

Cloud-Based Web Services

1st Given Edward Patch

Software Engineer (of BSc Year 3)

Cloud Computing

University of Wales Trinity St. Davids (of Dr. Tim Bashford/Mike Dacey)

Swansea, Wales

Student ID: 1801492

CONTENTS

I	Introduction	1
II	Web Services	1
II-A	Simple Object Access Protocol	1
	II-A1 Advantages and Disadvantages	1
II-B	Representational State Transfer	2
	II-B1 Advantages and Disadvantages	2
II-C	GraphQL	2
III	Conclusion	3
IV	Terminology	3
V	Appendices	3
VI	Reference List	3

Abstract—The fundamentals of web services, how they work and what different web services are out there. This paper covers and compares other protocols, which constructs a web service.

Index Terms—Cloud Computing, Web Services, SOAP, REST, GraphQL, API

I. INTRODUCTION

The following web services, Simple Object Access Protocol (SOAP), Representational State Transfer (REST) and Graph Query Language (GraphQL), are compared and detailed. A debate of each web service mentioned would show the drawbacks and benefits.

II. WEB SERVICES

A web service enables communication, connecting the Client-Side to the Server-Side within a Web Application. Web Service's allow information to be sent from the Client-Side and rendered on the Server-Side. Commonly, web services use a Markup Language such as, but not limited to Extensible Markup Language (XML), refer to 'XML in a Nutshell' Book by Elliotte Rusty Harold [1] to read further or JavaScript Object Notation (JSON), observe the 'Discovering JavaScript Object Notation' Journal by Charles Severance [2] to discover more.

See figure 1 to see a selection of the different architectures of a Web Service.

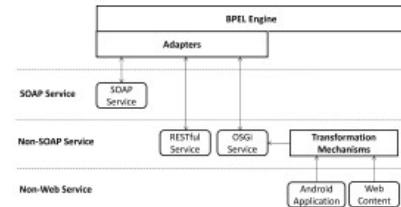


Fig. 1. A Web Service Architecture Design Example [3]

A. Simple Object Access Protocol

XML processes data to encode or decode data. SOAP processes the encoded data, which holds the client's requested information. Citing from 'SOAP Processing Performance and Enhancement' Journal by Joe M. Tekli, Ernesto Damiani, Richard Chbeir and Gabriele Gianini [4]:-

"WS mainly rely on two standard XML schemata

- Web Service Description Language (WSDL) [10] which supports the machine-readable description of a web service's interface. It allows the definition of XML grammar structures for describing WS as collections of communication endpoints capable of exchanging messages.
- Simple Object Access Protocol (SOAP) [82] is the protocol specification for message exchange among WS. It is based on the XML data model, and usually relies on existing application layer protocols (e.g., HTTP, FTP, SMTP, etc.) for message negotiation and transmission."

This quoted paragraph suggests that Web Services (WS) use the SOAP for communication more popularly than other protocols.

1) *Advantages and Disadvantages:* After further examination of the 'SOAP Processing Performance and Enhancement' Journal by Joe M. Tekli, Ernesto Damiani, Richard Chbeir and Gabriele Gianini [4], 'SOAP and Web Services' Article by the author Panagiotis Louridas [5] and 'A Framework for Composing SOAP, Non-SOAP and Web Services' written by Jonathan Lee, Shin-Jie Lee, Ping-Feng Wang [3], the following advantages and disadvantages were found. These are as follows:-

- **Advantage** - Due to SOAP built on XML and Hypertext Transfer Protocol (HTTP), the web service can communi-

cate through different methods and have some lightweight ability to transfer data from server to client.

- **Disadvantage** - SOAP increases network traffic and causes higher latency, thus meaning that SOAP loses performance on the communication side. Users may feel a 'lag', generally indicated by a loading symbol to indicate that the request has been made and is awaiting a response.

B. Representational State Transfer

REST Web Service communicates by endpoints. Endpoints are within Uniform Resource Locator (URL) links, which carry information after the trailing URL. After a request to the Server-Side, the response data is usually in JSON format, parsed on the Client-Side and formatted to how the developer would like.

REST Web Service has a configuration such as Headers and Parameters. If a service like Cross-Origin Resource Sharing (CORS) is available on the Server-Side, it can reject any requests made to the server from a third-party server or Client-Side.

Figure 2 demonstrates the structure of the REST Web Service. However, the figure is a rough diagram of its construction.

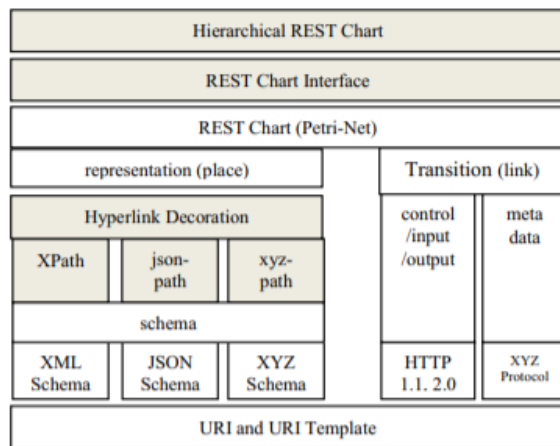


Fig. 2. Structure of REST [6]

1) *Advantages and Disadvantages:* Observations of Books and Journals lead to some advantages and disadvantages. The research follows Performance Analysis of GraphQL and RESTful in SIM LP2M of the Hasanuddin University by Dewi Ayu Hartina, Armin Lawi and Benny Leonard Enrico Panggabean [7], 'Designing Large Scale REST APIs Based on REST Chart' by Li Li and Wu Chou [6] and 'SOAP-Based vs. RESTful Web Services' written by Fatna Belqasmi, Jagdeep Singh, Suhib Younis Bani Malhem and Roch H. Glitho [8]. These are as follows:-

- **Advantage** - REST is popular, easy-to-use and can handle most situations.
- **Advantage** - REST enables the developer to parse the response data from any format. It is not limited to JSON, where SOAP is limited to XML.

- **Disadvantage** - Sending complicated requests to REST compared to SOAP Web Service proves difficult as REST holds data with Endpoints, where an XML document would store request data with SOAP.

C. GraphQL

GraphQL is similar to Structure Query Language (SQL) and other query languages. Facebook, now known as 'Meta', created and used the query language from the year 2012 [7] to improve data collection and performance within Meta's application platforms. According to Mike Bryant [9], 'GraphQL for Archival Metadata: An Overview of the EHRI GraphQL API' aims to '...reduce overhead of data transfer relative to REST-like web service models...', 'reduce the potential for errors caused by invalid queries on the part of the client' and 'support an evolving data model without the need for API versioning.' In turn, GraphQL increases '...efficiency in both power and data usage' [9] GraphQL popularity expanded among Website Developers in year 2017 [7]. To analyse the performance of GraphQL, 'REST vs GraphQL: A Controlled Experiment' by Gleison Brito and Marco Tulio Valente [10] and 'Performance Analysis of GraphQL and RESTful in SIM LP2M of the Hasanuddin University' by Dewi Ayu Hartina, Armin Law and Benny Leonard Enrico Panggabean [7] were looked at to answer the level of performance GraphQL offers. Figure 3 displays a flowchart of how the conduction of the performance test took place. [7].

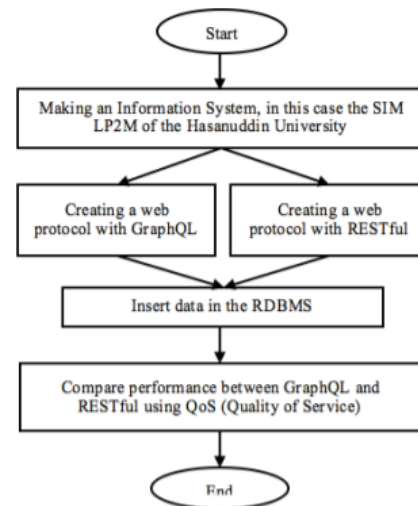


Fig. 3. Research Flowchart [7]

Figure 4 [7] shows that the '...Response Time at GraphQL has an average time in the range of 1810-2130 / ms, while in RESTful it has a faster time with a time range of 891-1000 / ms.' - Dewi Ayu Hartina [7]. Calculation of the response time:-

'Response Time = Execution Time + Data Transfer Time' [7]

Figure 5 uses the following calculation:-

$$\frac{\text{maximal number of request handled}}{\text{time}}, [7]$$

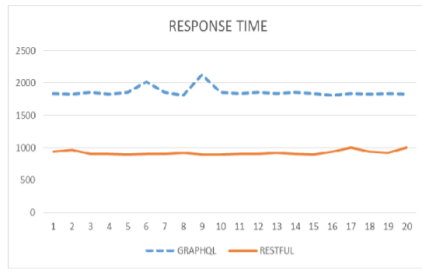


Fig. 4. Response Time Test [7]

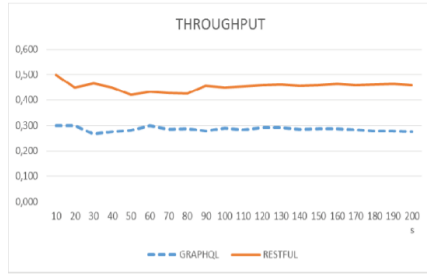


Fig. 5. Throughput Test [7]

The results display that GraphQL can make a request within $0.267-0.300 / s$ [7] and RESTful makes a request in $0.420-0.500/s$ [7]. These results show that RESTful can handle more requests in more seconds than GraphQL can.

To understand how GraphQL and REST compare in development, the Journal, ‘REST vs GraphQL: A Controlled Experiment’ by Gleison Brito and Marco Tulio Valente. The experiment Gleison Brito [10] puts together to test ‘Subjects Experience in REST and GraphQL’ show within table I indicates that from after the test, 50% of the participants preferred REST and 18.2% preferred REST and GraphQL. The selection of Gleison Brito’s [10] subject’s experience experiment demonstrates that RESTful web services are more understanding for the user experience than GraphQL is.

TABLE I
SUBJECTS EXPERIENCE IN REST AND GRAPHQL [10]

REST	GraphQL	REST and GraphQL	None
11 (50%)	0 (0%)	4 (18.2%)	7 (31.8%)

III. CONCLUSION

In conclusion, depending on the third-party server that the development requires data from, the choice of the WS changes. For example, if the server-side requires CORS requirements, a web service, which requires Headers and Parameters for authorisation like REST, is needed.

Unlike the REST web service, SOAP offers a lightweight interface that is more straightforward to implement. Although SOAP saves resources with the development side, SOAP handles optimisation poorly, resulting in a downfall of heavy traffic and higher latency with data transfer.

REST web service allows output data to be customisable by Headers and Parameters to retrieve data in JSON or any other desired format. Due to the nature of the REST web service using URL trailing, querying complex requests can prove difficult compared to the SOAP web service.

After analysing the performance of GraphQL compared with RESTful, the response time of GraphQL seems inconsistent and proves slower with the response. This analysis shows that RESTful can make more responses in the same timeframe than GraphQL.

IV. TERMINOLOGY

List of terminologies used in this document:-

- SOAP - Simple Object Access Protocol.
- REST - Representational State Transfer.
- GraphQL - Graph Query Language.
- XML - Extensible Markup Language.
- JSON - JavaScript Object Notation.
- WS - Web Services.
- HTTP - Hypertext Transfer Protocol.
- URL - Uniform Resource Locator.
- CORS - Cross-Origin Resource Sharing.

V. APPENDICES

[Frontend - Video Link](#)

VI. REFERENCE LIST

- [1] E. R. Harold and W. S. Means, *XML in a nutshell*, 3rd ed., ser. Safari electronic books. Beijing :: O'Reilly, 2004.
- [2] C. Severance, “Discovering JavaScript Object Notation,” *Computer*, vol. 45, no. 4, pp. 6–8, 2012.
- [3] J. Lee, S.-J. Lee, and P.-F. Wang, “A Framework for Composing SOAP, Non-SOAP and Non-Web Services,” *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 240–250, 2015.
- [4] J. M. Tekli, E. Damiani, R. Chbeir, and G. Gianini, “SOAP Processing Performance and Enhancement,” *IEEE Transactions on Services Computing*, vol. 5, no. 3, pp. 387–403, 2012.
- [5] P. Louridas, “SOAP and Web Services,” *IEEE Software*, vol. 23, no. 6, pp. 62–67, 2006.
- [6] L. Li and W. Chou, “Designing Large Scale REST APIs Based on REST Chart,” in *2015 IEEE International Conference on Web Services*, 2015, pp. 631–638.
- [7] D. A. Hartina, A. Lawi, and B. L. E. Panggabean, “Performance Analysis of GraphQL and RESTful in SIM LP2M of the Hasanuddin University,” in *2018 2nd East Indonesia Conference on Computer and Information Technology (EIconCIT)*, Nov. 2018, pp. 237–240.
- [8] F. Belqasmi, J. Singh, S. Y. Bani Melhem, and R. H. Glioth, “SOAP-Based vs. RESTful Web Services: A Case Study for Multimedia Conferencing,” *IEEE Internet Computing*, vol. 16, no. 4, pp. 54–63, 2012.
- [9] M. Bryant, “GraphQL for archival metadata: An overview of the EHRI GraphQL API,” in *2017 IEEE International Conference on Big Data (Big Data)*, Dec. 2017, pp. 2225–2230.
- [10] G. Brito and M. T. Valente, “REST vs GraphQL: A Controlled Experiment,” in *2020 IEEE International Conference on Software Architecture (ICSA)*, Mar. 2020, pp. 81–91.
- [11] S. Karlsson, A. Čaušević, and D. Sundmark, “Automatic Property-based Testing of GraphQL APIs,” in *2021 IEEE/ACM*

International Conference on Automation of Software Test (AST), May 2021, pp. 1–10.

- [12] A. Neumann, N. Laranjeiro, and J. Bernardino, “An Analysis of Public REST Web Service APIs,” *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 957–970, 2021.
- [13] J. Yasmin, Y. Tian, and J. Yang, “A First Look at the Deprecation of RESTful APIs: An Empirical Study,” in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2020, pp. 151–161.
- [14] H. Li, “RESTful Web service frameworks in Java,” in *2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, 2011, pp. 1–4.
- [15] B. M. Balachandar, *RESTful java web services : a pragmatic guide to designing and building RESTful APIs using Java*, third edition. ed. Birmingham, England: Packt Publishing, 2017.
- [16] E. Lee, K. Kwon, and J. Yun, “Performance Measurement of GraphQL API in Home ESS Data Server,” in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 1929–1931.