

Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics

Juris Tihomirovs¹, Jānis Grabis²

^{1,2} Riga Technical University

Abstract – The usage of Web services has recently increased. Therefore, it is important to select right type of Web services at the project design stage. The most common implementations are based on SOAP (Simple Object Access Protocol) and REST (Representational State Transfer Protocol) styles. Maintainability of REST and SOAP Web services has become an important issue as popularity of Web services is increasing. Choice of the right approach is not an easy decision since it is influenced by development requirements and maintenance considerations. In the present research, we present the comparison of SOAP and REST based Web services using software evaluation metrics. To achieve this aim, a systematic literature review will be made to compare REST and SOAP Web services in terms of the software evaluation metrics.

Keywords – Literature review, REST, SOAP, Web services.

I. INTRODUCTION

A. Problem Formulation

Nowadays the most common way to exchange data among information systems is to use Web services. Web services are self-contained, modular and dynamic applications by their nature [1]. Most common implementations are based on SOAP (Simple Object Access Protocol) and REST (Representational State Transfer Protocol) styles. Each of these approaches has its own advantages and disadvantages, so it is important to choose the right type of Web services, otherwise it can lead to certain problems in data exchange or impose some restrictions. This paper compares SOAP and REST approaches to give recommendations on how to opt the right Web service type at a project design phase.

There are a number of related studies on these protocols, primary benefits and final value [2], [3], [4]. Other articles review real implementation examples [5], [6]. This research paper summarises these studies and gives a comparison, which helps identify the key differences and benefits of SOAP and REST protocols.

B. Aim and Tasks

The **aim** of the paper is to summarise the main SOAP and REST protocol advantages and disadvantages.

To achieve this aim, the following **tasks** are set:

- (1) to define the metrics by which it is possible to mutually compare SOAP and REST protocols;
- (2) to compare SOAP and REST by found metrics;
- (3) to evaluate which protocol type is better on the basis of software evaluation metrics.

II. THEORETICAL BASIS

This section addresses the key concepts that will be used in the paper.

A. Web Services

A single Web service consists of service and service description where service is a software module offered by a service provider, which is available through the Web. A service description contains the details of the service interface and implementations, including data types, metadata, categorisation information and the location where the service is exposed [3], [1]. Service description is published in the service registry, where services are displayed and grouped by their purpose. A web service is the key element in the integration of different information systems, as information systems can be based on different platforms, programming languages and technologies.

B. SOAP

SOAP compared to REST is an older protocol, which was developed as an alternative to CORBA (Common Object Request Broker Architecture) standard. To ensure data transport in SOAP, protocols such as HTTP, SMTP, etc., are used, while the data are sent in XML format [3]. The main principle of this approach is as follows: a service provider publishes a service description or interface to the service registry, so the service requester can find a right service instance and use it [7].

The amount of data sent by SOAP can cause some performance problems because when forming the message SOAP adds an additional header and body parts to the message. SOAP-based Web services include a variety of standards, such as WSDL, WSBPEL, WS-Security, WS-Addressing (responsible for the Web service and message addressing) [8]. These standards were developed by standardisation organisations, such as W3C and OASIS.

C. REST

REST is a newer approach, which uses the HTTP protocol to transmit data, while the data are formed by XML, JSON, etc. formats [3]. It simplifies access to Web services by using the existing and well-known standards instead of adding a new data processing layers on the transmission and communication stack [9]. Thus, REST tends to be a lighter alternative to the heavy SOAP protocol. REST Web services are based on self-defining resources where the HTTP protocol is used to reach them. A service is provided as a resource that can be identified by URI (Uniform Resource Identifier) [8]. For example, if URI <http://www.example.com/rest/user/1> is opened in the Web

browser, the Web service will return information about a user whose ID is “1”. In order to perform data operations with the service, standard HTTP methods, such as GET, PUT, POST, DELETE, etc., are used [10].

D. Software Evaluation Metrics

There are two types of metrics, which are used in terms of software development [2]: direct, measurable metrics (errors, cost, etc.) and indirect or non-measurable metrics (complexity, maintainability, etc.). The most popular evaluation metrics are summarised in Table I.

