Evaluation of a Real-Time Buying Power Engine for Wealth and Asset Management

By

Mehraz Khan

Supervisor: Professor Joseph C. Paradi

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science
Graduate Department of Chemical Engineering and Applied Chemistry
University of Toronto

© Copyright by Mehraz Khan 2017
Evaluation of a Real-Time Buying Power Engine for Wealth and Asset Management
Masters of Applied Science, 2017
Mehraz Khan
Chemical Engineering and Applied Chemistry
University of Toronto

Abstract

Large banks provide wealth management services to their clients, through which investors can use cash and borrowed funds to invest in securities. The implementations of technologies for these services are long-term, capital projects which require rigorous assessment before selection and implementation. An evaluation framework was developed by incorporating individual frameworks and assessing capabilities. The goal of this framework is to provide a standardized approach for the evaluation of various technology vendors.

One potential vendor was evaluated using the developed framework, outlining performance and capability metrics, gaps and action items for compliance to the Bank’s needs. Using the results of the evaluation framework, a recommendation was provided to the Bank for further steps towards implementation of the vendor’s solution.
Acknowledgments

Dr. Paradi has been an exceptional supervisor, mentor and friend throughout this entire process. He has provided the help and expertise required to achieve my academic goals, the freedom to achieve my personal goals and sometimes, the necessary push for me to do both.

Dr. Fai Tam has been a great partner throughout this journey, providing his expertise, support and friendship during my time at the lab.

Additionally, the many members of the organizations I worked with throughout this thesis were a great help. Their time and commitment to this project has been invaluable.
# Table of Contents

Acknowledgments .................................................................................................................. iii
Table of Contents .................................................................................................................... iv
Glossary ................................................................................................................................... vii
List of Tables ............................................................................................................................ viii
List of Figures ........................................................................................................................... ix

1 Introduction .......................................................................................................................... 1
   1.1 Thesis Outline .................................................................................................................. 3

2 Literature Review .................................................................................................................. 5
   2.1 Canada's Securities Industry .......................................................................................... 5
   2.2 Investment Banking ......................................................................................................... 5
   2.3 Margin and Wealth Management ..................................................................................... 7
   2.4 Technology for Wealth Management ............................................................................. 7
      2.4.1 Automation of exchanges and trading ................................................................. 8
      2.4.2 Algorithmic trading ............................................................................................... 8
      2.4.3 Credit monitoring ................................................................................................. 9

3 Wealth and Asset Management ......................................................................................... 12
   3.1 Trading Accounts ............................................................................................................ 12
      3.1.1 Direct Investing (DI) Accounts ......................................................................... 12
      3.1.2 Personal Investment Advice (PIA) Accounts .................................................. 14
      3.1.3 Registered Accounts ......................................................................................... 14
   3.2 Financial Instruments .................................................................................................... 14
      3.2.1 Shares .................................................................................................................. 14
      3.2.2 Bonds .................................................................................................................. 14
      3.2.3 Mutual Funds ...................................................................................................... 15
      3.2.4 Exchange Traded Funds (ETFs) ........................................................................ 15
      3.2.5 Listed Options ..................................................................................................... 15
      3.2.6 Principal Protected Notes .................................................................................. 15
      3.2.7 Foreign Securities ............................................................................................... 16
      3.2.8 Precious Metals Certificates .............................................................................. 16
      3.2.9 Options ............................................................................................................... 16
      3.2.10 Strips .................................................................................................................. 16
   3.3 Orders ............................................................................................................................. 17
      3.3.1 Order Parameters ............................................................................................... 17
      3.3.2 Fill Type ............................................................................................................. 18
      3.3.3 Order Type ........................................................................................................ 18
   3.4 Regulatory Environment ............................................................................................... 19
      3.4.1 IIROC Rule 100 – Margin Requirements ......................................................... 19

4 Problem Overview .............................................................................................................. 20
   4.1 Opportunities Arising from the Current Process ......................................................... 20
      4.1.1 Non-Real-Time Solution ...................................................................................... 20
      4.1.2 Margin Optimization ......................................................................................... 21
   4.2 System Requirements .................................................................................................... 21
      4.2.1 Functional Requirements ................................................................................... 21
      4.2.2 Non-Functional Requirements ......................................................................... 24
Glossary

**Account Equity** – An account’s equity consists of the net assets of the account.

**Cash Equivalents** – Cash equivalents are highly liquid short term assets.

**Collateral** – Collateral is an asset that a borrower offers a lender in order to secure a loan. In the case that the borrower stops making loan payments, the lender can claim the collateral which is used to recoup any losses.

**Concentration Limit** – Concentration limits are the maximum value up to which an asset or asset class can be collateralized, which loans or margin on the asset or asset class do not exceed.

**Engine** – An engine is a component that performs a function for a different component, system or application.

**Exchange** – An exchange is a marketplace where financial instruments are traded. Exchanges ensure orderly trading as well as provide accurate pricing information.

**Execution** – An execution is the completion of a securities order. Execution occurs when an order is completely filled.

**Gap** – A gap is when the current capabilities of a system do not meet the required capabilities.

**Market** – A market refers to the virtual or physical environment where securities are traded.

**Maturity** – Maturity is the time period, at the end of which a financial instrument ceases to exist and any actions are taken and repayments are made.

**Settlement** – Settlement refers to when a buyer must pay for securities delivered by a seller.
List of Tables

Table 1. Canadian Banking Leaders by Assets................................................................. 7

Table 2. Option Types........................................................................................................ 16

Table 3. Order Parameters ................................................................................................. 17

Table 4. Fill Types ............................................................................................................ 18

Table 5. Conditional Order Types ....................................................................................... 19

Table 6. Client Information Inputs...................................................................................... 30

Table 7. Order Information Inputs ....................................................................................... 31

Table 8. Rule Engine Scenarios .......................................................................................... 33

Table 9. Header Tags ........................................................................................................... 39

Table 10. Trailer Tags ......................................................................................................... 40

Table 11. Database Table ................................................................................................... 45
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Client Assets Under Management in Canada (IIROC 2014)</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>TD Bank Block Trading (TD Securities 2016)</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Vendor Overall Architecture Diagram</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>Business Rules Framework Architecture (Microsoft 2016)</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Message Switch Diagram</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Message Switch Process Flow</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>Market Feed Diagram</td>
<td>44</td>
</tr>
<tr>
<td>8</td>
<td>Query Engine Diagram</td>
<td>47</td>
</tr>
<tr>
<td>9</td>
<td>Hybrid Supervisor Solution</td>
<td>51</td>
</tr>
<tr>
<td>10</td>
<td>Measured Latency for Regular (top) and Burst (bottom) Workloads (Marek, et al. 2014)</td>
<td>54</td>
</tr>
<tr>
<td>11</td>
<td>Test Scenario Message vs Time Graph</td>
<td>56</td>
</tr>
<tr>
<td>12</td>
<td>Production Data Messages vs Time Screenshot</td>
<td>57</td>
</tr>
<tr>
<td>13</td>
<td>Transaction Latency Data</td>
<td>58</td>
</tr>
<tr>
<td>14</td>
<td>REST Architecture</td>
<td>60</td>
</tr>
<tr>
<td>15</td>
<td>Client-Server Separation (Microsoft 2016)</td>
<td>61</td>
</tr>
<tr>
<td>16</td>
<td>Consumer Side Caching of Data (Microsoft 2016)</td>
<td>62</td>
</tr>
</tbody>
</table>
Figure 17. Dashboard Interface Organization (top) and mockup (bottom) ........................................ 72
1 Introduction

The wealth and asset management industry facilitates transactions where clients use cash to purchase financial securities. Among the major players in the industry are banks, which offer a wide range of financial services to consumers including wealth and asset management. In addition to facilitating these transactions by serving the role of intermediary between the client and the marketplace, banks also facilitate a client’s purchasing power by providing loans against collateral assets through “margin” accounts. In these accounts, the bank loans to the client an amount relative to the financial securities held in the account. As markets are continually fluctuating, the values of the assets upon which these loans are based are also continually changing, as is the loan value that is within the bank’s risk tolerance. To reduce risk, banks adopt technologies to monitor market changes against their accounts. As technology has advanced, banks are now moving from batch processes towards real-time monitoring technologies for this function.

The investment industry is a highly regulated environment, governed by the Investment Industry Regulatory Organization of Canada (IIROC) in Canada, and the Securities and Exchanges Commission (SEC) in the United States. IIROC’s mandate is to set and enforce rules regarding proficiency, business and financial conduct of dealer firms and their employees and enforce market integrity rules regarding trading activity on Canadian equity marketplaces.

The industry is highly regulated and the majority of offered products are similar among all banks. As the banks cannot distinguish themselves by the market structure and conventions available, they must offer a unique experience for the clients to gain more clients and increase market share.
Technology and its advancement have drastically changed the way in which Canadians bank. Online banking is now the primary means of banking for 55% of Canadians. According to the Canadian Banking Association, customers are embracing new technologies and say they see real value in increased convenience, choice and flexibility (Canadian Bankers Association 2015). The adoption of new technologies to improve user experience is important to make banks more competitive.

The study and research to be documented was initiated by a Canadian multinational banking and financial services corporation (to be referred to as “the Bank”). The Bank, as do all its competitors, operates a wealth management business, which offers investors a variety of investment products and services assigned to one or more accounts (portfolios). Through Wealth Management, clients can invest money in financial securities using both cash and funds borrowed from the Bank.

As technology is fundamental to the success of banks, they must carefully evaluate solutions in order to implement the most appropriate solution, factoring in all the banks’ and their clients’ needs. In order to evaluate all vendors in a manner that is most objective, a standardized approach is required, which is developed here.

The Bank’s current vendor solution used to perform capital calculations, determining the credit risk and approved credit terms for a wealth and asset management client, does not function in real-time and does not optimally calculate risk offsets, potentially resulting in greater risk to the bank and lower customer satisfaction to its clients. The bank wishes to determine if a new vendor solution is feasible for implementation while improving on the offering provided by the current vendor in the areas of real-time functionality, speed and increased client buying power, as well as aligning to the Bank’s overall internal technology strategy. Only one vendor was
evaluated as no other vendors currently offer a real-time solution to the Bank’s problem. The research will provide the Bank with the evaluation of the vendor solution, its ability to meet the Bank’s needs and a recommendation to proceed with further evaluation and implementation of the solution.

This thesis documents and evaluates a cutting edge technology for use in the wealth and asset management industry, in the aspects of communications, software and hardware architecture, as well as, performance. The Vendor is the only one to provide this solution which has not been used for large scale applications equivalent to the Bank’s wealth and asset management business.

