Lecture 7 : Service Oriented Architecture

Dr. Bibhas Ghoshal

Assistant Professor

Department of Information Technology

Indian Institute of Information Technology Allahabad

Cloud Computing

- Cloud Computing is the use of computing resources (hardware and software) that are delivered as a service over a network. (Wikipedia)

  Service is : logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit, provide weather data, consolidate drilling reports); is self-contained; may be composed of other services; is a "blackbox" to consumers of the service.

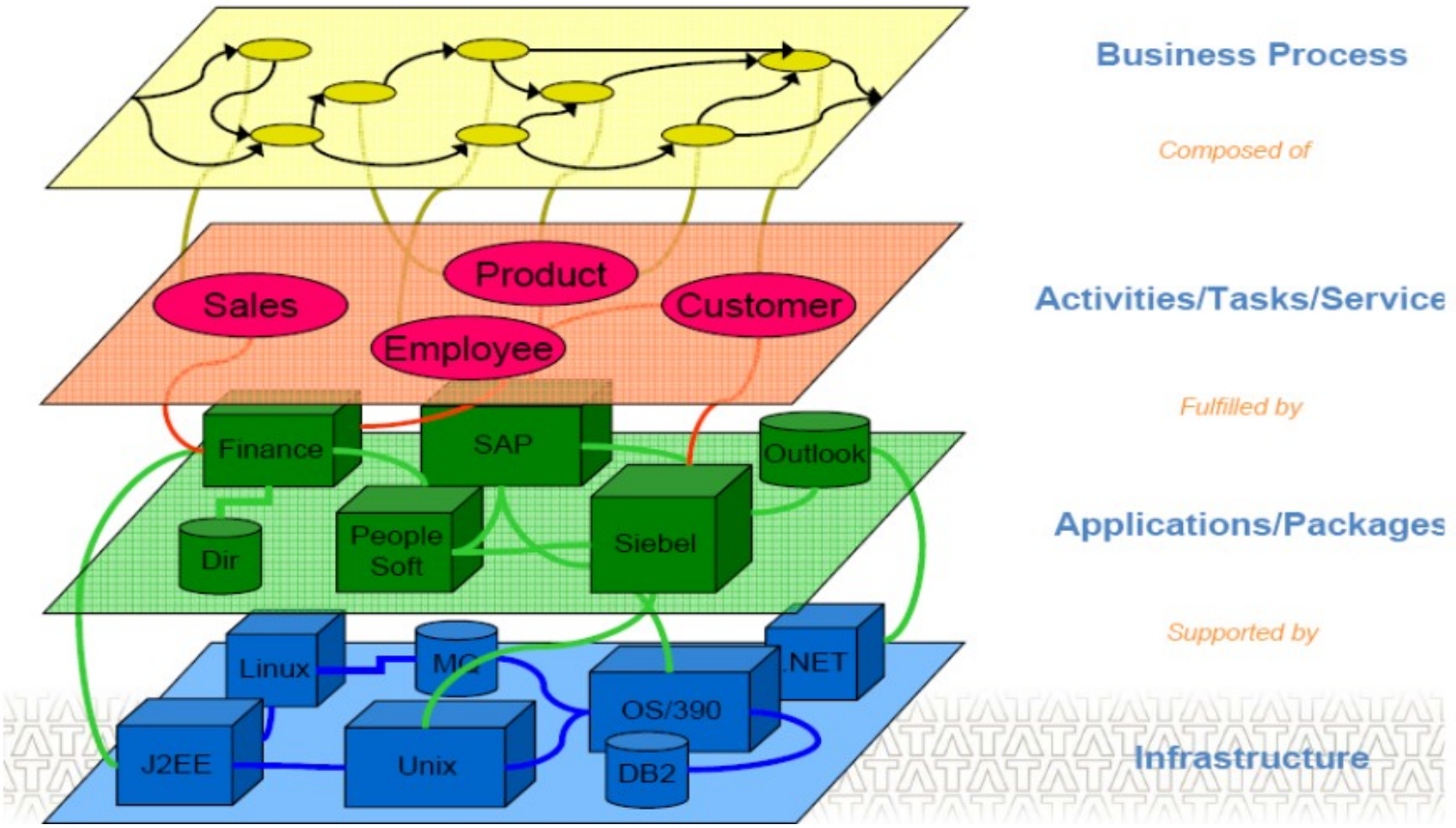
- Service contains

  Contract – message type def, constraint, description (comment)

  Interface – set of operations

  Implementation – Logic and data

Types of Service



Examples of Service

- Multiple business processes in an organization require the user authentication functionality. Instead of rewriting the authentication code for all business processes, one can create a single authentication service and reuse it for all applications.
- Almost all systems across a healthcare organization, such as patient management systems and electronic health record (EHR) systems, need to register patients. These systems can call a single, common service to perform the patient registration task.
- Creating a Purchase Order inside a mainframe application
- Requesting and reserving a room in a hotel
- Applying for a loan by filling out a loan request form
- Search books/music based on keywords

Web Service

A web service is a set of methods exposed through a web interface.

- Accessible through HTTP
- Provides internet access to RPC-like calls that define the service
- Web service messages are encoded in an XML dialect called Simple Object Access Protocol (SOAP)

Service model assumes services are always available

Benefits of Web Service

Web services use this special architecture because it:

- Can be used from any platform.
- Uses a standard, well-know channel.
- Is routable and will pass through most firewalls.
- Uses the same security mechanisms as any web site.

Web Service

A web service is a set of methods exposed through a web interface.

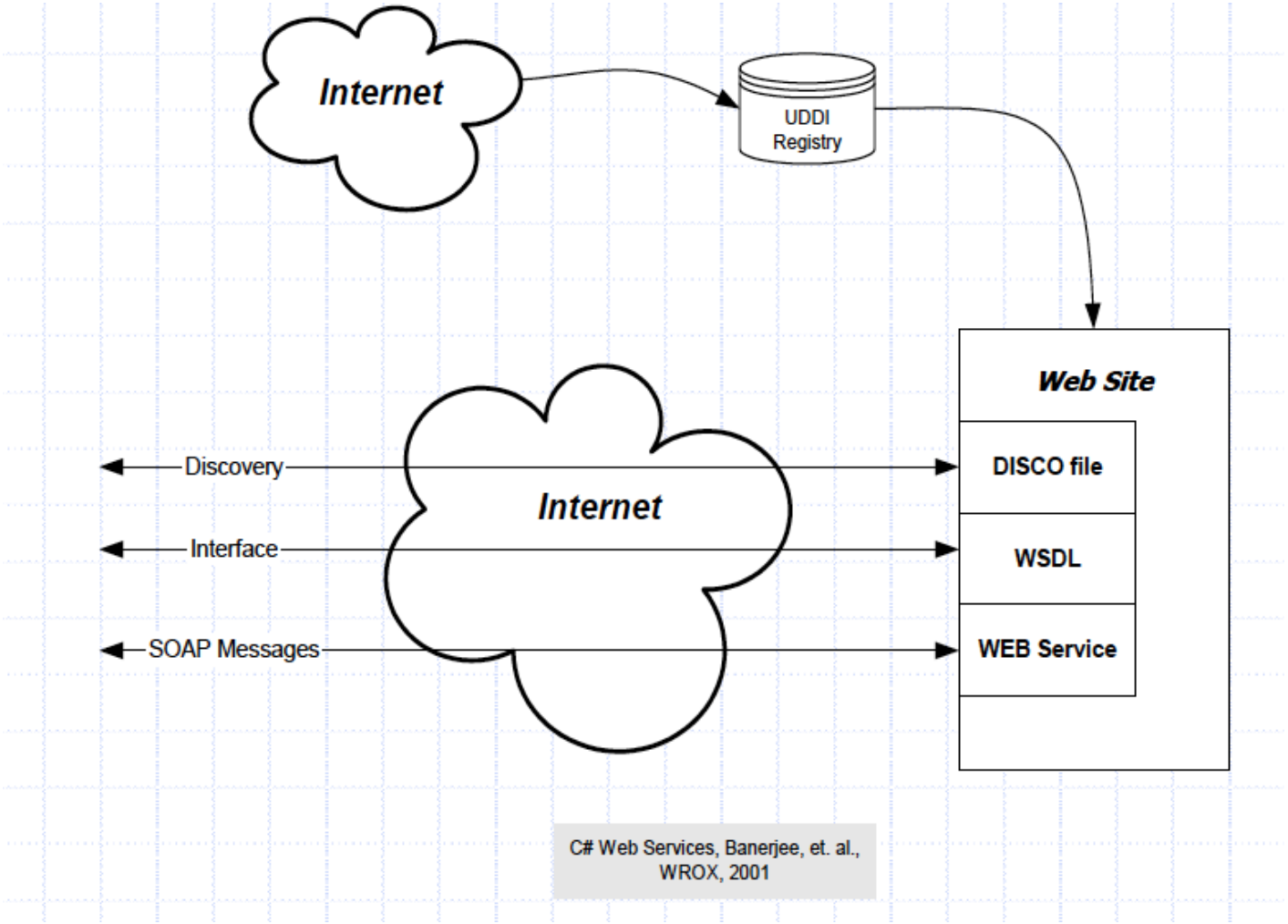
- Accessible through HTTP
- Provides internet access to RPC-like calls that define the service
- Web service messages are encoded in an XML dialect called Simple Object Access Protocol (SOAP)

