Data Factory
DRAGONFLY
Mine, Manage and Monetize Big Data

Dragonfly Data Factory

Stock Broking
Use-Case

Table of Contents

Stock Broking Limited- brief overview

Stock Broking Limited offers the largest online stock-trading platform, for buying and selling stocks and shares. It offers online trading for both key platforms – National Stock Exchange and Bombay Stock Exchange. The organization has created a very robust trading platform that facilitates customers to trade online not only in equities, but also to buy fixed deposits, mutual funds, commodities, currencies and participate in a public issue.

Problem Statement

Stock Broking is a retail business, with 800,000+ customers. They have a massive customer database and they maintain multiple database repositories for all this customer data. Integrating these multi-source data and to analyze the data on a real time for business insights is the major focus of the organization.

The organization is actively looking for a solution which would help them address the following:

- Real-time transaction data analysis
- Integration of multi-source data
- Data contextualization
- Real-time analytics
- Real-time and static dashboards to monitor the transactions and business

Database Repositories

The database repositories store different data related to trades, stocks, customers and holdings. The team has shared us the datasets from each repository, below is the screenshot of data descriptions.

Trade Details (Real-Time Transactions)

Column Name	*Column Description*	*Mapping Column*
GOTXRefNumber	System Referenace Number	
InvestorCode	Client Code	ClientMaster.ClientCode
DealerInvestorCode	Dealer ID	
TradeNumber	Trade Number	
ExchangeID	Exchange	
OrderNumber	Order Number	
Symbol	Symbol	ScriptMaster.Symbol
DerivativeType	To identity Equity/Fno orders	
BUYSELL	Buy or Sell Trade	
Price	Trading Price	
Qty	Trading Quantity	
TotalTradedQty	Total Traded Qty	
TradedTime	Traded Time	
TradeDate	Trade Date	
Brokerages	Brokerage earned for the trade	
OrderSource	Source from where trade is placed	

Holding Details

Column Name	*Column Description*	*Mapping Column*
ClientCode	Client Code	ClientMaster.ClientCode
BSECode	Bse Code	
Symbol	Symbol	
Series	Series	
ISIN	ISIN	ScriptMaster.ISIN
Quantity	No of Shares	
BoughtValue	Total Share Value	
BoughtPrice	Share Bought Price	
USQuantity	Under Settlement Quantity	
USValue	Under Settlement Value	
USBoughtPrice	Under Settlement Bought Price	
DPQty	DP Quantity	
DPPrice	DP Price	
DPValue	DP Value	
TotalQuantity	Total quantity	
TotalValue	Total Value	
TotalBoughtPrice	Total Bought Price	
ClosingPrice	Closing Price	
NetProfitLoss	net Profit or Loss	
LockIN	Lock in Shares	
Pledge	Shares in Pledge	

Client Master

Column Name	*Column Description*
ClientCode	To Identity Unique Client
ZoneName	Clients Zone
Area	Clients Area
Cluster	Clients Cluster
City	Clients City
DOB	Clients DOB
Occupation	Clients Occupation
NetWorth	Clients Networth

Script Master

Column Name	*Column Description*
SCRIPTMASTERID	Unique ID
EXCHANGE	Exchange Name
SYMBOL	SYMBOL Name
SERIES	SERIES
ISIN	ISIN
CompanyName	CompanyName

Trade Details: Real-time trading details of the customers, comprises of customer code, stock information, number of units purchased/sold, brokerage, traded date & time etc. The data is generated from the trading platform, where customers do the real-time trade. This data hits the database at every 2-3 seconds, when stock market is open. Around 1000-5000 trade orders are generated in less than 10 seconds, and these number increases during the opening and closing time of the stock market.

Client Master: The customer profile and demographic details like – date of birth, occupation, net worth, city, zone etc. This is a static database and gets appended, when new customer data gets into the system.

Script Master: The stock details like – the organization name, exchange, series, industry etc. This is a static database and gets updated periodically.

Holding Details: The entire trading history. It includes all the vital information for all the stocks which customers are holding.

Dragonfly Offering

About DataSwarm™

DataSwarm™ is a powerful and scalable real-time analytics solution designed to provide rapid application development through a rich graphical user interface. DataSwarm™ abstracts the complexity of programming real-time applications with its high degree of automation using a powerful drag and drop GUI. It enables the organizations to shift their focus from software development to data science and analysis, accelerating the time to market in the analytics value chain.

In an application topology, built on DataSwarm platform, **Components** are the fundamental building blocks of the platform.

Components could be of the following type:

- Sources
- Processors &
- Sinks

Sources read data from raw data sources like Streaming API's, Apache Kafka queue, Kestrel queue, MQTT and so on.

