Web Services and Cloud Based Systems - Assignment 1

Arumoy Shome
VU ID: 2636393

Shruti Rao
VU ID: 2631481

Group 13

Assignment 1.1

SOAP is a protocol that makes exchange of information over the web easy by standardizing the communication language. This means that services/applications written in any language can interact with one other using SOAP protocol.

Initially, Python was used to design the Calculator web service using SOAP protocol. While a SOAP client *Zeep* is available, Python does not support SOAP on the server side anymore [3]. An attempt was made to use an external library - *flask-spyne* [2]. However, the library was not compatible with the latest Python version (3.7).

The next alternative to Python was *NodeJS*. Using *node-soap*, `server.js` and the executable `calculator` files were written along with the `spec.wsdl` [6]. Users can call the service using `calculator` and specify the functions - `add`, `sub`, `mul`, `div` along with the two numbers they want to perform the function on.

Server

`server.js` creates a new http SOAP server that listens on the port *8080* with path specified at `/spec`. This server consists of the WSDL file that is defined by XML schema. On being accessed by the Client, it provides the SOAP message to the Client. Via the SOAP message, the Client will now know the name of the service - *CalculatorService*. It will also get information to expect the `String` data type and how to call various services in the future.

Client

`calculator` creates a new SOAP client and reads the WSDL from the provided url - http: //localhost: 8080/spec?wsdl. Once the Client is available, it calls the services `Add`, `Sub`, `Mul` and `Div`. Results are finally returned to the Client.

WSDL file

Using a working example as a blueprint, the WSDL file (spec.wsdl) was created manually due to the lack of any web tools for the same [5].

Stateful Design

To achieve Stateful design, the Calculator app would have to be able to save the activities of the client from one session, so that it can be retrieved in future sessions. To do so, the server can send a unique **session identifier** to the client. The client can append this token to all future requests it makes to the server. Based on this ID, the server can retrieve the client data from a persistent data store, recreate the application state and send it back to the client as a response (see Figure 1).

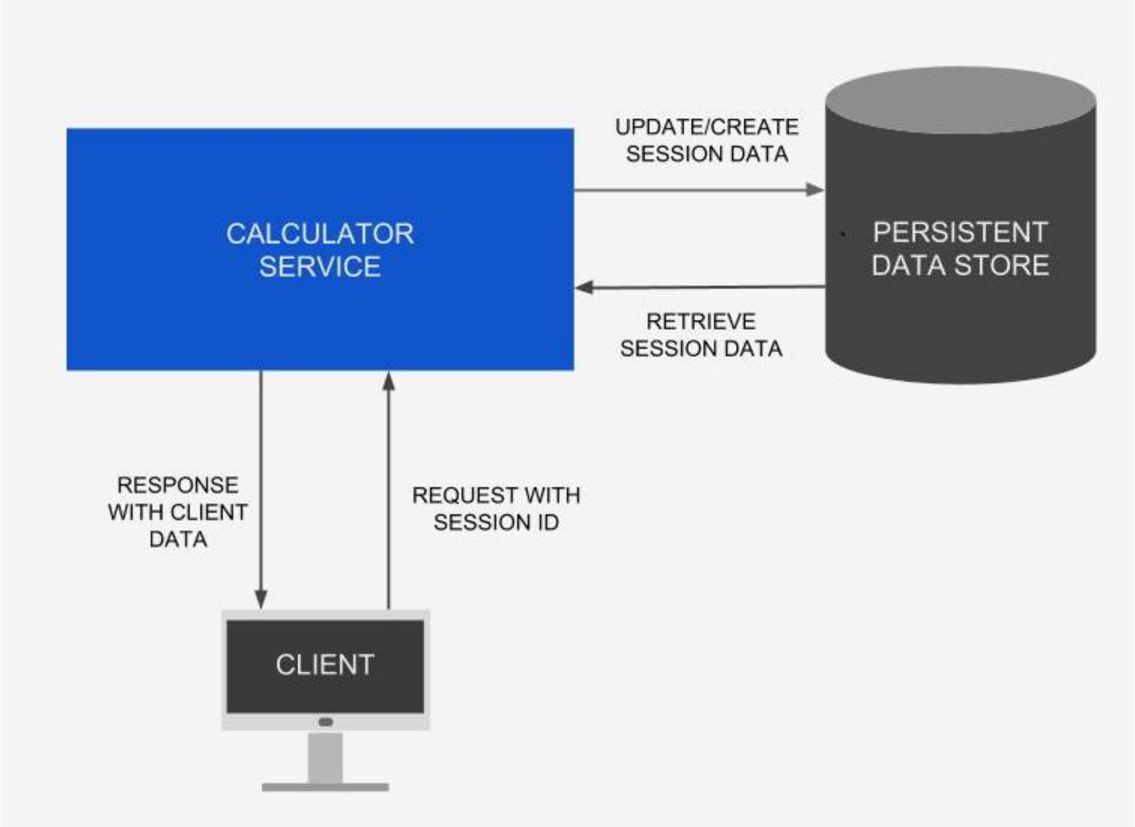

Figure 1: Stateful Calculator Sketch

References

[1] cawcaw et al. *How do I create a URL shortener?* URL: https://stackoverflow.com/questions/742013/how-do-i-create-a-url-shortener.

[2] *Flask-Spyne.* URL: https://pypi.org/project/Flask-Spyne/.

[3] *Python SOAP client.* URL: https://python-zeep.readthedocs.io/en/master/.

[4] *short$_u$rl.* URL: https://pypi.org/project/short_url/.

[5] Tutorialspoint.com. *WSDL Example.* URL: https://www.tutorialspoint.com/wsdl/wsdl_example.htm.

[6] Vpulim. *vpulim/node-soap.* Mar. 2019. URL: https://github.com/vpulim/node-soap.

[7] *Welcome.* URL: http://flask.pocoo.org/.