Service model assumes services are always available

Web Service Protocols

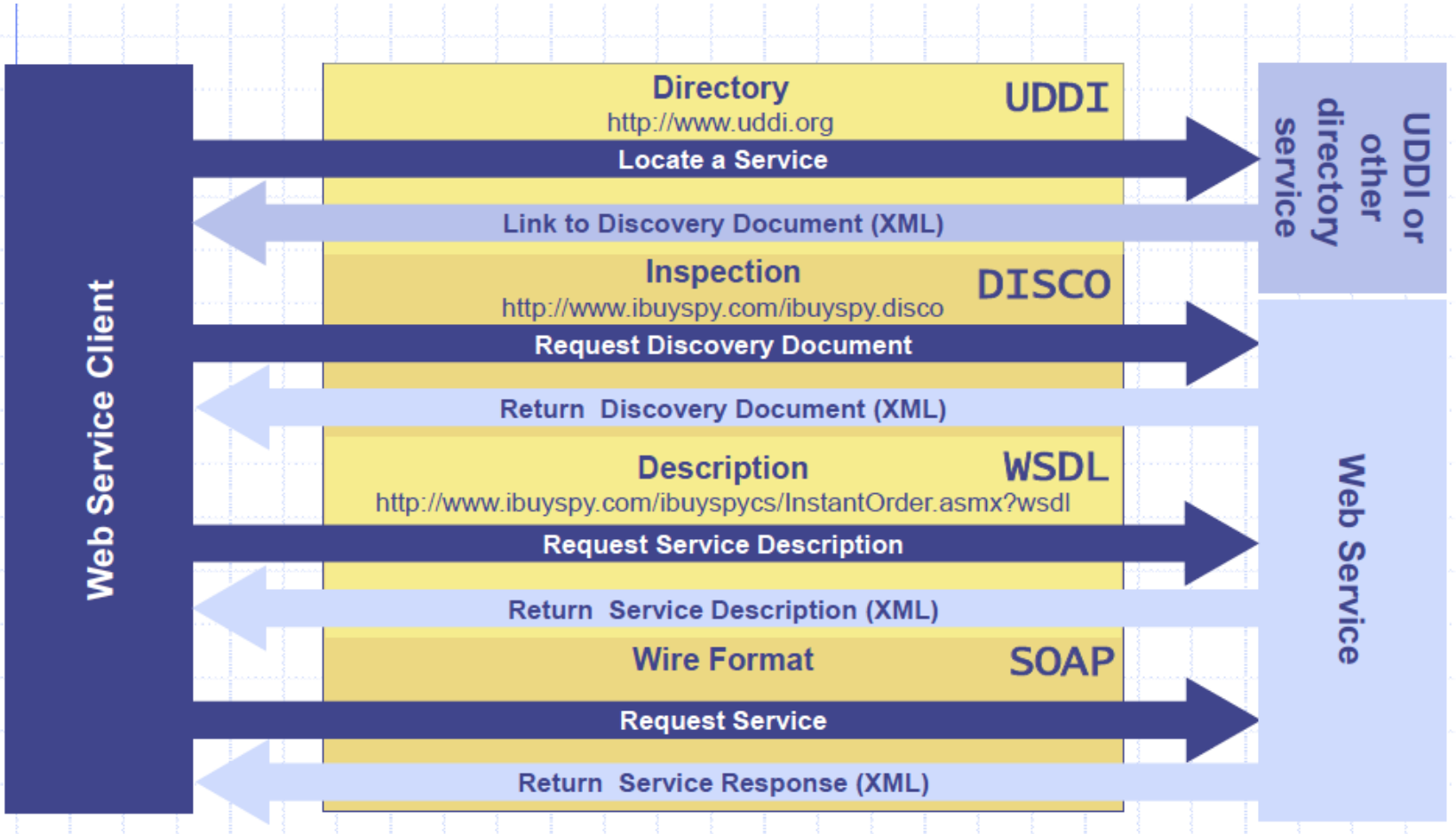
- Web services are based on four protocols:
  - Extensible Markup Language (XML)
    - defines complex data structures
  - Web Service Description Language (WSDL)
    - Specifies the interface of the web service
  - Discovery Protocol (DISCO)
    - Pointer to all web services on a particular web site
  - Universal Description, Discovery, and Integration (UDDI)
    - Central repository of web service descriptions

Web Service Structure



C# Web Services, Banerjee, et. al., WROX, 2001

Web Service Stack



SOAP Web Service

Simple Object Access Protocol (SOAP)

- Industry open-standard XML-based messaging protocol for exchanging information among computers – XML way of defining what information is sent and how is it sent

- Provides Data Transport for Web Services

- It can be delivered via a variety of transport protocols, but the initial focus of SOAP is remote procedure calls transported via HTTP

- SOAP messages are written entirely in XML and are therefore uniquely platform and language-independent

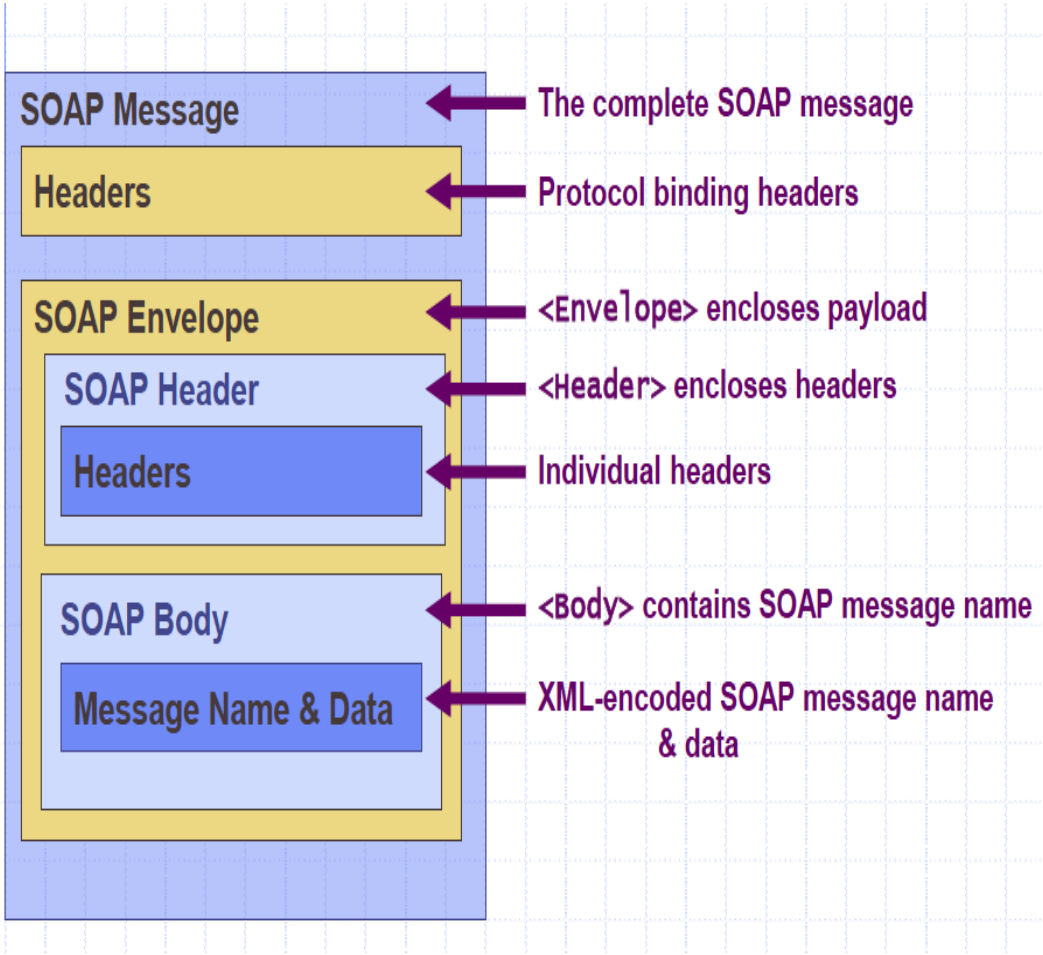
- SOAP enables client applications to easily connect to remote services and invoke remote methods

