Differentiating Parameters for Selecting Simple Object Access Protocol (SOAP) vs. Representational State Transfer (REST) Based Architecture

Muhammad Waqas Khan and Eram Abbasi

***Abstract*—In the modern era of globalization business are expanding across boundaries. This initiates a strong need to connect various business processes, people and technologies. This necessitates the need for Enterprise Integration (EI). There are various tools & technologies among them one is Web Services. These are one of the most promising and widely used technologies for developing enterprise integration solutions. One of the solutions for enterprise integration is function/method based integration and Web Services plays a major role in this area. Among various web Service architecture SOAP and REST are the most talked of and the most debatable one as well. Though both have value but generally speaking REST has always been preferred choice over SOAP among solution developers. Among experts an argument prevails as when to use each one. This paper describes the architecture of SOAP based and REST based web services and then compared both using different parameters. The aim is to help solution developers in selecting best web service considering a given scenario. The comparison parameters include many dimensions such as Coupling, Format, CRUD Operations and Network Latency. The paper also discusses an example of implementing both types of web services on middleware framework and compare results on the basis of performance with regard to both efficiency and scalability.**

***Index Terms*—Middleware framework, REST, SOAP, web services.**

I. Introduction

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. [1] It is recognized by a URL and XML is used to define its interfaces and bindings [2]. For the purpose of enterprise integration the enterprise applications use different software applications that can run on a variety of platforms/ frameworks. Web services help in such cases by providing a standard means of interpretability [3]. Web services allow the users to get information from the server without connecting to the server or using a proper GUI. Web services instead share business logic, data and processes through a programmatic interface across a network. Developers can later on add the Web service to a GUI (such as a Web page or an executable program) to offer specific functionality to users. Web services typically include XML, SOAP, WSDL and UDDI for achieving its features. XML for tagging the data, WSDL used to describe the structure of the service and UDDI defines the list of web services which are available and SOAP used for transferring the data [4].

Manuscript received September 5, 2014; revised December 20, 2014.

The authors are with Institute of Business Administration, Pakistan (e-mail: waqas.khan@khi.iba.edu.pk, eabbasi@iba.edu.pk).

SOAP is an acronym of Simple Object Access Protocol. It is a set of standards specified by the W3C and is an alternative to REST.

SOAP is a messaging protocol used to translate the information in a Web service message (like request and response) before sending them over a network. It is based on XML [1]. In simple words it can be said that SOAP is a means of communication for a program running in one kind of operating system (e.g., Windows) with another program in the same or another kind of an operating system (e.g., Mac) by using HTTP and XML as the mechanisms for communication [3].

SOAP based web services are web services which are located on remote machines to allow access from anywhere. [5] SOAP typically follows some standards for sending and receiving data. [4] SOAP based web services are platform independent, operating system independent and development platform independent. Fig. 1 depicts this architecture. This means one can access any type of web service from anywhere on any platform. For example, to access the web service that has been developed using Java on MAC from the .NET platform on Windows. SOAP commonly uses HTTP, but other protocols such as SMTP may also be used.

SOAP (Simple Object Access Protocol) is a way for a program running in one kind of operating system (such as Windows 2000) to communicate with a progam in the same or another kind of an operating system (such as Linux) by using the World Wide Web's Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML) as the mechanisms for information exchange.

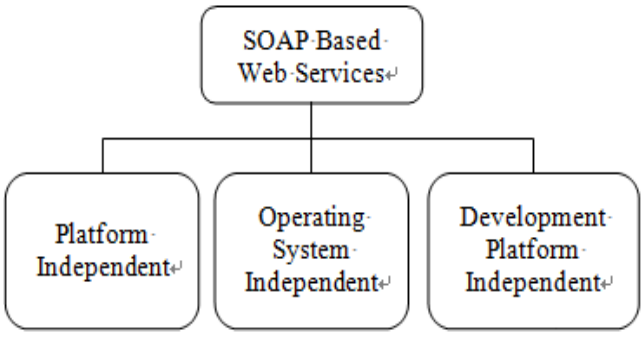

Fig. 1. SOAP architecture.

