Architectural Requirements Document:

Algorithmic Trading System

written by:

Stuart Gordon Reid

Systems Architect

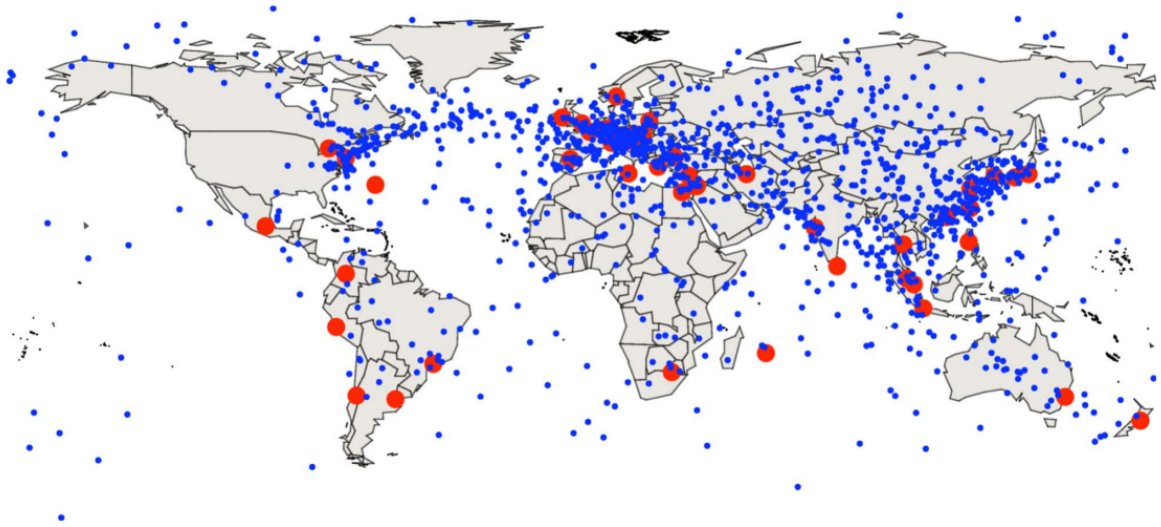
U1006942

for:

Open Source Algorithmic Trading Architectures (OSATA)

(29 September 2013)

V2.0



The blue points in the image above show the places where the network latency for algorithmic trading systems are minimized, and the red points show the locations of large international securities exchanges.

MIT open press: http://dspace.mit.edu/handle/1721.1/62859

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

Revision History

Date	Version	Description
10/09/2013	v0.1	Setup document with headings and began identifying focus areas
16/09/2013	v0.2	Finalized functional design use-cases for ATs
18/09/2013	v0.3	Defined the architectural scope of the ATs
20/09/2013	v0.4	Finalized the quality requirements for the ATs
22/09/2013	v1.0	Finalized first semi-final version of the document
23/09/2013	v1.1	Amended the quality, integration, and access requirements
28/09/2013	v1.2	Added compliance architectural constraints
29/09/2013	v2.0	Produced final version of the document

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

Table of contents

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

1. Introduction and Document Description

1.1. Purpose of this document

The purpose of this document is to describe, at a high-level of abstraction, the architectural requirements which will inform the specification of a systems architecture for a 'buy-side' algorithmic trading system. With respect to the algorithmic trading systems, this document will:

- ➔ Contextualize the system,
- ➔ Define the architectural scope,
- ➔ Discuss relevant quality requirements,
- ➔ Discuss access and integration requirements, and
- ➔ State the architectural constraints.

This document represents parts of an agreement between the Open Source Algorithmic Trading Architectures group, (OSATA), and Stuart Reid, the systems architect. As such, this document serves as a formal contract between the two aforementioned parties. The primary deliverable of this contract with the OSATA architecture.

1.2. Document conventions

This document follows the template prescribed by Dr Fritz Solms of the University of Pretoria
This diagrams in this document are created using the open source diagramming tool, Dia.

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

2. System Context

2.1. Background Information

In this document distinction will be made between programmed trading and algorithmic trading. Programmed trading is the activity of breaking up large markets orders into packets of smaller shares in order to disguise the presence of large orders that could distort the market. Algorithmic trading has more than one definition, in this document it is defines as use of computational algorithms to make trading decisions, submit orders, and manage those orders after submission. There are five types of entities which trade on financial markets. These are:

1. **Retail investors -** individuals who trade on financial markets using their own accounts. The majority of retail investors operate accounts with online securities brokers such as E*Trade, TradeKing, TD Ameritrade, and others.
2. **Proprietary traders -** institutions which trade on financial markets using the funds of the institution. Popular proprietary trading vehicles include hedge funds, mutual funds, and pension funds. Furthermore, proprietary traders which manage relatively large amounts of capital (e.g. Fidelity or Citadel) are called institutional traders.
3. **Market makers -** institutions which provide risk-neural buy and sell side liquidity to the market. Such institutions earn commissions on the 'bounce' between bid and ask prices.
4. **Buy side institutions -** all institutions who are principally concerned with the buying of securities and financial services. These are most often proprietary traders.
5. **Sell side institutions -** all institutions who are principally concerns with the selling of securities and financial services to buy side institutions. Such services can include brokerage services, underwriting services, research services, and in the case of algorithmic trading, even information technology services.

Most algorithmic trading systems are used by proprietary trading institutions, however, recent developments have made algorithmic trading more accessible to retail investors[1]. This document is strictly concerned with a 'buy-side' algorithmic trading system.

[1] In the first half of 2013, a number of brokerages and trading institutions announced plans to make virtual algorithmic trading systems available to retail investors as a service. Such services are exposed using a traditional buy-side algorithmic trading system augmented with a web-based front end and an increased focus on usability and accessibility quality requirements.

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

2.2. Purpose of algorithmic trading systems

Algorithmic trading systems have three primary purposes. First and foremost is the making of trading decisions, secondly is the submission of those orders to market exchanges, and thirdly is the management of those orders post submission.

2.2.1. Purpose #1 - Make trading decisions

Trading decisions are made according to a particular trading strategy. Trading strategies are generally based on one or more of the following techniques:

- ➔ **Technical analysis -** constructed using technical indicators based on historical trends in price, and volume data.
- ➔ **Fundamental analysis -** constructed using fundamental indicators based on the financial state of asset underlying the security.
- ➔ **Quantitative finance -** constructed from a mathematical basis of price movements and historical data and based on the concept of expected returns.
- ➔ **Event based -** based on analysis of historic and current events often profiles from textual news data and translated.
- ➔ **Sentiment based -** constructed using indicators that capture 'market sentiment'.

