Web Services

By

Anand Kulkarni

anand.pune38@gmail.com

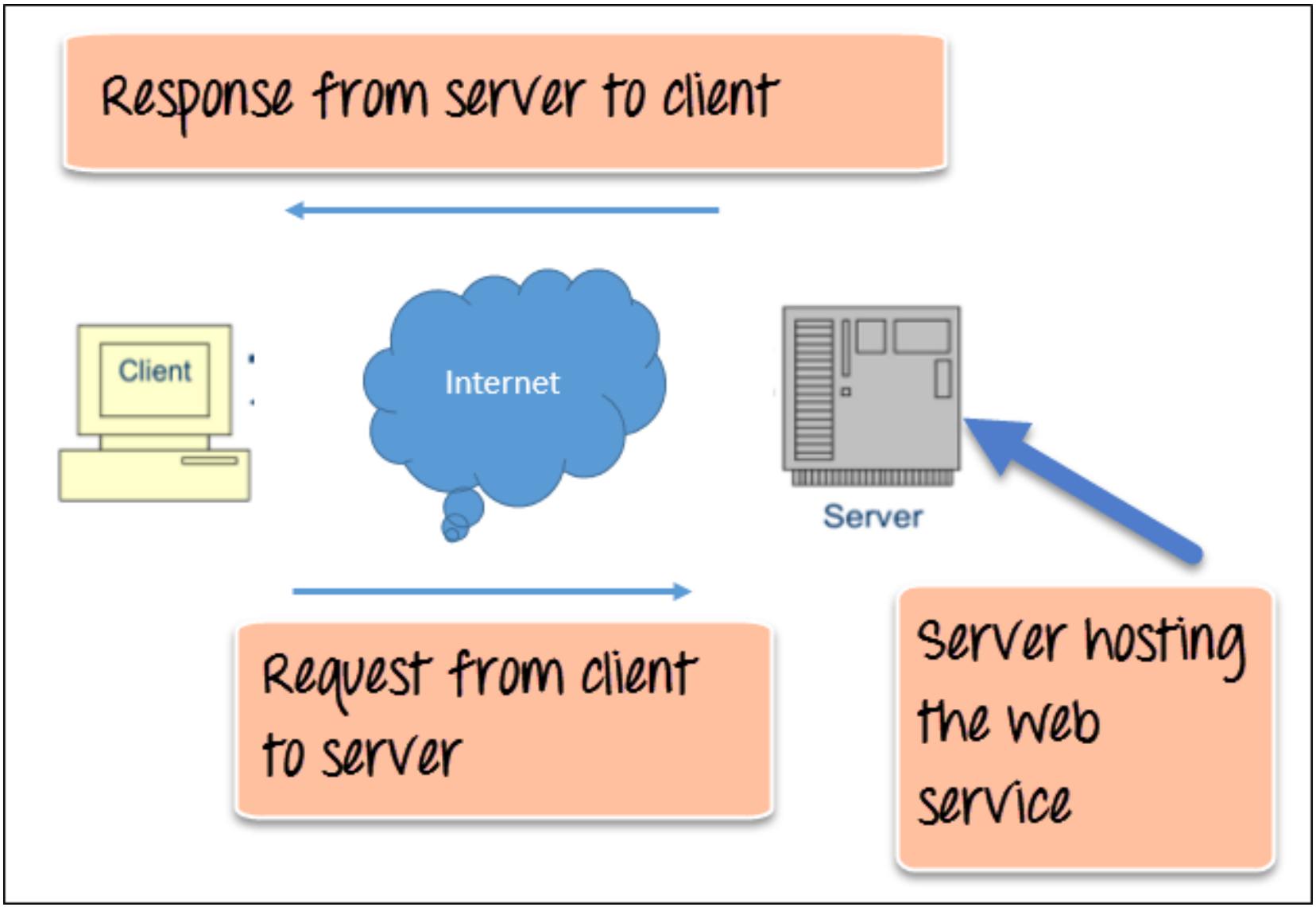
What is a Web Service?

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.

1. The web services can be searched for over the network and can also be invoked accordingly.

2. When invoked, the web service would be able to provide the functionality to the client, which invokes that web service.

How Web Services work?



How Web Services work?

1) The client would invoke a series of web service calls via requests to a server which would host the actual web service. These requests are made through what is known as remote procedure calls(RPC).

