International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

# A comparative study on SOAP and RESTful web services

**Sirsha Chatterjee[1], Mamatha T[2]**

[1]*Under Graduate Student, Dept. of Computer Science and Engineering, RV College of Engineering, Bengaluru, Karnataka*

[2]*Assistant Professor, Dept. of Computer Science and Engineering, RV College of Engineering, Bengaluru, Karnataka*

---

**Abstract -** *Modern enterprise applications nowadays need to build web-based applications using a wide variety of programming platforms, for different requirements. While backend service applications are developed using Java, .Net, or Node JS, front-end application are developed using ReactJS, AngularJS, etc. These heterogeneity in use of types of application platforms require a common communication service to transfer information and data. Web services provide this service which enables multiple applications to communicate with each other. Web services perform functions ranging from simple requests to complicate business processes. SOAP (Simple Object Access Protocol) and REST (Representational State Transfer) are the most common and popular types of web service protocols in use. Both have their own advantages and disadvantages.*

***Key Words*: SOAP, REST, XML, HTML, Server, Client.**

## 1. INTRODUCTION

In today's era of modern technology, web applications can be developed through several ways. The technical advantages are the deciding factor to prefer one web service protocol over the other alternatives. Web services is thus the most common and popular alternative for communication of information among the devices. [1] describes web services as self-describing and modular business applications. A web service is basically a software system which makes itself available over the internet. According to [1] and [3], programmable interfaces play a major role in the development of web services. IP address play a major part in web services, as in, it is responsible for invoking and consuming of web services. Since web services are mostly service oriented, SOAP and REST uses an API consuming services. This API is known as Web API. According to [4], web services are platform independent and loosely coupled services, that is, web services are not dependent on any prior knowledge of information and can be used by a variety of programming languages such as Java, React, Python, etc. SOAP is a web service protocol which is lightweight, platform independent. SOAP is built for decentralized and distributed environment.

SOAP is an XML based messaging protocol. According to [3], SOAP is a one-directional stateless communication between two communication nodes, the sender and the receiver.

Unlike SOAP web service, RESTful web service uses REST architecture, which is built mainly for point-to-point communication. REST is a client-server architecture. The request is sent by the client to the server, where the message is processed and appropriate response os sent back to the client. REST was developed in 2000 by Roy Fielding. [3] states that REST services is not only limited to XML but can also support JSON (JavaScript Object Notation), as well as plain text, unlike SOAP which only supports documents in XML format.

In this paper, the two services are analyzed and compared based on their underlying architecture, differences, advantages and disadvantages.

## 2. SOAP Web Service

Simple Object Access Protocol (SOAP) is an XML based messaging protocol that uses HTTP protocols to allow applications to communicate. SOAP is basically a one-directional, stateless communication between two communication nodes, the sender and the receiver, but, as concluded in [5] and [6], by making use of the addition features of the transport and/or application specific protocols and features, we could create interactions of higher complexity such as request/response interactions, single request/ multiple response interactions, etc. It is a structured XML messaging format used for data exchange. Currently there are two versions of SOAP:

- SOAP version 1.1
- SOAP version 1.2

Some of the advantages of SOAP 1.2 over the older SOAP 1.1 are [7]:

- SOAP Version 1.2 uses robust processing model.
- SOAP 1.2 provides better interoperability due to its better testing and implementation.
- SOAP Version 1.2 uses XML Information Set.
- SOAP Version 1.2 provides an HTTP based binding framework, thus giving the developers protocol independence.
- The binding leads to better integration to the World Wide Web
- SOAP Version 1.2 provides a well-defined extensibility model
- SOAP Version 1.2 provides better support for Web standards

SOAP nodes are of 3 types:

- SOAP Sender - responsible for producing and sending a SOAP message
- SOAP Receiver – responsible for obtaining and processing the SOAP message
- SOAP Intermediary – responsible for receiving and processing the SOAP header blocks.