Such strategies can be realized in a multitude of ways. The types of trading strategies available and the way in which they are realizable are responsibilities of the algorithmic trading system architecture (see section 3) for more details. Some algorithmic trading systems support machine defined strategies such as those strategies derived using machine learning algorithms.

2.2.2. Purpose #2 - Submit orders to market exchanges

Trade orders are requests to buy or sell securities. These get sent to exchanges or dark pools to be matched with other matching market orders. This matching done by complex order matching algorithms. The purpose of the algorithmic trading system is to submit appropriate trading orders that are accurate, timely, and transparent. There are six types of trading orders[2]:

1. **Long orders -** buy a security
2. **Short orders -** sell a security short
3. **Market orders -** buy or sell at market price

[2] For more information about each order type please read the Investopedia article on order types: http://www.investopedia.com/university/intro-to-order-types/introduction.asp

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

4. **Limit orders -** buy or sell at specified price or better
5. **Stop orders -** buy or sell only once a specific price is reached
6. **Conditional orders -** advanced orders to buy, sell, or cancel the order based on some specified condition occurring

The selection of order type, formatting and creation of the order, and the posting of that order are responsibilities of the algorithmic trading system architecture (see section 3) for more details.

2.2.3. Purpose #3 - Manage orders after submission

The third purpose of an algorithmic trading system is to manage the orders after they are submitted. This includes recording all profits and losses made, logging all transactions, and the monitoring of outstanding or unfulfilled orders. These will be described in more detail in section 3.

2.3. Variants of algorithmic trading systems

Since its inception algorithmic trading has produced numerous variants. The extent to which the selection of an algorithmic trading strategy affects the architecture is localized those strategies which result in a change to one or more quality requirements of the architecture. In this document distinction is made then between algorithmic trading systems and high frequency trading systems (HFTs). HFTs are algorithmic trading systems which fulfill their purpose(s) at a high frequency. That is to say they make trading decisions, submit orders to market exchanges, and manage those orders after all at high frequency. The system being architected for OSATA needs to be scalable up to HFT speeds (see section 4 for more information).

2.4. Users of the OSATA architecture

The users of a buy side algorithmic trading system are buy-side trading institutions wishing to execute their trading strategies automatically on market exchanges.

2.5. Stakeholders of the OSATA architecture

Stakeholders of the algorithmic trading system described in this document, and subsequent documents are parties with an interest in the system. Some direct stakeholders include, but are probably not limited to:

- ➔ Stuart Gordon Reid as the architect
- ➔ The Open Source Algorithmic Trading Architecture group (OSATA)
- ➔ System implementers who realize the OSATA architecture

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

- ➔ Buy side institutions which build frameworks based on the OSATA architecture
- ➔ Auditors of systems built on the OSATA architecture
- ➔ Exchanges interfaced to by systems built on the OSATA architecture
- ➔ Regulators of algorithmic trading systems

Please note that the above list only described 'direct stakeholders', in other words, those stakeholders that could directly have an interest in the OSATA architecture. An undefined number of stakeholders may have indirect interests in the OSATA architecture including investors, and IT technicians responsible for support.

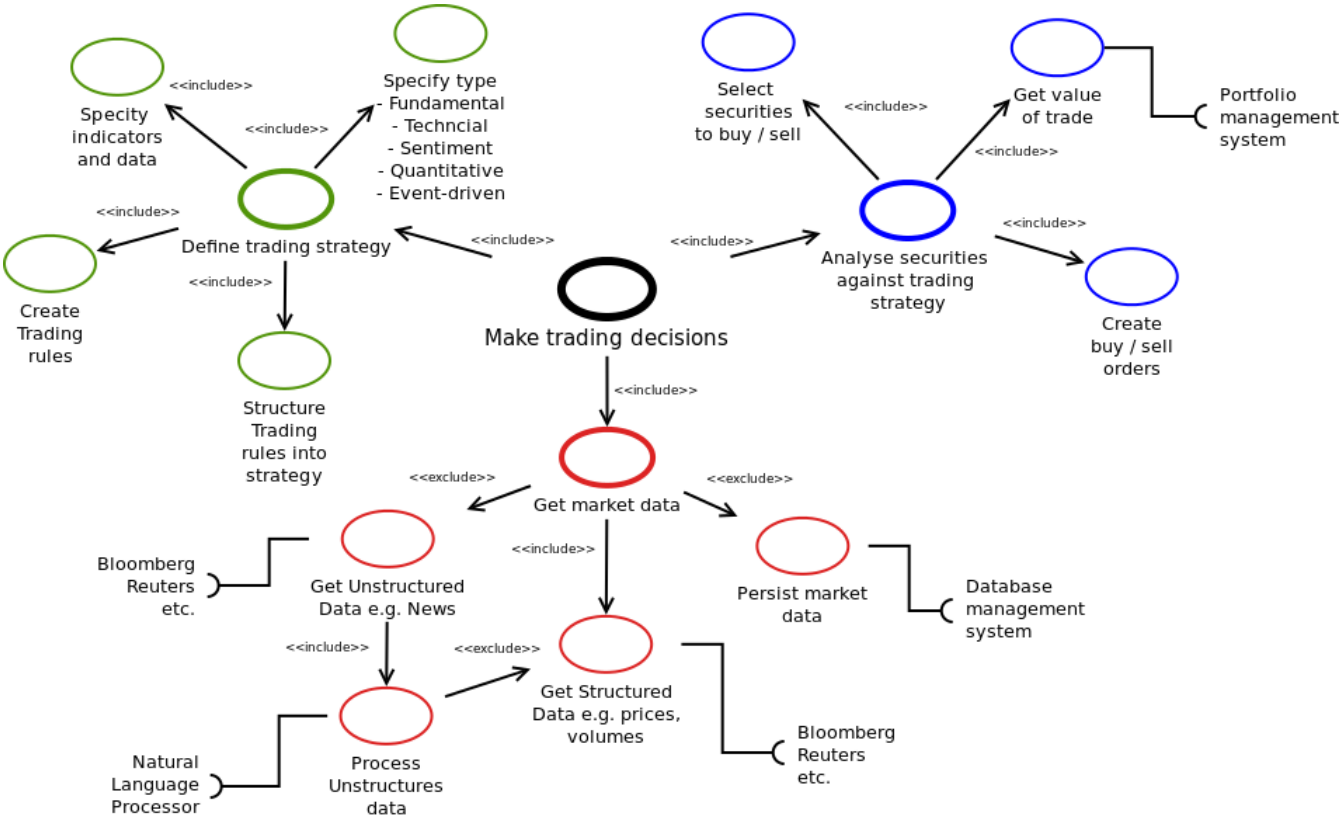
2.6. Functional scope of algorithmic trading systems

