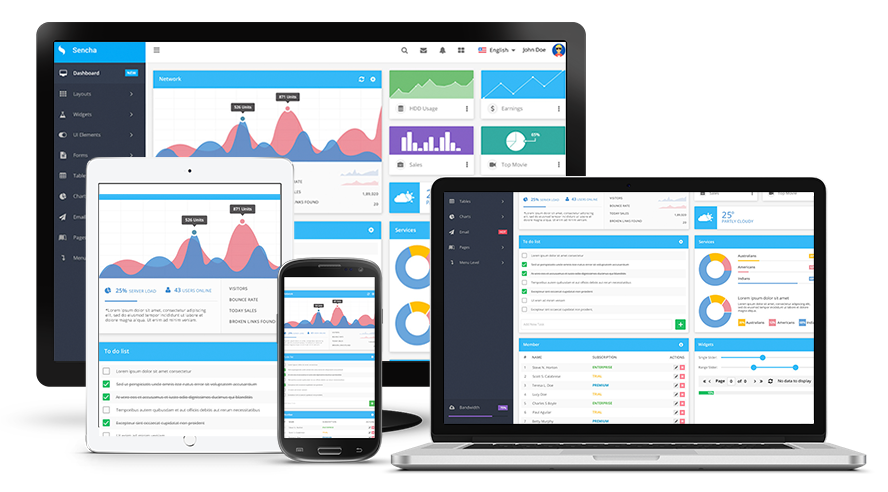
**WEB SERVICES AND SOAP**

**Web applications with web services**

**Web Applications**

****

**Web applications are defined by being interactive. You're supposed to use a web application in order to perform a function and use some of the web applications features. Lots of web applications don't even have real informative content or data exactly. People are just supposed to use them in order to perform additional tasks, using their features to accomplish something. You use a web application to check your incoming messages, for instance, or play a game.**

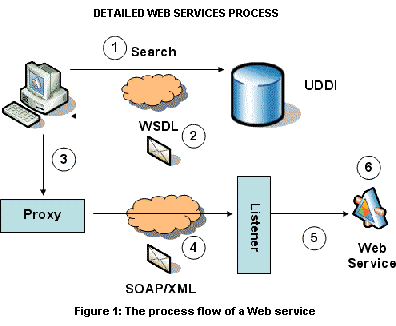
**The browser capabilities involved with web applications are significantly more high-tech, which is one reason why it's usually harder for people to design a web application than a website. Websites are all about getting more data, and web applications are all about doing things. One of your actions is probably going to be getting more information or learning more information, but the web application helped you perform that action. You got the information from a website.**

**The user interface of a web application is also usually much more complicated than the user interface of a website. Websites might have tags and categories that you need to understand, but you don't have to go through and learn any potentially complicated tasks in order to use websites. Web applications often require step-by-step guides, or you're not going to be able to complete them.**

**The setup of websites are completely different from web applications in most cases. The rhythm of typing in the address, loading websites, and going back and forth between websites is often absent with web applications.**

**Web applications, unsurprisingly, are usually harder to design and create than websites. Lots of people have their own websites today, and this was the case even ten years ago. The** **people who are able to create their own web applications can more easily make money off of them because it takes more work to learn how to code and create a web application.**

**Web Services**

****

**Web service is a technology by which two or more remote web applications interact with each other over network/internet. It can be implemented using Java, .net, PHP etc. Web pages allow people to communicate and collaborate with each other while web services allow programs to communicate and collaborate with each other.**

**A web service is essentially 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. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards (XML, SOAP, HTTP).**

**All the standard Web Services works using following components:**

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

**Need for web services**

**Re-usability**

**Once we develop some business logic, we can make it reuse for other applications**

**Example:**

**If 10 different applications requires to use our logic**

**We can expose our logic over a network as a web service**

**So all the 10 different application can access it from the network**.

**Interoperability**

**It provides the freedom for a developers to choose whatever the technology they want to use for development.**

**Web services uses a set of standards and protocols and enable us to achieve interoperability.**

**Hence applications developed in Java,Mainframe,Ruby or any other technology can call the web service and use it.**

**Loosely coupled**

**Web service exist independent of the other parts of the application that uses it.**

**So any changes to the application can be made without affecting the web service.**

**Deployability**

**It is very easy to deploy the web application as they are deployed over standard internet technologies.**

**WEB SERVICE DESCRIPTION LANGUAGE (WSDL)**

****

**•WSDL is an XML vocabulary for describing Web services allowing developers to describe Web Services and their capabilities, in a standard manner.**

