1 Design & Implementation

The Java API for XML Web Services (JAX-WS) is an API for creating and consuming web services that follow the SOAP (Simple Object Access Protocol) structure. In this protocol, requests and responses are in XML format. It is a widely accepted and applied protocol, which ensures applications on different operating systems can communicate, while using different technologies and programming languages. [1]
With help of the JAX-WS library, it is fairly straightforward to implement simple web services like a calculator. The calculator API that has been developed, contains four *webmethods* corresponding to a trivial calculator. These are addition, subtraction, multiplication and division. Due to the simplicity of the web service, no additional message handling has been implemented. By adding the *@SchemaValidation* annotation [2], all requests and responses of the web service are validated against the XML schema. This way, no input errors will occur within the defined methods. Additionally, the errors become more discriptive. Without the *SchemaValidation*, the addition of the number 5.0 and the character *a* would simply result in a response of 0.0. Now with this additional validation, the response contains a message that the second parameter is not valid for the type ’float’.
Bonus Additionally, we have also implemented the calculator service in Python. We have used the Spyne library to host the SOAP web-service and Zeep to make the requests. A naive front-end is implemented with Flask, in order to visualise (one of) the possibilities of a web service.

2 Introducing Statefulness

Web services are, in principle, stateless. The Web Services Resource Framework (WSRF) is a framework that explicitly defines a model for accessing stateful resources using web services. To make our Calculator implementation stateful and, for example, allow a user to use the answer of a previous calculation, we create a Calculator WS-Resource. This Calculator WS-Resouce is referenced by a WS-Addressing endpoint and will have a PortType element that declares the Resource Properties Document type that refer to the `first`, `second`, and `answer` real number fields and to the `operation` (e.g. addition) field, the state of the resource, which would be backed by a Java object instance.

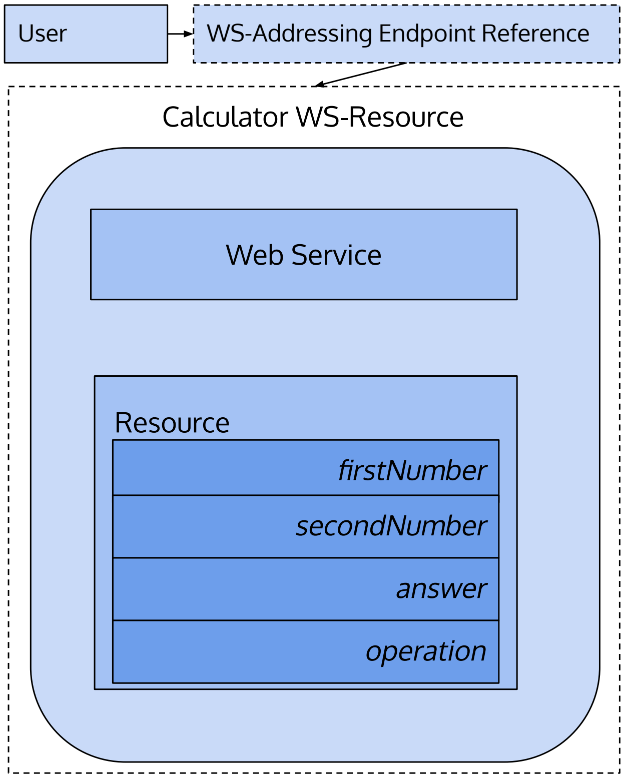

Figure 1: Diagram of proposed Calculator Web Service Resource

References

[1] building web services with jax-ws - the java ee 6 tutorial, Jan 2013.

[2] Getting started with weblogic web services using jax-ws.