**What do you mean by Web Service?**

Web services, as the name suggests, is simply a software system that is specially designed to propagate communication between the client and server applications on WWW (World Wide Web). In simple words, it is the method of communication among two or more devices over a network. It allows multiple applications built on different programming languages to communicate with each other without any trouble. It uses the internet for direct application-to-application interaction, and also allows you to expose business logic using API.

**Explain different types of Web Services.**

There are basically two types of web services:

**SOAP (Simple Object Access Protocol) Web Services**: It is also referred to as transport-independent messaging protocol whose main purpose is to transfer a message, and is based on XML protocol.   
  
**RESTful (Representational State Transfer) Web Services**: It is developed to fulfill the shortcomings of SOAP and to make the web services more effective.

**What are the important features of Web services?**

Some of the important features of web services include:

* Used to standardized XML messaging system.
* Not tied to any one programming language or any operating system.
* Discoverable via a simple find mechanism.
* Available over the internet or private networks.
* Supports loosely coupled connections between systems.
* Can be synchronous or asynchronous.
* Supports the transparent exchange of data to facilitate business integration.
* Supports communication among different apps with HTML, XML, WSDL, SOAP, etc.
* Supports RPC (Remote Procedure Calls).

**What are the different components of Web Services?**

There are various components of web services as given below:

* SOAP (Simple Object Access Protocol)
* UDDI (Universal Description, Discovery, and Integration)
* WSDL (Web Services Description Language)
* RDF (Resource Description Framework)
* XML (Extensible Markup Language)

**Write the difference between API and Web services.**

**API (Application Programming Interface)**: It acts as an interface between two devices so that they can communicate with each other without any user intervention. Some of its features include customizable, easy integration with GUI, time effective, language-independent, etc. All APIs are not web services.   
**Web Service**: It facilitates interaction between two devices over a network. They are widely used for exchanging data among systems or applications. Some of its features include loosely coupled, supports document exchange, interoperability, extensibility, etc. All Web services are APIs.

| **API** | **Web Service** |
| --- | --- |
| It can be online or offline. | It must use a network. |
| They are lightweight architecture. | They require SOAP to send and receive network data therefore, are not lightweight architectures. |
| It can use any design style or protocol. | It can only use SOAP but sometimes UDDI, XML, RPC, and REST also. |
| It supports HTTP/HTTPS protocol and also supports XML and JSON. | It supports HTTP protocol and also supports XML. |
| It doesn’t require any network for its operation. | It requires a network for its operation. |
| They are open source and are used for XML. | They are not open source and are used to understand JSON (JavaScript Object Notation). |