TABLE I
CATEGORIES OF THE METRICS [2]

Direct	Indirect
Cost	Functionality
Effort	Quality
Lines of code	Complexity
Execution speed	Efficiency
Memory	Reliability
Errors	Maintainability

These metrics will be used to compare SOAP and REST in this paper.

III. RESEARCH METHOD

This section describes the research method that will be used for a systematic literature review.

A. Data Sources

For our research, we have included the following three electronic library sources as data sources: SCOPUS, Web of Science and IEEE Xplore Digital Library. We have constructed a search string using “SOAP” and “REST” as the main keywords (see Table II) and included synonyms and related terms. The search string is constructed using Boolean **AND** and **OR** operators to connect these keywords in query string (1). Search query can be represented as follows:

$$(S_1 \text{ OR } S_2 \dots \text{ OR } S_n) \text{ AND } (R_1 \text{ OR } R_2 \dots \text{ OR } R_n), \quad (1)$$

where S covers SOAP keywords and R covers REST keywords.

TABLE II
SEARCH KEYWORDS AND SYNONYMS

SOAP	REST
Simple Object Access Protocol	Representational state transfer
SOAP Web services	RESTful
SOAP Web service	RESTful Web services
	RESTful Web service

The resulting search string is shown in (2):

(SOAP OR “Simple Object Access Protocol” OR “SOAP Web services” OR “SOAP Web service”) AND
(REST OR “Representational state transfer” OR RESTFUL OR “RESTFUL WEB SERVICES” OR “RESTFUL WEB SERVICE”)

The search string has been executed in the digital library services to titles, abstracts and metadata. In order to minimise the found data sets, we have defined the criteria that the article must be in the field of computer science or IT, as well as the article must be written after 2010, as the earlier publications may contain information that is no longer up to date. Number of the found information sources is displayed in Table III.

TABLE III
OVERVIEW OF INFORMATION SOURCES

No.	Source	Format	The number of information sources
1	Scopus	Mendelay	190
2	Web of Science	MS Excel	93
3	IEEE Xplore Digital Library	MS Excel	352

B. Study Selection

Since the result, which is obtained using the search query, contains 635 entries where not all the entries correspond to the theme of this study, it is necessary define the inclusion and exclusion criteria in order to limit the result set. The inclusion and exclusion criteria are defined in Table IV.

TABLE IV
INCLUSION AND EXCLUSION CRITERIA FOR STUDY SELECTION

Inclusion criteria	Exclusion criteria
I1. A study in the form of a scientific peer-reviewed paper. Motivation: A scientific paper guarantees high reliability and quality level.	E1. The study is reported in a language other than English.
I2. A study is focused on comparing SOAP with REST or one of these protocols is reviewed Motivation: The main focus of this research is to compare SOAP with REST. There is no particular need to look at other protocols.	E2. It is not a scientific publication (e.g., book chapter)
	E3. A study that is related to other protocol types than SOAP or REST

C. Data Synthesis

The search string was executed in the digital libraries and after that the study selection was extracted in an Excel spreadsheet. The following details were included about the study: title, authors, publication year, publication form (journal/conference/etc.) and abstract. First selection round was based on the “title and abstract” of the study, where the study was categorised as follows: (I) relevant, (II) irrelevant and (III) moderate. Articles in the moderate category were reviewed in full with the result that they were either included or not included in the relevant studies group. This step resulted in 122 studies.

In the next step, articles were evaluated based on inclusion and exclusion criteria, which resulted in 14 primary studies.

The review process can be summarised by four main steps:

Step 1: Creating a search string, executing search string in the digital libraries and saving the results into the Excel spreadsheet;

Step 2: The result relevance is rated based on the titles and abstracts. This step also includes the duplicate checking;

Step 3: The inclusion and exclusion criteria are applied;

Step 4: This step covers the article analysis and after that the conclusions are made.

Figure 1 shows the review process with a number of articles.