1.1 Thesis Outline

The document is organised as follows: Chapter 2, Literature Review, summarizes findings in literature which provides a background of the investment industry in Canada. Chapter 3, Wealth and Asset Management, provides a more technical and functional overview of the wealth and asset management industry. Chapter 4, Problem Overview, outlines the Bank’s problem, their current solution, the requirements of the potential solution and problem assumptions. Chapter 5, Introduction to Vendor’s Solution, details the Vendor’s proposed solution and their current capabilities. Chapter 6, Research Topics and Findings, summarizes the research done for this thesis which explores the motivation of the project and trends in technology within the wealth management industry. Chapter 7, Vendor System Evaluation, outlines the evaluation of the Vendor’s solution against findings in literature and the client’s needs. Chapter 8, Conclusion and Recommendations, revisits the project, summarizes the goals and findings of the thesis, outlining gaps in capabilities. The recommendation on the vendor and solution is made and explained in this section. Chapter 9, Future Work, discusses further steps
based on the findings of this thesis including evaluation of other vendors, further evaluations of the current vendor and implementation steps.
2 Literature Review

The literature review section first discusses technology and the critical role it plays within finance and wealth management. Future trends in technology as enterprise solutions are then discussed.

2.1 Canada's Securities Industry

The Canadian securities industry plays a key role in Canada’s financial services sector. It enables businesses and governments to raise debt and equity capital and allows investors to trade with confidence in open and fair capital markets. This is a large marketplace as seen below:

![Client Assets Under Management](image)

*Figure 1. Client Assets Under Management in Canada (IIROC 2014)*

2.2 Investment Banking

Financial institutions, in this case a Canadian Bank, execute client and proprietary orders in listed and over-the-counter cash securities, as well as convertible and derivative securities. In
order that the client has sufficient funds or credit, pre-execution credit evaluation is an essential part of the Bank's trading activities. An example of this is the activities of the TD Capital markets where TD Securities Inc. offers access to over 500 institutional equity asset managers in Canada, and key qualified institutional buyers in the U.S. and U.K. This substantial organisation serves as a foundation to execute both large and small transactions in order to provide the most liquidity for their clients. Here is an example from their website (TD 2016) showing block trading activities:

![Figure 2. TD Bank Block Trading (TD Securities 2016)](image)

Canada's three largest participants in this marketplace are: The Royal Bank of Canada (RBC), The Toronto Dominion Bank (TD) and the Bank of Nova Scotia. Other significant institutions in the industry include the Bank of Montreal (BMO) and Canadian Imperial Bank of Commerce (CIBC).
Table 1. Canadian Banking Leaders by Assets

<table>
<thead>
<tr>
<th>Bank</th>
<th>Assets Under Management ($ Billion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Royal Bank of Canada</td>
<td>$1,074 (Royal Bank of Canada 2015)</td>
</tr>
<tr>
<td>Toronto Dominion Bank</td>
<td>$1,104 (Toronto Dominion Bank 2015)</td>
</tr>
<tr>
<td>Bank of Nova Scotia</td>
<td>$856 (Bank of Nova Scotia 2015)</td>
</tr>
<tr>
<td>Bank of Montreal</td>
<td>$642 (Bank of Montreal 2015)</td>
</tr>
<tr>
<td>Canadian Imperial Bank of Commerce</td>
<td>$463 (Canadian Imperial Bank of Commerce 2015)</td>
</tr>
</tbody>
</table>

2.3 Margin and Wealth Management

Improper margining has been a topic of discussion related to past instabilities in the securities markets. John Galbraith concluded that the 1929 market crash was caused by improper margin management (Galbraith 1954). The U.S. Securities and Exchange Commission (SEC) concluded that improper margin management was also one of the factors that led to the 1987 crisis (SEC 1998).

Margin has been used by investors as a tool to lever their investments to increase expected profits. Conversely, when markets don’t behave as investors predict, the use of margin also exacerbates losses (Fortune 2001).

2.4 Technology for Wealth Management

The adoption of technology has been critical for the global financial industry – today it is essential as without it there would not be such an industry. In wealth and asset management, the adoption of specialized technology has revolutionized operations and capabilities for both institutions and clients alike by enabling on-line portfolio valuation and credit limits.
2.4.1 Automation of exchanges and trading

Trading of financial instruments has become a fully electronic process. Trades are now completed through communication between computer systems instead of face-to-face by traders. A review article by Dr. Vary Coates states that the securities market had essentially become information switches (Coates 1992). Academic studies reported that automation advances have changed the cost for trading services to the investors’ benefit (Domowitz and Steil 1999). In 2003, a study evaluated exchange revenue and cost efficiency in relation to technology and automation. They determined that the most technologically advanced exchanges were most efficient with respect to cost and revenue and that investment in technology developments positively influenced these factors (Hasan, Malkamaki and Schmeidel 2003).

The importance of updating technology is seen in practice, such as with the Tokyo Stock Exchange, which is implementing technologies used in the New York Stock Exchange (Investment Weekly News 2010).

2.4.2 Algorithmic trading

Investopedia defines this as:

"Algorithmic trading (automated trading, black-box trading, or simply algo-trading) is the process of using computers programmed to follow a defined set of instructions for placing a trade in order to generate profits at a speed and frequency that is impossible for a human trader".

Algorithmic trading (also known as AT), is the use of computers to automate one or more stages of the trading process. These algorithms attempt to profit from statistical patterns across financial markets by optimally assembling simple to complex orders. As of 2011, algorithmic trading systems accounted for approximately 50-60 percent of all stock trades in the US and EU (Nuti, et al. 2011).
Beyond making a profit, AT is effective in providing liquidity in stock markets and reducing price distortion ascribed to delay of information (Seo and Chai 2013).

Another computerised trading system is High Frequency Trading that is defined by Investopedia as: "High frequency trading is an automated trading platform used by large investment banks, hedge funds and institutional investors which utilizes powerful computers to transact a large number of orders at extremely high speeds.

High frequency trading is often considered a subset of algorithmic trading with differences in their operations. HFT strategies use algorithms to move in and out of positions at fractions of seconds and often do not hold inventory overnight (Carrion 2013). According to the New York Times, HFT accounted for approximately 50% of US equity trading volume in 2012 (New York Times 2012).

When observing the effect of HFT on liquidity, NASDAQ datasets displayed that HFTs benefit the market by providing liquidity when it is scarce, as well as, consuming liquidity by purchasing stocks when they are readily available (Carrion 2013). When observing the effect of HFT on price discovery and efficiency, the same NASDAQ dataset displayed that it increases the efficiency of prices by trading against the direction of pricing errors and in the direction of permanent price changes (Brogaard, Hendershott and Riordan 2014).

2.4.3 Credit monitoring

In the 2010 Commercial Lending Review, Michael Kozisek identified that lax collateral monitoring results in greater loan losses. One of the many variables that attribute to collateral monitoring is the frequency at which it is reviewed. In the event that collateral values are volatile, frequent revaluation of collateral reduces risk.
The 2010 Commercial Lending Review identified a number of common problems in credit processes (Hanley and Barrickman 2010):

- Analysis required to support a loan application is too extensive or identical for all applications regardless of the size of relationship exposure.

- The flow of loans through the credit approval process is jammed due to a number of factors including communication among different teams.

- The current credit monitoring processes used across a number of financial institutions are not fully optimized yet, and, in fact, still require some manual input which slows down the time it takes to review and take action on loan applications. Furthermore, manual intervention results in human errors that impact the entire process.

Hanley and Barrickman also concluded that by improving the credit process, customer experience, lender productivity and portfolio profitability could also be improved (Hanley and Barrickman 2010).

In the 2000 Commercial Lending Review, Deborah Williams highlights two major components of credit risk management solutions: First, determining whether a customer will default, and second, determine what the financial institution’s losses would be in the case of a default (Williams 2000). There are a number of different models used to quantify credit risks and many papers which compare the effectiveness and limitations of these models. One such paper led by Michel Crouhy finds limitations relating to fiscal factors such as finding data related to the country, sector and asset, as well as human factors such as internal expertise of credit departments in banks (Crouhy, Galai and Mark 2000). In the past, the industry focus has been on the first component, to determine if the investor would be able to meet their requirements of
financial contracts as monetary losses can be quantified, whereas predictive models to achieve the first component of the solution are constantly changing with varied accuracy and precision. This trend continues with the adoption of machine learning to limit human involvement within the credit process (Piramuthu 2006).
3 Wealth and Asset Management

The wealth and asset management industry is a complex business environment. There is a large variety of products and services offered by markets and institutions alike. There are many different types of securities available for purchase, which differ by type, value, location and exchange. These securities can be purchased by means of orders which have their own complexity. A number of orders and securities can be combined to form trading strategies. All this occurs in a highly regulated environment which controls the actions that can be taken and by whom.

3.1 Trading Accounts

The Bank’s Wealth Management business offers a variety of products and services to clients by means of different accounts. The capabilities of the client are limited by the type of account and the value of the account. Limitations come in the form of actions that can be taken such as the type of orders, the types of securities, the value of transactions and the trading strategies that can be used. The different types of accounts also determine the applicable regulations and house rules which effect capabilities and buying power. (Bank Employee, Project Introduction Meeting 2 2015)

3.1.1 Direct Investing (DI) Accounts

Direct investing accounts are non-registered accounts through which Bank clients are able to deposit and invest funds without input from a Bank Investment Representative. DI accounts are subcategorized as cash or margin accounts. (TD 2015)

3.1.1.1 Cash Accounts

Cash accounts are a category of DI accounts where investors must pay cash in full for each purchase. Funds must be available in the account on or before the settlement date for each
transaction. Buying power for cash accounts are typically twice the cash and money market investments in the account. The client can purchase up to this amount on credit but must have the purchase amount in cash in the account upon settlement. Cash equivalent to the full transaction cost must be present in the account upon settlement. (TD 2015)

\[ \text{Buying Power} = 2 \times \text{Cash and Money Market Investments} \]

3.1.1.2 Margin Accounts

Margin accounts are a category of DI accounts where investors may borrow funds against securities already in the account or borrow part of the purchase price of securities being bought as a loan, on which, interest is paid until the loan is repaid. Margin accounts have enhanced functionality, potentially allowing investors to trade options and short sell. Buying power in margin accounts are a function of the value of the account holdings and the margin rate of the securities, calculated as the cash in the account, plus the sum of securities and their margin rates up to the concentration limit, subtracted by any margin requirements, accounting for strategies and offsets. (Bank Employee, Project Introduction Meeting 2 2015)

When funds are borrowed, a certain net value must be maintained in the account, determined by the loan rates against which funds were borrowed. In the event that market fluctuates and the value of the account goes below this required net value, the account is considered “off-side” and results in a “margin call”. Funds must be transferred to the account to close the gap. (Bank Employee, Project Introduction Meeting 2 2015)

\[ \text{Buying Power} = \text{Cash} + \sum \text{Minimum of (margin rate x account holdings) or concentration limit} - \text{Short position margin requirements + offsets} \]
3.1.2 Personal Investment Advice (PIA) Accounts

PIA accounts are investment accounts where a financial advisor assists in the account by providing professional advice to the account holder. Trades are placed through the advisor. These accounts require an account value of at least $150,000. (TD 2015)

3.1.3 Registered Accounts

Registered accounts are a separate classification of trading accounts which follows specific rules to this type of account which are set by the governing regulatory body. Registered accounts in Canada consist of Registered Education Savings Plans (RESP), Registered Retirement Savings Plans (RSP), Tax Free Savings Accounts (TFSA), Registered Disability Saving Plans (RDSP) and Registered Retirement Income Funds (RIF). These accounts must have sufficient cash and cash equivalents for any transaction. (TD 2015)

\[ \text{Buying Power} = \text{Cash and Cash Equivalents} \]

3.2 Financial Instruments

Financial instruments are documents which represent a legal agreement involving some sort of monetary value (Arratia 2014). Instruments are often classified as equity based or debt based. More complex instruments exist as derivatives based on underlying assets.

