General Problem

The project is to implement a statistics generator for different trading engines located in different places across the world, such as New York, London and Tokyo. They may be interested in different stocks and different statistics and the goal of this project is to deliver the desired statistics information to the interested trading engine. This whole process of statistics information delivery involves routing, filtering, calculation and some other extra processing from the raw data. Also, messaging queues also needed in such process. The following sections will discuss in details the different steps in this process and potential design patterns and enterprise integration patterns (EIP's) during implementing such process.

Producer

The producer is to produce valid bid and ask data to calculate stock information statistics. The original data feeds are csv files, each of which contains a bid price, a bid quantity, an ask price, an ask quantity of a certain stock. The producer will do simple work to check each csv files for its data validity. The output from producer are fed to two message queues, one is for valid data, and the other for invalid data, which could be missing data, or errors. The message queue containing valid data will be further processed while the message queue containing invalid data are not handled in this project. In reality, they will also require some further processing, such as re-validation.

Inbound Endpoint: csv files
Outbound Endpoint: tow message queues
EIP Patterns: Splitter, Message Translator, Point to Point Channel
OO Design Patterns:

Consumer

The Consumer is to categorized data into different topics for each stock. It takes the input data from the message queue containing valid data produced by upstream Producer. It examines each data, does the statistics calculation, creates a message containing those statistics information and put it on a topic for each stock. So the consumer will create one topic for each stock, resulting in several topics. Different trading engines can subscribe to different topics they are interested in and get updates when there are new messages on those topics.

Inbound Endpoint: message queue
Outbound Endpoint: multiple message topics
EIP Patterns: Content Based Router, Message Translator, Log, Content Enricher, Publish-Subscribe Channel
OO Design Patterns: Strategy, Template Method

Subscriber

The subscribers are the trading engines. They subscribe to the topics they are interested in. Each topic contains statistics info for a given stock. Potentially they could further filter to only get the desired statistics information about the stock. Each trading engine will have a reporting engine that will periodically printout statistics data for its portfolio.

Inbound Endpoint: multiple message topics
Outbound Endpoint: NA
EIP Patterns: Log, Message Filter
OO Design Patterns: Singleton, Iterator, Observer, Composite

Summary

This project implements a statistics generator for different trading engines to fetch interested stock statistics information. This project consists a producer to produce stock data, a consumer to dispatch each stock statistics information to different topics, and each subscriber to subscribe the their interested stocks. This project leverages different object oriented design patterns and EIP's to build a simple yet functional messaging system.