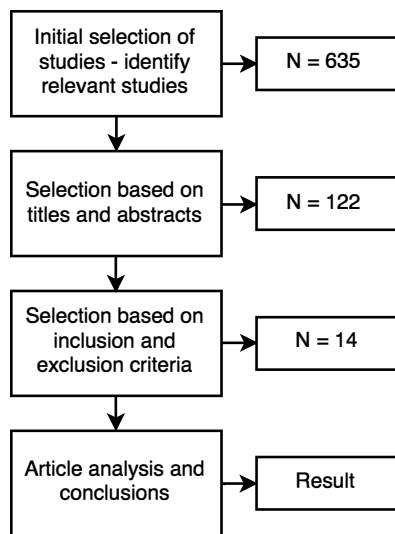


Fig. 1. The review process with a number of studies.

IV. COMPARISON METHOD

In the context of software development, there are direct and indirect metrics [2]. These metrics will be taken as the basis for the comparison method. These direct and indirect metrics will be used to point out the key principles that should be taken into the account when selecting between SOAP or REST approaches.

A. Cost

REST is based on a simple technological infrastructure, where the service can be developed with minimal development tool usage. It automatically reduces the cost that would be needed to achieve the same result with the SOAP approach [4]. In addition, it is possible to work without REST client at the

beginning of the development or testing phases, as access to the REST service can be ensured directly from the Web browser by accessing the resource address.

B. Effort

In order to reduce maintenance costs when developing and publishing Web services on the Web, it is advised to use the REST approach. In turn, if there is a need to use any other Web services to reduce maintenance costs, the recommendation would be to use a SOAP WSDL approach [2].

C. Lines of Code

A recent study [2] compared SOAP with REST to find out how many lines of code are needed to ensure the same functionality. In order to develop a REST service, there were 4016 lines of code written. In turn, to develop a SOAP service, there were 3844 lines of code written. Accordingly, in order to implement the REST client, there were 1117 lines of code written, and to realise SOAP client there were 509 lines of code written.

D. Execution Speed

Web service execution speed can be objectively measured by response time when a Web service returns a response to the client. Response time comparison between SOAP and REST approaches was made by the University of North Florida [8]: A SOAP client makes a request to the SOAP service, where it selects or adds data to the database, depending on the type of request. As a result, the response time from service was measured. In the same way, requests through the REST service were made. The experiment resulted in a conclusion that REST showed shorter response times and better data throughput than the SOAP protocol.

In [11] it is marked out that SOAP is not suitable in cases of slow data transfer speed or the network has a large load.

In [10] a response time measurement experiment was made by simulating a simple client-server scenario: (1) create a user, (2) user saves a message, (3) a request is made to query user's messages and (4) delete a user. The experimental results show that the SOAP-XML service processing time is 4–5 times higher compared to the REST approach (Fig. 2).

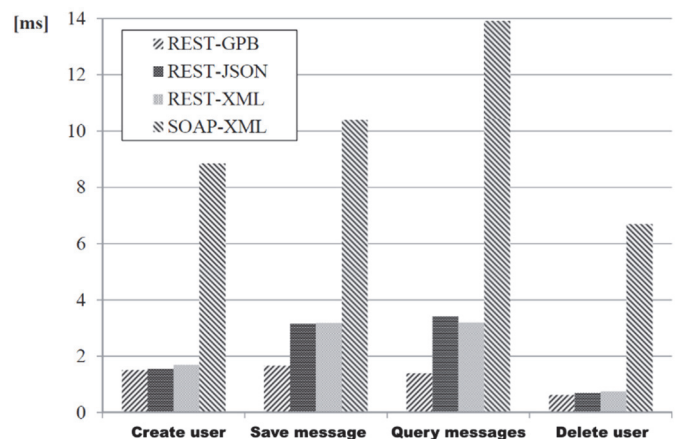


Fig. 2. REST and SOAP response time measurements [10].

In [7] an experiment was made to measure time when 100 and 1000 character messages were sent using REST and SOAP. REST showed better performance in this experiment: response time of 100 character message ($2.56 \text{ s} \pm 0.10 \text{ s}$) and 1000 character message ($3.45 \text{ s} \pm 0.10 \text{ s}$) was significantly lower compared to SOAP time, when 100 character ($5.64 \text{ s} \pm 0.17 \text{ s}$) and 1000 character ($5.83 \text{ s} \pm 0.17 \text{ s}$) messages were sent. This can be explained by the fact that it is not necessary to build an XML (so the time reduces), which is required for data transmission in the case of SOAP. Consequently, it can be concluded that REST has a faster execution speed than SOAP.

