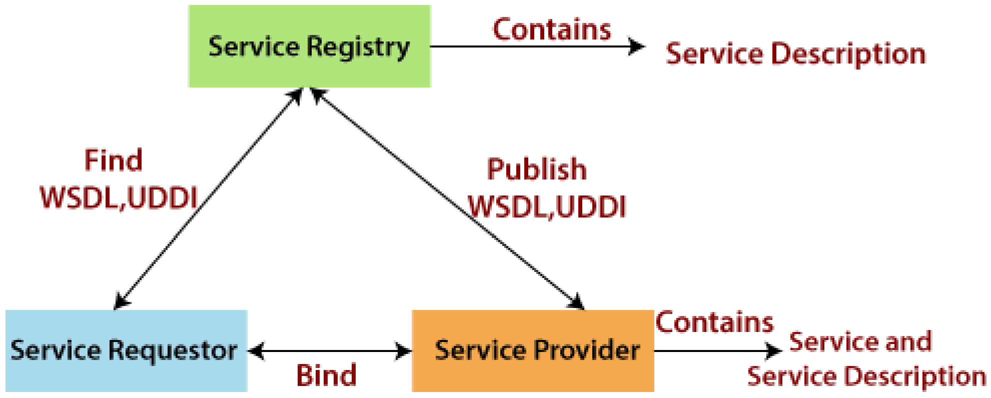
Architecture of Web Services

The Web Services architecture describes how to instantiate the elements and implement the operations in an interoperable manner.

The architecture of web service interacts among three roles: **service provider, service requester,** and **service registry**. The interaction involves the three operations: **publish, find,** and **bind**. These operations and roles act upon the **web services artifacts**. The web service artifacts are the web service software module and its description.

The service provider hosts a network-associable module (web service). It defines a service description for the web service and publishes it to a service requestor or service registry. These service requestor uses a find operation to retrieve the service description locally or from the service registry. It uses the service description to bind with the service provider and invoke with the web service implementation.

The following figure illustrates the operations, roles, and their interaction.



Web Service Roles, Operations and Artifacts

Roles in a Web Service Architecture

There are three roles in web service architecture:

- Service Provider
- Service Requestor
- Service Registry

Service Provider

From an architectural perspective, it is the platform that hosts the services.

Service Requestor

Service requestor is the application that is looking for and invoking or initiating an interaction with a service. The browser plays the requester role, driven by a consumer or a program without a user interface.

Service Registry

Service requestors find service and obtain binding information for services during development.

Operations in a Web Service Architecture

Three behaviors that take place in the microservices:

- Publication of service descriptions **(Publish)**
- Finding of services descriptions **(Find)**
- Invoking of service based on service descriptions **(Bind)**

Publish: In the publish operation, a service description must be published so that a service requester can find the service.

Find: In the find operation, the service requestor retrieves the service description directly. It can be involved in two different lifecycle phases for the service requestor:

- At design, time to retrieve the service's interface description for program development.
- And, at the runtime to retrieve the service's binding and location description for invocation.

Bind: In the bind operation, the service requestor invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact, and invoke the service.

Artifacts of the web service

There are two artifacts of web services:

- Service
- Service Registry

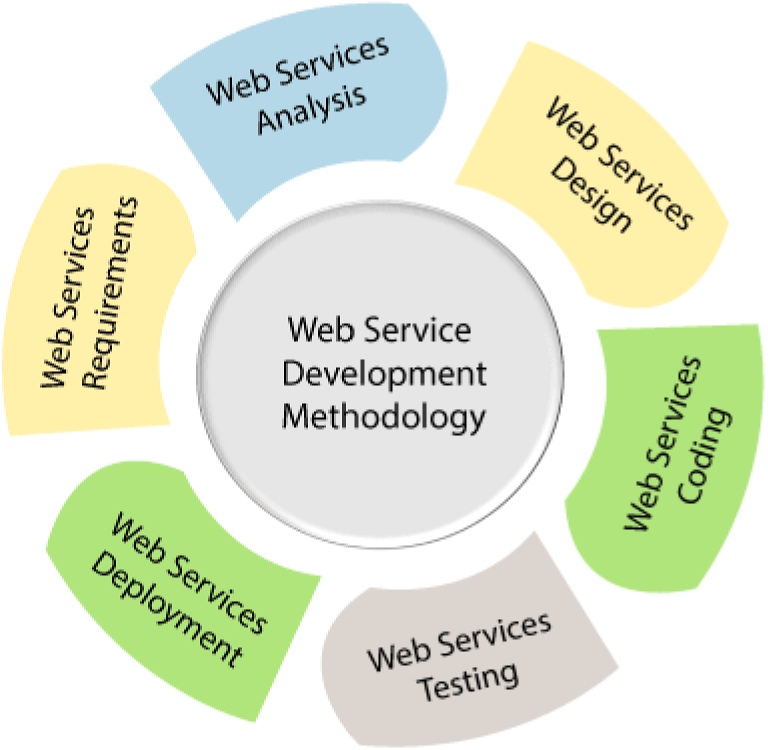
Service: A service is an **interface** described by a service description. The service description is the implementation of the service. A service is a software module deployed on network-accessible platforms provided by the service provider. It interacts with a service requestor. Sometimes it also functions as a requestor, using other Web Services in its implementation.