3.2.1 Shares

Shares are an equity security. The owner of the share owns a part of the company which issued the shares. Shares entitle owners to certain rights including the right to vote on members of the company’s board. (Bank Employee, Project Introduction Meeting 2 2015)

3.2.2 Bonds

Bonds are debt securities. The owner of a bond is entitled to the repayment of the face value of the Bond. Bond repayments may be in regular installments or in a lump sum payment.
But regular interest is also payable as the bond is really a renting of money, so the interest is the rent payment. Interest accumulation methods are also specified for bonds, including annually or semi-annually, quarterly, monthly or any other time periods that was offered when the bond was issued.

3.2.3 Mutual Funds

Mutual funds are portfolios comprised of a few or many different securities. When buying a mutual fund, investors, in effect, pool their money with other investors. Mutual funds are maintained by managers, who select specific securities based on specific portfolio strategies. But they are also compensated from the value of the fund for such management work.

3.2.4 Exchange Traded Funds (ETFs)

An ETF, or exchange traded fund, is a marketable security that tracks an index, a commodity, bonds, or a basket of assets like an index fund. ETFs are differentiated from mutual funds as ETFs are traded like common stocks on a stock exchange and net asset values are not calculated at the EOD.

3.2.5 Listed Options

Listed options, also known as exchange-traded options, are standardized options contracts that are traded on exchanges. Underlying securities include stocks, ETFs, indexes and futures.

3.2.6 Principal Protected Notes

A principal protected note (PPN) is a fixed-income security that guarantees protection of the investor’s initial investment. In PPNs, a guarantor guarantees that the investor receive the principal amount upon maturity. An income is possible if the market moves in its favour.
3.2.7 Foreign Securities

Foreign securities are securities denominated or expressed in foreign currencies. The Bank natively trades securities in Canadian and U.S. dollars. Bank rules specify that securities traded on the Tokyo and London stock exchanges are eligible for margin. Securities traded on other exchanges are not granted margin, though they are eligible for trading.

3.2.8 Precious Metals Certificates

Major Banks allow the purchase of certificates of ownership of precious metals: gold, silver and platinum. These certificates allow investors to invest in precious metals without the need to safely store or transport the metals.

Precious metals certificates are fairly liquid, with banks such as RBC claiming that certificates can be converted to equivalent gold or cash at any time.

3.2.9 Options

Options are securities that give the buyer the right, but not the obligation, to buy or sell an underlying security at a specified price within a specified time.

Table 2. Option Types

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call</td>
<td>A call option gives the holder the right to buy an asset at a certain price within a period of time.</td>
</tr>
<tr>
<td>Put</td>
<td>A put option gives the holder the right to sell an asset at a certain price within a period of time.</td>
</tr>
</tbody>
</table>

3.2.10 Strips

Strips are created when bonds have their coupons separated from the bond itself so we have a stripped bond and the coupons. The bonds are marketed as zero coupon bonds and
discounted when sold to derive a market yield. The coupons represent a sequence of cash flows which are discounted to offer the market yield for this investment.

Cost and timing make strips desirable to certain investors when compared to bonds. Strips offer an alternative investment opportunity to those with limited capital. The coupons require a lower capital cost than the bond, and provide a regular cash flow. The zero coupon bond provides a lump sum payment at a greater capital cost.

3.3 Orders
3.3.1 Order Parameters

During trading, there are certain parameters that are required. These parameters are consistent throughout the industry.

**Table 3. Order Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Underlying Asset</td>
<td>The underlying asset is the item upon which the instrument and action is based on. This is usually denoted by the exchange symbol for the asset.</td>
</tr>
<tr>
<td>Financial Instrument</td>
<td>The financial instrument is the type of contract that is being taken against the underlying asset. It can be classified by the type of asset such as the type of instrument for example stocks, options, futures, funds, etc.</td>
</tr>
<tr>
<td>Buy/Sell</td>
<td>This parameter describes the action being taken on the asset.</td>
</tr>
<tr>
<td>Price</td>
<td>The price is the value per unit of the transaction being taken.</td>
</tr>
<tr>
<td>Timing</td>
<td>The timing of the order sets conditions on when to cancel the order.</td>
</tr>
<tr>
<td>Fill Type</td>
<td>The fill type describes how the order is be filled.</td>
</tr>
<tr>
<td>Order Type</td>
<td>The type of order describes how the order is executed and any post-execution or cancellation steps.</td>
</tr>
</tbody>
</table>
3.3.2 Fill Type

The fill type determines whether an order must be fully or partially complete to be accepted.

Table 4. Fill Types

<table>
<thead>
<tr>
<th>Fill Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-Or-Nothing</td>
<td>In this case, the entire order must be filled or the trade will not settle.</td>
</tr>
<tr>
<td>Partial</td>
<td>In this case, a partial fill is accepted and the trade will be executed for as many assets as possible till the asset count is reached.</td>
</tr>
</tbody>
</table>

3.3.3 Order Type

3.3.3.1 Market Orders

Market orders are the most basic trade order. Market orders result in buying or selling at the best price that is currently available as determined by the market. These orders are most likely to get filled but do not guarantee the price.

3.3.3.2 Limit Orders

Limit orders result in buying or selling at a specified price, the limit price, or better. Buy limit orders are executed at the specified limit price or lower. Sell limit orders are executed at the limit price or greater. These orders provide a price guarantee within the tolerance as specified by the limit price but do not guarantee the order being filled.

3.3.3.3 Stop Orders

Stop orders result in buying or selling once a specified price, the stop price, is reached. Upon reaching the stop price, stop orders then function as either limit or market orders. This order provides an additional layer of control of the execution price but further reduces the likelihood of the order being filled.
3.3.3.4 Conditional Orders

Conditional orders are trade orders that are automatically submitted or cancelled if criteria are met. Multiple conditional orders can be combined to create a trading strategy.

Table 5. Conditional Order Types

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Sends Another</td>
<td>In this case, when one order is filled, it automatically sends another order(s).</td>
</tr>
<tr>
<td>One Cancels Another</td>
<td>In this case, when an order is filled, it automatically cancels another order(s).</td>
</tr>
</tbody>
</table>

3.4 Regulatory Environment

The Investment Industry Regulatory Organization of Canada (IIROC) is the national self-regulatory organization which oversees all investment dealers, in this case the Bank, and trading activity on debt and equity marketplaces in Canada. The rules established by IIROC dictate the conduct of companies and client capabilities with the goal of enforcing integrity, transparency and fairness. (Investment Industry Regulatory Organization of Canada 2015)

3.4.1 IIROC Rule 100 – Margin Requirements

IIROC Rule 100 is a regulation for margin requirements and is specifically required for the solution. This regulation specifies how buying power is calculated; specifically by listing limits for what can be used for margin and how much margin can be allocated. The regulation also covers how hedging and offsetting strategies can be implemented. Due to the Bank’s policies, only a subset of the IIROC Rule 100 applies. (Investment Industry Regulatory Organization of Canada 2015)
4 Problem Overview

The Bank has identified opportunities to increase its competitive advantage and reduce risk by adopting a real-time capital calculation engine in place of their current batch processing system.

Another vendor currently performs the functions to be performed by the new potential Vendor. The current vendor maintains the official Book of Record (BOR) and the real-time balances including cash transactions. Through various services, these cash balances are received and returned, intraday for use by the Bank. On a once-per-day basis, after hours, the current vendor receives all transactions within accounts and the BOR is updated to reflect the changes. The Bank reconciles their numbers of buying power, margin and position with the BOR to ensure no discrepancies.

The Bank has provided a list of functional and non-functional requirements on which to evaluate the solution.

4.1 Opportunities Arising from the Current Process

4.1.1 Non-Real-Time Solution

The largest gap from the current vendor is that their service does not have real time functionality. Transactions are processed on an overnight basis after the end of day. This results in the system not updating throughout the business day to reflect market conditions and changes within the account and holdings. In certain cases, a client may place orders for which they do not have sufficient funds due to the lag in the system. This current solution cannot be scaled to real time due to the inefficiency of the calculation algorithm and the age and inflexibility of the software. By implementing a real-time solution, the resulting financial risk can be reduced.
4.1.2 Margin Optimization

The Bank indicated that the capital calculation optimization, which is performed by the current vendor, is not done optimally. As a result, the required margin for clients is greater than should be, effectively reducing buying power. It is the goal of the Bank to provide the largest buying power to the client that is within regulations, house rules and their risk tolerance. By using a more optimal calculation, client buying power can be maximized while remaining within the Bank’s risk tolerance boundaries. By providing the maximum buying power, the Bank provides more value to their clients and results in a competitive advantage.

4.2 System Requirements

This section summarizes the full system requirements provided by the Bank into the different categories in which they are organized.

4.2.1 Functional Requirements

The functional requirements of the project are grouped based on the business operation which they help achieve. Though each of the business operations that shape the requirements are different, there are similarities between the functions required by each operation, such as the need for employee-authorized exceptions for both margin concentration requirements and trading restrictions placed on an account. Below is a summary of the functional requirements specific to each grouping.

4.2.1.1 Regulatory Requirements

The solution must comply with all regulatory requirements. The IIROC regulatory body dictates the functionality of the solution, which must conform to standards set by IIROC Rule 100. The rules that are pertinent to the solution are a specific subset of IIROC Rule 100 due to the Bank’s business practices or due to outsourcing of these regulatory functions to where the Book of
Record (BOR) is located. In the event that business practices do not require the implementation of IIROC Rule 100 sub-rules, the Bank reduces risk by not allowing the client actions that these rules dictate.

4.2.1.2 Margin Requirements
The solution must be able to implement IIROC margin rates as dictated by IIROC rule 100. Hedging and offsetting rules must be restricted to what is allowed by IIROC and in certain cases, more conservative rules set by the Bank. Margin requirements for options must be acceptable according to IIROC rules.

Margin requirements must be adjustable in order to implement the house rules and rates for all calculations. These requirements must be adjustable by type of security, specific securities, price ranges, strategies or contracts. Additional functionality includes the ability to update margin rates intraday. The solution must also reconcile to the end of day (EOD) margin rates with BOR rates.

