

RITSUMEIKAN ASIA PACIFIC UNIVERISTY

UNDERGRADUATE THESIS

---

# Utilizing a Single-Purpose Blockchain to Increase the Efficiency of Settlements in Exchange and Asset Management

---

*Author:*  
Hoon KIM

*Supervisor:*  
Dr. Damon DRUMMOND

*A thesis submitted in fulfillment of the requirements  
for the degree of Bachelor of Business Administration  
in the*

College of International Management

January 17, 2020



RITSUMEIKAN ASIA PACIFIC UNIVERSITY

## *Abstract*

APM

College of International Management

Bachelor of Business Administration

### **Utilizing a Single-Purpose Blockchain to Increase the Efficiency of Settlements in Exchange and Asset Management**

by Hoon KIM

The purpose of this research study is to answer the question of “Can we make the existing digitized settlement systems more efficient and secure for investors and managers?” One of the capabilities that Blockchain technology brings, is that it allows us to simplify the transaction process by removing the involvement of other parties and directly pay someone. This technology has the potential to reshape how traditional settlements work. With this in mind, this paper confirms that by developing a single-purpose blockchain for trading stocks can decrease the number of third-parties involved in a settlement and increase the integrity of the transaction data. Thus, increasing the overall efficiency of the system. The research question is answered via both primary research and secondary research. The results show that a blockchain-based system that is developed with the specifications described in this research study only requires the platform maintainer to ensure that the native token is pegged to another legal tender, as everything else is handled by the system itself. Furthermore, being able to implement a blockchain that can communicate with other blockchains in the stock exchange system can open the door for tokenization and easy transaction of company assets, becoming a supportive infrastructure for asset managers.

Keywords: Blockchain, Stock Exchange, Substrate Framework, Smart Contract, Asset Management



## *Acknowledgements*

It was a journey conceiving this research study, but it was a one that opened my eyes to the wider world. I hope this paper is the start of becoming something big in my life in academia. First and foremost, I want to thank my supervisor, Dr. Drummond Damon and the Damon Seminar members for supporting me both in and out of academics. I could not have started writing if it was not for Dr. Drummond or even be able to continue to enjoy my research. I am happy to be part of your seminar.

I want to thank Dr. John A. Rose for introducing me to the world of Blockchains, the marvels of computer science and supporting me in writing and finishing the thesis. As someone who never had a formal education in engineering, and someone who is majoring in social science, I consider myself very lucky to be able to learn from you.

Furthermore, I want to thank my best friend in APU, Kittidej “Gun” Chirabowornkul for introducing me to the world of stock exchanges and inspiring me to start this research. Finally, I want to thank my parents and everyone who is reading this paper. It may not be too much, but this is now a work that is part of my personal history, and thank you again to all the people that I was blessed to have in my life.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Background	1
1.2 Research Question and Hypothesis	2
1.3 Blockchain	2
1.3.1 Cryptocurrency	2
1.3.2 The Consensus Algorithm	3
1.3.3 Types of Blockchains	3
1.3.4 Applications	4
1.3.5 Current Issues with Blockchains	4
1.3.6 Substrate Blockchain Framework	5
1.4 Investment Management	5
1.5 The Stock Exchange	6
1.5.1 Behind the Scenes	6
1.5.2 Types of Orders	7
1.5.3 Involved Parties	8
1.5.4 Issues and Limitations	8
1.5.5 Blockchain Applications for Securities Exchange	9
1.5.6 Current Issues in Adopting Blockchain for Exchange	9
1.5.7 Why Single-Purpose Blockchain	10
1.5.8 Why Substrate	10
<b>2 Methods</b>	<b>11</b>
2.1 Setting Up the Development Environment	11
2.1.1 The Rust Language	11
2.1.2 Substrate Framework	12
2.2 Substrate Runtime Module Library	12
2.2.1 Substrate Consensus Algorithm	13
2.2.2 Chain Permissions	14
2.3 Implementing the Exchange System	14
2.3.1 Assumptions	15
2.3.2 Data Structure	15
2.3.3 Chain Storage	16
2.3.4 Function Types	18
2.3.5 Filling Orders	18
2.4 Function Development	19
2.4.1 Administrative Functions	19
2.4.2 Minting and Burning Shares	20
2.4.3 Trading and Fees	22
2.4.4 Starting and Testing the Node	22

<b>3</b>	<b>Results</b>	<b>25</b>
3.1	Trading and Escrows . . . . .	25
3.1.1	Placing Orders . . . . .	25
3.1.2	Order Expiration . . . . .	27
3.2	Confirming the Hypothesis . . . . .	29
<b>4</b>	<b>Discussion</b>	<b>31</b>
4.1	Advantages . . . . .	31
4.2	Challenges and Limitations . . . . .	32
4.2.1	Limitations from the Tests . . . . .	32
4.2.2	Privacy Issues . . . . .	33
4.2.3	Government Regulations . . . . .	33
4.3	Implications for Management . . . . .	34
4.4	Future Research . . . . .	35
4.4.1	Code Improvements . . . . .	35
4.4.2	Best Execution, Broker, Market Maker Module . . . . .	35
4.4.3	Primary Market Development . . . . .	36
4.4.4	Zero-Knowledge Proof Cryptography . . . . .	36
<b>5</b>	<b>Conclusion</b>	<b>39</b>
	<b>Bibliography</b>	<b>41</b>



# List of Figures

1.1	The Blockchain Trilemma . . . . .	3
1.2	Public Trading Process . . . . .	7
1.3	Counterparty Relationships . . . . .	8
2.1	Stock Exchange Chain's Trilemma . . . . .	13
2.2	Order Placing Process . . . . .	18
2.3	Order Filling Process . . . . .	19
2.4	Order Execution Process . . . . .	19
2.5	Giving Issue Rights . . . . .	20
2.6	Issuing Shares Function . . . . .	21
2.7	Retiring Shares . . . . .	21
2.8	Substrate Front-end UI . . . . .	23
2.9	Node running . . . . .	23
3.1	Orders Expiring . . . . .	29



# List of Tables

3.1	Initial Trading Value	25
3.2	First Trade Result	26
3.3	Second Order Placement	26
3.4	Second Trade Result	26
3.5	Third Order Placement	27
3.6	Third Trade Result	27
3.7	First Order with Expiration	27
3.8	First Expiration Order Result	28
3.9	Second Order with Expiration	28
3.10	Second Expiration Order Result	28



# List of Abbreviations

<b>ACH</b>	<b>A</b> utomated <b>C</b> learing <b>H</b> ouse
<b>API</b>	<b>A</b> pplication <b>P</b> rogramming <b>I</b> nterface
<b>CEX</b>	<b>C</b> oin <b>E</b> Xchange
<b>Dapps</b>	<b>D</b> ecentralized <b>a</b> pplications
<b>DLT</b>	<b>D</b> istributed <b>L</b> edger <b>T</b> echnology
<b>DMA</b>	<b>D</b> irect <b>M</b> arket <b>A</b> ccess
<b>ICO</b>	<b>I</b> nitial <b>C</b> oin <b>O</b> fferings
<b>OTC</b>	<b>O</b> ver <b>T</b> he <b>C</b> ounter
<b>PoA</b>	<b>P</b> roof of <b>A</b> uthority
<b>PoW</b>	<b>P</b> roof of <b>W</b> ork
<b>PoS</b>	<b>P</b> roof of <b>S</b> take
<b>NPoS</b>	<b>N</b> ominated <b>P</b> roof of <b>S</b> take
<b>RPC</b>	<b>R</b> emote <b>P</b> rocedure <b>C</b> all
<b>SRML</b>	<b>S</b> ubstrate <b>R</b> untime <b>M</b> odule <b>L</b> ibrary
<b>SUDO</b>	<b>S</b> uper <b>U</b> ser <b>d</b> o



*Dedicated to future generations...*





## Chapter 1

# Introduction

The stock market is arguably one of the most important factors for a company to grow and provide opportunities to further innovate (Beattie, 2019). It is natural that the stock exchange also sees innovation, which they did see throughout their 500 years of history. Starting from doing business in a coffee shop, innovating into trading physical slips, followed by the exchange of digital stocks, and the latest innovation, exchange through blockchains (Wall Street, 2018).

This paper is separated into five chapters. This chapter will discuss the background information that is required to understand this paper such as, the basic concepts of blockchain, the components that make a blockchain, basics of investment management and the basic concepts of the stock exchange.

**Chapter 2** presents the methodology of the research. This includes setting up the blockchain development environment, tokenization of stocks in code, the logic of how filling an order will work in the blockchain expressed in flowcharts, administrative functions, and other development-related work.

**Chapter 3** presents the results for the developed exchange platform. This section will talk about the types of tests conducted to confirm that the developed blockchain-based system works as expected through simulating a small number of traders selling and buying a single company's shares.

**Chapter 4** is the discussion section. This section will discuss how the blockchain-based system developed in this research study works than the traditional system, along with its limitations. This section will also talk about some potential risks that a blockchain-based system might face and further improvements required for this project to be production-ready.

Finally, **Chapter 5** presents the conclusion of this research, summarizing each point that was discussed in this paper.

## 1.1 Research Background

Over the past few years, the term **Blockchain** and **Cryptocurrency** was gaining public attention. Naturally, the number of researches on this topic has also increased. The topic that particularly attracted my attention was regarding the possible applications for blockchain technology in the stock market and the application for management, as it seemed to me that this opens up many opportunities in how ownership and settlement of assets of companies can work.

This thesis will adopt a similar approach to the work of Pop et al., 2018, in that, it will develop a fully functioning prototype that fills an order completely on the blockchain and conduct several tests to confirm its functionality. Furthermore, the goal of this paper will be similar to that of the work of Chiu and Koepl, 2018, in that it will aim to determine the feasibility of a blockchain-based settlement. This research study is different in that it will develop an entire blockchain that is designed

specifically for secondary market settlements, rather than developing a **Smart Contract**—a business logic that works within a blockchain’s ecosystem—on Ethereum. The other difference will be that the project developed in this paper will be a consortium blockchain with a **Proof-of-Authority** consensus algorithm, rather than a **Proof-of-Work** consensus, both of which will be discussed in detail later in this paper.

## 1.2 Research Question and Hypothesis

This research study aims to answer the question, "Can we make the existing digitized settlement systems more efficient and secure for investors and managers?"

The hypothesis for this question is that "Developing a single-purpose blockchain for trading stocks can decrease the number of third-parties involved in a settlement and increase the integrity of the transaction data." This research study will develop a blockchain module for the **Substrate Framework** (which will be discussed later in this paper) that will emulate the secondary market in the stock exchange. After the design and development of the system is finished, to confirm my hypothesis, I will test its functionality by simulating few traders each selling and buying stocks of a single company, and checking the resulting numbers of shares owned by each trader and their balance.

The finalized project that was developed in this research study is accessible via the following link: [https://github.com/hoonsubin/exchange\\_platform.git](https://github.com/hoonsubin/exchange_platform.git)

## 1.3 Blockchain

Blockchain, also known as **Distributed Ledger Technology**, is a category of a technology that provides consensus amongst a decentralized system. Please note that the term DLT and blockchain in its technical definition is different, but throughout this paper, they may be used interchangeably as some of the sources for this research study do not distinguish the two.

The blockchain concept was first introduced in 2008 with the Bitcoin White Paper written by Satoshi Nakamoto, 2008. After that, there have been countless innovations and variations of the same core technology. Arguably the most practical function of a blockchain is that regardless of what type of blockchain it is, it provides decentralized consensus in both permissionless or permissioned systems (Cong and He, 2019). Additionally, blockchain provides a data structure that links each data with **hash pointers**, which points to the previous data block. This effectively means that every successive block is "chained" together, allowing the data entry to be immutable, as changing one block will require that party to change the entire entry’s hash pointer which recursively points to the previous block (Bashir, 2018).

### 1.3.1 Cryptocurrency

We cannot talk about blockchain without addressing cryptocurrency.

Cryptocurrency is a newly emerging digital currency that is powered by a blockchain to ensure secure peer-to-peer transactions (Frankenfield, 2019). Think of cryptocurrency as a single type of value that is changed with every account that is stored in the global ledger called the blockchain. It is possible for a single blockchain to contain multiple cryptocurrencies, and it is also possible for a blockchain to have no cryptocurrency at all. The most important aspect of cryptocurrency is that it allows blockchains to incentivize the block validators and allow the tokenization of things

(Bashir, 2018). Having said this, it is my humble opinion that it may be a stretch to call every token in a blockchain a cryptocurrency or digital currency, as some are not meant to be directly traded.

### 1.3.2 The Consensus Algorithm

The consensus algorithm of a blockchain determines how distributed systems with multiple state changes in different times and places can reach a single consensus on which data to keep and which to reject. This is also called **block validation** and the nodes (computers) that perform are called **validators**. This effectively makes it one of the major security systems by which a blockchain can prevent malicious parties from taking control of the data and the major solution for the double-spending problem that was present in other related digital currency systems (Bashir, 2018).

Currently, there are several consensus algorithms that attempt to solve a problem that other consensus models could not. The first blockchain implementation, and perhaps the most popular one, would be the **Proof-of-Work** consensus, also known as the **Nakamoto consensus**, named after the creator of Bitcoin which, was the first blockchain to use this consensus model (Witherspoon, 2017). Other models of consensus include **Proof-of-Stake**, **Delegated Proof-of-Stake**, **Nominated Proof-of-Stake**, **Proof-of-Authority**, **Direct Acyclic Graphs** and more. Every consensus algorithm will have its strengths and weaknesses, hence why there are many types of blockchains. Most of them are developed for a specific task and employ a specific consensus algorithm that is the best fit for its purpose.

Furthermore, the consensus algorithm of a blockchain determines where that ecosystem will lie within the blockchain trilemma, which states that it is very difficult for a consensus algorithm to simultaneously obtain scalability, security, and decentralization. Therefore in most cases, a particular consensus model will involve a tradeoff (Buterin, 2018).

