



**FACULTY  
OF INFORMATION  
TECHNOLOGY  
CTU IN PRAGUE**

## ASSIGNMENT OF MASTER'S THESIS

**Title:** Cryptocurrencies exchange rates reporting tool  
**Student:** Bc. Adam Pečev  
**Supervisor:** doc. Ing. Štěpán Starosta, Ph.D.  
**Study Programme:** Informatics  
**Study Branch:** Web and Software Engineering  
**Department:** Department of Software Engineering  
**Validity:** Until the end of summer semester 2018/19

### Instructions

1. Survey existing applications which collect data about cryptocurrencies and exchange rates between them. Focus on the following cryptocurrencies: Bitcoin, Monero, Ethereum.
2. Design and implement an application which collects data on exchange rates of selected cryptocurrencies on selected markets. The application will be able to find the best sequences of exchanges (using the collected data) which satisfies a criterion input by the user. The available criteria include positive profit on the found path and identical first and last currency. The application shall work with real-time data in real-time, allow working with historical data and analyse these results; it shall have a GUI and visualize the results (that is, the found path).
3. Test the application.

### References

Will be provided by the supervisor.

Ing. Michal Valenta, Ph.D.  
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
Dean

Prague February 16, 2018





**FACULTY  
OF INFORMATION  
TECHNOLOGY  
CTU IN PRAGUE**

Master's thesis

# **Cryptocurrencies Exchange Rates Reporting Tool**

*Bc. Adam Pečev*

Department of Software Engineering  
Supervisor: doc. Ing. Štěpán Starosta, Ph.D.

May 9, 2019



---

## **Acknowledgements**

I would like to thank doc. Ing. Štěpán Starosta, Ph.D., my supervisor, for all his time and advice given. Moreover, I would like to thank my family and my Honey Bunny for full support during my studies.



---

# Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the “Work”), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work for non-profit purposes only, in any way that does not detract from its value. This authorization is not limited in terms of time, location and quantity.

In Prague on May 9, 2019

.....

Czech Technical University in Prague

Faculty of Information Technology

© 2019 Adam Pečev. All rights reserved.

*This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).*

### **Citation of this thesis**

Pečev, Adam. *Cryptocurrencies Exchange Rates Reporting Tool*. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2019.



---

# Abstrakt

Tato práce popisuje současný trh s kryptoměny. Následně definuje atributy kryptoměnových burz, které jsou relevantní pro provedení trojúhelníkové arbitráže. Na základě těchto atributů jsou analyzovány vybrané kryptoměnové burzy. Dále jsou specifikovány klíčové požadavky důležité pro vytvoření aplikace. Práce se také zabývá již existujícími aplikacemi. Poté je navržena a implementována nová aplikace. Implementace zahrnuje stahování dat z API třetích stran, zpracování dat a zobrazení potenciálních příležitostí vykonání trojúhelníkové arbitráže v reálném čase. Navíc umožňuje uživateli porovnávat jednotlivé kryptoměnové burzy v kontextu historických příležitostí vykonání trojúhelníkové arbitráže.

**Klíčová slova** kryptoměna, směnný kurz, kryptoměnová burza, trojúhelníková arbitráž, Bitcoin, aplikace v reálném čase

---

# Abstract

This work describes present cryptocurrency market. Then it defines attributes of cryptocurrency exchanges relevant to perform triangular arbitrage. On the basis of these attributes selected cryptocurrency exchanges are analysed. The core requirements important for the application are specified. The work also discusses the already existing applications. After that the new application is designed and implemented. The implementation includes downloading data from 3rd party APIs, processing them and displaying potential triangular arbitrage execution opportunities in real-time. Moreover, it allows a user to compare individual cryptocurrency exchanges considering the historical opportunities of triangular arbitrage execution.

**Keywords** cryptocurrency, exchange rate, cryptocurrency exchange, triangular arbitrage, Bitcoin, real-time application

---

# Contents

Citation of this thesis . . . . .	vi
<b>Introduction</b>	<b>1</b>
<b>1 Objectives</b>	<b>3</b>
1.1 Assignment Analysis . . . . .	3
1.2 Used Terms . . . . .	5
1.3 Motivation . . . . .	6
<b>2 Survey</b>	<b>7</b>
2.1 Cryptocurrencies . . . . .	7
2.1.1 Bitcoin . . . . .	7
2.1.2 Other Cryptocurrencies . . . . .	12
2.2 Cryptocurrency Market . . . . .	15
2.3 Cryptocurrency Exchanges . . . . .	15
2.4 Existing Solutions . . . . .	25
<b>3 Analysis</b>	<b>29</b>
3.1 Problem Definition . . . . .	29
3.2 Functional Requirements . . . . .	32
3.3 Non-Functional Requirements . . . . .	34
3.4 Use Cases and Scenarios . . . . .	38
<b>4 Design</b>	<b>43</b>
4.1 Selected Technologies . . . . .	43
4.2 Deployment Model . . . . .	49
4.3 Wireframes . . . . .	50
4.4 Application Structure and Algorithm . . . . .	52
<b>5 Implementation</b>	<b>59</b>
5.1 Client Side . . . . .	59

5.2	Server Side . . . . .	61
5.3	Triangular Arbitrage Library . . . . .	62
5.4	Initializer . . . . .	65
<b>6</b>	<b>Testing</b>	<b>67</b>
	<b>Conclusion</b>	<b>71</b>
	<b>Bibliography</b>	<b>73</b>
<b>A</b>	<b>Acronyms</b>	<b>87</b>
<b>B</b>	<b>GUI Snapshots</b>	<b>91</b>
<b>C</b>	<b>User's Manual</b>	<b>95</b>
<b>D</b>	<b>Contents of enclosed CD</b>	<b>97</b>

---

## List of Figures

2.1	Crypto Arbitrage Opportunity Expert website snapshot . . . . .	27
2.2	TriangularArbitrage application snapshot . . . . .	27
3.1	Triangular arbitrage scenario . . . . .	30
3.2	Use case diagram . . . . .	39
4.1	Polling, Long Polling and SSE comparison diagram . . . . .	47
4.2	WebSocket communication diagram . . . . .	47
4.3	Deployment diagram . . . . .	51
4.4	Wireframe of the <i>Real-time data page</i> . . . . .	53
4.5	Wireframe of the <i>Historical data page</i> . . . . .	54
4.6	Class diagram of the Triangular arbitrage library . . . . .	57
B.1	GUI snapshot of the <i>Real-time data page</i> . . . . .	92
B.2	GUI snapshot of the <i>Historical data page</i> with the chart . . . . .	93
B.3	GUI snapshot of the <i>Historical data page</i> with the table . . . . .	94



---

## List of Tables

2.1	Top cryptocurrencies by the market. cap. . . . .	14
3.1	Functional requirements . . . . .	32
3.2	Non-functional requirements . . . . .	35
3.3	Use cases . . . . .	38
3.4	Mapping of the use cases onto the functional requirements . . . . .	42





---

# Introduction

Since 2009, when Bitcoin as the first decentralized cryptocurrency and open-source software was released, a lot of various projects have been introduced. These projects involve new cryptocurrencies like Litecoin, Monero, Ethereum and hundreds of others, cryptocurrency exchanges, cryptocurrency markets, bots, whole new companies based on cryptocurrencies, etc. One of them – cryptocurrency exchanges – has attracted a lot of attention during last years. There are huge amounts of money on these exchanges [1], and this fact attracts people who want to somehow benefit from the market. On the one hand many frauds have happened [2, 3, 4] but, on the other hand, the whole cryptocurrency market expanded very much since then [5]. Even though the cryptocurrency exchanges operating Bitcoin and other cryptocurrencies are not so similar to the Foreign Exchange Market [6], there also do exist market inefficiencies [7, 8] from which one can benefit.

Apart from exploiting basic arbitrage opportunities<sup>1</sup>, calendar effects<sup>2</sup> or present economic bubbles driven by diverse incentives, one can also look further. One of the other possibilities is to look into multiple exchange rates on given cryptocurrency exchanges. The topic of this thesis is to visualise how one can take advantage of these exchange rates by collecting data on cryptocurrency trading pairs. It will be observed how the prices of these trading pairs change and we will attempt to make use of it by using triangular arbitrage [9, 10, 11]. The application will work with real-time data as well as with historical data. The final visualisation does not have to be used only for informational purposes but could be used for initiating a real trade as well.

---

<sup>1</sup><https://files.stlouisfed.org/files/htdocs/publications/review/02/11/EmmonsSchmid.pdf>

<sup>2</sup><http://aeconf.com/articles/may2005/aef060105.pdf>



---

# Objectives

The main objective of this thesis is to explore a possibility of making a profit from gathering data of various exchange rates on the cryptocurrency exchange market. The gathered data will be then processed and visualised for a user. The whole process includes five main phases – survey, analysis, design, implementation and testing. We will mainly focus on fine survey, analysis and design which are supposed to be the building blocks of the basic visualisation function. After it is possible to visualise the situation one will be able to build on top of it and make use of it – to trade the cryptocurrency if the profits are positive. On the other hand, if the profits are not positive then the thesis will show it is not worth using this technique on given cryptocurrency exchanges with given trading pairs.

## 1.1 Assignment Analysis

In this subchapter we describe the assignment in more detail and we focus on the best way how to accomplish that in its fullness. The next paragraph adds information about the Survey chapter. After that we state every part of the thesis assignment description and then we comment on it.

At the beginning of the Survey chapter we will write about cryptocurrencies as a phenomenon and quickly introduce the main and first cryptocurrency called Bitcoin. Then we will comment on cryptocurrency market as a whole – what types of the market are there and how large it is.

1. *Survey existing applications which collect data about cryptocurrencies and exchange rates between them. Focus on the following cryptocurrencies: Bitcoin, Monero, Ethereum.*

We will survey some of the existing websites and software applications which collect data about cryptocurrencies as Bitcoin, Ethereum, Monero, etc. and the rates between them and alternative cryptocurrencies. That means we will survey mainly the cryptocurrency exchanges with

## 1. OBJECTIVES

---

the APIs they provide. Then we will also survey programs which offer services like arbitrage trading or other techniques that involve exploiting various market inefficiencies.

2. *Design and implement an application which collects data on exchange rates of selected cryptocurrencies on selected markets. The application will be able to find the best sequences of exchanges (using the collected data) which satisfies a criterion input by the user. The available criteria include positive profit on the found path and identical first and last currency.*

Apart from the design and the implementation we will also analyse which functional and non-functional requirements are appropriate for the application. Furthermore, we will analyse the use cases. After the formulation of these foundations and possible options the technologies as well as the wireframes will be determined. Based on the given technologies the deployment model and the application structure will be designed. Finally, the implementation will be described.

3. *The application shall work with real-time data in real-time, allow working with historical data and analyse these results;*

In order to ensure that the application could be used as a basis for possible extension in the future, the real-time aspect of the collected data is needed. The real-time data will be processed in real-time and transferred into useful information flow. To make sure the historical data are correct a reliable source will have to be chosen. One will be able to view the collected historical data in given context.

4. *it shall have a GUI and visualise the results (that is, the found path).*

A Graphical User Interface will be integral to the whole application. As a matter of fact, it is a logical implication of the informational purpose of the thesis objective. The GUI will show relation between the trading pairs as well as the historical data.

5. *Test the application.*

A part of the thesis will also consist of the tests of the application – the usability testing.

## 1.2 Used Terms

### **Altcoin**

A combination of two words – *alt* and *coin* – meaning alternative cryptocurrency. Generally, it is a cryptocurrency other than Bitcoin [12]. Often it is also used for other than the most popular cryptocurrencies in terms of market capitalization.

### **Cryptocurrency exchange**

A trading exchange as an organized market where one can trade cryptocurrencies or other digital currencies or digital assets [13, p. 30][14].

### **Trading pairs**

A pair which represents two different units of value – in this case two different cryptocurrencies or one cryptocurrency and one national currency. A typical example is BTC/USD.

### **Triangulation**

A formation of a closed circle which represents a series of consecutive or even parallel buy orders. Used also for possibly profitable – in terms of triangular arbitrage – set of trading pairs.

### **Exchange market**

A trading exchange as an organized market where one can trade national currencies or derivatives. It can be a cryptocurrency exchange, Foreign Exchange Market<sup>3</sup>, etc.

### **Trading bot**

A computer program which evaluates a situation on a cryptocurrency exchange and acts on behalf of a user – the owner. It analyses, offers various recommendations and even buys or sells a (crypto)currency.

### **Application**

The software application, which is one of the objectives of this thesis, as a whole – both front-end and back-end.

---

<sup>3</sup><https://www.forex.com/en-uk/education/education-themes/trading-concepts/what-is-forex/>

### 1.3 Motivation

Cryptocurrency is a phenomenon which I have been interested in for years. The more I started to dive into this field the more I came across various cryptocurrency exchanges. After that at some point I started trading and started thinking about different market inefficiencies and how to exploit them. Since my major is Software Engineering, one of the exploitations that came to my mind has been an arbitrage made automatically by a bot. That led me to the triangular arbitrage. Since the success of the triangular arbitrage depends also on a response speed of a trader, a software solution might be the right way how to deal with this practice. Another part of my motivation came from the fact that I always wanted to connect my interest in cryptocurrencies with my university studies.

---

# Survey

In this section the context to the whole diploma thesis is given. There is a brief introduction of Bitcoin and other cryptocurrencies. Then the current major cryptocurrency markets and cryptocurrency exchanges are surveyed. In addition, we describe how the price of cryptocurrencies is created and how the exchanges provide information about trading through their API. We also present some of the existing software applications and websites which collect data about the cryptocurrencies such as Bitcoin, Ethereum, Monero and the rates among them.

## 2.1 Cryptocurrencies

This subchapter describes Bitcoin, the technology and principles behind it and Bitcoin difficulties in brief. Next to Bitcoin, it also describes alternatives to Bitcoin and difficulties of such alternatives. This subchapter is not indispensable for understanding of the realization of the work's objective. Yet, it provides information which ensures a comprehensive view and big picture – how the cryptocurrency works behind the buy and sell orders on a cryptocurrency exchange.

### 2.1.1 Bitcoin

Bitcoin is the world's first invented decentralized cryptocurrency [15]. It is a peer-to-peer, open-source worldwide payment system with current market capitalization of around \$76 billions [16]. It was invented by an unknown person or a group called Satoshi Nakamoto in 2008 and the open-source code was released in 2009 [17, 18]. Since then, many other cryptocurrencies have been created and launched. One of the Satoshi Nakamoto's incentives to create Bitcoin were an independence of centralized authority and a privacy protection [19]:

“The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the ‘tape’, is made public, but without telling who the parties were.”

### **Blockchain**

The main technology behind Bitcoin is blockchain. One can imagine the blockchain as a public ledger [20, 21, 22]. It records all the Bitcoin transactions which have happened within the payment network. The blockchain consists of blocks which are connected in a way that one block contains a hash of the previous block header, the previous block contains a hash of the block header before the previous block and so on. One can imagine this scheme as a linear connected sequence of blocks.

### **Block and Transactions**

The first block (also called the genesis block [23]) was created in 2009 [24]. The blocks are files which store transaction data. The block data can be divided into the following parts [25]:

- Magic number – value is always 0xD9B4BEF9 (4 bytes)<sup>4</sup>
- Blocksize – number of bytes following up to end of block (4 bytes)
- Blockheader – consists of other 6 items (80 bytes)
- Transaction counter – positive integer (1–9 bytes)
- Transactions – the non-empty list of transactions

The transactions are units as well. They are comprised of several items, such as: how much money is sent, from which address the money is sent, to which address the money is sent, etc., including witnesses [26]. The witnesses are data which are connected to signatures needed in order to verify the transactions [27]. These signature-related data were segregated from the block in 2017 because of the problems in the Bitcoin network as it is mentioned below in the Difficulties of Bitcoin.

---

<sup>4</sup><https://github.com/bitcoin/bitcoin/blob/master/src/chainparams.cpp>



## Wallet

One of the important aspects of bitcoin ownership is that an owner does not have the cryptocurrency (e.g. 10 bitcoins) on himself or herself. The owner possesses private keys which are required in order to be able to manipulate cryptocurrency on specific addresses.

When a user wants to send certain number of bitcoins the simplest way is to download a software wallet. According to [28], the main purpose of the software wallets is to create public keys and store the corresponding private keys which represent an ownership of a cryptocurrency. There are two main software wallet types:

1. desktop wallet – e.g. Bitcoin Core<sup>5</sup>, Electrum<sup>6</sup>, Exodus<sup>7</sup>
2. mobile wallet – e.g. Mycelium<sup>8</sup>, Coinomi<sup>9</sup>, Copay<sup>10</sup>

Some software solutions – wallets – can handle more cryptocurrencies than just one (e.g. Coinomi). They ensure sending transactions and recording new received transactions of the cryptocurrency. One can usually also see the transaction history and set transaction fees. When a new sending of a cryptocurrency is initiated the wallet creates a new transaction. This transaction is broadcast by the wallet to the broadcast network over TCP [29]. There are some disadvantages of the software solutions, e.g. the software is installed on an operating system on a computer which may be often connected to the internet. The computer can be infected with a malware and this malware can exploit this situation which could result in private keys theft [30, 31, 32]:

- It could take screenshots of the computer screen including the exposed private keys.
- If the software wallet is not encrypted the potential malware could send a command to the software in order to send the cryptocurrency to a specific address as soon as the software is launched.
- If the private keys are not encrypted (or incorrectly encrypted) a malware could steal them.