**•standard format to describe a Web Service (description stack)**

**•specifies three fundamental properties:**

**•What a service does -operations (methods) provided by the service**

**•How a service is accessed -data format and protocol details**

**•Where a service is located -Address (URL) details**

**•The document written in WSDL is also simple called a WSDL (or WSDL document) •WSDL is a contract between the XML (SOAP) Web service and the client who wishes to use this service**

**•The service provides the WSDL document and the Web service client uses the WSDL document to create the stub (or to dynamically decode messages).**

**Fundamental properties of a WSDL**

**•specifies three fundamental properties:**

**•What a service does -operations (methods) provided by the service •How a service is accessed -data format and protocol details**

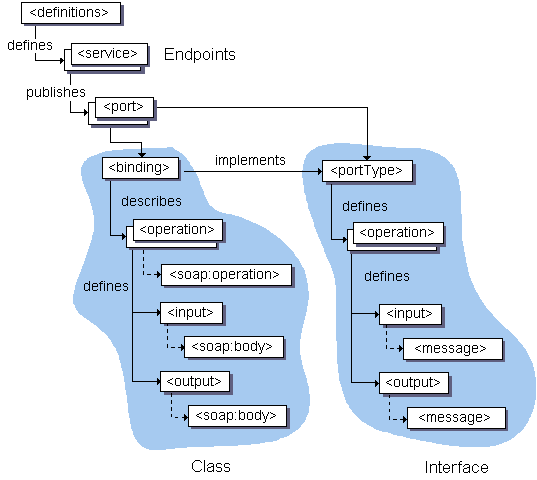
**•Where a service is located -Address (URL) details**

**Use of WSDL document in web services and client development**

**•WSDL is a contract between the XML(SOAP) Web service and the client who wishes to use this service**

**•The service provides the WSDL document and the Web service client uses the WSDL document to create the stub (or to dynamically decode messages)**

**Structure of the WSDL document**

****

**Web Services Description Language (WSDL) is an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages. The diagram below illustrates the elements that are present in a WSDL document, and indicates their relationships. To see an example of how this is implemented in a WSDL document,**

**Elements in WSDL**

**TYPE**

**•use the XML schema language to declare complex data types and elements that are used elsewhere in the WSDL document**

**•serve as a container for defining any data types that are not described by the XML schema built-in types: complex types and custom simple types,**

**•the data types and elements defined in the type’s element are used by message definitions when declaring the parts (payloads) of messages**

**MESSAGE**

**•any WSDL interface is the set of messages that the service behind the interface expects to send and receive**

**•messages are abstractions of request and response messages that clients and web services exchange partitioned into a number of logical parts to ease access to its contents**

**•messages are constructed from a number of XML Schema typed part elements, which can come from the types part of the description or an external schema that has been imported into the same WSDL document**

**•they contain types of other schemas and user defined types into input, output and fault messages that the web service will consume and produce**

**PORT TYPE**

**•define web service functionality at abstract level grouping sets of message exchanges into operations**

**•contain a set of operations that incorporates input, output and fault messages and parameter order**

**•WSDL supports at least a single input and output message, but permits the declaration of an arbitrary number of faults**

**•portTypeelement may have one or more operation elements, each of which defines an RPC-or document style Web service method**

**•Java equivalence:**

**•port Type-> java interface**

•**operation -> method name**

**BINDING**

•**map a PortType to a specific protocol, typically SOAP over http, using a specific data encoding style**

**•one portType can be bound to several different protocols by using more than one port**

**•bindings refer back to portTypes by name, just as operations point to messages.**

**•binding styles may be either “RPC” or “Document”(SOAP). •also specify SOAP encoding**

**SERVICE**

**•it binds the Web service to a specific network-addressable location, i.e. it takes the bindings declared previously and ties them to a port, which is a physical network endpoint to which clients bind over the specified protocol,**

**•contain one or more port elements, each of which represents a different Web service and the port element assigns the URL to a specific binding,**

**•reference a particular binding, and along with addressing information is wrapped together into a service element to form the final physical, network addressable Web service,**

**•access point for each port**

**•port type -The operations performed by the web service.**

**•message -The messages used by the web service.**

**•types -The data types used by the web service.**

**•binding -The communication protocols used by the web service.**

**WSDL Elements:PortType**

**Port type is a collection of one or more related operations describing the interface of a web service.**

**PortType definition is a collection of operation elements.**

**Generally,WSDL documents contain only one portType elements, because different web service interface defenitions are written with different documents.**