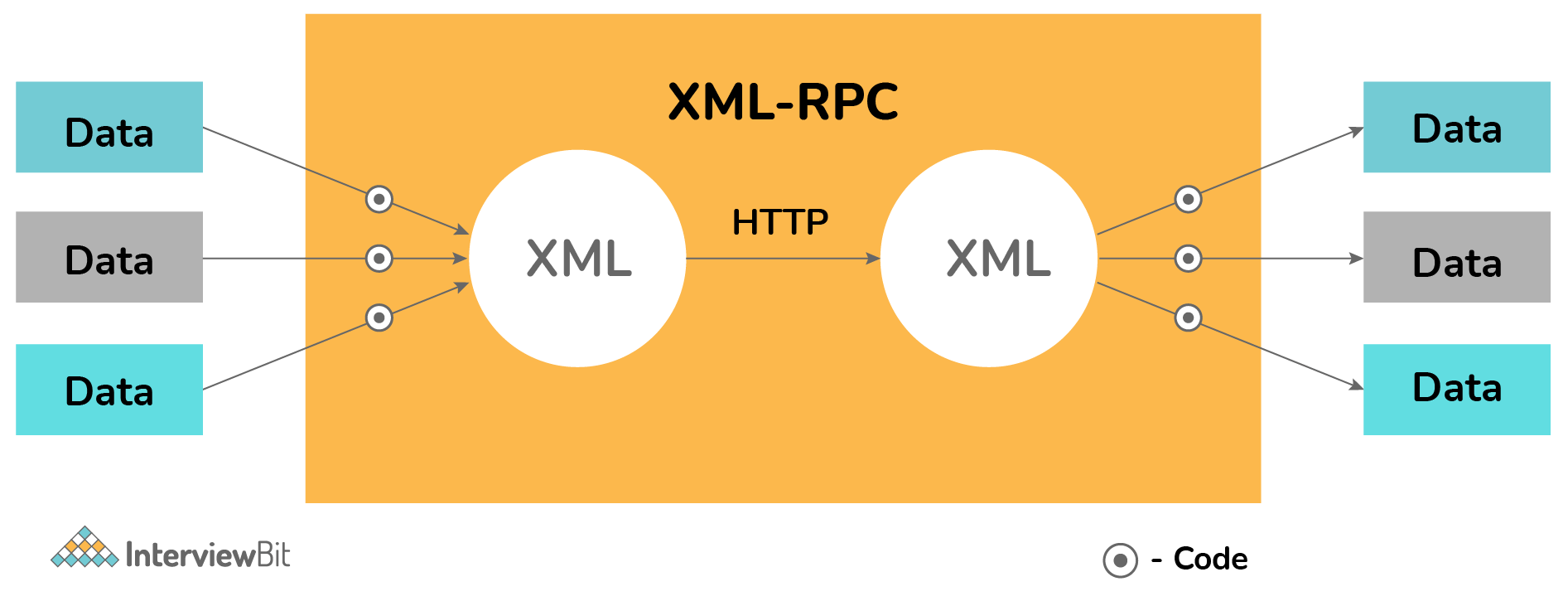
**What is WSDL?**

WSDL (Web Services Description Language), as the name suggests, is considered the standard format that is used to describe the availability of web services and how to access them. It is based on XML protocol for exchanging data in decentralized and distributed environments. It also describes the technical details or locates the user interface to the web service. WSDL document contains some important information as given below:

* Method name and parameters
* Port types
* Service endpoint
* Header information, etc.

**What is XML-RPC?**

XML-RPC (Remote procedure call) is considered the most basic and simplest XML-based protocol to exchange data among different devices on a network. It uses HTTP as a transport protocol for quickly and easily transferring the information or data between two devices. XML-RPC can also be used with different programming languages such as Perl, Java, Python, C, C++, PHP, etc.