In addition, there are also hardware (offline) wallets, e.g., Trezor<sup>11</sup> or Ledger<sup>12</sup>. They are supposed to be safer than the software wallets. The reason is that

---

<sup>5</sup><https://bitcoin.org/en/bitcoin-core/>

<sup>6</sup><https://electrum.org/#home>

<sup>7</sup><https://www.exodus.io/>

<sup>8</sup><https://wallet.mycelium.com/>

<sup>9</sup><https://www.coinomi.com/en/>

<sup>10</sup><https://copay.io/>

<sup>11</sup><https://trezor.io/>

<sup>12</sup><https://www.ledger.com/>

they are securely programmed stand-alone pieces of hardware which require a specific procedure in order to send cryptocurrency. They isolate the private keys of the cryptocurrency addresses on a secure chip when the device is connected (usually using USB) to the computer with internet connection [33]. When one wants to send cryptocurrency, a transaction is created in the affiliated application and subsequently sent to the device (hardware wallet) where it needs to be signed. The signature is done only when the user confirms the transaction on the device.

The last type of a wallet is a normal piece of paper (or an object like Cryptosteel<sup>13</sup>). It can carry the private keys or a Backup Seed Phrase which is usually a list of words sufficient enough to recover a wallet [34]. This type of a wallet is not secure on its own. There is some other supplementary item which provides safety of this type of wallet needed. For example a safe box where one can deposit the paper or the Cryptosteel-like item so nobody else can read it.

### **Network and Nodes**

The broadcast network is comprised of peers (nodes). In reality a node is usually a server or a storage device which has the Bitcoin client software installed. The node stores the entire blockchain as well [35, 36]. It also must be connected to the internet. The nodes communicate with each other and if they receive a message and evaluate it as a valid transaction, they broadcast it as well so the transaction is received by all the nodes. However, if this particular transaction has already been broadcast, they never broadcast it again. The nodes act also as checkers of incoming transactions – they check that the transactions comply with the Bitcoin protocol. Otherwise they reject them.

### **Miners, Nonce, Target and Proof of Work**

Another important component of the Bitcoin system are the miners. According to [37], they have two main purposes:

1. The primary purpose of miners is to record history of transactions in a way that is computationally and financially impractical to modify by anyone. That ensures security and confidence in the system. They basically hold the post of a distributed arbiter.
2. Secondary purpose is an issuance of new bitcoins.

Every miner needs to be connected to the network through one or more nodes and pick up the new transactions in the network. In addition, he or she needs

---

<sup>13</sup><https://cryptosteel.com/>

information about the last created block in the blockchain. Thus, some nodes are administered by the miners themselves, so they have immediate access to the broadcast network. When they are connected, they take the new transactions in the network (unconfirmed transactions or unordered transactions) and put them together along with other items such as Coinbase (Generation) transaction. Then they build a block header from these items [37]. One of the items in the block header is called Nonce. The miners hash this header and after hashing it they check if the hash is lower than the Target. The Target is a 256-bit number [38, 39, 40]. If it is lower than the Target, they just found (mined/generated) next block and broadcast this information to the network in the same way as the transactions are broadcast. This successful result is called Proof of Work [19, 41]. If the hash is not lower than the Target they change (usually just increase) the Nonce and hash the header again until they (or someone else) find the next block. The hashing algorithm that is used is SHA256 two times in a row –  $\text{SHA256}(\text{SHA256}(\text{block header}))$ . The lower the Target is the smaller chance to mine a block. For the reasons of low latency in transactions and transaction confirmation time predictability, the broadcast network strives for mining 1 block every 10 minutes [42, 43]. Since the attributes like the number of active miners and overall hashing power are changing over time, there also is a need to change the difficulty of finding the next block. In other words, the Target needs to be changed over time in order to comply with the 10-minute creation time. It is adjusted every 2016 mined blocks [44], i.e. approximately every 14 days. A successful miner gets two bonuses as a reward [19]:

1. The first one is the transaction fees included in the mined block.
2. The second one is a bounty of 12.5 BTC at the moment. This bounty is agreed upon by everyone in the network and it must comply with the Controlled Bitcoin Supply [45, 16, 46] – i.e. 50% reduction every 210,000 mined blocks.

This results in two important facts – the supply limit in total is 20,999,999.9769 bitcoins [47, 48] and 2140 is the year when the last bitcoins should be mined [49, p. 2].

### **Difficulties of Bitcoin**

Even Bitcoin has its drawbacks. Some of them appeared after a few years of existence and running. The adoption by the general public and so an increase of transactions per hour brought the scalability problem of Bitcoin [50, 51] to the surface. The core of the problem lays in two parameters:

- Average block creation time – Every 10 minutes on average is a block mined (created/found) and since the confirmed transactions are depen-

dent on the block mining there is only a limited number of transactions which can be confirmed in 10 minutes.

- Block size limit – Until 2017 was the system set as follows: 1 block can handle only 1 MB of data and consists of both data related to signatures (witness-related data) and the transactions. Since 2017 the Segregated Witness (SegWit) solution [27] by the inventor Pieter Wuille has been in practice. It segregates data related to signatures from the transactions data. It also cancels the block size concept and constructs a block weight concept which results in 4 MB as a maximum size of a block. All in all, the SegWit solution increases throughput of the Bitcoin network.

Another problem is the worldwide energy consumption of Bitcoin mining. For instance, according to [52], this consumption is comparable to the electricity consumption of whole Ireland.

### 2.1.2 Other Cryptocurrencies

There are hundreds of other cryptocurrencies apart from Bitcoin [53] – so-called Altcoins – e.g. Ethereum, Monero, Litecoin. They were launched usually after the Bitcoin's success and they are presented as:

1. better alternatives to Bitcoin
2. whole new cryptocurrencies based on different principles

A good representative of the first option, i.e., better alternatives to Bitcoin, is Litecoin<sup>14</sup> [54, 55]:

- It shows around 4x faster transaction processing.
- Its mining algorithm – Scrypt – is more suitable for basic user's CPUs compared to Bitcoin's SHA-256 which can be quickly calculated with the ASIC miners which complicates the network decentralization.
- Transaction fees are cheaper.

One of the representatives of the second option, i.e., new cryptocurrencies based on different principles, is Ethereum<sup>15</sup> [56]:

- Programmed to provide Smart Contracts – allowing to perform irreversible and traceable credible transactions without third parties involved. They are usually used to facilitate a transaction within the blockchain. That means it works much more as a platform than just a currency.

---

<sup>14</sup><https://litecoin.org/>

<sup>15</sup><https://www.ethereum.org/>

- Strives for changing into proof-of-stake system – that will change the rewarding system on the blockchain. It will also be more energy efficient and will be more about bandwidth capacity rather than hash rate. Lastly it should be also less competitive on the validator (former miner).
- The transaction confirmation lasts just seconds.

### Difficulties of Other Cryptocurrencies

Altcoins is a term which includes a wide range of different cryptocurrencies. On the one hand these can be interesting innovative projects with an active expanding community, but on the other hand they can also be pure scams. One of the main quality indicators is the white paper which is presented alongside the coin, a transparent active developer community and a competitive advantage. Next to the formal characteristics of the altcoin, in the beginning there is also a risk of Pump and Dump frauds [57, 58, 59]. The vulnerability occurs when low price and low market capitalization are present at the altcoin. A typical procedure is as follows:

- A closed group of people sets a time when everyone of them starts buying a predefined altcoin and spreading the news about the price movement (which is the “pump” part).
- That temporarily increases demand for the coin and also the price itself increases.
- After that the people in the closed group start selling the altcoin (which is the “dump” part) and the price decreases.

It is hard to stop this kind of activity because it is not illegal among the cryptocurrencies due to non-existent regulation. Despite the fact that it is difficult to see through this problem there are initiatives which try to show at least a suspicious behaviour of an altcoin as CoinCheckup<sup>16</sup>.

In order to give a big picture of the relationship between Bitcoin and the Altcoins – Monero, Ethereum and other cryptocurrencies – we attach a table 2.1 from CoinMarketCap website [60]. The table shows the top cryptocurrencies sorted by the market capitalization – it is comprised of a position, name, market capitalization<sup>17</sup>, price, volume<sup>18</sup> in the last 24 hours, circulating supply and a price graph. At first sight there is a similarity among various cryptocurrencies – Bitcoin, XRP, Ethereum, Stellar, Litecoin and Cardano – regarding the 7-day price graphs.

---

<sup>16</sup><https://pumpdump.coincheckup.com/>

<sup>17</sup><https://www.investopedia.com/terms/m/marketcapitalization.asp>

<sup>18</sup><https://www.investopedia.com/terms/v/volumeoftrade.asp>

Name	Market Cap	Price	Volume (24h)	Circulating Supply	Price Graph (7d)
Bitcoin	\$69,932,663,624	\$4,018.93	\$6,005,274,973	17,400,800 BTC	
XRP	\$14,652,245,641	\$0.3633333	\$523,970,380	40,327,341,704 XRP	
Ethereum	\$11,784,334,602	\$113.83	\$2,072,289,926	103,522,619 ETH	
Stellar	\$3,059,239,734	\$0.159714	\$96,417,066	19,154,499,889 XLM	
Bitcoin Cash	\$3,057,453,495	\$174.85	\$98,026,943	17,485,863 BCH	
EOS	\$2,607,337,029	\$2.88	\$849,686,463	906,245,118 EOS	
Litecoin	\$1,912,765,765	\$32.22	\$423,149,043	59,372,874 LTC	
Tether	\$1,841,411,846	\$0.991915	\$4,035,702,897	1,856,421,736 USDT	
Bitcoin SV	\$1,698,998,485	\$97.21	\$179,908,888	17,477,861 BSV	
Cardano	\$1,008,635,193	\$0.038903	\$30,192,241	25,927,070,538 ADA	
Monero	\$950,614,447	\$57.26	\$14,181,945	16,602,171 XMR	

Table 2.1: Top cryptocurrencies sorted by the market capitalization, from CoinMarketCap [60]

## 2.2 Cryptocurrency Market

As well as there are various physical and non-physical markets like farmers' market, e-commerce or Foreign exchange market (Forex), there is also a market with cryptocurrencies. This market has existed since the time people wanted to trade cryptocurrencies with each other. One can sell or buy (exchange) a cryptocurrency on a cryptocurrency market. It also works as a place where the prices of given cryptocurrencies are established.

The cryptocurrency market comprises many different types. The main one is definitely cryptocurrency exchange market. The total market capitalization according to CoinMarketCap [5] is over \$136 billion which is similar to some of the world's biggest companies like IBM [61]. There are some other smaller cryptocurrency markets as markets on the Darknet<sup>19</sup> [62] where cryptocurrencies can also be used as an anonymizing element of trading goods (including illegal ones). Other small cryptocurrency markets are the local ones. In the Czech Republic it could be places like Paralelní Polis<sup>20</sup> and events created by these local places and organizers. On these events people can learn about various cryptocurrencies and buy or sell them between each other without using any intermediaries like a cryptocurrency exchange as well. From the market capitalization and triangular arbitrage points of view, the most interesting are the cryptocurrency exchanges, and that is also the reason why we mainly focus on them from now on.

## 2.3 Cryptocurrency Exchanges

According to [63, p. 30] and [64, p. 2–3], one of the precursors of the triangular arbitrage opportunities are liquidity<sup>21</sup> and spread<sup>22</sup> on a market. The higher the liquidity on the market (with selected trading pairs), the better opportunity windows of triangular arbitrage may appear. The same but reversed applies to the spread as well – the lower the spread on the market (with selected trading pairs), the better the opportunity windows of triangular arbitrage may appear. These two concepts along with volume are connected with each other and affect each other [65]. In addition to liquidity and spread, there are other aspects which matter in order to execute a successful triangular arbitrage on a cryptocurrency exchange:

1. Trading pairs – There must be an opportunity to execute triangular arbitrage. The more trading pairs are available on a cryptocurrency exchange, the higher the possibility there will be an opportunity to create

---

<sup>19</sup>[https://en.wikipedia.org/wiki/Darknet\\_market](https://en.wikipedia.org/wiki/Darknet_market)

<sup>20</sup><https://www.paralelnipolis.cz/>

<sup>21</sup><https://www.investopedia.com/terms/l/liquidity.asp>

<sup>22</sup><https://www.investopedia.com/terms/s/spread.asp>

a triangle of cryptocurrencies. This attribute is more easily achieved on the cryptocurrency exchanges operating with various altcoins.

2. Trading fees – Trading fees have an influence on a success rate of triangular arbitrage opportunity. They differ across all the cryptocurrency exchanges. Some of them operate Maker-Taker model [66] (e.g. Kraken). It says that Takers are those who place an order which is immediately matched by an existing order. The Makers are the other ones – those who place an order which does not get filled immediately and is placed on the order book. Later it is matched by an order of another customer. The Taker fees are ordinarily higher than the Maker fees.
3. API capabilities – Since the cryptocurrency prices are changing every second or even faster, the communication between an API and a user is important. There can be various APIs provided and different protocols and architectures used by a cryptocurrency exchange in order to communicate with a user. The most common ones are these two:
  - REST – an architectural style based on HTTP protocol [67]
  - WebSocket – a communication protocol dependent on TCP [68]

REST API calls are usually limited on the number of requests that can be executed. WebSocket APIs work on a notification basis (or Push technology<sup>23</sup>) by definition. That means that a user is notified by the server when there is an update of data which the user subscribes to. Both approaches may be suitable – the bottom line is which cryptocurrency exchange is used and which functionality a user needs.

On the grounds of all the above-mentioned factors, several cryptocurrency exchanges and their attributes are discussed below. The attributes are the following:

- basic information about a cryptocurrency exchange
- volume ranking which indicates liquidity and spread [65] on a cryptocurrency exchange
- trading pairs listed on a cryptocurrency exchange
- trading fees charged on a cryptocurrency exchange
- APIs supported by a cryptocurrency exchange

---

<sup>23</sup><https://www.techopedia.com/definition/5732/push-technology>



### Coinbase Pro

Coinbase Pro<sup>24</sup> formerly GDAX is a San Francisco based cryptocurrency exchange launched in 2012. It is a product of Coinbase and is meant to be a cryptocurrency exchange for advanced traders. It has more than 20 million users. According to CoinMarketCap, it is the 19th among the cryptocurrency exchanges considering the volume of the last 30 days [1]. According to the Blockchain Transparency Institute it is the 5th in a similar ranking considering 24-hour volume [69].

1. Trading pairs: It operates more than 30 trading pairs which include BTC, LTC, ETH, ETC, BCH, ZRX and others [70].
2. Trading fees: The fees depend on a 30-day trading volume. We will take the lowest – worst – one (\$0m–10m) into consideration. That is 0.3% Taker fee and 0% Maker fee for all the trading pairs [71].
3. API capabilities: There are public and private endpoints. The private ones are intended for order and account management. The public ones are intended for public market data. The latter is the one which would be used. There are three possible APIs provided [72].
  - REST (with HTTP used) – The limit is 3 requests/sec. which is 180 requests/min per IP considering the public endpoints. Or there is up to 6 requests/sec. which is 360 requests/min. per IP in bursts. The typical options as Product Order Book or Product Ticker are available.
  - WebSocket – It is publicly available but there is maximum connections limit set as 1 connection per 4 sec. per IP. One needs to read the message stream and use only the messages relevant to him.
  - FIX (Financial Information eXchange) – This API is used to enter orders, receive fills and submit cancel requests so there is no applicability of that for the purpose of this thesis.

### Coinmate

Coinmate<sup>25</sup> is a London based cryptocurrency exchange with a technology background created by a Czech company – Profinit<sup>26</sup> and launched in 2014. It is one of the smaller (below the first 100) cryptocurrency exchanges worldwide according to CoinMarketCap [1]. According to the Blockchain Transparency Institute it is the 27th in a similar ranking considering 24-hour volume [69].

---

<sup>24</sup><https://pro.coinbase.com/>

<sup>25</sup><https://coinmate.io/home>

<sup>26</sup><https://profinit.eu/klienti/>

## 2. SURVEY

---

1. Trading pairs: There are 11 trading pairs containing BTC, LTC, ETH, BCH cryptocurrencies. Every one of them is in a trading pair with EUR, CZK and BTC (naturally except BTC/BTC) [73].
2. Trading fees: Only the basic fees (30-day trading volume is less than 10,000 EUR) are taken into consideration. There are two groups of trading pairs in the sense of the fees [74].
  - The first one contains trading pairs which include LTC and, moreover, BTC/EUR, BTC/CZK. The Taker fee is 0.25% and the Maker fee is 0.12%.
  - The second one contains trading pairs which include ETH and BCH. The Taker fee is 0.15% and the Maker fee is 0.05%.
3. API capabilities: There are three suggested approaches how to communicate with Coinmate [75].
  - REST (with HTTP used) – There is a limit of 100 requests/min. or the IP address is banned otherwise. One can easily get the Order Book, Ticker, etc.
  - WebSocket – Coinmate uses Pusher<sup>27</sup> which is a hosted mainly WebSocket API [76].
  - XChange Java library<sup>28</sup> – It is a Java library that streamlines API of more than 50 cryptocurrency exchanges and creates a consistent interface.

