CLOUD COMPUTING

Web Services, Service Oriented Architecture

PROF. SOUMYA K. GHOSH
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
IIT KHARAGPUR

What are "Web Services"?

"Software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts" – W3C Web Services Architecture Requirements, Oct. 2002

"Programmable application logic accessible using Standard Internet Protocols…" – Microsoft

"An interface that describes a collection of operations that are network accessible through standardized XML messaging …" – IBM

"Software components that can be spontaneously discovered, combined, and recombined to provide a solution to the user's problem/request … " - SUN

History!

- Structured programming

- Object-oriented programming

- Distributed computing

- Electronic Data Interchange (EDI)

- World Wide Web

- Web Services

Distributed Computing

- When developers create substantial applications, often it is more efficient, or even necessary, for different task to be performed on different computers, called N-tier applications:

  - A 3-tier application might have a user interface on one computer, business-logic processing on a second and a database on a third – all interacting as the application runs.

- For distributed applications to function correctly, application components, e.g. programming objects, executing on different computers throughout a network must be able to communicate.

  E.g.: DCE, CORBA, DCOM, RMI etc.

- Interoperability:

  - Ability to communicate and share data with software from different vendors and platforms

  - Limited among conventional proprietary distributed computing technologies

Electronic Data Interchange (EDI)

- Computer-to-computer exchange of business data and documents between companies using standard formats recognized both nationally and internationally.

- The information used in EDI is organized according to a specified format set by both companies participating in the data exchange.

- Advantages:
  - Lower operating costs
    - Saves time and money
  - Less Errors => More Accuracy
    - No data entry, so less human error
  - Increased Productivity
    - More efficient personnel and faster throughput
  - Faster trading cycle
    - Streamlined processes for improved trading relationships

Web Services

- Take advantage of OOP by enabling developers to build applications from existing software components in a modular approach:

    - Transform a network (e.g. the Internet) into one library of programmatic components available to developers to have significant productivity gains.

- Improve distributed computing interoperability by using open (non-proprietary) standards that can enable (theoretically) any two software components to communicate:

    - Also they are easier to debug because they are text-based, rather than binary, communication protocols

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Web Services *(contd...)*

- Provide capabilities similar to those of EDI (Electronic Data Interchange), but are simpler and less expensive to implement.

- Configured to work with EDI systems, allowing organisations to use the two technologies together or to phase out EDI while adopting Web services.

- Unlike WWW

    - Separates visual from non-visual components

    - Interactions may be either through the browser or through a desktop client (Java Swing, Python, Windows, etc.)

Web Services *(contd...)*

- Intended to solve *three* problems:
  - **Interoperability**:
    - Lack of interoperability standards in distributed object messaging
    - DCOM apps strictly bound to Windows Operating system
    - RMI bound to Java programming language
  - **Firewall traversal**:
    - CORBA and DCOM used non-standard ports
    - Web Services use HTTP; most firewalls allow access though port 80 (HTTP), leading to easier and dynamic collaboration
  - **Complexity**:
    - Web Services: developer-friendly service system
    - Use open, text-based standards, which allow components written in different languages and for different platforms to communicate
    - Implemented incrementally, rather than all at once which lessens the cost and reduces the organisational disruption from an abrupt switch in technologies

Web Service: Definition Revisited

- An application component that:
    - Communicates via open protocols (HTTP, SMTP, etc.)
    - Processes XML messages framed using SOAP
    - Describes its messages using XML Schema
    - Provides an endpoint description using WSDL
    - Can be discovered using UDDI