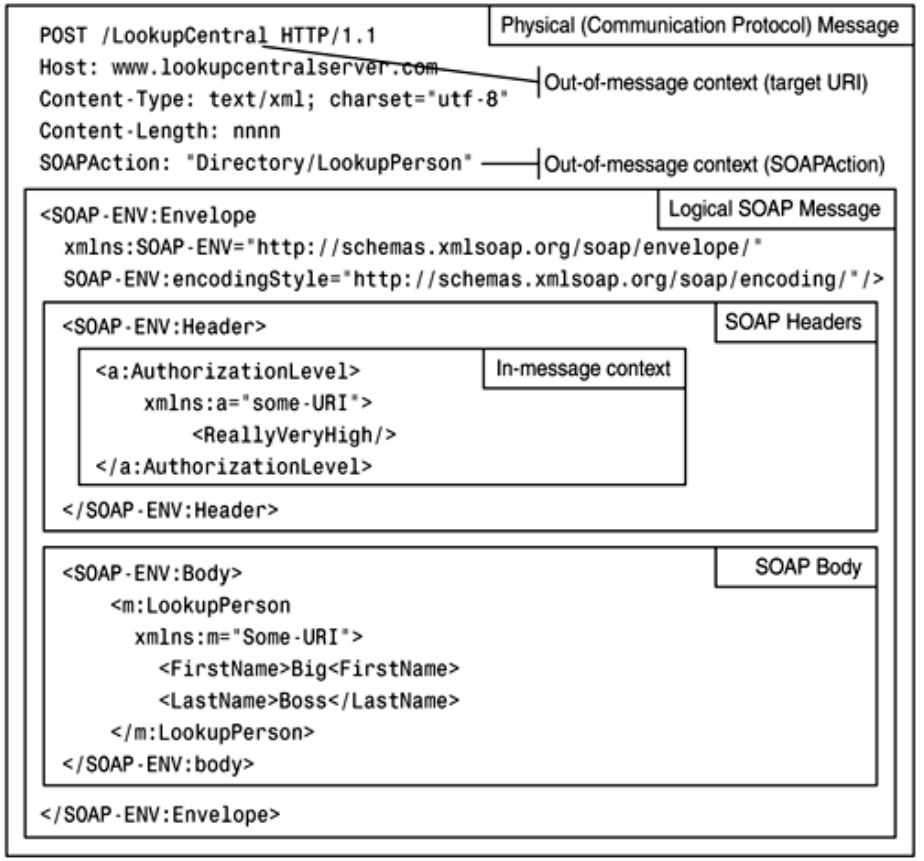
SOAP Message

A SOAP Message can be one of three types:

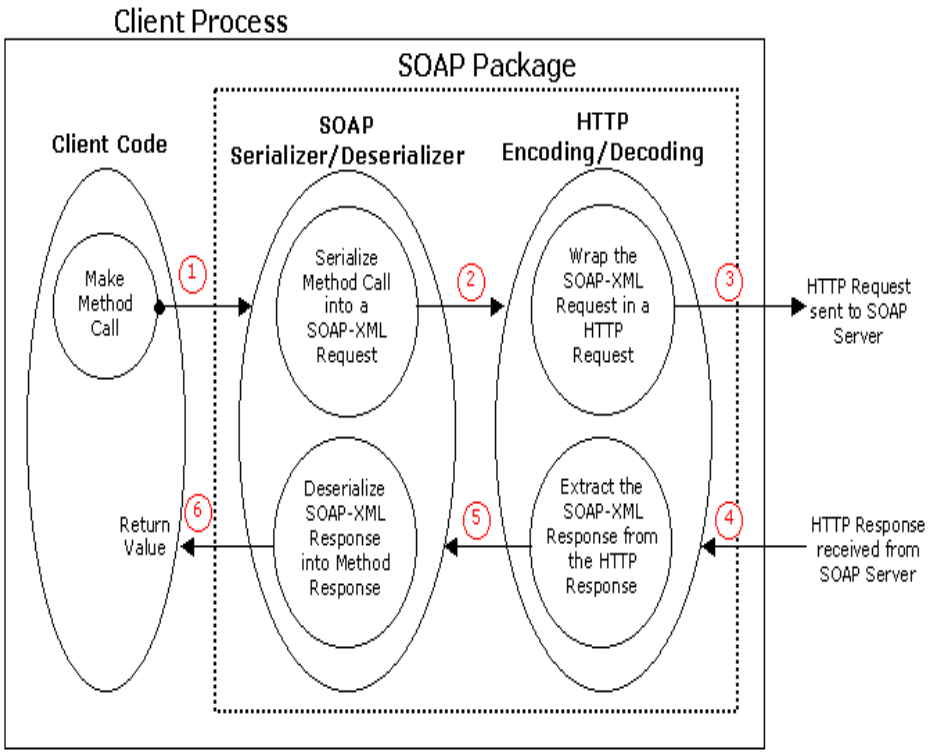
- Method call
  - Contains name of method and parameters
- Method Response
  - Return values
- Fault Message
  - SOAP fault message if service throws an exception
  - Will get standard HTTP message if transport fails.

