**Lab 1-Asterix Stock Bazaar Part 2 Design document**

* **Overview:**

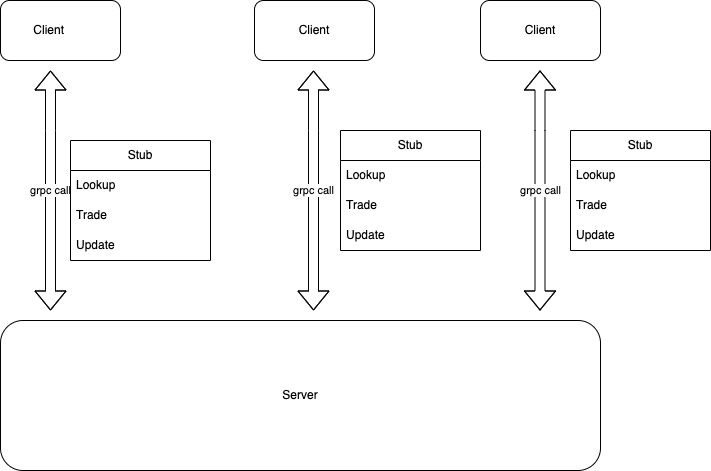
This is a stock bazaar implementation to demonstrate grpc protocol. Protobuf is used to define and create services such as lookup, trade and update along with the message structure required to generate stubs of request and response messages. Client uses grpc stubs to invoke remote object calls from the server.

* **Architectue:**

The model is designed in a single server multiple client manner. A single server instance is running, to which rpc calls are made by multiple client processes. At startup client program requires the hostname and port number of machine on which server process is running. For server program, the port number to listen for calls is need to be specified, along with maximum threads for threadpool and maximum threshold of trading for each stock are required at startup.

* **Inbuilt Threadpool:**

Python’s ***concurrent.futures.ThreadPoolExecutor*** is used to create a threadpool at server side to handle incoming requests. The ***max\_workers v***ariable used to specify the maximum number of concurrent threads in the threadpool is set at startup as per user input. The thread pool executor has an inbuilt queue that stores incoming requests and assigns them to idle workers in threadpool.



* References:
  + Wikipedia : <https://en.wikipedia.org/wiki/GRPC>
  + GRPC Python Documentation : <https://grpc.io/docs/languages/python/basics/>
  + Protobuf Documentation: <https://protobuf.dev/overview/>
  + Threadpoolexecutor : <https://docs.python.org/3/library/concurrent.futures.html>