Processors are the logical processing units. Processors receives input stream from sources, process and emits the output stream to another Processors or sinks. Processors can perform the operations of filtering, aggregation, joining, interacting with data sources and databases.

Sinks are the output data sources where the processed data will be written to. A processor processes the stream and writes the output data to any of the following sinks

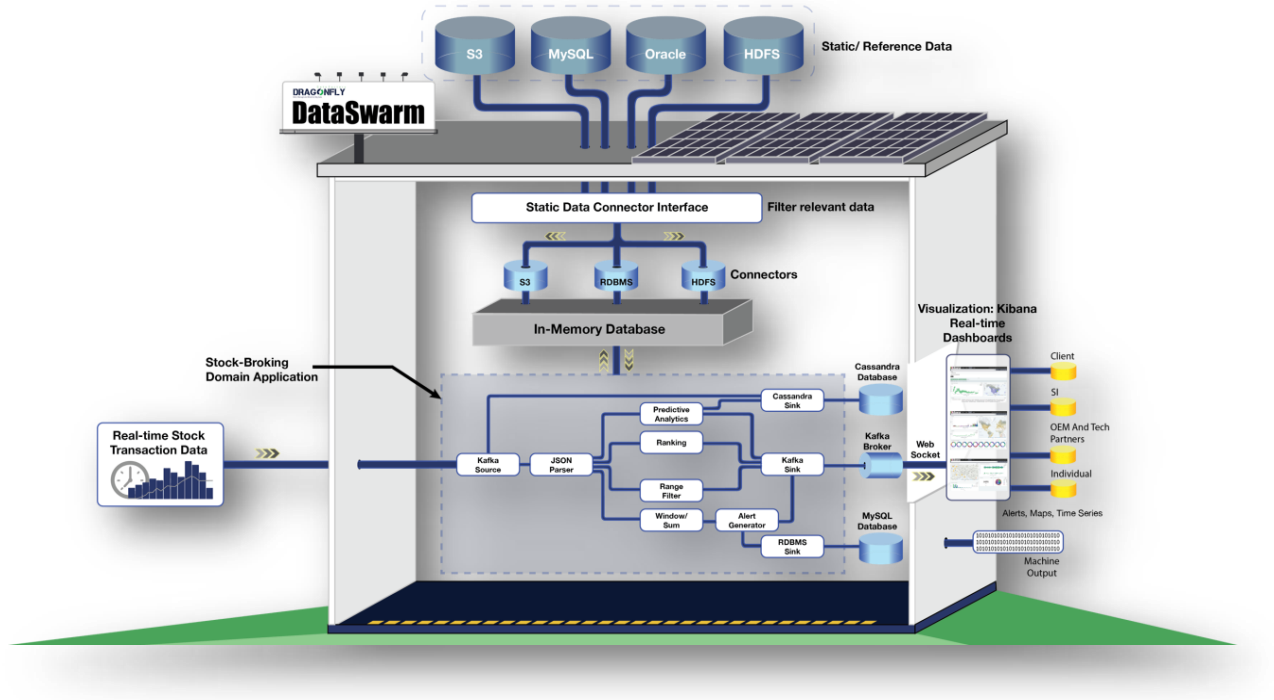
- Message Queues like Kafka, MQTT
- NoSQL databases like MongoDB, Cassandra and so on

Sources, Processors and Sinks are connected together to form an Application or Application Topology. Real-time application logic is specified inside the DataSwarm™ topology. In simple words, a topology is a directed graph where vertices are computations and edges are stream of data.

A simple Application starts with Sources which emits the data to one or more Processors. Processors represent a node in the topology having the smallest processing logic and the output of a Processor can be emitted into another Processor as input or can be written to any of the sinks.

For referencing and contextualizing the data, Users can load the static/reference data, from any data source to the In-memory databases like Hazelcast using **Connectors**.