4.2.1.3 Real-Time Functionality
The solution must run in real time, and in some cases, run at regular intervals intra-day. This real time functionality must be able to initiate capital calculations to report buying power in response to trading activity, market conditions and changes, client activity such as withdrawals and deposits, price changes and rule updates by the Bank.

4.2.1.4 Buying Power Calculations
The solution must accurately calculate a client’s buying power, incorporating a variety of factors. These calculations must be compliant with all policies and house rules as specified by the Bank (Bank Employee, Project Introduction Meeting 2 2015). The solution must be able to find an optimal pairing of positions in order to maximize buying power while limiting the financial risk.
The buying power of a client must also be able to be manually overridden by authorized employees in order to implement trades on an exclusive basis.

4.2.1.5 Trading Rules and Restrictions

Client trading must be regulated by this system to meet the requirements of the Bank. Trading restrictions must be able to be overridden by authorized employees based on user permissions. Each specific restriction must be able to be stacked with other individual restrictions, combining the effects of multiple restrictions. Restrictions may be implemented based on bulk criteria such as account criteria and order criteria, as well as individual criteria such as per client, transaction or order. Restrictions effect client permissions and capabilities including, but not limited to, types of trades, values of trades and the securities involved in trades.

In addition to implementation of rules and restrictions, the solution must also possess the ability to bypass or override restrictions by Bank employees.

4.2.1.6 Options

Options must be able to be adjusted in a variety of ways and must be handled in compliance with the Bank’s business practices. Option levels must be reconciled overnight with BOR and adjusted intra-day. The Bank practices identify a separate option level, which accounts for options for which investment advisors and authorized employees have made an exception.

4.2.1.7 Concentration Maintenance

The solution must be able to implement concentration limits and margin requirements in accordance to the Bank’s business practices and functional requirements. Concentration limits must take into consideration, related parameters such as loan rate of securities, security type and currency. The Bank must be able to implement exceptions to concentration limits for individual accounts for specific securities or by loan rate and security type within the account.
4.2.1.8 Deposits and Withdrawals
The solution must be able to include withdrawals, deposits and respective cancellations to the accounts in real-time. Deposited funds must also be identified as cleared or uncleared to account for uncleared funds trading.

4.2.1.9 Guarantee/Guarantor Relationships
As the Bank allows the linking of multiple accounts, the solution must be able to implement solution functions, treating multiple linked accounts to appear as one account or to only one individual account among a group of linked accounts. The Bank requires the solution to be able to consider multiple linked accounts as if one account for buying power, concentration limits, exceptions, margin requirements and restrictions.

4.2.1.10 Reporting
Data must be accessible to download throughout the day in real-time.

4.2.2 Non-Functional Requirements
The non-functional requirements dictate what the solution is, not its functions or the method through which it accomplishes its functions. Below is a summary of the non-functional requirements as were provided by the Bank (Bank Employee, Non-Functional Requirements 2015).

4.2.2.1 Engine Design
The Bank has requested the solution be designed as an engine. This design involves multiple software components that remain static until called upon by another module, in this case, in response to a request sent by the Bank. Once activated, the software performs its function. In this case, the solution should be listening to a data stream and upon reading specific sequences perform all necessary functions and return a usable output to the Bank.
4.2.2.2 Event Messaging Based Platform

The solution must be on event message based architecture. The solution must identify relevant messages from a bus, read the messages and create the appropriate outputs as events. The outputs must be useable by the solution as well as other consumer systems within the Bank.

The format of the messages must be determined based on the messaging framework and the event message producer and consumer systems within the Bank which will interact with the solution.

4.2.2.3 User Interface

The review and approval process requires that the solution have a process for identifying items that require review and a process for the Bank to review them and manually take the appropriate action. In order to complete this, a user interface is required. Through the interface, the Bank can review flagged items and take action. In addition, the Bank can query client and account information, as well as make adjustments where necessary.

Based on the intended use of the dashboard, we have identified a number of key functions which require a user interface:

1) View items of interest which are identified by the system
2) Monitor clients and account information
3) Manually enter orders
4) Query account information
5) Manually initiate recalculation of buying power and margin
6) Approve and reject items under review
7) Initiate or remove restrictions
8) Adjust buying power and margin parameters globally and by account
9) Implement exceptions in an account

Based on the identified functions, the associated user groups are:

- Investment advisors
- Risk advisors
- Supervisors
- IT personnel

4.2.2.4 Scaling

Due to prospective future demands which may be placed on the system, the system must be capable of upgrades to increase performance. The Bank has shown interest in horizontal scaling options as well as grid computing options to improve performance in the future due to increasing client base and related increases in computing loads (Bank Employee, Non-Functional Requirements 2015).

4.3 Stakeholders

By examining the stakeholders of the project, we can better understand their perspectives and values, allowing us to better evaluate the solution and its feasibility.

4.3.1 The Bank

The Bank is the most important stakeholder in this potential solution. The Bank has initiated the project through their request for proposals. The Bank is a major Canadian bank with millions of clients, tens of thousands of employees and hundreds of billions of dollars in assets.

From this solution, the Bank is aiming to reduce their risks and better improve their services to their clients. Beyond the specified requirements, the Bank values ease of use, high speed, high security, robustness, high reliability all at a low cost.
4.3.2 Bank Clients
The Bank has millions of clients within Canada who will use this service. The clients are the most important stakeholder in the Bank’s operations.

The Bank’s clients value high standards of service, high reliability and the accuracy of the calculation.

4.3.3 Vendor
The Vendor is a multi-national technology company whose expertise is centered on capital adequacy calculations. Currently, they service multiple institutions with a post-execution capital adequacy calculation but do not handle nearly the volume of clients that would be serviced through the Bank. They have recently created a pre-execution capital adequacy solution which they are planning to adapt to meet the needs of the Bank.

4.4 Problem Assumptions
To determine a solution, assumptions are used:

It is assumed that all the required inputs for the correct function are available. Inputs include:

- Client info
- Account information
  - Account ID
  - Holdings
  - Cash
  - Positions
  - Open orders
  - Unsettled transactions
  - Withdrawals and deposits
• Securities pricing
• Margin information
  o Margin rates
  o Concentration limits
• Exchange rates
• Commission and fee structure
• Tax information

It is assumed that the output buying power and margin requirement numbers to be received by the Bank as event messages in a messaging standard of their choosing. The Vendor will be in charge of ensuring the capability of receiving and outputting messages in the specified format.
5 Introduction to Vendor’s Solution

After developing the requirements for the solution, an understanding of the Vendor’s systems architecture, performance and operation was built. By establishing the facts, the literature review was tailored to the solution requirements. By focusing the literature review, we were able to develop knowledge about the solution from sources external to the Vendor.

The Vendor is a leading provider of capital and credit data management solutions, including capital adequacy, regulatory reporting, and client and portfolio margin calculations, cross asset margining and real-time credit monitoring. The Vendor has recently created a pre-execution capital calculation product which is the proposed solution for the Bank.

![Vendor Overall Architecture Diagram]

Figure 3. Vendor Overall Architecture Diagram

5.1 Required Inputs

The proposed solution functions as an engine which is triggered by a number of events including client activity events, portfolio fluctuation events, market events and time deadlines.
5.1.1 Client Information

Client information inputs are described as data used to accurately identify a client of the Bank. By identifying the customer, the Vendor is able to correctly identify account holdings and select rules and permissions which apply to the customer for the purposes of the requested calculation.

Table 6. Client Information Inputs

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Identifier</td>
<td>This customer identifier identifies the Bank as the Vendor’s customer. This information allows the Vendor to identify and segregate data provided by exchanged with the Bank.</td>
</tr>
<tr>
<td>Client Information</td>
<td>The client information must contain all pertinent information to accurately identify a client. This client information must include a unique identifier, in order to identify and segregate data exchanged with the Bank, specific to each client.</td>
</tr>
<tr>
<td>Account Information</td>
<td>The account information must be provided in a format that allows the Vendor to uniquely identify the account, information and account permissions.</td>
</tr>
<tr>
<td>Branch</td>
<td>The branch identifier allows the Vendor to segregate accounts by branch.</td>
</tr>
</tbody>
</table>

5.1.2 Order Information

Order information inputs are classified as data used to perform a capital calculation due to an order placement. These inputs are not necessary to perform the calculation but are required in order to produce an order-initiated output such as an approval, rejection or drop-for-review for an order. These data are essentially order parameters. Client information is also required to perform an order-initiated calculation.
### Table 7. Order Information Inputs

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security code</td>
<td>The security identifier identifies the asset in a transaction and applies the correct permissions, pricing and calculations.</td>
</tr>
<tr>
<td>Order type</td>
<td>The order type specifies the action taken and the calculations that need to be performed to determine acceptance or rejection of the order.</td>
</tr>
<tr>
<td>Quantity</td>
<td>The quantity is the number of units of the security being transacted.</td>
</tr>
<tr>
<td>Price</td>
<td>The price is the monetary value per security for the transaction. This input can also be listed as market value, which can be provided by the Vendor.</td>
</tr>
<tr>
<td>Duration</td>
<td>The duration is the time for which the order will remain active without being filled.</td>
</tr>
<tr>
<td>Fill type</td>
<td>This input determines how the order must be filled. Possible iterations of this input include all or nothing or partial fill.</td>
</tr>
</tbody>
</table>

#### 5.1.3 Security Data

Securities data is described as information related to the management of actions involving all marginable securities. This includes information such as margin or loan rates, security classifications and concentration limits. This data provides the Vendor with the necessary information to perform accurate capital calculations in accordance with house rules. Upon the first transfer of security data, Vendor must modify their security master to represent the Bank’s policies and operations. Should there be any updates to the Bank’s policies, the updated security data must be communicated by the Bank to the Vendor.

#### 5.2 Business Rule Engine Design

The Vendor’s systems function as a business rule engine. A Business Rules Engine is a Microsoft .NET-compliant class library. The framework allows the implementation of readable, declarative, semantically rich rules to business objects, XML documents or database tables.
Business logics, such as those used by the Bank, are transformed into rules which are implemented on information contained in the system. (Microsoft 2016)

Figure 4. Business Rules Framework Architecture (Microsoft 2016)

The design framework promotes code reuse. This is preferred for the nature of the Bank’s project due to the iterative nature of the rules: for each client a subset of all the possible rules are implemented with parameters specific to that client. In many cases the subsets are the same but parameters are changed. This framework is also suitable as house rules limit the trading and hedging strategies used by Bank clients, which are a subset of all possible strategies possible under the IIROC set of strategies. The Vendor already possesses these strategies within their rule engine, reducing the time required to implement the system with the limitations of house rules.
Table 8 below illustrates this. Concentration limits are the same rule applied to different clients and different securities criteria up to a limit depending on the client and security. For all possible concentration limit rules, the same subset applies to clients 0001 and 0002 but with different parameters.

