Web Service Assignment

SOAP vs REST

Nicholas Randles – B00058026

Contents

Abstract

This research paper was written to compare SOAP and REST within the area of web services. The paper starts off by explaining what each web service is and how they both work. The paper discusses the SOA and ROA architectures in relation to SOAP and REST. The paper then discusses security for the web services. It then discusses the different description languages used by the web services and why they are used. The paper then discusses when it is suitable to use one web service over the other. Finally, the paper finishes off by discussing web services growth in the real world and which web service has gained an edge over the other.

Research Paper

Introduction

The focus of this research paper is to explore the differences between SOAP and REST web services. This research paper will focus on six main topics of research. The six topics are:

- What are SOAP and REST and how they work
- SOAP vs REST in SOA/ROA Architecture
- Web Services Security for SOAP and REST
- WSDL vs WADL
- When to use SOAP versus REST
- Who uses what in the real-world

What are SOAP and REST and how they work

SOAP and REST both provide us with ways to access web services in their own way. Simple Object Access Protocol (SOAP) is an object access protocol which was created in 1998 by Microsoft [1]. SOAP is an XML based protocol which was designed to send and receive XML information. All messages sent in SOAP must be in XML [2]. All messages are made up of three sections, which are the envelope, header and body [3]. The envelopes job is to encapsulate the message header and body and provide important information such as how the message should be processed to the target.

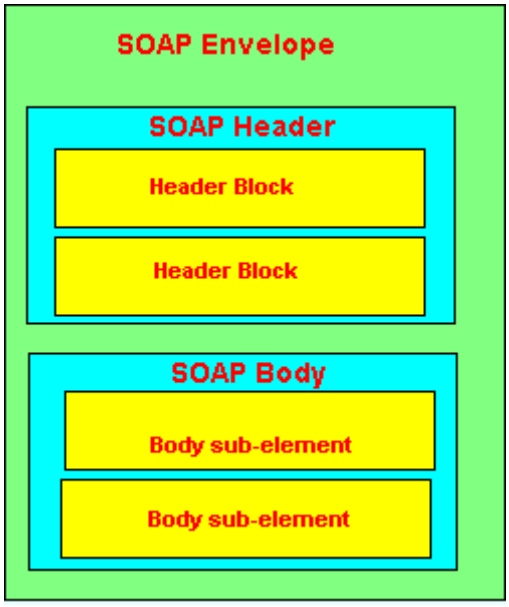

Figure 1: SOAP message structure [3]

The header in a SOAP message is not required but messages will put them in when SOAP is being used in web services. Headers are used to provide information which can help with

tasks such as authentication. Header data is put into a message in header blocks. There can be more than one header block in the message. The body in a SOAP message is the part that contains that actual data which is being send. This data can be a request for information or it can be a response to a request for information. This data is placed into one or many sub elements.

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:hypotenuse xmlns:ns2="http://Hypotenuse/">
            <arg0>3.0</arg0>
            <arg1>4.0</arg1>
        </ns2:hypotenuse>
    </S:Body>
</S:Envelope>
```

Figure 2: SOAP request example

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:hypotenuseResponse xmlns:ns2="http://Hypotenuse/">
            <return>5.0</return>
        </ns2:hypotenuseResponse>
    </S:Body>
</S:Envelope>
```

Figure 3: SOAP response example

Representational State Transfer (REST) is an architectural style which was created in 2000 by Roy Fielding at the University of California, Irvine [4]. They are web services designed in accordance with a set of architectural principles that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a wide range of clients written in different languages [5]. It is called the representation state transfer because it puts the client into a different state with each resource representation. In rest web services everything is considered a resource. Instead of using complex mechanisms like SOAP to communicate between devices, REST uses simple HTTP requests. They use HTTP requests to create data, read data, update data and remove data (CRUD).

