

Web Services Protocol: SOAP vs REST

Vibha Kumari

Abstract – Web service makes the work easier by binding common functionality into one entity. But the question arises as what to which to use and when? Basically SOAP and REST are two protocol used to communicate for exchange of message. SOAP (Simple Object Access Protocol) and REST (Representational State Transfer) approaches work, but both have advantages and disadvantages to interacting with web services, and it is up to the web developer to make the decision of which approach may be best for each particular case. SOAP (Simple Object Access Protocol), was created in 1998 by Dave Winer in Collaboration with Microsoft. Develop by a large software company, this protocol addressed the goal of addressing the needs of enterprise market. REST (Representational State Transfer) was created in 2000 by Roy Fielding in UC, Irvine. Developed in an academic environment, this architectural style embraces the philosophy of the open web. Sometimes it's very difficult to analyze which protocol to use, so this paper might help you to find the situation where we need to use which protocol.

Index Terms – SOAP (Simple Object Access Protocol), REST (Representational State Transfer), Protocol, Web services, Architectural Style

I. INTRODUCTION

Web Services: A Web service is a method of communication between two electronic devices over a network. Web services are application components can communicate using open protocols, self-contained and self-describing, can be discovered using UDDI, and HTTP and XML is the basis for Web services. For communication we can use SOAP or REST. But the question arises which one to use and when? This paper will give you answer for this question.

We have something called as "Web API" which helps in development for Web services. In this, we are preferring simpler representational state transfer (REST) based communications. Basically RESTful APIs do not require XML-based Web service protocols (SOAP and WSDL) to support their interfaces.

So now we will define **SOAP** and **REST**

"SOAP: SOAP is a messaging protocol that allows programs that run on disparate operating systems (such as Windows and Linux) to communicate using Hypertext

Transfer Protocol (HTTP) and its Extensible Markup Language (XML)."

"REST: Representational State Transfer (REST) is a software architecture style consisting of guidelines and best practices for creating scalable web services. REST is a coordinated set of constraints applied to the design of components in a distributed hypermedia system that can lead to a more maintainable architecture. REST efficiently uses HTTP verbs"

Before proceeding ahead let's have a look at software architecture style. An architectural pattern is a solution to a frequently occurring problem in software architecture within a given context.[1] Architectural patterns are similar to software design patterns but have a broader scope. The architectural patterns address various issues in software engineering, such as computer hardware performance limitations, high availability and minimization of a business risk. Some architectural patterns have been implemented within software frameworks.

II. Pros and Cons Of SOAP and REST

I would prefer SOAP because:

1. Its platform and language independent.
2. Uses XML to send and receive messages.
3. Its vendor neutral.
4. Utilizes WS-* efficiently along with security.
5. Its firewall friendliness.
6. Universally accepted i.e. cost is not too high for implementation.
7. It also supports asynchronous messaging.
8. It makes data available as services.
9. WSDL fully describes SOAP.

Why I would not prefer SOAP:

1. Too much reliance on HTTP
2. It's not stateless.
3. At times it's slow too because of XML generation. Also the bandwidth gets heavier due to its format for message generation.

Moving ahead let's see pros and cons of REST. Following are advantages of REST:

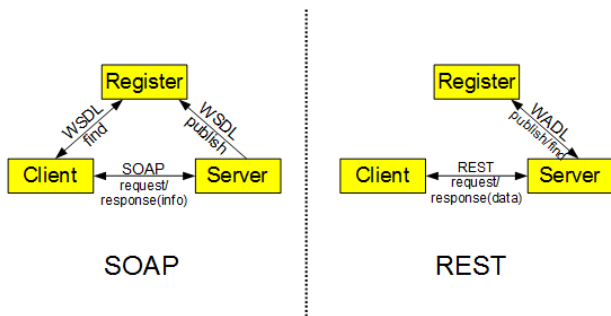
1. It supports stateless communication.
2. It's simple to learn and implement.
3. Efficiently uses HTTP verbs.

4. Light bandwidth since its passes message is JSON (JavaScript Object Notation) format also it can use multiple other formats.
5. For security it uses HTTP standards.
6. REST can be consumed by any client.
7. It makes data available as resource.

Disadvantages of REST:

1. It's not suitable for large amount of data.
2. Comparative SOAP it does not cover all varieties of web service standards like Security, Transactions etc.
3. REST is not reliable.