**Table 8. Rule Engine Scenarios**

<table>
<thead>
<tr>
<th>Client</th>
<th>Stock Valuation (CAD)</th>
<th>Concentration Limit (CAD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>&lt; 5.00</td>
<td>100,000</td>
</tr>
<tr>
<td>0001</td>
<td>&gt;= 5.00</td>
<td>500,000</td>
</tr>
<tr>
<td>0002</td>
<td>&lt; 5.00</td>
<td>150,000</td>
</tr>
<tr>
<td>0002</td>
<td>&gt;= 5.00</td>
<td>1,000,000</td>
</tr>
<tr>
<td>0003</td>
<td>&lt; 5.00</td>
<td>150,000</td>
</tr>
<tr>
<td>0003</td>
<td>&gt;= 5.00</td>
<td>1,000,000</td>
</tr>
</tbody>
</table>

The Vendor’s specific rule engine is a web-service provided by Microsoft. This specific rule engine uses a forward-chaining inference engine to execute rules, meaning that an available set of data is used in rules to extract more data until an end goal is reached.

Rule engines do not impose on the architecture or design of business applications. This means that integration with the rule engine does not require changes to the Bank’s systems.

5.2.1 Business Rule Engine Key Terms

**Facts**

Facts are discrete pieces of information. Facts can be .NET objects, XML documents or database row sets.
Rules

A business rule is a declarative statement that governs the conduct of a business process. Rules consist of a condition and resulting action. Actions are only executed if the condition evaluates as true. Conditions are Boolean expressions that are applied to information accessed by the system. Actions are the functional consequences of the condition evaluation.

Policy

A policy is a logical grouping of rules. Policies can be made in the Business Rule Composer by constructing rules. Policies are grouped to pertain to a specific business domain or depending on the application of the policy. Policies can be classified in versions, and when stored, any given version of the policy can be implemented.

Vocabulary

Rule engines vocabulary includes the terms used to define conditions and actions. The benefit of rule engines is that vocabularies can be tailored to be industry or domain-specific, based on the technology artifacts (objects, database tables and XML documents).

5.2.2 Business Rule Engine Components

A rule engine is a production system, defined by four components: a rule repository, a rule editor, a database and an execution core (Microsoft 2016).

A rule repository is an unordered compilation of rules, written as if-then statements, where certain set of conditions result in a set of actions. The rules define initial conditions and the executable action(s) for each initial condition. It is from this repository from which rules are sourced for execution.
The business rule editor is a front-end application and user interface which allows users to define, design, document and edit business rules for storage and execution. This interface is usually designed for business users without a programming background.

The database serves as data storage for elements which are used by the engine. In the case of this problem, the database would consist of definitions, client data and financial data, with which calculations are made and the rule engine executes rules.

The execution core runs selected rules. The core functions by matching initial conditions to applicable rules, resolving any conflicts and running the rule agenda.

5.2.3 Condition Evaluation and Execution

Business rule engines have three stage algorithms for policy execution: Match, Conflict Resolution and Action. (Microsoft 2016)

Matching

Facts are matched against predicates (initial conditions) that use fact type, using the predicates defined in the rule conditions. Rules are selected from the rule repository when initial conditions match those specified in the rules.

In this matching process, pattern matching occurs over all the rules in the policy and conditions that are shared across rules are matched only once. By using pattern matching, the system can pre-determine the set of rules that the initial conditions meet. Partial condition matches can be stored to expedite future pattern-matching and rule policy operations.
Conflict Resolution

In the conflict resolution stage, all rules that are candidates for execution are evaluated to determine the set of rule actions to execute based on a predetermined scheme. Conflict resolution can also be based on rule priorities within policies, which can be edited in the Business Rule Composer. A greater priority number translates to a higher priority for the rule, causing it to be run before those of lower priority. At the end of this stage, the rule agenda is finalized for execution.

Action

In the action stage, the rule agenda is executed in a batch. Rules can assert new facts into the engine, resulting in another cycle of evaluation and execution, but cannot pre-empt the current batch of rules. The action stage results in new facts upon which more rules are executed or the final result is output.

5.2.4 Applicability to the Bank’s Solution

Due to the nature of the Bank’s application requirements, a solution that implements a rule engine seems appropriate. All the business logic that determines the Bank’s processes can be transformed into business rules. For each individual client and transaction, the appropriate set of rules can be extracted from the repository to apply the correct permissions and perform the correct calculations.

The Vendor already has all hedging and trading strategies permitted by IIROC in a rule repository. Any subset of these rules can be implemented to match the Bank’s house rules.

The front-end for rule creation and editing is oriented towards business users who have limited computer science knowledge. This allows business rules to be created with business vocabulary,
allowing business users to create, edit and modify rules as well as their execution with limited or no programming knowledge.

5.3 Message-Oriented Middleware

The Vendor’s systems comprise of multiple hardware and software components. In order to communicate with one-another, the systems all operate on a proprietary event message protocol. The message format is a proprietary version of the Financial Information Exchange (FIX) protocol, which has been modified for integration with the Vendor’s business terms, definitions and systems. With the exception of values with a RAW data type, which is unconverted or is not encoded data, all data is transmitted as text and encoded using a 7-bit ASCII character set.

5.3.1 Financial Information Exchange Protocol

The Financial Information Exchange (FIX) protocol is a non-proprietary, free and open-source standardized language used by global financial markets to communicate trade information. The protocol consists of messaging specifications to be used in trade communications. FIX protocol is maintained by member firms which includes financial institutions which update the standard to meet emerging requirements. (FIX Protocol Ltd 2016)

5.3.2 Message Based Platform

Message based platforms build on event platforms by including the ability to route the events to specific recipients. In the case of event based platforms, events are published globally and all systems within the pipeline can read the data. In message events, the recipients are specified and transferred to them only. The capability for global publishing is retained.

Messages are packages of data which are made up of zero or more parts used for communication between computer systems in a variety of applications. The format of the message is depended on the protocol which it follows, in our case, FIX. One of the parts must be defined as a body,
which consists of binary data which can represent an XML document, a flat file, a serialized .NET class or other binary stream of data. The body is used to identify the type of the message that can be used for routing.

Messages are not editable. Once a message is constructed, it cannot be changed. Should a change be required, a new message is created with the change and it is used from that point onwards.

5.3.2.1 Message Structure

Messages used for communication between technology components follow a certain structure in order to correctly identify, route and understand the content. The general structure of a message includes the header, the body and the trailer. Each message is made of tag=value pairs which define the information and its type contained within the message.

5.3.2.1.1 Header

The header is the first component in the message and is considered the message context. It contains a variety of data which helps identify and decipher the rest of the message. In many cases it includes information to identify the sender, parameters regarding the body such as length and tags to identify the message type. Table 9 shows a number of possible tags that may be used in the header of a vendor’s solution.
Table 9. Header Tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Field Name</th>
<th>Type</th>
<th>Mandatory (Yes or No)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Begin String</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>Body Length</td>
<td>Int</td>
<td>Y</td>
</tr>
<tr>
<td>3</td>
<td>Message Type</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>Sender ID</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>5</td>
<td>Target ID</td>
<td>String</td>
<td>Y</td>
</tr>
<tr>
<td>41</td>
<td>Sender SubID</td>
<td>Int</td>
<td>N</td>
</tr>
<tr>
<td>51</td>
<td>Target SubID</td>
<td>Int</td>
<td>N</td>
</tr>
<tr>
<td>60</td>
<td>Secure Data Length</td>
<td>Int</td>
<td>N</td>
</tr>
<tr>
<td>61</td>
<td>Secure Data</td>
<td>Raw Data</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>Possible Duplicate Flag</td>
<td>Boolean</td>
<td>N</td>
</tr>
<tr>
<td>80</td>
<td>Sending Time</td>
<td>UTC Time</td>
<td>Y</td>
</tr>
<tr>
<td>81</td>
<td>Original Sending Time</td>
<td>UTC Time</td>
<td>N</td>
</tr>
<tr>
<td>90</td>
<td>Xml Data Length</td>
<td>Int</td>
<td>N</td>
</tr>
<tr>
<td>91</td>
<td>Xml Data</td>
<td>Raw Data</td>
<td>N</td>
</tr>
<tr>
<td>100</td>
<td>Raw Data Length</td>
<td>Int</td>
<td>N</td>
</tr>
<tr>
<td>101</td>
<td>Raw Data</td>
<td>Raw Data</td>
<td>N</td>
</tr>
</tbody>
</table>

5.3.2.1.2 Body

The body contains the majority of the information in the message. The contents of the message body can be anything that a sender wishes to communicate. Bodies often have substructures depending on the messaging protocol which is followed. In financial industries, the standard information protocol is the FIX protocol.

5.3.2.1.3 Trailer

The trailer is a non-essential component of the message. The trailer simply concludes the message and is often used to indicate the communication of a follow-up message, as is the case with message bodies that are too long for one message, resulting in the communication of two
messages. Table 10 shows a number of possible tags that may be used in the trailer of a vendor’s solution.

**Table 10. Trailer Tags**

<table>
<thead>
<tr>
<th>Tag</th>
<th>Field Name</th>
<th>Type</th>
<th>Mandatory (Yes or No)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>Signature Length</td>
<td>Int</td>
<td>N</td>
</tr>
<tr>
<td>201</td>
<td>Signature</td>
<td>Raw Data</td>
<td>N</td>
</tr>
<tr>
<td>202</td>
<td>Check Sum</td>
<td>Int</td>
<td>Y</td>
</tr>
</tbody>
</table>

5.3.2.2 Message Processing

Messages are processed in a pipeline as they are received and as they are published. The processing allows for the transformation of formats and verification. The pipeline also provides a formalized procedure for receiving and understanding or building and publishing messages. In message processing, maps can be applied to messages in order to transform the message into a different format suitable for the receiving system.

5.3.2.2.1 Receiving Messages

The adapter initiates the process of receiving messages by reading a data stream and creating the message. The message and its context are created and put through the receive pipeline. Within the pipeline, the message is decoded, disassembled and validated. After being processed by the pipeline, the message is published.

5.3.2.2.2 Sending Messages

When a message is ready to be sent, it undergoes a process similar to receiving messages. Any maps required by the receiver are applied to the message, which allow the message to be transformed to an application or customer-specific format. The message is pre-assembled, forming the different message components. It is then assembled, encoded and published through the transmit adapter.
5.4 Message Switch

The message switch serves as the router through which all information exchanged between the Vendor systems and the Bank systems. A message listener waits for incoming messages which are sent from the Bank. By analyzing the message header, body and trailer, it performs work queuing and routing functions to the appropriate consumer systems. Figure 5 illustrates the high level process flow for the message switch.

The message hub uses both base routing and dynamic routing. COMPIDs serve as unique identifiers for different components within the solution. The different COMPID routes are learned by analysis of message traffic. Static routes are not supported.

Messages are routed using a store and forward algorithm. Residual data and partial messages are buffered until the entire message arrives and framing can be verified. In the case that there are multiple routes for a COMPID, then the message is forwarded once in each route. In the event that there is no route for a COMPID, then the message is broadcast in all connections excluding the connection through which it was received and any isolated connections.