Functional requirements describe the functionality of a system. Given the context of algorithmic trading systems the top level functional requirements are the three purposes of an algorithmic trading system. Beneath each one of these top level requirements there are more high-level requirements. What follows is three use cases diagrams which illustrate the scope of high-level requirements for the OSATA algorithmic trading system architecture.

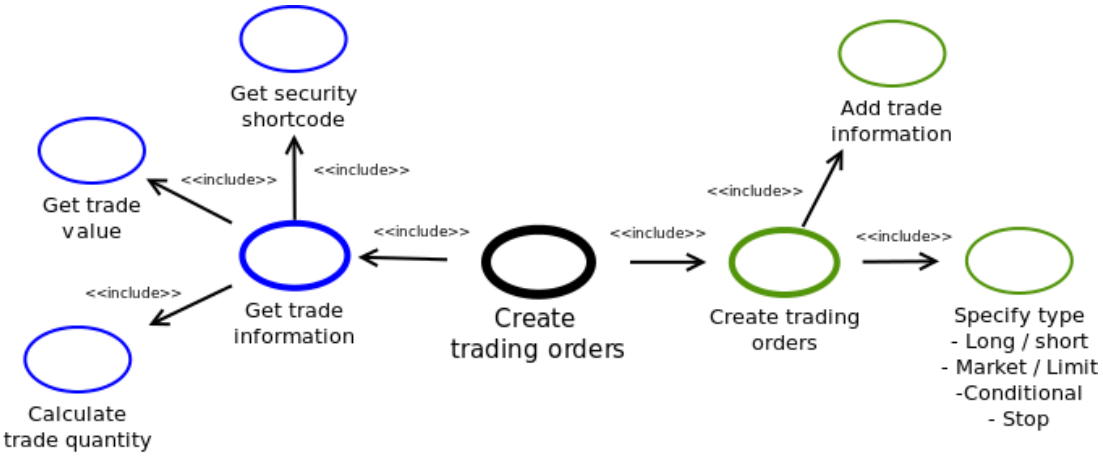
Note: I have also included a use-case diagram for portfolio management. These services are not core to the algorithmic trading system and would be outsourced to another system that integrates with the algorithmic trading system, especially if only not all of the portfolio is algorithmically traded e.g. options are algorithmically traded and bonds are not. That having been said, the algorithmic trading system is high dependent on the portfolio management system which would therefore need to conform to the same quality requirements as the algorithmic trading system so as to not impact the algorithmic trading system.

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

2.6.1. Make trading decisions: use case scoping

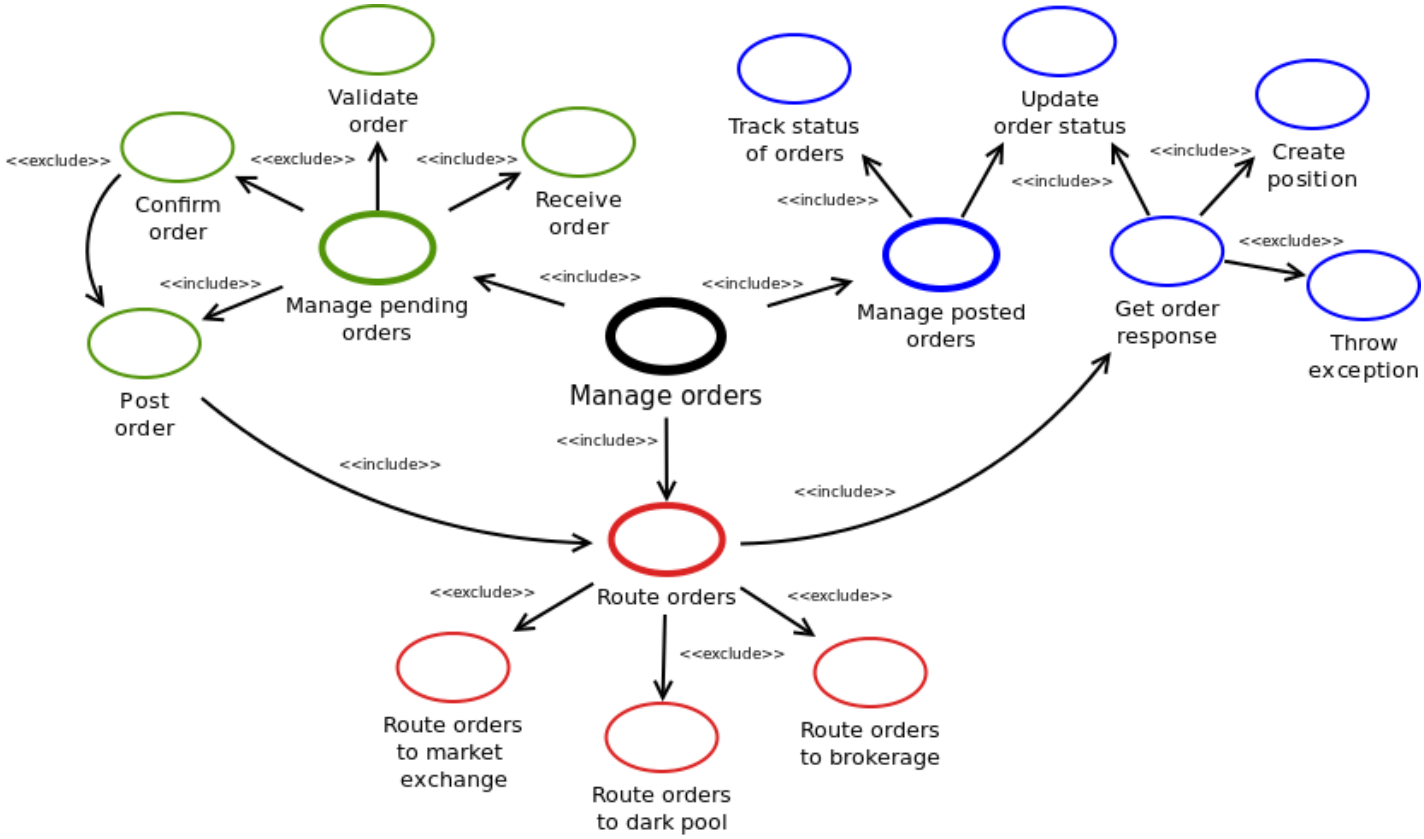


2.6.2. Create trading orders: use case scoping



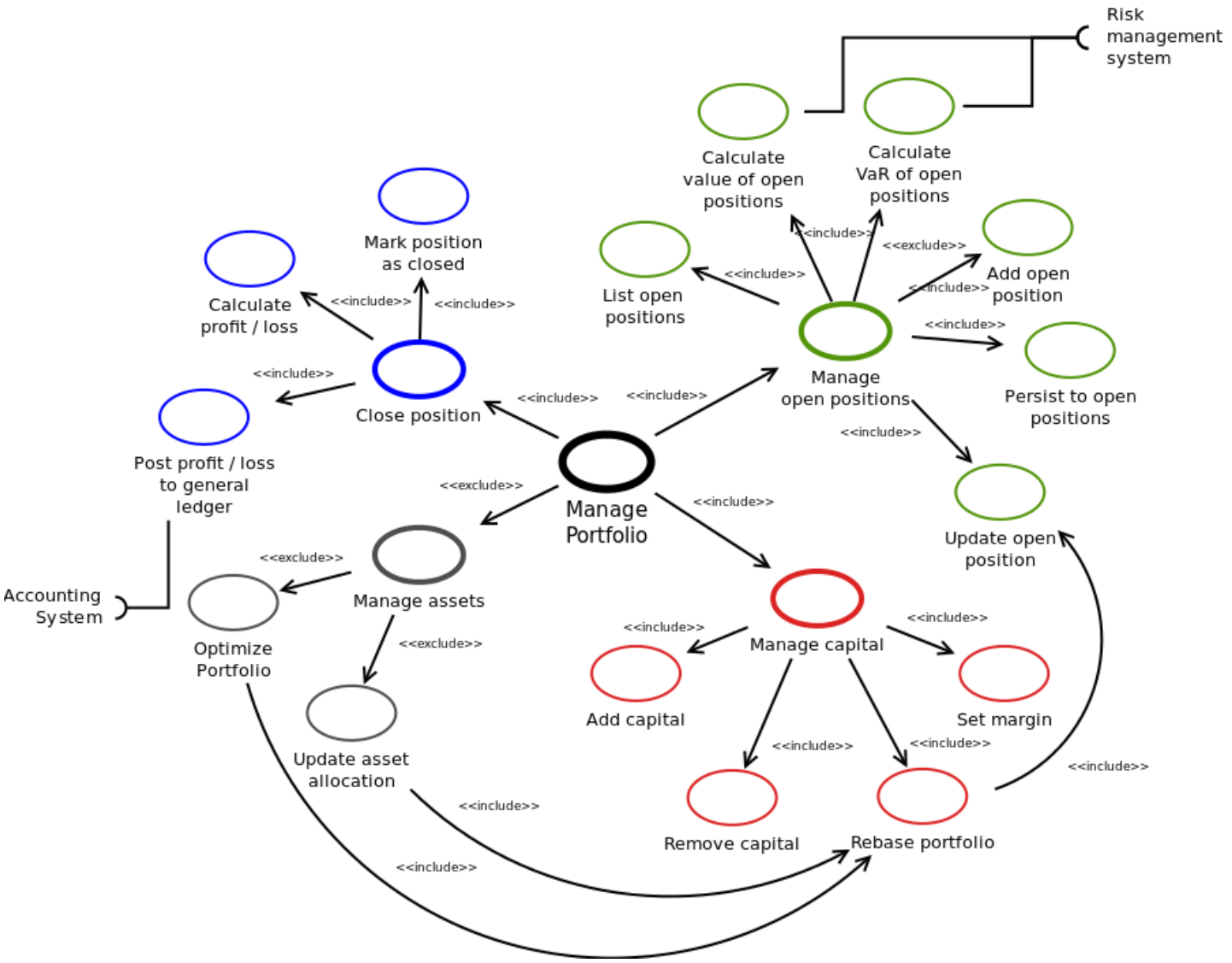
Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

2.6.3. Manage and monitor orders: use case scoping



Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

2.6.4. Portfolio management: use case scoping



Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

2.7. A note on complexity and dependencies

Systems are seldomly deployed in isolation of other systems. In a recent paper entitled 'the global financial markets: an ultra-large-scale systems perspective', researchers David Cliff and Linda Northorp of the University of Bristol, U.K, and Carnegie Mellon, U.S., argued that because of the complex network of financial systems, financial markets can be described as 'adaptive ultra large scale socio technical system of systems (SoS)'. They cited the rise in so-called Ultra-fast Extreme Events (UEE's) as evidence as this. As such, financial markets can be classified as complex adaptive systems which tend to exhibit emergent and chaotic behaviours. In light of this, standard engineering practices were called into question.

Algorithmic trading systems contribute to this SoS and in order to remain competitive they must also be adaptive. This is complicated by dependencies between the algorithmic trading system and other related systems. It is outside of the scope of this project to architect these external systems, but it is noted that their poor performance could impact upon the performance of the proposed algorithmic trading system. A non-exhaustive list of dependencies is shown below.

- ➔ Portfolio management systems
- ➔ Risk management systems
- ➔ Order management systems[3]
- ➔ Telecommunications networks and ISP's
- ➔ Accounting systems

[3] Note this would be if the order management system were to be built separately of the algorithmic trading system which would be possible

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

3. Architectural Scope

3.1. Core architectural responsibilities

The core responsibilities of the OSATA algorithmic trading system architecture will be to provide the following products and services:

3.1.1. Data responsibilities

- ➔ The architecture provides a data pre-processing zone which:
    - Supports multiple data input streams
    - Filters out all irrelevant input data
    - Performs temporal data partitioning
        - Producing temporal datasets

3.1.2. Processing responsibilities

- ➔ The architecture provides a distributed processing environment of clusters which:
    - Monitors connected clusters in real-time
    - Uses message oriented communications
    - Schedules temporal datasets between clusters
    - Is load balanced and has passive redundancy
- ➔ The architecture deploys clusters into the distributed processing environment which:
    - Provides in-memory queues between threads
    - Support virtualized processing units:
        - Threaded processing units for temporal data analysis:
            - Process unstructured data into events
            - Process structured data into signals
            - Post signals and events into queue
        - Processing units for the event and signals queue:
            - Persist events and signals to storage (memory)
            - Analyze events and signals to derive orders
            - Posts orders to the P-OMS
    - Provides an integration channel between the processing units and the P-OMS

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

3.1.3. Storage responsibilities

- ➔ The architecture provides a storage area network (SAN) which:
  - Performs temporal data aggregation
  - Supports continuous querying
  - Maintains a full audit trail
- ➔ The architecture provides disaster recovery (DR) storage which:
  - Replicates the SAN

3.1.4. Integration responsibilities

- ➔ The architecture provides an integration environment which connects:
  - Exposes a standard API for integration
  - Internal services and components to one another, and
    - Data preparation zone to clusters
    - Clusters to the SAN
    - Clusters to the P-OMS
  - External services and components to the algorithmic trading system
    - The data providers to the data preparation zone
    - The P-OMS to the markets, exchanges, and brokerages
    - Other external dependency systems that require integration

3.1.5. Order management responsibilities

- ➔ The architecture provides a parallel Order Management System (P-OMS) which:
  - Supports multiple input streams
  - Supports load-balancing
  - Has passive redundancy
  - Conforms to the ACID criteria
  - Maintains a full audit trail
  - Is fully replicated

3.1.6. System usage responsibilities

- ➔ The architecture provides a fully enabled front-end that:
  - Defines access profiles and access rights
  - Enables users to use the system in appropriate ways

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

4. Quality Requirements

4.1. Scalability

The OSATA architecture needs to be scalable in a number of different areas. This is because the architecture must scale in lock-step with the institutions trading activities. The rate at which financial markets change means that the trading activities of the institution could vary wildly from year to year. A system able to support this changing landscape must be scalable.

4.1.1. Data input streams must be scalable

The number of data inputs streams must be scalable so as to support:

1. Additional securities markets being added e.g. Stocks, Bonds, Options, Forex, etc.
2. Additional exchanges being added e.g. Nasdaq, NYSE, BSE, JSE, etc.
3. Additional data providers being added e.g. additional RSS feeds, private bank feeds

This would be required by the buy-side institution implementing an algorithmic trading system based on the OSATA architecture as well as the traders using that system.

Metric → number of supported data input streams

4.1.2. The distributed processing environment must be scalable

The processing environment needs to be scalable to support an increased scope in trading activities as well as an increased scope in the data being used to inform those trading activities. This would be required from the buy-side institution.

Metric → number of additional clusters it takes to process each new input stream over time, the ideal situation would be linear or near linearly scalable processing.

4.1.3. The

4.2. Performance

4.2.1. The OSATA architecture must guarantee high frequency response times

The OSATA architecture needs to achieve a very high degree of performance. This would largely depend on the holding periods in the trading strategies being used by the buy-side institution implementing OSATA. If the organization wishes to engage in High Frequency Trading (HFT) then the system would need to achieve response times to the market of less than 40 microseconds, but this is an extreme case. Many algorithmic trading has holding periods lasting hours, days, or even weeks. In which case performance is less critical.

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

Metric → Response time to market (measured in bips)

Metric → Transactions per second (TPS) or transactions per minute (TPM)[4]

4.3. Modifiability

4.3.1. Business logic for trading strategies must be modifiable

Maximizing the ease with which new algorithmic trading strategies can be added to the system represents a competitive advantage to the buy-side institution. In the OSATA architecture, the logic that defines the trading strategy is decoupled from the processing environment. In other words the processors interpret whatever strategy they are presented and analyze the data they received against that strategy. This allows the organization to adapt faster to changing market conditions and increase revenues as a result.

Metric → The time to market for new trading strategies (measured in days)

4.3.2. Data preprocessing and filtering must be modifiable

Not all of the data being received from the data subscriptions (especially the unstructured data produced by news aggregators) will be relevant to the trading being done by the buy-side institution. As such, data filtering and pre-processing will be a responsibility of the OSATA architecture. This should be easily modifiable through configuration files wherein specific feeds and filters can be applied or removed on the fly.

Metric → The ease with which filters can be applied and removed from the system whilst live

4.4. Reliability

4.4.1. Data preprocessing must be decoupled from data formatting and containers

The data coming into the algorithmic trading system needs to be of the highest quality as this directly impacts the reliability and accuracy of the output produced. One risk which is introduced when the system design is tightly coupled to the format in which data arrives, is that when that formatting changes, the system implementation breaks. In order to avoid this risk and improve the reliability of the system outputs, an architectural requirement is that the data be loosely coupled from the implementation of the data preprocessor(s).

Metric → Failure testing should be done with datasets using different formats / containers

[4] Benchmarks from IBM are available on this topic: http://publib.boulder.ibm.com/infocenter/idshelp/v10/index.jsp?topic=/com.ibm.perf.doc/perf43.htm

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

4.4.2. Transactions and orders must be 100% reliable

Because algorithmic trading systems deal with large sums of money which may or may not belong entirely to the buy-side institution doing the trading, the orders posted to the parallel order management system. The order management system should be accessible in parallel and represent the single entry and exit through which orders can be made. Test cases with known inputs and expected outputs should be regularly run through the system so as to establish the reliability of the services provided.

Metric → Pass rate on known inputs to expected outputs (100%)

4.5. Auditability

4.5.1. The OSATA architecture must be IT Auditable

This requirement stems from the auditors of such systems. The system should expose interfaces through which the IT General Controls may be tested and verified. ITGC are a set of controls that apply to systems components, processes, and data. The objectives of applying ITGC is to ensure proper development and implementation of systems, the integrity of programs, data files, and computer operations.

Metric → subjective

4.5.2. The OSATA architecture must be financially auditable

Since the 2008 stock market crash and the so-called 'flash crash' of 2010, the financial auditability of algorithmic trading systems has been a key requirement. As such, certain critical models and processes should be made visible to auditors:

1. Value securities in light of new information
2. Modelling of expected returns of securities

Metric - ease with which valuation assurance can be done on historic trades

4.6. Security

4.6.1. Intellectual Property (IP) must be secured

Because of the sensitivity and confidential nature of proprietary trading firms' algorithms, appropriate logical and infrastructural security controls surrounding should be implemented as a part of the OSATA architecture.

Metric → Number and severity of vulnerabilities identified during black-box and white-back

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

penetration testing.

4.6.2. Confidentiality, Integrity, and Availability of insider information is guaranteed

The confidentiality, integrity, and immediate availability of all information that might constitute insider information must be guaranteed:

1. Current open positions e.g. what securities the institution currently has on its books
2. Pending orders that have not been posted onto the market

This is a legal requirement, as both the knowing dissemination and use of insider information to inform investment activities is illegal.

Metric → Number and severity of vulnerabilities identified during black-box and white-back penetration testing.

4.6.3. Large orders must be obfuscated and obscured

Earlier in this document I mentioned the distinction between programmed trading and algorithmic trading. Because large orders posted by institutional traders have the potential to significantly impact the price of a security, large orders are broken up into many smaller pieces and executed over an extended period of time. This is done to reduce the risk of "whale hunting", which is the activity (done by other predatory algorithmic trading systems) of looking for large orders which will impact the stock price and "jumping" in front of them to profit from the rising or lowering tide. Large orders must therefore be obfuscated and obscured to improve security. This requirement is one shared by any large institutional buy-side trading institution.

Metric → This is very difficult to measure

4.7. Fault and Failure Tolerance

4.7.1. Significant components are fault tolerant

Historically speaking, a number of trading systems have seriously impacted the financial markets as a result of faults. Faults are to blame for the Knights Capital debacle of 2012. The risk of faults is not just monetary, but also legally and reputationally serious. The issue with Knights Capital resulted in a fine of $400 million.

Metric → Fault testing should be done on the OSATA architecture to see how it handles faults

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

4.7.2. Significant components have built in redundancy

A risk facing algorithmic trading systems is that certain components fall over. Consider the financial impact if the component responsible for managing trading orders failed. A number of positions would be held longer than was ideal which may result in trading losses. Similarly, consider the opportunity cost of the component which makes trading decisions failing which might result opportunities to trade being missed. A third, potentially more serious example, would be if the component responsible for logging transactions had to fail as this could result in misrepresented financial statement and legal ramifications.

Metric → Measuring and monitoring the probability of component failure

Metric2 → Failure testing on redundant components to test that strategies implemented perform as expected after a failure

4.8. Interoperability

4.8.1. The system must be interoperable with other financial systems

Interoperability is the characteristics of a system that allows is it to work together with a diverse range of related systems. As mentioned previously in this document there are numerous systems with which the algorithmic trading system must work with. To reiterate, these systems include, but are not limited to:

- ➔ Financial risk management systems
- ➔ Banking systems (for liquidity)
- ➔ Portfolio management systems
- ➔ Accounting systems
- ➔ Data service provider systems
- ➔ Order management systems
- ➔ Etcetera.

As such the system needs to be interoperable. This is required by the stakeholders of the interfacing systems as well as the buy-side institution.

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

5. Access and Integration Requirements

5.1. Integration channels

5.1.1. Data service providers protocols (protocols)

The protocols described in this section are standardized across the financial services industry internationally. They are used by market exchanges, sell-side institutions, buy-side institutions, and retail investor applications. As such, OSATA must provide support for them.

5.1.1.1. Support the financial information exchange (FIX)

The Financial Information eXchange (FIX) protocol was introduced in 1992 and is an electronic communications protocol for international real-time exchange of information related to the securities transactions and markets. A FIX packet contains two parts: the body and the trailer. The body contains the order details and the trailer contains the security information including a checksum. FIX is a self-describing protocol similar to XML. The OSATA architecture must support the FIX protocol both in its ability to construct and send FIX packets as well as receive and process them.

5.1.1.2. Support the FIX adapted for streaming (FAST)

The FAST protocol was developed by FIX Protocol Ltd. It aims to optimize the data representation on the network, support high-throughput, and low latency data communications between financial institutions. The OSATA architecture must also support the FAST protocol.

5.1.1.3. Support FIX Algorithmic Trading definition language (FIXatdl)

In addition to the FIX and FAST protocols, a new FIX-based protocol specifically targeted at Algorithmic Trading systems called FIXatdl has been created. The OSATA architecture must support version 1.1 of FIXatdl.

5.1.1.4. Support <unstructured> data

In addition to structured financial information protocols, the OSATA architecture must provide support for unstructured data streams such as those related to standard news. Included in this list should be the transportation protocols for news including NNTP (Network News Transfer Protocol) and RSS (Rich Site Summary).

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

5.1.2. Integration channels with enterprise financial systems

5.1.2.1. Integrates with enterprise systems

As mentioned a number of times in this document, the OSATA architecture needs to be interoperable with a number of enterprise financial systems. In order to achieve this quality requirement the OSATA architecture should:

- ➔ Expose a standardized application programming interface (API)
- ➔ Support XML and JSON file-based integration
- ➔ Support a standard messaging format
- ➔ Database integration (see 5.1.2.3.)

5.1.2.2. Integrates with banking systems via the SWIFT protocol

The Society for Worldwide Interbank Financial Telecommunication (SWIFT) provides a network that enables financial institutions worldwide to send and receive information about financial transactions in a secure, standardized and reliable environment. One critical part of an Algorithmic Trading system is accessibility to liquidity for trading purposes. As such, the OSATA architecture should support the SWIFT protocol so that transferral of funds to and from bank accounts to the Algorithmic Trading system is easier. This should be an optional feature of the algorithmic trading system, as the buy-side institution may already have a system connected to the global SWIFT network.