Service Description: The service description comprises the details of the **interface** and **implementation** of the service. It includes its **data types, operations, binding information,** and **network location**. It can also categorize other metadata to enable discovery and utilize by service requestors. It can be published to a service requestor or a service registry.

Web Service Implementation Lifecycle

A web service implementation lifecycle refers to the phases for developing web services from the requirement to development. An Implementation lifecycle includes the following phases:

- Requirements Phase
- Analysis Phase
- Design Phase
- Coding Phase
- Test Phase
- Deployment Phase



Web Service Implementation Lifecycle

Requirements Phase

The objective of the requirements phase is to understand the business requirement and translate them into the web services requirement. The requirement analyst should do requirement elicitation (it is the practice of researching and discovering the requirements of the system from the user, customer, and other stakeholders). The analyst should interpret, consolidate, and communicate these requirements to the development team. The requirements should be grouped in a centralized repository where they can be viewed, prioritized, and mined for interactive features.

Analysis Phase

The purpose of the analysis phase is to refine and translate the web service into conceptual models by which the technical development team can understand. It also defines the high-level structure and identifies the web service interface contracts.

Design Phase

In this phase, the detailed design of web services is done. The designers define web service interface contract that has been identified in the analysis phase. The defined web service interface contract identifies the elements and the corresponding data types as well as mode of interaction between web services and client.

Coding Phase

Coding and debugging phase is quite similar to other software component-based coding and debugging phase. The main difference lies in the creation of additional web service interface wrappers, generation of WSDL, and client stubs.

Test Phase

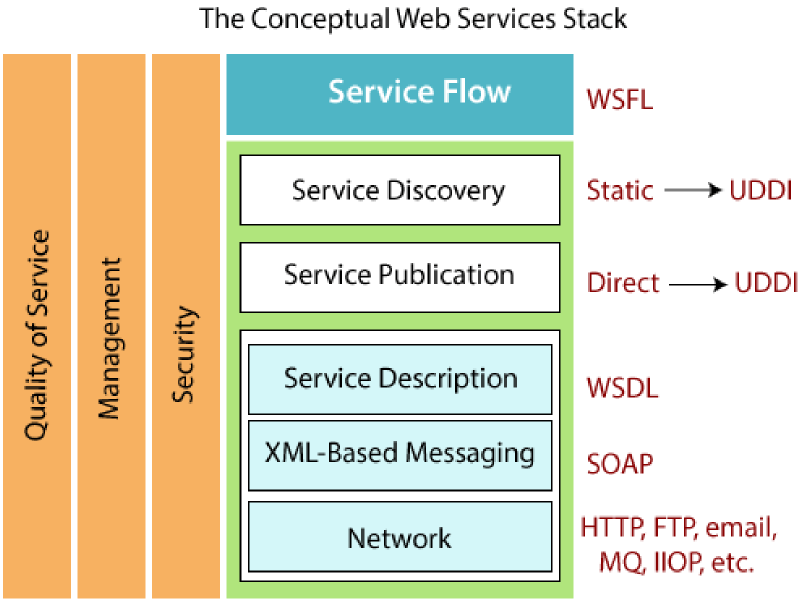
In this phase, the tester performs interoperability testing between the platform and the client's program. Testing to be conducted is to ensure that web services can bear the maximum load and stress. Other tasks like profiling of the web service application and inspection of the SOAP message should also perform in the test phase.

Deployment Phase

The purpose of the deployment phase is to ensure that the web service is properly deployed in the distributed system. It executes after the testing phase. The primary task of deployer is to ensure that the web service has been properly configured and managed. Other optional tasks like specifying and registering the web service with a UDDI registry also done in this phase.

Web Service Stack or Web Service Protocol Stack

To perform three operations: publish, find, and bind in an interoperable manner, there must be a **web service stack**. The web service stack embraces the standard at each level.



Web Services conceptual Stack

In the above figure, the top most layers build upon the capabilities provided by the lower layers. The three vertical towers represent the requirements that are applied at every level of the stack. The text on the right represents technologies that apply at that layer of the stack. A web service protocol stack typically stacks four protocols:

- Transport Protocol
- Messaging Protocol
- Description Protocol
- Discovery Protocol

(Service) Transport Protocol: The network layer is the foundation of the web service stack. It is responsible for transporting a message between network applications. HTTP is the network protocol for internet available web services. It also supports other network protocol such as **SMTP, FTP,** and **BEEP** (Block Extensible Exchange Protocol).

(XML) Messaging Protocol: It is responsible for encoding message in a common XML format so that they can understand at either end of a network connection. SOAP is the chosen XML messaging protocol because it supports three operations: publish, find, and bind operation.

(Service) Description Protocol: It is used for describing the public interface to a specific web service. WSDL is the standard for XML-based service description. WSDL describes the interface and mechanics of service interaction. The description is necessary to specify the **business context, quality of service,** and **service-to-service** relationship.

(Service) Discovery Protocol: It is a centralized service into a common registry so that network Web services can publish their location and description. It makes it easy to discover which services are available on the network.

The first three layers of the stack are required to provide or use any web service. The simplest stack consists of HTTP for the network layer, SOAP protocol for the XML-based messaging, and WSDL for the service description layer. These three-layer provides interoperability and enables web service to control the existing internet infrastructure. It creates a low cost of entry to a global environment.

The bottom three layers of the stack identify technologies for compliance and interoperability, the next two layer- **Service Publication** and **Service Discovery** can be implemented with a range of solutions.