E. Memory

Creating a SOAP request message from a mobile device, ten times longer processing time is consumed in comparison with the REST service [5], [6]. Thereby up to 8 times more memory resources are consumed. This can be explained by the fact that mobile devices are resource limited, which means that more time to process a message is needed compared to the standard PCs. In addition, it is mentioned that REST has an advantage over SOAP when bandwidth and resources are limited [7].

F. Errors

It is possible to import a SOAP WSDL file when using IDE tools, so the client-side code can be automatically generated by the tools. This helps reduce the number of errors that could be tainted if the code were written by a programmer manually. However, it is possible to describe REST Web service using WADL definition. WADL is the REST equivalent of SOAP's WSDL. In the study [2], which is about SOAP and REST service comparison, it is mentioned that the automatic code generation using the WSDL file helped students from the University of Sao Paulo reduce the number of errors and made source code maintenance easier. In addition, SOAP has built-in error handling mechanism [7] that is an advantage over REST.

G. Functionality

One of the advantages of SOAP is that the same SOAP message in the same format can be sent through several middleware systems, which may be based on HTTP or any other protocol [4]. There can be cases where the transport protocol may change along middleware systems – it is not a nuisance to deliver SOAP messages from point A to B. As another advantage of SOAP, it can be mentioned that it supports asynchronous requests, which in turn is not possible when using REST [7].

Nowadays the widely used system integration approaches are point-to-point and distributed computing. By these two approaches, the differences between REST and SOAP are marked – REST is not applicable in the distributed computing environments and is suitable for point-to-point integrations, while SOAP is suitable for use in the distributed computing environments, where the message can be sent through one or more intermediaries [7], [12].

H. Quality

The main Web service quality characteristics [13] are: service time, reliability, execution price, availability,

performance, security, accessibility, transaction (atomicity, consistency, isolation, and durability), capacity, integrity, regulatory (conformance and compliance to the rules, laws, standards and specifications) and reputation.

Since many of these qualities have been defined in this study as a comparison metrics, this point is not viewed in detail.

I. Complexity

From the point of view of complexity, it is easier to develop and implement Web services using REST. REST protocol uses the Web architectural style that is closely related to the HTTP protocol. Thus, the Web service development using REST is more “Web-friendly” [10].

At the same time, SOAP is based on closely specified data structures, API (Application Programming Interface), which is specified in WSDL files [10], and the range of standards, such as WSDL, WSBPEL, etc. [8]. In REST, this all has been achieved by simplifying the API to the HTTP basic operations, such as GET, PUT, POST, etc. [10], [14].

XML Schema is an effective language to define the data model, but sometimes when developing the Web service it is difficult to identify the right construct to express a data model that later will be used in SOAP / WSDL implementations [4].

From the point of view of complexity, REST approach is simpler since it uses widely known W3C / IETF standards (HTTP, XML, URI, and MIME).

In the case of REST, HTTP clients and servers are available for all major programming languages and operating systems, as well as HTTP port 80 is generally open in most firewall configurations.

J. Efficiency

The REST protocol has caching, clustering and load balancing mechanisms [4]. This ensures that the REST Web service can be used by a large number of users at the same time. It should be considered that REST uses JSON (which is a lightweight data format) or even plain text to send messages that in total reduces the server load.

K. Reliability

In the SOAP protocol, the WS-Security is used as one of the standards [3] for signing and encrypting messages so that the data transfer becomes safer. In turn, the REST approach is missing its own security model. Communication model is based on the HTTP (or HTTPS) built-in security mechanisms, and additional security mechanisms can be written manually depending on the project specifics. Transport layer security mechanisms protect the point-to-point communication channels, but if there are mobile devices involved, it becomes problematic because TLS (Transport Layer Security) channel must be frequently reset [9].

L. Maintainability

In [2] SOAP and REST are compared in terms of maintainability perspective. It was concluded that from the maintainability point of view it was easier to maintain RESTful Web services on the server side. In turn, SOAP-WSDL Web services were more maintainable on the client-side. This is

explained by the fact that the SOAP Web services use WSDL file to describe a service interface. In today's integrated development tools, such as Eclipse, a client-side code can be automatically generated based on the available WSDL file.

