

Diagrams: Format is a cross between: a class diagram (rectangles show the classes, dashed rectangles are system classes) and a sequence diagram (ellipses show methods corresponding to an object, in a lane indicating the level of operation)

- 1) “REST Design.pdf” (link [here](#)) shows the use of exchange RESTful APIs to poll for data every minute, then aggregate it to compare prices and volumes, then trade various strategies if there are opportunities.
- 2) “Streaming Design.pdf” (link [here](#)) shows the use of a websocket to gain every transaction in Bitstamp markets.

Overview: Client wanted to be able to trade Bitcoin across Cryptocurrency exchanges, both in the West and Emerging markets. Client offered the work to me and another contact, he only intended to proceed with one of us.

Objective: Develop software to be able to monitor for and trade Bitcoin Arbitrage opportunities across exchanges

Results: Gained go-ahead to work exclusively with client. Surpassed expectations;

The platform (written in Java and JavaScript) can:

- identify and automatically trade across 6 different exchanges
- trade both cross-sectional (across exchanges in a given time period) and time series (across time periods for a given exchange) strategies
- trade Bitcoin and other cryptocurrencies e.g. Ether
- account for market depths and collect both Limit Order Book and Tick information

The platform can be extended relatively easily to incorporate further exchanges and cryptocurrencies due to the time spent on OOD

I have gained interest from a hedge fund for developing the platform further

Steps: Created a Minimum Viable Product to demonstrate capability to client

Designed and Implemented extensions to enable full functionality. Created 90% of the system in Java. Outsourced 10% for JavaScript calls to 1) Coinsecure and Currency Layer RESTful APIs and 2) Bitstamp websocket