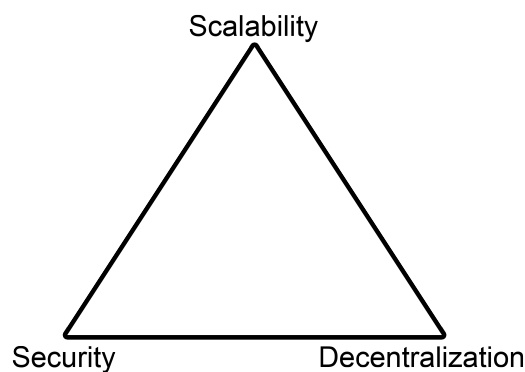


FIGURE 1.1: The trilemma that every blockchain will need to face for each consensus mechanism

### 1.3.3 Types of Blockchains

Countless blockchains have been introduced throughout the years, so trying to discuss all of the models will be impractical. Instead, I will only discuss the general types of blockchains and some of the more famous and innovative projects that were developed over the past decade.

According to Bambara and Allen, 2018, blockchains can be categorized into three different types.

- **Public Blockchains** – A permissionless blockchain where anyone with the right condition can contribute to the consensus to become a block validator, making it nearly impossible to regulate. These include Bitcoin, Ethereum, Zcash and many more.
- **Private Blockchains** – A permissioned blockchain where the block validators are selected by a centralized organization or an individual. Private blockchains are generally considered to be more scalable and fast but are very centralized, making it unusable to the general public.
- **Consortium Blockchains** – These are a combination of both of the above two, where the validators are chosen by a centralized organization, but the data that is stored within the blockchain is made accessible by the general public.

### 1.3.4 Applications

According to Antonopoulos, 2017, a decentralized blockchain that has been operating correctly for a certain time has the characteristics of no double-spending, immutability, neutrality, secure timestamping, authorization, auditability, accounting, non-expiration, integrity, transaction atomicity, discrete units of value, timed transactions, consistency, etc. Many industries that require these characteristics tend to benefit greatly from a blockchain-based system.

As blockchain is considered to be an industry-shaping innovation, many existing businesses are looking into, or have already started to implement the technology in their environment (Clohessy and Acton, 2019). One example is The **Nasdaq** stock exchange, starting to utilize the blockchain technology for their infrastructure from 2015 (Khandelwal, 2018).

A **Smart Contract** is a piece of business logic that is operated on top of a blockchain, given that the blockchain is capable of hosting such programs (Bambara and Allen, 2018). Thanks to this, the applicability of blockchains has increased exponentially, as the immutable nature of blockchain allows these contracts to reduce uncertainties and risks in business with blockchain-related tasks (Kim and Laskowski, 2017). There is also a category of newly emerging applications called **Decentralized Applications**, or most commonly referred to as **Dapps**, which utilize either the smart contract or any other features of the blockchain to tokenize real-life things and provide a related service (Dannen, 2017). We can see that the ecosystem of blockchains has a wide variety of applications that are not only for financial institutions but also for many parts of the government, real estate, trade invoices, advertisement, accounting, auditing, web applications, gaming, etc. (Bashir, 2018).

### 1.3.5 Current Issues with Blockchains

Even with the aforementioned benefits that the blockchain technology provides us, it does have some flaws that the community is working hard to overcome, such as the **scalability problem** which, the biggest public chains like Bitcoin and Ethereum are starting show (Huillet, 2019). The scalability problem of a blockchain, in general, refers to the point at which the limited transaction rate of a public blockchain becomes a bottleneck for the system. Another symptom for a bottleneck is when the transaction fees for payments and smart contracts become exponentially large, to the

point where it is no longer feasible to use it. There are many community efforts to solve this issue, such as the **Layer 2** solutions like the **Plasma Network**, where the transactions are delegated to a child chain, to be handled as their own transactions, and a finalized state is periodically sent to the main chain (or the root chain) via a connection made with a smart contract (Poon and Buterin, 2017).

In addition, there are various regulatory issues that blockchains have, which will be discussed in detail in Section 4.2.3.

### 1.3.6 Substrate Blockchain Framework

**Substrate** is a blockchain development framework written in **Rust**, first introduced as a concept for a heterogeneous multi-chain network blockchain named **Polkadot**, in a white paper written by Dr. Gavin Wood, 2017, who is one of the founders of Ethereum. We can think of substrate as a pre-developed blockchain with different components and an **Application Programming Interface** (API) that complies with a common standard. Polkadot is also developed with the substrate framework as well, meaning that other blockchains developed with the substrate framework will be able to utilize the interoperability with the Polkadot blockchain. The main idea was to develop a system where multiple different blockchains would connect with one another through what they call **Parachains** which can complement each other, rather than compete for more users (Schulz, 2019).

Through interoperability, substrate has solved the scalability issue that was plaguing other blockchains. This is accomplished through the delegation of certain tasks on the blockchain to another parachain. Another advantage in substrate is that it takes a modular approach with the design of its architecture. This allows developers to easily develop a blockchain functionality without having to work from scratch or make drastic changes to the entire system and share their work with other developers. Furthermore, thanks to **Webassembly**, an intermediate binary language that runs on the browser, this allows the software to run with almost no overhead to the system. In addition, this allows substrate to have forkless updates (Schulz, 2019). In the past, the users of a blockchain must manually download a new client in order to apply ongoing changes to the system. This process is also known as a **Hard Fork**. Without this, future transactions will not be accepted by other nodes (Dannen, 2017), but substrate was able to solve it. Overall, substrate is a very scalable blockchain solution that is also flexible in supporting changes to customize its use. Substrate is arguably one of the best frameworks for developing a single-purpose blockchain that is future-proof and easy to develop.

## 1.4 Investment Management

The term **Investment Manager**, sometimes called **Financial Manager**, has a different definition depending on the context it is spoken in, but in general, financial management is a form of management that specializes in managing an organization's or an individual's assets, investment portfolios, investment decisions, equity and such (Kennon, 2019). This type of work is generally categorized as part of corporate finance.

The main role of investment managers is to provide oversight of the client's assets and equity. Unlike financial advisers, investment managers will formulate investment strategies to achieve the client's goals. They also act as financial planners who provide insight on topics like real estate, cash-flow, insurance, risk analysis, and

internal managers. Furthermore, they may make investment decisions on behalf of their company as well (Corporate Finance Institute, 2015). Because most firms rely on debt to operate, being able to understand and manage their capital structure is considered to be important for any business.

While this will be discussed in detail in Section 1.5.4, the current challenge that is facing financial managers, especially in regard to investment management and financial management, is the risk that is involved in exchanging assets such as shares, real estate, and cash (FinTech Futures, 2017).

## 1.5 The Stock Exchange

The stock market has been the backbone for the growth of the global economy for centuries. The history of the stock market is debatable as the concept of trading "notes of future growth" or debt issues was a somewhat common occurrence starting from the 1100s and 1300s (Beattie, 2019). Having said this, it is generally accepted that the modern style of trading ownership of companies started from the late 16th century, famously by the Dutch East India Company (Bramble, 2018). At the time there were no formal locations, or even a name to describe the act of exchanging notes representing ownership of a company. Instead, investors would have to seek a broker to either sell or buy the papers. In England, these brokers would often conduct business in various coffee shops in London. Later in 1773 the **London Stock Exchange** was officially formed and another 19 years later, the world's largest stock exchange, the **New York Stock Exchange** was formed (Beattie, 2019). In 1971, almost two centuries after the world was officially introduced to the concept of the stock market, **NASDAQ** commenced its operations. This was significant as NASDAQ was the world's first exchange to trade securities on a computerized system (Chen, 2019b). After a few decades, the technology has improved exponentially and now the world is seeing **Automated Clearing Houses**, **Direct Market Access**, and electronic trading platforms such as the **Globex**, where trading is less physically involved and more automated, eliminating the need for a trading floor (Simpson, 2019).

### 1.5.1 Behind the Scenes

In general, there are two types of stock markets, the unlisted market, and the listed market. There is a third type that is called the **Primary Market**, which stands in between the unlisted market and the public market (also known as the **Secondary Market**).

The Unlisted Market is a market where investors would trade shares, securities and any other assets that are private companies, or other companies that are not listed in the exchanges like the NYSE or LSE for various reasons. The unlisted market is traded through a process called **Over-The-Counter**. Transactions will take place through the OTC Bulletin Board, which is an electronic quotation service. Additionally, OTC trading is usually done via a broker-dealer network rather than a centralized exchange (Murphy, 2019).

The Listed Market is where the shares of public companies are traded. This market can be divided into the primary market and the secondary market. Keep in mind that the primary market is technically not part of the listed market, but rather a prerequisite for a company's assets to be traded in the listed market. The primary market is known for the sales of newly issued shares for companies looking to go public,



also known as **Initial Public Offering**. This is facilitated by underwriting groups like investment banks, and the primary market is where traders purchase shares directly from the issuing firm (Chen, 2019c). After the IPO, the shares in the market are further traded in the secondary market by other investors or the public companies themselves, though the latter is uncommon. The secondary market is where most of the complicated transactions occur, as there are many parties with many tasks. Additionally, the client may have several options to consider when trading in the secondary market (Hayes, 2019).

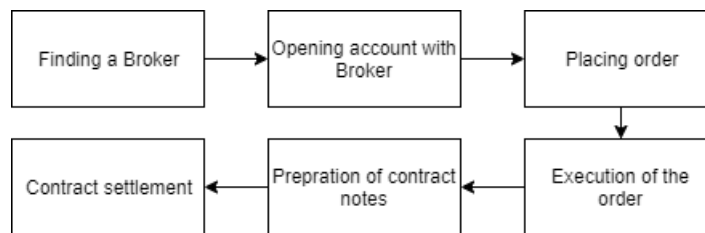


FIGURE 1.2: A simplified representation of the process of how an asset is bought and settled (Aditya Trading Solutions, 2017)

The process of trading can be divided into four main parts.

- Selecting Brokers
- Receiving/Placing Orders
- Executing Orders
- Settlement

Within those processes, there are several parties involved who work together to ensure fast and secure transactions.

### 1.5.2 Types of Orders

There are several types of orders that an investor can request to the broker which allows the clients to have more control over what time and price the assets will be traded, given that the exchange that the investor is trading in allows it. (Aditya Trading Solutions, 2017). In this section, I will give a short explanation of some of the more well-known orders that are supported by most exchanges.

- **Limit Order** – an order in which the investor can choose the minimum or the maximum price point for a particular security. This order is not guaranteed to be executed if there are no orders in the market with the investor's requested price point.
- **Stop Loss Order** – this order requests to buy/sell a particular asset when the market price hits a particular price point.
- **Market Order** – this order simply sells/buys the assets at the current market price.
- **Good Till Day Order** – this order specifies the duration (usually in a number of days) in which the order will remain open for trading and close the order when it hits the requested deadline.

### 1.5.3 Involved Parties

As the public market relies on interconnected processes that span through multiple organizations, it is natural that this process will involve many parties.

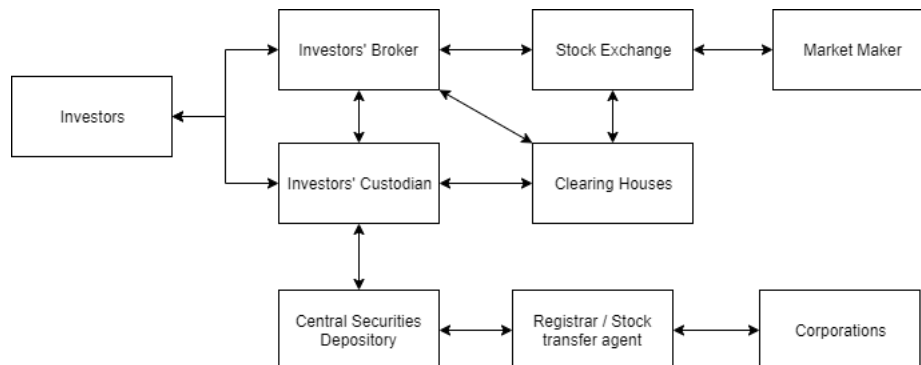


FIGURE 1.3: The relationship of major parties that are involved in trading a company's asset (Brown, 2014)

As we can see in Figure 1.3, excluding the Taxation Authorities, there are around nine types of parties involved in trading assets in a public market. Most organizations excluding the traders and corporations that are involved in this transaction serve the purpose of reducing risk in the transaction, ensure satisfactory orders and provide market liquidity (Brown, 2014). Thanks to recent technological advancements, the process of communicating has been made much more efficient and effective, reducing the risk of human error (Simpson, 2019). Having said this, there are remaining fundamental issues and risks in this process, which I will discuss in the next section.

### 1.5.4 Issues and Limitations

As we have seen in Figure 1.3, there are several parties interconnected with each other when exchanging assets, which may imply several points of failure. Arguably, the greatest risk comes from the **Settlement** process and the **Clearing Houses**.

Clearing House is an intermediary between the buyer and the seller. Their main role is to reduce the risk in transactions from start to finish and to consolidate all transactions leading to settlements. They take the opposite role of the client in each trade, which allows the transaction process to be fast with reduced cost, but they have to pay the cost in default risks as the Clearing Houses are subject to default risk from the counterparties. The solution to this risk is by imposing margin requirements on those parties (Ganti, 2019).

The major issue here is that the integrity of the transaction solely relies on the credit of the counterparty and the margins imposed by the Clearing House. When the Clearing Houses are met with defaulting counterparties, the gap that was made in this process is expected to be filled with the Clearing Houses' own capital (IFCI Risk Institute, 2004), meaning that they are designed to have huge capital backings so big losses are absorbed by their members, who are mostly banks. Considering how these institutions work in a monopoly or duopolies, it is very possible that one huge economic bubble may compromise the entire system, with the operations of the Clearing Houses themselves being the cause of starting and amplifying this problem (Cohn, 2015).



Another issue with the current system from both the Clearing Houses and the Stock Exchange is the risk of either a glitch in the system or a cyber attack. For instance, an infrastructure with a centralized database requires communications from multiple sources, where each source is a point of failure for the entire system. Because of how the network is structured, any error in one of the systems may have the risk of data loss, which has happened in the past (Pisani, 2019).

### 1.5.5 Blockchain Applications for Securities Exchange

Due to these potential issues with the traditional exchange system, in recent years many stock exchanges have started to consider the benefits that a blockchain-based system might bring to the table. For instance, Nasdaq is one of the first stock exchange to document private securities issuance on the blockchain (Briganti and Wells, 2015).