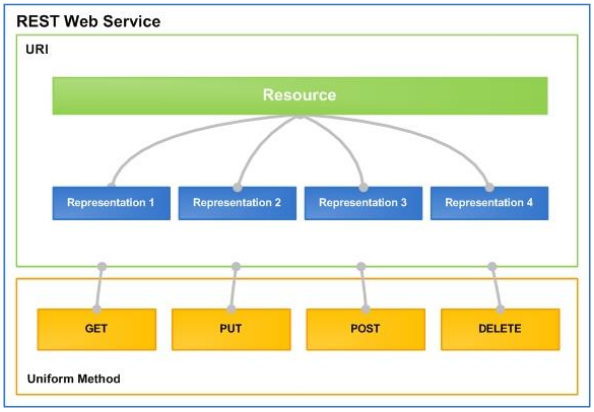

Figure 4: Design structure of a rest web service [6]

SOAP vs REST in SOA/ROA Architecture

Service Oriented Architecture (SOA) and the Resource Oriented Architecture (ROA) are architectural approaches [7]. SOA is commonly used for SOAP whereas ROA is commonly used for REST. They are the structure of the application and they describe how it should work. SOA is a set of guidelines that describes how to design a system to use business processes as services. Each services carries out its own task and does not interact with any of the other services. Rather than using an object oriented approach, SOA aims to allow large blocks of functionality to be rearranged to form new applications. ROA is set of guidelines to implement a RESTful approach. They develop software in the form of resources. These resources can be reused for different purposes.

Both SOA and ROA have advantages over each other and are suitable in different scenarios. One of the main advantages SOA has, is its more mature tool support. This could change overtime as ROA continues to evolve. One of the main advantages ROA has over SOA is that it is easier to implement and takes a more lightweight approach. SOAP-based SOA is more commonly used for businesses, especially in industries such as accounting and finance whereas REST-based ROA is more commonly used by people who want something up and running easily and quickly, that has low overhead and good performance [8].

Web Services Security for SOAP and REST

While using web services it is important that we are able to keep are data safe. We want to show are data but not to the wrong people. SOAP and REST both use HTTP requests and responses for data transfer but they use different formats so therefore they have different security problems that need to be considered. SOAP has been around a long time and continues to improve its security features over the years. It has implemented security patterns such as XML-Encryption and XML-Signature to help tighten up security while sending and receiving messages. On the other hand REST has not implemented any specific security patterns because the patterns focus on how to send and receive data and not how to build in safety into the way you exchange data [9].

WSDL vs WADL

WSDL is the Web Services Description Language. It is mostly used to outline the details of the services provided by a SOAP server [10]. It is XML based and specifies the location of the service and its methods using XML elements. Originally it was not very suitable for REST web services as it did not support HTTP requests other than GET and POST. This made it unsuitable for REST as other HTTP requests such as PUT and DELETE might be used. Although this problem was later fixed in WSDL version 2.0.

```
<?xml version="1.0" encoding="UTF-8"?>
    <definitions name="BookQuoteWS" targetNamespace="http://www.Monson-
    Haefel.com/jwsbook/BookQuote" xmlns:mh="http://www.Monson-
    Haefel.com/jwsbook/BookQuote"
    xmlns:soapbind="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <!-- message elements describe the input and output parameters -->
    <message name="GetBookPriceRequest">
          <part name="isbn" type="xsd:string" />
    </message>
    <message name="GetBookPriceResponse">
          <part name="price" type="xsd:float" />
    </message>
    <!-- portType element describes the abstract interface of a Web service -->
    <portType name="BookQuote">
          <operation name="getBookPrice">
                <input name="isbn" message="mh:GetBookPriceRequest"/>
                <output name="price" message="mh:GetBookPriceResponse"/>
          </operation>
    </portType>
```

Figure 5: WSDL example [11]

Web Application Description Language (WADL), was developed as an alternative for WSDL. WADL has several advantages over WSDL, it is more lightweight, easier to understand and write. It may not be as flexible as WSDL in some respects but it is good enough for all REST web services as they tend to not need as much explanation.

```
<application xmlns:xsi="…">
     <grammars>
        <include href="ticker.xsd"/>
     </grammars>
     <resources base="http://www.example.org/services/">
        <resource path="getStockQuote">
           <method name="GET">
                <request>
                   <param name="symbol" style="query"
                       type="xsd:string"/>
                </request>
                <response>
                   <representation mediaType="application/xml"
                       element="ex:quoteResponse"/>
                   <fault status="400" ediaType="application/xml"
                       element="ex:error"/>
                </response>
           </method>
        </resource>
     </resources>
</application>
```

Figure 6: WADL example [12]

When to use SOAP versus REST

Both SOAP and REST are good choices to use when developing web services. They can both help us solve a large amount of problems and challenges we might encounter while developing web services. Each one has advantages over the other and are more suitable in different situations. REST is the easier of the two to understand and implement but does lack industry standards. Whereas SOAP is considered a well-defined protocol with an industry standard and a set of established rules [13].