M. Comparison Summary

Taking into account the previously viewed metrics relating to the SOAP and REST service types, the results are presented in Table V. Summarising the results, it is not possible to clearly identify the best protocol to use to integrate internal information systems and applications with other systems. This means that each project has to be assessed individually by functional and non-functional requirements. Taking into account these requirements, it is possible to identify which of the approaches – SOAP or REST – to use.

TABLE V
COMPARISON SUMMARY

Metric	SOAP	REST
Cost (lower)		✓
Effort (lower)		
Server side		✓
Client side	✓	
Lines of code (fewer)	✓	
Execution speed (faster)		✓
Memory (low consumption)		✓
Errors (less)	✓	
Functionality		
Protocol independent	✓	
Complexity (lower)		✓
Efficiency		
Performance (better)		✓
Reliability	✓	
Maintainability		
Server side		✓
Client side	✓	

V. RECOMMENDATIONS

Having performed the research result analysis, it has been concluded that that it is not possible to clearly identify the best approach to ensure data exchange because each integration project should be assessed individually. Each type – SOAP or REST – has its own advantages and disadvantages. However, it is possible to highlight the main characteristics to ease the choice of the right approach.

A. REST

If the project requires greater scalability, compatibility and performance, it is better to choose REST. REST

implementation complexity, execution speed, consumed memory resources and performance were better compared to SOAP protocol. Thus, if the project requires a simple point-to-point integration or large-scale availability from the mobile devices, REST is the right choice. Based on these reasons, the major Web service providers use exactly REST: Twitter, Yahoo, Flickr, del.icio.us, etc. [7].

B. SOAP

SOAP is a better choice when the project requires security and reliability, easier maintainability on the client side, as well as a lower number of possible errors. In addition, many business system integration projects require asynchronous data processing requests, which are the SOAP advantage. Consequently, the conclusion is that SOAP is more suitable in large information system integration, such as banking information system integration projects.

The choice of SOAP and REST approach, depending on the project aims, is summarised in Table V. It should be noted that this is not a strict rule how projects should choose the right data exchange approach. Table V contains selection criteria based on the literature analysis.

TABLE V
SELECTION OF SOAP AND REST PROTOCOLS

	Main criteria	Project example
REST	Greater scalability; Compatibility; Performance; Simplicity; Point-to-point communication model; Limited bandwidth.	Mobile app integration with the information systems; Simple client-server data exchange solutions.
SOAP	Higher security and reliability; Lower number of errors; Asynchronous requests; Distributed computing; Support from other standards (WSDL, WS).	Business information systems (B2B); Banking information systems; Payment systems.

VI. CONCLUSION

This paper provides a general comparison of SOAP and REST services based on software evaluation metrics. The paper contains a systematic literature review, where a conceptual comparison method is applied to compare the SOAP and REST approaches. Results do not clearly show which of the approaches should be selected to make integration among the systems. To select the right approach, functional and non-functional requirements should be analysed before making the choice about SOAP or REST. For example, if the project is about to integrate two simple information systems, the right choice will be REST. However, if the systems are complex and they should have additional security levels, the right choice will be SOAP, which supports many different standards (for example WS-Security).

The present paper only focuses on SOAP and REST service types. In future, the research focus can be made on other Web service types because it was not possible to find detailed information on some software evaluation metrics, so this is left for future research.