5.1.2.3. Database integration

The OSATA architecture should support a number of database connectors including the following popular connectors:

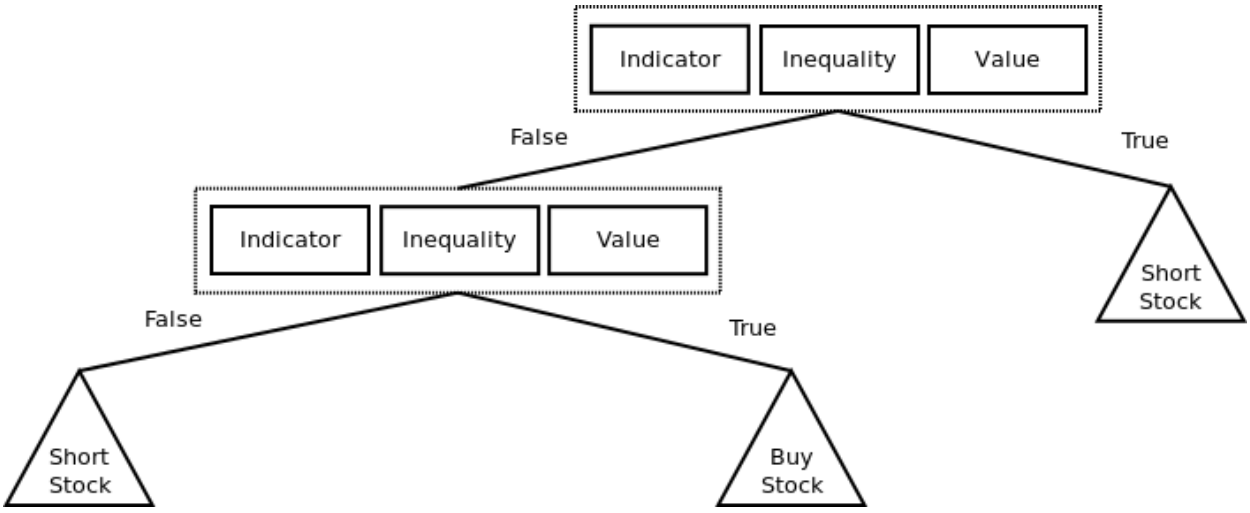
- ➔ ODBC (Open Database Connectivity) - is a standard programming language middleware API for accessing database management systems.
- ➔ JDBC (Java Database Connectivity) - is a standard API for the Java programming language for accessing database management systems.
- ➔ ADO (ActiveX Data Objects) - built by Microsoft this is a set of Component Object Model (COM) objects for accessing data sources.
- ➔ XQC (XQuery API for Java) - this is common Java API for the W3C XQuery 1.0 specification. It allows programs to execute XQuery against an XML data source.

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

5.2. Access Channels

5.2.1. Algorithmic trading strategy definition

One access channel is required to allow the users of the system to define new algorithmic trading strategies. The definition of such strategies would involve the structuring of trading rules into a decision making format (scripts, decision trees, quantitative models, etc.) Each rule would need to consist of some indicator, an inequality, and a value against which the indicator is compared using the inequality. In order to simplify this process, the access channel should be web-based where the web-page represents the front-end to a database. Through this interface the user could define new trading rules and then structure these rules into new strategies. This process is illustrated conceptually below for clarity.



Example decision tree structure for trading strategies

5.2.2. Administrative interface

An administrative interface should be exposed which allows system administrators to configure, control, and monitor components of the system. A non-exhaustive list of functionalities controlled through this access channel is listed below:

- ➔ Adding and removing data streams
- ➔ Adding filters to the data pre-processing environment
- ➔ Setting up of new clusters in the processing environment
- ➔ Real-time performance monitoring of components
- ➔ Configuration of logging (e.g. verbosity of logging)
- ➔ Definition of security policies

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

- ➔ Management of security components (access, firewalls, etc.)
- ➔ Network configuration (DNS, Routing, etc.)
- ➔ Management of data-backups
- ➔ Other system administration functions.

5.2.3. Auditor interface

A third interface should be exposed specifically for auditors. It should allow the user (auditor) to generate a number of compliance reports in a succinct, technology neutral format e.g. CSV / Excel. It should also expose a limited interface to the database management systems which store historical trade information and security controls (e.g. users' access rights).

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

7. Architectural Constraints

7.1. Compliance to financial and technological standards

7.1.1. Financial Standards

The financial record keeping done by the OSATA architecture should be compliant to relevant financial standards. These standards will vary depending on the country in which the system is deployed. However, as a good rule of thumb the system should conform to the International Financial Reporting Standards (IFRS) these are designed as a common global language for business affairs to make company accounts understandable and comparable across international boundaries.

7.1.2. Technological Standards

There are a number of standards for algorithmic trading systems. These standards are normally set out by the exchanges through which the systems trade. As such, for each exchange the standards should be researched and conformed to. A couple such standards are listed below:

7.1.2.1. The SEC's Regulation Systems Compliance and Integrity

The Securities and Exchange Commision (SEC) of the USA has proposed that before trading on U.S. exchanges algorithmic trading systems need to conform to Regulation SCI. This also requires that algorithmic trading systems undergo a series of mandatory tests. These tests were proposed in March of this year and are likely to be finalized within the next couple of years. According to the SEC's fact sheet "Regulation SCI is intended to reduce the chance of technology problems occurring in the first place and ensure that key entities are well-positioned to take appropriate corrective action if problems do occur."[5]

7.1.2.2. The ESMA guidelines for Algorithmic Trading systems

The European Securities and Markets Authority (ESMA) has a publicly available consultation paper proposing standards for Algorithmic Trading systems operating in Europe. The second round of this paper was published in 2002 and has been adopted for european markets[6].

7.1.2.3. The ISO Algorithmic Trading 9000 (AT9000) standard

A working group from the ISO (International Organization for Standardization) has begun work on

[5] Fact sheet is available here: http://www.sec.gov/News/Article/Detail/Article/1365171517000#.Ukgnnxiv87x
[6] Download a copy of the ESMA guidelines here: http://www.esma.europa.eu/content/Proposed-Standards...

