Final Project Proposal

1. General Idea

This project is about simulating the whole process starting from distributed servers' collecting real-world stock price data, to consultants' providing buy/sell/hold advices applying different trading strategies, to clients' deciding trading actions based on their own risk preference, finally to master server's passing or rejecting the requested actions.

2. Detailed Description

This project involves multiple phases.

First, there will be at least five servers acting as Producers, which collect real-world stock price data simultaneously, but from different kinds of data sources. One server will fetch stock price data from the Internet using third-party Java library API's and the data format will be Java objects. Another four servers will read in data from local csv files, json files, xml files, and database via JDBC.

Then, every time when a server gets one piece of stock price data, it will send this message through its point-to-point channel (Queue) to the Canonical Data Model acting as the Consumer, where the data formats will be unified, and then the stock price data will be Published and multiple consultants acting as Subscribers will get this piece of data and generate buy/sell/hold advices applying different strategies.

Then, these consultants will publish the advices about different stocks to corresponding Topics, and the clients who are interested in some of these topics (i.e. stocks) will receive the advices and decide their trading actions based on their own risk preference.

Finally, these requested actions will be sent through point-to-point channels to the master server, which keeps the records of these clients including their cash flow and owned stocks quantities etc. The master server will make the decision to either pass or reject the requested trading actions.

3. Related EIP Patterns

1) Point-to-Point Channel: used to communicate the collected stock price data and clients' trading actions.

2) Publish-Subscribe Channel: used to distribute and deliver the unified stock price data to consultants, and consultants' buy/sell/hold advices to clients.

3) Message Broker/Message/Message Endpoint/Message Channel: used to collect raw stock price data and make final trading decisions etc.

4) Other EIP Patterns such as Content-Based Router, Invalid Message Channel, Dead Letter Channel etc. will also be needed when controlling the data flow during the whole process.

4. Related Design Patterns

1) Singleton Pattern: used to make sure that there is only one master server in charge.

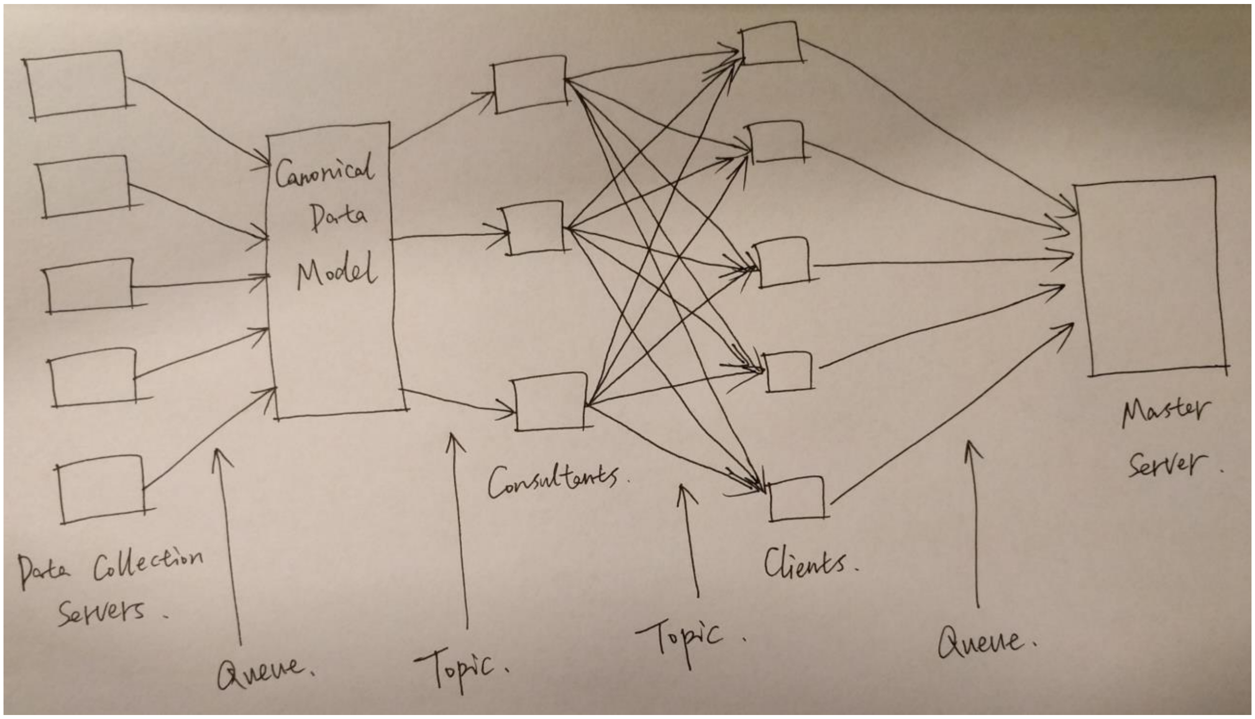
2) Strategy Pattern: used by consultants to make it easy to apply different trading strategies and generating corresponding buy/sell/hold advices.

3) Composite Pattern: used by clients to store their portfolio of the different stocks concerned.

4) Template Method Pattern: used by the master server to check if the client has enough cash or owned stock quantities, and either pass or reject the requested trading actions, and calculating the cash and stock quantities changes if applicable.

5) State Pattern: used by clients so that they behave differently under different states of risk preference.

5. Concept Graph


Canonical Data Model
Data Collection Servers.
Queue.
Topic.
Consultants.
Topic.
Clients.
Queue.
Master Server.