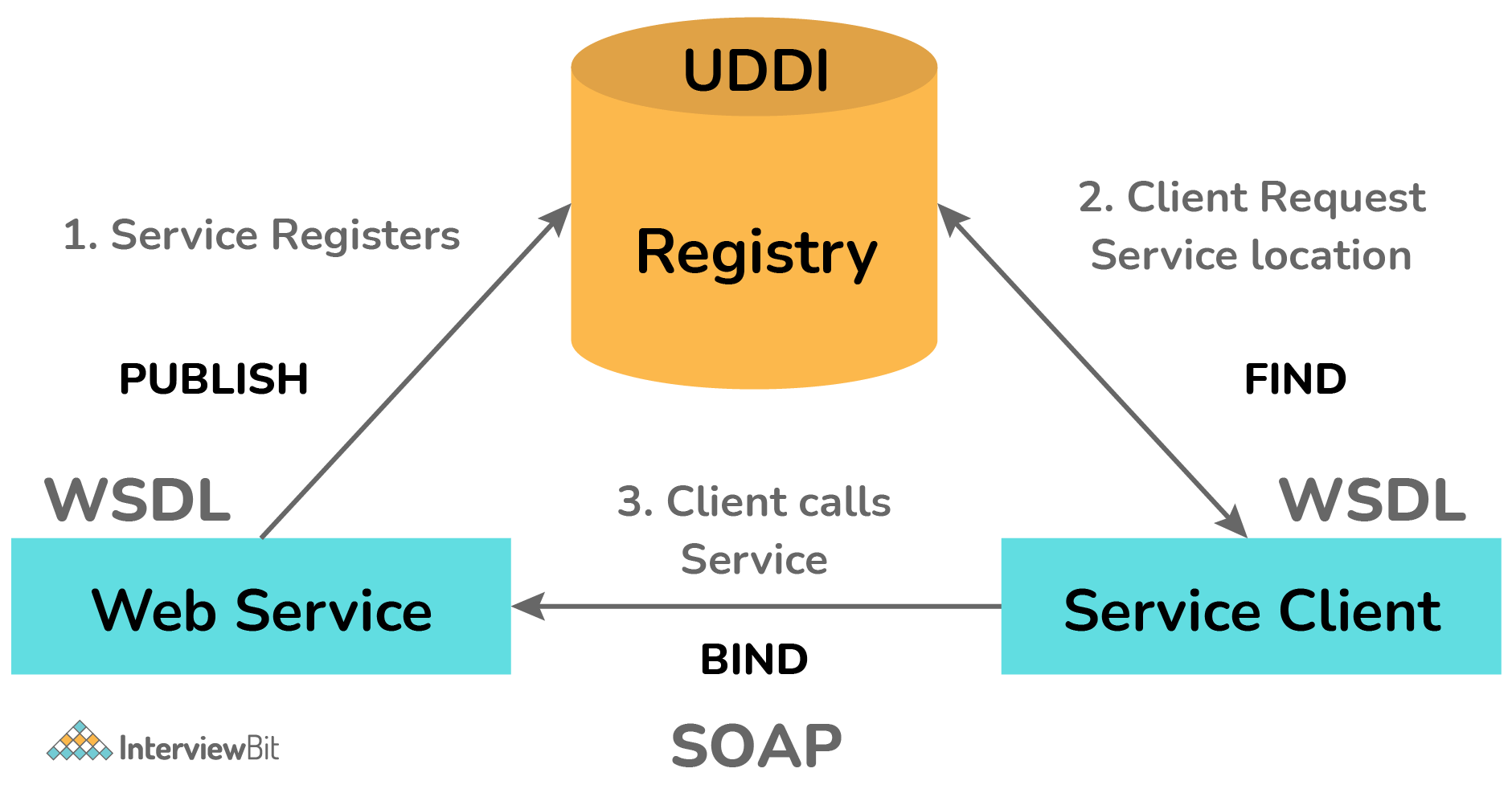
**What are the features of XML-RPC?**

There are various features of XML-RPC that includes:

* Platform independent
* Allows diverse applications to communicate
* Considered as the easiest and simplest way to get started with web services
* Uses XML to encode its calls and HTTP as the transport protocol

**What do you mean by UDDI?**

UDDI (Universal Description, Discovery, and Integration) is a directory service used for describing, publishing, and finding web services. It is also used for creating business registries. It is based on a set of web standards including HTTP, XML, SOAP, WSDL, XML Schema. Its main goal is to streamline digital transactions and e-commerce among company systems.



**10. What are the important features of UDDI?**

Some important features of UDDI include:

* Platform independent
* Uses WSDL for describing interfaces to web services
* Can communicate through SOAP, Java RMI, and CORBA Protocol
* The delineation between interface and implementation
* Neutral in terms of protocols