IIT KHARAGPUR

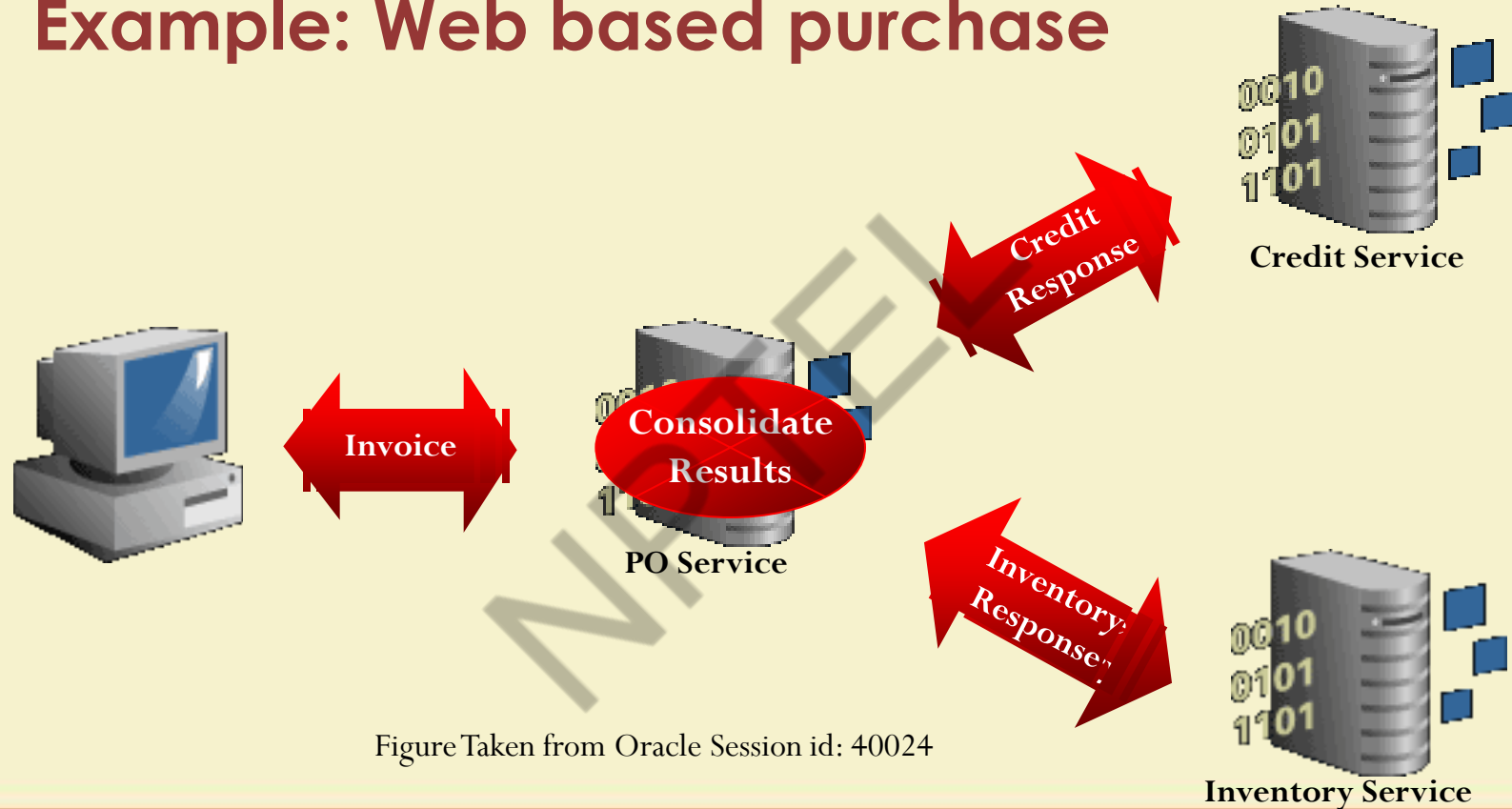NPTEL ONLINE
CERTIFICATION COURSES

Example: Web based purchase

Figure Taken from Oracle Session id: 40024

Credit Service

Credit Response

Invoice

Consolidate Results

PO Service

Inventory Response

Inventory Service

Service Oriented Architecture (SOA)

- IBM has created a model to show Web services interactions which is referred to as a Service-Oriented Architecture (SOA) consisting of relationships between three entities:

  - A service provider;

  - A service requestor;

  - A service broker

- IBM's SOA is a generic model describing service collaboration, not just specific to Web services.