A typical SOAP communications architecture comprises of:

1) SOAP client: an application that sends and receives requests messages to and from the SOAP server.
2) SOAP server: an application that accepts requests from

DOI: 10.7763/JACN.2015.V3.143

the client and based on the requests it invokes services. Then sends the response from the service back to the client.

3) Actual service: is the actual functionality being performed

REST (RESTful) web services use client-server architecture. REST does not restrict client-server communication to particular protocol but is most commonly used with HTTP. [6] REST web services are based on URLs and four HTTP methods that are POST (For insertion of new record), PUT (For updating any existing record), DELETE (For removing any existing record) and GET (For getting any record's information) [7].

If both SOAP and REST are to be compared then the major differentiating areas are:

1) REST accesses resources whereas SOAP accesses operations.
2) In case of REST when handling CURD operations on data, it exposes only a single public API for handling these operations. SOAP exposes pieces of business application logic as service or it can be said that SOAP exposes operations.
3) REST accesses resources through a single consistent interface, whereas SOAP accesses operations which implement business logic through different interfaces
4) REST authorizes many different data formats whereas SOAP only authorizes XML
5) REST has better performance and scalability as REST reads can be cached but SOAP based reads cannot be cached [8].

Both types of web services have their advantages and disadvantages. Deciding which one to be preferred over the other is a difficult decision to make. Comparing both architectures using multiple parameters would aid the solution developer to make a better decision as which one to use and in what situations. The parameters selected for comparing both architectures are:

1) Weight
2) Coupling
3) Network Latency
4) Overhead
5) CRUD operations and
6) Security.

II. Parameters for Comparison

A. Weight

It usually defined the size of the service the more the weight the more difficult to achieve better performance. SOAP is heavier than the REST because of standards (which may be meaningless to user). [4] For example, if the response of any service is "Name" then SOAP response would be "Name" + SOAP standards while the response of REST would be just "Name". Therefore, SOAP is heavy weight than REST.

B. Coupling

It defines the binding or dependency of service to the server. SOAP based web services are usually tightly coupled to the servers whereas REST don't. We can say that REST services bind dynamically and SOAP web services are statically bound to the servers. This clearly gives REST an advantage over SOAP.

C. Overhead

REST based web services have no additional payload that means no additional overhead while in the case of SOAP there are additional tags of 'envelop' set and adds some headers which causes the additional payload and overhead to SOAP message. [4] Additional payload increases the latency time for requests, this makes REST more suitable than SOAP.

D. CRUD Operations

CRUD operations are database operations which are CREATE, READ, UPDATE and DELETE. CREATE used for inserting record, READ for getting selected record, UPDATE for updating any existing record and DELETE for removing any record. [6] REST take advantage of these simple operations and use them efficiently than SOAP. For example, if we are willing to remove any record from back-end data (such as item with particular ID) then for REST we have to just call DELETE method with item's ID in URL which allows REST service to omit internal XML payload, which results in reducing its packet size. Whereas for SOAP we have to include internal XML payload in order to achieve this functionality. This provides edge to REST over SOAP.

E. Network Latency

It refers to response time from the server for particular resource (service). In this dimension REST has clear edge over SOAP. REST latency rate increase in linear fashion but for SOAP it increases exponentially. And for small number of requests the difference between both the services is not much high but as the number of requests increases the difference between the services also increases [1].

Considering above discussion we can say that REST found to be more useful than its counterpart SOAP.

F. Security

Here security means the security of data which is moved from one place to other. As we know that SOAP services use specific standard for data transformation and they have specific security format as well. [4] While REST services are not using any specific security format, therefore they are easy to temper. SOAP services are more securable than REST because of the security format SOAP has.

III. Portal

Portal is an interaction layer which works as an intermediate between application code and the devices they support to perform operations. Portal also saves developers from writing the code for each other specific device to write only generic code and Portal do the rest for them. Its architecture typically divided into two main parts one is client and other is server. Client usually contains the plug-ins which translates input from supported devices for application code. Portal server contains the data which includes interaction techniques, application, and device and user profiles. Communication between the portal client and server will be done through Service Oriented Architecture [1].

IV. Portal Framework

Portal framework comprises of four major components named Plug-in manager, Portal controller, Data transmission component, and the server-side data model. [1] This framework is shown in Fig. 2.