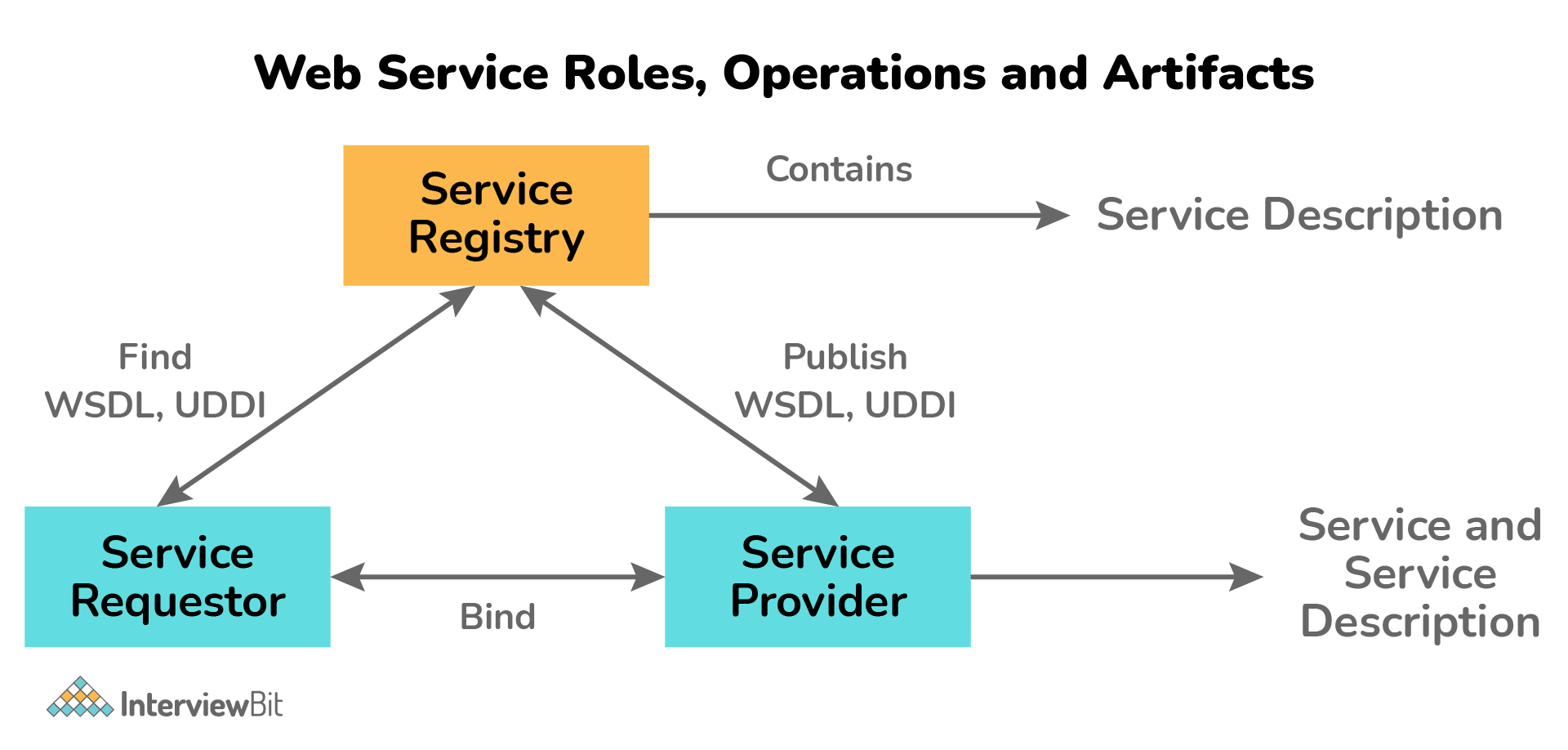
**Name the language that is commonly used by UDDI.**

The language commonly used by UDDI is WSDL (Web Service Description Language).

**Explain web service Architecture.**

Every framework requires some type of architecture to ensure that the entire framework works perfectly as desired, the same goes for web services. Web service architecture is used to assist the developer with steps and procedures that are essential to complete the creation. Web service architecture includes three distinct roles i.e., service provider, service requester, and service registry. It also includes three different operations that include:    
  
**Publish (Publication of Service Descriptions)**: A service description needs to be published so that the service requestor can locate and have access to it. It can be published anywhere depending upon the requirements of the application.   
  
**Find (Finding of Services Descriptions)**: A service description is retrieved directly by a service requestor. The requestor consults the broker to locate a web service that is already published.

