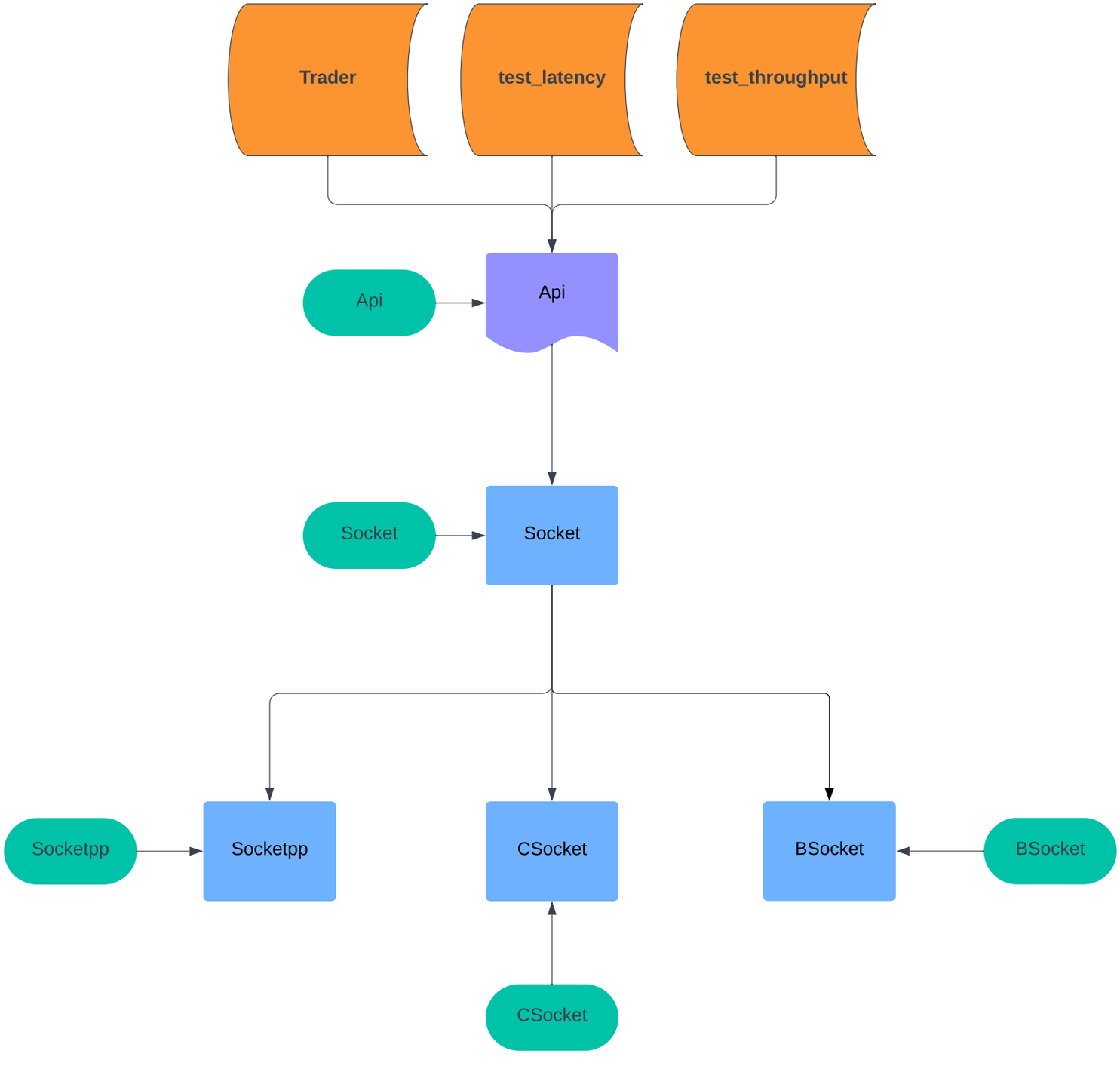
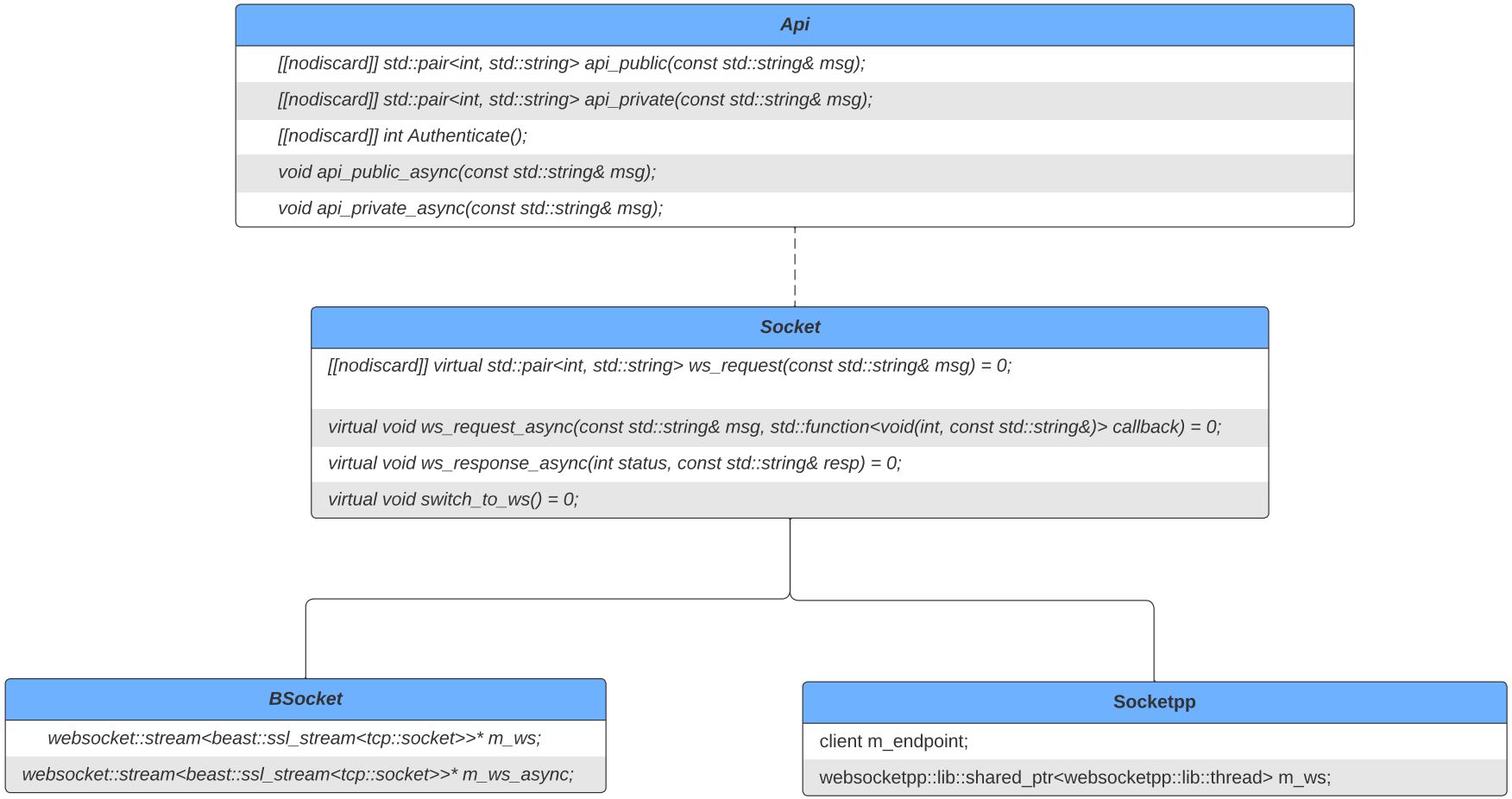