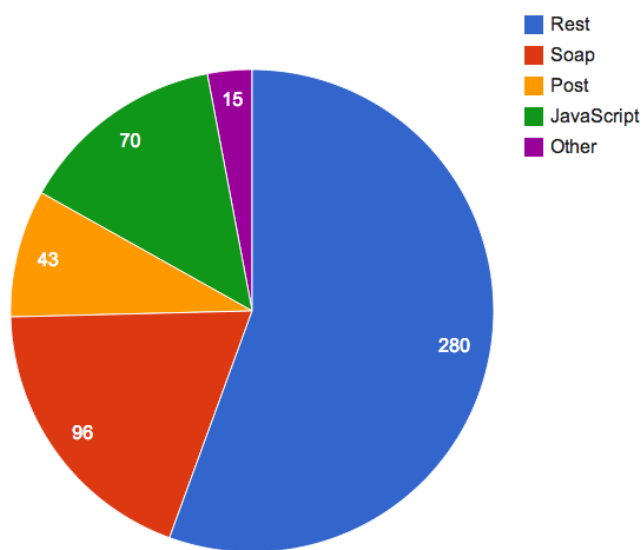
III. Flow of SOAP and REST



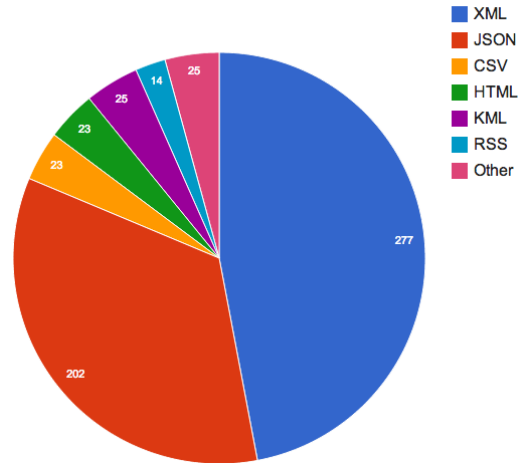
IV. Compare SOAP and REST

Following are below examples shown the need of individual protocol in different sectors. First is Protocols used in mapping APIs

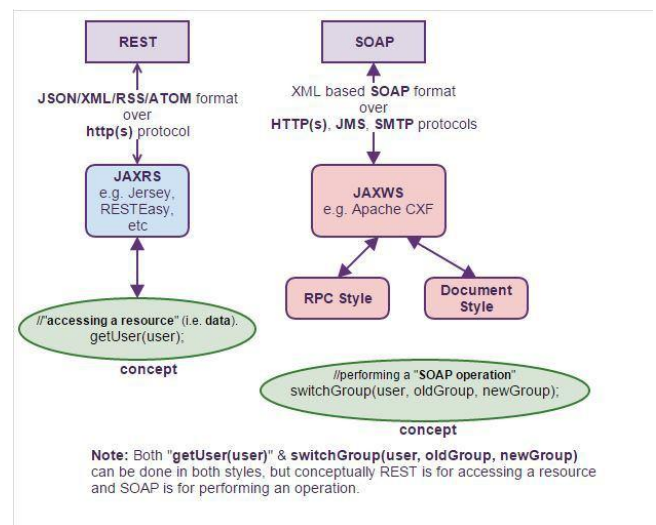
Protocols Used in Mapping APIs



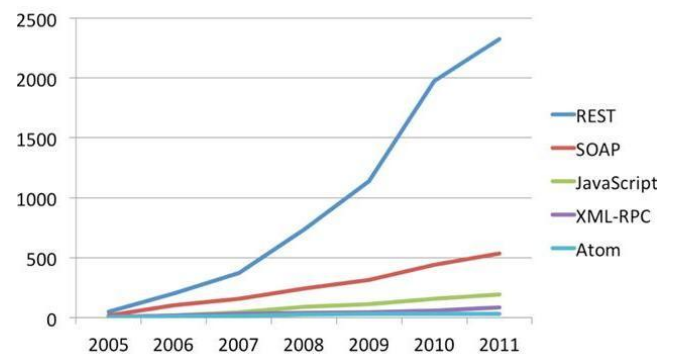
Data Formats Used in Mapping APIs



Let's have a look at below flow chart, which shows REST and SOAP accessing method in their own way.



Below is the graph shown, indicating how REST took over SOAP.



V. CONCLUSION:-

When to use REST:

1. When client and servers operate on a web environment;
2. When information about objects doesn't need to be communicated to the client.

When to use SOAP:

1. Client needs to have access to objects present on server.
2. Impose formal contract between client and server.

Do not prefer SOAP if majority of developer are dependent on APIs. Also do not use REST when you need multiple calls to perform a transaction.

Mostly SOAP's are preferable for financial, banking, telecommunication services, and REST preferred for Social interaction, WEB chat, and mobile services.

Basically use of either of protocol depends on the use of it in your project. For sensitive and security of data prefer SOAP otherwise if utilizing API more often then go for REST.

VI. ACKNOWLEDGMENT

This research wouldn't be possible without the guidance and support from our faculty and help from friends. Also would like to thanks to one and all for directly and indirectly supporting us in completion of research paper.

VII. REFERENCES

- [1] <http://en.wikipedia.org/wiki/SOAP>
- [2] http://en.wikipedia.org/wiki/Representational_state_transfer
- [3] http://en.wikipedia.org/wiki/Web_service
- [4] <http://stackoverflow.com/questions/19884295/soap-vs-rest-differences>
- [5] <http://stackoverflow.com/questions/4163066/rest-vs-soap-has-rest-a-better-performance>
- [6] <http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/>
- [7] REST vs. SOAP: Making the Right Architectural Decision by Cesare Pautasso from Faculty of Informatics University of Lugano (USI), Switzerland;
- [8] Sharing Service Semantics using SOAP-Based and REST Web Services by Xuan Shi
- [9] SOAP and Web Services by Panagiotis Louridas
- [10] SOAP-Based vs. RESTful Web Services by Fatma Belqasmi, Jagdeep Singh, Suhib Younis Bani Melhem, and Roch H. Glitho from Concordia University

- [11] [https://msdn.microsoft.com/en-us/library/vstudio/hh273094\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/vstudio/hh273094(v=vs.100).aspx)
- [12] <http://www.slideshare.net/Muratakal/rest-vs-soap-15854355>
- [13] <https://www.api2cart.com/blog/rest-vs-soap-feel-difference-infographic/>



Vibha Kumari born in Mumbai, Maharashtra India. She is pursuing MCA final year in ASM's Institute of Management & Computer Studies ,IMCOST, Thane