  - See: http://www-106.ibm.com/developerworks/webservices/
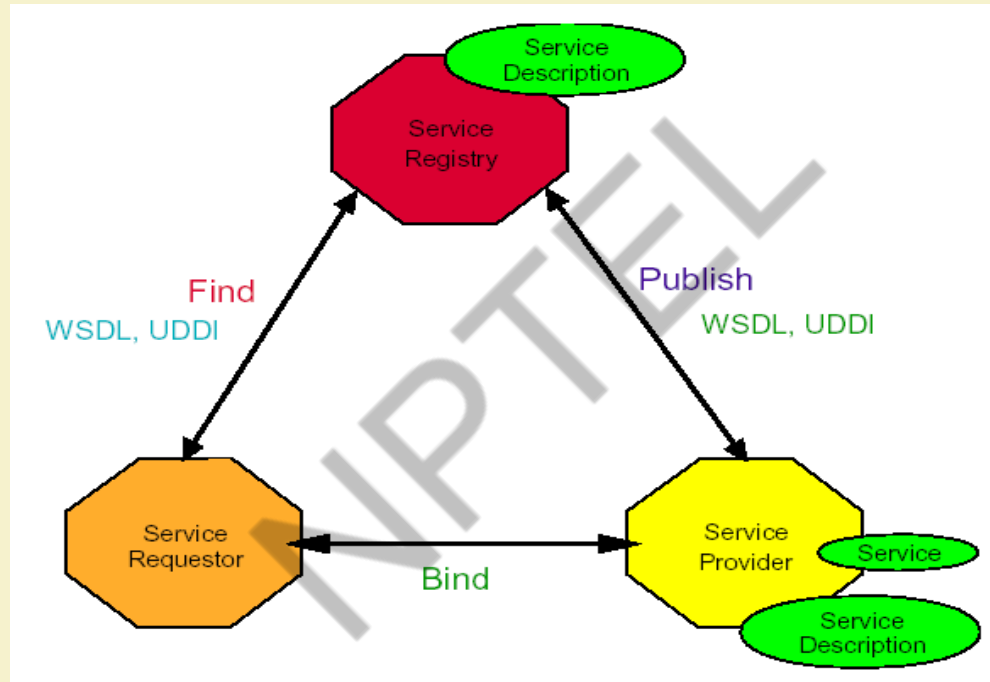
Web Service Model

Web Service Model *(contd...)*

- Roles in Web Service architecture

  - Service provider

    - Owner of the service

    - Platform that hosts access to the service

  - Service requestor

    - Business that requires certain functions to be satisfied

    - Application looking for and invoking an interaction with a service

  - Service registry

    - Searchable registry of service descriptions where service providers publish their service descriptions

Web Service Model *(contd…)*

- Operations in a Web Service Architecture

  - Publish

    - Service descriptions need to be published in order for service requestor to find them

  - Find

    - Service requestor retrieves a service description directly or queries the service registry for the service required

  - Bind

    - Service requestor invokes or initiates an interaction with the service at runtime
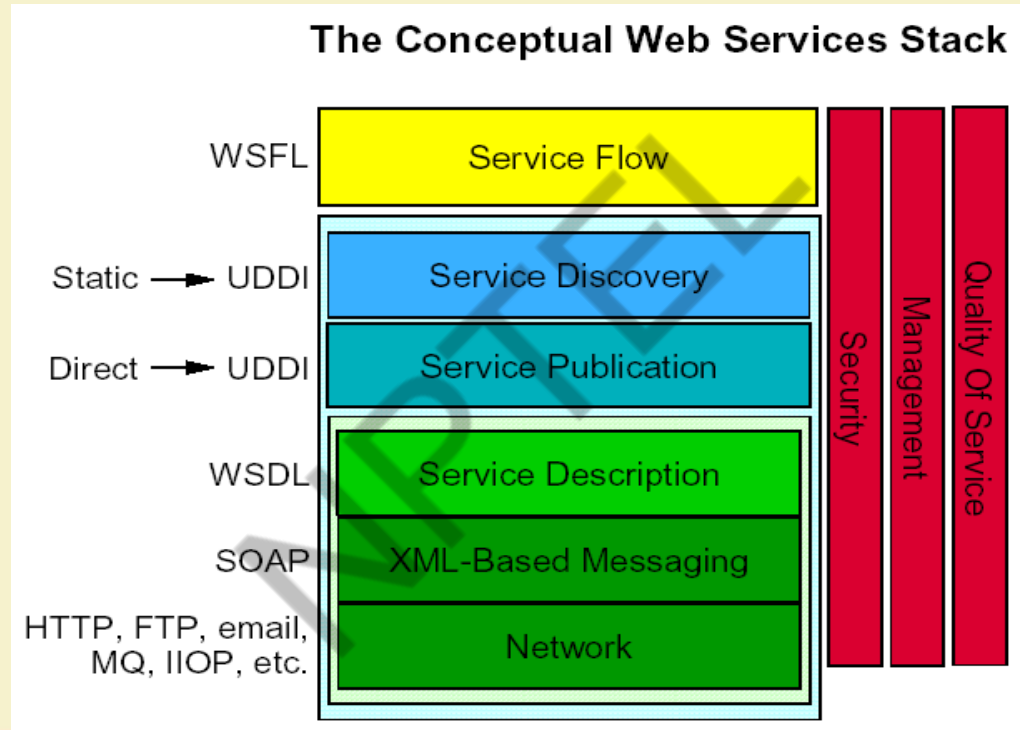
Web Service Components