DataSwarm™ Solution Architecture

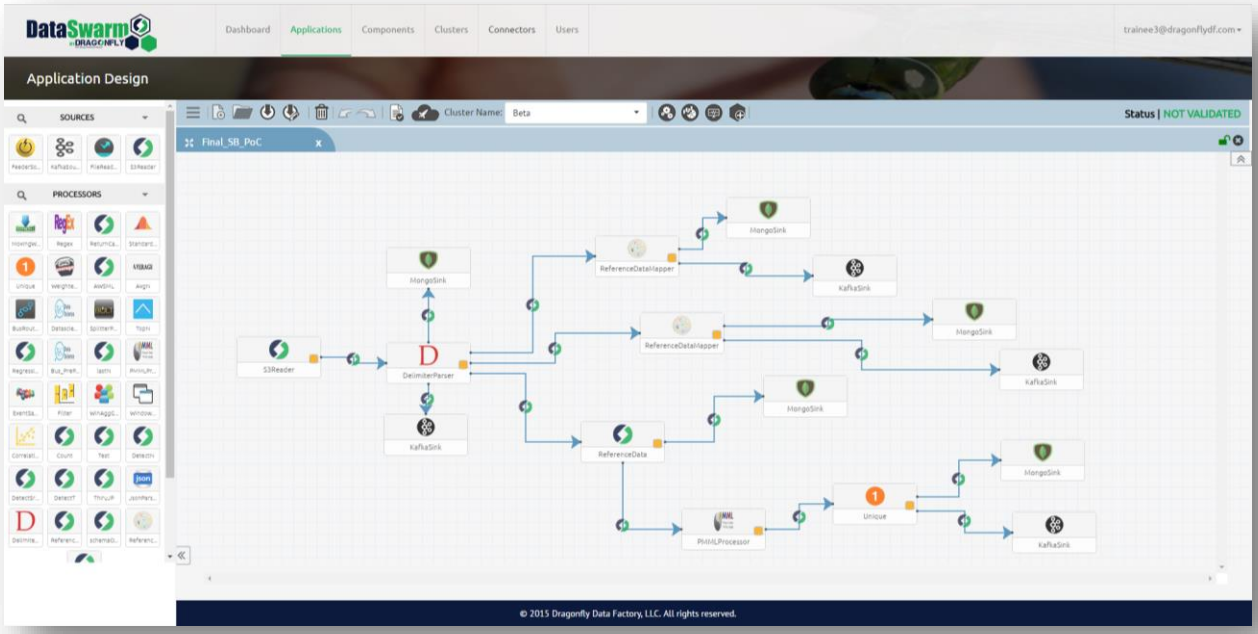


DataSwarm processes the real-time transactions data, and loads the static/reference data onto the In-Memory Database (IMDB), from various relational databases like – S3, MySQL etc., and can be scheduled using connectors.

Using built-in processor component, DataSwarm can do the contextualization, and integrate different static data in IMDB with the real-time stream, using primary key concept.

Processing, analysis and predictions can be performed on the data, using built-in components and machine learning models. The output and results of the application, can be stored in different databases, for different purposes, either for visualization (static/real-time), or for triggering and sending alerts to machines/sensors.

DataSwarm™ Topology Design



A simple application is developed on DataSwarm, to address the stock-broking problem statement, and also to perform some real-time analytics on the transaction data for smart decisions.

For the prototype, real-time transaction events are passed into the system from S3 using **S3 Reader** component as a source, and after parsing the data, the event is moved to another built in component - **Reference Data Mapper**, which lookup and integrates the real-time event with the static/reference data stored in IMDB. After all the processing and analysis, the data is stored in different sinks, either in **Mongo DB** - for future reference and static dashboards, or in **Kafka** - for real-time dashboards.

Real-time Analytics

The data science component - **PMML Processor**, is used to include machine learning models in the application in the Predictive Model Markup Language (PMML) format, for instant predictions.

For a given use case, prediction model - to predict the probability of a contact performing the intraday trading on a given day, based on his/her past trading behavior, is included in the topology.

Visualization

Real-Time Dashboard

Processed output data is passed to Kafka, for generating real-time dashboard using open source visualization tool – **KIBANA.**