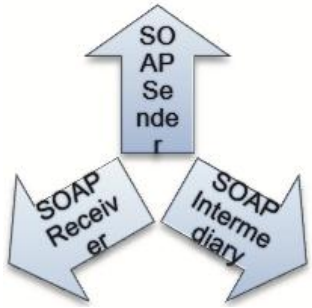


**Fig - 1:** SOAP nodes

The Structure of the SOAP message consists of the following elements:

- SOAP Envelope – this is the most important SOAP element. It defines the start and end of the SOAP message
- SOAP Header – this is an option element which is used specify any specific attributes to be used while processing the message.
- SOAP body – This is the element which contains the actual message.
- SOAP Fault – this is an optional element which is used to give information about errors that might occur while processing the message.

SOAP is an independent and lightweight protocol because it does not depend specifically to any particular OS or platform. XML and HTTP protocols are the key reasons of making this possible.

SOAP messaging requests consists of two types: Remote Procedure Call and Document request. They are discussed in the following sections.

## 2.1 Remote Procedure Call

Remote Procedure call is a type of call which is uses client-server model. In a RPC, a program can request a service from another program without any prior knowledge of any of the network's information, such as its address.

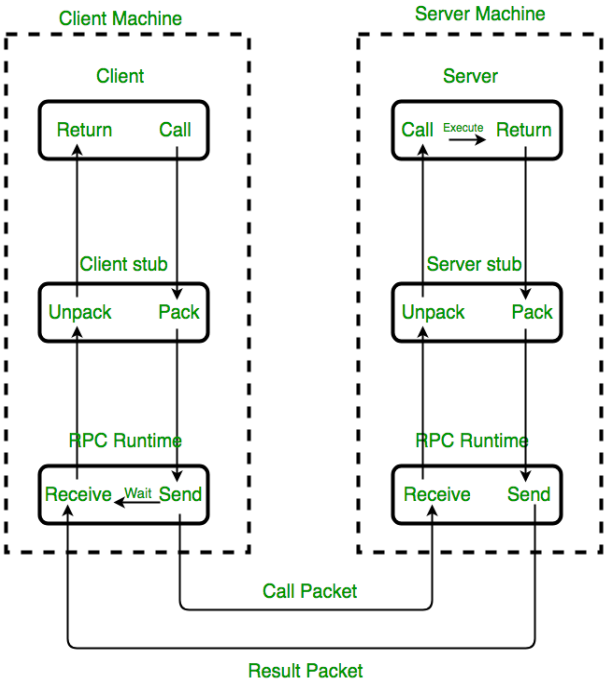


**Fig - 2:** RPC Architecture

## 2.2 Document Requests

In this type of SOAP messaging request, the XML file is forwarded in the SOAP body instead of as a parameter, like in RPC.

A service called 'GetCountries,' for example, includes a document based on XML as its input message. The request is transmitted via SOAP message, specifying which operation of the service it requires the reponse for. Let's say it requests the 'GetCountryCode' operation. The document should contain a request payload having an XML document with country name as input in SOAP message. The request is sent to the server and is processed. Finally, when the processing is done, anther XML document is produced containing the response payload for the data requested.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap
/envelope/" xmlns:web="http://www.oorsprong.org/
websamples.countryinfo">
   <soapenv:Header/>
   <soapenv:Body>
      <web:CountryISOCode>
         <web:sCountryName>India</web:sCountryName>
      </web:CountryISOCode>
   </soapenv:Body>
</soapenv:Envelope>
```

**Fig - 4**: Document request example

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/">
   <soap:Body>
      <m:CountryISOCodeResponse
      xmlns:m="http://www.oorsprong.org/websamples.countryinfo">
         <m:CountryISOCodeResult>IN</m:CountryISOCodeResult>
      </m:CountryISOCodeResponse>
   </soap:Body>
</soap:Envelope>
```

**Fig - 5**: Document response example

Fig - 4 is a sample document request payload providing 'India' as the input parameter field.

Fig – 5 is the response payload for the given request. It contains the Country's ISO code, i.e. 'IN'.