When talking about blockchains and shares, it is important to discuss **Initial Coin Offerings**, more commonly known as ICO. ICO is a new form of fundraising for companies or development teams that are working on a service that operates in some relationship with a blockchain, which could be either a Dapp or a new blockchain. This is done by selling the native token (cryptocurrencies) that can be used in the blockchain that the company or team is working on to the outside world (Frankenfield, 2019). The major difference from an ICO to an IPO is that the blockchain tokens do not necessarily represent the ownership of the company and because of this, there are some problems when viewing tokens as assets for regulatory reasons (Global Law Library of Congress, 2018).

In the context of utilizing the blockchain technology to improve the traditional settlement system, exchanges from different parts of the world are in serious consideration of its implementation, as the benefits that it brings are clear (French, 2018). For instance, in 2018, Nasdaq and Singapore have collaborated in the development of a blockchain-based settlement system (Parsons, 2018). They are looking to efficiently link up funds transfers and tokenized securities transfer to reduce counterparty risks that I have discussed in the previous section.

### 1.5.6 Current Issues in Adopting Blockchain for Exchange

The issue with the current implementation of blockchain technology for settlements is that most of them are focused on implementing the system as a smart contract (Trustnodes, 2018), or any other method that evolves utilizing existing public chains' security and the number of nodes (Seijas, Thompson, and Mcadams, 2016). It is true that being able to use existing public blockchains will provide the system with decentralization, increased security and reduced cost for the infrastructure. However, it will also reduce the degree of control the stock exchange has over transaction fees and transaction speed (Pop et al., 2018). This implies that such a system will lack a consistent incentive for the platform maintainer, as the fees in a smart contract or any other public blockchain will go to the validator, which in this case is an unknown node that is in this world (Chiu and Koepl, 2018).

In this research study, I propose a solution to this issue. By developing a single-purpose blockchain with the substrate framework, it is possible to give enough control and economic incentives to the platform maintainers, but also provide the security benefits and transparency that a blockchain-based system can provide.

### 1.5.7 Why Single-Purpose Blockchain

Most of the blockchains and cryptocurrencies that we hear in the news are called general-purpose public blockchains. Their main purpose is to offer a digital currency solution that is reliable and ubiquitous enough to be used for any transactions on the web or to provide a platform to host smart contracts that can be used for any purpose. S&P Global, 2019 points out that public blockchains have the potential to pose compatibility risks regarding economic incentives for miners and the consensus algorithm. Any software updates that result in a fork is another factor in the risk of adopting a public blockchain. There are existing projects like **Counterparty**, which is a peer-to-peer financial platform that adds a smart contract layer to the Bitcoin blockchain and aims to reduce the counterparty risk in settlements and to tokenize company ownerships (Kelleher, 2014). There is also **Ripple**, which is a settlement system that utilizes blockchain technology to ensure fast, reliable, and low-cost alternative to the traditional system (Cata, 2018). At first, it is easy to think that these blockchain-based services aim to solve the same issue that was described, making them a "Single-Purpose" blockchain. However, the main difference from what this research aims to achieve and their services are that both Counterparty and Ripple are a single type of blockchain that is designed to serve multiple types of settlements. Both Ripple and Counterparty have their own cryptocurrencies—XRP for Ripple and XCP for Counterparty—that is required to exchange within the platform. This has the risk of added liquidity into the system and this may not be optimal for securities exchanges with daily high-frequency transactions. The blockchain that I develop and discuss in this paper is a single-purpose blockchain in that every stock exchange will have its own blockchain with its own cryptocurrency and tokens that are based on the framework developed in this paper. This means that every exchange will have segregated transactions which will prevent the blockchain from being filled with unrelated data (Huillet, 2019), which will increase its scalability. Details will be discussed later in this paper.

### 1.5.8 Why Substrate

As mentioned in Section 1.3.6, the substrate framework is fast and easy to develop complex blockchain projects, and the interoperability that it provides with the Polkadot network and other substrate-based blockchains can benefit the exchange platform with added security and integrity to the system. Additionally, a substrate-based system can prove to be scalable compared to other public blockchains (Schulz, 2019). Lastly, the Substrate framework supports forkless updates to the blockchain. This is done by uploading the WASM binary to the blockchain, which is then executed directly from the blockchain, similar to how a smart contract would operate.

## Chapter 2

# Methods

In this section, I will be going through the development process of the project for the research. The software that will be developed is a simple but effective decentralized application (also known as Dapp) on a semi-public blockchain that emulates a stock exchange system, where investors and traders can directly interact with the order book of exchange and the system itself can handle the settlement as well. This will be completely powered by the blockchain, and will be able to function as a distributed system, meaning there is no single point of failure, and the service can operate without a single governing body, or a broker-dealer, at a minimum execution time.

Due to time constraints for this research, I will be exclusively focusing on implementing the Secondary Market that can only handle Limit Orders. The focus on the Secondary Market was a choice due to the fact that the stock price in the Secondary Market is almost completely determined by the supply and demand of the security, while primary markets require other third parties' evaluation (Kenton, 2019). The reason for choosing to implement a limit order for this project is due to the reason that limit orders do not require an initial market price, as it is completely represented by the traders' perception of the value of the stock (Mitchell, 2019).

## 2.1 Setting Up the Development Environment

In this section, I will describe the process of setting up the development environment for the Substrate Blockchain Framework, starting from downloading the language compiler to running a template node.

### 2.1.1 The Rust Language

The Rust Language is the main programming language that is used to develop the Substrate Blockchain Framework. Rust allows us to develop a system that is memory safe and has less overhead to the system (Klabnik and Nichols, 2019). This is a crucial part of both blockchains and stock exchange platforms. System failures for stock exchanges have the potential of a massive financial loss that cannot be recovered, which may damage the integrity of the system as a whole, causing confusion for all the parties involved (Pisani, 2019).

The Rust compiler can be found from the following link.

<https://www.rust-lang.org/tools/install>

The installation process will be different from Unix-based operating systems like Darwin (MacOS) and Linux from that of Windows. Because I will be developing from a Linux system (Ubuntu 19.10), I will use the following command to install Rust and the required toolchain.

```
$ curl https://sh.rustup.rs -sSf | sh
$ rustup target add wasm32-unknown-unknown --toolchain nightly
```

Note that the Rust compiler and any other dependencies that are required to develop for the Substrate Framework will be automatically downloaded and installed when we install Substrate. This step is only required if you are working on an existing Substrate node on a different computer.

### 2.1.2 Substrate Framework

As mentioned in the previous section, substrate is only a framework for developing a blockchain and not necessarily a full blockchain by itself.

The source code for the entire Substrate Framework can be found on Parity Technologies' GitHub page: <https://github.com/paritytech/substrate>.

For developing a custom Substrate-based blockchain node, we only need to download the template node that I will be using as a starting point of this project. That can be done by executing the following commands on the command line for Linux systems.

```
$ curl https://getsubstrate.io -sSf | bash -s -- --fast
$ substrate-node-new <project title> <author>
```

The first command will install the packages that are required for running a Substrate node (including Rust and git). You may notice the `-fast` flag in the first command. This will skip the `cargo install` stage of the entire Substrate code, and `subkey`, which are not required for this project.

For the second command, it will (at the time of this writing) create a 1.0 stable release version of the node template inside the current working directory that only has the basic runtime modules added, with the provided name (`exchange_platform` in the case of this research study). The Substrate framework is a rapidly growing community project, and will always have changes that may or may not break the existing codebase. So it is recommended that we use the stable release version for production-ready projects.

## 2.2 Substrate Runtime Module Library

The substrate blockchain can be divided into two main components, the blockchain core, and the runtime. The core system of the blockchain includes the hashing algorithm, block number, header, extrinsic data, etc., where the extrinsic data is the piece of information that represents data outside of the blockchain which is recognized by the blockchain, i.e., the runtime. The runtime component of substrate can be thought of as a collection of software that determines how the state of the blockchain will be. This includes the logic of how a transaction can be handled, the logic for the cryptocurrency, smart contract functionality, and much more (Eluard, 2019). Unlike smart contracts, runtime modules give us much more control over the entire blockchain without needing to consider gas fees.

The substrate runtime module library (SRML) is a collection of different pre-made runtime components that can be implemented for substrate-based blockchains' functionality, such as determining what type of data will be stored and how it will be stored. With SRML, it is easy and convenient for any new substrate blockchain developers to add different functionality to their blockchain (Petrowski, 2019). One

example would be to add smart contract functionality to the blockchain by implementing the `srml-contract` module to the blockchain project.

Most developers aiming to develop dapps with the substrate blockchain will be developing on the runtime layer, which is the case for this project (Petrovski, 2019). The code that this paper is working on will be stored in the following directory.

```
runtime/src/secondary_market.rs
```

### 2.2.1 Substrate Consensus Algorithm

Before we discuss the development of this project, I would like to discuss the consensus algorithm that is used for this project.

This project will be using **Aura**, a **Proof of Authority** consensus algorithm that was developed by Parity Technology, the developers behind substrate framework. This was initially implemented as part of the Parity Ethereum client and is a default consensus algorithm for substrate (Barinov, 2018). PoA is somewhat similar to PoS in the sense that it chooses a group of validators to optimize the block time, rather than allowing complete decentralization like PoW, where anyone who solves a cryptographic puzzle the fastest can be a validator. This effectively prevents anyone from spamming data into the blockchain, which is one of the causes of Bitcoin and Ethereum's scalability problem (Curran, 2018). The main difference between PoA and PoS is that PoS chooses block validators (the node that can validate transactions generated from the users) by accounts with the most staked coins. On the other hand, PoA chooses validators by the identity of the group in the network, meaning the blockchain will have pre-selected accounts that are allowed to validate ongoing transactions from the blockchain (Curran, 2018). This means that PoA will be more centralized than PoW or PoS consensus, and because of this, PoA is mostly used for test nodes or closed blockchain applications rather than a public blockchain. However, as the priority of a trading platform is to maintain security and scalability rather than decentralization, this consensus model will work well for this research study (Barinov, 2018).

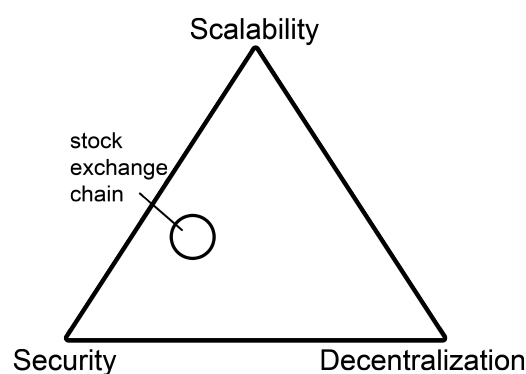


FIGURE 2.1: How a blockchain-based exchange platform with PoA consensus will look represented in a blockchain trilemma. We can see that the system will have to sacrifice decentralization to increase scalability and security.

According to Santo et al., 2016, the consensus algorithms of public blockchains that were mentioned in Section 1.3.2 have issues when implementing the traditional

stock exchange system. Notably, consensus algorithms that require economic incentives to fund the nodes (such as PoW) will affect the overall circulation of the currency within the system, hence giving the currency volatility, given that the exchange market was to use the native currency for trading (Trustnodes, 2018). The ability to handle high traffic volume markets in a stable manner will be another issue, as many blockchains that use the PoW consensus algorithm have inherent scalability issues (Piccolo, 2017). Additionally, other consensus algorithms with a probable finality like PoS and PoW have the potential to damage the integrity of the securities exchange system because of the weakness against cartel attacks (Parity Technology, 2019). With a PoA consensus algorithm, these shortcomings, in the context of utilizing blockchain technology as a stock exchange, can be circumvented. Having said this, there are some technical problems and limitations with PoA, specifically Aura (Aumasson, 2017). This means that the project developed in this paper may be subjected to future updates to address this issue.

The detailed block propagation and the consensus algorithm can be set up in the substrate framework from `src/service.rs` in the source code.

### 2.2.2 Chain Permissions

The traditional trading platforms have special functions that only the platforms can execute, such as closing the market or authorizing firms to issue shares. In order for this project to accurately represent the traditional system, we will be needing a special account that has the permissions of the platform maintainer. This is where the **sudo key** is required.

**sudo** is a keyword that is used in Unix-like computer systems to refer to users with special privileges (i.e. superuser), similar to that of the Administrative permissions from Windows operating systems (Cohen, 2008). In substrate, sudo is a runtime module that is part of the SRML. The main functionality of a sudo module is to store an address that will be marked as the sudo key.

```
decl_storage! {
    trait Store for Module<T: Trait> as Sudo {
        /// The 'AccountId' of the sudo key.
        Key get(key) config(): T::AccountId;
    }
}
```

This is the storage macro of the sudo module that comes with substrate. As we can see from the `config()` keyword from the source code of the sudo module, this data can only be defined from the `GenesisConfig`, which ensures that the sudo key can only be defined during the development phase of the application. Using this ensures integrity in the system by only allowing platform maintainers to execute crucial functions, which no other accounts can. These settings can be adjusted and defined from `src/chain_spec.rs` file inside the source code.

## 2.3 Implementing the Exchange System

In this section, I will be describing the typical structure of a secondary market in a traditional stock exchange platform, and how this will be expressed in Rust in the context of developing the substrate blockchain.



### 2.3.1 Assumptions

Before we continue on, I will have to make several assumptions about this system in order to ensure the functionality is similar to that of real-life exchanges.

**Assumption 1** – The account with the sudo key is the platform maintainer and the owner of the stock exchange that it is trying to represent.

**Assumption 2** – The cryptocurrency that is used in this system is a stable coin that is pegged to the currency of the country that the stock market resides in (for example, if the owner is NASDAQ, then the currency will be in US dollars).

**Assumption 3** – Only the platform owner can mint/burn coins.

**Assumption 4** – Anyone can download and run a node of this blockchain.

**Assumption 5** – At least 2/3 nodes are honest nodes.

**Assumption 6** – The initial balance of each account is traded through the platform maintainer from real cash to cryptocurrency and vice-versa. This is a representation of how much currency a person (entity) owns. This effectively makes the platform maintainer a **Coin Exchange** as well.

**Assumption 7** – All firms (accounts that can issue shares) are publicly traded companies.

Assumptions 1, 2, 3, and 6 are to make sure that the entire system has a stable circulation of currency, and platform maintainers have complete control over any market functions. Assumptions 4 and 5 will ensure that the system is distributed throughout different nodes. This increases both accessibility and transparency of the system, which ensures integrity as even the blockchain maintainer cannot change any entry within the blockchain. Lastly, this project requires Assumption 7 due to the lack of a primary market system built-in to it.

From these assumptions, we can see that this will be a semi-public blockchain, where a single trusted organization (i.e. the stock exchange) has the ability to manage the flow of the currency, but the ability to access the blockchain is available to anyone who wishes to trade, removing the involvement of a broker to access the books.

### 2.3.2 Data Structure

For the order type, this project will only implement the **Limit Buy Orders** and **Limit Sell Order** Good till canceled, with an expiration date (block number in the case for this project) to prevent the list from overflowing with the increase in traders.

In the traditional market, the broker will require the following information for a typical limit order (Mitchell, 2019).

- number of shares
- max (minimum for sell orders) price per share
- name of the company
- expiration date

There may be additional variables in the back-end portion of the system, but for the end-user, these are typically the minimal information that the broker requires in order to register the order (Securities and Exchange Commission, 2013).

For this project, information of a limit order will be represented as a Rust data structure like the following:

```
#[derive(Encode, Decode, Default, Clone, PartialEq)]
#[cfg_attr(feature = "std", derive(Debug))]
pub struct BuyOrder<AccountId, Balance, Hash, BlockNumber> {
    firm: AccountId,
    owner: AccountId,
    max_price: Balance,
    amount: u64,
    order_id: Hash,
    expire: BlockNumber,
}

#[derive(Encode, Decode, Default, Clone, PartialEq)]
#[cfg_attr(feature = "std", derive(Debug))]
pub struct SellOrder<AccountId, Balance, Hash, BlockNumber> {
    firm: AccountId,
    owner: AccountId,
    min_price: Balance,
    amount: u64,
    order_id: Hash,
    expire: BlockNumber,
}
```

The attributes for all the data within the data structure is as follows:

- **firm** – The address of the company that issued the share. This can be thought of as the blockchain equivalent of the stock market symbol.
- **owner** – The address of the trader that registered this order. This is the account where the money/shares will be sent and taken from depending on whether it is a sell order or a buy order.
- **price** – The price for each share the owner is willing to sell/buy. For buy orders, it will be the maximum price, and for sell orders, it will be the minimum price.
- **amount** – The number of shares the owner of the order is willing to buy/sell.
- **order\_id** – A unique ID for the order. This allows the order to be treated as a **non-fungible** coin.
- **expire** – The expiring block number for this order. This is the blockchain equivalent of an expiration date. Orders that are past their expiration block will be canceled and all the locked shares/money will be returned to the owner.

### 2.3.3 Chain Storage

Chain storage is the data that will be stored within the blockchain as part of the extrinsic chain state. Everything that is stored can be accessed by anyone via the RPC command, and any other substrate front-end wrapper so long as it's connected to a full node (Eluard, 2019). This can be thought of as the entry for the immutable ledger. In substrate and Ethereum smart contracts, the relationship between multiple sets of data can be referenced via a **hash map**, where a key can be given to a hash map that will return a value associated with the given key. For instance, we can map the data



type `AccountId` with `Balance`, and when we want to get the value of the `Balance`, we would just provide the key `AccountId`. This will not work the other way where we provide `Balance` to get `AccountId`.

The data that will be stored and read for this project will be as the following:

- Issuer list – a vector of `AccountId`. These will be the addresses of firms with floating shares in the market.
- Floating shares – Hash map of `AccountId` to unsigned 64-bit integer. This will return the number of floating shares for the given firm.
- Is allowed to issue – Hash map of `AccountId` to a boolean value. This will return whether or not the given account is allowed to issue shares, i.e., is the account owned by a publicly-traded firm or not.
- Owned shares – Hash map of (`AccountId`, `AccountId`) to a unsigned 64-bit integer. The key will be a tuple of `AccountIds`, where the first one is the account of the holder of the asset, and the second account is the address of the firm. This will return the number of shares the account has for the firm.
- Locked shares – Hash map of (`AccountId`, `AccountId`) to a unsigned 64-bit integer. This is the escrow storage, where the number of shares that are put out for sale will be stored, and can only be retrieved when the order is satisfied or expired.
- Last bid price – a Hash map of `AccountId` to `Balance`, where the given account is the address of a firm. This value will be updated every time an order is satisfied with a firm. This value can be considered as a market price for that particular block.
- Authorized shares – Hash map of `AccountId` to unsigned 64-bit integer. By giving the address of a firm, it will return the number of authorized shares that the firm has. This is the maximum number of shares the firm can issue, and this value can only be changed by the **sudo** account.
- Market closed – A boolean value. This shows whether or not the market is closed. Before every order, this value will be checked, and the order will be executed only if this value returns false. This value can only be changed by the **sudo** account.
- Sell orders list – a Hash map of `AccountId` to a vector of sell orders. This will return a list of all the sell orders for a given firm account.
- Buy orders list – a Hash map of `AccountId` to a vector of buy orders. This will return a list of all the buy orders for a given firm account.
- Sell orders expiring – Hash map of `BlockNumber` to a vector of sell orders. This will return all the sell orders that are set to expire in the given block number.
- Buy orders expiring – Hash map of `BlockNumber` to a vector of buy orders. This will return all the buy orders that are set to expire in the given block number.
- Nonce – This will return an unsigned 64-bit integer that will increment with every hash generation for the order ID. A nonce is part of the hashed data, and it is used to ensure that the generated hash is truly random.

One thing the reader might have noticed from the storage value is that, from the perspective of the system, both traders' accounts and firms' accounts are of the same type. Technically the firms' accounts are just normal addresses in the blockchain that are allowed to issue shares of their own, buying and selling a share can be done by anyone given that they have the assets to afford it. This allows the system to be flexible and easily manageable by other runtime modules in the substrate blockchain, giving us expandability which the paper discusses in Chapter 4.

### 2.3.4 Function Types

The functions of this secondary market exchange platform will be mainly divided into three parts.

**Administrative functions** – These include functions that only the platform maintainer can execute. Such as giving/revoking accounts the right to issue shares, adjusting authorized shares for a firm, close/open market, and other features that require permission from a trusted third party like the government (Chen, 2018).

**Trading functions** – For this project, the trading function will only support limit sell orders and limit buy orders. The trading function will place an order with the provided information from the trader, and either be saved in the chain storage for it to be filled by later orders or be executed on the spot if the order is already filled.

**Corporation functions** – This will be functions that only publicly traded companies can execute, such as retiring shares or issuing new shares below the authorized share amount.

### 2.3.5 Filling Orders

Placing and filling in the orders are the main parts of trading in the exchange market (Chen, 2019a). Thus, it is important to focus on the algorithm for placing and filling orders, which will have to work all within this blockchain-based system.

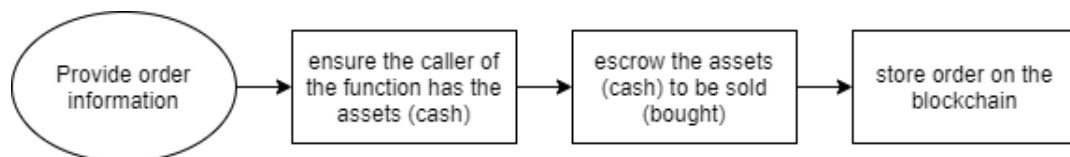


FIGURE 2.2: The high level view of how an order will be placed

Escrowing the shares for sell orders will be done by implementing a custom private function that will store the shares to the `LockedShares<T>` chain storage. Escrowing the trader's currency for buy orders will be done by the built-in **ReservableCurrency**, which will simply flag a certain amount of currency as **reserved**, and ensure that the amount will not be spent until the order is filled or canceled.

For the algorithm for how both sell and buy orders will be filled within the system, refer to Figure 2.3.

We can see that from Figure 2.3, every time the trader invokes the `PlaceOrder` function, the system will check for a fill in the order and execute it. This allows us to save space by only doing the check when it is required, rather than every time the block is finalized.

Figure 2.4 shows how the execution process will work within this system. One thing to note is that the term **send** works differently in the blockchain world, as in the case of substrate, "sending" a value represents the transition in the blockchain

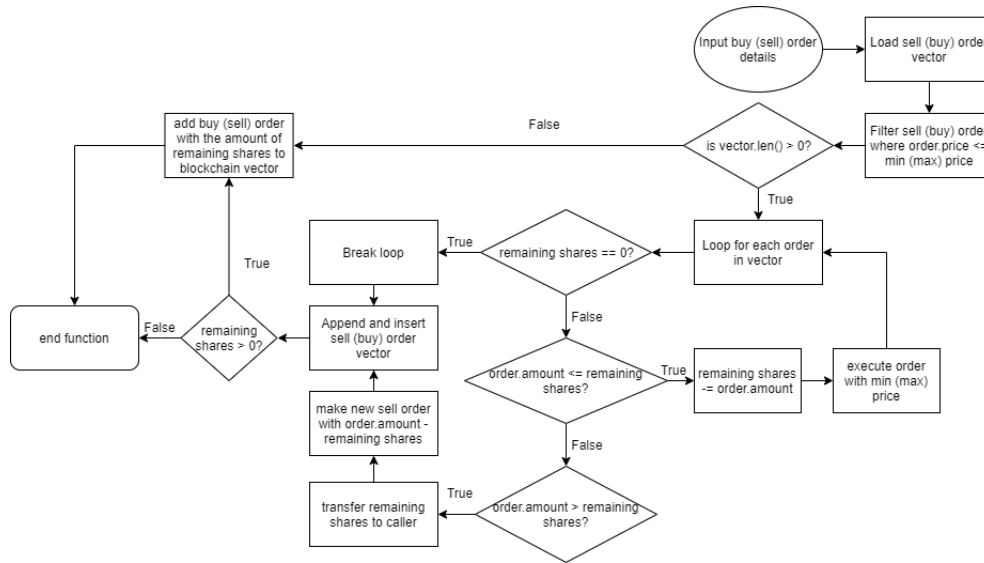


FIGURE 2.3: The flowchart of how both the buy order and sell will be filled within the blockchain

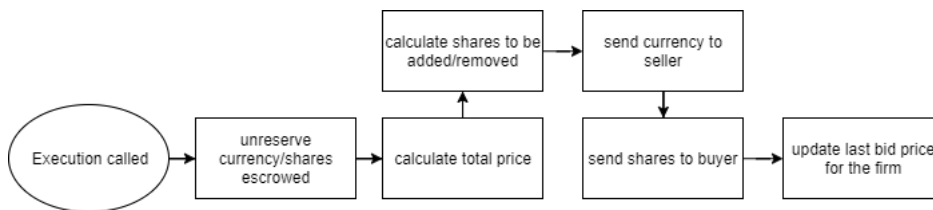


FIGURE 2.4: The process of how an order is executed once it is filled

state, meaning the hash map for OwnedShares and Balance will just be updated to current balance + balance to add or subtract.

## 2.4 Function Development

In this section, I will go through the detailed logic of how the features mentioned above will be coded for the project. The full source code can be found from the following GitHub repository.

[https://github.com/hoonsubin/exchange\\_platform](https://github.com/hoonsubin/exchange_platform)

### 2.4.1 Administrative Functions

For this project, there will be a total of five administrative functions.

- give issue rights
- revoke issue rights
- change authorized shares
- close market
- open market

The function for giving issue rights to organizations will be shown in Figure 2.5.

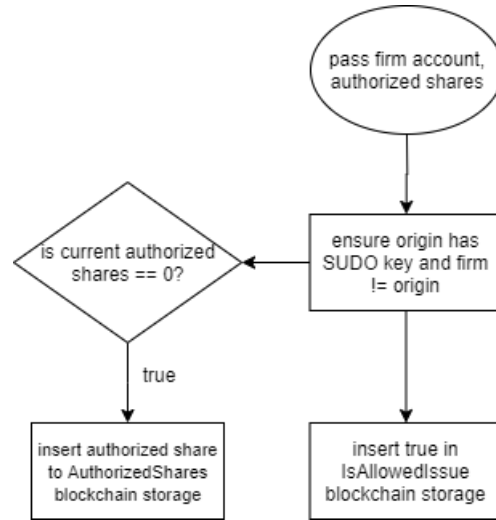


FIGURE 2.5: The logic for how giving issue rights will work

The parameters for this function are *firm* and *authorized shares*. Once the information is given, it will first ensure that the caller of the function has the sudo key and that the caller is not a firm. If the given address of the firm does not have any authorized shares—meaning that this is the first time the firm is given the right to issue shares—the number of authorized shares will be set to the number that is given in the parameter.

The **revoke issue rights** function will start with the same state checks, such as caller sudo keys, firm address. Once the check has passed, it will simply change the *IsAllowedIssue* value for the firm to false.

The **change authorized shares** function will take the firm address and the number of authorized shares as parameters. Before it mutates to blockchain storage, it will first check if the caller of the function has sudo keys, if the given firm’s address is allowed to issue shares, the sender is not changing their own authorized shares, and the given number of newly authorized share is not below the floating share of the firm. When all the conditions are met, the function will replace the *AuthorizedShares* blockchain storage for the given firm with the given number of authorized shares.

Finally, both **close market** and **open market** functions will check the caller’s sudo key and change the *CloseMarket* blockchain storage value to either *true* or *false* respectively.

## 2.4.2 Minting and Burning Shares

The following two functions are only callable by an account that is registered as a firm.

- issue shares
- retire shares

First, the **issue shares** function will check if it is allowed to issue shares. You can see the flowchart for this in Figure 2.6. The function starts with making sure

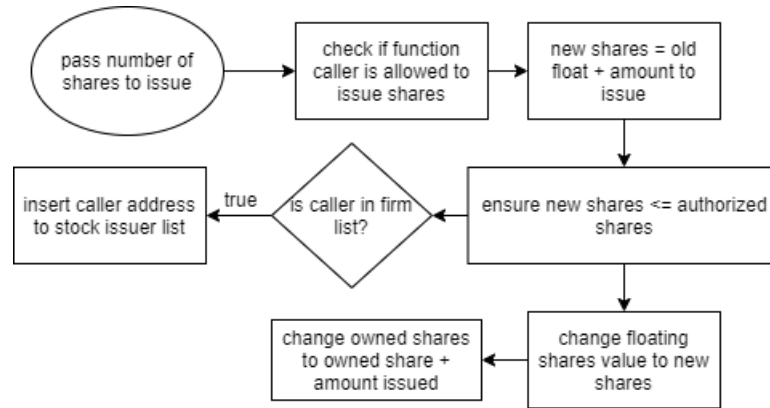


FIGURE 2.6: The logic for how a firm issues a share in the blockchain

that the given amount of shares to mint (issue) + current float is smaller or equal to the number of authorized shares. This prevents the firm from issuing more than they are allowed to issue within the system. After that, the function will update the value of floating shares to add the number of newly minted shares, and also change the number of the shares the calling firm currently owns, sending the newly issued shares to its own account. Additionally, if the calling firm is not part of the firm's list in the blockchain storage, it implies that this is the first time that the firm has issued shares, meaning this firm will be added to the list of issuers.

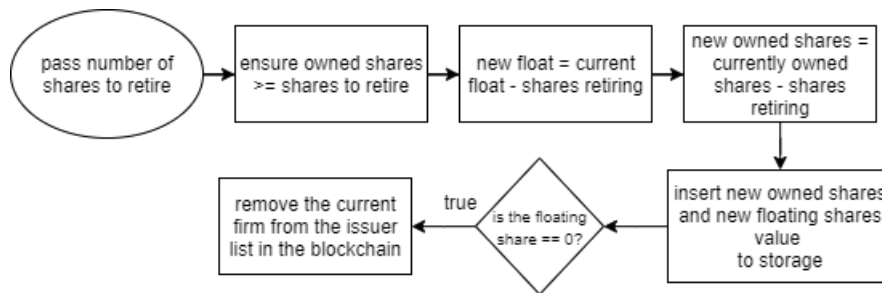


FIGURE 2.7: The flowchart for retiring shares in the blockchain system

Next is the **retire shares** function. The flowchart for how the function is programmed into the blockchain system can be seen from Figure 2.7. We can see that this function only takes the number of shares to retire as its parameter. This is because the function will only look into the number of owned shares for the function caller, that the caller issued. This effectively restricts accounts from burning (retiring) other companies' shares. Next, it will calculate how many shares the company will have after they burn the shares, and the number of floating shares in the market as well. If the company has no floating shares recorded in the platform, it will take down the company's address from the IssuerList. This only means that the company does not have any floating shares and cannot be traded on the platform, and will not change the state of the company having issuing rights since that can only be set by the sudo account.

### 2.4.3 Trading and Fees

One thing about this platform is that it does not require a broker or any other third-party entities in order to fill and execute an order. This allows the platform to function with minimal underlining costs.

As mentioned in Section 2.2.1, typical public blockchains like Ethereum or Bitcoin require economic incentives for block validators to maintain a full blockchain node. These "incentives" are presented as transaction fees, and gas (which are the main mechanism for preventing infinite loops in a smart contract) for blockchains with smart contract functionality like Ethereum. However, one issue is that, as the blockchain gains massive transactions, the fees will also scale in the proportion of the transaction volume (Bashir, 2018). This means that for an exchange platform, the fee for each transaction may vary when implemented for a public blockchain, giving less control for the platform maintainer (Pop et al., 2018). Fortunately, because our platform will be using a Proof of Authority consensus in a single-purpose blockchain, we have more control over how the transaction fees perform as we do not have to consider incentivizing the validator in a growing blockchain. The fees for the exchange platform can be changed from the `src/chain_spec.rs` file, in the genesis configuration.

Looking into the source code, the balance module in substrate framework has four fee-related genesis configurations.

- **Transfer fee** – The fee required to make a transfer
- **Creation fee** – The fee required to create an account
- **Transaction base fee** – The base fee for a transaction
- **Transaction byte fee** – The fee to be paid for making a transaction in a per-byte portion

The fees can be changed by adjusting the variables in the genesis config. The fees will be transferred to the block validator, which will act as a means for incentivizing the platform maintainer.

### 2.4.4 Starting and Testing the Node

It is possible to see the blockchain state and storage data via the **Remote Procedure Call (RPC)**. However, for this paper, I will use the front-end UI that is developed for interacting with the substrate blockchain, provided by Polkadot, like the one shown in Figure 2.8. The source code for the front-end UI can be accessed from <https://github.com/polkadot-js/apps>, or the web implantation from <https://polkadot.js.org/apps/>.

To build the project I use the following command on the terminal.

```
$ ./exchange_platform/scripts/build.sh && cargo build --release
```

The `./scripts/build.sh` command will build the Wasm binary that will be required for the blockchain to operate. The `cargo build -release` command will build the actual binary for the entire blockchain node.

To connect with the substrate blockchain, I will have to start the blockchain developer local node via the following command in the command line.

```
$ ./exchange_platform/target/release/exchange_platform --dev
```

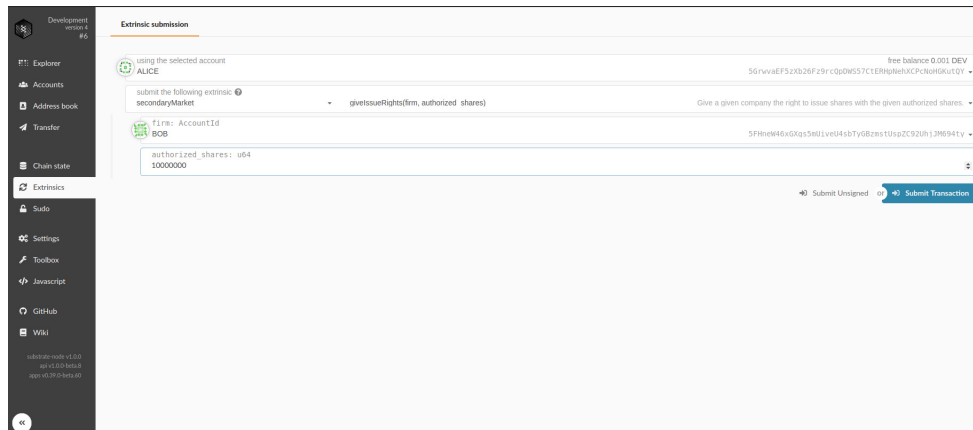


FIGURE 2.8: Alice giving Bob the right to issue shares via a Substrate web frontend that is connected to a local node

Once we check that the node is correctly running, we just need to go to the front-end UI and in the **Settings**, change node to **Local Node**.

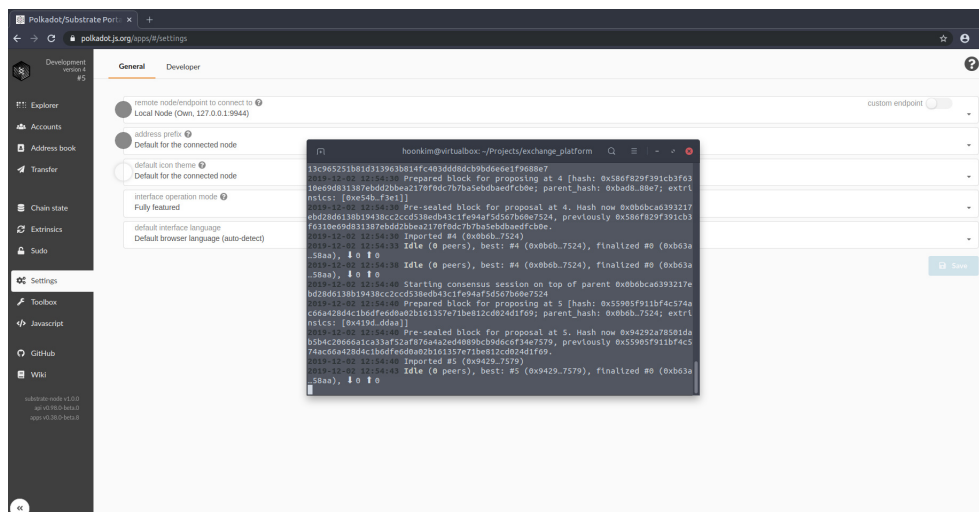


FIGURE 2.9: A local node running with a front-end webpage connected

Inputting values and calling functions can be done through the **Extrinsics** tab. Reading the chain value can be done through the **Chain State** tab.





## Chapter 3

# Results

In this section, I will present the results for the blockchain-based stock exchange platform. The purpose of this test is to show that placing, filling, and executing shares can work all within the blockchain without third-parties such as a brokerage firm or a Clearing House. It will also test for counterparty risks through checking if share escrow and block-number-based order expiration function properly. I will test this by setting up a single local node with multiple accounts accessing that node. This will allow us to focus only on the market functionality of the blockchain. The involved accounts will be **Alice**, **Bob**, **Charlie**, and **Dave**. Alice will act as the platform maintainer, holding sudo keys and the sole account that can invoke administration functions such as giving stock issue rights to other accounts and will have a role as a trader like the other accounts as well. Bob will act as a company listed in the public stock exchange, issuing 100,000 shares for this test. All the other accounts excluding Bob will act as the traders for the shares, buying and selling Bob's stock on the market. Every account will start with 1,000,000,000 exchange platform native tokens (assumed to be pegged to JPY for this research). Furthermore, to keep things simple, the blockchain fees will be set as 0 JPY for this test. Finally, the block time will be set to 10 seconds, meaning that the platform will generate one block every 10 seconds on average. This also shows the minimum time it takes for an order or any blockchain function to take effect in a single node.

### 3.1 Trading and Escrows

#### 3.1.1 Placing Orders

After Alice has given Bob the right to issue shares, and Bob has issued 100,000 shares, first I have inputted the values shown in Table 3.1 for the first three traders. Even though it is unlikely in real-life scenarios, in this test the platform maintainer (Alice) will also trade. There are no particular reasons for this decision other than to make the test simple and straight-forward to conduct.

TABLE 3.1: The content for the first trade orders from three different accounts.

Order no.	Address	OrderType	Share no.	Price (Yen)
1	Alice	Buy	1,000	5,000
2	Dave	Buy	1,500	4,000
3	Bob	Sell	1,300	2,500

We can see that the results in Table 3.2 shows us the shares have been successfully traded with the currency from buyer to sellers.

TABLE 3.2: Owned assets for three accounts after the first order transaction

Address	OwnedShares	Balance (Yen)	Escrowed
Alice	1,000	995,000,000	0 Yen
Bob	98,700	1,006,200,000	0 shares
Dave	300	994,000,000	4,800,000 Yen

If we recall back to Figure 2.3, you might notice that if an order is not fully filled, it will partially execute it by placing a new order with the difference in the left share amount. Just like the flowchart, the blockchain exchange platform has filled Bob's order of selling 1,300 shares with Alice's order of 1000 shares and filled the remaining 300 shares with Dave. Once the order has executed, the platform has placed a new order in Dave's name with the remaining 1,200 shares, hence why Dave has 4.8 million Yen (1200 shares \* 4000 Yen) of escrowed (reserved) currency.

TABLE 3.3: Bob places another sell order

Order no.	Address	OrderType	Share no.	Price (Yen)
4	Bob	Sell	1,500	3,000

After Bob has placed a new order with the values of Table 3.3, the results are in Table 3.4. We can see that the balance for Dave has not changed from Table 3.2.

TABLE 3.4: Owned assets for three accounts after Bob's second sell order has been executed

Address	OwnedShares	Balance (Yen)	Escrowed
Alice	1,000	995,000,000	0 Yen
Bob	97,200	1,011,000,000	300 shares
Dave	1,500	994,000,000	0 Yen

This is because Bob's order was filled from Dave's reserved balance, hence why now Dave has no escrowed balance while Bob's entire balance has increased by 4.8 million Yen. We can also see that Bob's leftover shares have been successfully escrowed by 300 shares (1500 shares - 1200 shares).

Now we add a new trader named Charlie to our platform who will buy 2,000 shares for 6,500 Yen each. Alice will be selling her shares for more than Charlie's asking price.

We can see the results for the new orders in Table 3.6. The first thing to note is that as we want the platform to do, Alice has 800 shares locked, while her balance has no change from the previous state. This is because Alice's order has not been filled due to the max price for Charlie's buy order being lower than the minimum price for her sell order. Now Alice's 800 shares will be locked away from her until

TABLE 3.5: Alice, Bob and Dave places a sell order, and a new trader Charlie places a buy order.

Order no.	Address	OrderType	Share no.	Price (Yen)
5	Alice	Sell	800	7,000
6	Bob	Sell	1,100	5,500
7	Dave	Sell	1,000	5,000
8	Charlie	Buy	2,000	6,500

either another buy order with a higher maximum price is placed, or when the order becomes expired, meaning that when the blockchain's current block number is greater than the order's expiration block number.

TABLE 3.6: Owned assets for all four accounts in the platform

Address	OwnedShares	Balance (Yen)	Escrowed
Alice	200	995,000,000	800 shares
Bob	96,100	1,015,750,000	100 shares
Charlie	2,000	990,250,000	0 Yen
Dave	500	999,000,000	0 Yen

There is another part in Table 3.6 that we should note. Bob has 100 shares locked despite in table 3.5, he has placed a sell order earlier than Dave and the asking price is within Charlie's limit. This is because the platform will re-order the order list to the lowest minimum price, or highest maximum price when filling an order, attempting to give the best possible order for the later trader, hence why in this case Dave's order was filled first rather than Bob's order.

### 3.1.2 Order Expiration

Testing the block time-based order expiration function will be straight-forward. The basic idea will be the same as before, but this time I will have different expiration blocks for each order, and placing orders in specific block times to check if some orders have successfully been expired in the given block number.

TABLE 3.7: Bob will sell shares while the others will buy them with the given expiration number

OrderBlock	Address	OrderType	Share no.	Price(Yen)	Exp.Block
7	Bob	Sell	5,000	3,000	20
10	Alice	Buy	500	5,000	40
12	Dave	Buy	3,000	5,500	45
21	Charlie	Buy	1,500	4,000	60

In the test results, **OrderBlock** refers to the block number in which the order was placed. Additionally, **Exp.Block** refers to the block number in which the placed

order will expire. Now we have finished the first test, we'll look into what the results are.

TABLE 3.8: The results after the order has been placed.

Address	OwnedShares	Balance (Yen)	Escrowed
Alice	500	998,500,000	0 Yen
Bob	96,500	1,010,500,000	0 shares
Charlie	0	994,000,000	6,000,000 Yen
Dave	3,000	991,000,000	0 Yen

After the test in Table 3.7, we can see the results in Table 3.8. We can see that Bob's sell order has been filled with Alice and Dave's buy order, but despite the fact that Bob still should have 1,500 shares available for trade with Charlie, his order was not filled. This is because before Charlie placed his order in block number 21, Bob's order expiration number has passed. The other thing to notice is that Bob now has 96,500 shares, even though he placed an order of selling 5,000 shares. This proves that the block expiration function has worked successfully and returned any escrowed securities/balances to the owner of the order. Additionally, because Charlie was not able to make it in time for Bob's expiration number, his order will still be in the blockchain until it is filled with another trader's order. Hence why he has an escrowed balance of 6 million JPY (1,500 shares \* 4,000 Yen).

TABLE 3.9: Everyone except Dave will each sell Bob's shares while Dave buys Bob's shares in block 40

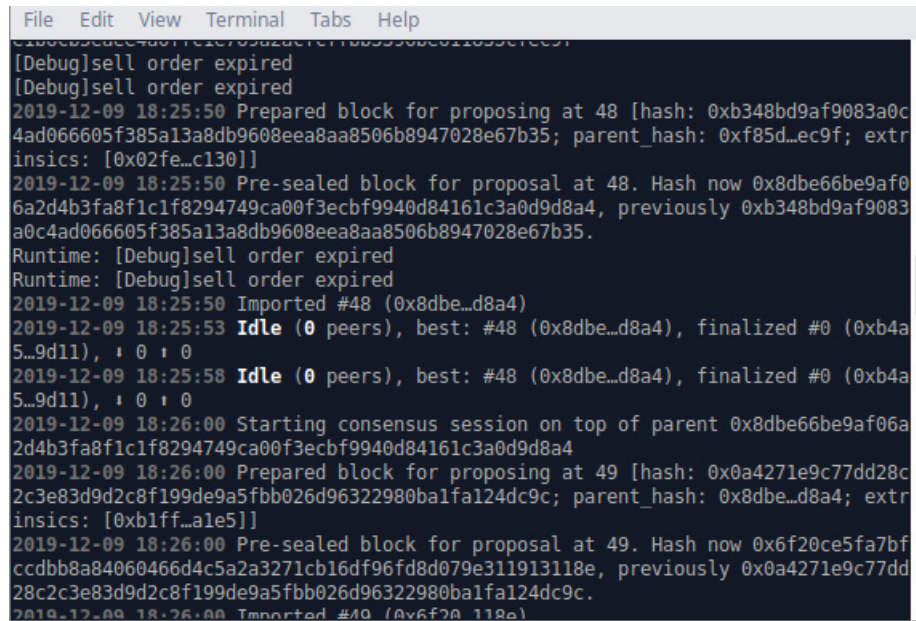
OrderBlock	Address	OrderType	Share no.	Price(Yen)	Exp.Block
36	Bob	Sell	1,500	3,000	60
38	Alice	Sell	500	3,000	48
39	Bob	Sell	500	3,000	48
40	Charlie	Sell	500	3,000	60
50	Dave	Buy	1,500	4,000	60

TABLE 3.10: The results right after the order has been placed.

Address	OwnedShares	Balance (Yen)	Escrowed
Alice	500	998,500,000	0 Yen
Bob	95,000	1,016,500,000	0 shares
Charlie	1,000	995,500,000	0 Yen
Dave	3,500	985,500,000	4,000,000 Yen

After the orders with the values given in Table 3.9 were placed, we have the results listed in Table 3.10. As we can see here, after Bob has placed his sell order in block 36, the order was then filled with Charlie's buy order from Table 3.7.

Later Charlie placed a sell order of 500 shares that is set to expire at block number 60, hence why after the orders have been executed, he is left with 1,000 shares as



```

File Edit View Terminal Tabs Help
[Debug]sell order expired
[Debug]sell order expired
2019-12-09 18:25:50 Prepared block for proposing at 48 [hash: 0xb348bd9af9083a0c
4ad066605f385a13a8db9608eea8aa8506b8947028e67b35; parent_hash: 0xf85d...ec9f; extr
insics: [0x02fe...c130]]
2019-12-09 18:25:50 Pre-sealed block for proposal at 48. Hash now 0x8dbe66be9af0
6a2d4b3fa8f1cf8294749ca00f3ecbf9940d84161c3a0d9d8a4, previously 0xb348bd9af9083
a0c4ad066605f385a13a8db9608eea8aa8506b8947028e67b35.
Runtime: [Debug]sell order expired
Runtime: [Debug]sell order expired
2019-12-09 18:25:50 Imported #48 (0x8dbe...d8a4)
2019-12-09 18:25:53 Idle (0 peers), best: #48 (0x8dbe...d8a4), finalized #0 (0xb4a
5...9d11), + 0 : 0
2019-12-09 18:25:58 Idle (0 peers), best: #48 (0x8dbe...d8a4), finalized #0 (0xb4a
5...9d11), + 0 : 0
2019-12-09 18:26:00 Starting consensus session on top of parent 0x8dbe66be9af06a
2d4b3fa8f1cf8294749ca00f3ecbf9940d84161c3a0d9d8a4
2019-12-09 18:26:00 Prepared block for proposing at 49 [hash: 0x0a4271e9c77dd28c
2c3e83d9d2c8f199de9a5fbb026d96322980baf1a124dc9c; parent_hash: 0x8dbe...d8a4; extr
insics: [0xb1ff...ale5]]
2019-12-09 18:26:00 Pre-sealed block for proposal at 49. Hash now 0x6f20ce5fa7bf
ccdbb8a84060466d4c5a2a3271cb16df96fd8d079e311913118e, previously 0x0a4271e9c77dd
28c2c3e83d9d2c8f199de9a5fbb026d96322980baf1a124dc9c.
2019-12-09 18:26:00 Imported #49 (0x6f20...118e)

```

FIGURE 3.1: The debug message of the blockchain telling us that Alice and Bob's sell order has expired in block 48.

Dave's buy order was placed in block number 50. With this, we can also see that Alice's order in block number 38 and Bob's order in block number 39 does not get satisfied as both of them still end up with their initial shares and balance (excluding the transaction from Bob's first order in block 36 to Charlie). The fact that they have been expired can be seen within the custom made debug message from the blockchain in Figure 3.1.

After going through the tests for the blockchain-based stock exchange platform, we can see that both the trading functions and the block time-based expiration functions are working exactly how it was described in Chapter 2.

## 3.2 Confirming the Hypothesis

The hypothesis for this research was "Developing a single-purpose blockchain for trading stocks can decrease the number of third-parties involved in the settlement and increase the integrity of the transaction data". The variables in this hypothesis are **number of third-parties** and **data integrity**.

As we have seen from Chapter 2 and test results in this chapter, using a blockchain-based system allows the platform to function without a broker or a Clearing House. This means that the parties involved have the flexibility to choose their fees without having to consider the overhead costs and still provide incentives for the platform maintainers, which were some of the concerns in adopting a blockchain-chased platform (S&P Global, 2019). Starting from order input to settlement, the system developed in this paper does not require any third parties that are usually involved in the traditional system, outside of the platform maintainer who is responsible for fixing the prices of the native token. I can safely say that even though the current system is quite lacking in allowing the investors to control how their order is filled, it clearly reduces the parties required in the settlement process, consequently reducing the risk that is involved as well.

The second variable, data integrity was not deeply explored from the tests, but as mentioned in Section 1.3.4, every block of data, which includes the trade information in this project, is referenced by the next data block through a hash pointer that is unique to each nonce and data sets inside the block. This means that changing one block will require the hacker to change the entire chain of blocks just to allow the blockchain to not reject the out-of-place block. With a test blockchain like the one described in this paper, this is not a hard task to do. But as more users use this platform and more blocks are added, the data become tightly linked with each other. Making the integrity of the platform proportional to the number of transactions that were settled in the platform.

With this, I can conclude that indeed, developing a single-purpose blockchain as a stock exchange platform does in fact reduce the number of parties involved in a transaction. Furthermore, as mentioned previously and in Section 1.3.4, it is confirmed via a secondary source that a blockchain-based system has stronger data integrity than that of the traditional system because of the aforementioned mechanism of block propagation (Antonopoulos, 2017).

## Chapter 4

# Discussion

### 4.1 Advantages

In the traditional system, through the help of modern technology, most systems are able to handle millions of transactions in less than 10 milliseconds. However, even with ACH and DMA systems in place, the settlement process still requires up to 2 business days. In this system that I developed, the transaction requires up to 7 seconds, but the settlement happens as soon as the order is filled without any third-parties such as the Clearing House involved (Investopedia Staff, 2019).

Having a system that does not require constant maintenance, such as the one developed in this research, makes the system act in similar ways as the Forex market, where trades are conducted by a network of computers (nodes in the case of blockchain) without having a need for a centralized system (Lioudis, K., 2019).

Additionally, because this system is built on top of a blockchain, it has most of the benefits that come with what a typical blockchain has, including the benefits that come from the Substrate framework. Such benefits include data immutability, secure timestamping, auditability, authorization, integrity, forgery protection, consistency (Antonopoulos, 2017) and interoperability through communications with a relay chain if made into a parachain of the Polkadot network, which will also solve the scalability issue that is prominent in most public blockchains (Wood, 2017). Additionally, it is possible to implement Smart Contracts to automate types of orders that the investor wishes to utilize, giving full freedom to develop a custom order type that is not supported by traditional brokers in many exchanges (Kim and Laskowski, 2017). This expandability also opens up the possibility of developing a built-in **Market Maker** algorithm that will buy and sell shares to create market liquidity, which is not implemented at this stage of development.

The immutability of blockchain blocks (data) comes from the timestamp server. This works by hashing the information of a block with a timestamp in the block propagation stage including the hash of the previous block. By 'chaining' these hashes by having new blocks have the hash reference of the previous block, it makes it easy for everyone to know if the data is intact just by checking if the hashes are properly chained. Additionally, it is computationally infeasible for malicious parties to change any entries that are already recorded in the blockchain, as they will have to change the entire data up to the entries that they want to change in the blockchain (Nakamoto, 2008).

Bhandarkar, Bhandarkar, and Shiva, 2019 states that the blockchain's nature of being secure, transparent, low transaction costs, and fast transaction speed allows it to be more advantageous than the existing exchange platforms, making tokenizations of stocks a possible future direction for many exchange platforms (Grody, 2017). This sentiment is backed by the research conducted by Aitken, Frederick,



and Ji, 2015, who prove that market quality does have a positive relationship with market integrity.

With these facts and through multiple tests that were shown in Chapter 3, I can safely conclude that a blockchain-based platform is indeed

- Cost effective
- Secure
- Fast settlement
- Flexible

Which are some of the variables that increase the efficiency in operations of stock exchange platforms according to the hypothesis for this research study.

By developing a single-purpose blockchain for a specific capital market rather than using an existing public blockchain/protocol like Ethereum smart contracts, or Counter Party for all the markets, I was able to develop a platform that has more control over its shares, which other platforms would struggle due to dynamic gas prices or fees (Chiu and Koepl, 2018). This also allows us to issue stable coins that are not affected by the price of other cryptocurrencies (Bashir, 2018).

Finally, through developing and implementing a stock and token escrow function, block time-based expiration function and having a firm-wide account to represent a single account, I was able to mitigate some of the counterparty risk that well-known public blockchains have, which was one of the potential risks for utilizing the blockchain technology for security exchanges (S&P Global, 2019).

## 4.2 Challenges and Limitations

There are several limitations of the project discussed in this paper that mostly stem from the lack of research time and resources and a couple of others stemming from the inherent limitations with the blockchain itself. This section will also discuss challenges that a blockchain-based system will face such as legal ambiguity and its public nature.

### 4.2.1 Limitations from the Tests

Even if the developed blockchain-based exchange platform does function the way it was intended to, there are several limitations within the conducted tests and the lack of some tests that may challenge the integrity of the final implementation of the platform.

- Lack of proper benchmarking – Being able to benchmark the exact speed for each trade would contribute in determining exactly how the blockchain-based platform would compare with the traditional system. Unfortunately obtaining information for proprietary exchange platforms was a huge challenge with my given time and resource for this research study.
- Low trade volumes in tests – The test was done with four traders placing their orders in a one-by-one fashion, which makes it hard to know the capacity of the blockchain-based platform when this technology is applied in real life.



- Lack of testing with multiple nodes – Being able to test with multiple nodes that connect to other nodes with a single consensus would allow us to understand exactly how the consensus algorithm can work, and its potential effect in performance.
- Lack of a Primary Market – Because this research only focuses on developing and testing a secondary market purely based in the blockchain, we cannot determine the effectiveness of using a blockchain-based system in the primary market.

Additionally, there are several questions about how the platform will scale when there are trillions of trades on the book, or if there will be any integer overflows as the data type for share is an unsigned 64-bit integer, which can hold up to around 18 quintillion. Any value that goes beyond that will overflow back to 0, and for an exchange platform where it is designed to handle real legal tender, this may be a huge flaw and can potentially cause devastating economic damage.

#### 4.2.2 Privacy Issues

It was possible to mitigate the issues with volatile tokens and fees in blockchains via developing a single-purpose blockchain-based system. However, one other inherent limitation with a distributed ledger technology is its lack of anonymity (Chiu and Koepl, 2018). This means that anyone is able to look into which trader has how many coins and assets of a certain company. Having a completely transparent platform does contribute to ease of investigations and auditing, but it also raises the issue of privacy for its traders. According to Comerton-Forde, Putniņš, and Tang, 2011, the anonymity of a market does affect the quality of the market, though to which degree it does is debatable. Thus, to develop a blockchain-based market that can act as a substitute for the existing system, the platform should be able to offer the option to the traders to trade anonymously or publicly.

#### 4.2.3 Government Regulations

Due to blockchain technology being a relatively new concept at the time of writing this paper, there are no established legal frameworks to distinguish a system working on the blockchain from cryptocurrencies, though in recent years many have started to appear in different nations (Yafimava, 2019). Because the blockchain is considered as a class of technology, some legal regulations will have to apply different measurements in a case-by-case situation (European Parliamentary Research Service, 2019). Therefore, the regulations that I refer to in this section will be those that can potentially affect a semi-public single-purpose blockchain used as a platform to trade securities like the one discussed in this paper.

According to the findings from Global Law Library of Congress, 2018, they claim that the majority of the regulations stem from concerns with money laundering, terrorism, alternate corporate funding methods like **Initial Coin Offerings**, and the high volatility in cryptocurrencies which prompted a warning from several governments in investing in cryptocurrency markets. Having said this, not every jurisdictions were negative about blockchains, as some have developed cryptocurrency-friendly regulations, or even going far as to accepting it as legal tender (Global Law Library of Congress, 2018).

In 2017, due to the lack of proper methods for taxing ICOs, both the South Korean government and the Chinese government has banned any form of ICO (FLETA

Blockchain, 2019). The potential risk that a blockchain-based exchange platform might have may come to form the definition of an ICO. This paper only focuses on developing the secondary market, but we can think of a hypothetical situation where we do have a primary market that is based on blockchain technology. Frankenfield, 2019 defines ICO as a method to raise funds for the cryptocurrency industry by issuing a new token for the platform which may either serve the purpose of using the service/product the company is offering, or act as a stake in the company or project. Some may ask the question, can we considered a primary market based on blockchain technology to be an ICO? One key point for an ICO is that it must issue tokens for a specific platform that utilizes the blockchain, as tokens are the major incentive to power the validation process of a block, given that it is a sovereign blockchain (Schulz, 2019). In the case of the project developed in this paper, the native token is assumed to be pegged to the official currency where the market resides and focused on emulating an **Initial Public Offering** in the primary market. Trading may not be highly regulated in this market, but the circulation of the native token is highly controlled by the platform maintainer, i.e., the sudo address holders. Native tokens are only used to trade with a form of meta-coin that is the firm's stock, and the stocks cannot be traded directly with other stocks or accounts as it must go through the limit order with the native currency. Even in the technical sense, an ICO raises funds via the sales of **coins**, while a blockchain-based stock exchange does not represent company ownership through native tokens. The other issues such as native tokens with high volatility and money laundering can be countered by the fact that this is a centralized blockchain that gives the most control over to the platform maintainers. Having said this, it is possible that a blockchain-based capital market can be viewed as a typical ICO Thus giving the platform a regulatory risk (S&P Global, 2019).

### 4.3 Implications for Management

The implications of a blockchain-based settlement system with a built-in auction platform for management is huge. Having said all the advantages and potential problems that this system has, it is my opinion that the system developed in this paper will be most useful for **Tokenization of Assets** and its exchange with other owners.

Tokenization of company assets through blockchain is nothing new, as there are existing projects that satisfy that purpose (Clohessy and Acton, 2019). However, the work that I propose is not just an immutable database with limited circulation, but a platform that is capable of utilizing a smart contract, or a runtime module that can accept information from other blockchains and migrate data between them for exchange or liquidation. Through applying the framework from the works of Poon and Buterin, 2017, this platform will require two type of blockchain, one parent chain, and one or more child chain(s). The parent chain will be the blockchain where the exchange is held (in this case the blockchain that was developed for this paper). The child chain will be a permissioned private blockchain that is used inside the firm for the tokenization of assets. By developing a smart contract or a runtime module that can communicate between the child chain and the parent chain, it is possible to migrate tokenized assets from the child chain to the parent chain. To prevent fraudulent behaviors, I propose that in order to migrate and trade assets from the child chain to the parent chain, the child chain must deposit a sum that is proportionate to the assets that are being moved in the native token of the parent chain, which in this

context will mean the local currency of where the exchange platform is operational. Through this system, it is possible for companies to not only tokenize their assets in their permissioned blockchain that can have other customized functionality but to transfer those assets for auction in the public exchange. One limitation from this is that, a system with this architecture will not guarantee the transactions of physical objects. For that we must rely on removing the collateral funds, similar to how the **Staking** function works in the Polkadot network (Laboon, 2019).

Regardless, even with the aforementioned limitation, this system will prove to be a useful tool for asset management and investment management in terms of tracking company assets and the exchange of assets.

## 4.4 Future Research

In this section, I will discuss the possible future research direction and any additional developments or implementations that could contribute to creating a complete substitute for the traditional capital market system in the future.

### 4.4.1 Code Improvements

There are several improvements that can be made for both the code and the design of the project discussed in this paper. Due to time constraints, I was not able to put too much effort into optimizing the code or to make the code more readable. Thus my first improvements would be to refactor several blocks of code, especially the function for placing an order, and the private function for exchanging shares. For instance, when the order list gets too long, I can split the list into a smaller and reasonable chunks and execute only those orders to optimize the order filling process, rather than filling the entire list of orders. Next, I would change the data type for the share to an unsigned 128-bit integer, as this would extend the integer limit that an unsigned 64-bit integer has by an exponential amount. This can allow the system to handle more integers without overflowing. Finally, I would add different types and classes of stocks like the common stock, rather than just a preferred stock like the current system. This will allow us to develop functions for payments of dividends to investors. However, this functionality will require more than just a single blockchain if we consider its compatibility with existing systems.

### 4.4.2 Best Execution, Broker, Market Maker Module

The current system developed in this paper has a simple order execution logic, where it will fill orders that will satisfy the minimum (maximum for buy orders) price requirements of the early trader and the maximum (minimum for buy orders) request of the late trader as shown in Figure 2.3. This algorithmic execution does not guarantee the best execution for all the investors involved, which is one of the biggest limitations of the system developed in this paper (Securities and Exchange Commission, 2013). However, this is not an inherent limitation of the system, but rather a result of limitations in development time, as the substrate framework is flexible enough to allow development of **off-chain workers**, which an external code or software that runs before the block propagation stage of the blockchain, connected via either a runtime module or a smart contract which is similar to how other Layer 2 blockchain scalability framework works (Poon and Buterin, 2017). Thus, it is possible to allow the development of external execution logic as a module for filling orders with its own consensus algorithm to communicate with other execution logic,

which can allow investors to choose what type of "broker" works best for them. With the same principle, it is also possible to develop a Market Maker module that is built-in to the blockchain-based exchange, which will algorithmically trade shares to create liquidity in the market, where the actual code will be some variant of existing automated trading software. Furthermore, it is also possible to communicate and trade with other stock exchanges that do or do not implement the blockchain-based system, assuming that they accept the token that is handled in this system, as everything is just made out of interconnected APIs, which also provides an easy method for dividend payments.

#### 4.4.3 Primary Market Development

In order to develop a fully functioning blockchain-based stock exchange platform, it would be important to implement not only the secondary market but also the primary market.

To develop a primary market based on my currently developed platform, it may be optimal to utilize a modified staking function in substrate, which would allow for any firms to participate in an IPO given that they have enough currency to stake (Laboon, 2019). It would be possible to use the **Democracy** substrate runtime module to give anyone in the blockchain with enough currency to vote on which company can hold an IPO, which I believe that it may increase the autonomy of the market, without having to involve underwriters and put all trust in a single third-party. However, the feasibility of such system will require further research to confirm or deny. I do want to say that it is an area that deserves further research as this has the possibility to reduce the time and money by a huge margin in contrast to the current system.

For the primary market, it may not be wise to use the current secondary market's order filling mechanism, considering that the primary market has a very different approach when it comes to how shares are valued and traded (Kenton, 2019). I suggest that for the primary market, it is important to add an auction function, where the value of shares are determined by investors' bids, with different types of stocks being only available to certain addresses (investors) rather than being completely open to everyone. Implementing the primary market on a blockchain-based system will require many adjustments and trial-and-error as a small compromise in security or system functionality will lead to real-life repercussions considering that we are talking about ownership of a company with real money. Thankfully the substrate framework is flexible enough to support a forkless update, which means that the sudo address can directly upload the web assembly binary to the blockchain to change and update its functionality, meaning any modifications required can be easily applied until we are able to perfect the system (Kashitsyn, 2018).

#### 4.4.4 Zero-Knowledge Proof Cryptography

As mentioned in Section 4.2.2, the blockchain is a public and transparent database that anyone can view. However, in October of 2016, Zcash, the first blockchain that allows anonymous transactions through a special zero-knowledge proof cryptography named zk-SNARKs was launched to the public (Bashir, 2018). Zero-knowledge proof cryptography is (to oversimplify it) a form of cryptography where the prover can prove to the verifier that a certain statement is true (in case of blockchain, that a certain account does possess a certain value) without having to reveal any important information like the identity of any of the parties (Reitwiessner, 2016). This essentially changes the publicly readable nature of a blockchain and allows anonymous

transactions, where the identity of the counterparty is hidden, but also assures that the value is safely traded. Using the same technique, in 2019 a project named **Zerochain** was developed. Zerochain aims to implement the zero-knowledge proof cryptography in substrate framework by adding a privacy layer on top of the existing substrate blockchain core functionality, which includes the project that was developed for this paper (Osuke, 2019). The Zerochain project is still in its infancy, and due to the complicated mechanism of the secondary market discussed in this paper, implementing a privacy layer to the capital market blockchain will require us to modify Zerochain's source code so that it can also encrypt the number of shares a certain account holds (Osuke, 2019). Having said this, it is my opinion that this technology will be beneficial to a blockchain-based settlement system, and worth further development.



## Chapter 5

# Conclusion

In conclusion, the current stock exchange requires numerous parties to operate. Clearing houses are always faced with default risks from the clients, which shows us that their business model solely relies on the fact that they are backed by a centralized financial institution with massive capital. Because of this fact, the settlement process has several overheads. This also affects the workflow of investment managers and asset managers, as the risks and overheads within the current system forces managers to spend more just to overcome this issue. Furthermore, exchanging assets of a company was made hard due to a lack of a proper settlement system and a trusted database. The research question that this paper aims to answer was, how can we make the existing digitized settlement systems more efficient and secure for investors and managers?

My proposed solution and the hypothesis for this research is to develop a single-purpose blockchain for trading stocks. Developing a single-purpose blockchain can decrease the number of third-parties involved in the settlement and increase the integrity of the transaction data. Using the substrate framework, it is possible to develop a blockchain that is designed for specific exchanges and can only be managed by the exchange platform itself while allowing data access to anyone, giving it the characteristics of a consortium blockchain. The substrate framework provides a simple and scalable blockchain solution that has interoperability with other blockchains, making asset transactions much more efficient and compatible.

As a proof of concept, I have developed a full blockchain and a substrate module that simulates the secondary market with several assumptions given, that eliminates all the involved parties from the traditional system excluding the stock exchange, investors and firms. The blockchain is able to give special permissions to the platform maintainers such as allowing a firm to issue shares, changing authorized shares, closing and opening the market. Every other party will only have the permission to issue/retire their own shares (if they are given the permission to do so by the platform maintainer) and buy/sell shares. After the development was finished, I have tested its functionality and it was shown that the blockchain with given assumptions, can indeed accept orders, escrow assets, fill orders, execute orders, consolidate the transaction without any involvement from a broker or Clearing House with a scalable and expandable system. My hypothesis is confirmed via primary data from tests and secondary research, which proves that a blockchain-based system has stronger data integrity than that of the traditional system. Furthermore, I confirm my hypothesis by developing a system that reduces the number of third-parties required for a single transaction transaction via developing a single-purpose blockchain to handle settlements. Because a blockchain-based system only requires the validators (platform maintainer and owner) to operate without a broker or a Clearing House.

For managers, this opens the opportunity for the tokenization of assets and exchange of those assets through a centralized and trusted system, which would also reduce the risks that were present in the traditional system. Additionally, this will make transactions easy to record.

For future research directions, I propose the possibility for implementation of smart contracts which provides the opportunity for the development of a broker smart contract. This smart contract will operate on top of the blockchain developed in this research study, which will add the freedom for investors to develop their own broker that will work within the boundaries set by the exchange owners, allowing investors to use the precise logic that they desire. With the same framework, it is also possible to develop an alternative dividend payment system. Additionally, I discuss the potential for a primary market development that implements an action function. For the privacy issue that stems from the blockchain's public nature, it is worth researching the possibilities for implementing the zk-SNARKs algorithm that will hide all the important information regarding who invested how much to which firm.



# Bibliography

- Aditya Trading Solutions (2017). *Trading mechanism of securities | Trading cycle*, BSE, NSE | ATS. URL: <https://adityatrading.in/knowledge/introduction/trading-mechanism-of-securities.aspx> (visited on 01/04/2020).
- Aitken, Michael J., Frederick H. Frederick, and Shan Ji (2015). "A Worldwide Examination of Exchange Market Quality: Greater Integrity Increases Market Efficiency". In: *Journal of Business Ethics* 132.1, pp. 147–170. ISSN: 15730697. DOI: 10.1007/s10551-014-2294-5.
- Antonopoulos, Andreas M. (2017). *Mastering Bitcoin : programming the open blockchain*. 2nd. O'Reilly Media, p. 371. ISBN: 1491954388.
- Aumasson, Jean-Philippe (2017). *Aura Consensus Protocol Audit*. URL: <https://github.com/poanetwork/wiki/wiki/Aura-Consensus-Protocol-Audit> (visited on 11/21/2019).
- Bambara, Joseph J. and Paul R. Allen (2018). *Blockchain Practical Developing Technology Solutions*. United States.
- Barinov, Igor (2018). *What is POA · poanetwork/wiki Wiki*. URL: <https://github.com/poanetwork/wiki/wiki/What-is-POA> (visited on 11/21/2019).
- Bashir, Imran (2018). *Mastering Blockchain : Distributed ledger technology, decentralization, and smart contracts explained, 2nd Edition*. Packt Publishing, p. 656. ISBN: 9781788839044.
- Beattie, Andrew (2019). *The Birth of Stock Exchanges*. URL: <https://www.investopedia.com/articles/07/stock-exchange-history.asp> (visited on 12/28/2019).
- Bhandarkar, Vinith V, Akshay A Bhandarkar, and Aditya BE Shiva (2019). "Digital Stocks Using Blockchain Technology the Possible Future of Stocks?" In: *International Journal of Management (IJM)* 10.3, pp. 44–49. ISSN: 0976-6510. URL: <http://www.iaeme.com/IJM/index.asp44http://www.iaeme.com/IJM/issues.asp?JType=IJM{\&}VType=10{\&}IType=3JournalImpactFactorwww.jifactor.comhttp://www.iaeme.com/IJM/issues.asp?JType=IJM{\&}VType=10{\&}IType=3>.
- Bramble, Laura (2018). *How the Stock Market Was Started & by Whom*. URL: <https://smallbusiness.chron.com/stock-market-started-whom-14745.html> (visited on 01/03/2020).
- Briganti, Will and Ryan Wells (2015). *Nasdaq Linq Enables First-Ever Private Securities Issuance Documented With Blockchain Technology | Nasdaq, Inc.* URL: <http://ir.nasdaq.com/news-releases/news-release-details/nasdaq-linq-enables-first-ever-private-securities-issuance> (visited on 11/19/2019).
- Brown, Richard (2014). *A Simple Explanation of How Shares Move Around the Securities Settlement System | Richard Gendal Brown*. URL: <https://gendal.me/2014/01/05/a-simple-explanation-of-how-shares-move-around-the-securities-settlement-system/> (visited on 01/04/2020).
- Buterin, Vitalik (2018). *Sharding FAQ · ethereum/wiki Wiki*. URL: <https://github.com/ethereum/wiki/wiki/Sharding-FAQ> (visited on 12/29/2019).

- Cata, Justin (2018). *Everything to know about Ripple — Part 1: How Ripple Works*. URL: <https://medium.com/@jcata018/everything-to-know-about-ripple-part-1-how-ripple-works-f7404aa4a8d1> (visited on 01/05/2020).
- Chen, James (2018). *Investopedia - Shares*. URL: <https://www.investopedia.com/terms/s/shares.asp> (visited on 11/22/2019).
- (2019a). *Execution Definition*. URL: <https://www.investopedia.com/terms/e/execution.asp> (visited on 11/22/2019).
- (2019b). *Nasdaq Definition*. URL: <https://www.investopedia.com/terms/n/nasdaq.asp> (visited on 01/03/2020).
- (2019c). *Primary Market*. URL: <https://www.investopedia.com/terms/p/primarymarket.asp> (visited on 01/04/2020).
- Chiu, Jonathan and Thorsten V. Koepl (2018). *Blockchain-Based Settlement for Asset Trading*. DOI: 10.1093/rfs/hhy122.
- Clohessy, Trevor and Thomas Acton (2019). “Investigating the influence of organizational factors on blockchain adoption An innovation theory perspective The influence of organizational factors”. In: *Industrial Management & Data Systems* 119,7, pp. 1457–1491. DOI: 10.1108/IMDS-08-2018-0365. URL: [www.emeraldinsight.com/0263-5577.htm](http://www.emeraldinsight.com/0263-5577.htm).
- Cohen, Noam (2008). *This Is Funny Only if You Know Unix - The New York Times*. URL: <https://www.nytimes.com/2008/05/26/business/media/26link.html> (visited on 11/20/2019).
- Cohn, Gary (2015). *Clearing houses reduce risk, they do not eliminate it* | *Financial Times*. URL: <https://www.ft.com/content/974c2c48-16a5-11e5-b07f-00144feabdc0> (visited on 01/04/2020).
- Comerton-Forde, Carole, Talis J. Putniņš, and Kar Mei Tang (2011). “Why do traders choose to trade anonymously?” In: *Journal of Financial and Quantitative Analysis* 46.4, pp. 1025–1049. ISSN: 00221090. DOI: 10.1017/S0022109011000214.
- Cong, Lin William and Zhiguo He (2019). “Blockchain Disruption and Smart Contracts”. In: *Review of Financial Studies* 32.5, pp. 1754–1797. ISSN: 14657368. DOI: 10.1093/rfs/hhz007.
- Corporate Finance Institute (2015). *Corporate Finance - Overview of Main Activities in Corporate Finance*. URL: <https://corporatefinanceinstitute.com/resources/knowledge/finance/corporate-finance-industry/> (visited on 01/09/2020).
- Curran, Brian (2018). *What is Proof of Authority Consensus? (PoA) Staking Your Identity*. URL: <https://blockonomi.com/proof-of-authority/> (visited on 11/21/2019).
- Dannen, Chris (2017). *Introducing ethereum and solidity: Foundations of cryptocurrency and blockchain programming for beginners*. Apress Media LLC, pp. 1–185. ISBN: 9781484225356. DOI: 10.1007/978-1-4842-2535-6.
- Eluard, Julien (2019). *Substrate Developer Hub Docs: Extrinsics*. URL: <https://substrate.dev/docs/en/overview/extrinsics> (visited on 11/13/2019).
- European Parliamentary Research Service (2019). *Blockchain and the General Data Protection Regulation Can distributed ledgers be squared with European data protection law?* Tech. rep. Scientific Foresight Unit (STOA). DOI: 10.2861/535. URL: <http://www.europarl.europa.eu/thinktank>.
- FinTech Futures (2017). *Analysis: the trend toward best practice in institutional FX settlement – FinTech Futures*. URL: <https://www.fintechfutures.com/2017/09/analysis-the-trend-toward-best-practice-in-institutional-fx-settlement/> (visited on 01/09/2020).
- FLETA Blockchain (2019). *Cryptocurrency and Blockchain Regulations around the World (Part 1)*. URL: <https://medium.com/fleta-first-chain/cryptocurrency-and->

- blockchain-regulations-around-the-world-part-1-a07a890ceb6f (visited on 12/13/2019).
- Frankenfield, Jake (2019). *Initial Coin Offering (ICO) Definition*. URL: <https://www.investopedia.com/terms/i/initial-coin-offering-ico.asp> (visited on 12/14/2019).
- French, Jordan (2018). *Nasdaq Exec: Exchange Is 'All-In' on Using Blockchain Technology - TheStreet*. URL: <https://www.thestreet.com/investing/nasdaq-all-in-on-blockchain-technology-14551134> (visited on 11/19/2019).
- Ganti, Akhilesh (2019). *Clearing House Definition*. URL: <https://www.investopedia.com/terms/c/clearinghouse.asp> (visited on 01/04/2020).
- Global Law Library of Congress (2018). *Regulation of Cryptocurrency Around the World*. Tech. rep. Global Law Library of Congress. URL: <http://www.law.gov>.
- Grody, Allan D. (2017). *What's really holding back blockchain in financial services*. URL: <https://www.americanbanker.com/opinion/whats-really-holding-back-blockchain-in-financial-services> (visited on 11/20/2019).
- Hayes, Adam (2019). *How Does the Stock Market Work?* URL: <https://www.investopedia.com/articles/investing/082614/how-stock-market-works.asp> (visited on 01/04/2020).
- Huillet, Marie (2019). *Vitalik Buterin Talks Scalability: 'Ethereum Blockchain Is Almost Full'*. URL: <https://cointelegraph.com/news/vitalik-buterin-talks-scalability-ethereum-blockchain-is-almost-full> (visited on 12/30/2019).
- IFCI Risk Institute (2004). *Sources and Types of Risk to an Exchange Clearing House*. URL: <http://ifci.ch/138700.htm> (visited on 01/04/2020).
- Investopedia Staff (2019). *What Do T+1, T+2, and T+3 Mean?* URL: <https://www.investopedia.com/ask/answers/what-do-t1-t2-and-t3-mean/> (visited on 12/28/2019).
- Kashitsyn, Dmitriy (2018). *Substrate in a nutshell | Parity Technologies*. URL: <https://www.parity.io/substrate-in-a-nutshell/> (visited on 11/20/2019).
- Kelleher, John (2014). *Medici, The Blockchain Stock Exchange*. URL: <https://www.investopedia.com/articles/investing/121014/medici-blockchain-stock-exchange.asp> (visited on 01/05/2020).
- Kennon, Joshua (2019). *How Asset Management Companies Work*. URL: <https://www.thebalance.com/asset-management-companies-for-beginners-4048203> (visited on 01/09/2020).
- Kenton, Will (2019). *Secondary market Importer*. URL: <https://www.investopedia.com/terms/s/secondarymarket.asp> (visited on 11/20/2019).
- Khandelwal, Nikhil (2018). *Nasdaq Inc. (NASDAQ:NDAQ) and Blockchain Technology*. URL: <https://blockchainstocks.com/nasdaq-nasdaq-ndaq-and-blockchain-technology/> (visited on 11/19/2019).
- Kim, Henry and Marek Laskowski (2017). "A perspective on blockchain smart contracts: Reducing uncertainty and complexity in value exchange". In: *2017 26th International Conference on Computer Communications and Networks, ICCCN 2017*. Institute of Electrical and Electronics Engineers Inc. ISBN: 9781509029914. DOI: 10.1109/ICCCN.2017.8038512.
- Klabnik, Steve and Carol Nichols (2019). *The Rust Programming Language*. No Starch Press. ISBN: 978-1-7185-0044-0.
- Laboon, Bill (2019). *Staking · Polkadot Wiki*. URL: <https://wiki.polkadot.network/docs/en/learn-staking> (visited on 12/16/2019).
- Lioudis, K., Nick (2019). *Why the Forex Market Is Open 24 Hours a Day*. URL: <https://www.investopedia.com/ask/answers/how-forex-market-trade-24-hours-day/> (visited on 11/15/2019).

- Mitchell, Cory (2019). *Buy Limit Order Definition and Example*. URL: <https://www.investopedia.com/terms/b/buy-limit-order.asp> (visited on 11/20/2019).
- Murphy, Chris (2019). *Over-The-Counter – OTC Definition*. URL: <https://www.investopedia.com/terms/o/otc.asp> (visited on 12/27/2019).
- Nakamoto, Satoshi (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Tech. rep. URL: [www.bitcoin.org](http://www.bitcoin.org).
- Osuke (2019). *Announcing Zerochain: Applying zk-SNARKs to Substrate*. URL: <https://medium.com/layerx/announcing-zerochain-5b08e158355d> (visited on 11/15/2019).
- Parity Technology (2019). *Polkadot Consensus · Polkadot Wiki*. URL: <https://wiki.polkadot.network/docs/en/learn-consensus> (visited on 11/20/2019).
- Parsons, Joe (2018). *SGX, Nasdaq and Singapore regulator to develop blockchain settlement system*. URL: <https://www.thetradenews.com/sgx-nasdaq-singapore-regulator-develop-blockchain-settlement-system/> (visited on 11/19/2019).
- Petrowski, Joe (2019). *Substrate Runtime Module Library · Substrate Developer Hub*. URL: <https://substrate.dev/docs/en/runtime/substrate-runtime-module-library> (visited on 11/20/2019).
- Piccolo, Alessandro (2017). “Distributed ledger technology in the capital market”. In: *HSBC Securities Services*.
- Pisani, Bob (2019). *‘Tape’ glitch means it’s not clear where the Dow and S&P closed Monday*. URL: <https://www.cnbc.com/2019/08/13/stock-tape-glitch-means-its-still-not-exactly-clear-where-the-dow-sp-500-closed-on-monday.html> (visited on 11/15/2019).
- Poon, Joseph and Vitalik Buterin (2017). *Plasma: Scalable Autonomous Smart Contracts*. Tech. rep. Plasma.io. URL: <https://plasma.io/>.
- Pop, Claudia et al. (2018). “Decentralizing the stock exchange using blockchain an ethereum-based implementation of the bucharest stock exchange”. In: *Proceedings - 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing, ICCP 2018*. IEEE, pp. 459–466. ISBN: 9781538684450. DOI: 10.1109/ICCP.2018.8516610.
- Reitwiessner, Christian (2016). “zkSNARKs in a Nutshell”. In: *Ethereum Blog*, pp. 1–15. URL: <https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/>.
- Santo, Atsushi et al. (2016). *Applicability of Distributed Ledger Technology to Capital Market Infrastructure*. Tech. rep. Japan Exchange Group.
- Schulz, Fabian (2019). *A brief summary of everything Substrate and Polkadot*. URL: <https://medium.com/polkadot-network/a-brief-summary-of-everything-substrate-and-polkadot-f1f21071499d> (visited on 11/20/2019).
- Securities and Exchange Commission (2013). *Trade Execution: What Every Investor Should Know*. URL: <https://www.sec.gov/reportspubs/investor-publications/investorpubstradexechtm.html> (visited on 11/20/2019).
- Seijas, Pablo Lamela, Simon Thompson, and Darryl Mcadams (2016). *Scripting smart contracts for distributed ledger technology*. Tech. rep.
- Simpson, Stephen (2019). *The Death Of The Trading Floor*. URL: <https://www.investopedia.com/financial-edge/0511/the-death-of-the-trading-floor.aspx> (visited on 01/03/2020).
- S&P Global (2019). *What Blockchain Could Mean For Structured Finance*. URL: <https://www.spglobal.com/en/research-insights/articles/what-blockchain-could-mean-for-structured-finance> (visited on 11/15/2019).
- Trustnodes (2018). *Hackers Present Stock Trading Prototype on Ethereum Smart Contracts*. URL: <https://www.trustnodes.com/2018/10/09/hackers-present-stock-trading-prototype-on-ethereum-smart-contracts> (visited on 11/15/2019).

- Wall Street (2018). *How Stock Trading Has Evolved* - Wall-Street.com. URL: <https://wall-street.com/stock-trading-evolved/> (visited on 12/28/2019).
- Witherspoon, Zane (2017). *A Hitchhiker's Guide to Consensus Algorithms* - By. URL: <https://hackernoon.com/a-hitchhikers-guide-to-consensus-algorithms-d81aae3eb0e3> (visited on 12/29/2019).
- Wood, Gavin (2017). "Polkadot: Vision for a Heterogeneous Multi-Chain Framework". In: *Whitepaper*, pp. 1–21. URL: <https://github.com/w3f/polkadot-white-paper/raw/master/PolkaDotPaper.pdf>.
- Yafimava, Darya (2019). *Blockchain And The Law: Regulations Around the World*. URL: <https://openledger.info/insights/blockchain-law-regulations/> (visited on 12/13/2019).