- **XML** – eXtensible Markup Language

  - A uniform data representation and exchange mechanism.

- **SOAP** – Simple Object Access Protocol

  - A standard way for communication.

- **WSDL** – Web Services Description Language

  - A standard meta language to described the services offered.

- **UDDI** – Universal Description, Discovery and Integration specification

  - A mechanism to register and locate WS based application.

Steps of Operation
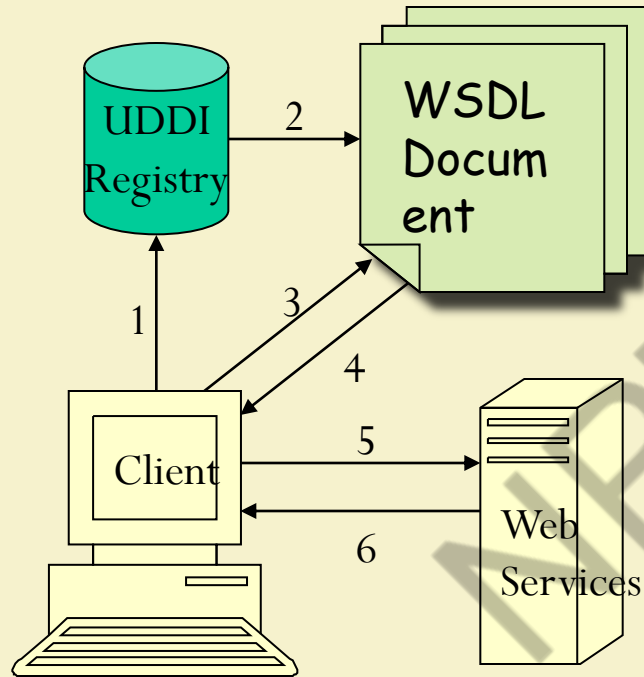
1. Client queries registry to locate service.

2. Registry refers client to WSDL document.

3. Client accesses WSDL document.

4. WSDL provides data to interact with Web service.

5. Client sends SOAP-message request.

6. Web service returns SOAP-message response.

Web Service Stack

The Conceptual Web Services Stack

WSFL	Service Flow	
Static → UDDI	Service Discovery	Security · Management · Quality Of Service
Direct → UDDI	Service Publication	
WSDL	Service Description	
SOAP	XML-Based Messaging	
HTTP, FTP, email, MQ, IIOP, etc.	Network	

XML

- Developed from Standard Generalized Markup Method (SGML)

- Widely supported by W3C

- Essential characteristic is the separation of content from presentation

- Designed to describe **data**

- XML document can optionally reference a *Document Type Definition (DTD),* also called a *Schema*

  - XML parser checks syntax

  - If an XML document adheres to the structure of the schema it is *valid*

XML *(contd…)*

- XML tags are not predefined
  - You must **define your own tags.**

- Enables cross-platform data communication in Web Services

XML vs HTML

An HTML example:

```
<html>
<body>
    <h2>John Doe</h2>
    <p>2 Backroads Lane<br>
        New York<br>
        045935435<br>
        john.doe@gmail.com<br>
        </p>
</body>
</html>
```

XML vs HTML *(contd…)*

- This will be displayed as:

John Doe

2 Backroads Lane

New York

045935435

John.doe@gmail.com

- HTML specifies how the document is to be displayed, and not what information is contained in the document.