**Bind (Invoking of Service based on Service Description)**: Every service needs to be invoked. To locate, contact, and invoke the service, the service requestor initiates the interaction with the service at runtime using details of binding in the service description.



**What is a Web Service Provider?**

Web service generally creates web services and provides access to the client application who wants it. Its main purpose is to implement the service and make it available on the Internet so that client applications can use it whenever required. In simple words, it is a platform that creates and hosts web services.

**What is a Web service Requestor?**

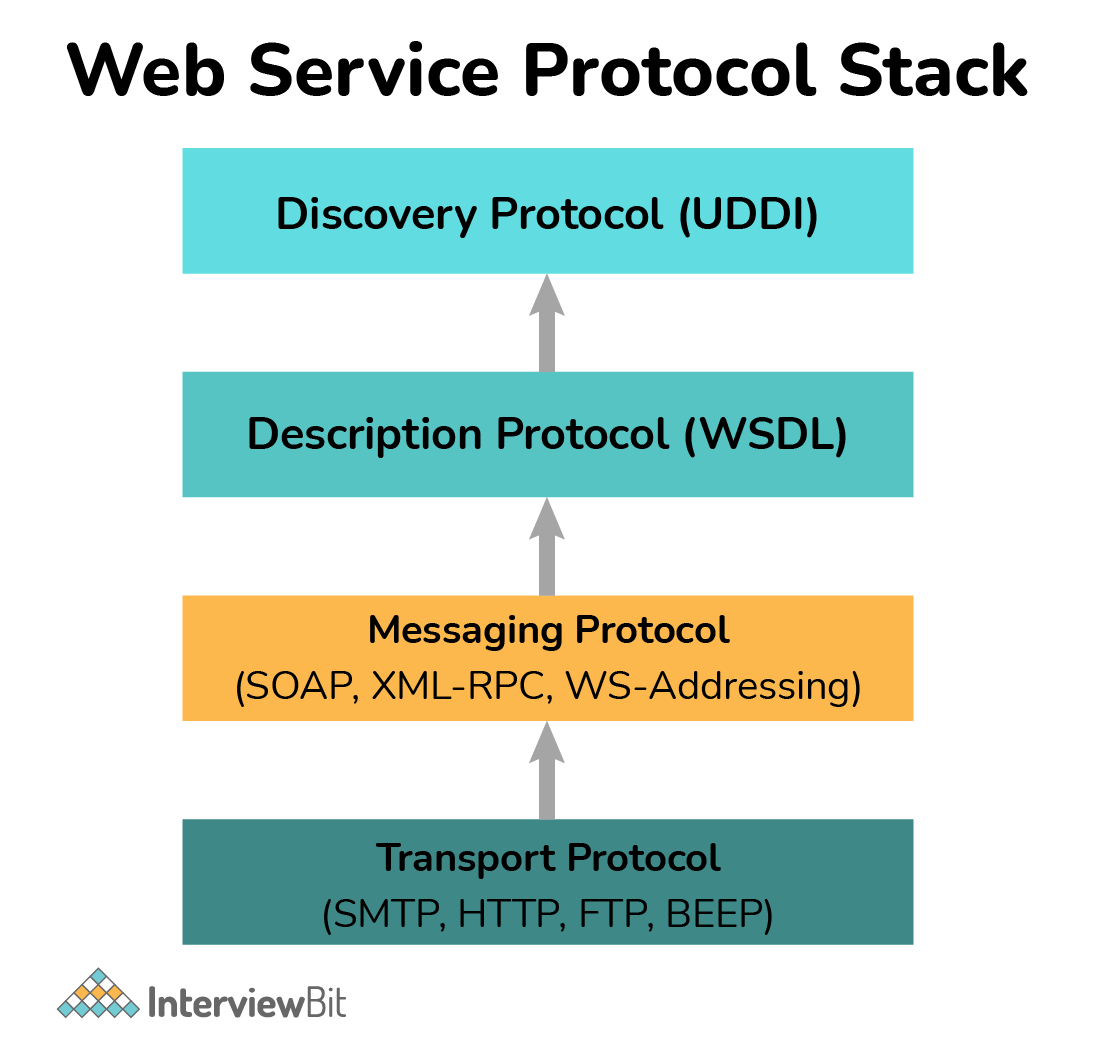
Web service requestor is the client application that requests for web service to use it. Its main purpose is to use an existing web service by opening a network connection and sending an XML request. In simple words, they are consumers of the web service.

