

# CMPT 431: Distributed Systems

## Assignment 2, Spring 2019

This assignment is to be done in groups of 2 students!

In assignment 1, we used RPC or RMI to communicate between a client and a server. This of course required both the client and the server to be written in the same programming language.

In this assignment, we will repeat the same problem (clock synchronization), but using standard web services. Although the examples and resources below are in Python, you can choose any programming language of your choice.

### **Suggested Resources:**

- [Web Services](#)
- [SOAP](#)
- [REST](#)
- [JSON](#)

### **Q1 (70 marks) Web service using SOAP and WSDL:**

Create a SOAP service using the Web Service Definition Language [WSDL](#) (for creating WSDL, you can use [Eclipse IDE](#) or do it by hand) and deploy it in using [SimpleHTTPServer](#) (a python web framework) or Apache Tomcat server or another web server of your choice. Then, write a client Web application (see example [here](#)) to access the service you have created, in order to perform the same clock synchronization scheme as in Assignment 1. Exchanged data should be formatted in JSON.

Useful References: A sample [WSDL of Google Search SOAP service](#), and sample [SOAP client message to access Google Search SOAP service](#).

### **Q2 (30 marks) Web service using REST:**

[Create a REST-based service](#) to perform the same clock synchronization scheme in Assignment 1. Deploy the web service either using SimpleHTTPServer or Apache Tomcat server or another web server of your choice to host/deploy the RESTful service. Then, develop a client Web application to access the deployed RESTful web service to synchronize its clock. Exchanged data should be formatted in JSON.

Suggestion: you can use the *Flask* web framework to deploy your web service if you are using python.

### **Q3 (bonus 15 marks) Web service using standard REST API- Google calendar API:**

Develop a client Web application using either SimpleHttpServer or Flask or another tool of your choosing to locally host/deploy your web app. The application will access the [Google Calendar API \(API reference\)](#).

**3.1)** Your web app should first get all the upcoming events in you Calendar using the Google Calendar API and list them neatly in a readable manner.

**3.2)** Your app should then create a meeting event for February 6<sup>th</sup> 2019 to be held at Simon Fraser University, Burnaby, Canada with the presenter being Prof. Michael Collins. The response from the server on successful insertion of the event on to the google calendar will be a URL; display that in your app.

**Submission:** In your solution to all 3 questions, describe your design, include screenshots of the output, and attach your source codes. There is no demo for this assignment.

**Note:** If you score more than 100% on this assignment, it will be carried over to the final grade calculation.