Figure 5 illustrates the design of the message switch. As shown by figure 6, the engine listens, queues events and establishes connection sessions, through which messages are routed.
Figure 5. Message Switch Diagram
5.5 Market Feed

By default, the Vendor implements a pricing engine based off of trades handled in-house. The Vendor consolidates pricing from trades processed by their systems and uses these trades to update their ticker. This pricing system has some discrepancy from actual market prices but has shown a negligible effect on the outcome of capital adequacy calculations.

The Vendor possesses the capability to implement market feeds such as a 20 minute delayed market feed or a real-time market feed for asset pricing. The implementation of these feeds would require the Bank to relay their feed or it can be sourced by the Vendor at an increased cost for the service.

Figure 7 illustrates the function of the market feed system. What’s important to note is that a feed, internal or external, updates a number of databases with pricing data. These databases are accessed and used by the engines to perform calculations or to query data.
Figure 7. Market Feed Diagram
5.6  Security Master
The security master serves as the database of all securities and assets and pertinent information such as margin rates, concentration limits and price thresholds.

5.7  Database
The Vendor has multiple databases which store a variety of data for their services and for specific clients. Data sources include external sources such as exchanges and feeds, clients and internal sources such as results, security and administration data. The Vendor stores data on in-house servers to improve speed and efficiency. Table 11 lists and describes all databases utilized and maintained by the vendor.

5.8  Query Engine
The query engine handles all requests for data. Sessions are opened in response to messages that are routed by the message hub. This results in verification of the message and sender including security, opening a work task, gathering data from the databases and consolidating it into a report and output it through a message. For reports required regularly, report queries can be saved in the report database and accessed through the query engine.

Figure 8 illustrates the design of the query engine. The engine listens for events and creates sessions where requests are handled. The requests are verified and the data is collected. The session accumulates the data and the reply is sent through the engine to the recipient.
### Table 11. Database Table

<table>
<thead>
<tr>
<th>Database</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account</td>
<td>The account database contains information about customers and account structures, including accounts, clients, branches, advisors, account links and designated clients, such as those being monitored or with non-standard alerts.</td>
</tr>
<tr>
<td>Decision</td>
<td>The decision database contains information about securities, control center tasks, and rules. This database also contains data for the current processing day.</td>
</tr>
<tr>
<td>Finance</td>
<td>The finance database contains information about transactions and positions.</td>
</tr>
<tr>
<td>Import</td>
<td>The import database contains staging tables and procedures used for ETL (extract, transform and load) routines. These routines are used when incoming data is too large for normal tools.</td>
</tr>
<tr>
<td>Price History</td>
<td>The price history database contains information about the valuations of securities.</td>
</tr>
<tr>
<td>Report</td>
<td>The report database contains data describing the results of rule calculations and the queries to extract them in different ways.</td>
</tr>
<tr>
<td>System Administration</td>
<td>The system administration database contains information of status codes and alarms used within the system.</td>
</tr>
<tr>
<td>System Security</td>
<td>The system security database contains information of users and respective permissions.</td>
</tr>
<tr>
<td>Test Data</td>
<td>The test data database is used for system testing and related administrative activities.</td>
</tr>
</tbody>
</table>
Figure 8. Query Engine Diagram
5.9 Outputs

5.9.1 Buying Power and Breakdown

This output is a client-facing “buying power” value, which can be used by the Bank to show the client the value of non-marginable securities that they can purchase with their current account holdings and open positions. This value is a result of buying power equations provided by the Bank for the type of account.

For analysis by the Bank, a full capital-adequacy calculation is provided based on account holdings, margin requirements and acceptable hedging strategies. A breakdown of the full buying power value can be provided including hedging strategies used and risk reduced by hedging pairs to complete the calculation.

5.9.2 Approval/Rejection/Review Message

This output is provided when a calculation is initiated by an order and it is in a message format and includes an action and in the case of rejection or review, an explanation for non-acceptance.

5.10 Solution Architecture

In determining the solution architecture, we first established an understanding of the current system and the components involved from the Bank’s systems.

The Bank's clients use front-end applications to access the trading platform and place orders. The orders are made available through the order API which is accessed by the order management system (OMS). The OMS communicates with the exchange, where orders are routed and executed. Information is received from the exchange such as order status, any exchange-side rejections and fill status.
The architecture options stem from the placement of the solution systems and interfaces with the above-mentioned components.

The solution architecture was evaluated based on non-functional criteria provided to the Researcher by representatives from the Bank.

Two possible solution architectures were evaluated for feasibility and suitability for the Bank’s needs: a pipeline architecture and a supervisor architecture.

5.10.1 Pipeline Architecture
In the pipeline architecture, the solution systems serve as an intermediary between the Bank and the exchanges, receiving, evaluating and forwarding orders. In the supervisory architecture, the solution receives inputs and provides outputs to the Bank based on received transactions, where the Bank must take action based on the outputs by communicating rejection, dropping for review or routing the order to the exchange.

A pipeline architecture was explored. In this solution architecture, the solution would serve as an intermediary in the pipeline for orders from the Bank to the exchanges. This architecture provides the following benefits when compared to the supervisor system:

- Lower latency
- Fewer implementation requirements. The Bank does not need to expand the functionality of their system to understand and utilize outputs sent by the Vendor.

5.10.2 Supervisor Architecture
A supervisor system architecture was also explored. In this solution architecture, the Vendor’s and Bank's systems are in constant communication. As the Vendor's system receives calculation
requests from the Bank, it produces an output which is communicated back to the Bank. This architecture provides the following benefits when compared to the pipeline system:

- Reduced reliance on solution systems. All order routing and reception of information from the exchange is controlled by the Bank.
- The system is more reliable. Should there be a failure in communication lines to the Vendor, orders can still be routed to the exchange. In the event there is an error in the Bank’s communication line with exchanges, the Vendor possesses the capability to route orders.
- The process can be expedited. Should an order be expedited in order to gain a competitive advantage, the path from the client to the exchange is shorter and less time-intensive in the supervisory architecture.

5.10.3 Comparison

Through discussion with the Bank, the supervisor system architecture was chosen for further exploration (Bank Employee, Buying Power Engine Architecture Review 2015). The supervisory system architecture seems appropriate for the solution. The two possible and evaluated architectures were a pipeline and a supervisory architecture.

Within the supervisory architecture, multiple sub-architectures were also considered.

5.10.4 API Supervisor

The API Supervisor architecture proposes that the Vendor’s systems communicate directly with the Bank’s order API. As all order management is handled by the order management system, it was deemed that outputs should not be communicated to the API directly from the Vendor, making this sub-architecture unsuitable.
5.10.5 OMS Supervisor

The OMS supervisor architecture proposes the Vendor’s systems communicate directly with the Bank’s OMS in a similar fashion to the API supervisor architecture. Though feasible, this solution is not favourable as order information must be routed through the OMS before reaching the solution, increasing the time required from order placement to execution.

5.10.6 Hybrid Supervisor

The hybrid supervisor architecture, as seen in figure 9, proposes that inputs be communicated to the Vendor from the Order API and outputs be communicated to the OMS. This solution provides the most comprehensive performance as well as the interfacing required. The information is communicated directly to the solution from the API instead of being routed through the OMS, resulting in quicker results. This solution also communicates outputs to the OMS, which is the system which utilizes outputs. Due to these factors, the hybrid supervisor architecture seems most appropriate for implementation by the Bank.

Figure 9. Hybrid Supervisor Solution
6 Research Topics and Findings

Throughout the process, literature was reviewed in order to understand concepts related to the design, operation and methods for evaluating the system.

6.1 Message-Based Middleware Performance

From a functional perspective, suitability of the solution takes into account performance under high message loads. In the Bank’s scenario, a large number of users use systems, which requires frequent communication between the Bank’s and Vendor’s computer systems. With the steadily increasing adoption of algorithmic trading, there are more and more cases of high-frequency trading instances, putting greater loads on technology infrastructure.

Middleware is a term used for software that serves to connect different software and hardware together. In the Vendor’s case, the middleware is facilitated through the use of the message switch.

In the literature review, middleware performance models were reviewed to develop different test scenarios and understand factors that affect solution performance.

Messaging channels resemble service queues, where messages are parsed one by one at a speed determined by the processing power. As messages are received, they are addressed by the system rapidly, but one at a time. In this scenario, the latency depends on the size of the queue and ability to address each individual message quickly. The average resource demand at the bottleneck determines the achievable throughput (Marek, et al. 2014). Regardless of individual capabilities of the system, the bottleneck determines the ability to process events. The accumulated effects of queuing at the messaging channels dominate observed latencies (Marek, et al. 2014). In a complex system, messages are repeatedly queued for consumption or parsing by
different components. The repeated queuing dominates the observed latency from start to finish in a processing cycle (Marek, et al. 2014).

Fitted performance models will need to be developed after implementation. Preliminary models can predict performance but during implementation, a number of miniscule effects which are difficult to predict accumulate and may degrade performance such as environmental factors and physical orientation of the hardware (Marek, et al. 2014).

The ability for a system to handle a message workload can be measured using a set message throughput in two possible configurations: regular and burst. In the regular configuration, an individual message is sent at set intervals. In the burst configuration, a group of messages are sent together at larger time intervals so that overall throughput per time in the study remains the same. It has been observed that for a set number of message queues, message bursts result in significantly more latency. Martinec et al (2014) measured latency distribution in a test system, clearly demonstrating the effect of message burst workloads in comparison with regular workloads as seen in figure 10.
6.2 Real-Time Functionality

The real-time functionality of the engine is based on the ability to perform calculations on-demand and on a continuous basis in response to initiating triggers. The feasibility of a constantly running system is not favourable due to the energy and power demands to run the system.
To create an application that functions in real-time, a reactive computing framework can be adopted. In a reactive computing environment, the system responds to requests on-demand in response to initiating events. Initiating events can vary in their nature. A system component must constantly be active to respond to any new events. When the new event is raised, the computation invokes the rest of the system and the required computations are performed.

Time deadlines can be implemented to raise events in the case of low usage or activity to prevent age-related obsolescence of data.

### 6.3 Vendor Performance

The performance of the vendor’s solution was tested using code written and compiled in a development environment using the vendor’s hardware and proprietary software. In all test cases, when multiple calculations are performed in the same test, the same transaction is being used. This adds extra stress to the system as the transactions are in contention in the database to access data, theoretically increasing latency.

The test environment poses the ideal environment and will produce better results than those found in a production environment, due to some factors. In production, calculations are in contention with one-another and must queue in order to gather data from other resources such as securities databases. Calculation requests are also prioritized in the production environment, which will lead to latency increases for certain calculation requests and decreasing latency for others. As discussed in section 6.1, many physical factors also play a role in determining latency which cannot be accounted for in a development environment such as ambient temperature or hardware setup.
6.3.1 Test Scenarios

To evaluate the vendor’s system performance, two test cases were used. In the first scenario, 20 requests were sent in succession, manually at evenly spaced intervals of .5 seconds. In the second test scenario, 1000 messages were sent to the system as fast as they could be generated once the test was initiated. Tests scenarios were created in order to compare performance observed with the trends described in section 6.1.