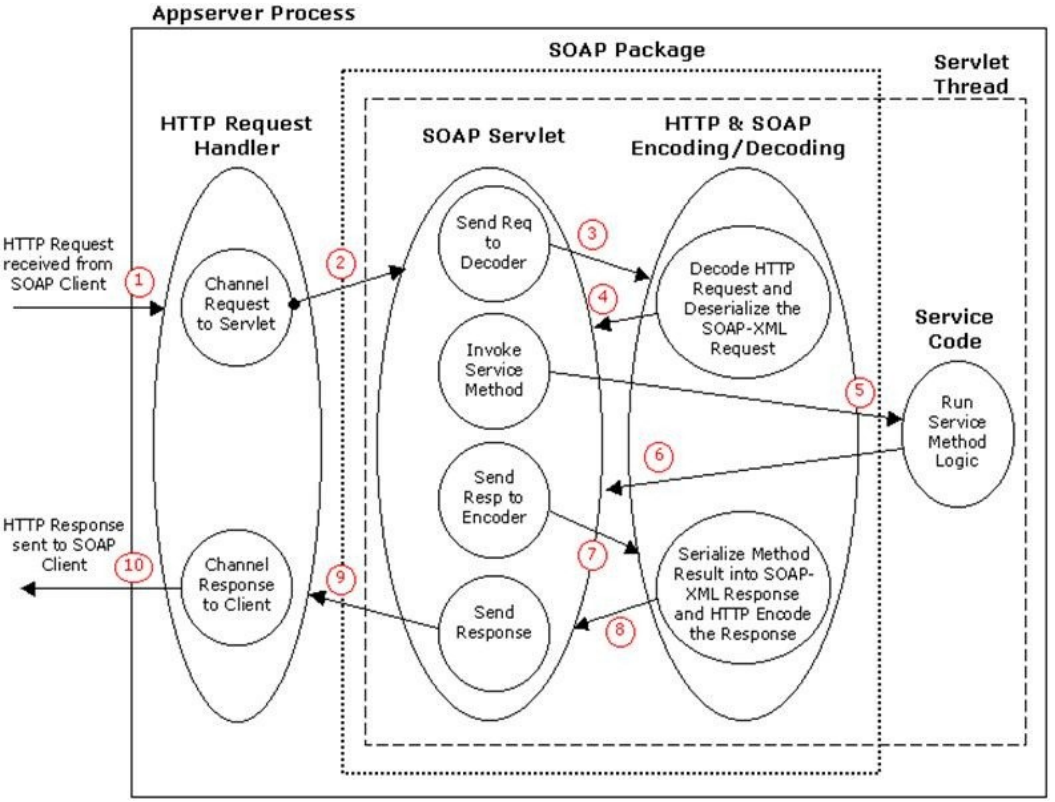
SOAP Message Structure



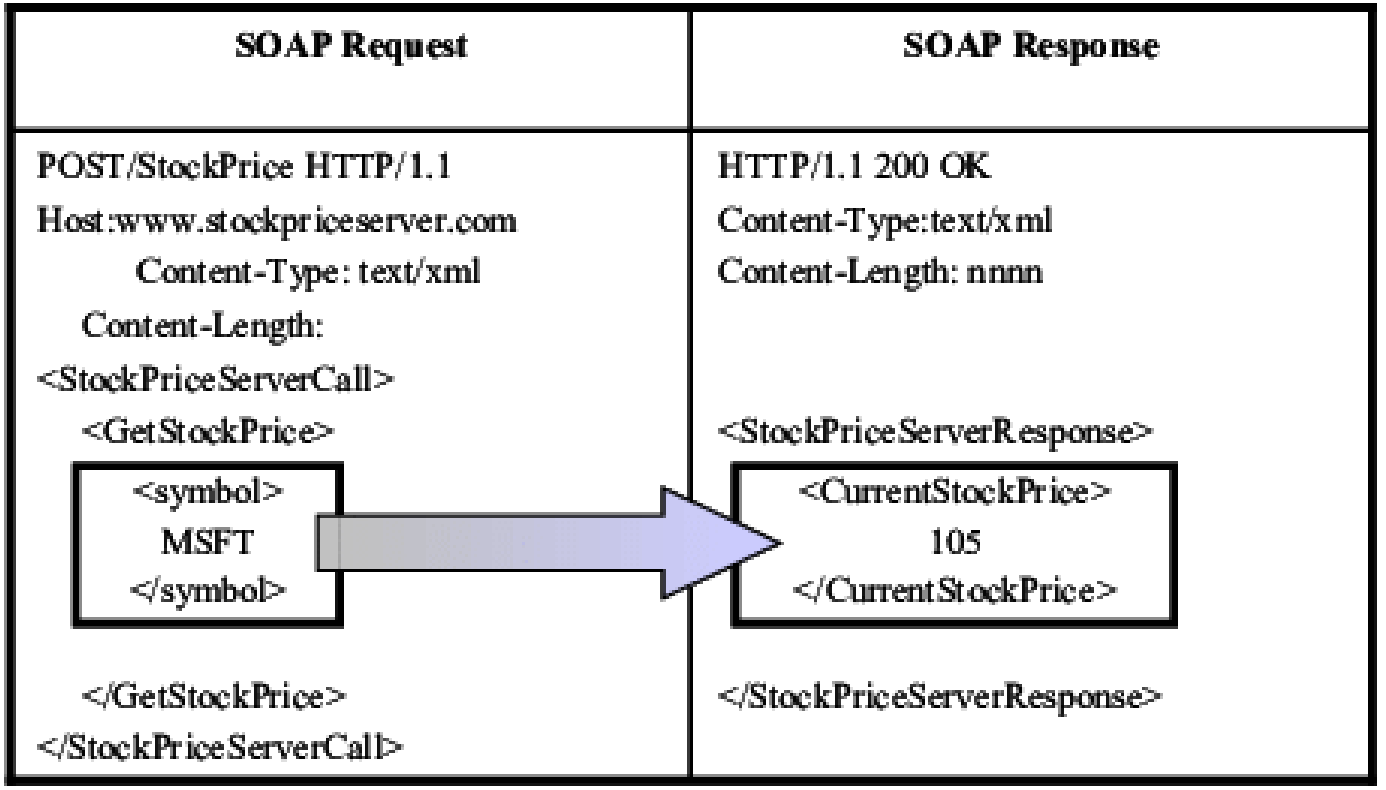


SOAP Processing





SOAP Request Response



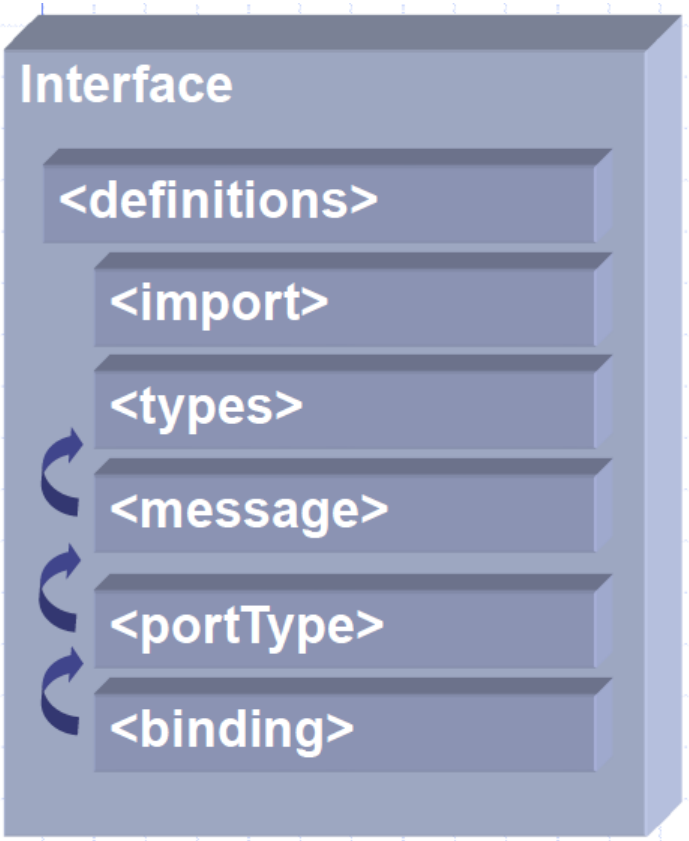
Web Services Description Language (WSDL)

WSDL - XML-based interface definition language that is used for describing the functionality offered by a web service;
• Current version - WSDL 2.0

WSDL File - WSDL description of a web service, provides a machine-readable description of :
• *how the service can be called*
• *what parameters it expects*
• *what data structures it returns*

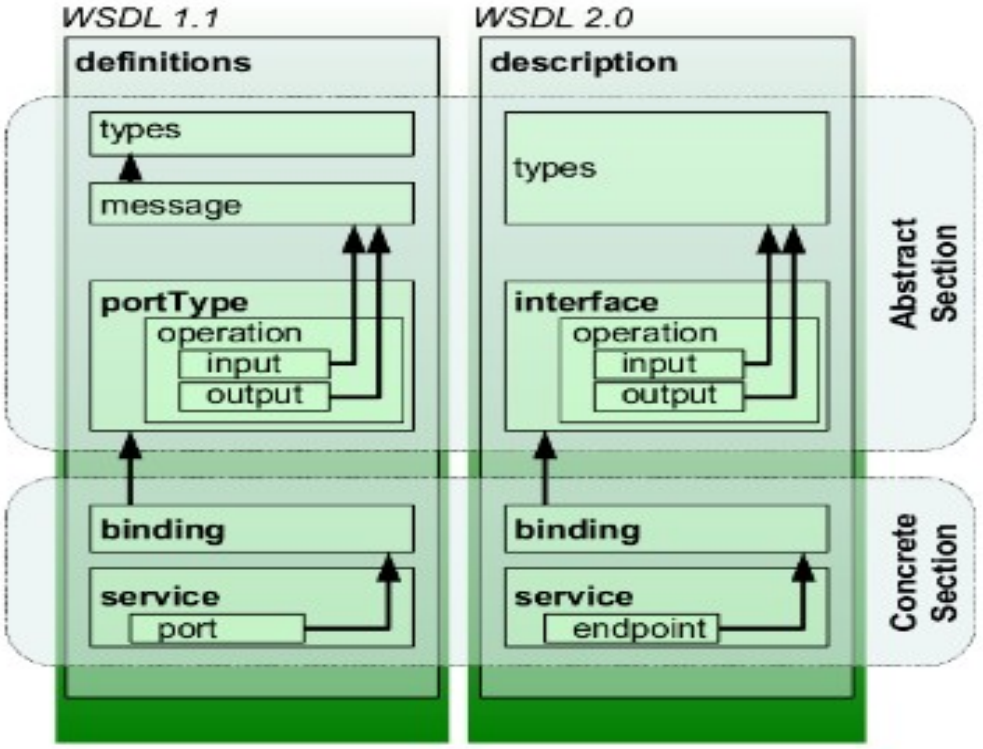
Services described as collection of endpoints or ports
- Abstract definitions of ports and messages are separated from their concrete use
- A port is defined by associating a network address with a reusable binding
- Collection of ports is Service
- Messages are abstract descriptions of the data being exchanged
- port types are abstract collections of supported operations
- the concrete protocol and data format specifications for a particular port type
constitutes a reusable binding, where the operations and messages are then
bound to a concrete network protocol and message format.
- WSDL is often used in combination with SOAP and an XML Schema to provide Web services over the Internet.
- A client program connecting to a Web service can read the WSDL file to determine what operations are available on the server.
• Any special datatypes used are embedded in the WSDL file in the form of XML Schema.
•The client can then use SOAP to actually call one of the operations listed in the WSDL file using for example XML over HTTP.

WSDL Schema



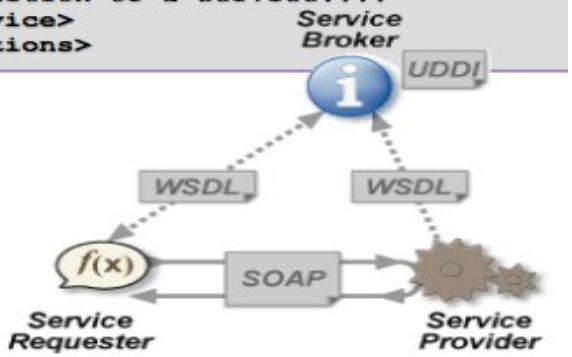
- <definitions> are root node of WSDL
- <import> allows other entities for inclusion
- <types> are data definitions - xsd
- <message> defines parameters of a Web Service function
- <portType> defines input and output operations
- <binding> specifies how each message is sent over the wire

Web Services Description Language (WSDL)