### **Bittrex**

Bittrex<sup>29</sup> is a Seattle based cryptocurrency exchange launched in 2014. According to CoinMarketCap, it is the 40th among the cryptocurrency exchanges considering the volume of the last 30 days [1]. According to the Blockchain Transparency Institute it is the 9th in a similar ranking considering 24-hour volume [69].

1. Trading pairs: It operates more than 300 trading pairs which includes the major (in terms of market capitalization) cryptocurrencies as BTC, LTC, XMR, ETH, BCH, XLM, etc. as well as the minor ones as XZC (Zcoin) or EXP (Expanse). It is divided into four groups of markets – USD market, Bitcoin market, Ethereum market and USDT market [77].
2. Trading fees: There is one global fee applied in all the trades – 0.25% [78].

---

<sup>27</sup><https://pusher.com/>

<sup>28</sup><https://github.com/knownm/XChange>

<sup>29</sup><https://international.bittrex.com/>

3. API capabilities: There are three distinct groups of APIs – Public, Market, Account. The only one needed for the purposes of this thesis is the Public group which does not require any API key [79, 80].
  - REST (with HTTP used) – There is a limit of 1 request/sec. because of Bittrex itself. It updates the data at regular intervals of 1 sec. The REST API provides common functionality such as Order Book, Ticker, Market Summary, etc.
  - WebSocket – The Websockets API is also possible to use. One needs a SignalR<sup>30</sup> client. SignalR is a library mainly for ASP.NET developers.

### Bitstamp

Bitstamp<sup>31</sup> is a Luxembourg based cryptocurrency exchange launched in 2011. It started as an alternative to the then dominant Mt. Gox<sup>32</sup> – by contrast primarily focusing on the European market [81]. According to CoinMarketCap, it is the 28th among the cryptocurrency exchanges considering the volume of the last 30 days [1]. According to the Blockchain Transparency Institute it is the 6th in a similar ranking considering 24-hour volume [69].

1. Trading pairs: It operates exactly 15 trading pairs. There are XRP, LTC, ETH, BCH where each of them is in a trading pair with USD, EUR and BTC. Then there are three more trading pairs – BTC/USD, BTC/EUR and even EUR/USD [82].
2. Trading fees: There are various options according to 30-day USD volume. We will take into consideration only the one which is less than 20,000 USD. That means it is the worst one and is 0.25% for all 15 trading pairs. There is also a minimum order size 5 EUR, 5 USD and 0.001 BTC for Euro, USD and Bitcoin denominated trading pairs respectively [83].
3. API capabilities: There are three accessible APIs – REST, WebSocket, FIX [84].
  - REST – There is a limit of 600 requests/10 min. which is 1 request/sec. on average. It provides common functionality such as Ticker and Order Book as well as Hourly Ticker, Trading Pairs Info, etc.
  - WebSocket – Bitstamp uses Pusher for the WebSocket streaming. It handles Live Orders, Live Ticker, etc.

---

<sup>30</sup><https://www.asp.net/signalr>

<sup>31</sup><https://www.bitstamp.net/>

<sup>32</sup><https://www.mtgox.com/>

- **FIX** – It is based on FIX 4.4 standard and it works as a gateway. It performs routing and translation of FIX messages from clients to appropriate calls to REST or WebSocket APIs and vice versa. It enables to get data of Live Ticker, Order Book, etc.

### **Bitfinex**

Bitfinex<sup>33</sup> has its headquarters in Hong Kong and was founded in 2012. Although it has experienced an extensive security breach [85], according to CoinMarketCap it is the 6th among the cryptocurrency exchanges considering the volume of the last 30 days [1]. According to the Blockchain Transparency Institute it is the 2nd in a similar ranking considering 24-hour volume [69].

1. **Trading pairs:** There are more than 300 trading pairs operating. Almost every (crypto)currency is in a trading pair with BTC, ETH and USD. In addition, one can also trade JPY (Japanese Yen), however, only with eight cryptocurrencies altogether [86].
2. **Trading fees:** The fees depend on a USD equivalent of the order execution during the last 30 days. We will take into consideration only the one which is less than 500,000 USD. Bitfinex also divides the fees into two types depending on a trade type – Taker fee and Maker fee. The first one is 0.2% and the latter is 0.1%. One can also place a hidden order always with the Taker fee. If (another) limit order matches this hidden order then the trader pays the Maker fee [87].
3. **API capabilities:** There are two accessible APIs – REST and WebSocket [88].
  - **REST** – There is a limit between 10 and 90 requests/min. which is between 1 request/6 sec. and 3 requests/2 sec. on average. Each endpoint has its own limit. For example, limit of the Ticker is 20 requests/min. If a user exceeds the limit his or her IP address is blocked for 10 to 60 seconds. There are two types of endpoints again – Public and Authenticated. Only the first type would be used. It provides common functionality such as Ticker and Order Book as well as Trades and Stats.
  - **WebSocket** – Bitfinex also supports WebSocket protocol where all the sent and received messages via the WebSocket channel are encoded in JSON format. It divides into Public channels and Authenticated channels. Public channels provide Order Books, Raw Order Books, Trades and Ticker.

---

<sup>33</sup><https://www.bitfinex.com/>

### **Binance**

Binance<sup>34</sup> is a Malta based [89, 90] cryptocurrency exchange founded in 2017. It is capable of sustaining 1,400,000 orders/sec. [91]. According to CoinMarketCap, it is the 1st among the cryptocurrency exchanges considering the volume of the last 30 days [1]. According to the Blockchain Transparency Institute it is also the 1st in a similar ranking considering 24-hour volume [69].

1. Trading pairs: It is possible to trade more than 100 cryptocurrencies through more than 300 trading pairs [92]. Even though it does not support any fiat currency [93], it intends to support it in the future [94].
2. Trading fees: Binance runs Maker-Taker model as well. There are multiple levels of fees depending on 30-day BTC volume and BNB (Binance coin) balance. One can use BNB for transaction fees and in that case he or she has a discount of various percentage rates depending on time. We will take into consideration the General level which is set as less than 100 BTC or greater than or equal to 0 BNB. Then the fees equal 0.1% both for Maker and Taker fees [95].
3. API capabilities: There are two accessible APIs – REST and WebSocket [96].
  - REST – The limits are weighted in favour of a specific request type. Maximum of the request weight is 1200/min. In addition, there is a maximum limit of raw requests which is 5000/5 min. If a user exceeds the limits his or her IP address is blocked for 2 minutes up to 3 days. The desirable endpoint would be Symbol order book ticker which has a weight of 2 in order to get all trading pairs on the cryptocurrency exchange.
  - WebSocket – Binance offers Trade Streams and Ticker Streams. The latter is pushed every second to a client. All the stream connections are valid for 24 hours maximum. On top of that, the connection must be maintained by ping pong messages (heartbeat) every 10 minutes.

### **Kraken**

Kraken<sup>35</sup> is a cryptocurrency exchange founded in 2011 and based in San Francisco. It is the largest Bitcoin exchange in view of euro volume and liquidity [97, 98]. According to CoinMarketCap, it is the 18th among the cryptocurrency exchanges considering the volume of the last 30 days [1]. According

---

<sup>34</sup><https://www.binance.com/en>

<sup>35</sup><https://www.kraken.com/>

## 2. SURVEY

---

to the Blockchain Transparency Institute it is the 4th in a similar ranking considering 24-hour volume [69].

1. Trading pairs: Kraken operates 20 cryptocurrencies and 5 fiat currencies [99]. In total there are 73 trading pairs [100]. Regarding fiat currencies, one can trade Canadian dollars, US dollars, British pounds, Euros and Japanese yen.
2. Trading fees: The fees comply with Maker-Taker model. There is no common trading fee for all the trading pairs. Each trading pair has its own trading fees. We will take into consideration only the first tier of each trading pair. The tiers are determined according to the last 30-day “Fee Volume Currency” volume. All the realized trades are always converted to the equivalent value of this Fee Volume Currency. Fee Volume Currency is USD for all the trading pairs right now. In the first tier there is 0.16% Maker fee and 0.26% Taker fee for each cryptocurrency except two cases – dark pools<sup>36</sup> and USDT/USD trading pair. Dark pools do not follow Maker-Taker model and has 0.36% trading fees. USDT/USD trading pair has both 0.2% Maker and Taker fee [101].
3. API capabilities: There are two accessible APIs – REST and WebSocket [102]. The latter was added in 2019 [103].
  - REST – The limit for Public calls is 1 request/sec. One does not need an API key in case of using the Public calls. API key is needed only for Private calls where there is a call counter. Call counter may exceed a maximum (which varies depending on user’s verification level) and then the user’s API access is suspended for 15 minutes. Endpoint calls increase the call counter by zero, one or two depending on the endpoint type. The call counter is also decreased by one automatically every 1 or 2 or 3 seconds again depending on user’s verification level [104].
  - WebSocket – Kraken WebSocket API offers real-time market data updates encoded in JSON format. All the supported pairs including the aliases are mentioned in [105]. The API supports the common market information such as Ticker information or Order book levels, etc.

### Huobi

Huobi<sup>37</sup> is a cryptocurrency exchange founded in 2013. It is based in Singapore and founded in China. In 2018, Huobi had to create the Communist Party committee. This was the first case when a Chinese blockchain company had

---

<sup>36</sup><https://www.investopedia.com/terms/d/dark-pool.asp>

<sup>37</sup><https://www.hbg.com/en-us/>

to do this [106]. According to the Blockchain Transparency Institute, the real 24-hour trading volume of BTC/USDT trading pair is around 26% of the reported one [107]. One of the reasons might be wash trading<sup>38</sup> as stated by BTI. According to CoinMarketCap, it is the 3rd among the cryptocurrency exchanges considering the volume of the last 30 days [1].

1. Trading pairs: Huobi operates more than 400 trading pairs. There are five quote cryptocurrencies<sup>39</sup> – USDT, HUSD (Huobi Stable Coin), BTC, ETH and HT (Huobi Token). It does not support fiat currencies [108].
2. Trading fees: Huobi comply with Maker-Taker model. Each currency has its own Maker fee and Taker fee. Nevertheless, all the trading fees – both Maker and Taker – are equal to 0.2% [109].
3. API capabilities: Huobi’s API supports all the trading pairs. There are two typical APIs provided – REST API and WebSocket API [110].
  - REST – There are two types of requests. Those which need a signature and those who do not. Market Data and Symbols&Currency data requests do not need a signature and are public. The request limits are specified as 100 requests/10 sec. which is 10 requests/sec. for each API key on average. Since there is no need to provide an API key for the public requests and there are no other request limits provided, the public request limits are unknown. In this case we would follow the already mentioned provided limits.
  - WebSocket – There are two types of WebSocket data. Market Data and Accounts and Orders. The first one would be used and there is no authentication (API key) required for it. One can subscribe to Market Data or request it at once.

### OKEx

OKEx<sup>40</sup> is a Belize based cryptocurrency exchange founded in 2014 with offices in Hong Kong and Malta [111, 112]. According to the Blockchain Transparency Institute, its official volume is not true and its top trading pairs are engaging in wash trading [107]. The BTI claims that the real volume of BTC/USDT trading pair is around 11% of the reported one. According to CoinMarketCap, it is the 2nd among the cryptocurrency exchanges considering the volume of the last 30 days [1].

1. Trading pairs: OKEx operates more than 300 trading pairs. There are four quote cryptocurrencies – BTC, USDT, ETH, OKB (OKEx Utility

---

<sup>38</sup><https://www.investopedia.com/terms/w/washtrading.asp>

<sup>39</sup><https://www.investopedia.com/terms/q/quotecurrency.asp>

<sup>40</sup><https://www.okex.com/>

## 2. SURVEY

---

Token) [113]. It does not support fiat currencies except the OTC Trading [114].

2. Trading fees: OKEx runs 30-day trading volume-based Maker-Taker model. The volume is counted in BTC. We would take Tier level 1 into consideration. That means the cumulated volume is less than 100 BTC, Maker fee is 0.1% and Taker fee is 0.15% [115].
3. API capabilities: There are two types of endpoints – Public and Private. Since the Public one is available for acquiring information and market data, it is sufficient. There is no authentication needed. OKEx supports both REST API and WebSocket API [116, 117].
  - REST – The current version of the API is v3. The limit for the REST API requests varies depending on a specific endpoint. If there is no specific endpoint limitation the general REST API limit is applied. That is 6 requests/sec. One of the prospective endpoints which could be used is Get All Token Pairs Information which is limited to 20 requests/2 sec. which is 10 requests/sec. on average.
  - WebSocket – This API is recommended for accessing market related information and trading depth. All the incoming messages – from the client point of view – has to be decompressed. There is a connection limit of 1 connection/min. and subscription limit of 240 subscriptions/hour. In addition, there is a limit of 50 outgoing message commands/sec. One needs to maintain a connection by ping pong messages (heartbeat) every 30 seconds.

### DigiFinex

DigiFinex<sup>41</sup> is a cryptocurrency exchange founded in 2017 in Seychelles with offices in Singapore and China [118]. According to the Blockchain Transparency Institute, DigiFinex most likely participates in wash trading. E.g. the BTC/USDT trading pair evinces 99% of the 24-hour traded volume is not real [107]. According to CoinMarketCap, it is the 5th among the cryptocurrency exchanges considering the volume of the last 30 days [1].

1. Trading pairs: DigiFinex operates more than 100 trading pairs. There are five quote cryptocurrencies – USDT, BTC, ETH, DFT, TUSD. It supports only stablecoins<sup>42</sup> instead of fiat currencies.
2. Trading fees: In November 2018 DigiFinex introduced Maker-Taker model. Next to this model they also implemented level-based fees. There are 7 levels – from General through VIP1 to VIP6. The levels

---

<sup>41</sup><https://www.digifinex.com/en-ww/>

<sup>42</sup><https://www.investopedia.com/terms/s/stablecoin.asp>



depend on 30-day trading volume and DFT (DigiFinex Token) holdings. General level would be taken into consideration which means less than 100,000 USDT and less than 100 DFT. Then the Maker fee is 0% regardless of DFT condition, Taker fee is 0.2% and Taker fee in compliance with DFT condition is 0.15% [119].

3. API capabilities: There is only REST API available [120].
  - REST – In order to execute any request API key must be provided. It means there is no public endpoint so one needs to register even for Market Information such as Ticker or Market Depth. The limits are 60 POST requests/min. which is 1 request/sec. on average and 180 GET requests/min. which is 3 requests/sec. on average. If any limit is exceeded the 5 min. ban will be applied.

## 2.4 Existing Solutions

Apart from the survey of cryptocurrency market itself and the individual cryptocurrency exchanges, the already existing solutions are surveyed as well. In this subchapter several applications are discussed. They consist of basic (cross-exchange) arbitrage applications and triangular arbitrage applications.

### Basic Cross-Exchange Arbitrage Applications

There are many basic arbitrage applications available on the internet. That includes both web and desktop applications:

- Web applications – One of the well-arranged web applications is Crypto Arbitrage Opportunity Expert [7] which shows the present arbitrage opportunities (in %) among many cryptocurrency exchanges like Bitfinex, Bitstamp, etc. It allows to choose various cryptocurrency exchanges and trading pairs to watch. The snapshot of the application is in the figure 2.1. The menu of the website refers to other paid services such as Arbitrage Paths and Arbitrage Spread Signals. Then there are similar web applications which are less well-arranged such as Arbitrage Opportunities for Cryptocurrencies by CryptoCoinCharts<sup>43</sup> or Cryptocurrency Arbitrage opportunities by CoinCheckup<sup>44</sup>. Finally, there are also robust applications such as HaasOnline<sup>45</sup> which offer various pricing plans depending on the available features.

---

<sup>43</sup><https://coin.market/arbitrage.php>

<sup>44</sup><https://arbitrage.coincheckup.com/>

<sup>45</sup><https://www.haasonline.com>

- Desktop applications – There are projects on GitHub which can be downloaded and installed such as Blackbird [121] published under the MIT license. It does automatic arbitrage on Bitfinex, Kraken, Bitstamp, etc. and it is programmed in C++. Another project is CryptoArbitrage-Trader [122] programmed in Python.

### Triangular Arbitrage Applications

Unlike the basic arbitrage applications, there are not many triangular arbitrage applications available on the internet. Typically, there are two types of them. Firstly, there are poorly-documented projects without any GUI which have not been maintained for months, and secondly, there are paid applications with various pricing plans.

An example of the second type is the already mentioned HaasOnline software [123]. It allows a user to choose the trading pairs which are to be involved in the triangular arbitrage, the trade amount in a specific (crypto)currency and the profit target (in %). Then the bot automatically tries to execute the triangular arbitrage when an opportunity exists.

An example of the first type is the TriangularArbitrage [124] project on GitHub written in Python. The snapshot of the application is in the figure 2.2. It shows the triangular arbitrage opportunities on Binance cryptocurrency exchange only, it is published under ISC license and it is neither documented nor maintained. A user has to set the starting (crypto)currency and select the (crypto)currencies present in the triangle. This fact forces the user to be rather specific. Another similar project is Crypto Trading Bots in Python<sup>46</sup>.