Trader
test_latency
test_throughput
Api
Api
Socket
Socket
Socketpp
Socketpp
CSocket
BSocket
BSocket
CSocket

GoQuant Demo

Lakshay Bansal | December 31, 2024



Trader
std::pair<int, std::string> place_order(const std::string&, double, int);
std::pair<int, std::string> cancel_order(const std::string&);
std::pair<int, std::string> modify_order(const std::string&, double, int);
std::pair<int, std::string> get_orderbook(const std::string&, int);
std::pair<int, std::string> view_position(const std::string&);
std::pair<int, std::string> get_openorders(const std::string&);
std::pair<int, std::string> get_marketdata(const std::string&, int);

helper
int handleCancelOrder(const std::function<std::pair<int, std::string>(std::string)>& action);
int handlePlaceOrder(const std::function<std::pair<int, std::string>(std::string, double, int)>& action);
int handleModifyOrder(const std::function<std::pair<int, std::string> (std::string, double, int)>& action);
int handleGetOrderBook(const std::function<std::pair<int, std::string> (std::string, int)>& action);
int handleViewPosition(const std::function<std::pair<int, std::string> (std::string)>& action);
int handleOpenOrders(const std::function<std::pair<int, std::string> (std::string)>& action);
int handleMarketData(const std::function<std::pair<int, std::string>(std::string, int)>& action);