## 3. RESTful web service

Representational State Transfer (REST) is specifically concerned with the relationship between the client and the server and how the state is processed. REST architecture mainly uses server client relationship. In REST architecture, a REST Server is responsible for providing resource access to the client, while the REST is responsible for reading and modifying or writing the resources. Thus, the requests and responses are developed keeping in mind the process transfer of the resources. Uniform Resource Identifier (URI) is usually used for identifying resources to contain a document that captures the resource state.

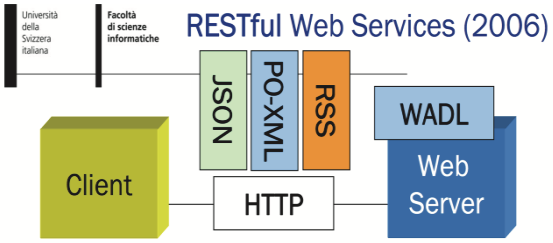


**Fig - 6**: Architecture of RESTful web services

REST is usually much lighter than SOAP architecture. It requires no formats such as headers, as required by the SOAP architecture, to be included in the message. Besides, it parses JSON, a human-readable language that enables data sharing and makes it easier to interpret and use data, instead of XML as used by SOAP architecture. Although REST does support responses in a variety of formats other than JSON, like XML, plain text, etc.

```
{
    "currencies": [
        {
            "code": "INR",
            "name": "Indian rupee",
            "symbol": "₹"
        }
    ],
    "name": "India",
    "capital": "New Delhi"
}
```

**Fig - 7**: A sample JSON representation format

There are several principles for designing RESTful Web Service, some of them are:

- **Addressability** – Each resource should have at least one URI associated with it in REST service. URI is used to determine a resource or a set of resources.
- **Statelessness** - A RESTful service is a stateless service. A request is assumed independent of any past requests by the service. HTTP is by design a stateless protocol, and additional efforts needs to be made to design a stateful service with help of HTTP. A clear idea needs to be there about a stateful and stateless design to avoid confusion.
- **Cacheable** - Data marked as cacheable are stored in the system and can be reused in the future as the response to the same subsequent request instead of producing the same results repeatedly. Cache constraints are thus used to enable response data to be marked as cacheable or non-cacheable.
- **Uniform interface** - allows a common and standardized interface to be used to access the services, i.e. using defined collection of HTTP methods. If the service designer abides by these concepts, then the REST implementation is almost guaranteed to be simple and lightweight. For this purpose, HTTP 1.1 gives a set of methods, called as verbs. Few of important verbs:

| HTTP METHOD | IDEMPOTENCE | SAFETY |
|---|---|---|
| GET | YES | YES |
| HEAD | YES | YES |
| PUT | YES | NO |
| DELETE | YES | NO |
| POST | NO | NO |
| PATCH | NO | NO |

**Fig - 7**: Verbs

### 3.1 Advantages of REST

Some of the advantages of RESTful web service are:

- Uses simpler format for messages
- Relatively cheap
- Offers stronger long-term efficiency
- Easy to learn
- It supports stateless communication
- Uses HTTP standards and grammar
- Messages, being in JSON format, and other formats, leads to less bandwidth.
- HTTP standards for security
- Flexible in consumption, any client can consume.
- Data available as resource.

### 3.2 Disadvantages of REST

Some of the disadvantages of RESTful web service are:

- Fails in standards of Web service such as Security, Transactions etc.
- REST requests (especially GET) are not scalable
- Can't handle large amount of data.
- Unreliable
- Delay in processing requests.
- Latency in bandwidth use.
- Though the REST APIs rely on headers for details based on state, it is clumsy to use headers and link the API to http as a transport.