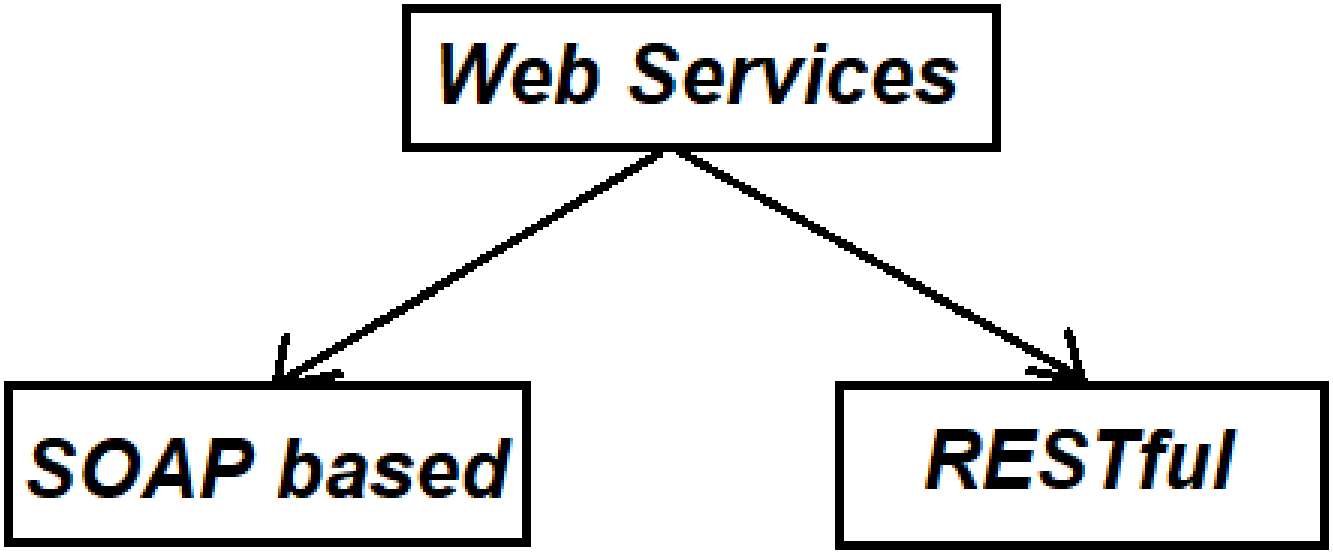
2) The main component of a web service is the data which is transferred between the client and the server, and that is XML. So when applications talk to each other, they actually talk in XML. This provides a common platform for application developed in various programming languages to talk to each other.

How Web Services work?

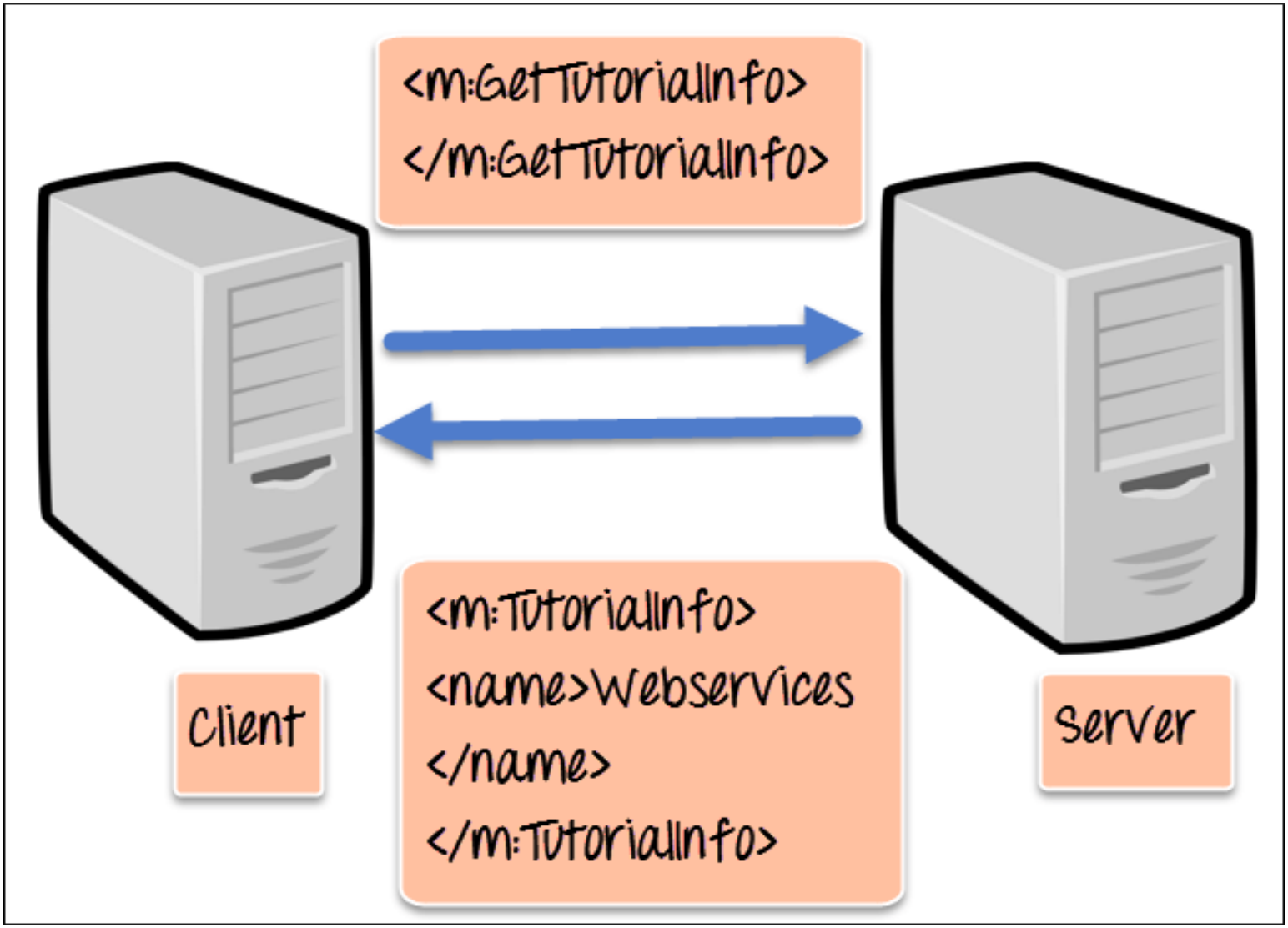
3) Web services use something known as SOAP (Simple Object Access Protocol) for sending the XML data between applications.

4) The data which is sent from the web service to the application is called a SOAP message. The SOAP message is nothing but an XML document. Since the document is written in XML, the client application calling the web service can be written in any programming language.

Types of How Web Services



Client-Server communication using SOAP



What is SOAP?

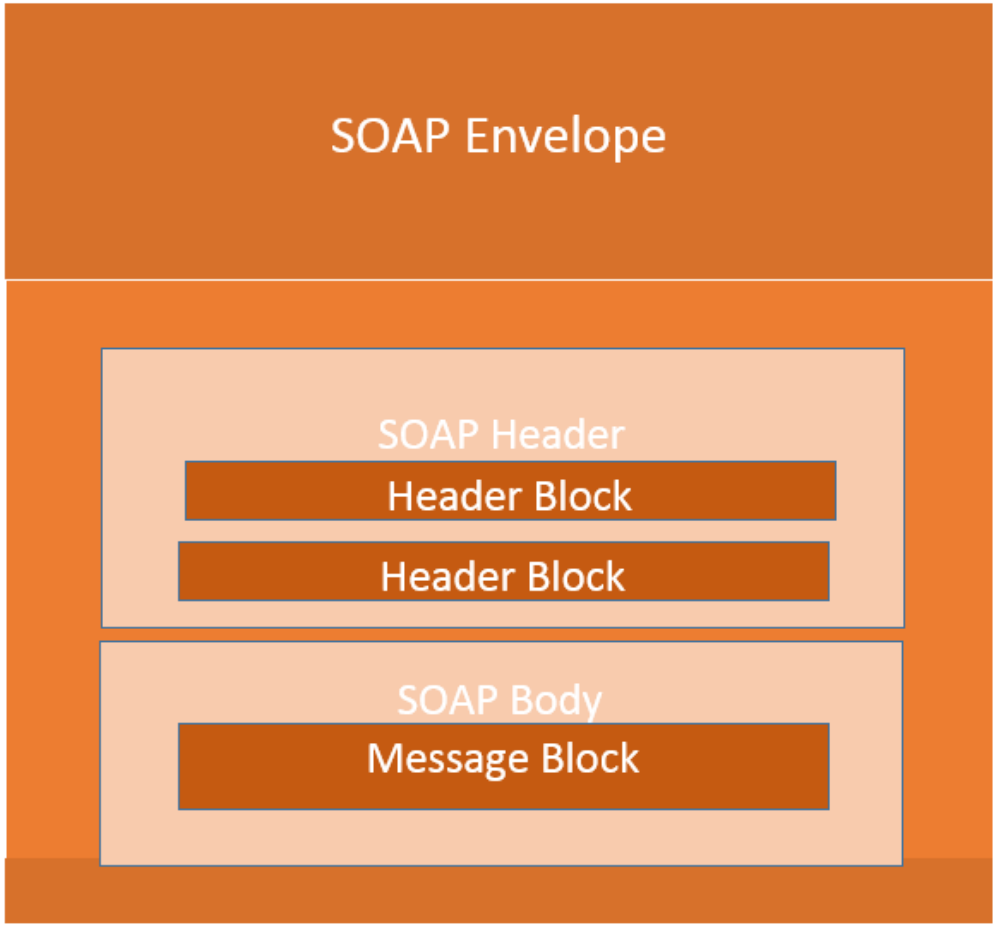
1. SOAP is an XML-based protocol for accessing web services over HTTP.
2. SOAP is known as the Simple Object Access Protocol.
3. In order to establish communication between heterogeneous applications, we use XML. But there are no standard specifications on use of XML across all programming languages for data exchange. That is where SOAP comes in.

