreateSOAPMessage {
    public static void main(String[] args) throws Exception {
        System.out.println("Tao SOAP message");
        MessageFactory msgFactory = MessageFactory.newInstance();
        SOAPMessage message = msgFactory.createMessage();
        message.getSOAPHeader().detachNode();//bỏ header
        SOAPBody body = message.getSOAPBody();
        QName qn = new QName("http://vn.com.ty/", "Cong", "q");
        SOAPElement calc = body.addChildElement(qn);
        calc.setEncodingStyle(SOAPConstants.URI_NS_SOAP_ENCODING);
        SOAPElement a = calc.addChildElement("a");
        a.addTextNode("3");
        SOAPElement b = calc.addChildElement("b");
        b.addTextNode("4");
        message.writeTo(System.out);

        System.out.println("\nGửi SOAP message đến server");
        URL endpoint = new URL("http://localhost:8080/WS_Ex01/CalcWS");
        SOAPConnectionFactory sfc = SOAPConnectionFactory.newInstance();
        SOAPConnection connection = sfc.createConnection();
        SOAPMessage response = connection.call(message, endpoint);
        SOAPPart part = response.getSOAPPart();
        Source src = part.getContent();

        Transformer trans = TransformerFactory.newInstance().newTransformer();
        trans.transform(src, new StreamResult(System.out));
        System.out.println("Finish");
    }
}
```

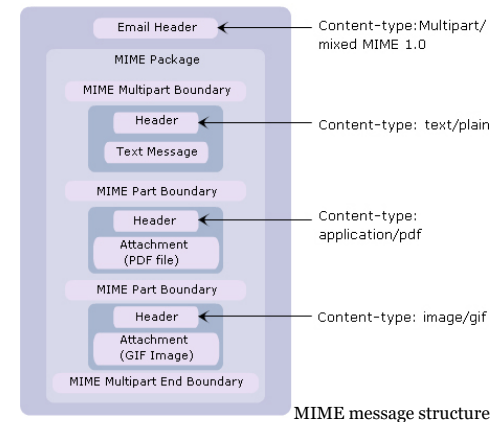
50

## 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.

51

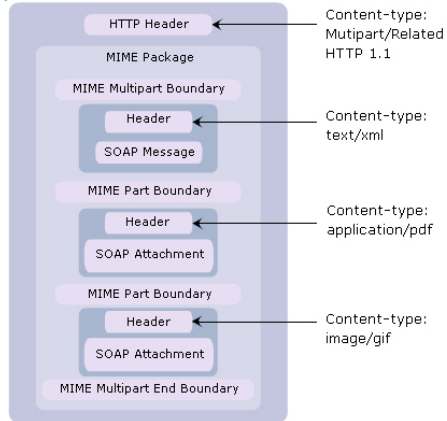
## SOAP with Attachments(2)



MIME message structure

52

## MIME msg with SOAP attachment



53

## Example

```

MessageFactory msgFactory = MessageFactory.newInstance();
SOAPMessage message = msgFactory.createMessage();
message.getSOAPHeader().detachNode();//bỏ header

//đính kèm text
SOAPPart part = message.getSOAPPart();
SOAPEnvelope envelop = part.getEnvelope();

SOAPBody body = message.getSOAPBody();
SOAPBodyElement elem = body.addBodyElement(
    envelop.createName("blog", "v", "http://vovanhai.wordpress.com"));
elem.addTextNode("This is a text message");

//Đính kèm hình
URL url = new URL("http://localhost:8080/images/kem1_2.jpg");
DataHandler dataHandler = new DataHandler(url);
AttachmentPart attachment = message.createAttachmentPart(dataHandler);
attachment.setContentId("attached_image");
message.addAttachmentPart(attachment);