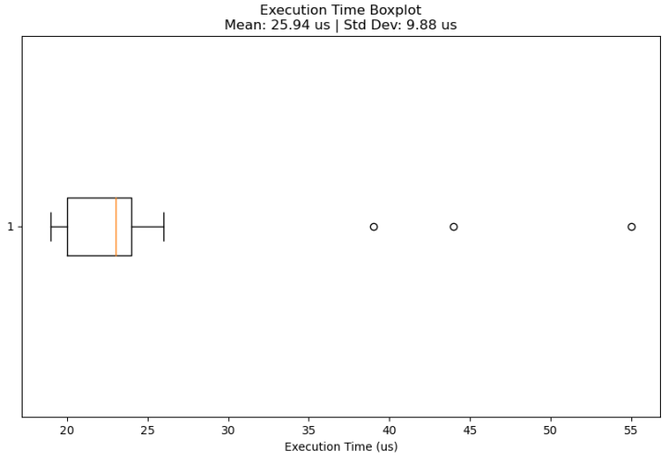
test_latency
void place_order(int &orders, Api& api);
void place_order_async(int& orders, Api& api);
void clear_orders(Api& api);
int main();

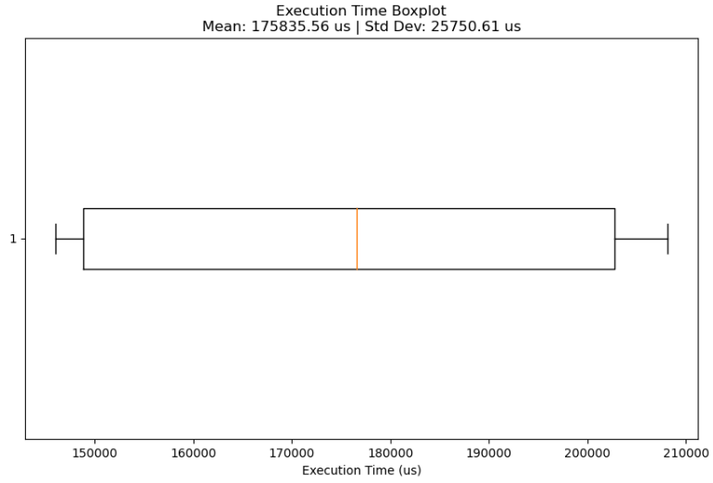
test_throughput
void place_order(int &orders, Api& api);
void place_order_async(int& orders, Api& api);
void clear_orders(Api& api);
int main();

Async

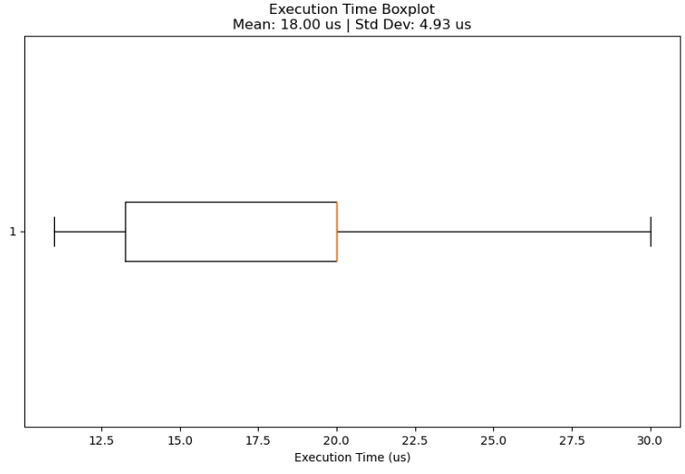
Sync

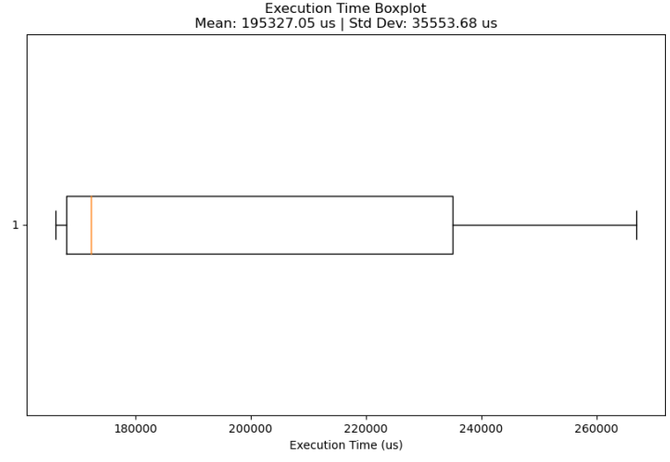
Websocketpp





BSocket





The Sync results are **End-to-end trading loop latency** on estimation propogation delay for request and response were roughly the same

Optimizations

Use of char literals for requests and response (minimal use of nlohmann/json library):
I figured out use of the library for parsing may slow down the application so Just for show casing results to user in case of trader the library is used to pretty print the response.

Passing by reference wherever possible to avoid unnecessary copies made during transfer of objects.

Proper Benchmarking to get a better idea regarding what are the bottle necks.

Future Work:

Continue Developing my own Socket library tailor mad for our needs.

Rectify the bugs or issues in the code.