WEBSERVICE ARCHITECTURE

List of Topics

What are web services?

- Web Services can Convert your application into a web application which can publish its function or message to the rest of the world.
- The basic Web Services platform is XML+HTTP
- A application which run on web (Internet or Intranet) and provides generic services.
- The services provided are through the web and in a standardized format which makes it generic and independent on the platform or the protocol on which the service was requested.

- Web Services are open standard (XML, SOAP,HTTP etc) based web applications that interact with other web applications for the purpose of exchanging data.
- These are two types of Web services
- SOAP (JAX-WS, Java API for XML Web Services)
- REST (JAX-RS, Java API for RESTful Web Services)

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.

Why Web Services?

- Exposing the existing function on to network:
- A web service is a unit of managed code that can be remotely invoked using HTTP requests. So, Web Services allows you to expose the functionality of your existing code over the network.
- once it is exposed on the network, other application can use the functionality of your program.
- Connecting Different Applications i.e. Interoperability
- Web Services allows different applications to talk to each other and share data and services among themselves. Other applications can also use the services of the web services. For Example, VB or .NET application can talk to java web services and vice versa.
- So, Web Services is used to make the application platform and technology independent.

Standardized Protocol:

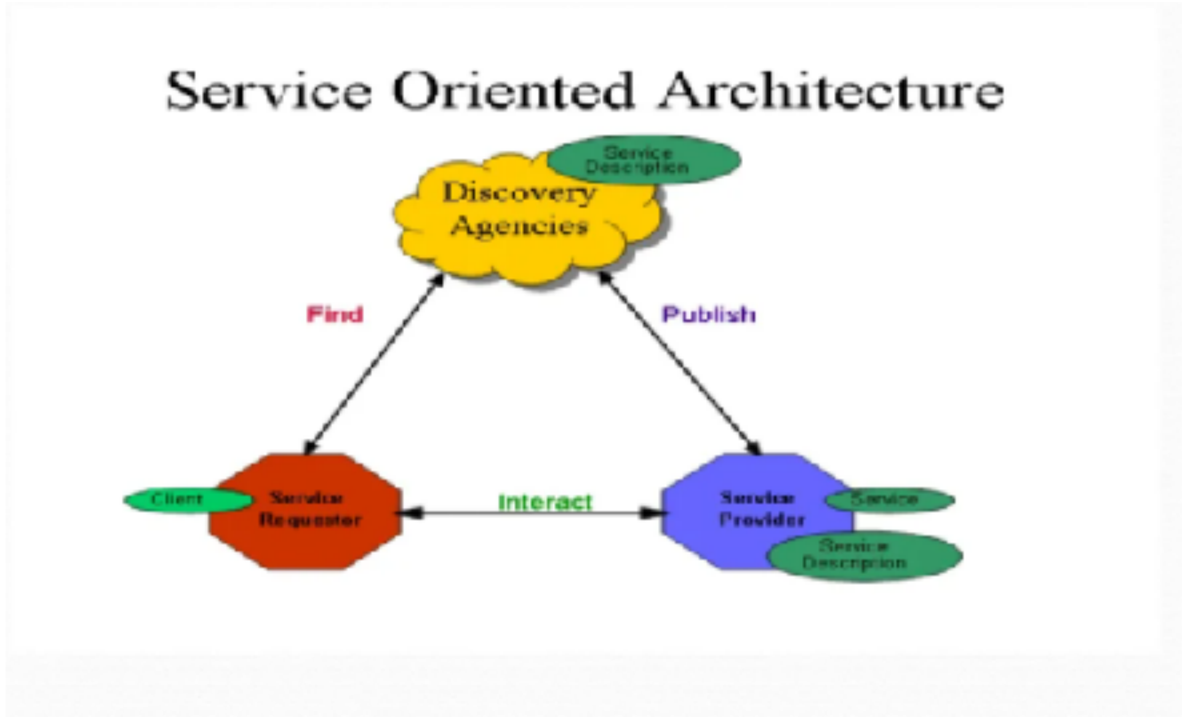
Web services uses standardized industry standard protocol for the communication. All the four layers (Service Transport, XML Messsaging, Service Description and Service Discovery layers) uses the well defined protocol in the web services protocol stack. This standardization of protocol stack gives the business many advantages like wide range of choices, reduction in the cost due to competition and increase in the quality.

Low Cost of Communication:

Web Services uses SOAP over HTTP protocol for the communication, so you can use your existing low cost internet for implementing web services. Beside SOAP over HTTP, Web Services can also be implemented on other reliable transport mechanisms like FTP etc.

Architecture

Service Oriented Architecture



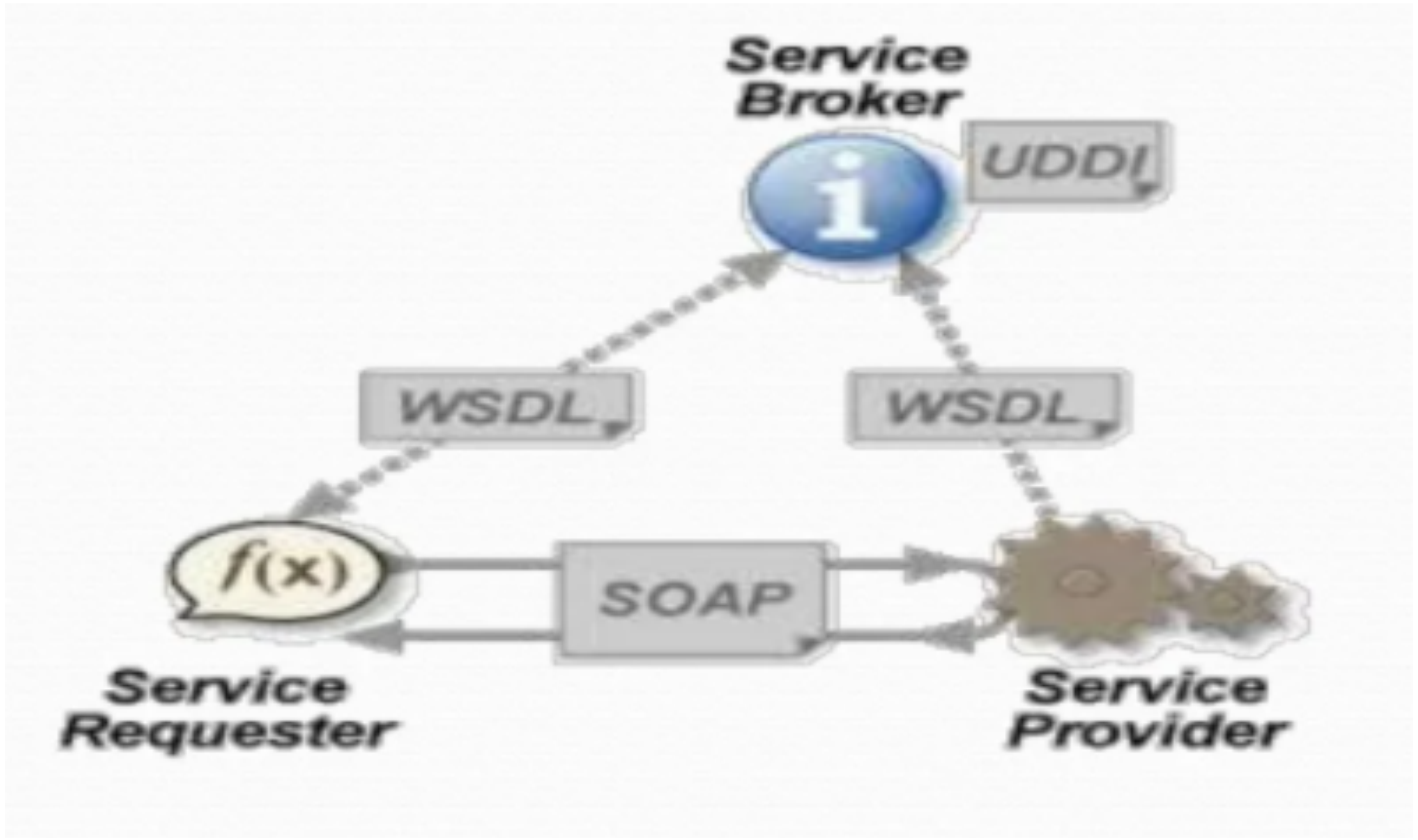
Web Service roles

There are three major roles within the web service architecture

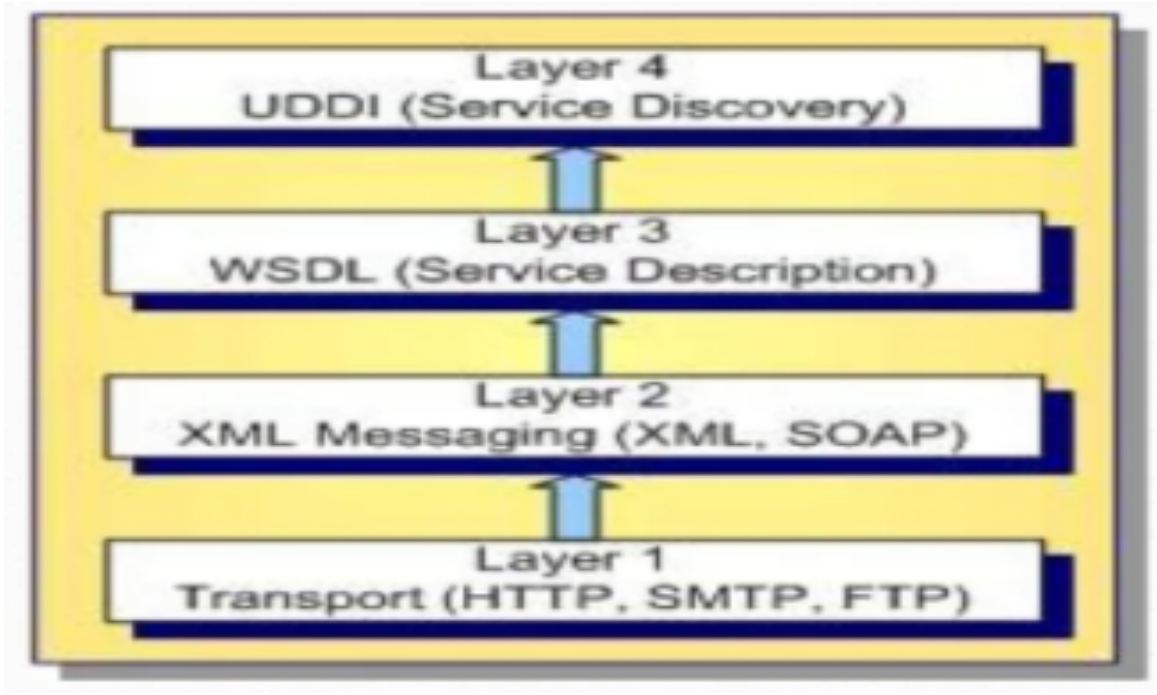
Service provider: This is the provider of the web service. The service provider implements the service and makes it available on the Internet.

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.


Service Broker
UDDI
WSDL
WSDL
f(x)
SOAP
Service Requester
Service Provider

Web service protocol stack


Layer 4
UDDI (Service Discovery)
Layer 3
WSDL (Service Description)
Layer 2
XML Messaging (XML, SOAP)
Layer 1
Transport (HTTP, SMTP, FTP)

Service transport: This layer is responsible for transporting messages between applications. Currently, this layer includes hypertext transfer protocol (HTTP), Simple mail Transfer Protocol (SMTP), file transfer protocol (FTP), and newer protocols, such as Blocks Extensible Exchange Protocol (BEEP)

XML messaging: This layer is responsible for encoding messages in a common XML format so that messages can be understood at either end. Currently, this layer includes XML-RPC and SOAP.

Service description: This layer is responsible for describing the public interface to a specific web service. currently, service description is handled via the Web Service Description Language (WSDL).

Service Discovery: This layer is responsible for centralizing services into a common registry, and providing easy publish/find functionality. Currently, service discovery is handled via Universal Description, Discovery and Integration (UDDI).

Components of Web Service

SOAP (Simple Object Access Protocol)

Protocol based on XML, used for message transfer

WSDL (web service description language)

XML file used to describe the Web service and how to access them.

UDDI (Universal Description and Discovery Integration)

used to register and search for web service

Directory of web service.

JAX-RPC

For intercommunication

HTTP

For message transfer

What is SOAP?

SOAP is an XML-based protocol to let applications exchange information over HTTP or more simple: SOAP is a protocol for accessing a web service.

- SOAP stands for Simple Object Access Protocol.
- SOAP is a communication protocol.
- SOAP is a format for sending messages.
- SOAP is designed to communicate via Internet.
- SOAP is platform independent.
- SOAP is language independent.
- SOAP is based on XML.
- SOAP is simple and extensible.
- SOAP allows you to get around firewalls.
- SOAP is a W3C standard.

What is WSDL?

- WSDL is an XML-based language for locating and describing Web Services.
- WSDL stands for Web Service Description Language.
- WSDL is based on XML.
- WSDL is used to describe Web Services.
- WSDL is used to locate Web Services.
- WSDL is a W3C standard.

What is UDDI?

UDDI is a directory service where companies can register and search for Web services.

- UDDI stands for Universal Description, Discovery and Integration.
- UDDI is a directory for storing information about web services.
- UDDI is a directory of web service interfaces described by WSDL.
- UDDI communicates via SOAP.

XML-RPC

This is the simplest XML based protocol for exchanging information between computers.

XML-RPC is a simple protocol that uses XML messages to perform RPCs.

Requests are encoded in XML and sent via HTTP POST.

XML responses are embedded in the body of the HTTP response.

XML-RPC is platform-independent.

XML-RPC allows diverse applications to communicate.

XML-RPC is the easiest way to get started with web services.

Security

Confidentiality

If a client sends an XML request to a server, then question is that can we ensure the communication remains confidential?

Answer lies here

XML-RPC and SOAP run primarily on top of HTTP.

HTTP has support for secure sockets layer (SSL).

Communication can be encrypted via the SSL.

SSL is a proven technology and widely deployed.