![Testing Performance Graph](image)

**Figure 11. Test Scenario Message vs Time Graph**

6.3.2 Production Data Sample

Actual performance data was provided by the vendor to aid the analysis of the solution. Data logs contain incoming messages as calculation requests and outgoing messages indicating the completion of the calculation.

When observing trends for incoming 3 days which were random selected within a period of 1 month, the prevailing trend is that over the period of each day, the vendor receives over 1 million calculation requests in a burst type sequence. Figure 12 is a visualization based on a direct view of the user interface found in the Vendor’s performance evaluation software. The display
indicates that messages are received at intervals with inactive periods between the intervals. The impact of this finding is that the burst type message testing findings will be more likely to be representative of the values in a production environment.

In production, there is no timestamp on outgoing messages which indicate a calculation was completed and data transmitted to the vendor’s client. For tests in the development environment, the code was added to create this feature. Due to the lack of a timestamp on outgoing messages, reliable latency calculations are unavailable. One aggregate sample was available of average calculation latency, which was 42 milliseconds.

![Message over Time (Production Environment)](image)

**Figure 12. Production Data Messages vs Time Scenario**

6.3.3 Test Results

Results of the tests were received as a log file with incoming messages and outgoing messages. The latency was determined by using the identification tag for each individual calculation request and the timestamp on the outgoing message which indicates that the calculation is completed and the data was sent out to the vendor’s client. The timestamp is in a numerical format which indicates a point in time in units of the processor clock. The difference between the
two timestamps was divided by the processor clock speed in order to determine the latency in seconds.

For the single message request the average time of calculation was 24.856 milliseconds.

For the burst type message requests, the average time of calculation was 136.2871 milliseconds.

![Transaction vs Latency](image)

**Figure 13. Transaction Latency Data**

6.3.4 Discussion

Figure 13 plots the latency of each message in the sequence that the message was received. We see an initial near-zero latency which is the observed latency of the single message. This is followed by a switch to a burst type message sequence which results in a step change to higher latency which decreases to a local minimum, from which latency then increases. The latency then experiences a step change to near-zero latency which concluded the test. The near-zero observed latency is not zero but very small due to the immediate consumption and resultant computation of the calculation without queuing.
There is a significant increase in the latency figure between single and burst type calculations which is to be expected. Compared to the Marek et. al study, the latency experienced in the vendor’s solution has a much narrower spread between single and burst type configurations. On the low end, the latency is higher in the vendor’s solution. This is most likely attributable to the complexity of the calculation. In the study, only the transport of messages was considered, whereas in our testing, a rule engine was used to perform a calculation. For the burst type configuration, the latency was much lower in the vendor’s system. This is possibly due to the optimization of the vendor’s solution to meet high throughput low-latency industrial applications and a greater processing capacity compared to the setup used in the Mark et al. study.

When comparing test data to production data, the average latency is lower in production as shown by the daily average we obtained. This is attributable to two factors: contention for resources and complexity of calculations. In the test used by the vendor, the same transaction is executed a specified number of times. For the test, access to the database becomes the bottleneck as only so many connections are possible to specific tables. Input messages are processed but are queued at the calculation stage as data cannot be accessed right away, resulting in latency. The test used by the vendor uses a combination of currencies, derivatives as well as underlying assets for each transaction which is more complex than the transactions that the majority of layman investors execute. This results in a greater number of calculations performed by the rule engine, resulting in a greater time from input to output.

6.4 Interface

The Bank has indicated a preference for a streamlined dashboard interface, facilitated through Representational State Transfer (REST) methodology (Bank Employee, Non-Functional Requirements 2015).
Two types of user interface solutions are available for software as a service solutions: an in-house, Bank built approach and a Vendor built approach.

An in-house REST approach involves deeper connectivity and access of Bank and Vendor systems. Data is accessed in Vendor-side systems by Bank-side software and hardware. Data is read and representations are then created, altered or consumed as necessary. Any changes to the representations are then communicated back to the Vendor’s systems where the data is edited to reflect the change. For this approach, the Bank must build the necessary software and hardware in order to manage the user-interface, flow of data and computing load. This also requires that the Vendor have the ability to facilitate a web-service.

A Vendor-built approach involves the implementation of Vendor’s software in the Bank's computer systems. Data is accessed and manipulated through software applications on the Bank employee’s desktop which connects to the Vendor’s systems.

6.4.1 Representational State Transfer

The web is generally used by humans to access resources remotely through their browser. Representational State Transfer (REST) is an architecture in which machines communicate and resources are manipulated remotely using HTTP (Vinoski 2008).

![Figure 14. REST Architecture](image-url)
As illustrated in figure 14, in REST, data is copied from its source to a remote server. This copy of the data is known as a representation. The representations are used and manipulated as necessary, with the changes relayed and reflected to the data storage. This communication happens in milliseconds, resulting in a seamless experience for the user. The service provided by the Vendor acts as the backend to a client-developed front-end application.

In order to adopt the REST architecture, there are certain constraints that must be met (Vinoski 2008):

- **Client-Server**: this constraint requires the separation of service and consumer logic, as shown in figure 15. The service must listen for requests from the client, which invokes a capability with a request message.

![Figure 15. Client-Server Separation (Microsoft 2016)](image)

- **Stateless**: this constraint requires that all request messages contain all of the information necessary to understand the request. Client-side session state is unavailable to the server and the service cannot retain state data related to the consumer instance.

- **Cache**: Service outputs must be explicitly labeled as cacheable or non-cacheable, which will result in its storage for later requests. An example is illustrated in figure 16.
Uniform Interface: this constraint states that all services and service consumers must share a single, overarching technical interface.

Layered System: this constraint requires that systems be set up in layers, where applications only directly communicate with the next adjacent layer. A client sends a request message to the next application layer, which may be an intermediary or the service.

Code-On-Demand: this optional constraint requires that servers be able to customize or extend client functionality by the transfer of executable code. Consumer architectures require an execution environment to implement this constraint.
7 Vendor System Evaluation

Using the information from the foregoing chapters, the system was evaluated against the requirements specified by the Bank. Requirements were checked and evaluated one-by-one. The findings of the evaluation are summarized according to the groupings in which the Bank provided them. The summaries for each group include the Bank's requirements, Vendor capabilities and a gap analysis.

7.1 Regulatory Requirements

The Bank requires that the solution comply with all regulatory requirements as set by IIROC. IIROC dictates how the system performs the capital adequacy solution. IIROC Rule 100 sets out all regulations with regards to margin and margin accounts, which the Bank must adhere to. Due to house rules and account limitations, there are subsets of IIROC Rule 100 which apply. The system must possess the ability to implement these rules as required.

The Vendor’s capabilities allow their offered solution to implement all required regulatory controls, applying regulations for both IIROC and any subset of rules from IIROC rule 100. All house rules are checked for compliance with IIROC when implemented. The Vendor updates rules according to regulatory changes and a check for compliance is done.

The Vendor's systems adequately meet all the Bank’s current regulatory requirements and possess the capabilities to meet possible future requirements and changes.

7.2 Margin Requirements

The solution must be able to implement and apply IIROC and house margin rules as applicable. Hedging and offsetting rules should allow maximization of client buying power while complying with regulations and house rules. Margin requirements for options should be acceptable
according to IIROC rules. Additional functionality requires that margin requirements be adjustable intraday by the Bank by any desired singular or grouped classification.

The Vendor’s solution has the capabilities to meet all regulatory requirements including audit functions. IIROC rules are fully implementable and house-rules are cross-referenced with regulations to ensure compliance with regulations, including margin rates, hedging and strategies.

The solution has the ability to implement all the Bank House Rules, including margin rates, calculations and exceptions. For regulatory compliance, margin rates are cross-referenced with regulatory requirements to ensure compliance.

Margin rates can be adjusted intraday, initiated by authorized Bank users or by the Vendor upon receipt of written communication from an authorized Bank user. Margin requirements can be adjusted based on bulk criteria or individual criteria, globally or individually. Turnaround time can vary from immediately to a few hours depending on workload and priority.

The Vendor currently has a system in place to reconcile end of day data with the book of record, already proven by other work performed for the Bank.

The Vendor’s capabilities extend beyond the functionality required by the Bank. The Bank’s policies limit the hedging and offsetting practices to a subset of what is acceptable under IIROC. Additionally, the Bank may use parameters for these practices that are more conservative than those allowed under IIROC. The Vendor would have to, and is able to, limit their hedging algorithms and adjust the calculations based on the applicable Bank policies.
7.3 Real Time Functionality

The solution has to run in real time and at regular intervals intra-day when uninitiated by the Bank. This real time functionality must perform capital calculations in response to triggers that are external to the Vendor, such as trading activity, market conditions, client activity and rule updates by the Bank. Additionally, the calculations must be performed in accordance with time deadlines.

The Vendor’s systems currently operate in real time and possess the capabilities to perform pre-execution calculations for transactions. Buying power calculations can be initiated by client or Bank activity, pre-determined initiators such as time deadlines and account changes, or external factors, such as market events or price changes in account holdings. Action-triggered recalculation such as orders, account changes and requests can be specified by the Bank. Passive-triggered recalculation such as market events and time intervals can be set specified by the Bank by either mass data edit or by specific account or client.

The solution currently meets the requirements by the Bank related to real-time functionality.

7.4 Buying Power Calculations

The solution must accurately calculate a client’s buying power. These calculations must be compliant with all Bank and regulatory policies. Where possible, the solution should use hedging and offsetting strategies approved by both IIROC and the Bank to maximize buying power while limiting risk. The implications of calculation results should be able to be manually overridden by authorized Bank employees to execute trades on an exclusive basis.

Buying power calculations are this Vendor’s specialty. Specific house rules can be implemented to ensure only house approved strategies are used. As with other considerations, calculations are
checked to ensure compliance to regulatory standards. Calculations and protocols can be adjusted to create exceptions for accounts or clients by authorized Bank users or by the Vendor's users upon receipt of written requests.

The solution currently meets the functionality required by the Bank related to the calculation of buying power and capital adequacy calculations.

7.5 Trading Rules and Restrictions
Client trading should be regulated by this system, in accordance with the Bank’s policies. When multiple restrictions apply, the effects should stack. Restrictions should be able to be implemented based on bulk or individual criteria. Trading restrictions should be able to be overridden by authorized Bank employees. The solution should also possess the ability to bypass or override restrictions by authorized Bank employees.