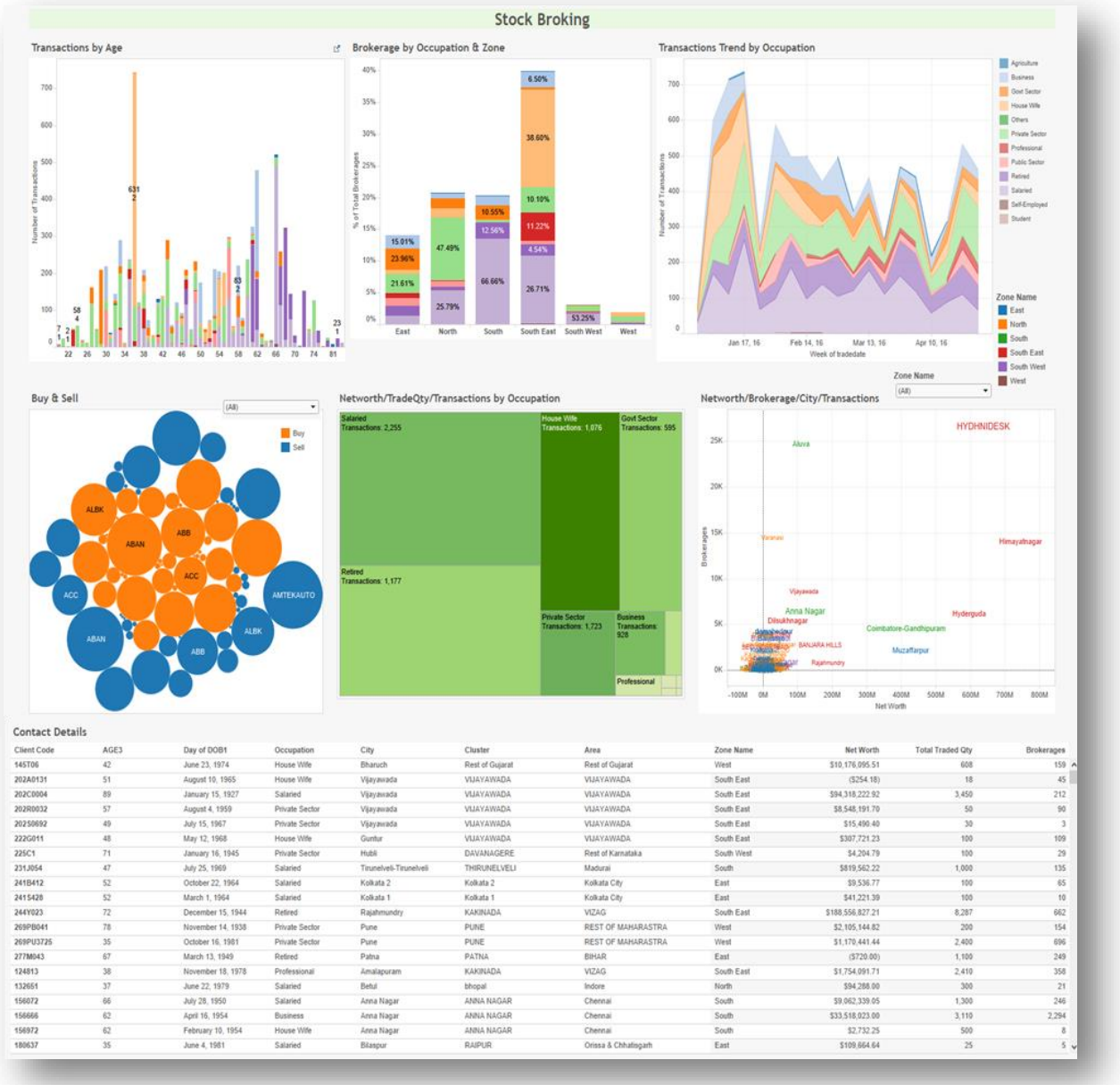
This dashboard shows a visualization being automatically updated with the latest data. Basic descriptive graphs were built and included in the dashboard, to monitor aspects like- customer segments and locations which bring in more or less transactions/revenue, real-time change in the data traffic etc. The dashboard also highlights the top traders and top traded stocks, with respect to time, and depicts the probability of the customer performing the intraday trading.

The real-time dashboard enables the organization to take immediate data driven decisions, based on the trend and the customer behavior, and also enables them to monitor the Key Performance Indicators (KPI) of the business on a real-time basis.

Static Dashboard

MongoDB sink can be connected to any static visualization tool, to visualize the historical data.

This interactive dashboard is built on **Tableau**, which would allow the user to view different KPIs, on the historical data stored in MongoDB. With respect to days, months, or years, the dedicated team within the organization will be able to observe the changes in customer behavior and business, and will also be able to identify profitable customer segment and focus areas. Static dashboard will help the team in taking strategic long term decisions efficiently.



Conclusion

With the given stock broking use-case, the organization will be able to take instantaneous business decisions related to but not limited to **revenue generation, customer satisfaction, customer retention and acquisition** and **resource optimization,** by using above dashboards and application.

To provide efficient customer service, customers with high and low probability of intraday trading can be treated and served differently, which would indirectly lead to an increase in revenue generation and customer satisfaction. With the help of the real-time dashboards, the dedicated teams within the organization, will be able to optimize the resource allocation based on the change in number of transactions or orders. The observations from the trend of the traders, will enable the teams to target the potential customer segments/focus areas, and distribute their marketing activities efficiently.

DataSwarm™ empowers organizations, to solve their complex business problems by taking data driven decisions- both short-term and long-term. The ease to use the product helps the user to address complicated data issues efficiently and effectively. With just the basic knowledge of the product, anyone can easily build and deploy the application with very minimal support.