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

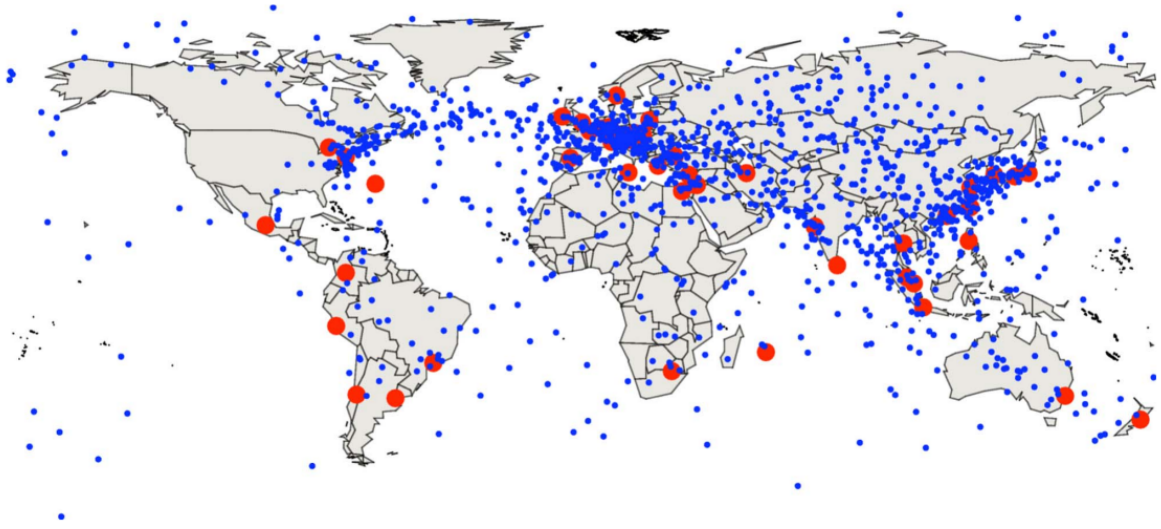
an ISO 9000 standard for Algorithmic Trading. The ISO 9000 family addresses various aspects of quality management and contains some of ISO's best known standards. For more information about the working group visit their website: http://www.at9000.org/.

7.2. Compliance to standard international integration protocols

As mentioned in the quality requirements section of this document there are a number of international standards for financial information exchange. These standards are governed and set out by the FIX community. Compliance to these standards represents an architectural constraint on the OSATA architecture.

7.3. Physical network constraints

As illustrated in the diagram on the cover page of this article (and shown again below), the performance of an algorithmic trading system is dependent on its geographic location. This is because of network latencies. In order to implement a high frequency trading system, the location of that system should be taken carefully. One option is to co-locate the algorithmic trading system in the same datacenter as the exchange. This is a service offered by most international exchanges, but can be quite costly. To co locate with the JSE costs R51,000.00 per month for a single rack[7].



A recent study done by MIT identified those physical locations where the network latency is minimized (blue dots)

[7] For some more information on co-locating with the JSE please download the following fact sheet http://www.jse.co.za/Libraries/Colocation_Documents_2013/JSE_Co-location_fact_sheet.sflb.ashx

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

7.4. Organizational constraints

Additional organizational constraints may constrain the realization of the OSATA architecture. This could include the use of specific technologies, dependent systems, budgetary constraints, and resource constraints.

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

8. Conclusion

Algorithmic trading system architectures are complicated by the strict quality requirements expected of the system and the wide range of regulatory and compliance requirements governing automated trading. Because of these complexities, careful consideration should be payed to the design and implementation of the system architecture. In designing an open source algorithmic trading architecture, OSATA, hopes to point out those architectural requirements that are often overlooked at the onset of designing such systems. The requirements identified in this document are unlikely to be complete and will inevitably evolve as the project progresses.

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com

References

- ★ Algorithmic Trading, available definitions:
  Investopedia: http://www.investopedia.com/terms/a/algorithmictrading.asp
  Wikipedia: http://en.wikipedia.org/wiki/Algorithmic_trading
  Automated Trader: http://www.automatedtrader.net/Algorithmic_Trading.xhtm
- ★ The global financial markets:an ultra-large-scale systems perspective, The Future of Computer Trading in Financial Markets driver review – DR 4, D. Cliff and L. Northorp. Available at: http://arxiv.org/pdf/1109.3444.pdf
- ★ Algorithmic Trading, G.Nuti, M. Miraghaemi, P. Treveleaven, and C. Yingsaeree. Available at: http://www.ccrc.wustl.edu/~roger/361S.f11/mco2011110061.pdf
- ★ A. Chaboud et al., “Rise of the Machines: Algorithmic Trading in the Foreign Exchange Market,” Int’l Finance Discussion Papers, Board of Governors of the Federal Reserve System, Oct. 2009; www.federalreserve.gov/pubs/ifdp/2009/980/ifdp980.pdf
- ★ Algorithmic and High-frequency trading: an overview, Marco Avellaneda, New York University & Finance Concepts LLC. Available at: http://www.math.nyu.edu/faculty/avellane/QuantCongressUSA2011AlgoTradingLAST.pdf
- ★ Algorithmic Trading Trends and Drivers, Tellefsen Consulting Group, January 2005. Available at: http://www.tellefsen.com/Algorithmic_Trading_TCG.pdf
- ★ FIX Trading Community Homepage → FIX, FAST, FIXatdl
  FIX Available at: http://www.fixtradingcommunity.org/pg/structure/tech-specs/fix-protocol
  FAST Available at: http://www.fixtradingcommunity.org/pg/structure/tech-specs/fast-protocol
  FIXatdl Available at: http://www.fixtradingcommunity.org/pg/structure/tech-specs/fixadtl
- ★ Algorithmic Trading ISO 9000 standard
  Homepage: http://www.at9000.org/
  ISO 9000: http://www.iso.org/iso/iso_9000
- ★ SEC Compliance for Algorithmic Trading
  Article available at: http://marketsmedia.com/sec-wants-algos-tested-before-deployment/
  SCI Regulation: http://socfinance.wordpress.com/2013/08/16/regulation-sci-systems-compliance-and-integrity/
- ★ IFRS (International Financial Reporting Standards)
  About IFRS: http://www.ifrs.org/IFRS-for-SMEs/Pages/IFRS-for-SMEs.aspx
  Wikipedia: http://en.wikipedia.org/wiki/International_Financial_Reporting_Standards
- ★ Database connectors information
  ODBC: http://en.wikipedia.org/wiki/ODBC
  JDBC: http://en.wikipedia.org/wiki/Java_Database_Connectivity
  ADO: http://en.wikipedia.org/wiki/ActiveX_Data_Objects
  XQJ: http://en.wikipedia.org/wiki/XQuery_API_for_Java
- ★ SWIFT Standards
  Homepage: http://www.swift.com/index.page?lang=en
  FIX on SWIFT: http://old.fixprotocol.org/vendors/4870

Contact: http://www.stuartreid.co.za or stuartgordonreid@gmail.com