Assignment No. 7

Aim: Create a simple web service and write any distributed application to consume the web service.

Objective: To understand web services, and how distributed applications can be developed to consume web services.

Infrastructure: Python environment.

Software Requirements: Python 3.0, Flask, request library

Theory:

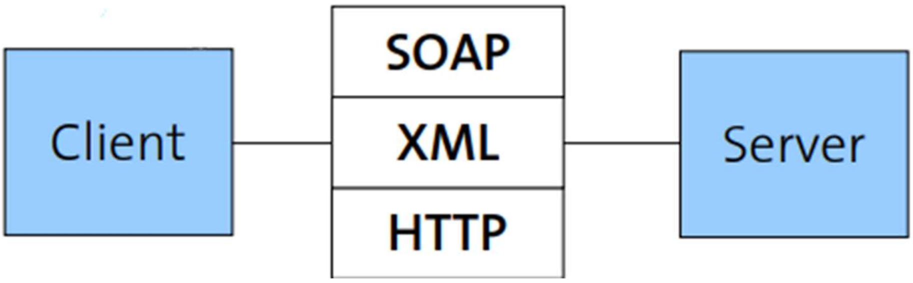
What are Web Services?

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.

Components of Web Services

The basic web services platform is XML + HTTP. All the standard web services work using the following components −

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)



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.

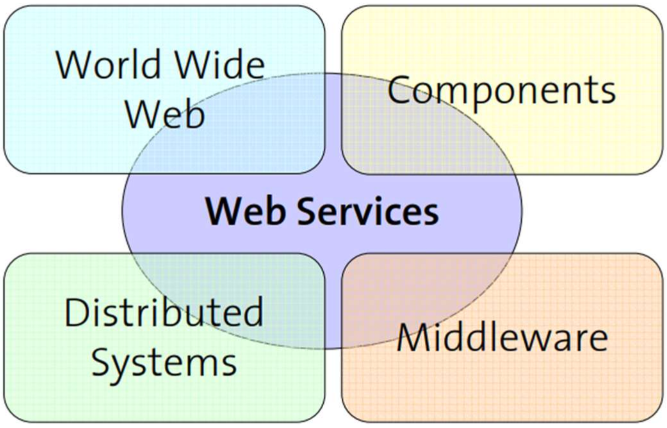
Types of Web Services

There are mainly two types of web services.

(I) SOAP web services.
(II) RESTful web services.

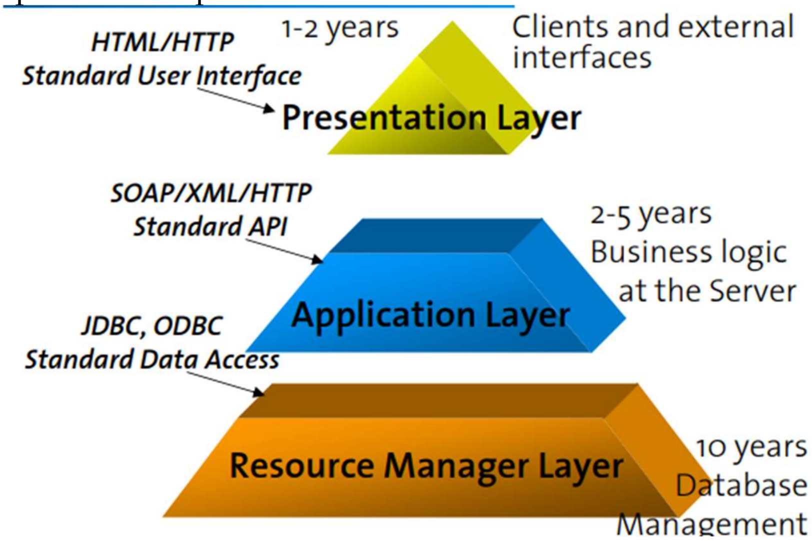
Distributed Systems and Web Services

- Web services provide standards for developing large scale distributed system.
- Web services on the path of success while CORBA distributed objects failed (This is nothing technical, only a matter of widespread industry acceptance)



Layers in Distributed Systems

- Client is any user or program that wants to perform an operation over the system. To support a client, the system needs to have a presentation layer through which the user can submit operations and obtain a result.
- The application logic establishes what operations can be performed over the system and how they take place. It takes care of enforcing the business rules and establish the business processes. The application logic can be expressed and implemented in many different ways: constraints, business processes, server with encoded logic ...
- The resource manager deals with the organization (storage, indexing, and retrieval) of the data necessary to support the application logic. This is typically a database but it can also be a text retrieval system or any other data management system providing querying capabilities and persistence.



Steps involved in development of Web service, and a distributed application to utilize this Web services are:

1. Setting up the Web Service:

- Choose a programming language and framework for your web service.
- Define the functionality and endpoints of your web service. For simplicity, let's assume you want to create a basic calculator API with two endpoints: POST /add and POST /multiply.

2. Deploying the Web Service:
   You can deploy the web service on any cloud platform.

Alternatively, you can run a local server of the web service.

3. Building the Distributed Application:
    - Define the functionality of your distributed application. In this case, you'll create an application that consumes the calculator API endpoints.
    - Implement the logic to make HTTP requests to the web service endpoints. You can use libraries like axios in JavaScript or requests in Python to send HTTP requests.
    - Parse the responses from the web service and handle any errors that may occur.
4. Test and Run the Distributed Application:
    - Set up the development environment for your distributed application.
    - Run the distributed application and ensure it consumes the web service correctly.
    - Debug and fix any issues that may arise.

Conclusion:

We learnt:

- how the web service works,
- how to use web service in a distributed application,
- implementation of web service and distributed application.