**portType has a single name attribute.**

**The name of portType together with the namespace of the WDSL document define a unique name for the portType.**

**SOAP is used with HTTP**

****

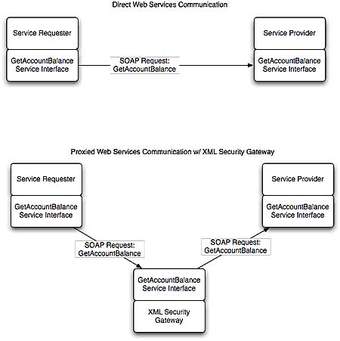
**SOAP is a messaging protocol and in a nutshell is just another XML language.  
Its purpose is the data exchange over networks. Its concern is the encapsulation of these data and the rules for transmitting and receiving them.**

**HTTP is an application protocol and SOAP messages are placed as the HTTP payload.  
Although there is the overhead of HTTP, it has the advantage that it is a protocol that is open to firewalls, well-understood and widely-supported. Thus, web services can be accessed and exposed via technology already in-place.**

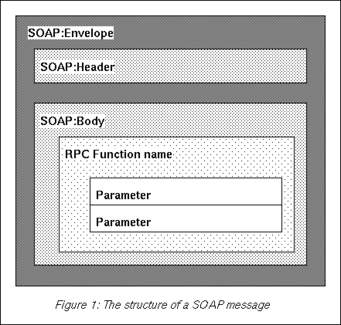
**SOAP messages are usually exchanged via HTTP. Although it is possible to use other (application) protocols, e.g. SMTP or FTP, the non-HTTP bindings are not specified by SOAP specs and are not supported by**[**WS-BP (interoperability spec)**](http://en.wikipedia.org/wiki/WS-I_Basic_Profile)**.  
You could exchange SOAP messages over raw TCP but then you would have web services that are not interoperable (not compliant to WS-BP).**

**Nowadays the debate is why the SOAP have overhead at all and not send data over HTTP (RESTful WS).**

**SOAP can be used for functional oriented communication**

****

**Structure of SOAP message in message oriented communication and elements used**

****

**•SOAP messages**

**•consistent envelope -header and body**

**•consistent data encoding -based on XML Schema type system •protocol binding framework**

**•has encoding rules**

**•defines a serialization mechanism that can be used to exchange instances of application-defined objects**

**•Envelope**

**•wraps entire message and contains header and body**

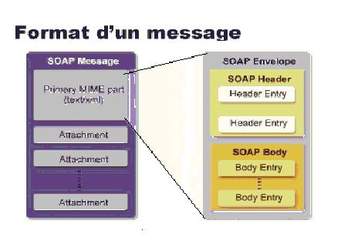
**•defines an overall framework for expressing what is in a message; who should deal with it, and whether it is optional or mandatory**

**•Header**

**•optional element with additional info such as security or routing •Body**

**•application-specific message content being communicated as arbitrary XML payloads in the request and response messages**

**•fault element provides information about errors that occurred while processing the message**



**Importance of the SOAP attachments**

**SOAP (abbreviation for Simple Object Access Protocol) is a messaging**[**protocol**](https://en.wikipedia.org/wiki/Protocol_(computing))**specification for exchanging structured information in the implementation of**[**web services**](https://en.wikipedia.org/wiki/Web_service)**in**[**computer networks**](https://en.wikipedia.org/wiki/Computer_network)**. Its purpose is to provide**[**extensibility**](https://en.wikipedia.org/wiki/Extensibility)**,**[**neutrality**](https://en.wikipedia.org/wiki/Neutrality_(philosophy))**and independence. It uses**[**XML Information Set**](https://en.wikipedia.org/wiki/XML_Information_Set)**for its**[**message format**](https://en.wikipedia.org/wiki/Message_format)**, and relies on**[**application layer**](https://en.wikipedia.org/wiki/Application_layer)**protocols, most often**[**Hypertext Transfer Protocol**](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)**(HTTP) or**[**Simple Mail Transfer Protocol**](https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol)**(SMTP), for message negotiation and transmission.**

