Exchanging data between applications is crucial in today's networked world. But data exchange between these heterogeneous applications would be complex. So will be the complexity of the code to accomplish this data exchange.

One of the methods used to combat this complexity is to use XML (Extensible Markup Language) as the intermediate language for exchanging data between applications.

Every programming language can understand the XML markup language. Hence, XML was used as the underlying medium for data exchange.

But there are no standard specifications on use of XML across all programming languages for data exchange. That is where SOAP software comes in.

SOAP was designed to work with XML over HTTP and have some sort of specification which could be used across all applications. We will look into further details on the SOAP protocol in the subsequent chapters.

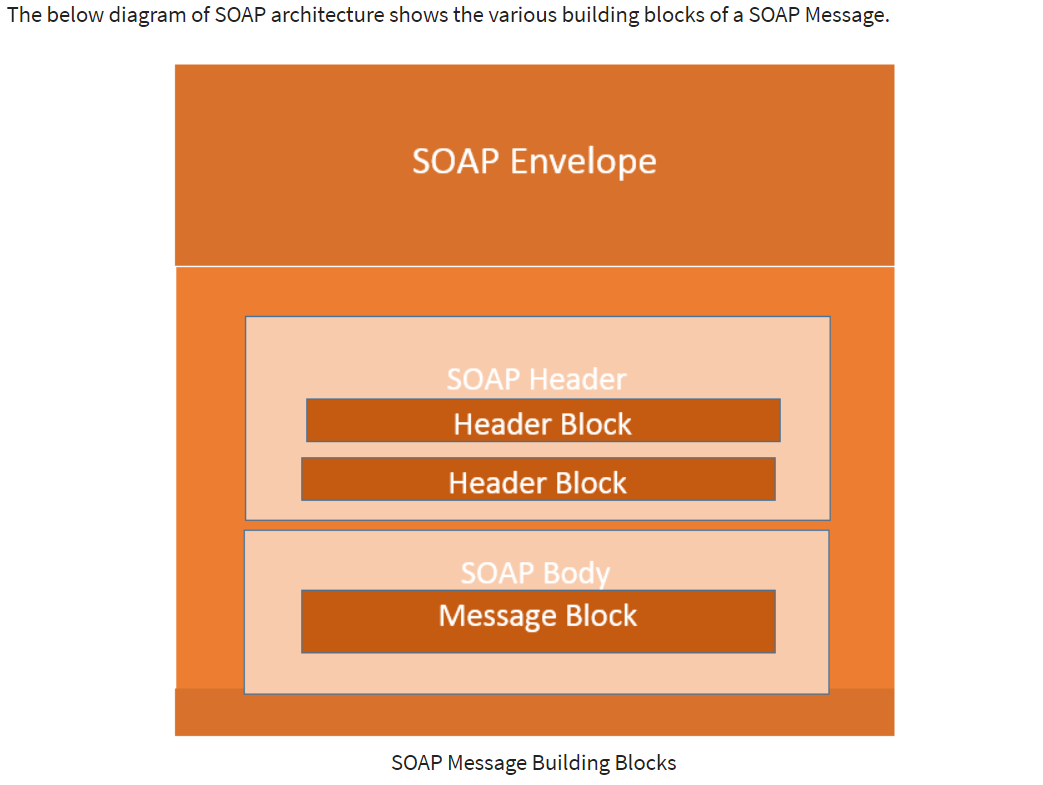
**Advantages of SOAP**

SOAP is the protocol used for data interchange between applications. Below are some of the reasons as to why SOAP is used.

* When developing SOAP based Web services, you need to have some of language which can be used for web services to talk with client applications. SOAP is the perfect medium which was developed in order to achieve this purpose. This protocol is also recommended by the W3C consortium which is the governing body for all web standards.
* SOAP is a light-weight protocol that is used for data interchange between applications. Note the keyword '**light**.' Since SOAP programming is based on the XML language, which itself is a light weight data interchange language, hence SOAP as a protocol that also falls in the same category.
* SOAP is designed to be platform independent and is also designed to be operating system independent. So the SOAP protocol can work any programming language based applications on both Windows and[Linux](https://www.guru99.com/unix-linux-tutorial.html)platform.
* It works on the HTTP protocol –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 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.



The SOAP message is nothing but a mere XML document which has the below components.

* An Envelope element that identifies the XML document as a SOAP message – This is the containing part of the SOAP message and is used to encapsulate all the details in the SOAP message. This is the root element in the SOAP message.
* A Header element that contains header information – The header element can contain information such as authentication credentials which can be used by the calling application. It can also contain the definition of complex types which could be used in the SOAP message. By default, the SOAP message can contain parameters which could be of simple types such as strings and numbers, but can also be a complex object type.

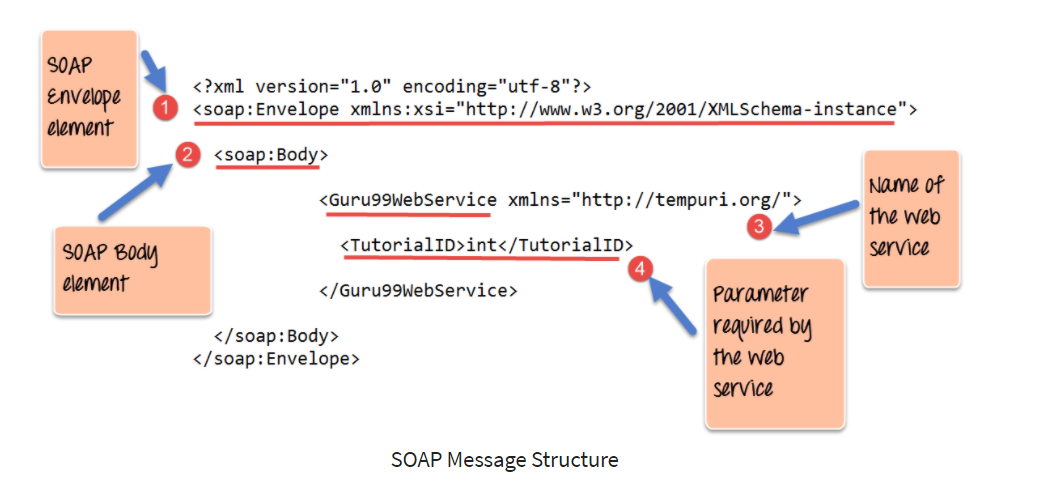
**SOAP Message Structure**

One thing to note is that SOAP messages are normally auto-generated by the web service when it is called.

Whenever a client application calls a method in the web service, the web service will automatically generate a SOAP message which will have the necessary details of the data which will be sent from the web service to the client application.

As discussed in the previous topic of this SOAP tutorial, a simple SOAP Message has the following elements –

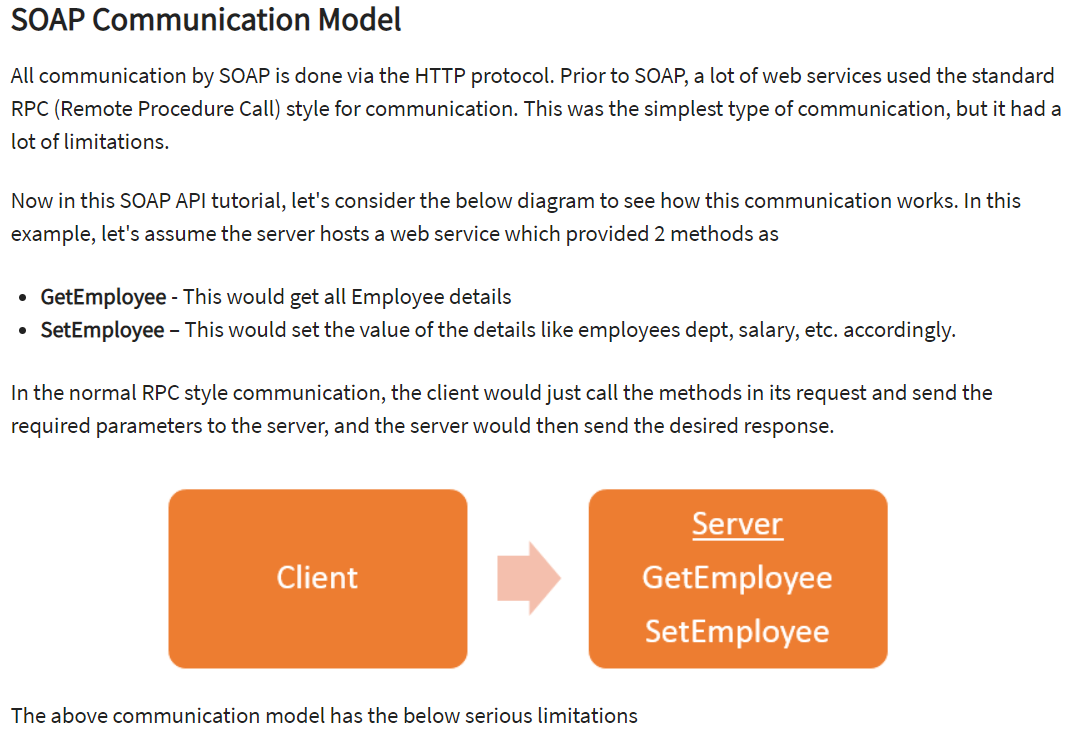
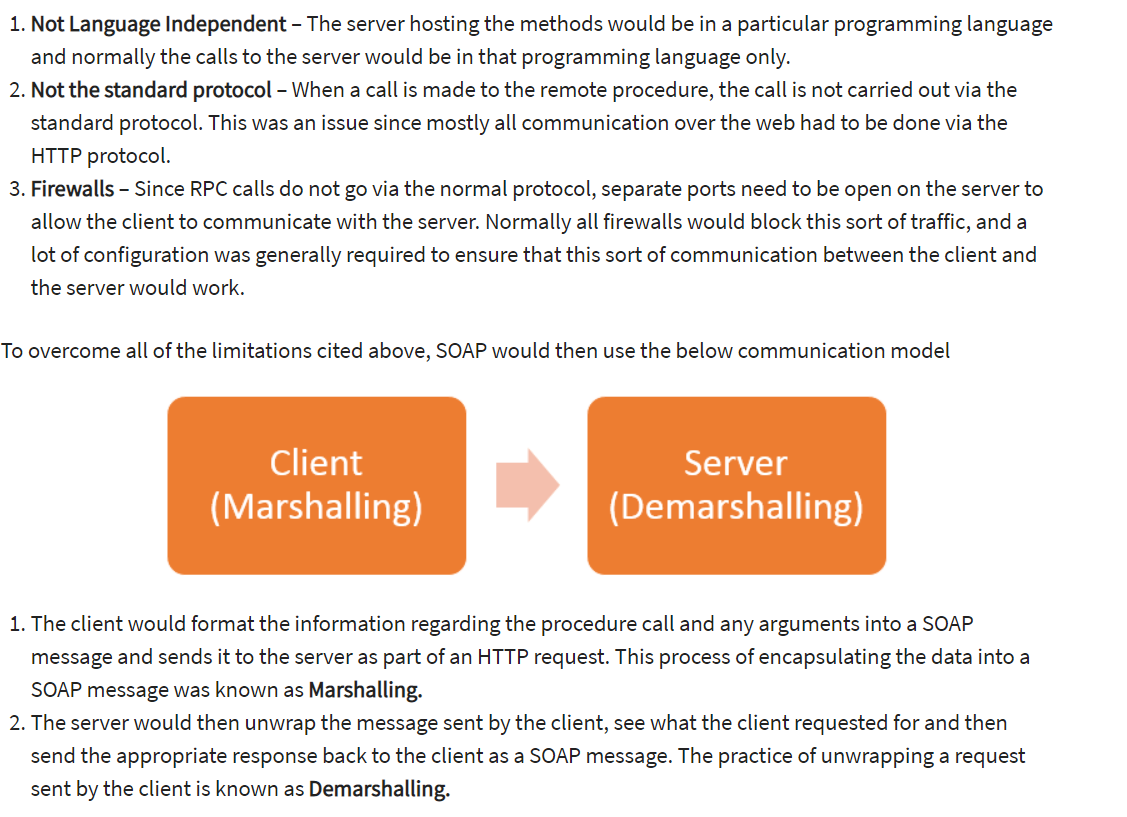
* The Envelope element
* The header element and
* The body element
* The Fault element (Optional)