## 4. COMPARISON

**Table -1:** Feature Based comparison of SOAP and REST web services

| SOAP Web Service | REST Web Service |
|---|---|
| Independent of language, platform, and transportation. | Independent of language, platform, but dependent on transportation. |
| It supports parsing of binary attachment files. | It explicitly supports all of the data forms. |
| Requires heavy input payload compared to REST. | REST is lightweight as it uses URI for data transfer. |
| SOAP supports only POST requests. | Restful APIs supports all types of requests using either GET, POST, PUT, DELETE, etc. |
| More difficult to generate, needs tools. | Much easier web than SOAP in creating web services. |
| Designed to manage the paradigms of distributed computing | Designed to handle point-to-point communication model. |
| Change in SOAP services often leads to a significant change in code on the client side. | Client-side code is not affected by change in services in REST web provisioning. |
| Return type is always of XML type. | Provides versatility with respect to the form of the returned data. |
| SOAP is not good with wireless network. | REST is wireless infrastructure friendly. |
| Due to heavy payload, SOAP web services has more bandwidth requirement. | Due to REST being lightweight, REST consumes less bandwidth. |
| SOAP gets better assistance from other vendors such as WSDL, WS, etc. | REST lacks support for security, policy, reliable messaging and so on in standards. |

## 5. CONCLUSIONS

After going through the analysis and comparison, it is likely to be concluded that RESTful web services would be the one to dominate in the coming years. As was seen, REST web services are lightweight, flexible and fast and has support for most of the datatypes. REST web services definitely provide services which are alluring for the technological space. However, REST is still in its very early phase of its design and innovation. There is still improvement needed forits scalability, increase in accommodation of web standards, security, transactions, etc. In recent times, SOAP based web services have moved more into the legacy class of web services architecture, while a gradual shift is being made towards migrating to the REST web services. However, there are still enterprises which seek to accommodate the best of both the web architectures and use both protocols for achieving different services. But in the long run, RESTful web services are the protocol you need if you want something to stay online for a decade.

## REFERENCES

[1] Wagh, K. "A Comparative study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host", Journal of Information Engineering and Applications, ISSN 2225- 0506 (Online), 2(5), 2012.

[2] Mironela, P., "The Importance of Web Services using the RPC and REST Architecture", IEEE, International Conference on Computer Technology and Development, 2009.

[3] Halili, F., & Dika, A. (2012). "Choreography of Web Services and Estimation of Execution Plan", IEEE International Conference of Information Technology and e-Services (ICITeS), pp.168-174, Sousse, Tunis, March 2012. ISBN: 978-1-4673-1166-3, IEEE CN: CFP1245S-ART, 2012

[4] Adamopoulos, V. T., "The Effectiveness of Promotional Events in Social Media.", Workshop on Information Systems and Economic, 2014

[5] Mumbaikar, S., Padiya, P., "Web Services Based On SOAP and REST Principles", International Journal of Scientific and Research Publications, 3(5), ISSN 2250-3153, 2013.

[6] Halili, F., Rufati, E., & Ninka, I., " Service Composition Styles – Analysis and Comparison Methods", Proc. Book of IEEE CICSyN2013 5th International conference on Computational Intelligence, Communication Systems and Networks, pp.278-284, 5-7 June, Madrid, Spain.

ISBN: 978-0-7695-5042-8, BMS Number: CFP1381H-CDR, 2013.

[7] "From SOAP/1.1 to SOAP Version 1.2 in 9 points", https://www.w3.org/2003/06/soap11-soap12.html

[8] Halili, F., & Kasa, M., "Analysis and Comparison of Web Services Architectural Styles, and Business Benefits of their Use", Proc. Book of International Conference of Information Technologies and their importance in the economic development, 701-712, Tirana. ISBN: 978-99956-59-13-4, 2011.

[9] Kumari, V., "Web Services Protocol: SOAP vs REST", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2015.

[10] Wagh, K., & Thool, R. (2015). A Comparative Study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host. Marathwada Mitra Mandals Institute of Technology Pune, India - Shri Guru Gobind Singhji Institute of Engineering & Technology, Nanded, India.

[11] Wagh, K., & Thool, R., "A Comparative Study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host", Marathwada Mitra Mandals Institute of Technology Pune, India - Shri Guru Gobind Singhji Institute of Engineering & Technology, Nanded, India, 2015.

[12] Suda, B., "SOAP Web Services". University of Edinburgh, School of Informatics, Computer Science, 2003.