Authentication

If a client connects to a web service, how do we identify the user? And is the user authorized to use the service?

Following options can be considered but there is no clear consensus on a strong authentication scheme.

HTTP includes built-in support for basic and digest authentication, and services can therefore be protected in much the same manner as HTML documents are currently protected.

SOAP Security Extensions: Digital Signature (SOAP-DSIG). DSIG leverages public key cryptography to digitally sign SOAP messages. This enables the client or server to validate the identity of the other party.

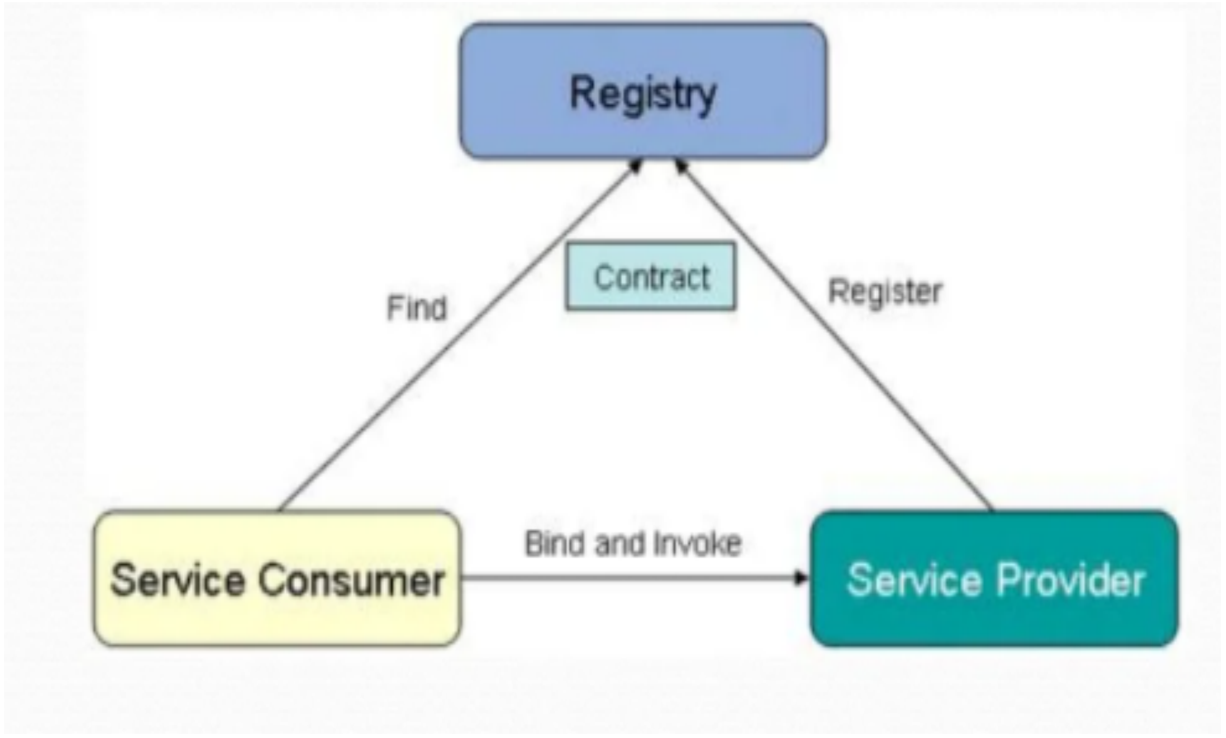
The Organization for the Advancement of structured Information Standards (OASIS) is working on the Security Assertion Markup Language (SAML, It is an XML-based open standard data format for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider).

Network Security

There is currently no easy answer to this problem, and it has been the subject of much debate. For now, if you are truly intent on filtering out SOAP or XML-RPC messages, one possibility is to filter out all HTTP POST requests that set their content type to text/xml.

How web service is implemented?

Build and Publish
Find
Bind and Invoke



Web Service Implementation

Pre Conditions

No one will be able to use your service

Step 1 : Build and Publish

Build

Create your application

Create your contract file (WSDL)

Publish

Register your application as a web service onto any registry

This process happens on UDDI using a separate SOAP request.

This process is useful only if your web service should be accessible using Internet.

Post Conditions

Your web service will be available to the public (If published to any registry) or will be accessible to intranet users.

Pre Conditions

The user needs to access a service but is not aware of the service details.

Step 2 : Find

Find

Search in the registry for a service which provides your needs

Obtain the necessary details about the service'

Post Conditions

The user will have all the details about the service and gets ready to contact the service.

Pre conditions

The user will have the contract necessary to identify all the service

Step 3: Bind

Bind

Use the contract file to build the request message.

Invoke

Send a request to the service and request for the necessary operation available from the service.

The request should be sent in the protocol which is required by the service.

Post Conditions

The user will receive the response from the service in a format specified in the contract file.