Taking into consideration the already existing applications, there has not been found any open-source web application which displays the real-time triangular arbitrage opportunities on several cryptocurrency exchanges and which also visualises historical data consisting triangular arbitrage opportunities.

---

<sup>46</sup><https://github.com/Roibal/Cryptocurrency-Trading-Bots-Python-Beginner-Advance>

2.4. Existing Solutions

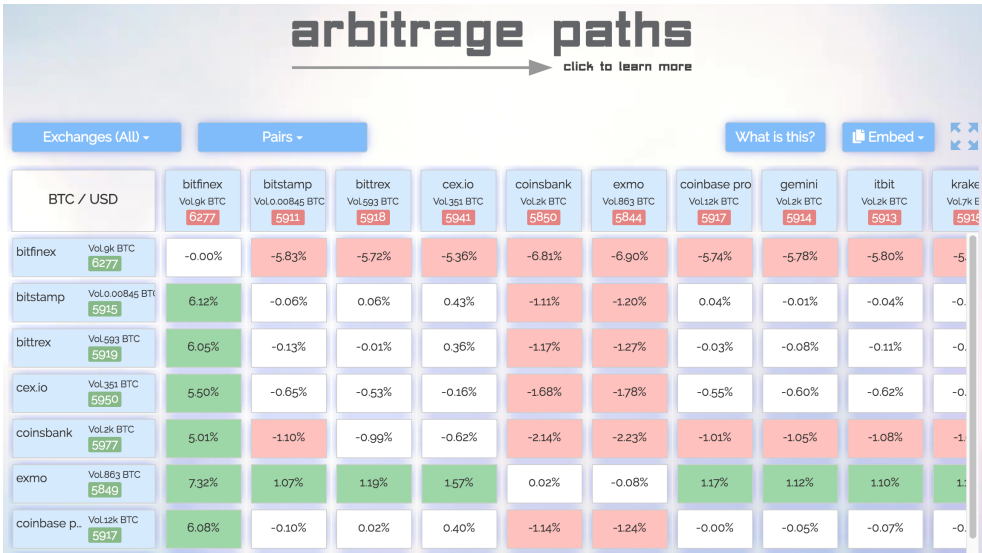


Figure 2.1: Crypto Arbitrage Opportunity Expert website snapshot [7]

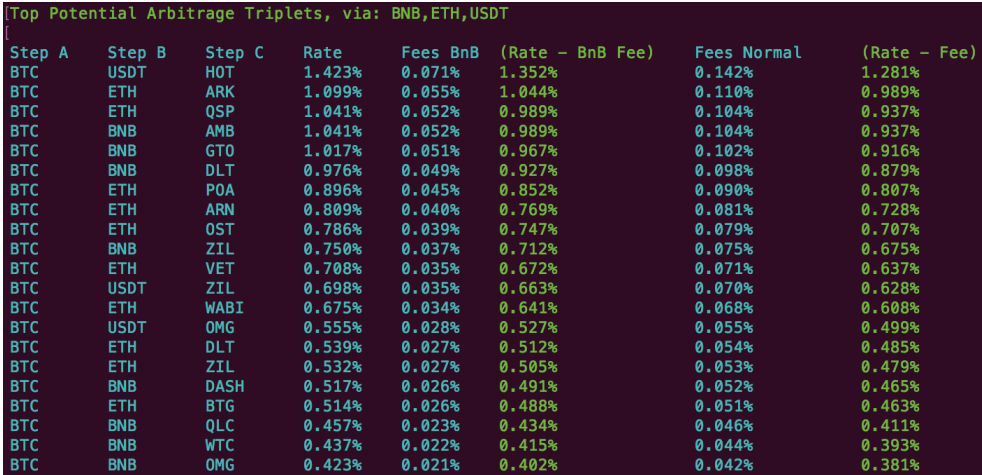


Figure 2.2: TriangularArbitrage application snapshot [124]



---

# Analysis

In this chapter the problem is formulated (not only) from a mathematical point of view. Then the functional and non-functional requirements are determined. They establish the core properties and characteristics of the work. They also set the limits of this work and describe it in more detail. There are tables in both subchapters dealing with requirements in order to outline each requirement briefly. After that the requirements are described more thoroughly one by one. This chapter also mentions use cases and scenarios of the application – what will the user do with it.

## 3.1 Problem Definition

The triangular arbitrage as well as the basic arbitrage is dependent on trading pairs which are available at the specific (cryptocurrency) exchange. These trading pairs are comprised of two different (crypto)currencies usually in one of the following forms:

- $XXX/YYY$
- $XXX-YYY$
- $XXXYYY$

The first part ( $XXX$ ) is called base currency and the second part ( $YYY$ ) is called quote currency. The trading pair itself is a quotation of the relative value of the  $XXX$  unit against the  $YYY$  unit. For example, quotation  $BTC/EUR = 4870$  means that 1 BTC is equal to 4870 EUR, where 4870 is usually the price of the last trade. Here, BTC is the base currency and EUR is the quote currency [125]. The trading pairs on the analysed cryptocurrency exchanges are never quoted both as  $XXX/YYY$  and as  $YYY/XXX$ .

Next to the trading pairs and the corresponding quotations each trading pair also has a bid price and an ask price. Bid price represents the best (highest) price for which one is willing to buy a unit of the selected (crypto)currency.

### 3. ANALYSIS

Ask price represents the best (lowest) price for which one is willing to sell a unit of the selected (crypto)currency [125]. The difference between these two prices is called bid-ask spread. Considering the example in the previous paragraph, the bid price would be e.g. 4869.9 and the ask price would be e.g. 4870.1.

Triangular arbitrage exploits a pricing discrepancy – contained in the trading pairs – among three different currencies, i.e. using three different trading pairs. Graphical example of the triangular arbitrage in the figure 3.1 shows how it works. The difference between the example in the figure and the situation we explore is that there are no banks. All the trading pairs are quoted on selected cryptocurrency exchange. In addition, there are also cryptocurrencies considered in the triangular arbitrage instead of fiat currencies only. Moreover, there are fees present in the situation we explore.

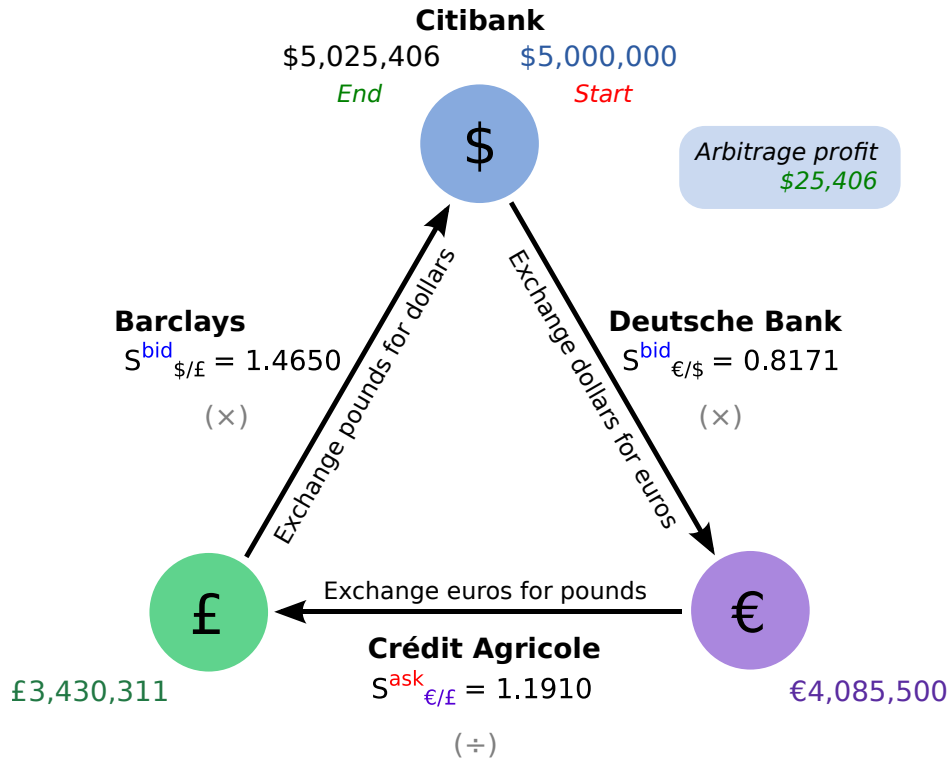


Figure 3.1: Triangular arbitrage scenario using sample bid and ask prices quoted by international banks [11]

From a trader's (arbitrageur's) perspective the triangular arbitrage involves three trades. In the beginning the arbitrageur has some amount of currency  $X_1$ . He or she exchanges the currency  $X_1$  for currency  $X_2$ . Then he or she exchanges  $X_2$  for currency  $X_3$  and finally  $X_3$  is exchanged again for the currency  $X_1$  so the triangle is closed. The triangular arbitrage is successful if

the final amount of the currency  $X_1$  is higher than it was in the beginning. That can happen only if there exist market imperfections. The individual exchange orders within the triangular arbitrage may be acts of either selling or buying. The deciding factor of the choice of selling or buying is the order of currencies in the trading pairs. Let us consider one has the currency  $X_1$ , wants to exchange it for the currency  $X_2$  and the trading pair is quoted as  $X_1/X_2$ . Since the base currency is  $X_1$ , then the operations (buy/sell) are executed in terms of this currency (buy/sell some amount of  $X_1$ ). Thus he or she needs to sell  $X_1$  and get some  $X_2$  for it. It means he or she needs to check the bid price, execute the sell order and get some  $X_2$  in return. In other words, the arbitrageur multiplies his or her amount of  $X_1$  by the bid price. On the other hand, if the trading pair is quoted as  $X_2/X_1$  then the arbitrageur needs to buy  $X_2$ , so he or she checks the ask price. In other words, the arbitrageur divides his or her amount of  $X_1$  by the ask price. We can express this paragraph as follows.

Let us define

$$\gamma_{X_1, X_2, X_3}(t) = \prod_{i=1}^3 r_i(t) \quad (3.1)$$

as the exchange rate product at time  $t$  and  $r_i(t)$  as one of the following:

- $i$ -th exchange rate from currency  $X_i$  to  $X_{i+1}$  at time  $t$  where the exchange rate is the *bid price*, if the (crypto)currency we possess before the order is placed is the base currency
- $i$ -th exchange rate from currency  $X_i$  to  $X_{i+1}$  at time  $t$  where the exchange rate is  $\frac{1}{\text{ask price}}$ , if the (crypto)currency we possess before the order is placed is the quote currency

considering  $X_4 = X_1$ . If the result of the exchange rate product is:

- $\gamma_{X_1, X_2, X_3}(t) > 1$ , there is a triangular arbitrage opportunity
- $\gamma_{X_1, X_2, X_3}(t) < 1$ , there is no triangular arbitrage opportunity and the trade would be loss-making
- $\gamma_{X_1, X_2, X_3}(t) = 1$ , there is no triangular arbitrage opportunity and the trade would not be loss-making

In addition, the majority of the popular cryptocurrency exchanges charge taker fees (see 2.3). Either there is one global taker fee for all the trading pairs within the cryptocurrency exchange or each trading pair has its own taker fee. This fact changes the definition of the exchange rate product as follows:

$$\gamma'_{X_1, X_2, X_3}(t) = \prod_{i=1}^3 (r_i(t) \cdot (1 - f_i)) \quad (3.2)$$

where  $0 \leq f_i < 1$  is a taker fee which corresponds to the  $i$ -th exchange rate from currency  $X_i$  to  $X_{i+1}$ .

## 3.2 Functional Requirements

All the functional requirements are defined based on the thesis assignment and the already explored functionality of the existing data collecting applications. First, we provide a table 3.1 of the functional requirements together at one place and then we describe them in full detail one by one.

#	Name of the functional requirement
F1	Collect data from selected cryptocurrency exchanges
F2	Process collected data in order to display potential profit
F3	Store the processed data in a database
F4	Display the processed data using GUI
F5	Display collected historical data
F6	Sort the processed data
F7	Highlight the profitable paths
F8	Choose data sources

Table 3.1: Functional requirements

### **F1 – Collect data from selected cryptocurrency exchanges**

The application will collect (download) selected cryptocurrency data – the trading pairs – from the selected cryptocurrency exchanges. This action will be performed on a regular basis. If some of the selected cryptocurrency exchanges do not respond it will collect data from the other ones.

### **F2 – Process collected data in order to display potential profit**

The collected data will be processed in a way within the meaning of the triangular arbitrage [9, 10, 11]. That means a situation where there is a succession of three new buy/sell orders on different cryptocurrency pairs on



a cryptocurrency exchanges – e.g. a successive buy of ADA for BTC, ETH for ADA and BTC for ETH (i.e.  $BTC \rightarrow ADA \rightarrow ETH \rightarrow BTC$ ). Both at the beginning and at the end there is the same (crypto)currency (e.g. BTC). There are three different possible outcomes. Either the final amount (balance) can be higher (positive) or lower (negative) or the same as the initial one. In other words, there is some percentage of earnings. The application will not buy or sell any of the cryptocurrencies. It will display potential percentage of earnings of such an operation instead. In addition, it will store the data in a database. Hence, the collected data will be processed in order to satisfy two separate requirements. Firstly, in a way which is suitable for displaying the required data. Secondly, in a way which is suitable for storing the data.

### **F3 – Store the processed data in a database**

The application will store selected data in a database in order to display them for the purpose of potential comparison or for an analysis of cryptocurrency exchange inefficiencies development with time. The data will be stored in a format which is suitable for the subsequent displaying. Owing to correctness of the historical data the already gathered data will be used.

### **F4 – Display the processed data using GUI**

The processed data (from F2) will be displayed using Graphical User Interface. The application will display various triangles (sequences of cryptocurrencies) and potential instant profit/loss from the corresponding transaction orders including mentioning the cryptocurrency exchange fees. One can get remotely to the processed data, thus to the GUI over the internet accessing a web page. That is convenient for any user who wants to examine the data without installing any additional software (except the web browser). It also improves readability and intelligibility of the processed data.

### **F5 – Display collected historical data**

Not only real-time data will be displayed. Along with them will also be possible to search for and display historical data. A user will be able to choose a point in time and thereafter the corresponding data will be displayed. The historical data will be presented in the context of percentage of earnings at that point in time.

### **F6 – Sort the processed data**

The collected and processed data will be sorted primarily in descending order in terms of the potential percentage of earnings. The reason is that the profit is the core of the problem we address. So the first thing one would like to know is if the triangular arbitrage is profitable at the selected point in time.

Ascending order will also be possible to choose because of knowing the worst sequences of cryptocurrencies on the list might be valuable to a user as well.

#### **F7 – Highlight the profitable paths**

The most interesting part of the exploration is whether there will be profitable triangles (paths) on the selected cryptocurrency exchanges with the selected trading pairs. That is the reason why the highlighting of the profitable paths is interesting and important for a user. Hence, the application will highlight these paths.

#### **F8 – Choose data sources**

The application will allow a user to choose which particular cryptocurrency exchange will display the real-time data. The reason is the user might trade just on few of the offered cryptocurrency exchanges so he or she might be interested only in these particular cryptocurrency exchanges. Besides the real-time data, the user will be able to choose all the cryptocurrency exchanges he or she wants to view in the graphical representation of the historical data. In addition, the cryptocurrency market has been changing since it came into existence. As a result of it, various cryptocurrency exchanges have changed in the volume of trade or they even discontinued [2] by the proprietors. Under these circumstances it is possible that the suggested cryptocurrency exchanges could act similarly in the future and so it is reasonable to let the user choose.

### **3.3 Non-Functional Requirements**

All the non-functional requirements delimit major capabilities of the program from a distance. They set criteria that can be used to evaluate various operations of the system. In addition, they are chosen in a way which minimizes contradiction among each one of them in spite of the fact that non-functional requirements often go against each other. They are also defined based on the thesis assignment and the already explored functionality of the existing data collecting applications. They generalize and support the functional requirements. First, we provide a table 3.2 of the non-functional requirements together at one place and then we describe them in full detail one by one.

#	Name of the non-functional requirement
N1	Real-time communication
N2	Modifiability of the sources
N3	Fault-tolerant system
N4	Data reliability
N5	Open-source code
N6	Platform independence
N7	Source code readability
N8	Usability
N9	Extensibility
N10	Accessibility

Table 3.2: Non-functional requirements

### **N1 – Real-time communication**

In an application like this the response time is an important element in order to fulfil one of the core requirements. One of the key attributes of cryptocurrency trading is the speed – how fast one can buy or sell assets. In the spirit of this attribute it is important to keep up with the traders and provide real-time information about potential profit of the triangular arbitrage. Real-time does not mean a precise period of time. The time period is relative to a specific cryptocurrency exchange – e.g. how often individual exchanges refresh or provide the trade-related data.

### **N2 – Modifiability of the sources**

This non-functional requirement is moderately connected with the functional requirement F8. The cryptocurrency exchanges change often in time in terms of the traded volume or they even close and stop working due to various reasons. Considering these potential issues, the modifiability of the sources is desired. An administrator of the application will be able to easily change (add, delete or adjust) the sources of data – i.e. cryptocurrency exchanges and their APIs – used for collecting and subsequently processing the finally displayed data.