- Hard for machine to extract the embedded information. Relatively easy for human.

XML vs HTML *(contd...)*

- Now look at the following:

```
<?xml version=1.0?>
<contact>
    <name>John Doe</name>
    <address>2 Backroads Lane</address>
    <country>New York</country>
    <phone>045935435</phone>
    <email>john.doe@gmail.com</email>
</contact>
```

- In this case:
    - The information contained is being marked, but not for displaying.
    - Readable by both human and machines.

SOAP

- **S**imple **O**bject **A**ccess **P**rotocol

- Format for sending messages over Internet between programs

- XML-based

- Platform and language independent

- Simple and extensible

- Uses mainly HTTP as a transport protocol
  - HTTP message contains a SOAP message as its payload section

- Stateless, one-way
 - But applications can create more complex interaction patterns

SOAP Building Blocks

- Envelope (required) – identifies XML document as SOAP message

- Header (optional) – contains header information

- Body (required) –call and response information

- Fault (optional) – errors that occurred while processing message

Transport protocol

MIME header

SOAP ENVELOPE

SOAP HEADER

SOAP BODY

FAULT
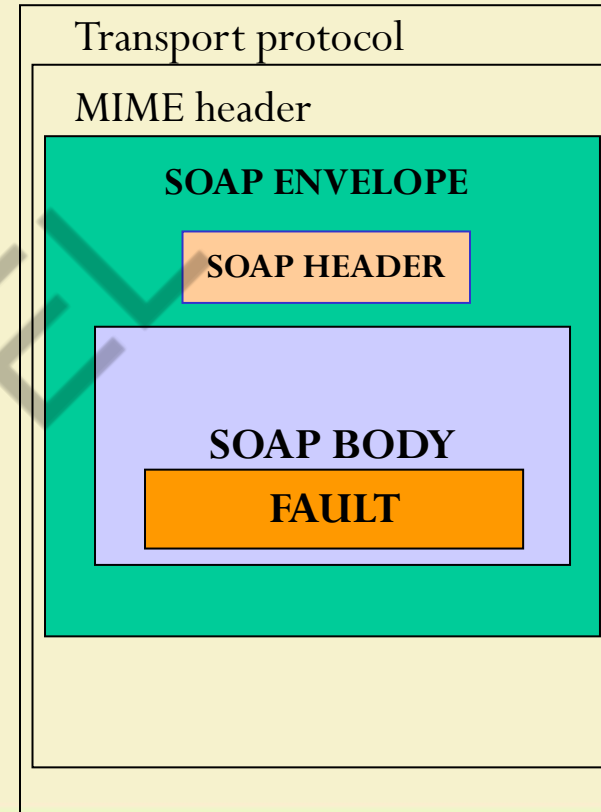
SOAP Message Structure

- Request and Response messages

  - Request invokes a method on a remote object

  - Response returns result of running the method

- SOAP specification defines an "envelop"