REFERENCES

- [1] S. Kumari and S. K. Rath, "Performance comparison of SOAP and REST based Web Services for Enterprise Application Integration," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, 2015, pp. 1656–1660. <https://doi.org/10.1109/ICACCI.2015.7275851>
- [2] R. R. de Oliveira, R. V. Vieira Sanchez, J. C. Estrella, R. Pontin de Mattos Fortes and V. Brusamolín, "Comparative Evaluation of the Maintainability of RESTful and SOAP-WSDL Web Services," in *2013 IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA)*, Eindhoven, 2013, pp. 40–49. <https://doi.org/10.1109/MESOCA.2013.6632733>
- [3] N. Serrano, J. Hernantes and G. Gallardo, "Service-Oriented Architecture and Legacy Systems," *IEEE Software*, 2014.
- [4] C. Pautasso, O. Zimmermann and F. Leymann, "RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision," in *Proceedings of the 17th international conference on World Wide Web*, Beijing, China, 2008, pp. 805–814. <https://doi.org/10.1145/1367497.1367606>
- [5] F. Belqasmi, J. Singh, S. Y. B. Melhem and R. H. Glitho, "SOAP-Based Web Services vs. RESTful Web Services for Multimedia Conferencing Applications: A Case Study," *IEEE Internet Computing*, vol. 16, issue 4, pp. 54–63, July–Aug. 2012. <https://doi.org/10.1109/MIC.2012.62>
- [6] F. AlShahwan and K. Moessner, "Providing SOAP Web Services and RESTful Web Services from Mobile Hosts," in *2010 5th International Conference on Internet and Web Applications and Services (ICIW)*, Barcelona, 2010, pp. 174–179. <https://doi.org/10.1109/ICIW.2010.33>
- [7] P. A. Castillo, J. L. Bernier, M. G. Arenas, J. J. Merelo, P. Garcias-Sanchez, "SOAP vs REST: Comparing a master-slave GA implementation," in *The 1st International Workshop of Distributed Evolutionary computation in Informal Environments*, 2011.
- [8] P. K. Potti, S. Ahuja, K. Umapathy, Z. Prodanoff, "Comparing Performance of Web Service Interaction Styles: SOAP vs. REST," in *Proceedings of the Conference on Information Systems Applied Research*, New Orleans Louisiana, 2012, ISSN 2167-1508.
- [9] G. Serme, A. S. de Oliveira, J. Massiera and Y. Roudier, "Enabling Message Security for RESTful Services," in *2012 IEEE 19th International Conference on Web Services*, Honolulu, HI, 2012, pp. 114–121. <https://doi.org/10.1109/icws.2012.94>
- [10] T. Aihkialo and T. Paaso, "Latencies of Service Invocation and Processing of the REST and SOAP Web Service Interfaces," in *2012 IEEE 8th World Congress on Services*, Honolulu, HI, 2012, pp. 100–107. <https://doi.org/10.1109/SERVICES.2012.55>
- [11] J. Kangasharju, S. Tarkoma and K. Raatikainen, "Comparing SOAP performance for various encodings, protocols, and connections," in *Personal Wireless Communications* (Lecture Notes in Computer Science (LNCS)). Springer-Verlag, vol. 2775, 2003, pp. 397–406. https://doi.org/10.1007/978-3-540-39867-7_38
- [12] K. Wagh and R. Thool, "A Comparative Study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host," *Journal of Information Engineering and Applications*, vol. 2, no. 5, pp. 12–16, 2012, ISSN 2224-5782 (print), ISSN 2225-0506 (online).
- [13] W. N. Wan Ab Rahman and F. Meziane, "Challenges to Describe QoS Requirements for Web Services Quality Prediction to Support Web Services Interoperability in Electronic Commerce," *Communications of the IBIMA*, vol. 4, issue 6, pp. 55–58, 2008.
- [14] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess and D. Savio, "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services," *IEEE Transactions on Services Computing*, vol. 3, no. 3, pp. 223–235, July–Sept. 2010. <https://doi.org/10.1109/TSC.2010.3>

Juris Tihomirovs is a student at the professional Master programme "IT Project Management" at the Institute of Information Technology of Riga Technical University (Latvia). He also works as a Systems Analyst at ZZ Dats Ltd. E-mail: Juris.Tihomirovs@edu.rtu.lv

Jānis Grabis holds a Doctoral Degree and is a Professor at Riga Technical University (Latvia) and the Head of the Institute of Information Technology. His main research interests lie within the application of mathematical programming methods in information technology, enterprise applications and system integration. He has published more than 80 scientific papers, including a monograph on supply chain configuration. He has led a number of international and national projects and has participated in five projects in collaboration with the University of Michigan-Dearborn (USA) and funded mainly by industrial partners, such as SAP America and Ford Motor Company. E-mail: grabis@rtu.lv