Advantages of SOAP

1. SOAP protocol is also recommended by the W3C consortium which is the governing body for all web standards.
2. SOAP is a *light*-weight protocol that is used for data interchange between applications.
3. SOAP is designed to be platform & language independent.
4. SOAP works on the HTTP protocol, which is the default protocol used by all web applications. Hence, there is no sort of customization which is required to run the web services built on the SOAP protocol to work on the World Wide Web.

SOAP message building blocks

The SOAP specification defines something known as a "SOAP message" which is what is sent to the web service and the client application.



SOAP message building blocks

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
.org/sample/">
 <soapenv:Header/>
 <soapenv:Body>
 <sam:loginResponse>
 <sessionid>4074721112005223</sessionid>
 </sam:loginResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

SOAP message building blocks

SOAP Envelope - This is the root element in the SOAP message. It is used to encapsulate all the details in the SOAP message.

SOAP Header element – It can contain information such as authentication credentials & definition of complex types if applicable.

SOAP Body - It contains the actual data which needs to be sent between the web service and the calling application.

SOAP Fault message – If SOAP response returns error then it will add Fault message inside SOAP body.

SOAP Fault Message

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xm
lns:xsi="http://www.w3.org/1999/XMLSchema-instance" xmlns:xsd="http://www.w3.org
/1999/XMLSchema">
      <SOAP-ENV:Body>
         <SOAP-ENV:Fault>
         <faultcode xsi:type="xsd:string">SOAP-ENV:Client</faultcode>
        <faultstring xsi:type="xsd:string">
            Failed to locate method (GetTutorialID) in class (GetTutorial)
         </faultstring>
    </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