Under the supervisor architecture, client trading is not directly regulated by this system but provides the necessary outputs for the Bank’s OMS to regulate client trading by accepting, rejecting or dropping orders for review. The Vendor has the ability to implement any number of rules and restrictions within their systems. The architecture of the current systems can accommodate specific subsets of restrictions to apply to different clients based on bulk or individual criteria. All trade restrictions are considered when performing calculations and producing an output. All restrictions that were specified in the project document are possible. Overrides can be implemented by the Bank by authorized users or by the Vendor through written requests. The supervisor architecture can allow the Bank to directly bypass buying power calculations by allowing the OMS to route orders to the exchange regardless of approval by the Vendor's systems.
The Vendor possesses the capability to implement the rules and restrictions with some additional work. A full list of rules, regulations and respective conditions must be supplied to the Vendor for implementation.

7.6 Options

The Bank requires that options should be able to be adjusted and handled in compliance with its business practices. Option approval levels must be reconciled overnight with the Book of Record (BOR) and adjusted intra-day. The solution should be able to implement custom options levels with exceptions introduced by the Bank.

The Vendor possesses the ability to implement any set of business rules to securities based on type, including options and their levels. The Vendor also has the ability to reconcile data with the BOR on an overnight basis.

The Vendor’s systems conform to the Bank’s options requirements. For implementation, the specific rules and levels have to be supplied.

7.7 Concentration Maintenance

The Bank requires that the solution must be able to implement concentration limits and margin requirements in accordance to the Bank’s business practices. Concentration limits should be taken into consideration; related parameters such as loan rate of securities; and security type and currency. The Bank should be able to implement exceptions to concentration limits by bulk and individual criteria.

The Vendor’s systems possess the capabilities to implement concentration limits and margin requirements. All used margin and concentration values are maintained in the security master and updated according to market changes or when initiated by the Bank. Exceptions can be
implemented using bulk or individual criteria. Updates and changes to maintenance concentrations are provided by the Bank and implemented through the Vendor. Exceptions by security, security type, account and/or client can be implemented by authorized users or by the Vendor upon receipt of a written request from an authorized user.

The Vendor’s systems meet the capabilities required by the Bank. For implementation, an initial update is required to set all values for concentrations and margins to create the Bank’s specific security master. An update process should be created for future updates.

7.8 Deposits and Withdrawals

The solution should be able to include withdrawals, deposits and respective cancellations in real-time. Deposited funds should be identified as cleared or uncleared to account for uncleared funds trading.

The Vendor’s systems are able to accept deposits and withdrawals for client accounts. Cleared and uncleared funds can be segregated. Enhanced functionality includes client and holdings based exceptions. For example, high net-worth clients with deposits can be accepted as usable funds as acceptance is almost certain.

The solution possesses the capabilities to meet the Bank’s needs. For implementation, specific details should be discussed and a connection to the BOR should be established.

7.9 Guarantee/Guarantor Relationships

As the Bank allows the linking of multiple accounts, the solution has to link multiple accounts according to the Bank’s business practices. The solution must be able to reconcile differing parameters among linked accounts. Specific to the Bank’s business practices, guarantee-
guarantor relationships are either two-way or one-way. For linked accounts, the pool of accounts should be counted as one account for calculating concentration limits.

The Vendor’s systems analyze guarantee/guarantor relationships after an initial calculation. Guarantee processing is done after an account calculation is completed. Guarantee-guarantor relationships are only one-way, so a two-way linkage is considered as two one-way linkages. In the event the guaranteed account is offside, the guarantor’s account(s) are evaluated one-by-one until the deficiency is covered. In the event of a loop, calculations are considered completed.

The solution meets only some of the capabilities required by the Bank. Though the Vendor is able to link accounts, the relationships used for calculations are only one-way compared to the combination of one way and two-way relationships required by the Bank. For implementation, the guarantee/guarantor relationships used by the Vendor would require detailed specifications for account linkages and such capability would have to be developed.

7.10 Reporting

Data should be available to download throughout the day in real time.

The Vendor’s systems allow data access through client applications which is accessible through authorized user access and automated queries can be initiated for reports commonly used within the financial services industry.

The solution meets the reporting and data accessibility requirements required by the Bank. For implementation, users need to be authorized and the applications set up on their devices.
7.11 Engine Design

The Bank has indicated the solution should be designed as an engine. The solution must function in a runtime environment. Upon request, it has to perform the appropriate functions and return an output to the Bank.

The Vendor’s systems function as an engine and polls for requests in the message pipeline.

The solution meets this design criteria outlined by the Bank. For implementation, an information link will need to be established between the Vendor and the Bank.

7.12 Event Messaging Based Platform

The solution should be on an event message based architecture accepting inputs and publishing outputs through messages in a format determined by the Bank. Currently, the Bank has not specified the format of the messages that will be used to communicate internally or externally.

The Vendor’s systems are built on a message based platform. The Vendor uses a proprietary version of the Financial Information Exchange (FIX) format internally but has indicated that they can implement a method to accept any form of message that the Bank specifies.

The solution can meet the Bank’s requirements but cannot be fully implemented until the Bank issues specifications for their messaging formats. An information link will need to be established for message communication.

7.13 User Interface

The solution should have a user-interface that meets the requirements specified by the Bank. It should allow for the identification and review of flagged accounts and transactions and accepts inputs from the users in response. The user-interface must also allow users to query data.
The Vendor’s systems interface with customer users through proprietary client applications. Through these applications, users can access and modify both systems and data. Exceptions can be made through these features.

The Bank has specified its preference for a dashboard, through which its users can interface with the solution. This is not ideal as the Vendor’s applications are sufficient in fulfilling the functions accomplished by a dashboard. A dashboard would also require development and add a layer of complexity to implementation and future updates to the Vendor’s systems and applications updates can adversely affect the dashboard. The Vendor and the solution do not possess this capability. The Vendor has indicated an interest in creating this capability but will require substantial work and cost.

As shown in figure 17, the dashboard interface combines functions that would normally be provided through different applications in a unified interface. This unified interface is desired to streamline the operation of the many functions of the solution and minimize the learning required by bank employees. The example interface shows a securities pricing ticker, a feed populated by event types the user subscribes to, a feed of flagged items for review and an area which is populated by information of selected flagged items, through which, the user might approve, reject or escalate items.
Figure 17. Dashboard Interface Organization (top) and mockup (bottom)
7.14 Scaling

The solution should be scalable using horizontal scaling or grid computing to improve performance.

The Vendor’s systems operate in a high throughput low-latency environment and can be scaled to accommodate a higher throughput.

The solution meets any scaling requirements which may be implemented in the future.
8 Conclusions and Recommendations

As Banks compete in a highly competitive marketplace and technology evolves, the exploration of feasibility of implementing technologies continues to play a crucial role for the advancement of the industry. The proprietary nature of cutting edge technology for financial applications has made the analysis of this Vendor solution a unique and informative experience.

8.1 Conclusions

In this thesis, we examined the needs of one of Canada’s largest financial institutions and evaluated a Vendor’s solution to meet this need. The functional requirements presented by the Bank were explored with consideration given to the business processes that motivated them. The Vendor’s proposed solution was evaluated for its ability to meet these, either by fulfilling the function directly or by providing another avenue of accomplishing the business process. Where the solution did not meet the requirement, the gap was identified and explored.

The proposed solution meets most of the requirements set forth by the Bank. The solution does not meet three requirements specified by the Bank: the guarantee guarantor account relationships, the user interface and the message protocol.

The Bank has specified the requirement of two-way relationships, whereas the Vendor handles guarantee-guarantor relationships as a combination of one-way relationships. Functionally, the result is the same. The result that the bank wishes to achieve is accomplished by using a combination of two one-way relationships. Due to the functionality, it was concluded that this is not a reason for a negative recommendation of the Vendor.

The Bank has specified the requirement to adhere to a messaging protocol but as the Bank has not provided a messaging protocol as of yet and the vendor operates on a proprietary
protocol; this needs to be resolved at a later stage of the negotiations. Through no fault of the Vendor or their solution, this need was not met but the capability does exist once the protocol is provided. Due to this, it was concluded that this is not a reason for a negative recommendation of the Vendor’s solution.

The Bank indicated its preference to implement an in-house uniform interface when accessing the system. The Vendor currently does not possess the capability to implement a web-service through which the Bank can access the system to implement a RESTful interface. Currently, the Vendor provides a client application, through which their clients change settings and access data. In this case, though hesitant, the Vendor has indicated the ability to produce a custom-built solution. In this case, the process for implementation is lengthy and costly. This major issue led us to conclude that the implementation of the Vendor’s solution may not be the most ideal solution.

8.2 Recommendations

First and foremost, this study recommends that other vendors be considered as solution providers and their solutions be evaluated on the same needs, methodology and criteria. By evaluating other vendors, the ideal solution may be identified, where there is a current ability to implement a web service without custom-development, saving time and money in the long run. As stated, no other vendor currently provides a real-time solution to the Bank’s problem, it is possible that the solution may be in development, which can only be known by closely working with other possible vendors.

Secondly, this study recommends the evaluated Vendor as a feasible solution provider for the Bank. Their solution adequately meets all major functional requirements declared by the Bank. Compared to the Bank’s current vendor, in all evaluation criteria, the proposed vendor is
as adequate as the current vendor due to the regulatory environment created by IIROC and industry standards such as message based communications following the FIX protocol. The evaluated vendor highly exceeds the Bank’s current vendor in the areas of buying power calculations and real-time functionality. For buying power calculations, the Bank has identified the Vendor provides a more optimized buying power than their current vendor. The real-time functionality of the evaluated vendor far exceeds the capabilities of the current vendor which uses batch processing. Where gaps were initially identified, such as with the inability to facilitate REST interface due to not currently implementing a web API, the Vendor has proposed alternate methods for meeting the functionality, as well as, custom-built capabilities that will meet the needs.
9 Further Work Implementation Steps

This study is not comprehensive enough. Other Vendors and solutions could be evaluated against the same criteria and requirements to compare the adequacy of this solution against all other available solutions. Based on the findings of the multiple studies, the best available solution can be identified.

As the Vendor has indicated an interest in providing a means of allowing the Bank to access its services through a web-service connection which would allow the implementation of a uniform interface, it is recommended that the Vendor’s ability to build this custom solution be explored. Considerations for in-depth analysis for the Vendor’s capability include foremost, the feasibility of the solution against constraints of security, vulnerability and reliability. Then the timeline should be considered, accounting for building any required capabilities and timeline for implementation. Lastly, cost must be analyzed. Though the capability for the vendor to provide a REST API does not exist now, it is worth noting that their current offering is in-line with the current Bank practice of individual applications and interfaces for each individual service, making this solution adequate in the short-term while a uniform interface solution capability is developed.

For implementation, the only issue stems from the missing message format specification but the Bank has indicated it will be available to the Vendor once it has been finalized. If the specification is provided, the capability to interpret and translate Bank and Vendor messaging standards must be built. This is a custom-built solution by the Vendor to understand the incoming message standard and output responses in the Bank’s standard. The majority of identified work required for implementation relate to initiation of the project and applying the
Bank’s rules and business practices. It is reasonable to assume that this work would be required by all Vendors.
Works Cited


—. "Rule 100 - Margin Requirements." 2015.