### **N3 – Fault-tolerant system**

The application will show signs of a fault-tolerant system in terms of a potential number of data sources. Firstly, the functionality will enable using multiple sources of data – i.e. cryptocurrency exchanges and their APIs – which will prevent the application from collapsing due to a reason there is only one source which stops with data provisioning. Secondly, when one of

the viable sources stops providing data then the application will not stop working and will continue with the rest of the viable sources.

#### **N4 – Data reliability**

The application will comply with the data reliability owing to the fact that we will gather the data on our own. No intermediaries will be used. The reason is that there have been issues of reliability of the cryptocurrency exchange intermediaries like CoinMarketCap in the past [126, 127, 128]. One of the appropriate methods to overcome the problem is to choose credible cryptocurrency exchanges and connect to them directly or using a library. This shifts the potential reliability problem to the particular sources (cryptocurrency exchanges).

#### **N5 – Open-source code**

Whole application will be managed as an open-source project which has several advantages. Some of them are:

- Transparency – everyone who wants to use the software can read the source code and assure himself of not-malicious intentions of the project. In addition, progress of the project will be trackable (e.g. via timestamps) and everyone will be able to see it.
- Increase in use and collaboration – since the project will be open-source, it will be easy to share it and more persons will be able to learn about it or to join it.
- Reliability – software will have greater potential to be reliable because more people who are interested will be able to contribute easily and test the application in the future.
- Independence – even if we stop developing the software it will continue to exist and will be developed by its users.

#### **N6 – Platform independence**

The program will work as a web application which means one will be able to run it on a web browser. The platform independence will be ensured by the web browser which can be installed on all the major popular operating systems (Windows, OS X/macOS, other Unix-based systems). The advantage of using the web browser is not only the platform independence but also the accessibility which is described in the non-functional requirement N10 below.

#### **N7 – Source code readability**

The code of the application will be easy to follow and will support the best practices of the code readability (e.g. commenting on complicated parts, meaningful variable naming, skilful use of programming language facilities, short functions which meet the Separation of Concerns principle<sup>47</sup>

#### **N8 – Usability**

User testing which will be held at the end of the application creation process will ensure the usability of the application. Several human testers will verify that the application is easy to understand and one can move with ease within it. They will provide us with feedback on what is not fully clear.

#### **N9 – Extensibility**

It will be easy to add or modify a subset of already included sources – cryptocurrency exchanges – in the application. This requirement is particularly important in case of a cryptocurrency exchange’s API maintenance or change. In addition, it is also important to fulfil this requirement in case there is a new promising cryptocurrency exchange coming into existence. As a matter of course, the code will take a future growth into consideration.

#### **N10 – Accessibility**

The application functionality will be accessible (in terms of how to get to the application’s outputs) over the internet and WWW<sup>48</sup>. That enables working with the data and the whole application from any place connected to the internet and so increases its accessibility.

---

<sup>47</sup><https://books.google.cz/books?id=pFHYkOKWAEgC>, p. 85

<sup>48</sup><https://www.w3.org/TR/webarch/>

### 3.4 Use Cases and Scenarios

This subchapter describes various options how the application may be used by a user. A user is supposed to be a common person who wants to watch triangular arbitrage opportunities in context of various cryptocurrency exchanges in real-time. Apart from the real-time data he or she may also study the historical data, i.e. the opportunities over time. He or she might want to compare those cryptocurrency exchanges in terms of the opportunities occurring in them. All the use cases are shown in the use case diagram 3.2 and summarized in the table 3.3. After that they are mapped onto the functional requirements.

#	Name of the use case
UC1	Choose among cryptocurrency exchanges
UC2	Sort real-time triangular arbitrage opportunities
UC3	View triangular arbitrage opportunities
UC4	View historical data on specific day
UC5	Compare historical triangular arbitrage opportunities among cryptocurrency exchanges
UC6	View historical record on specific hour among the cryptocurrency exchanges
UC7	View the highest potential profit during a specific day

Table 3.3: Use cases

The use case diagram 3.2 is one way how to clearly outline all the user's interactions with the application where he or she is involved. The user is pictured as the figure. The use cases are represented by the ovals. Some use cases extend other use cases. That is the extended use cases may be concretized in one or more ways. More detailed use case descriptions are introduced below.

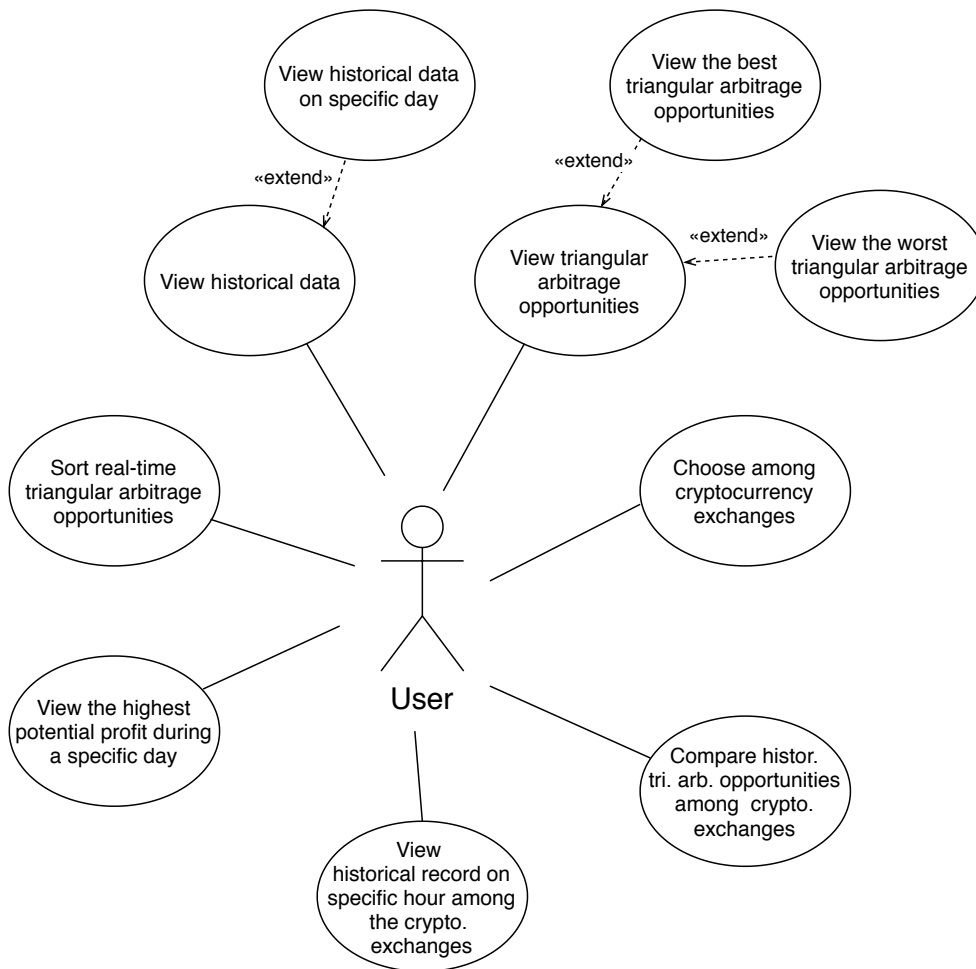


Figure 3.2: Use case diagram containing all the user's use cases

### UC1 – Choose among cryptocurrency exchanges

A user will be able to choose among various cryptocurrency exchanges. There are more reasons to do that – one of the exchanges might be out of order, the user might want to compare real-time data among the exchanges, specific exchange adds a new cryptocurrency, etc. Simplified scenario for this use case:

1. User navigates to the page with real-time data.
2. Clicks on a button labeled with the specific cryptocurrency exchange name.
3. Waits until the data are fetched.

#### **UC2 – Sort real-time triangular arbitrage opportunities**

A user will have an option of changing the order of showing data – ascending and descending order. He or she might want to see the best and the worst combinations (triangles) of (crypto)currencies for executing triangular arbitrage. That is the main reason for enabling it. Only a subset of data will be shown due to the high number of the records. Simplified scenario for this use case:

1. User navigates to the page with real-time data.
2. Selects the desired order.
3. Waits until the data are fetched and sorted properly.

#### **UC3 – View triangular arbitrage opportunities**

This is one of the major and probably most frequent use cases of the application. A user will be able to view the triangular arbitrage opportunities in real-time. A typical situation arises when a new coin is introduced on a cryptocurrency exchange – its value is usually volatile in the beginning so triangular arbitrage opportunities might appear. Another typical situation arises when a price of one or more cryptocurrencies rapidly changes. Simplified scenario for this use case:

1. User navigates to the page with real-time data.
2. Scrolls down or up the page to see desired records (opportunities).

#### **UC4 – View historical data on specific day**

Next to the real-time data a user may also need to view historical data for various reasons. One of them might be he or she does not have time to view the real-time data on time or wants to view a specific day when something happened on cryptocurrency market. Another reason might be the historical data are condensed so they provide big picture on the selected day. Simplified scenario for this use case:

1. User navigates to the page with historical data.
2. Selects the desired day and confirms the selection.
3. Views the data for selected day (2 possibilities).
  - Either there is an option of using graphical representation of historical data.
  - Or one can scroll the page in order to read all the historical data from that day.



### **UC5 – Compare historical triangular arbitrage opportunities among cryptocurrency exchanges**

Since there is more than one cryptocurrency exchange available for a user, he or she has an opportunity to compare historical triangular arbitrage opportunities among those exchanges at one place. A typical situation is again when price rapidly changes at some point in time – different exchanges then record different profits. In addition, one can view profit development over time on the selected day among the exchanges and compare them. Simplified scenario for this use case:

1. User navigates to the page with historical data.
2. Selects the desired day and confirms the selection.
3. Views the graphical representation of historical data which allows the user to compare cryptocurrency exchanges over time.

### **UC6 – View historical record on specific hour among the cryptocurrency exchanges**

A user might want to check a specific hour in a specific day in year – e.g. an event like in December 2017 when Bitcoin was getting to its peak <sup>49</sup>. The application is used in this way for viewing the opportunities among cryptocurrency exchanges during separate events which are somehow unique. Simplified scenario for this use case:

1. User navigates to the page with historical data.
2. Selects the desired day and confirms the selection.
3. Views the specific hour on that day (2 possibilities).
  - One way is to view the specific hour in the graphical representation of data.
  - Another way is to scroll down the page until he or she sees the specific record.

### **UC7 – View the highest potential profit during a specific day**

One might want to view the best potential profit of a day. One of the reasons might be to see at what hours the best potential profits across the cryptocurrency exchanges usually appear. According to those periods of time might one choose the most suitable cryptocurrency exchange and be ready to execute triangular arbitrage. Simplified scenario for this use case:

---

<sup>49</sup><https://www.coindesk.com/sell-news-bitcoin-price-tests-20k-ahead-cmes-futures-launch>

### 3. ANALYSIS

---

1. User navigates to the page with historical data.
2. Selects the desired day and confirms the selection.
3. Views the specific hour with highest profit on that day (2 possibilities).
  - One way is to notice the peak in the graphical representation of data and check the corresponding hour.
  - Another way is to scroll down the page and search for the record with the highest potential profit.

The following table 3.4 represents mapping of the already defined use cases onto the functional requirements. It documents which functionalities are used by specific user's interactions with the application. It also shows and verifies whether all the functional requirements have been utilised. Every functional requirement is mapped.

#	UC1	UC2	UC3	UC4	UC5	UC6	UC7
<b>F1</b>	•	•	•	•	•	•	•
<b>F2</b>		•	•	•	•	•	•
<b>F3</b>				•	•	•	•
<b>F4</b>	•	•	•	•	•	•	•
<b>F5</b>				•	•	•	•
<b>F6</b>		•					
<b>F7</b>			•				
<b>F8</b>	•				•		

Table 3.4: Mapping of the use cases onto the functional requirements

---

# Design

In this chapter the choice of various technologies is made. Design of individual components is created. The components are shown in the context of the whole application in the deployment model. It also takes the previous chapters – Survey and Analysis – into consideration, builds upon them and specifies them. Several wireframes are proposed which act as a template for the Implementation phase. In this chapter a design of the used algorithm and an application structure is discussed as well.

## 4.1 Selected Technologies

The technology choices depend mainly on functional and non-functional requirements. Moreover, they also depend on suitability and convenience to use. Finally, if there still exists more than one appropriate possibility after the preceding considerations, the choice may become a matter of personal preference.

In this subchapter all the technologies as frameworks and main libraries are mentioned. Next to the names, also the reasoning behind the choices is discussed. The other similar technologies that were not selected but taken into consideration are mentioned as well.

### Ruby on Rails

Since the non-functional requirements contain platform independence (N6) and accessibility (N10) and the functional requirements contain displaying data using GUI (F4), a web application seems to be the most appropriate solution for this problem. After the type of application was chosen, the next decision was about whether to use a web framework or not. And if so, which one to choose. This choice was again influenced by the functional and non-functional requirements. The code should be able to be extended easily (N9) and it should be open-source (N5). Moreover, it should store data in a

database. A web framework helps a programmer with meeting these requirements. It provides a robust application structure in case the work is extended as an open-source project and it simplifies working with a database. There are quite a lot of suitable frameworks using various programming languages which are widely used and supported – Django<sup>50</sup>, Rails<sup>51</sup>, Spring<sup>52</sup>, Laravel<sup>53</sup>, Express<sup>54</sup>, etc.

Ruby on Rails is the final decision. The current version of the framework is 5.2.3 and it works with Ruby version 2.6.1. It is a model-view-controller framework based on Ruby language which supports dependencies – called gems – that can extend application functionalities really easily. It supports the real-time communication – through its Action Cable<sup>55</sup> component – as a major building block of this work as well. Another advantage is one can write both back-end and front-end (in large part) in one programming language Ruby. In addition, it supports the web server–database communication through its Active Record<sup>56</sup> component which allows using various databases such as SQLite, PostgreSQL, MariaDB, etc. There also exist officially supported ODM<sup>57</sup> frameworks dealing with other types of databases, such as Mongoid<sup>58</sup> for MongoDB.

### SQLite

Since the database model is not going to be complex, basic RDBMS is sufficient to use. The current version of the sqlite3 gem<sup>59</sup> is 1.4.0 and it supports SQLite 3.6.16 database engine or newer. The advantages of SQLite<sup>60</sup> are that the database is public domain and open-source, it is light-weight and fast, its file format is cross-platform and the maximum database size is 140 terabytes. Finally, Rails framework comes with a build-in support for SQLite.

### WebSockets

There is a need to send the real-time data to a client’s web browser. There are several approaches which allow this kind of communication – Polling, Long Polling, Server-Sent Events and WebSockets. First three are compared

---

<sup>50</sup><https://www.djangoproject.com/>

<sup>51</sup><https://rubyonrails.org/>

<sup>52</sup><https://spring.io/>

<sup>53</sup><https://laravel.com/>

<sup>54</sup><https://expressjs.com/>

<sup>55</sup>[https://guides.rubyonrails.org/action\\_cable\\_overview.html](https://guides.rubyonrails.org/action_cable_overview.html)

<sup>56</sup>[https://guides.rubyonrails.org/active\\_record\\_basics.html](https://guides.rubyonrails.org/active_record_basics.html)

<sup>57</sup>[https://en.wikipedia.org/wiki/Object-relational\\_mapping#Object-oriented\\_databases](https://en.wikipedia.org/wiki/Object-relational_mapping#Object-oriented_databases)

<sup>58</sup><https://docs.mongodb.com/mongoid/current/>

<sup>59</sup><https://rubygems.org/gems/sqlite3/versions/1.4.0>

<sup>60</sup><https://www.sqlite.org/index.html>

in the diagram 4.1. The last one – WebSockets – is depicted in the diagram 4.2. The reasoning behind why WebSockets have been chosen is as follows:

- Regular polling – The regular polling works with the HTTP protocol which is stateless. A client creates and sends a request to a server. The server calculates the response and sends it back to the client. This would repeat in regular interval for all the real-time data client requests for. The disadvantage is that this approach is quite demanding – each new request needs a creation of a new incoming connection that must be established, HTTP headers must be sent, each new request must be processed on the server, new data must be generated and sent back to the client and then connection must be cleaned and closed. So the client would have to create the same requests all over again – that is something we do not want.
- Long Polling – This technique is basically a variation on the regular polling except for it is improved [129]. The beginning of the communication is the same – a client sends a request and waits until he or she gets a response. The difference is the server does not respond immediately after getting the request but waits. It waits until there are new (different) data or a request timeout has been reached and only then it sends them back to the client, thereby completing the request–response cycle. As soon as the client receives the new data it sends a new request again. The disadvantages are a possible timeout, a time-consuming scaling – the whole server may be blocked if there are lots of clients connected (lots of threads used) and again superfluous requests created by a client.
- Server-Sent Events – The SSE is another technique utilising HTTP protocol which is quite new (2004). It is standardized as part of HTML5 by the W3C [130]. This technique allows the server to asynchronously send (push) data to a client as soon as the client-server connection is established. It is a one-way communication from server to client as one can see in the diagram 4.1. Implementation of SSE is quite simple – one uses EventSource object on client’s side to open connection to the server and then just listens for messages using `onmessage` Event<sup>61</sup>. At first sight it looks like the optimal technology for our problem. Unfortunately, there are significant disadvantages emerging when taking a closer look at it. SSE suffers from a limitation to the maximum number of open connections. Even though there exist polyfills<sup>62</sup>, SSE is not supported by some of the major web browsers including Edge (it

---

<sup>61</sup>[https://www.w3schools.com/jsref/event\\_onmessage\\_sse.asp](https://www.w3schools.com/jsref/event_onmessage_sse.asp)

<sup>62</sup><https://developer.mozilla.org/en-US/docs/Glossary/Polyfill> and <https://github.com/Yaffle/EventSource>

is marked as “Under consideration”<sup>63</sup>), Edge Mobile and Internet Explorer. Furthermore, the support from Ruby on Rails is quite weak. The framework contains a class which deals with this type of approach – `ActionController::Live`<sup>64</sup> – yet it is not well-documented and it is less used in the Rails community than Action Cable representing the WebSockets.