REST is suitable to use when:

- Limited resources and bandwidth: The return structure can be in any format that has been defined. Any browser can be used for REST because it uses HTTP requests.
- Total stateless operations: Good for carrying out stateless operations such as CRUD (Create, Read, Update, Delete).

SOAP is suitable when:

- Asynchronous processing: Can provide applications with a guaranteed level of security and reliability. Provides additional standards that make sure this is carried out.
- Stateful operations: Good for applications which need operations to be continued such as transactions.

Who uses what in the real-world

As the web service trend continues to grow more and more web services are available to developers every year. 'Programmable web' [14] has been tracking the availability of web services online since 2005 and they have discovered that they have grown from approximately 105 to 5000 plus in 2011. The most recognized ones has been provided by companies such as Amazon, Google and Tesco. This growth is shown in figure 7.

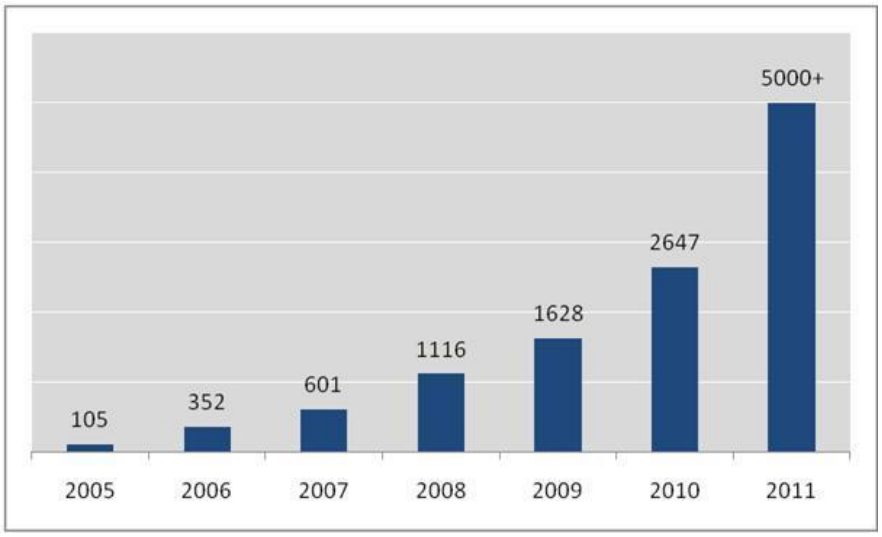

Figure 7: Web services growth [14]

‘Programmable web’ [14] was also able to find out which web service type was more popular as well. They discovered that some companies provide both SOAP and REST web services, but this has started to decline over the years. More companies are now just providing REST web services. This is shown in figure 8.