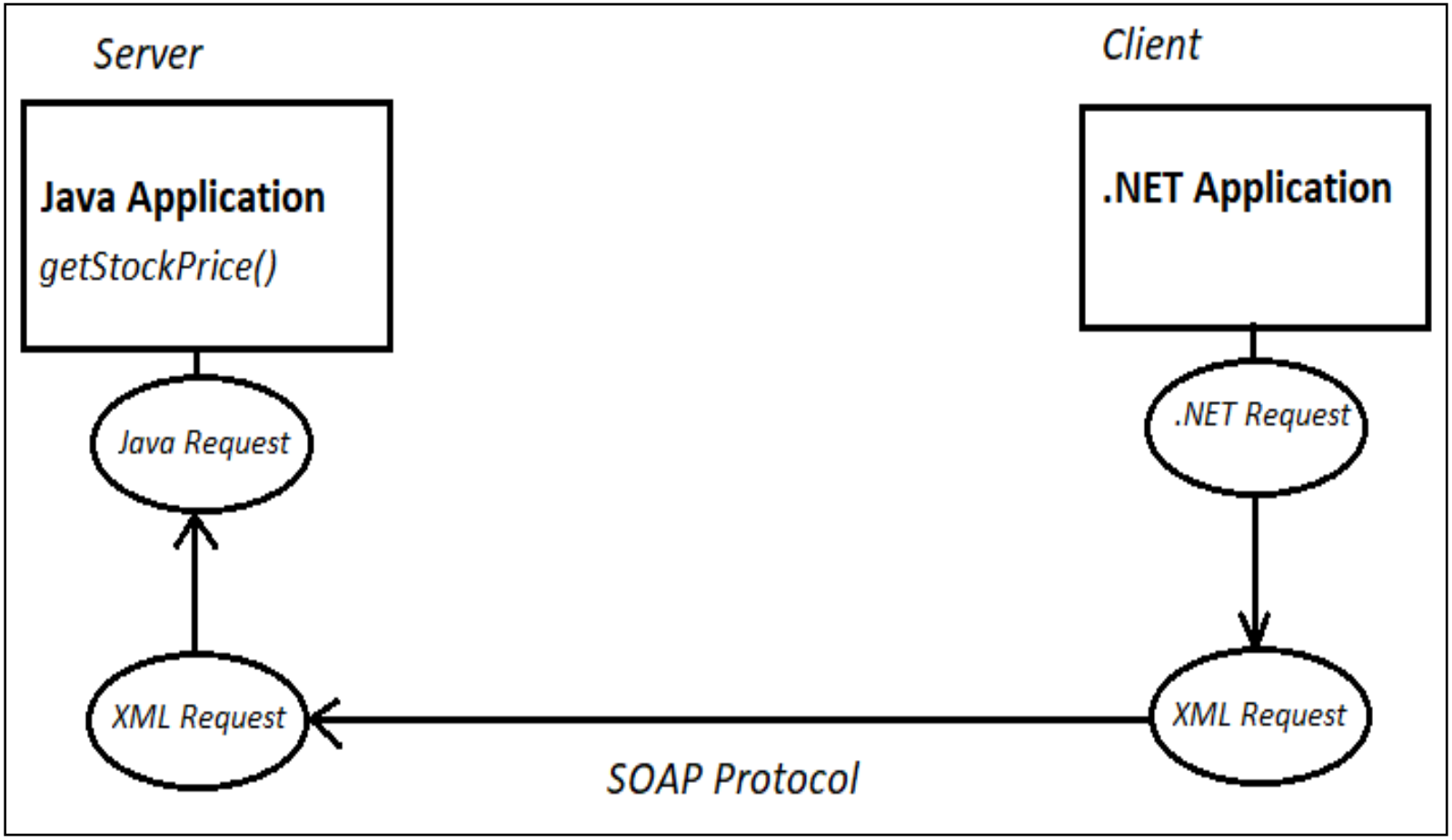
WSDL (Web Service Description Language)

WSDL is an XML format for describing network services as a set of endpoints operating on messages.