- WebSockets – This type of communication represents a new type of protocol which was standardised in 2011 and is located in the 7th layer of OSI model as well as HTTP [68]. It introduces a full-duplex communication channels where a client can send messages to a server and vice versa over a TCP connection. The connection consists of three steps as you can see in the diagram 4.2: handshake (using HTTP Upgrade header), communication itself and closing the connection by either server or client side. Since it provides bidirectional communication, the application could be easily extensible in future work if necessary. It is supported by all the major web browsers. It represents a low latency solution and the messages come with little overhead. Ruby on Rails comes with a native built-in implementation of WebSockets called Action Cable<sup>65</sup>. Action Cable component is well-documented and maintained by the framework. It offers publish–subscribe pattern which is optimal solution for this application. Any client which wants to view the real-time data just subscribes to a specific channel and so becomes a subscriber. The server acts as a publisher and broadcasts different data to various channels. Every client which is connected (subscribed) to a specific channel receives data from that channel.

In the end, WebSockets as a new client–server communication type is the most suitable solution for our application since it supports important properties: scalability, potential future work, lightweight and fast communication, modern technology, wide web browser support, well written documentation with support of Rails community.

---

<sup>63</sup><https://developer.microsoft.com/en-us/microsoft-edge/platform/status/serversenteventseventsourc/?q=sort%3AVotes>

<sup>64</sup><https://api.rubyonrails.org/classes/ActionController/Live/SSE.html>

<sup>65</sup>[https://guides.rubyonrails.org/action\\_cable\\_overview.html](https://guides.rubyonrails.org/action_cable_overview.html)

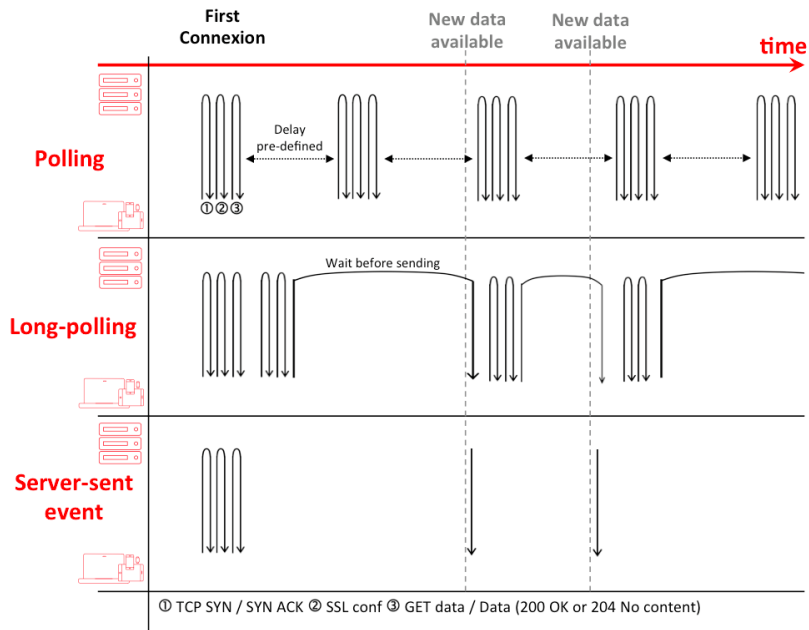


Figure 4.1: Polling, Long Polling and Server-Sent Events comparison diagram [131]

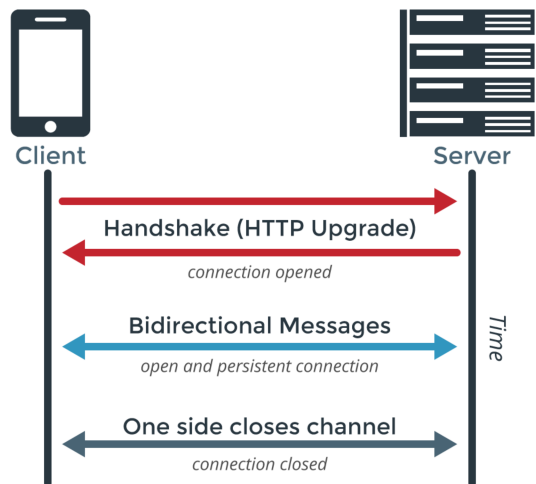


Figure 4.2: WebSocket communication diagram [132]

## JavaScript & jQuery

JavaScript as a language and jQuery as a JavaScript library will be used. The reason is Ruby on Rails offers CoffeeScript<sup>66</sup> or JavaScript to program the client side. Furthermore, Action Cable's client side is also based on JavaScript. Since jQuery is imported to the framework as a gem, the jquery-rails gem version 4.3.3<sup>67</sup> will be used. It will utilise channel management as a part of the Action Cable as well as it will be used to change the website dynamically. Besides that, it will also handle necessary JavaScript elements datepicker, graphical representation of data, etc.

## Graphs & Datepicker

In order to depict the historical data in a well-arranged way for a user a graphical representation of the data will be used. There are plenty of JavaScript libraries providing sufficient graph visualisation support such as: C3.js<sup>68</sup>, Flot<sup>69</sup>, Tau charts<sup>70</sup>, Chart.js<sup>71</sup>, HIGHCHARTS<sup>72</sup>, etc. In the end the Chart.js was chosen owing to several reasons. A functioning Rails gem named chart-js-rails (version 0.1.6<sup>73</sup>) exists and will be used. The library is simple, interactive and good-looking. There is also a possibility to create the graphs responsive. It is supported by all the major desktop and mobile web browsers (it uses HTML5 Canvas). The code is open-source, too. Lastly, there is an important feature which allows a user to hide or show individual data sets in the graph.

Another useful element is a datepicker. A user will be able to choose a date to view historical data on that day. There are several options how to approach this problem. One way is to let the user fill in the date. Apart from the fact that it is not a user-friendly solution, other problems occur – we would have to parse and check given input. Another way is to rely on web browser implementation of `<input type="date">` or use a 3rd party solution. To rely on a web browser solution which shows a calendar where a user can choose a date is not a good idea. Safari and Internet Explorer do not offer this feature. In the end, the 3rd party solution was chosen – jQuery UI library<sup>74</sup>. The Datepicker is one of its widgets and can be easily adjustable. There is a gem dealing with this library – jquery-ui-rails version 5.0.5<sup>75</sup>. The advantages are it is open-source, one can restrict available dates to be selected, a format of the date can be chosen, etc.

---

<sup>66</sup><https://coffeescript.org/>

<sup>67</sup><https://rubygems.org/gems/jquery-rails/versions/4.3.3>

<sup>68</sup><https://c3js.org/>

<sup>69</sup><http://www.flotcharts.org/>

<sup>70</sup><https://www.taucharts.com/>

<sup>71</sup><https://www.chartjs.org/>

<sup>72</sup><https://www.highcharts.com/products/highcharts/>

<sup>73</sup><https://rubygems.org/gems/chart-js-rails/versions/0.1.6>

<sup>74</sup><https://jqueryui.com/>

<sup>75</sup><https://rubygems.org/gems/jquery-ui-rails/versions/5.0.5>



## JSON, Network Library & Sass

Taking the REST APIs into consideration, all the analysed cryptocurrency exchanges use JSON as a default data response format. Thus, one will need to work with JSON while parsing the incoming data. That should not be a problem because the newer Ruby on Rails versions has the JSON gem<sup>76</sup> already bundled by default.

In order to actually get data from a cryptocurrency exchange one will need to use a network library. Ruby provides a HTTP client API through its standard library named `Net:HTTP`<sup>77</sup>. It allows to create a HTTP request easily and handle the corresponding response.

Sass (Syntactically awesome style sheets) is a preprocessor scripting language which is compiled into CSS. Since Ruby on Rails version 3.1 all the Rails projects come with Sass by default. It is again imported as `sass-rails` gem version 5.0<sup>78</sup>. Its newer syntax SCSS (Sassy CSS) is used in Rails (with file extension `.scss`). The advantages of this language are variable definition possibility, loops, code nesting, etc.

## 4.2 Deployment Model

The deployment model represents a relation between a software and a hardware when the application is in operation. It shows which software unit works with which hardware unit and how the software or hardware units are connected among each other. The model is depicted clearly by the deployment diagram 4.3. Individual units such as modules, components, libraries, frameworks and devices are shown in the context of the whole system.

There are two main nodes depicted:

1. Client's computer – On one side there is a client who possesses a computer or a mobile phone or other similar device which runs a web browser. There are several libraries used on the client's side such as `Chart.js` or `jQuery UI` in order to show specific elements on the historical data page. In addition to the libraries, there exists also the `Action Cable` component which ensures `WebSocket` connection maintained between the client and the server when the client wants to view the real-time data.
2. Server – On the other side there is a running server. The client communicates with the server through the web browser. There are two protocols used for communication between the server and the client – `HTTP` and the already mentioned `WebSocket`. `HTTP` is used for basic

---

<sup>76</sup><https://rubygems.org/gems/json>

<sup>77</sup><https://ruby-doc.org/stdlib-2.6.3/libdoc/net/http/rdoc/Net/HTTP.html>

<sup>78</sup><https://rubygems.org/gems/sass-rails/versions/5.0.0>

request-response cycle – getting all the HTML pages and getting the historical data from the database to the client’s web browser. Ruby on Rails framework uses Puma as an application server and SQLite3 as a relational database management system. Communication between the database and the application itself is ensured by ORM using Active Record pattern. The server also runs Triangular arbitrage library module which communicates with 3rd party APIs in order to get both the real-time and the historical data.

### 4.3 Wireframes

Wireframes are one of the possibilities how to clearly visualise the design of the application. They show a programmer’s idea how the GUI should look like. There is supposed to be more than one wireframe – each screen is visualised. There is no need to design all the details. Basic ideas and graphics are provided. No excess colours or effects are depicted. In addition, it shows the elements’ layout on the page.

There are two wireframes depicted – wireframe 4.4 and wireframe 4.5. Each one of them represents a GUI of the specific web page (screen):

1. Real-time data page – There is a heading on the top of the page indicating where a user is located. Under the heading there is a control panel where the user can choose from several options. The first element on the left is a button which redirects the user to the page with Historical data. In the middle of the control panel there is an option of changing the order of the real-time data by the potential profit. The right side of the control panel shows the cryptocurrency exchanges to be selected. When a user selects specific cryptocurrency exchange only real-time data from that exchange will be displayed. Under the control panel there is a table containing all the displayed real-time data structured to five columns – Profit, Currencies, Pair #1, Pair #2 and Pair #3. The profit is depicted in percentage including cryptocurrency exchange’s fees. The currencies represent the triangle of the potential triangular arbitrage execution. The pairs represent the quoted pairs involved in the potential triangular arbitrage execution on the specific cryptocurrency exchange. Since there may be a lot of potential triangles to show, only the first 20 triangles will be displayed.
2. Historical data page – Again there is a heading on the top of the page indicating where a user is located. There is a control panel as well. The control panel consists of two elements – a Realtime data button and a datepicker. The Realtime data button redirects the user to the page with real-time data. The datepicker allows the user to pick a date.

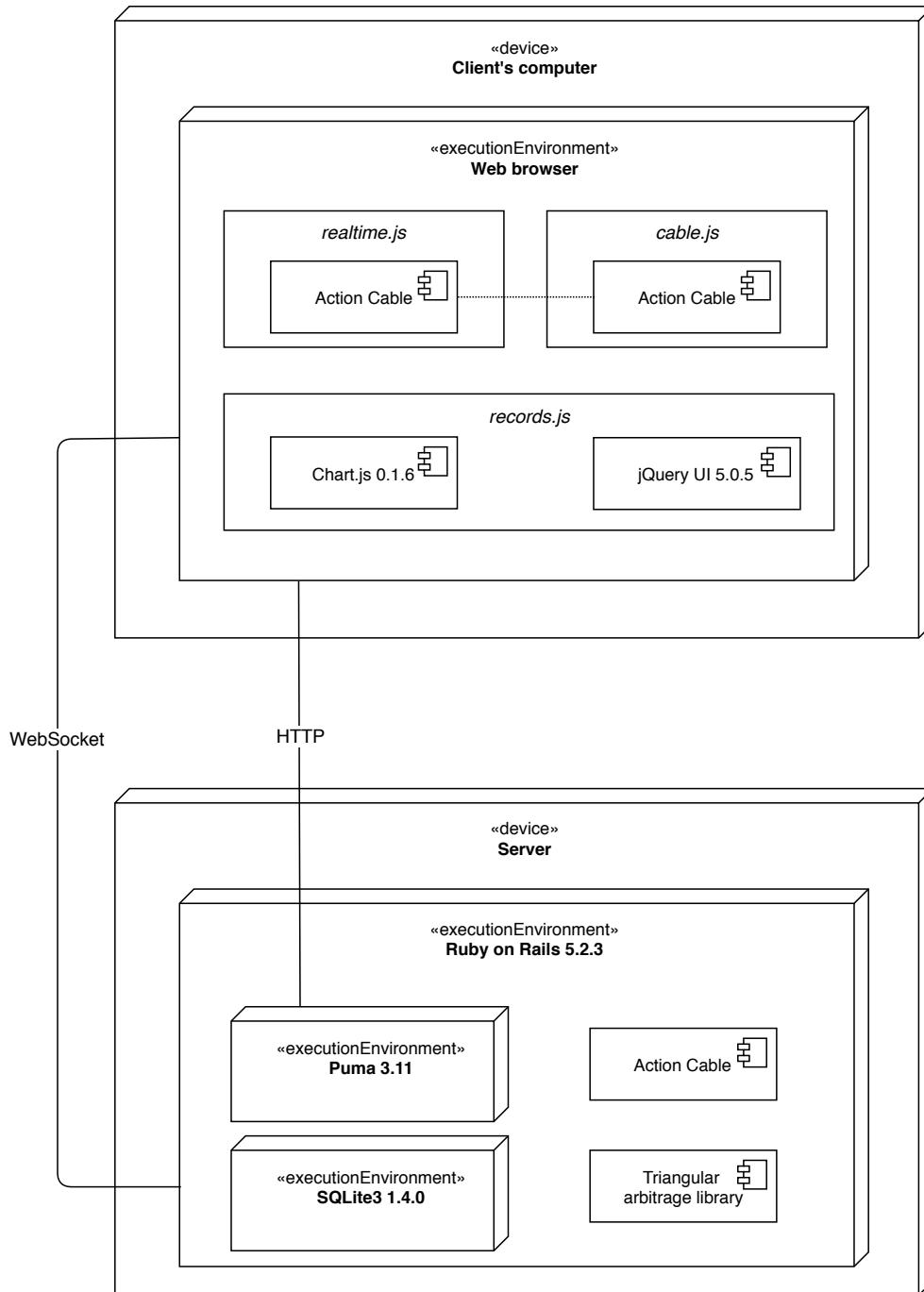


Figure 4.3: Deployment diagram

The date represents the historical data from that day. When the date is confirmed the historical data from that day are displayed – there is one record per hour per cryptocurrency exchange. The returned data appear both in the table and in the chart below the control panel. Under the control panel there is a chart depicting all the data from the table below the chart in a well-arranged way. Under the chart there is the already mentioned table containing the historical data. Besides the columns contained in the *Real-time data page* there are two more columns – Exchange and Date – adding the exchange’s name and exact date. The exact date represents a moment when the best triangular arbitrage opportunity appeared within the specific hour.

## 4.4 Application Structure and Algorithm

The structure of the application – all the major parts – are designed and described in this subchapter. The files and modules will be located in the framework’s recommended directory, taking into consideration the strict approach of the Ruby on Rails framework. Moreover, the main logic is designed and the used algorithm along with possible alternatives is discussed.

The application will be divided into the following main parts:

- Client side – view, supporting logic
- Server side – channels, controller, model
- Triangular arbitrage library
- Initializer

### Client side

The Client side will consist of two main parts. The first one – view – will form the content of the particular web pages. Not only the static HTML elements will be included in the view files. But also the eRuby (Embedded Ruby) language will be used in order to add some dynamic content – created in the controller – on the web pages. The content will be styled with `.css` (or `.scss`) files.