A. Plug-in Manager Component

The Portal plug-in manager component is charged with ascertaining which Portal-supported devices a user decides to utilize for a similarly-supported application, monitoring user-initiated device input, translating this input into corresponding interaction techniques which the framework is configured to recognize, and relaying instances of these interaction techniques to the Portal controller component [1].

B. Controller Component

The controller component essentially acts as the central hub for interaction technique invocation events between devices and the applications [1].

C. Data Transmission Component

Responsibility of Data Transmission Controller is to act as a media between communication of client and server [1].

D. Data Model Component

This component represents the model for the back-end Data storage which contains the all related data and information [1].

E. Data Flow

On the client-side, the Portal controller is responsible for invoking commands related to manipulating application, device, or user profile information in a particular server's data model via this component. On the server-side, a data access object (DAO) is implemented as part of the data model component with, again, an interface method provided for each service in the Portal data transmission component service set [1].

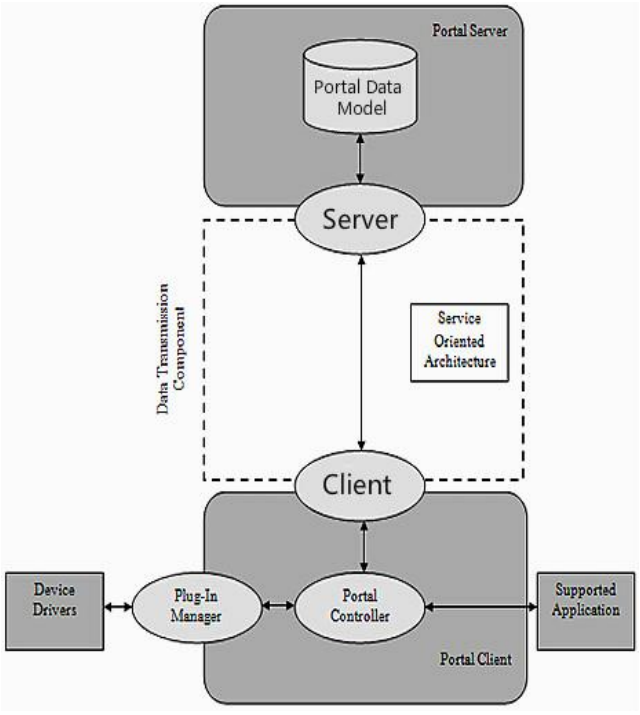

Fig. 2. Displays portal framework architecture [1].

V. Portal Test Results

REST and SOAP are implemented one by one by using the following strategies:

SOAP based architecture typically revolves around XML-encoded message transmitted over HTTP. First of all, WSDL file has to be written which contains Server side data model in the form of XML schema types. Once all this has set, now WSDL to be made public so that every portal client can reach the particular portal server, this can be done by using SOAP-enabled HTTP web server. Moreover server side code to be written in way that it will handle incoming service requests and formulating appropriate responses [1].

In REST case, any HTTP client library may be used to interact with REST server instead of specific SOAP client in case of SOAP based web services. Furthermore, there is no need of interpretation on the client's part [1].

A. CRUD Operations

Results for this test are shown on two factors one is latency and other is packet size.

Table I shows the comparison between SOAP and REST services on the basis of Latency. Latency rate of SOAP is more than the REST on every operation.

TABLE I: Table Showing Comparison between SOAP and REST on Latency Basis When Performing CRUD Operations

Latency (Milliseconds)	Acid	Update	Get	Remove
SOAP	450	225	125	115
REST	325	200	100	90

Table II shows the comparison between SOAP and REST services on the basis of Packet Size. SOAP requires more packet size than REST for completing its operations.

TABLE II: Table Showing Comparison between SOAP and REST on Packet Size Basis When Performing CRUD Operations

Packet Size(Bytes)	Acid	Update	Get	Remove
SOAP	2750	2900	2950	1100
REST	2500	2500	2500	500

For portal communication between the client and server, REST seems to be more obvious choice than the SOAP considering the above statistics.

B. Synchronous Request Test

In our scenario, synchronous request refers to the simultaneous number of queries to the plug-in for service.