1. As seen from the above SOAP message, the first part of the SOAP message is the envelope element which is used to encapsulate the entire SOAP message.
2. The next element is the SOAP body which contains the details of the actual message.
3. Our message contains a web service which has the name of "Guru99WebService".
4. The "Guru99Webservice" accepts a parameter of the type 'int' and has the name of TutorialID.

Now, the above SOAP message will be passed between the web service and the client application.

You can see how useful the above information is to the client application. The SOAP message tells the client application what is the name of the Web service, and also what parameters it expects and also what is the type of each parameter which is taken by the web service.

**Limitations of SOAP:**

SOAP relies exclusively on XML to provide messaging services. Microsoft originally developed SOAP to take the place of older technologies that don’t work well on the internet such as the Distributed Component Object Model (DCOM) and Common Object Request Broker Architecture (CORBA). These technologies fail because they rely on binary messaging. The XML messaging that SOAP employs works better over the internet.

After an initial release, Microsoft submitted SOAP to the Internet Engineering Task Force (IETF) where it was standardized. SOAP is designed to support expansion, so it has all sorts of other acronyms and abbreviations associated with it, such as WS-Addressing, WS-Policy, WS-Security, WS-Federation, WS-ReliableMessaging, WS-Coordination, WS-AtomicTransaction, and WS-RemotePortlets. In fact, you can find a whole laundry list of these standards on Web Services Standards.

## ****Solution to SOAP: Overview of REST****

REST provides a lighter-weight alternative. Many developers found SOAP cumbersome and hard to use. For example, working with SOAP in JavaScript means writing a ton of code to perform simple tasks because you must create the required XML structure every time.

Instead of using XML to make a request, REST (usually) relies on a simple URL. In some situations you must provide additional information, but most web services using REST rely exclusively on using the URL approach. REST can use four different HTTP 1.1 verbs (GET, POST, PUT, and DELETE) to perform tasks.

Unlike SOAP, REST doesn’t have to use XML to provide the response. You can find REST-based web services that output the data in Command Separated Value (CSV), JavaScript Object Notation (JSON) and Really Simple Syndication (RSS). The point is you can obtain the output you need, in a form that’s easy to parse within the language you’re using for your application.