Apart from the view, the second one – supporting logic – will be programmed. JavaScript language will be used for that. Each view file will be accompanied by a JavaScript file. Elements like the chart, datepicker, etc. will be set. The type of the chart will be a line chart with several data sets. The user will be able to hide or show any data set in the chart. One data set will represent data from one cryptocurrency exchange. The units in the data

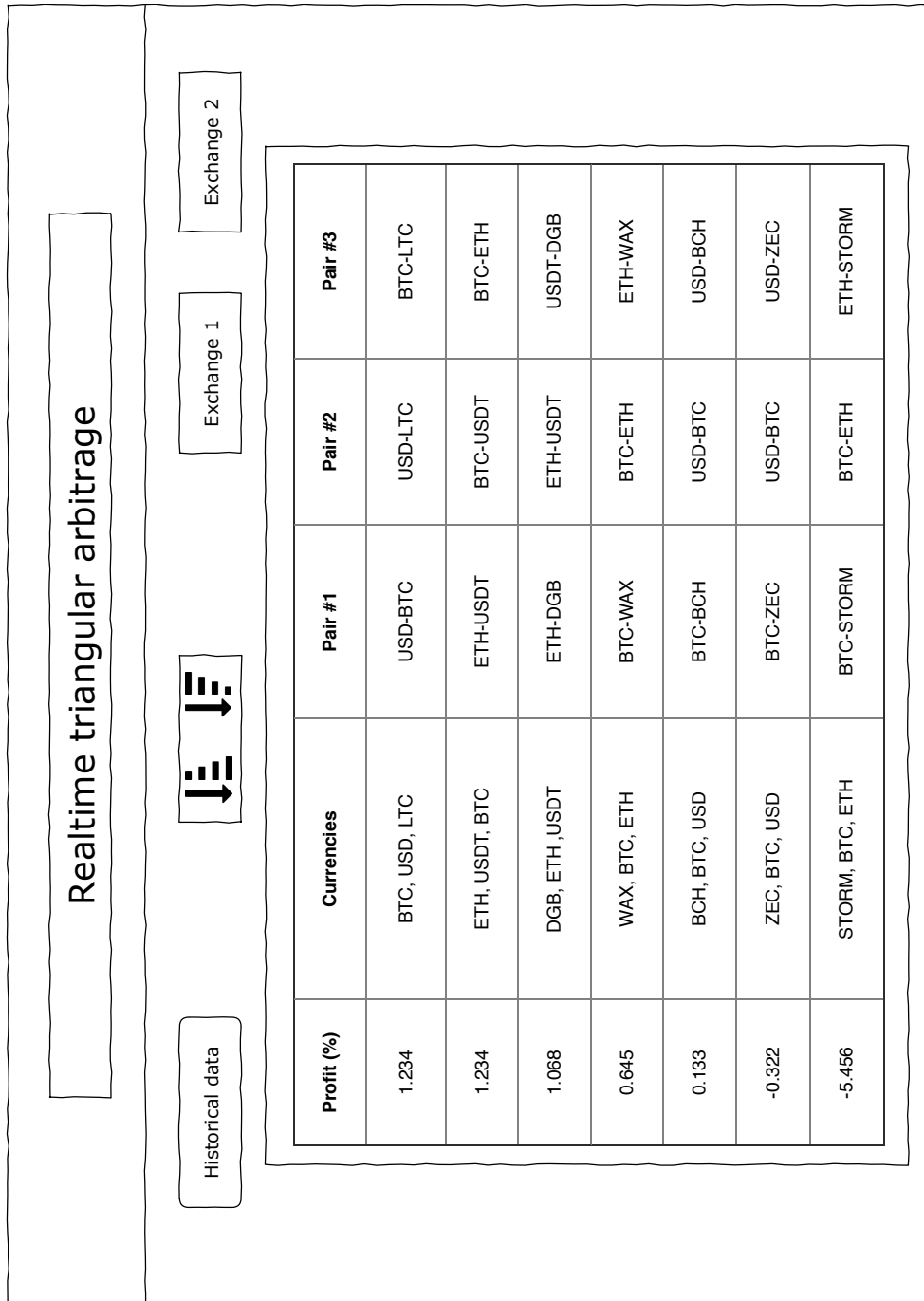


Figure 4.4: Wireframe of the *Real-time data* page

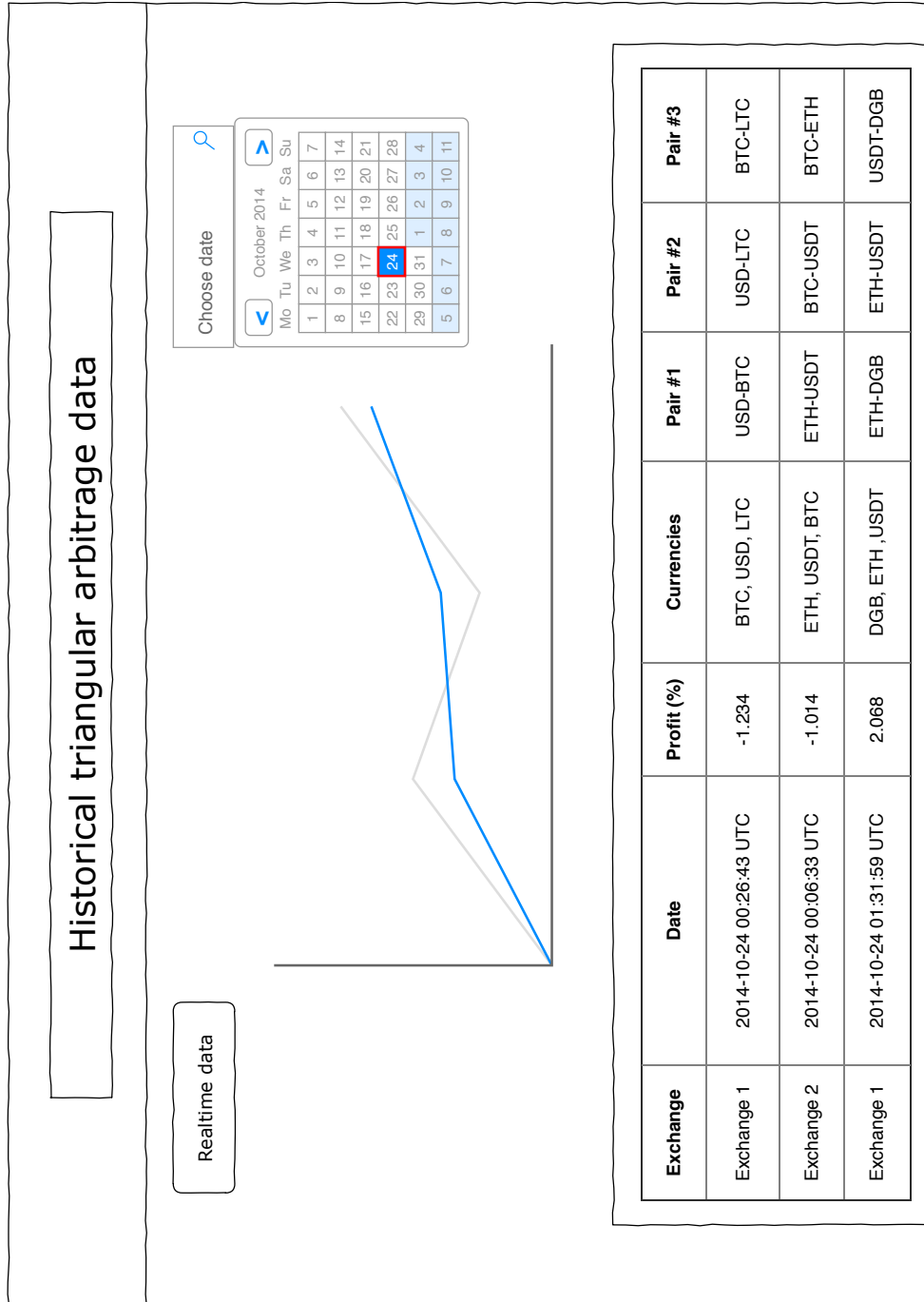


Figure 4.5: Wireframe of the *Historical data* page

set will be the database records from the selected date. The datepicker will fill the HTML `input` for the user and the user will not be able to write anything to it. The user will also not be able to submit the HTML `form` until a date is picked. The JavaScript will also handle the WebSocket (Action Cable) logic on the client's side and the typical interaction between the user and the web browser as well.

### Server side

The Server side will consist of the three main parts – channels, controller and model. The channels will be created so the server will be able to broadcast the real-time data to the client via the Action Cable component. Each cryptocurrency exchange will have its own channel. When another cryptocurrency exchange will be selected by a user the server will be noticed and will be able to react to that change.

The controller will handle the business logic when a new HTTP request will be received. Especially the *Historical data page* will have to handle the data from the database. The date selection logic and the logic of the request of this page will be handled in the controller as well. When a user accesses the *Historical data page* the server will respond with the data of the day the request is sent (UTC time).

The model will be straightforward and quite simple. Since there will not be any complicated interconnection needed among the downloaded data, one table in the database will be sufficient. The table will consist of:

- Exchange name – type `:string`
- Date – type `:datetime`
- Profit – type `:float`
- Currencies – type `:string`
- Pair #1 – type `:string`
- Pair #2 – type `:string`
- Pair #3 – type `:string`

### Triangular arbitrage library

The Triangular arbitrage library will represent a module which will communicate with the 3rd party APIs – cryptocurrency exchanges. It will implement methods which download real-time data from the chosen cryptocurrency exchanges. If a REST API is used the real-time data will be downloaded periodically as often as possible (preferably every second). Otherwise WebSockets

would be used and in that case the real-time data would be pushed to the client as soon as there is a change.

There will be two cryptocurrency exchanges chosen and this library will download real-time data from them. The preferred cryptocurrency exchange is Kraken because of the fact only the REST API is offered despite its large volumes while writing this text. Therefore, there might be good triangular arbitrage opportunities since the limit for the public calls is 1 request/sec. and the traders might call the API at different points in time. In addition, Kraken represents the largest Bitcoin exchange in view of euro volume and liquidity. Another exchange suitable for this task is Bittrex because its REST API offers an endpoint (<https://api.bittrex.com/api/v1.1/public/getmarketsummaries>) which returns all the data we need for our task. Unfortunately the WebSocket option requires SignalR and now there are no (maintained) SignalR clients for Ruby. In addition, Bittrex is well known for its safety and security – it has never been hacked [133, 134]. It is one of the biggest cryptocurrency exchanges in terms of the trading pairs count – it offers more than 300 of them. Moreover, it is well respected across the blockchain community.

On top of the real-time data downloading the module will also process the data in order to get the potential profit. There will be `Exchange` class which will be a superclass to all the implemented cryptocurrency exchanges. It will create requests, send them to the 3rd party APIs and get the responses. It will also compute the potential profit and prepare the processed data to be sent to the client. Then there will be `KrakenExchange` and `BittrexExchange` that will inherit from the `Exchange`. Each one of them will implement methods which will process the response data and create special structures from them. There will be `pairs` and `triangles` collections needed. The `pairs` collection will keep all the pairs on the given cryptocurrency exchange. The `triangles` collection will keep all the possible triangles (e.g. `BTC → LTC → USD`) with which one can execute triangular arbitrage. In addition, the mentioned superclass will also declare methods which will need to be overridden by the subclasses such as `create_pairs`, `update_pairs`, `create_triangles`. The big picture is shown in the diagram 4.6.

### Initializer

The Initializer ensures the Triangular arbitrage library module is initialized as soon as the server is up and running. It involves an initialization of all the implemented cryptocurrency exchange classes such as `KrakenExchange`, `BittrexExchange` and the other ones implemented in the future. Then it will create all the corresponding channels and it will start broadcasting the real-time data to that channels. In addition, a logic of adding new records to the model (to the database) will be implemented here. During each hour the best potential profit for each cryptocurrency exchange will be kept. At the end of



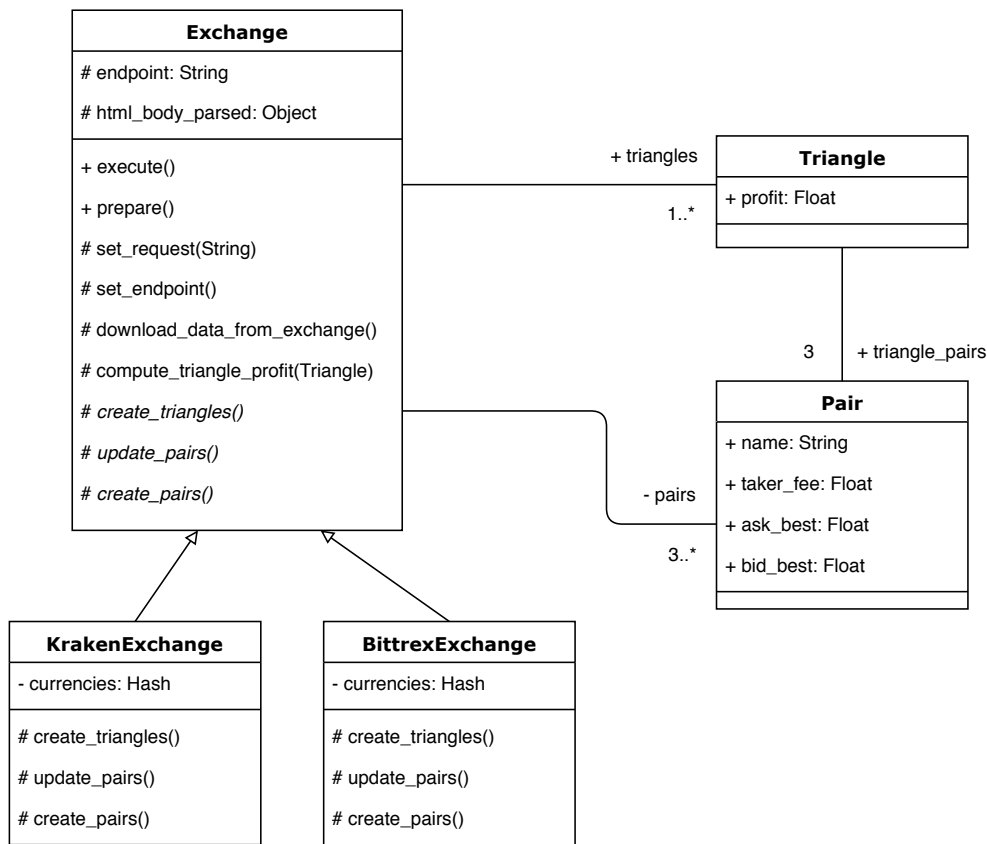


Figure 4.6: Class diagram of the Triangular arbitrage library

each hour all these best potential profits will be stored to the database. Since the application will store the new records every hour, we should estimate how much data will be stored during a time period. Let us consider there will be 20 cryptocurrency exchanges implemented in the application in the future. In addition we assume one record takes around 100 bytes. Then the data size per year equals:

$$\begin{aligned}
 &20 \text{ records} \cdot 100 \text{ bytes} \cdot 24 \text{ hours} \cdot 365 \text{ days} = \\
 &= 17520000 \text{ bytes} \approx 16.7 \text{ MiB}
 \end{aligned}$$

Because 16.7 MiB per year is not much in terms of both server disk space and the maximum database size the database will not trigger any periodic deletion tasks.

## Algorithm

The problem of finding the best triangular arbitrage opportunity can be converted to a graph problem in the following manner. The vertices are individual (crypto)currencies. The edges are weighted and represent the exchange rates as  $r_i(t) \cdot (1 - f_i)$  ratio as expressed in the equation (3.2). The task is then to find the best simple cycle having 3 vertices containing a given vertex. The best here means having the product of weights (exchange rates) maximal. Usually, in classic algorithms that solve this problem (such as the Bellman-Ford algorithm), the weights are added, not multiplied. We can overcome this by considering the opposite of logarithms of the exchange rates (ratio) for the weights. Since the logarithm of a product is the sum of the logarithms as stated in equation (4.1), we obtain a problem of finding the shortest path where the weight of a path is the sum of the weight of its individual edges.

$$\log_b(x \cdot y) = \log_b(x) + \log_b(y) \quad (4.1)$$

Nevertheless, the application will need to get the potential profit of all the triangles (triangular arbitrage opportunities) of the specific cryptocurrency exchange. In other words, we do not need to find only the shortest simple cycle, but all the simple cycles starting from a given vertex. Therefore, the final approach will use the following algorithm (instead of an algorithm for finding the shortest path). It will precompute all the possible triangles at the beginning. After that the application will only periodically update the exchange rate of the trading pairs and recalculate the potential profit of the already precomputed triangles.

Since the number of the precomputed triangles will not be large, neither the time complexity nor the space complexity should be a problem. The precomputation will take  $\mathcal{O}(n^3)$  time where  $n$  represents the number of the (crypto)currencies present on a cryptocurrency exchange. Even though  $n$  usually equals to tens or hundreds (surely less than 1000), the cryptocurrency exchanges have usually less than 10 different markets. That means there are only less than 10 (crypto)currencies which constitute one part (the base currency in most cases) of all the trading pairs quoted on the cryptocurrency exchange. This fact significantly reduces the number of all the possibilities and thus it reduces also the complexity of the algorithm. The recalculation of the profit and sorting will take  $\mathcal{O}(m + m \cdot \log_2(m)) = \mathcal{O}(m \cdot \log_2(m))$  time where  $m$  represents the number of precomputed triangles. That is another positive fact since  $m$  usually equals to tens or hundreds as well.