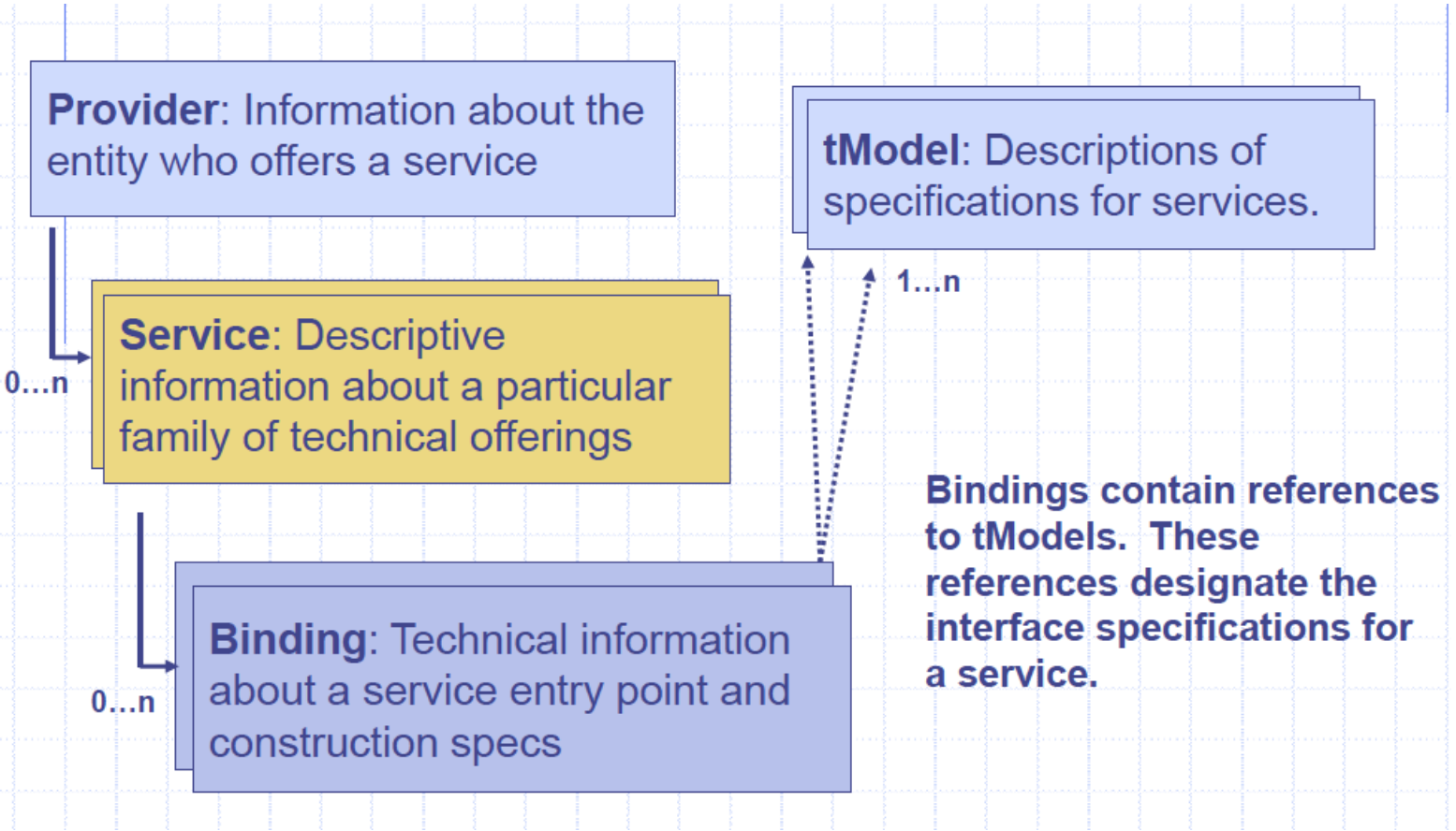



```
<definitions>
   <types>
    definition of types.....
   </types>
   <message>
    definition of a message....
   </message>
   <portType>
      <operation>
       definition of a operation.......
      </operation>
   </portType>
   <binding>
    definition of a binding....
   </binding>
   <service>
    definition of a service....
   </service>
</definitions>
```

WSDL Document Structure

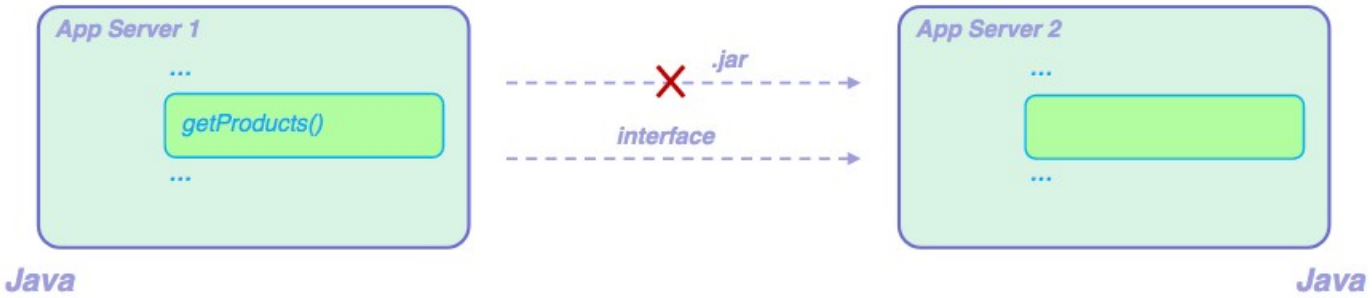


UDDI Information Model



SOAP Web Services

Why Web Services?..

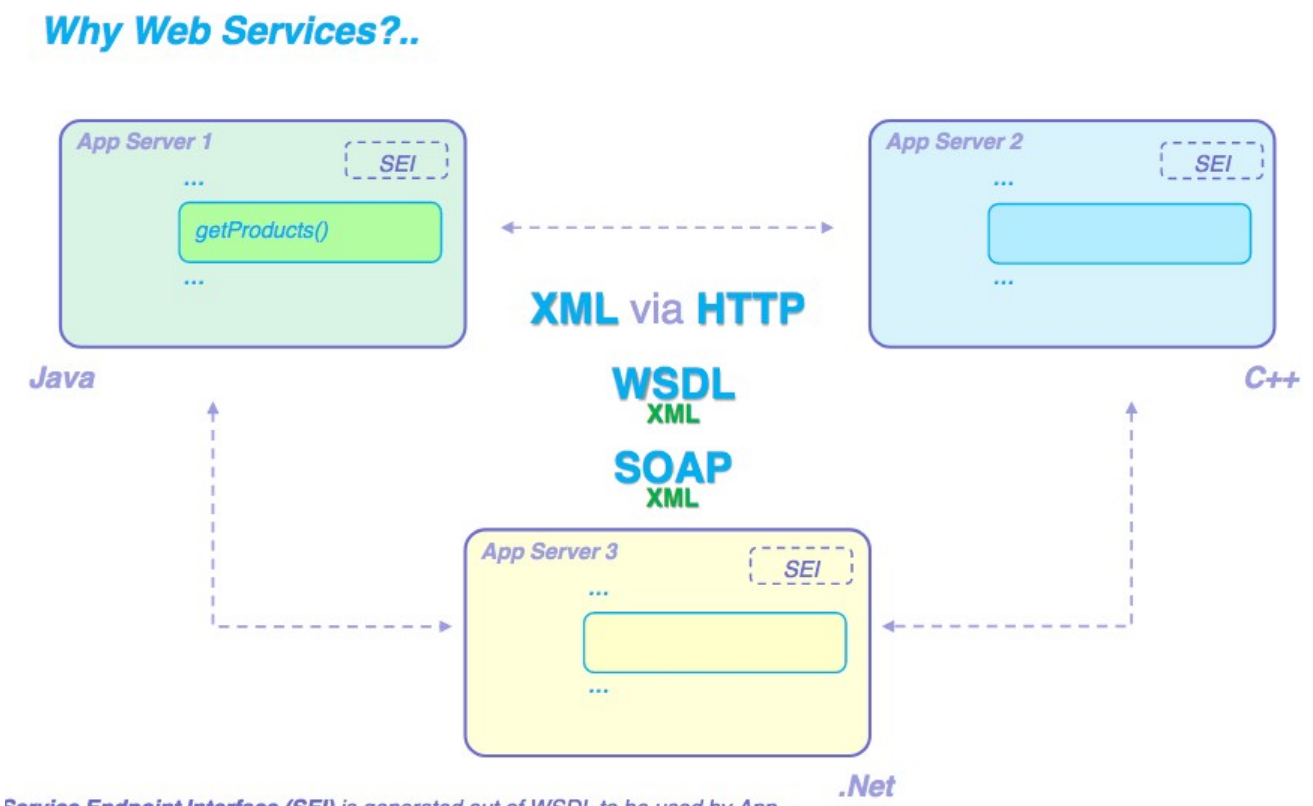


- *Simple packaging of business logic into **.jar** file and porting it to other server does not work in case of dependency of business logic on other resources (applications, database, etc.) from the mother server.*
- *To allow consumer to use an implementation of some business logic, provider should provide an **Interface** as a form of contract.*

... but, if we consider clients that are implemented with different technology (different programming language, platform, etc.),