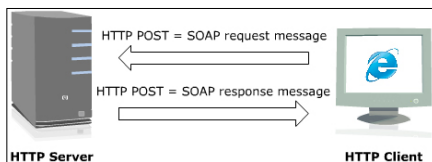
//xuất nội dung SOAP
message.writeTo(System.out);

```

54

## 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.



55

## SOAP request over HTTP

```

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

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns: SOAP-ENV="http://schemas.xmlsoap.org/soap/
envelope/">
...
<SOAP-ENV:Body>
  <m:GetOrderStatus
    xmlns:m="http://www.flamingo.com/methods">
    <m:OrderNo>34347</m:OrderNo>
  </m:GetOrderStatus>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

56

## SOAP response over HTTP

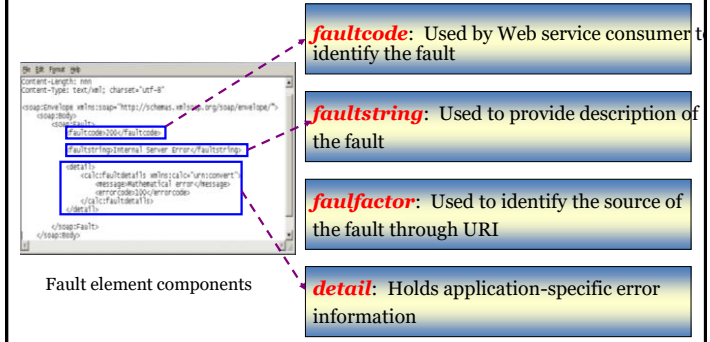
```
HTTP/1.1 200 OK
Connection: close
Content-Length: 659
Content-Type: text/xml; charset=utf-8

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

57

## Fault Element

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



The diagram illustrates the structure of a SOAP fault element within a SOAP message. The XML snippet shows a `<fault>` element with attributes `faultcode` and `faultstring`, and a `<detail>` child element containing `faultactor` and `faultinfo`.

**Fault element components:**

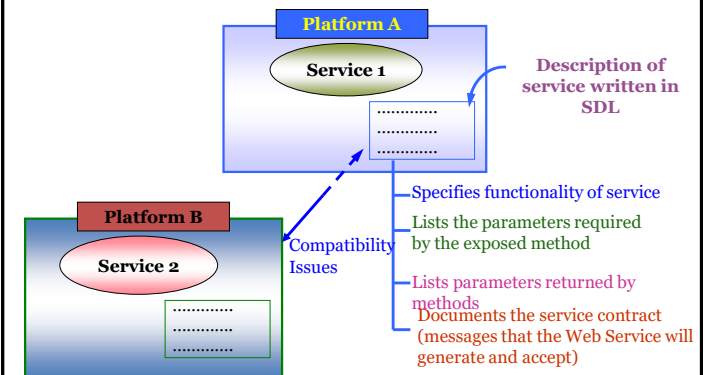
- faultcode:** Used by Web service consumer to identify the fault
- faultstring:** Used to provide description of the fault
- faultactor:** Used to identify the source of the fault through URI
- detail:** Holds application-specific error information



## Web Service Description Language

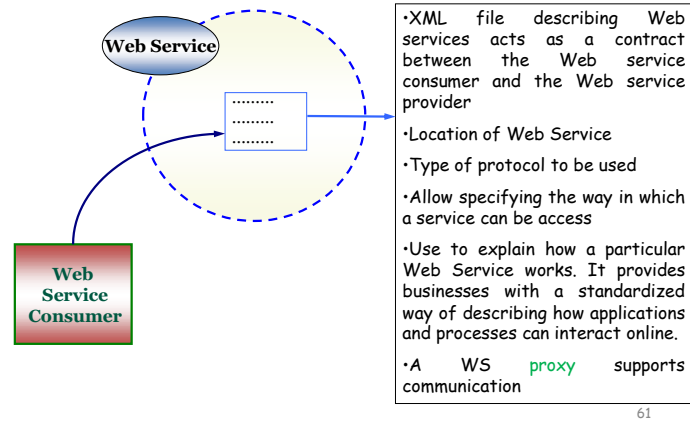
59

## Service description language - SDL

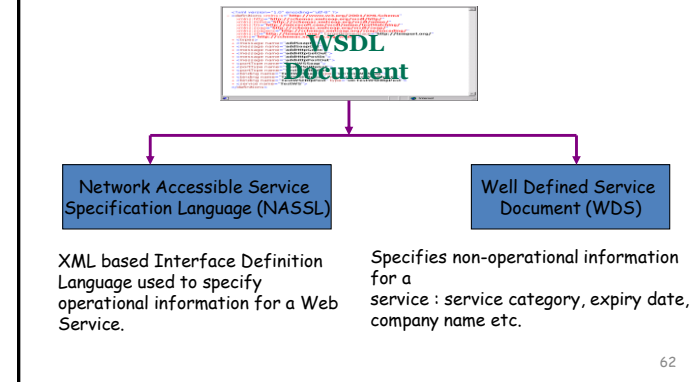


60

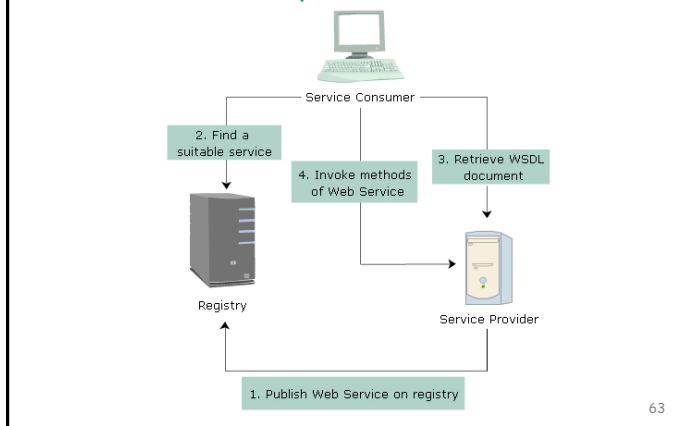
## Web service description language



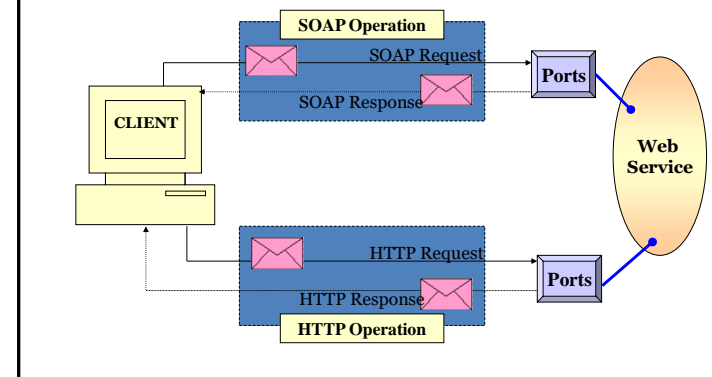
## Web Service Description Language



## WSDL for service providers & consumers



## WSDL Terminologies (1)





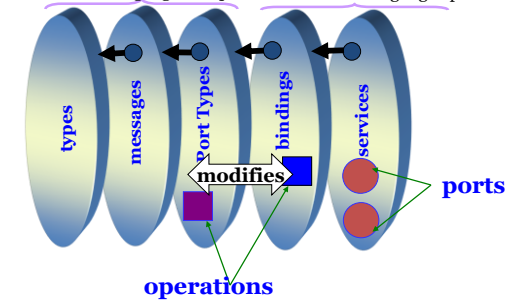
## WSDL Terminologies (2)

- **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

65

## WSDL document structure

**Abstract Definitions**      **Concrete Definitions**  
 Platform and Language Independent      Machine or Language Specific



66

## Sample WSDL

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

```
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://www.webservicex.net"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  targetNamespace="http://www.webservicex.net"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Get one week
    weather forecast for valid zip code or Place name in
    USA</wsdl:documentation>
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
      targetNamespace="http://www.webservicex.net">
      <s:element name="GetWeatherByZipCode">
```

67

## 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.

```
<?xml version="1.0" encoding="utf-8" ?>
<definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:s0="http://tempuri.org/" targetNamespace="http://tempuri.org/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
```

## 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.

```
- <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>
```

69

## 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

```
- <message name="addSoapIn">
  <part name="parameters" element="s0:add" />
</message>
- <message name="addSoapOut">
  <part name="parameters" element="s0:addResponse" />
</message>
.....
```

70

## 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 .

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

71

## 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

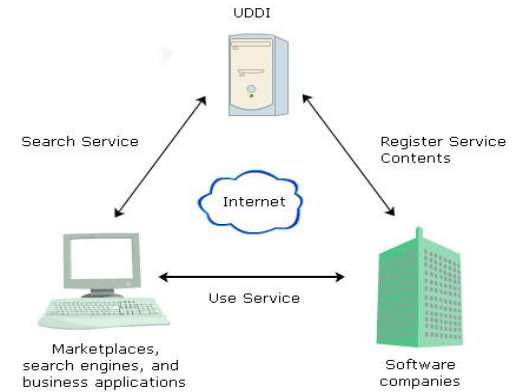
```
<binding name="TestWSSoap" type="s0:TestWSSoap">
  <soap:binding transport=
http://schemas.xmlsoap.org/soap/http style="document" />
  <-operation name="add">
    <soap:operation soapAction=http://tempuri.org/add
      style="document" />
  - <input><soap:body use="literal" /></input>
  - <output>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



## Web Service Registry

73

## Registering and Using Web Services



74

## UDDI



- Stands for **Universal Description, Discovery and Integration**
- Helps businesses to:

**Describe** the business and the services offered by them

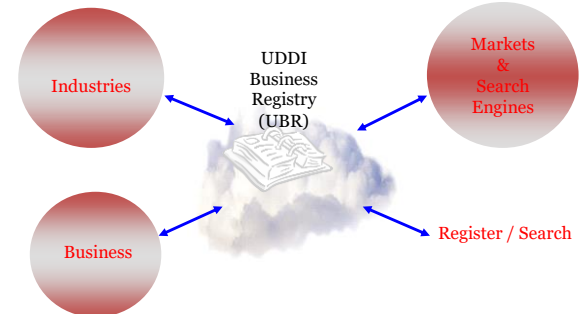
**Discover** other businesses, which may be of some assistance

**Integrate** with other businesses for expansion

75

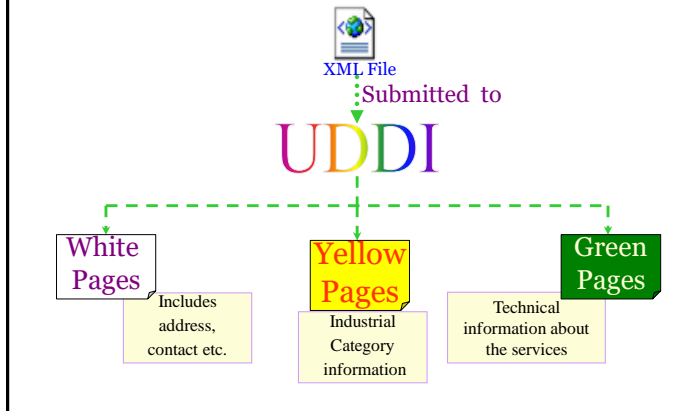
## UDDI Business Registry

Aims at *integrating e-commerce sites*

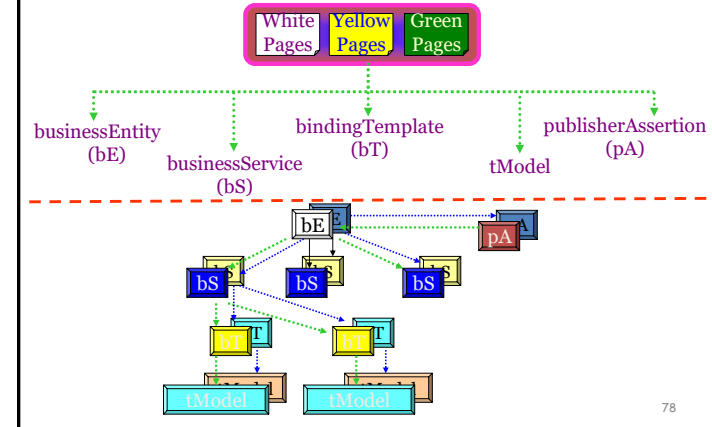


76

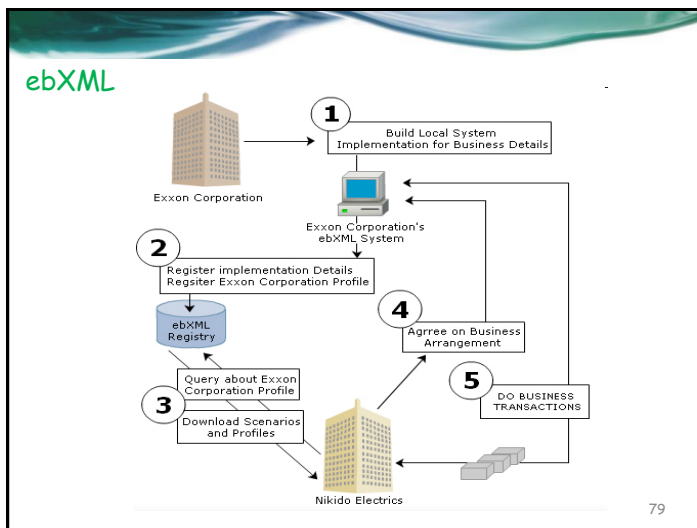
## Working of UDDI



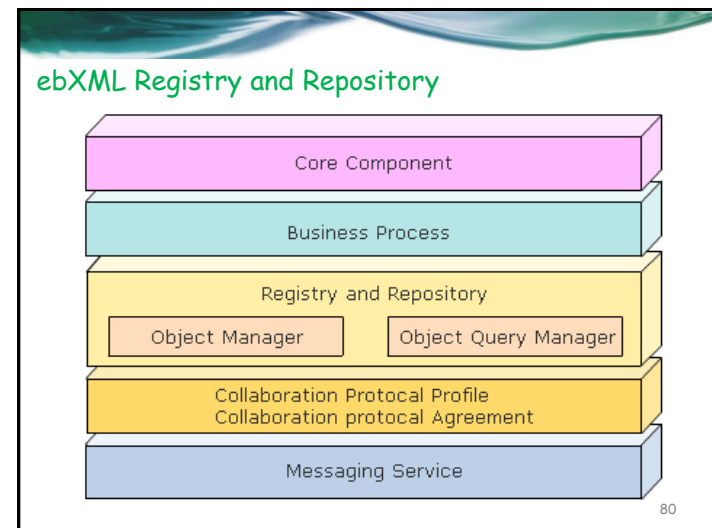
## Registry Architecture and Data Structures



## ebXML

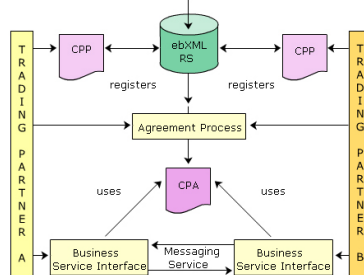


## ebXML Registry and Repository



## 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 *CPP*s.



## Live Demo

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

82

## RESTful Web Services



83

## What is REST ?

- REST stands for **RE**presentational **S**tate **T**ransfer. 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.

84

## 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.

85

## 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.

86

87