**WEB SERVICE:**

Web service is a standardized medium to propagate communication between the client and server applications on the world wide web. A web service is a software module that is designed to perform a certain set of tasks. The main component of a web service is the data which is transferred between the client and the server, and that is XML(easy to understand intermediate language that is understood by many language). There are mainly two different type of webservices(way to transfer this XML file through the internet), SOAP webservices and RESTful web services.

Some advantages of web services:

* Exposing business functionality on the network
* Interoperability amongst applications
* A standardized protocol which everybody understands
* Reduction in cost of communication

**SOAP(Simple Object Access Protocol):**

SOAP is based on transferring XML data as SOAP Messages and it is all sent via HTTP. Each of this soap message have to follow a specific XML structure but the content shouldn’t be altered. Each SOAP message structure consist of: Envelope element, which is the root element and the first element in the XML document. Header, which contains the routing data which is basically the information which tells the XML document to which client it needs to be sent to. Body, that contain the actually message client requested. SOAP have two communication model called Marshalling and Demarshalling. The client would format the information regarding the procedure call and any arguments into a SOAP message and send it to the server as part of an HTTP request as Marshalling. The server would then unwrap the message sent by the client, see what the client requested for an then send the appropriate response back to the client as a SOAP message is Demarshalling.

**WSDL:**

WSDL are used so the client application would be able to understand where the webservice is located and how it can be utilized. Below are main WSDL declarations:

* <message>: Used to define the different data elements for each operation performed by the webservice.
* <portType>: Describes the operation which can be performed by the web service.

Client access these WSDL file through UDDI, standard for describing, publishing, and discovering the web services that are provided by a particular service provider.

**REST(Representational State Transfer):**

Restful web service, expose API from your application in a secure, uniform, stateless manner to the calling client. The calling client can perform predefined operations using the restful service. REST uses the HTTP protocol.

REST is a way to access resources which lie in a particular environment. For example, you could have a server that could be hosting important documents or pictures or videos. All of these are an example of resources. If a client, say a web browser needs any of these resources, it has to send a request to the server to access these resources. Now REST defines a way on how these resources can be accessed. Key element of restful elements are shown below:

* Resources: UID of a specific resource when requesting an information
* Request Verbs: describe what you want to do with the resource. Get,Put,post, delete, etc..
* Request Headers: Additional instructions sent with the request. These might define the type of response required or the authorization details.
* Request body: normall sent with a post request where you want to tell the webservice that it wants to add a resource to the server. In this case, the body will contain information of that.
* Response body: main body of the response with data requested.
* Response status code: returned along with the response of web server. Ex code 200 returned when there is no error.

Why do we want to keep our web services RESTful?

* Heterogeneous languages and environment using XML(same as SOAP)
* The event of device does not matter.
* Everything is moving toward cloud now, and all of it is REST based, so it is better to make it RESTful.

Principle of RESTful services

* RESTful client server: server will have a RESTful web service which would provide the required functionality to the client.
* Stateless: it's up to the client to ensure that all the required information is provided to the server. This is required so that server can process the response appropriately. Every request is atomic.
* Cache: The cache is a concept implemented on the client to store requests which have already been sent to the server. So if the same request is given by the client, instead of going to the server, it would go to the cache and get the required information.
* Layered System: The concept of a layered system is that any additional layer such as a middleware layer can be inserted between the client and the actual server hosting the RESTFul web service
* Interface/uniform contract: uses key verbs to work with resources on the server.
* Code on demand: REST allows extension of functionality by the downloading of script.