**What is a Web Service Registry?**

The web service registry is basically like a ‘phone book’ for web services. It allows client applications to be able to publish new services or can locate the already existing ones. Two widely-used registry standards are generally supported by application servers i.e., ebXML (Electronic Business using XML) and UDDI (Universal Description, Discovery, and Integration).

**What are the different layers of the web service protocol stack?**

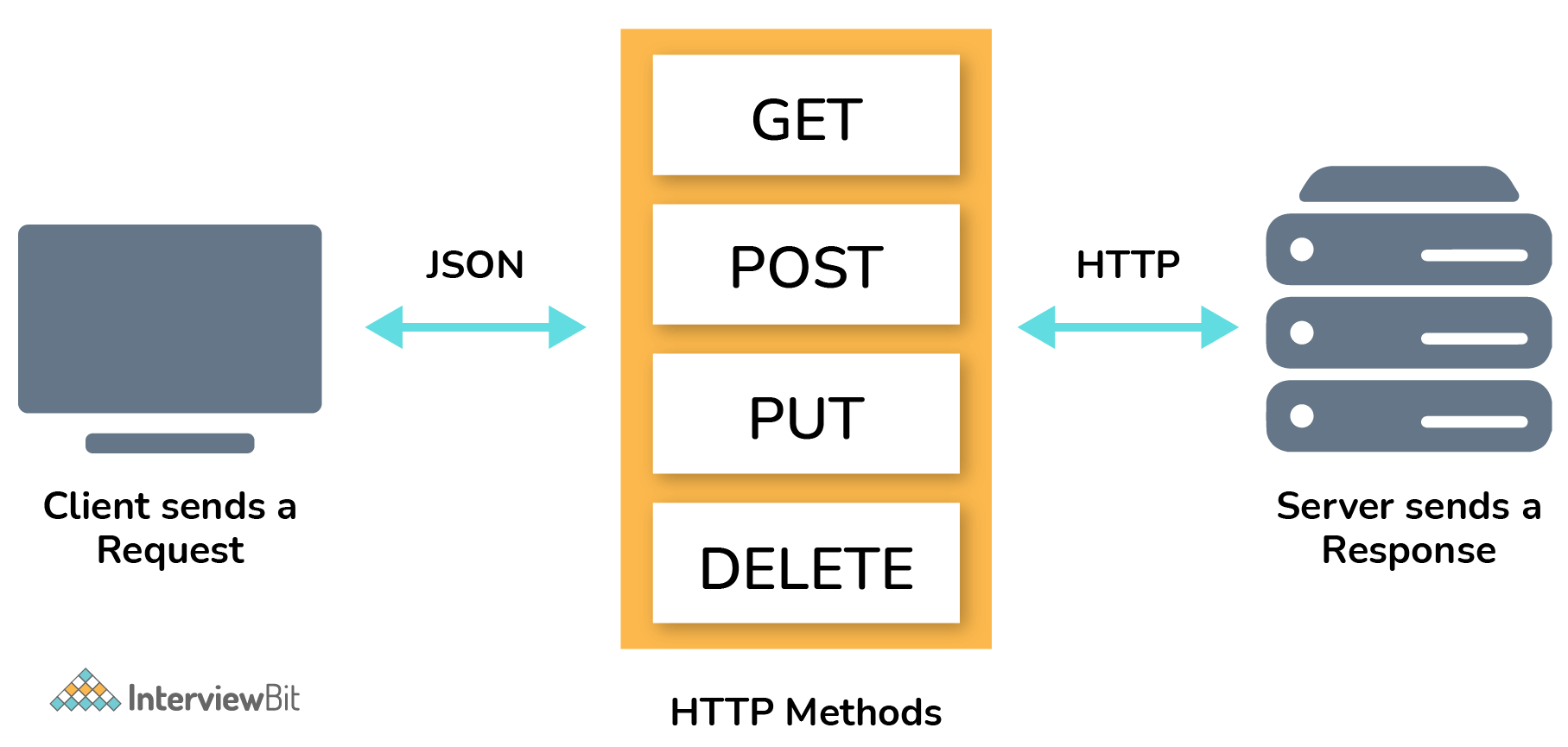
The implementation of web services generally depends on technologies that are often organized in a layered stack. Examining the web service protocol stack is considered as the second option for viewing the web service architecture. In simple words, it is the set of protocols that are used to explore and execute web services. Currently, the web service protocol stack has four layers as given below:



**Service Transport**: It is generally responsible to transport messages between applications. It basically defines technology standards for communication and allows messages or information to move across the network without any difficulty. It uses HTTP, SMTP, FTP, and DEEP protocols to transfer information.    
**XML Messaging**: It is generally responsible to encode messages into XML format so that one can understand these messages at either end. This layer generally includes XML-RPC and SOAP.    
**Service Description**: It is generally responsible to describe the public interface to a specific web service. WSDL is generally used to handle service descriptions.  
**Service Discovery**: It is generally responsible to centralize service into a common registry and provides easy functionality for publishing or finding web services. UDDI is generally used to handle service discovery.

**What are RESTful Web Services?**

REST (Representational State Transfer) is a stateless client-server architecture style used for developing applications that are accessible over the web. It is a type of web service whose main goal is to make web services more effective. It can be defined as the web service that uses HTTP methods for implementing the REST architecture. Unlike SOAP which is protocol-based, Restful services are architecturally based. It does not contain any contract or WSDL file.



**What are the advantages of RESTful web service?**

There are several advantages of RESTful web services as given below:

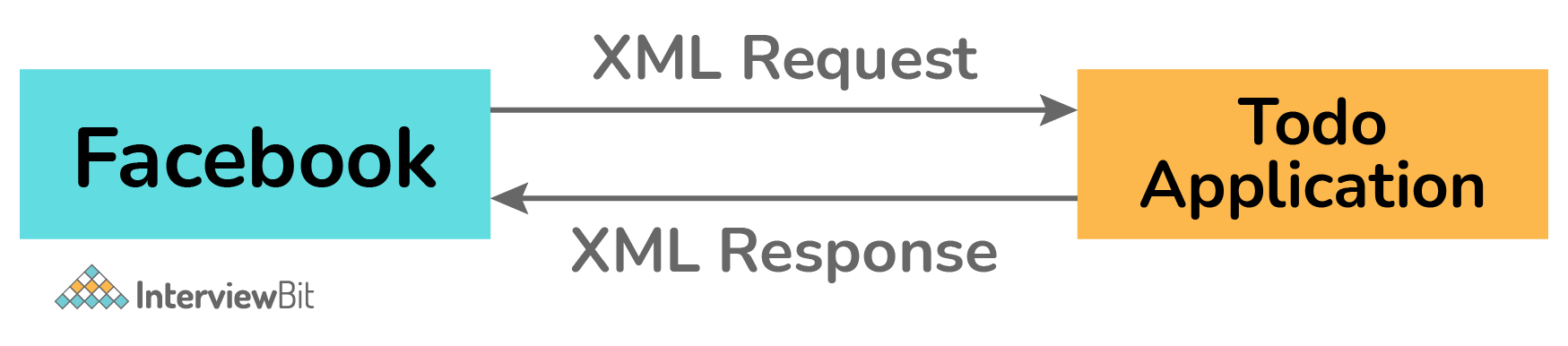
* Platform independent.
* Simple and easy to implement and test.
* Support different formats such as JSON, XML, HTML, etc.
* Can be written in different programming languages and executed on any platform.
* Lightweight, manageable, scalable, and reusable.
* Faster and provide better performance.
* Consume less bandwidth and resources.
* A lot of automation framework is available

**Which protocol is used by RESTful web services?**

The protocol used by RESTful web services is HTTP.