  - "envelop" wraps the message itself

  - Message is a different vocabulary

  - Namespace prefix is used to distinguish the two parts

Application-specific message vocabulary

SOAP Envelop vocabulary
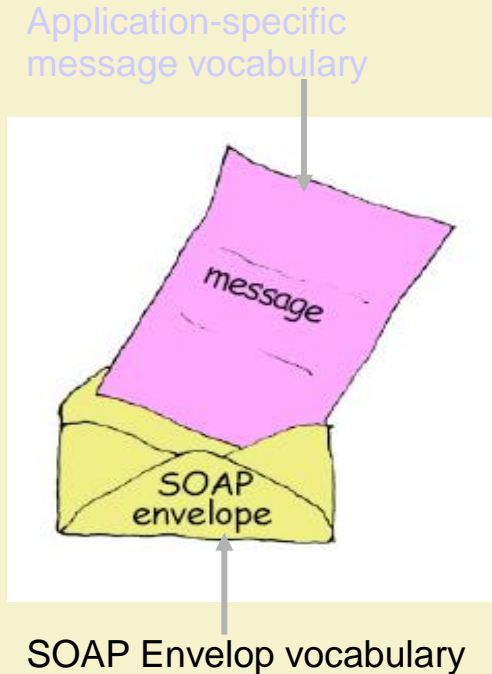
SOAP Request

```
POST /InStock HTTP/1.1
Host: www.stock.org
Content-Type: application/soap+xml; charset=utf-8 Content-Length: 150

<?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.stock.org/stock">
        <m:GetStockPrice>
             <m:StockName>IBM</m:StockName>
        </m:GetStockPrice>
   </soap:Body>
</soap:Envelope>
```

SOAP Response

```
HTTP/1.1 200 OK
Content-Type: application/soap; charset=utf-8
Content-Length: 126


<?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.stock.org/stock">
        <m:GetStockPriceResponse>
                <m:Price>34.5</m:Price>
        </m:GetStockPriceResponse>
    </soap:Body>
</soap:Envelope>
```

Why SOAP?

- Other distributed technologies failed on the Internet

  - Unix RPC – requires binary-compatible Unix implementations at each endpoint

  - CORBA – requires compatible ORBs

  - RMI – requires Java at each endpoint

  - DCOM – requires Windows at each endpoint

- SOAP is the platform-neutral choice

  - Simply an XML wire format

  - Places no restrictions on the endpoint implementation technology choices

SOAP Characteristics

- SOAP has three major characteristics:

  - Extensibility — security and WS-routing are among the extensions under development.

  - Neutrality - SOAP can be used over any transport protocol such as HTTP, SMTP or even TCP.

  - Independent - SOAP allows for any programming model.

SOAP Usage Models

- RPC-like message exchange

    - Request message bundles up method name and parameters

    - Response message contains method return values

    - However, it isn't required by SOAP

- SOAP specification allows any kind of body content

    - Can be XML documents of any type

    - Example:

        - Send a purchase order document to the inbox of B2B partner

        - Expect to receive shipping and exceptions report as response

SOAP Security

- SOAP uses HTTP as a transport protocol and hence can use HTTP security mainly HTTP over SSL.

- But, since SOAP can run over a number of application protocols (such as SMTP) security had to be considered.

- The *WS-Security specification* defines a complete encryption system.

WSDL - Web Service Definition Language

- WSDL : XML vocabulary standard for describing Web services and their capabilities

- Contract between the XML Web service and the client

- Specifies what a request message must contain and what the response message will look like in unambiguous notation

- Defines where the service is available and what communications protocol is used to talk to the service.

WSDL Document Structure

- A WSDL document is just a simple XML document.

- It defines a web service using these major elements:

    - **port type** - The operations performed by the web service.

    - **message -** The messages used by the web service.

    - **types -** The data types used by the web service.

    - **binding -** The communication protocols used by the web service.