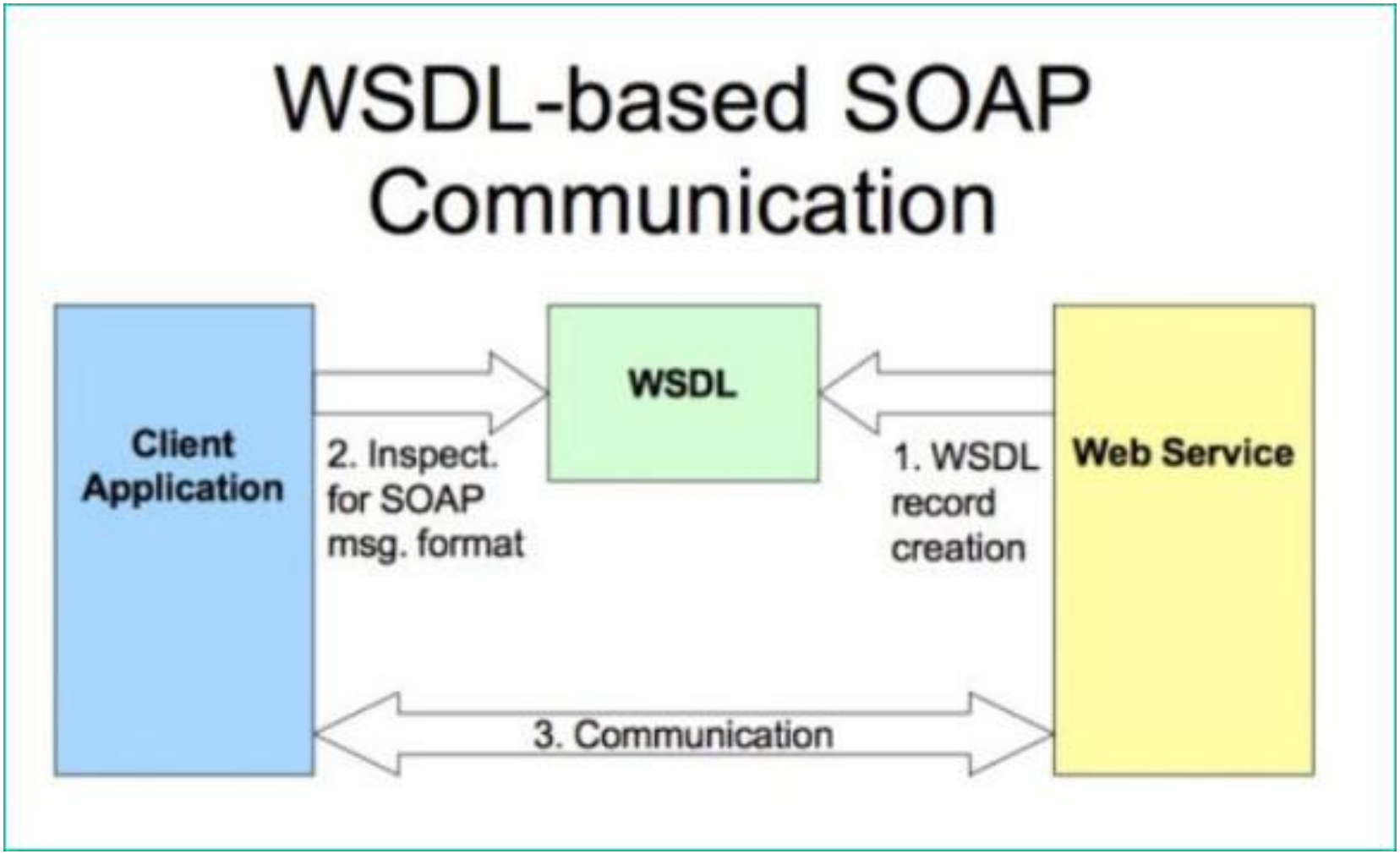
WSDL describes operations & messages abstractly.

```
<wsdl:definitions targetNamespace="http://math.example.com" name="MathFunctionsDef">
<wsdl:message name="addIntResponse">
        <wsdl:part name="addIntReturn" type="xsd:int" />
</wsdl:message>
<wsdl:message name="addIntRequest">
        <wsdl:part name="a" type="xsd:int" />
        <wsdl:part name="b" type="xsd:int" />
</wsdl:message>
<wsdl:portType name="AddFunction">
        <wsdl:operation name="addInt" parameterOrder="a b">
                <wsdl:input message="impl:addIntRequest" name="addIntRequest" />
                <wsdl:output message="impl:addIntResponse" name="addIntResponse" />
        </wsdl:operation>
</wsdl:portType>
<service name="MathFunctions"/>
</wsdl:definitions>
```