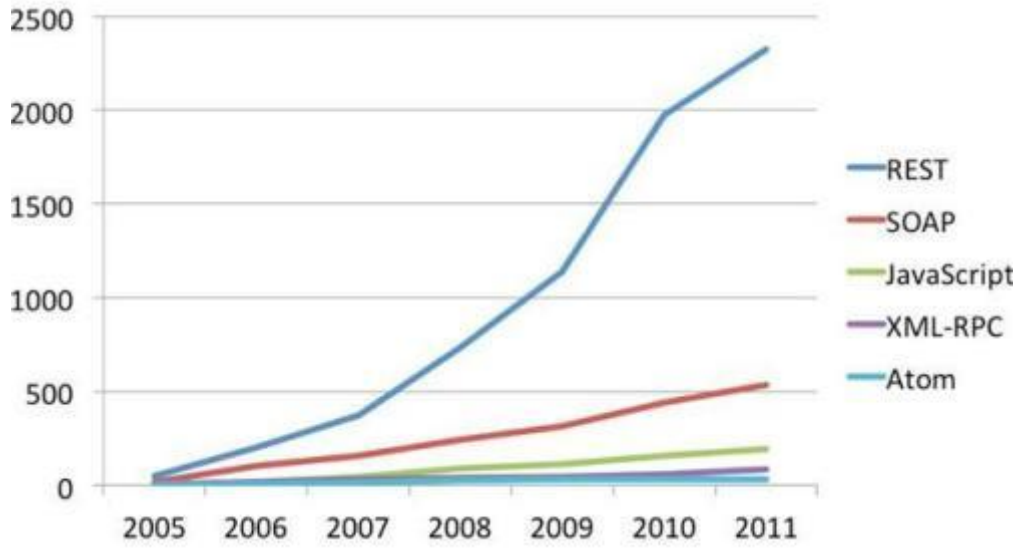

Figure 8: Popularity of different web service types [15]

Conclusion

Based upon the research that has been currently undertaken in the field of web services it is clear that REST web services have over taken SOAP web services in terms of popularity over the years, mainly down to their simple and easier approach. SOAP till does have its place in the web development world though as it provides a different approach which is more suitable in some scenarios. They both provide the developer with a different way to create a web service. They both have their own structures and ways of requesting data and responding to requests. They have their own security problems and different ways of handling them.

References

[1] "Exclusive .NET Developer's Journal "Indigo" Interview with Microsoft's Don Box | Microsoft Cloud " [online]. Available: http://dotnet.sys-con.com/node/45908. [Accessed 4/4/2016].

[2] "Understanding SOAP and REST Basics and Differences " [online]. Available: http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/. [Accessed 4/4/2016].

[3] "Understanding how SOAP Works " [online]. Available: http://searchsoa.techtarget.com/tip/Understanding-how-SOAP-works. [Accessed 4/4/2016].

[4] "Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST) " [online]. Available: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm. [Accessed 4/4/2016].

[5] "Learn REST: A Tutorial " [online]. Available: http://rest.elkstein.org/. [Accessed 4/4/2016].

[6] "Apache Wink : 1 Introduction to Apache Wink " [online]. Available: http://wink.apache.org/1.0/html/1 Introduction to Apache Wink.html. [Accessed 4/4/2016].

[7] "AKF Partners | SOA Vs ROA " [online]. Available: http://akfpartners.com/techblog/2008/04/08/soa-vs-roa/. [Accessed 4/4/2016].

[8] "Learn REST: A Tutorial: 10. ROA Vs. SOA, REST Vs. SOAP " [online]. Available: http://rest.elkstein.org/2008/02/roa-vs-soa-rest-vs-soap.html. [Accessed 4/4/2016].

[9] "API Security in REST Vs SOAP " [online]. Available: http://blog.smartbear.com/security/api-security-in-rest-vs-soap/. [Accessed 4/4/2016].

[10] "Learn REST: A Tutorial: 11. Documenting REST Services: WSDL and WADL " [online]. Available: http://rest.elkstein.org/2008/02/documenting-rest-services-wsdl-and-wadl.html. [Accessed 4/4/2016].

[11] "Jee Course Web Services " [online]. Available: http://www.slideshare.net/odedns/jee-course-web-services. [Accessed 4/4/2016].

[12] "Putting SOAP to REST " [online]. Available: http://www.slideshare.net/igor.moochnick/put-soap-to-rest-2774452. [Accessed 4/4/2016].

[13] "REST and SOAP: When should I use each (Or both)? " [online]. Available: http://www.infoq.com/articles/rest-soap-when-to-use-each. [Accessed 4/4/2016].

[14] "How REST Replaced SOAP on the Web: What it Means to You " [online]. Available: http://www.infoq.com/articles/rest-soap. [Accessed 4/4/2016].

[15] "API Growth, SOAP v REST, etc" [online]. Available: https://www.dinochiesa.net/?p=259. [Accessed 4/4/2016].