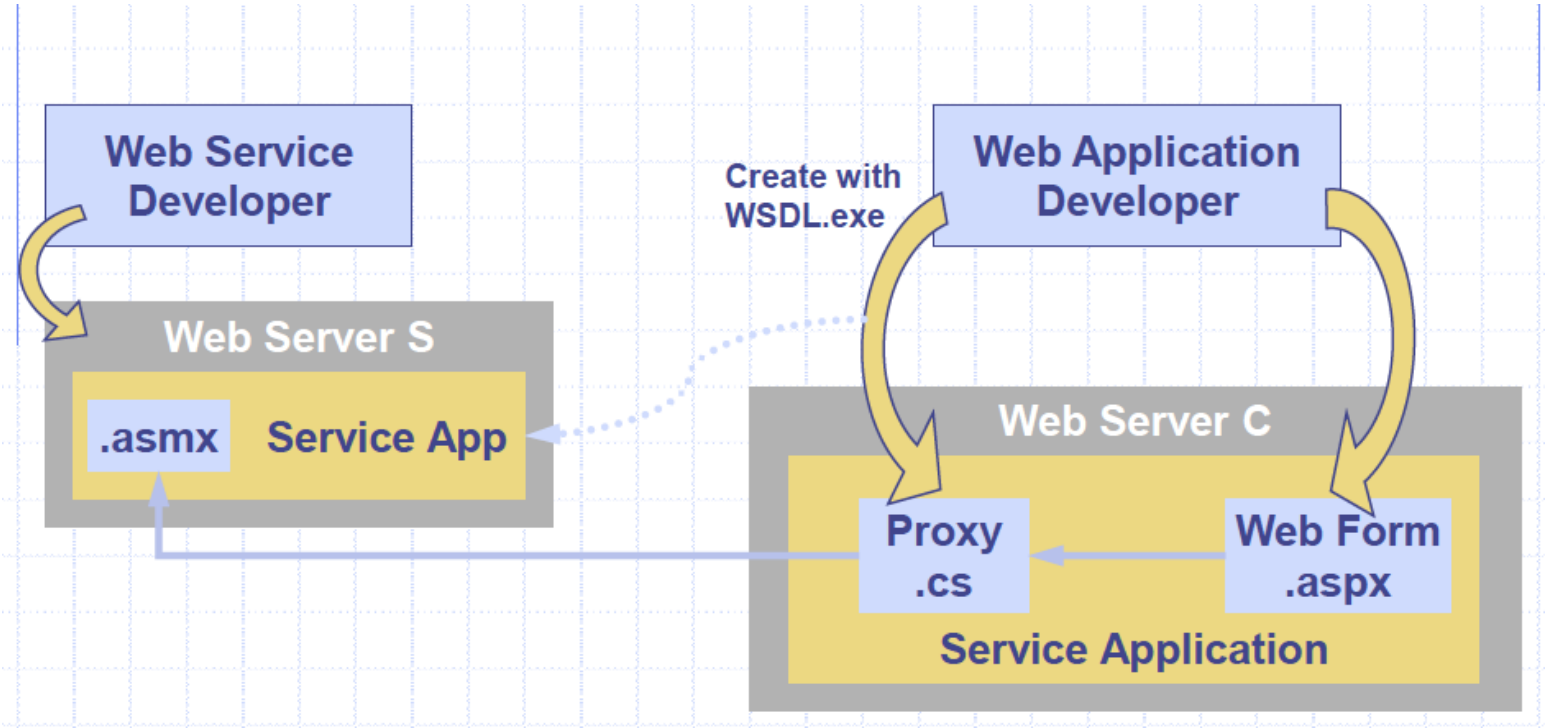
we have to provide Technology Independent Interface!!!

SOAP Web Services



Service Endpoint Interface (SEI) is generated out of WSDL to be used by App.
It covers all the complexity of web service (converting objects and access to web service into SOAP message).

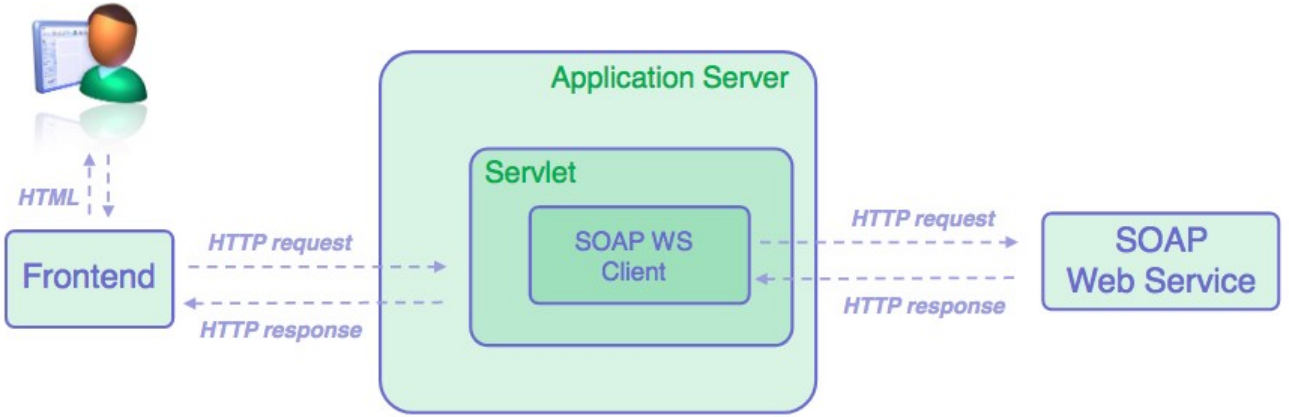
Consuming Web Services



JAVA Servlets

Component-based, platform-independent method for building Web-based applications.

- extension to a server that enhances the server's functionality.
- normal Java class which extends from the **javax.servlet.http.HttpServlet** class
- In the class : override the **doPost** or **doGet** method depending on the type of requests this Servlet needs to answer to
- In the method : call methods on the request and response objects to get the query parameters, set status codes and write the response itself.
- use for a servlet : extend a web server by providing dynamic web content.



Tutorial : https://www.codejava.net/java-ee/servlet/handling-html-form-data-with-java-servlet

Client SOAP Web Service : Demo

Web Service: Number Convertor

Service is available at : *https://www.dataaccess.com/webservicesserver/numberconversion.wso*
WSDL is available at : *https://www.dataaccess.com/webservicesserver/numberconversion.wso?WSDL*

wsimport

Tool to parse an existing Web Services Description Language (WSDL) file and generate :
required files (JAX-WS portable artifacts) for web service client to access the published web services

wsimport example : *http://www.mkyong.com/webservices/jax-ws/jax-ws-wsimport-tool-example/*

Structure of Web Service Client

myService Proxy code

Generated using disco.exe and wsdl.exe (see CalcClient.cs code comments)

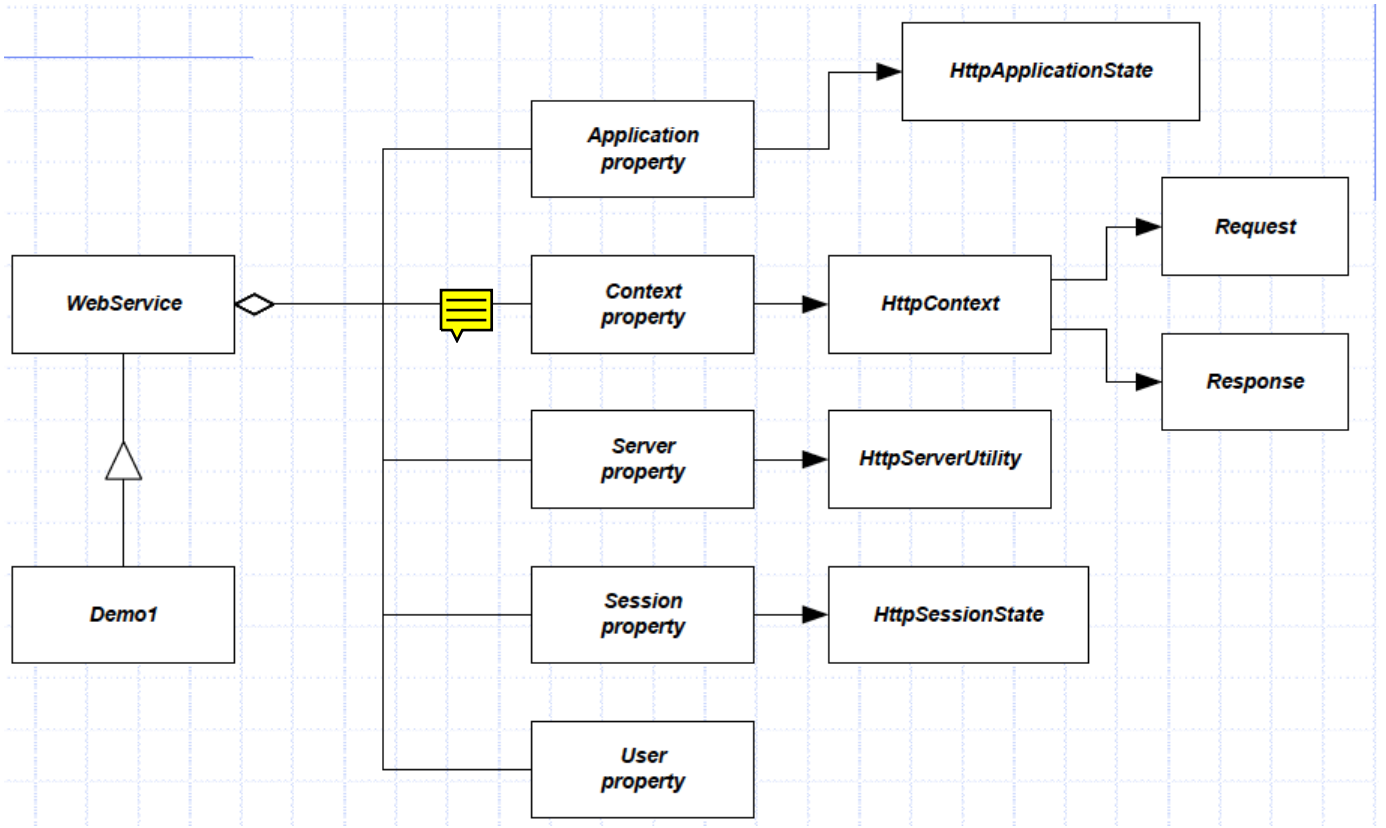
myServiceClient code

Ordinary ASP or Winform application

myService Proxy = new myService();

Result = Proxy.myMethod(args);

Web Service Application Structure



Web Service Properties

HttpApplicationState
- Share state among all users of an application.