SOAP communication



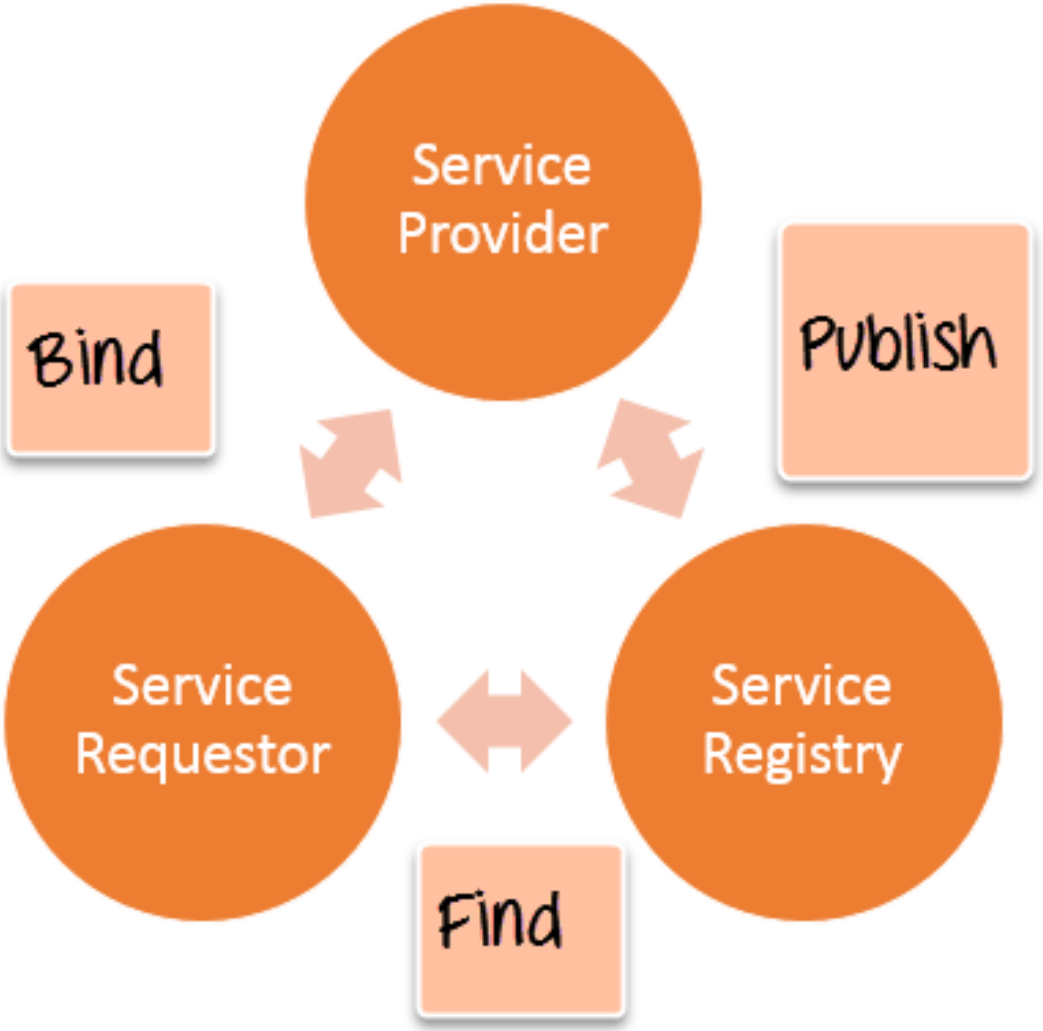
WSDL in SOAP communication



UDDI (Universal Description, Discovery, and Integration)

1. UDDI is a standard for describing, publishing, and discovering the web services that are provided by a particular service provider.
2. UDDI provides a repository on which WSDL files can be hosted.
3. Client application will have complete access to UDDI, which acts as a database containing all the WSDL files.
4. Like a telephone directory has contact information about persons, same way UDDI registry will have the relevant information for the web service.

Web Service Architecture



Web Service Architecture

1. **Service Provider –** It creates the web service and makes it available to client application who want to use it.
2. **Service Requestor –** It is the client application that needs to contact a web service. The client application can be a .Net, Java, or any other language based application which looks for some sort of functionality via a web service.
3. **Service Registry (Broker) –** It is the application which provides access to the UDDI.

RESTful Web Services

1. REST stands for REpresentational State Transfer.
2. Restful Web Service is a lightweight, maintainable, and scalable service that is built on the REST architecture.
3. Restful Web Service, expose API from your application in a secure, uniform, stateless manner to the calling client.
4. The underlying protocol for REST is HTTP.

SOAP vs REST

SOAP	REST
SOAP uses service interfaces to expose its functionality to client applications	REST uses Uniform Service locators to access to the components
SOAP needs more bandwidth	REST needs less bandwidth
SOAP only works with XML formats	REST work with plain text, XML, HTML and JSON

SOAP vs REST

Sr. No.	Functionality	SOAP based web services	REST Service
1	Create a new stock	createStock()	/stock [POST]
2	Update an existing stock	updateStock(stock_id)	/stock/{id} [PUT]
3	Delete all stocks	deleteAllStocks()	/stock [DELETE]
4	Delete a specific stock	deleteStockById(stock_id)	/stock/{id} [DELETE]
5	Read all stocks	getAllStocks()	/stock [GET]
6	Read a specific stock	getStockById(stock_id)	/stock/{id} [GET]

Thank you!!!