**What do you mean by SOAP? Write its advantages.**

SOAP (Simple Object Access Protocol) is an XML-based protocol that is used to access web services. It is simply used to interchange data or information between two devices or computers using request and response based on XML format over transport protocols like HTTP, SMTP, etc.



**Advantages:**

* Language and Information independent.
* Can be written in different programming languages or operating systems.
* Provide data transport for web services.
* Can extend HTTP for XML messaging.
* Defines and uses its own WS security.
* Easy to debug and eliminate firewall issues.

**What are the different elements of the SOAP Document or message?**

The SOAP message is basically an ordinary XML document consists of three parts as given below:   
  
**SOAP Envelope**: It is a mandatory element that identifies XML documents as a SOAP message. It simply defines the start and the end of the message.    
  
**SOAP Header**: It is an optional element that contains header information.    
  
**SOAP Body**: It is a mandatory element that contains call and response information. It includes XML data consisting of the message that is being sent.

The following block depicts the general structure of SOAP XML request and response:

**XML Request**

<**Envelope** xmlns=?http://schemas.xmlsoap.org/soap/envelop/?>

<**Body**>

<**getCourseDetailRequest** xmlns=?http://udemy.com/course?>

<**id**>course1</**id**>

<**getCourseDetailRequest**>

</**Body**>

</**Envelope**>

**XML Response**

<**SOAP-ENV:Envelope** xmlns:SOAP-ENV=?http://schemas.xmlsoap.org/soap/envelope/?>

<**SOAP-ENV:Header** /> <!?empty header-->

<**SOAP-ENV:Body**> <!?body begin-->

<**ns2:getCourseDetailsResponse** xmlns:ns2=?http://in28mi> <!--content of the response-->

<**ns2:course**>

<**ns2:id**>Course1</**ns2:id**>

<**ns2:name**>Spring<**ns2:name**>

<**ns2:description**>10 Steps</**ns1:description**>

</**ns2:course**>

</**ns2:getCourseDetailResponse**>

</**SOAP-ENV:Body**> <!?body end-->

</**SOAP-ENV:Envelope**>

**What do you mean by SOA?**

SOA (Service Oriented Architecture) is basically an architectural approach that is specially designed to support service orientation. It enabled services to communicate or interact across different platforms and languages to form applications. Applications in SOA are developed on the basis of services. It can be easily implemented using different protocols such as HTTP, JMS, HTTPS, RPC, RMI, etc.

**What are the advantages of using SOA**

Some of its advantages include:

* Easy to integrate.
* Services are platform-independent.
* Manage complexity so that integration becomes more manageable.
* Services are easier to test and debug.
* Easily available to any requester.

**Name three primary security issues of Web Services?**

The three primary security issues of web services include:

* Confidentiality
* Authentication
* Network Security

**Give an example of working for a Web Service Provider.**

**using** System;

**using** System.Web.Services;

**using** System.Xml.Serialization;

[WebService(Namespace="http://localhost/MyWebServices/")]

**public** **class** **FirstService** : **WebService**

{

[WebMethod]

**public** int **Add**(int a, int b)

{

**return** a + b;

}

[WebMethod]

**public** String **SayHello**()

{

**return** "Hello World";

}

}

**What is the importance of URI in REST based web services?**

In REST-based web services, URI (Uniform Resource Locator) is generally used to locate resources on the server that is hosting the web service. Each resource in service will have at least one URI that is used to identify it. Web services clients generally use URI to access the resource. Its format is given below:

<protocol>://<service-name>/<ResourceType>/<ResourceID>