HttpSessionState
- Share state from page to page for one user.

HttpContext
- Provides access to the server Request and Response objects.

HttpServerUtility
- Provides CreateObject, Execute, and MapPath methods.

User
- Supports authentication of user.

Web Methods

WebMethod methods can pass any of the C# and CLR types

User defined objects can also be passed if they are serializable:

- .Net XML serializer will not serialize non-public members
- User defined types can only be passed with SOAP. GET and POST won't work.
- The WSDL contract contains a schema description of any user defined objects passed by a WebMethod

Web Service Clients

Web Service Clients use Web Service proxies to communicate with the remote service:

```
// create proxy instance

demo1WebService.demo1 proxy = new
demo1WebService.demo1();

// use proxy
string result = proxy.demoMethod("string from client");
```

Web Services Versus Remoting

Web Services:

- Can be used by any platform that understands XML, SOAP, and WSDL.
  - Metadata (types) provided by WSDL
- Hosted by IIS and inherits ASP's security model.
- Uses HTTP protocol so accessible by web pages and can pass through most firewalls.
- Can only pass a limited set of user-defined objects
  - Can't serialize an object graph or all .Net containers.

Web Services Versus Remoting

Remoting:

- Requires .Net platform on client as well as server.
- Requires custom security (notoriously hard to get right).
- Metadata provided by assembly.
  - Can pass any .Net type, including object graphs and all .Net containers.
  - Rich, but none portable types.

Service Oriented Architecture (SOA)

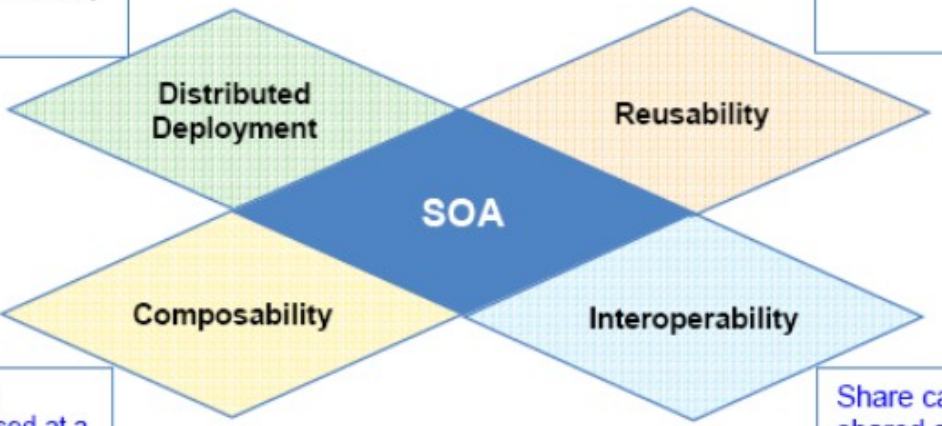
- Framework (method of software development) that provides a set of fundamental operations via web services to create business applications.
- A client/server design approach in which an application consists of software services and software service consumers (also known as clients or service requesters).
- All applications based on that framework share the common services - Don't have to recreate the same functionality for each new application
- Can provide those same services to Partner businesses, suppliers, and customers
- Cloud-based services use web services for accessing platform facilities on cloud servers
- SOA differs from the more general client/server model in its definitive emphasis on loose coupling between software components, and in its use of separately standing interfaces (Gartner).

Service Oriented Architecture (SOA)

SOA is an architectural approach that allows to:

Expose enterprise data and business logic as loosely coupled, discoverable, structured, standards-based, coarse-grained, stateless units of functionality called services

Choose a services provider and access to existing resources exposed as services



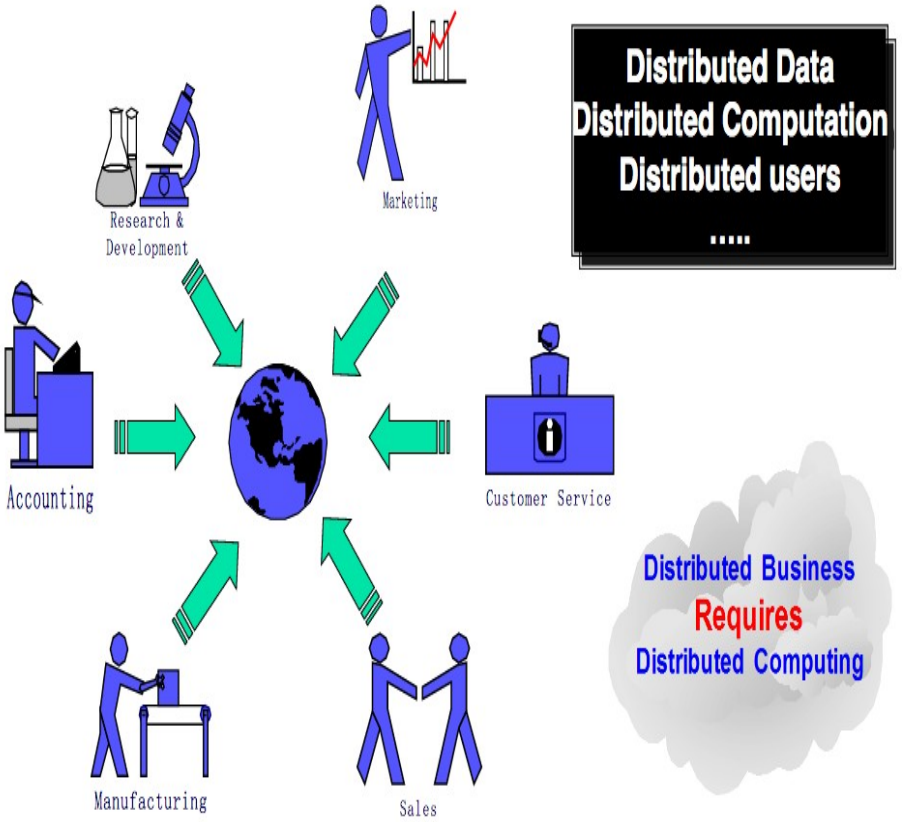
Assemble new processes from existing services that are exposed at a desired granularity through well-defined, published and standards-compliant interfaces

Share capabilities and reuse shared services across a network irrespective of underlying protocols or implementation technology

Why SOA ?

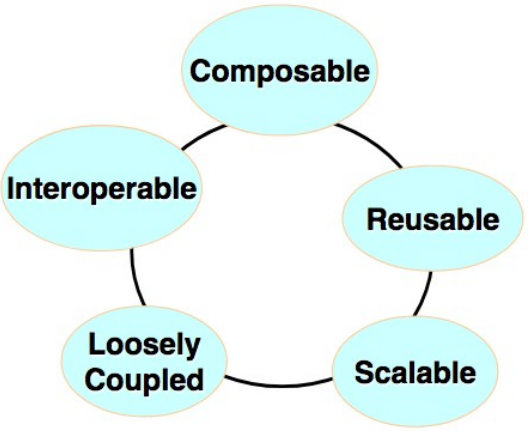
Interoperation issues

- Heterogeneous network protocols
- Heterogeneous hardware platforms
- Heterogeneous operating systems
- Heterogeneous application formats
- .....
- Increased Competitions
- Enhancement of Business Capabilities
- There must be consensus On Interoperability