**SOAP allows processes running on disparate operating systems (such as**[**Windows**](https://en.wikipedia.org/wiki/Microsoft_Windows)**and**[**Linux**](https://en.wikipedia.org/wiki/Linux)**) to communicate using**[**Extensible Markup Language**](https://en.wikipedia.org/wiki/XML)**(XML). Since Web protocols like HTTP are installed and running on all operating systems, SOAP allows clients to invoke web services and receive responses independent of language and platforms.**

### Advantages

* **SOAP's neutrality characteristic explicitly makes it suitable for use with any transport protocol. Implementations often use HTTP as a transport protocol, but other popular transport protocols can be used. For example, SOAP can also be used over SMTP,**[**JMS**](https://en.wikipedia.org/wiki/Java_Message_Service)**and**[**message queues**](https://en.wikipedia.org/wiki/Message_queue)**.**
* **SOAP, when combined with HTTP post/response exchanges, tunnels easily through existing firewalls and proxies, and consequently doesn't require modifying the widespread computing and communication infrastructures that exist for processing HTTP post/response exchanges**
* **SOAP has available to it all the facilities of XML, including easy internationalization and extensibility with XML Namespaces.**

**MIME header**

**•SOAP messages may have one or more attachments**

**•each Attachment Part object has a MIME header to indicate the type of data it contains.**

**•it may also have additional MIME headers to identify it or to give its location, which can be useful when there are multiple attachments •when a SOAP message has one or more Attachment Part objects, its SOAP Part object may or may not contain message content**

* **Text in**[**character sets**](https://en.wikipedia.org/wiki/Character_set)**other than**[**ASCII**](https://en.wikipedia.org/wiki/ASCII)
* **Non-text attachments: audio, video, images, application programs etc.**
* **Message bodies with multiple parts**
* **Header information in non-ASCII character sets**

**Frameworks/libraries for SOAP web service development**

•**JAX-WS (Java API for XML Web Services) is a java package/library for developing web services**

**•It supports both Functional oriented and message oriented communication via SOAP**

**•JAX-WS API hides the complexity of SOAP from the application developer**

**•JAX-WS uses the javax.jwspackage**

**•It uses annotations**

**•@WebService**

**•@WebMethod**

**•@OneWay**

**•@WebParam**

**•@WebResult**



**Annotations in JAX-WS**

****

**Java API for XML-Based Web Services (JAX-WS) relies on the use of annotations to specify metadata associated with web services implementations and to simplify the development of web services. Annotations describe how a server-side service implementation is accessed as a web service or how a client-side Java class accesses web services.**

**The JAX-WS programming standard introduces support for annotating Java classes with metadata that is used to define a service endpoint application as a web service and how a client can access the web service. JAX-WS supports the use of annotations based on the Metadata Facility for the Java Programming Language (Java Specification Request (JSR) 175) specification, the Web Services Metadata for the Java Platform (JSR 181) specification and annotations defined by the JAX-WS 2.0 and later (JSR 224) specification which includes JAXB annotations. Using annotations from the JSR 181 standard, you can simply annotate the service implementation class or the service interface and now the application is enabled as a web service. Using annotations within the Java source simplifies development and deployment of web services by defining some of the additional information that is typically obtained from deployment descriptor files, WSDL files, or mapping metadata from XML and WSDL into the source artifacts.**

**Use annotations to configure bindings, handler chains, set names of portType, service and other WSDL parameters. Annotations are used in mapping Java to WSDL and schema, and at runtime to control how the JAX-WS runtime processes and responds to web service invocations.**

**Web service can be tested using different approaches**

**Testing of web services is one of the important type of software testing approach, which is mainly used to determine the expectations for reliability, functionality, performance, etc.**

**As these days automated testing is considered as one of the most trending methodology in the field of software testing, hence testing web apps based on RESTful APIs through automation will provide effective test results. Some of the best & popular tools for web services testing are:**

1. **SoapUI,**
2. **TestingWhiz**,
3. **SOATest**
4. **TestMaker,**
5. **Postman, etc.**
6. **As mentioned in my previous posts related to web services or API testing, among above mentioned tools, I consider Testingwhiz and SoupUI as the most user-friendly tools for API testing. Automating API testing, as a part of**[**automated web services testing**](http://www.testing-whiz.com/web-services-testing?utm_campaign=Pratik&utm_medium=QnA&utm_source=Quora)**solution that helps you to test whether your application communicates and accesses functions correctly from the Web by effectively verifying the behavior of web services connected to them.**
7. **For better clarity and more information, you can go through the following informative article**: