SOAP vs REST

A translator app in REST and SOAP

Project presentation link: <https://drive.proton.me/urls/HM8EQSJVN4#0YM4zh3XNjSH>

Experiment:

To compare REST API and SOAP API by developing a language translator application using REST API and SOAP API.

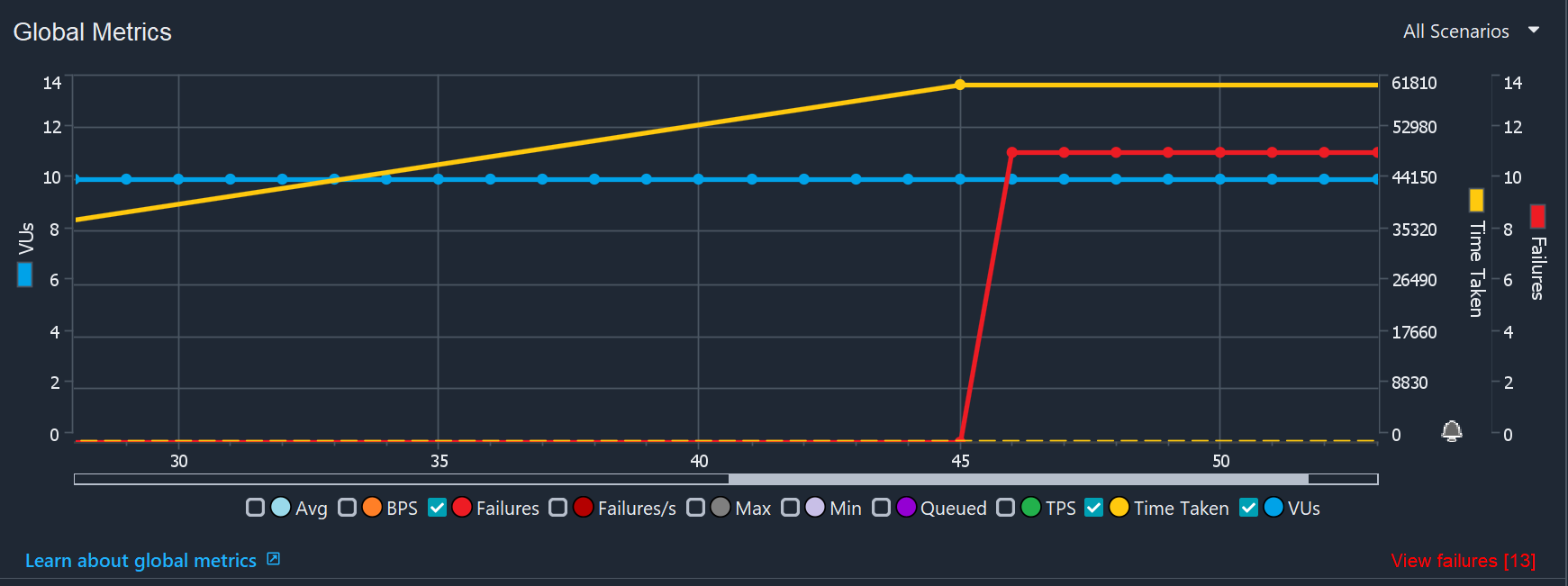
Experiment set up:

So for the set up, we need to develop the language translator application first. I wanted to translate English to French. I developed two different versions of it. One using REST API and another using SOAP API. So after developing it, I used node js to host and run both of them. The translator app worked perfectly in both REST and SOAP API. After this, I wanted to test how good the APIs are by themselves. So I wanted to do a test load on both of them. For doing this, I started using SmartBear READY API (version 3.42.1). From using that, I got the following results for a load test on SOAP API and REST API.

Load test for SOAP



Load test for REST



As we can see, load test for SOAP and REST have a wildly different result. We can see that the failure rate grows linearly for SOAP whereas its very less in REST. This proves that REST can handle much higher loads far effectively that SOAP. For the next test, I wanted to see how much the latency would be for both the APIs. I used POSTMAN to test it. REST had comparably low latency by a huge difference than SOAP. This would again affect scalability of apps built using SOAP and REST as well. These test results show that REST is much more effective and better than SOAP for developing and scaling applications.

**Conclusion:**

Applications that demand the following feature are recommended to be developed using **SOAP API:**

* If the enterprise application is language independent and platform independent
* If the application requires to work in a distributed enterprise environment
* Standard security standards.
* Defined error handling, so that the enterprise application becomes robust.

**REST API offers the following benefits:**

Though there are several pros and cons of using these APIs, SOAP API is highly recommended for enterprise application where it requires security and reliability.

* REST APIs are very closer to web technologies and most of IIOT and mobile applications uses REST APIs.
* Uses easy to understand standards like swagger and OpenAPI Specification 3.0
* REST APIs can be learnt very easily.
* REST uses formats like JSON and hence it is considered very effective.
* Response is very quicker.
* Closer to other Web technologies in design philosophy.

APIs implementation comprises of a set of definitions and protocols. APIs integrate various applications, and facilitate data exchange between applications developed in various technologies. So, fundamentally producer creates a response through an API and when the client calls the API, it sends a response.  Typically, applications call API to communicate, process the data, or perform certain operations. APIs allow the applications to work with various systems to return the processed information. API acts as an interface between users (clients) and resources (servers).

SOAP and REST are ways to design web services APIs. SOAP stands for Simple Object Access Protocol, and the primary objective of designing SOAP during 1990s was to enable the programs that are built on different platforms and programming languages to exchange data in an easy manner.  The REST APIs fully utilize the uniform resource locator (URL) to make data available using the web, maximizing usage of HTTP methods, headers, the body, and the other essential web building blocks. But in addition, REST can be easily adopted within mobile applications.

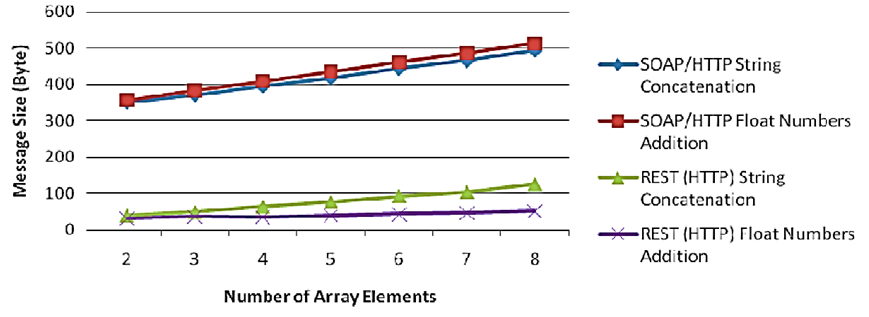
Though delivery of features is the focus of any business, the non-functional requirements directly impact the digital business outcome. For instance, 2 seconds of delay, in web page loading, can result in the loss of 32% of the customers.

Non-functional requirements or the technical requirements must be considered while designing a system and comparison of these 2 technologies is given below:

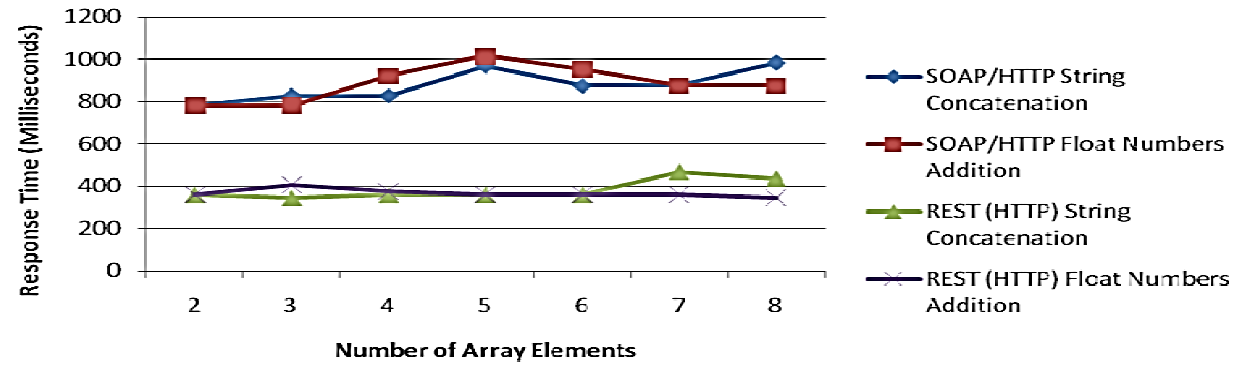
| Non-Functional requirements | Description | SOAP API | REST API |
| --- | --- | --- | --- |
| Usability | How easily the system can be integrated and how the integration can be effectively managed for the changes. | Open Standards: There are fewer open-source industry standards within SOAP universe, with a more proprietary stance on interoperability. | The main benefits of REST APIs are that they rely on the HTTP standard. It is format-agonistic, and we can use XML, JSON, HTML, etc. Because of this standard agonistic nature, it can be used in Internet of Things applications, and mobile devices. |
| Security | Security focuses on the end-to-end life cycle of creation of design to operations, compliance, data security, application access, and auditability | SOAP standard of SOAP 1.2 provides security features with a guaranteed level of security and reliability. | REST is appropriate for publicly available URLs and does not impose any security standards like SOAP. If the data is breached in the network layer, or because of any other layer breach, will not be prevented by REST API. |
| Portability | Portability assesses the degree of flexibility to change, openness, integrate and add workloads together, that can be achieved with open standards | SOAP APIs are Platform-independent  and can run on Linux, Windows and macOS | REST API are ideal for cloud applications due to their statelessness. Because of statelessness, they can be deployed and made scalable in the cloud. |
| Performance | Performance include availability, reliability, response time, throughput, and other requirements of speed and coping with the workloads | It is slow because it uses XML format, whereas the payload for a simple string message is large. | REST is just Http and there is no overhead on the network. |
| Elasticity | Elasticity refers to the ability to cope with an increasing amount of workload, users, and more deployments | SOAP uses XML, which requires more bandwidth to transmit over networks and is less desirable for mobile developers. | Since there are no server-side sessions maintained, scalability is high |

**Research on REST and SOAP API performance:**

Below is the research done by Hamad, Hatem & Saad, Motaz & Abed, Ramzi. (2010). Performance Evaluation of RESTful Web Services for Mobile Devices, published in International Arab Journal of e-Technology. They implemented two benchmarks using two different data types as parameters to the web service: float data type, and string data type.  Total session time and a message size of service call measured are listed below.



**Figure 1: RESTful vs SOAP message sizes of string concatenation and float addition service**



**Figure 2: RESTful vs SOAP Response time of string concatenation and float addition service**

**Citation - Hamad, Hatem & Saad, Motaz & Abed, Ramzi. (2010). Performance Evaluation of RESTful Web Services for Mobile Devices. International Arab Journal of e-Technology. 1.**