Table III shows results on different number of synchronous requests for analyzing the performance of REST and SOAP services in terms of Latency.

TABLE III: Table Showing Comparison between SOAP and REST on Basis of Synchronous Requests Response

Number of Synchronous Requests	10	25	50	100
SOAP Latency(Milliseconds)	175	225	350	600
REST Latency(Milliseconds)	150	200	300	325

REST founds to be more efficient when considering the number of synchronous requests to handle.

VI. Conclusion

REST and SOAP both sorts of services have their own advantages and disadvantages. REST found to be more light weight and easy to develop which makes REST good for mobile applications. But when security is a prime concern then SOAP would definitely be a better choice, like in case of banking applications. Generally speaking SOAP has a lot more complexity attached to it as compared to REST. SOAP is first step towards achieving Service Oriented Architecture and set the platform for applications to move towards this dimension. REST follows the steps and now largely become first choice over SOAP and the reasons could be CRUD operations, Latency, and other factors.

Both SOAP and REST have their own positive characteristics and this is why it has been both have been compared. In fact the debate should not be about whether SOAP or REST is better, but in future the debate should be on the question; as to when can both be combined in order to reap maximum benefits of both together.

References

[1] G. Mulligan and D. Gracanin, "A comparison of SOAP and REST implementations of a service based interaction independence middleware framework," in *Proc. IEEE Simulation Conference*, 2009, pp. 1423-1432.

[2] S. Dustdar and W. Schreiner, "A survey on web services composition," *Int. J. Web and Grid Services*, vol. 1, no. 1, 2005.

[3] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard. (2004). Web services architecture. [Online]. Available: http://www.w3.org/TRws-arch/

[4] X. Shi, "Sharing service semantics using SOAP-based and REST web services," *IT Professional*, vol. 8, pp. 18-24, Mar. 2006.

[5] T. Ahikisalo and T. Paaso, "Latencies of service invocation and processing of the REST and SOAP web service interfaces," in *Proc. Services IEEE Eighth World Congress*, Jun. 2012, pp. 100-107.

[6] F. Belqasmi, J. Singh, S. Y. B. Melhem, and R. H. Glitho, "SOAP-based vs. RESTful web services: A case study for Multimedia Conferencing," *Internet Computing*, vol. 16, pp. 54-63, Aug. 2012.

[7] H. Su, B. Cheng, T. Wu, and X. Li, "Mashup service release based on SOAP and REST," in *Proc. Computer Science and Network Technology (ICCSNT) International Conference*, Dec. 2011, vol. 2, pp. 1091-1095.

[8] S. Francia. REST Vs SOAP, the difference between soap and rest. [Online]. Available: http://spf13.com/post/soap-vs-rest

Muhammad Waqas Khan was born in Hyderabad, Pakistan in 1986. He completed his bachelor's degree in computer science from Department of Computer Science University of Karachi, Pakistan, in 2009. Now he is an M.S. student of Computer Science Department at Institute of Business Administration, Karachi, Pakistan. His research focuses on web development and data warehousing.

He has working experience of over 4 years as a software engineer. Currently he is working as a senior software engineer in Karachi, Pakistan.

Eram Abbasi received her MS degree in computer science from SZABIST Karachi and now is a PhD candidate at the same university. Her area of research is knowledge management, project management and technology entrepreneurship.

Currently she is a faculty member at IBA Karachi. She has over 15 years working experience of corporate as well as academic experience. Her work portfolio comprises of a working experience with organizations like Hewlett Packard (HP), IBM and universities in Pakistan like IBA, Bahria University, IoBM and PAF KIET. She has worked on various research projects and represented Pakistan at various international platforms, presented research papers at international conferences and has also served as reviewer and session chair for many international conferences.

She has research papers and articles published in international publications like IEEE, ACM, and Springer. She is the chief editor of a monthly newsletter by IEEEP. She has contributed chapters in three books published in USA and wrote reviews for several books published by international publishers.

Prof. Abbasi is also an active member of various professional bodies like IEEEP, OPEN (USA), QPSP (Quality & Productivity Society of Pakistan), IRMA–USA (Information Resource Management Association). In recognition of her academic & educational efforts she was nominated for the "Women Educationist of the Year" award in 2009.

Network Security