A Sample WSDL

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

Binding to SOAP

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>

<binding type="glossaryTerms" name="b1">
<soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
  <operation>
    <soap:operation
     soapAction="http://example.com/getTerm"/>
    <input>
     <soap:body use="literal"/>
    </input>
    <output>
     <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```
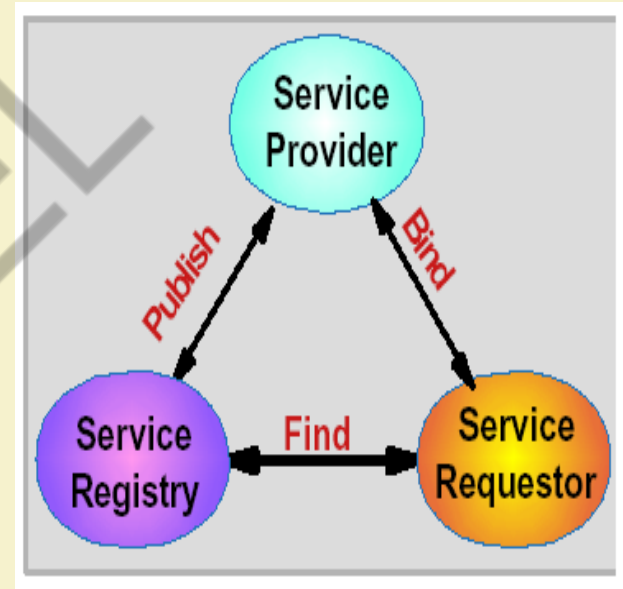
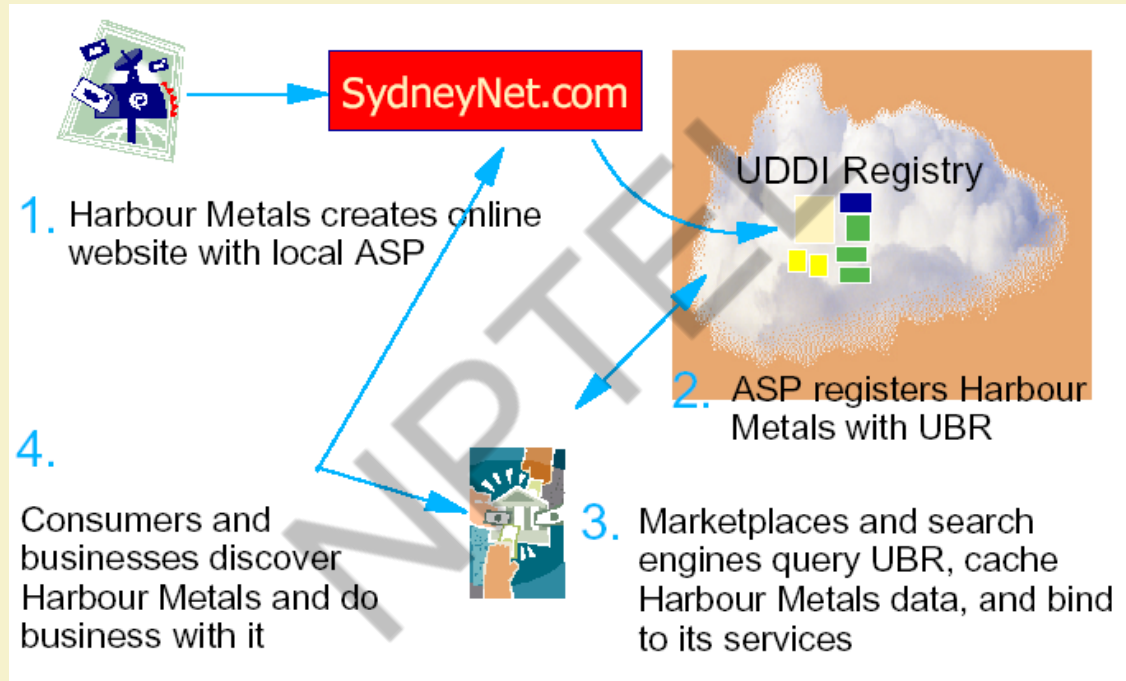# UDDI - Universal Description, Discovery, and Integration

- A framework to define XML-based registries

- Registries are repositories that contain documents that describe business data and also provide search capabilities and programmatic access to remote applications

- Businesses can publish information about themselves and the services they offer

- Can be interrogated by SOAP messages and provides access to WSDL documents describing web services in its directory

UDDI Roles and Operations

- Service Registry
  - Provides support for publishing and locating services
  - Like telephone yellow pages
- Service Provider
  - Provides e-business services
  - Publishes these services through a registry

- Service requestor
  - Finds required services via the Service Broker
  - Binds to services via Service Provider

How can UDDI be Used?

SydneyNet.com

UDDI Registry

1. Harbour Metals creates online website with local ASP

2. ASP registers Harbour Metals with UBR

3. Marketplaces and search engines query UBR, cache Harbour Metals data, and bind to its services

4. Consumers and businesses discover Harbour Metals and do business with it

UDDI Benefits

- Making it possible to discover the right business from the millions currently online

- Defining how to enable commerce once the preferred business is discovered

- Reaching new customers and increasing access to current customers

- Expanding offerings and extending market reach