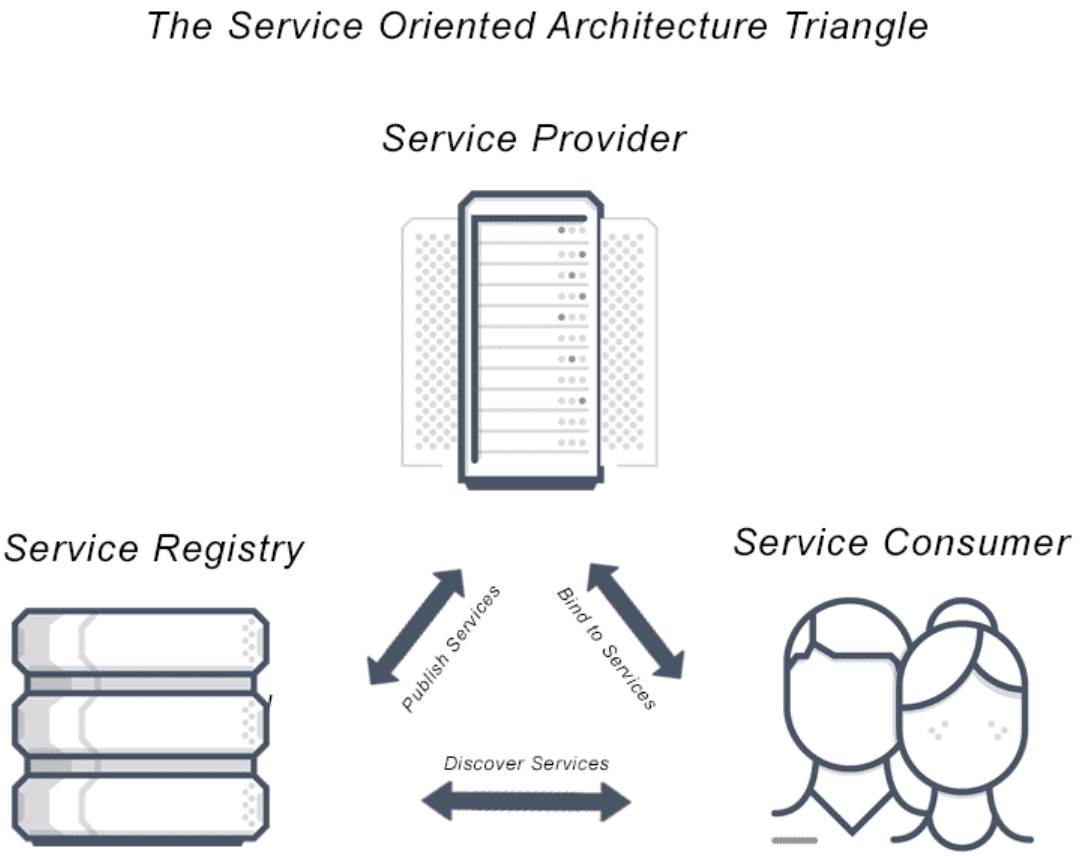


Service Oriented Architecture Characteristics

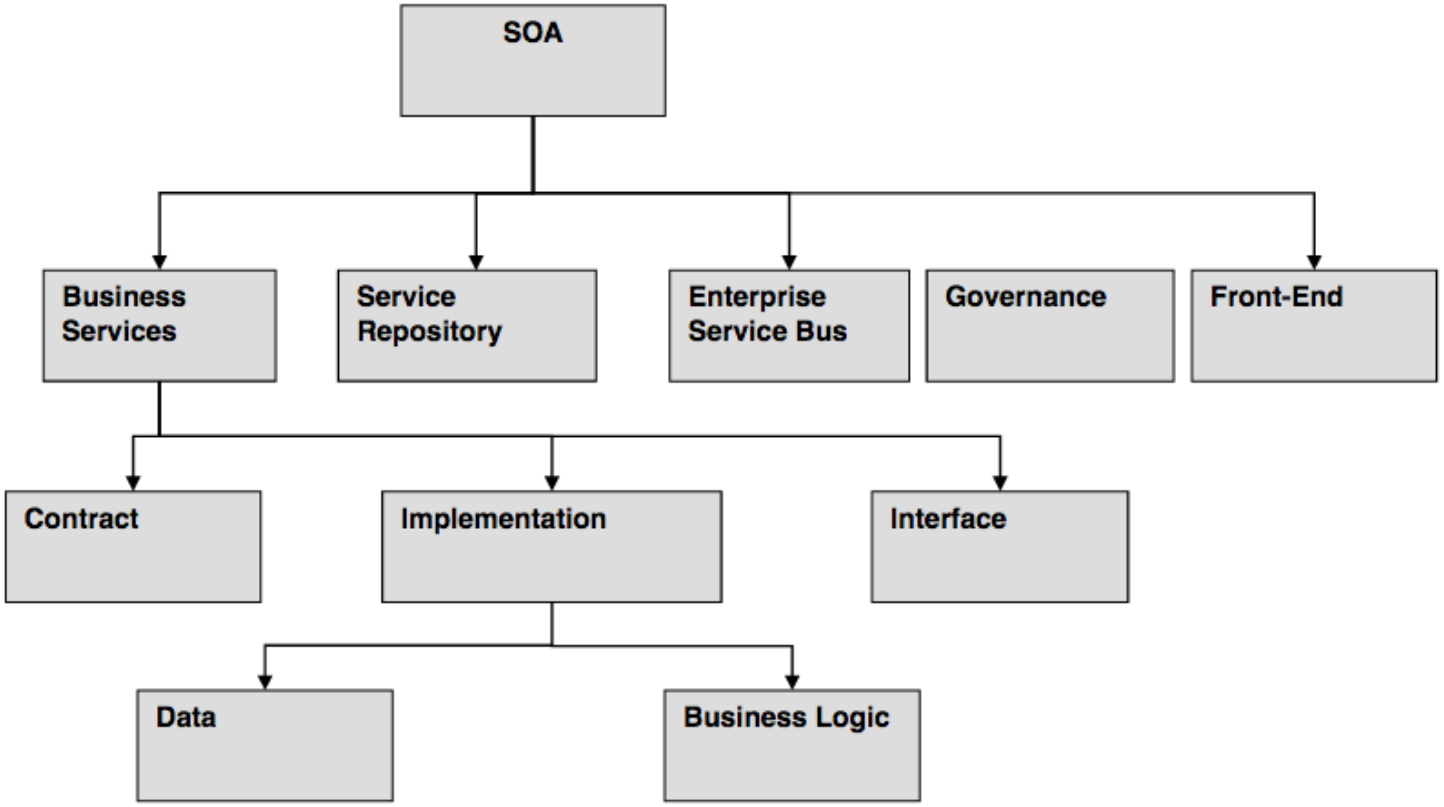
- Services are platform independent, self describing interfaces (XML)
- Messages are formally defined
- Services can be discovered
- Services have quality of service characteristics defined in policies
- Services can be provided on any platform
- Can be governed



Service Oriented Architecture (SOA)



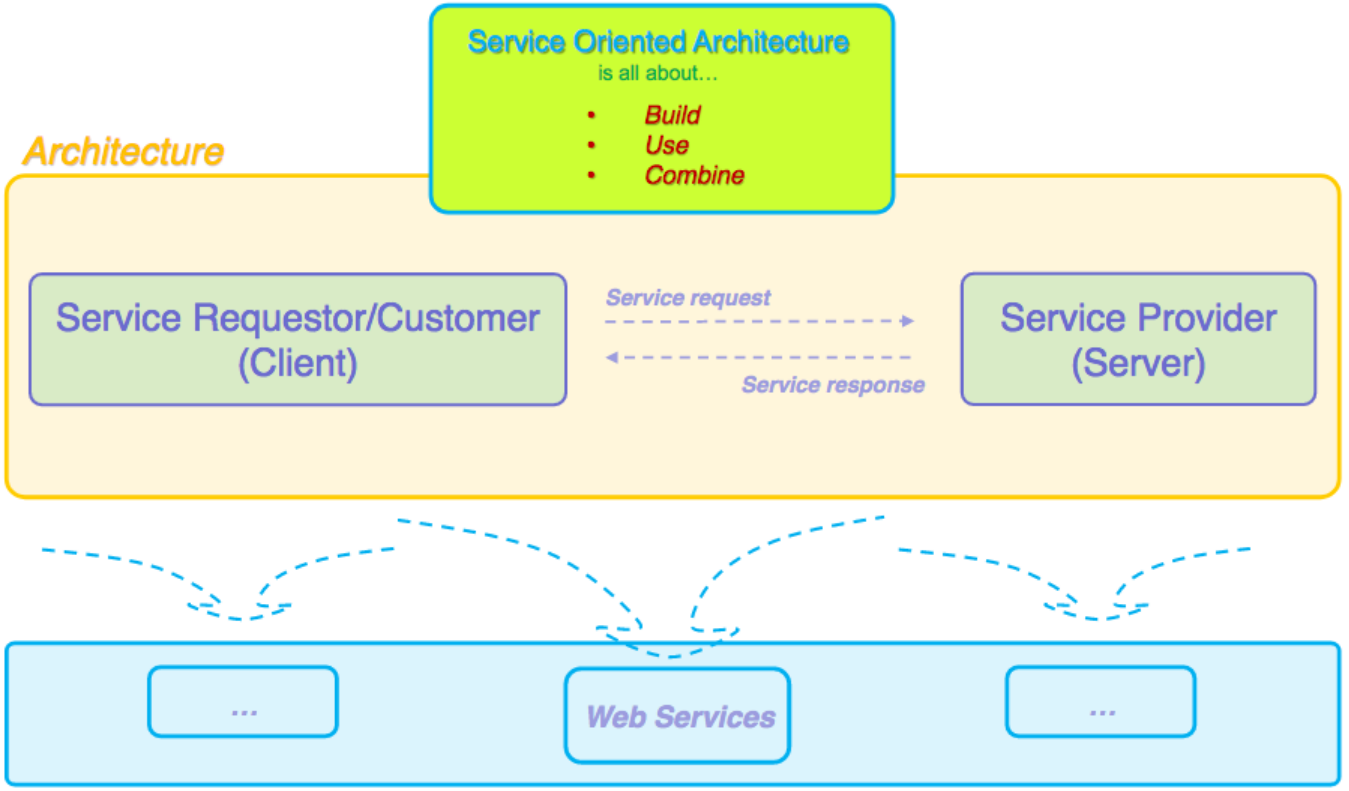
Key Components of SOA



Key Standards and Technology of SOA

XML	• Markup Language designed to carry/transport data • Structure of the document i.e. the tags can be user defined based on the data being transported
Web Services	• Loosely coupled software components delivered over Internet standard technologies
SOAP	• Message format communication between parties involved in a web service
WSDL	• Mechanism for describing a web service in a platform independent way
UDDI	• Facilitates registration and organization of web service descriptions into a searchable directory

SOA and Web Services



Implementation of SOA

Register yourself on Google Classroom

For any queries : bibhas.ghoshal@iiita.ac.in

Web : profile.iiita.ac.in/bibhas.ghoshal