---

# Implementation

In this chapter the application is divided into several logical parts and each part is commented. The parts are comprised of files. The files are also commented and important functions and methods are explained. The technologies such as libraries, frameworks, etc. have been used according to the subchapter 4.1. All the already mentioned versions have been employed. The whole application is maintained as a repository on GitHub website [135].

The program structure generally corresponds to the designed one in the subchapter 4.4. It complies with the Ruby on Rails framework structure so all the directories mentioned in the text are derived from it. The program structure can be divided as follows:

- Client side
- Server side
- Triangular arbitrage library
- Initializer

## 5.1 Client Side

The Client side contains three types of files:

- Logic – JavaScript files with `.js` extension
- Content – eRuby files which are basically HTML files enriched with Ruby code with `.html.erb` extension
- Styles – Cascading Style Sheets files and Syntactically awesome style sheets files with `.css` and `.scss` extensions

## 5. IMPLEMENTATION

---

The files with *Logic* are located in the same directory and are the following:

```
app/assets/javascripts/realtime.js
app/assets/javascripts/records.js
app/assets/javascripts/cable.js
```

The `realtime.js` file belongs to the view with the real-time data. It executes as soon as a user accesses the home page (resource path: `/`) or the *Real-time data page* (resource path: `/realtime`). It creates the default subscription to Bittrex cryptocurrency exchange channel named `BittrexChannel` and the user becomes a subscriber. There is `receivedData(data)` callback function which is called as soon as the real-time data are received by the client. The callback function parses the incoming data which are sent in JSON format and generates the HTML table with those data. If the subscribed cryptocurrency exchange experiences some problems the error message is displayed. The code also handles the user–web page interaction such as selecting another cryptocurrency exchange or redirecting to the *Historical data page*.

The `records.js` file belongs to the view with the historical data. It executes as soon as a user accesses the *Historical data page* (resource path: `/records`). It sets the datepicker widget in a way the last day which can be picked is the current date and the format is `YY-MM-DD`. Moreover, the data sets for the chart are created, the chart is initialized and set to be responsive. It also guarantees the date submit button can be clicked only when the date is picked.

The `cable.js` file ensures the client is connected to the server properly. It sets up a consumer instance of the connection. Nevertheless, the connection itself is not established until a subscription is specified.

The files with *Content* are located in the `views` directory and are the following:

```
app/views/realtime/index.html.erb
app/views/records/index.html.erb
app/views/layouts/application.html.erb
app/views/layouts/_footer.html.erb
```

These files form the whole HTML content of the application with the aid of eRuby. They ensure a user can move around the website. The table with historical data is created in `records/index.html.erb` file.

The files with *Styles* are located in the same directory and are the following:

```
app/assets/stylesheets/application.css
app/assets/stylesheets/realtime.scss
app/assets/stylesheets/records.scss
```

They provide the styling of all the HTML elements on the page. Since CSS3 also supports the animations, these are provided as well as the transitions and transformations. That involves the table and the error notice pulsing whenever the new data are received and the transitions when a user selects another cryptocurrency exchange or data order.

## 5.2 Server Side

The Server side consists of three parts:

- Controller
- Model
- Channels

The files representing the *Controller* part are in the same directory and are the following:

```
app/controllers/records_controller.rb
app/controllers/realtime_controller.rb
app/controllers/application_controller.rb
```

The `records_controller.rb` file contains the logic which is related to the *Historical data page*. When a user accesses it the application displays the data from the current day. If another day is picked and confirmed by the user the application shows data from that day.

The `realtime_controller.rb` has just the `index` method (action) declared because the framework requires it.

The files representing the *Model* part are the following:

```
db/migrate/YYYYMMDDHHMMSS_create_records.rb
db/schema.rb
app/models/record.rb
```

The `YYYYMMDDHHMMSS_create_records.rb` file is responsible for creating the database structure. It is called a database migration file. The migration files help to modify or create the database tables easily.

The `schema.rb` file represents the database. There are 7 major columns:

- `triangle_name` – consists of the three involved (crypto)currencies
- `pair1` – represents the Pair #1
- `pair2` – represents the Pair #2
- `pair3` – represents the Pair #3
- `profit` – represents the profit of this triangle
- `exchange` – represents the name of the cryptocurrency exchange
- `date` – represents the date and time when the triangular arbitrage opportunity was present

The files representing the *Channels* part are the following:

```
app/channels/application_cable/channel.rb
app/channels/application_cable/connection.rb
app/channels/bittrex_channel.rb
app/channels/kraken_channel.rb
```

All the four files are empty because the application (its server side) does not need to react either when a user subscribes to a channel or unsubscribes from it. Moreover, the application also does not react to any incoming messages since the client does not send any messages to the server for now.

### 5.3 Triangular Arbitrage Library

The library works as a middleware between the server and the cryptocurrency exchange APIs. It communicates with the APIs, downloads data from them in real-time, processes them and provides them to the *Initializer*. The library is located in this directory:

```
app/lib_triangular_arbitrage
```

The code in the library is divided into two modules – `TriangleDownloader` and `Structures`. The modules correspond to the directories. The first module consists of:

- `Exchange` class – located in `exchange.rb` file
- `KrakenExchange` class – located in `kraken_exchange.rb` file
- `BittrexExchange` class – located in `bittrex_exchange.rb` file

The `Exchange` class is an ancestor of `KrakenExchange` and `BittrexExchange` classes. If anyone wants to implement another cryptocurrency exchange that will also be a descendant of the `Exchange` class. The functionality of this class is divided into several methods.

The `execute` method is called every second by the `Initializer` and it updates all the trading pairs within the cryptocurrency exchange and finds the potential profits for all the triangles.

The `prepare` method is called at the beginning when the object is created, it sets the API's endpoint and creates all the necessary structures that are needed later.

The `set_request(endpoint)` method sets the network objects (endpoint URL and request) in order to connect to the endpoint.

The `download_data_from_exchange` method sends the request and gets the response from the API. It also parses the response.

The `remove_duplicate_triangles` method removes all the duplicates among the triangles which have been created so the application is less memory-intensive and less processor-intensive.

The `triangulate` and `compute_triangle_profit(triangle)` methods together compute potential profit (including the fees) of all the triangles, sort them in descending order and save the best 20 and the worst 20 of them. The code of `compute_triangle_profit(triangle)` which is important for the whole application is as follows:

```

1 def compute_triangle_profit(triangle)
2   profit = 1.0
3   triangle.triangle_pairs.each_index do |i|
4     if triangle.is_reversed[i]
5       profit = profit *
6         triangle.triangle_pairs[i].bid_best
7     else
8       profit = profit /
9         triangle.triangle_pairs[i].ask_best
10    end
11    profit = profit -
12      (profit * triangle.triangle_pairs[i].taker_fee)
13  end
14  return ((profit - 1) * 100).round(4)
15 end

```

Then there are methods which need to be overridden in the descendants: `set_endpoint`, `create_pairs`, `update_pairs` and `create_triangles`.

The **KrakenExchange** and **BittrexExchange** classes implement two main functionalities – downloading and parsing the real-time data – which are both inherently cryptocurrency exchange specific.

The `set_endpoint` method creates a valid endpoint which is then used for requesting the API.

The `create_pairs` method is called at the beginning right after the object is created. It saves all the useful pairs the cryptocurrency exchange offers.

The `update_pairs` method downloads the real-time data from the cryptocurrency exchange and refreshes the old saved `pairs`. It is called every second.

The `create_triangles` method is called at the beginning right after the object is created. It creates all the triangles which are present on the cryptocurrency exchange. Later on, the triangles are used every second to check the triangular arbitrage opportunities.

The second module – **Structures** – consists of:

- **Pair** class – located in `pair.rb` file
- **Triangle** class – located in `triangle.rb` file

The **Pair** class keeps all the necessary attributes of a pair on a cryptocurrency exchange:

- `name` – represents the pair name (e.g. BTC-ETH)
- `bid_best` – represents the best bid (e.g. 0.123)
- `ask_best` – represents the best ask (e.g. 0.124)
- `taker_fee` – represents the fee of this particular pair (e.g. 0.25%)

The **Triangle** class keeps all the necessary attributes of a triangle on a cryptocurrency exchange:

- `triangle_pairs` array – represents all the pairs included in the triangle (e.g. BTC-ETH, ETH-STORM, BTC-STORM)
- `is_reversed` array – determines if the pairs involved are quoted reversely or not (e.g. if a pair is quoted as ETH-BTC or BTC-ETH)
- `profit` – represents the potential profit of the triangle (e.g. 1.2%)



## 5.4 Initializer

The Initializer is a part of the application where the *Triangular arbitrage library* module is used. It is created and it runs as soon as the server is up and running. All the exchange classes (`Exchange`, `KrakenExchange` and `BittrexExchange`) are initialized and prepared here. Moreover, the logic of storing data to the database and also the logic of broadcasting to the channels are located here. There is only one class in the *Initializer* named `TriangularArbitrageInitializer` with the following methods.

The `start_all_channels` method creates a new thread which handles all the communication with the 3rd party APIs.

The `start_broadcast_to_all_channels` method consists of the infinite loop with 1 second sleep where several things are checked. It is checked whether the hour has changed or not, then it broadcasts the real-time data to the channels and it checks whether there is a new highest potential profit or not.

The `prepare_all_channels` method creates and prepares the objects representing all the cryptocurrency exchanges and sets the corresponding channel names.

The `check_hours` method checks whether the hour has changed. If it has changed the `save_last_hour_record` method is called in order to store the best potential profit from the last hour to the database.

The `check_highest_profit` method checks whether the present best profit on a cryptocurrency exchange is better than the so far best one. This method is called every second.

The `save_last_hour_record` method works with the model. It saves the best potential profit from the last hour to the database.



---

# Testing

There are several options of testing the application. In this chapter the usability testing is performed. Since the application is not closed to the public and it is designed as a website, the usability testing is one of the most important types of testing. The aim is to identify the application's weak points through observing selected people (personas) using the application. Afterwards, improvement of the application based on the found weak points is done.

There are three persons testing the application. Each one of them represents a type of person (persona) who might use the website. The first one is an expert (with alias: Expert) who understands the ecosystem of cryptocurrencies and cryptocurrency exchanges. The second one is a young person who does not know cryptocurrencies well but in general is an early adopter type (with alias: Adopter). The third one is a middle-aged person (with alias: Middle-aged) who works with a computer on a regular basis but strictly on user level.

The testing follows on from the chapter 3.4 where use cases are defined. The testing is based on observing the already mentioned personas one by one. First a cryptocurrency arbitrage along with cryptocurrency exchanges are explained briefly. Then a list of tasks is presented to the personas and then they are observed how they are dealing with the tasks. After the session the personas are asked about their overall impression of the application. There are the tasks and corresponding summaries of how the personas dealt with the tasks in the text below.

- 1. Describe what you can see on the website and what it represents.** [current page: /realtime]

All the three personas did not have any serious problems with describing what they see. The Expert recognised and described everything on the page. He was only not sure if the Profit column in the table included the cryptocurrency exchange fees or not. He also suggested to change the order of the columns so the Currencies column was the first one. Both

the Adopter and the Middle-aged were not sure what the ascending and descending symbols mean. In addition, the Adopter suggested to highlight the Profit column better.

- 2. Choose among the cryptocurrency exchanges.** [current page: /realtime]

All the three personas did not have any problems with this task and reacted quickly.

- 3. Sort the real-time triangular arbitrage opportunities in ascending order.** [current page: /realtime]

Both the Adopter and the Middle-aged were hesitating and after several seconds the Adopter clicked on the right button. The Middle-aged was completely stuck in the task and asked what the right symbol for sorting meant. The Expert had no problem with that.

- 4. Interpret the first row (below the table header) in the table.** [current page: /realtime]

All the three personas knew the meaning of all the columns. The Expert was not sure about the already mentioned Profit column – if it included fees or not. That is the reason why we accompanied the task by a question “Do you think the profit does include fees?” Both the Adopter and the Middle-aged were not sure. The Adopter answered it included fees and the Middle-aged answered it did not include any fees.

- 5. Navigate to the web page with historical data.** [current page: /realtime]

All the three personas did not have any problems with this task and reacted quickly.

- 6. Describe what you can see on the website and what it represents.** [current page: /records]

All the three personas were describing the page correctly but were not sure about certain things. The Expert was complaining about the missing chart title. He was also not sure what the chart meant and asked if it depicted the best profits during the specific hours or if it depicted some special moments on the market. The Adopter had similar problem as the Expert and was not sure about the information in the chart. The Middle-aged did not see immediately the connection between the chart and the table below it. However, she interpreted (more likely guessed) the chart correctly.

- 7. View the historical data on the April 26, 2019.** [current page: /records]

---

All the three personas did not have any problems with this task and reacted quickly.

**8. Compare historical triangular arbitrage opportunities among cryptocurrency exchanges.** [current page: /records]

All the three personas did not have any major problems with that. In addition, Expert and Adopter clicked on the chart a few times and discovered it was dynamic and that they could include any cryptocurrency exchanges they wanted.

**9. View the historical records between 14:00 and 15:00 on the day you just see among all the cryptocurrency exchanges.** [current page: /records]

The Expert and the Middle-aged did not have any problems with this task and reacted quickly. The Adopter was a little slower while viewing all the records between 14:00 and 15:00. The Expert suggested to rename the Date column to Time column.

**10. View the highest potential profit during the day you just see.** [current page: /records]

All the three personas did not have any problems with this task, they used the chart and reacted quickly.

**11. Navigate back to the page with the real-time data.** [current page: /records]

All the three personas did not have any problems with this task and reacted quickly.

Based on the responses above and the final impression of the personas several changes have been made on both /realtime and /records pages. On the /realtime page these changes have been made:

- The heading has been changed from “Realtime triangular arbitrage” to “Realtime triangular arbitrage opportunities”
- The layout of the control panel has been changed – Historical data button and Exchange selection (radio buttons) have been swapped.
- The symbols used for sorting the data have been replaced with the “BEST  $\Delta$ ” and “WORST  $\Delta$ ” buttons.
- The Profit column has been renamed so the user knows it includes the fees.

## 6. TESTING

---

- The Currencies column has been renamed to the Triangle column.
- The content of the (formerly) Currencies column has been changed – the commas have turned into dashes.
- The column order has been changed to: Triangle, Pair #1, Pair #2, Pair #3, Profit (%) incl. fees

On the `/records` page these changes have been made:

- The layout of the control panel has been changed – the Realtime data button and the Date input field have been swapped.
- The title of the chart has been provided – “Best profit during each hour on YYYY-MM-DD”.
- The description of the page has been added to the control panel.
- The Date column has been renamed to the Time column.
- The column order has been changed to: Time, Profit (%) incl. fees, Triangle, Exchange, Pair #1, Pair #2, Pair #3

The GUI snapshots of the final version (including the modifications listed above) of the application are in the appendix B.

---

# Conclusion

We described the fundamental principles of Bitcoin and other selected cryptocurrencies which are helpful to understand the big picture of the thesis. What they are, how they work and what difficulties they face. After that we surveyed the cryptocurrency market and determined the largest part of it – cryptocurrency exchanges. There were attributes of cryptocurrency exchanges selected based on significance to triangular arbitrage successful execution. We then surveyed several high-profile cryptocurrency exchanges according to the selected attributes. In addition, we surveyed the already existing applications and found out they did not provide the functionality we needed. Afterwards, the problem of the triangular arbitrage was formulated (also from mathematical point of view) and the work's core requirements and use cases were analysed and defined. Based on the survey and analysis we designed the application – the technologies were chosen, the wireframes were created and the algorithm was discussed. We then programmed the application, tested it and modified it according to the tests.

The main objective of this thesis is to implement a web application which displays the real-time triangular arbitrage opportunities along with the historical data. The implementation has shown the triangular arbitrage opportunities still exist, may equal to a few percentage points and are common both on Kraken and Bittrex exchanges. Furthermore, it has provided a suitable comparison between the mentioned cryptocurrency exchanges as well.

## **Future Work**

Given the design of the application, a future work is expected and welcome. Since the application is open-source and maintained as a repository on GitHub, anyone can become a contributor. The potential new features include adding the other cryptocurrency exchanges such as Binance, Bitfinex, etc. That carries an implementation of another type of communication (WebSocket) between the application and the individual 3rd party APIs. It also includes the newly offered Kraken exchange WebSocket API which could be implemented instead of the REST API. Another feature that could be implemented in the future is the trading itself. That would involve the user account creation and other subsequent services implementation.



---

## Bibliography

- [1] CoinMarketCap. Top 100 Cryptocurrency Exchanges by Trade Volume. [online], [Accessed 14 December 2018]. Available from: <https://coinmarketcap.com/rankings/exchanges/>
- [2] Harney, A. and Stecklow, S. Twice burned – How Mt. Gox’s bitcoin customers could lose again. *Reuters*, November 2017, [Accessed 23 November 2018]. Available from: <https://www.reuters.com/investigates/special-report/bitcoin-gox/>
- [3] Gertrude Chavez-Dreyfuss. Cryptocurrency exchange theft surges in first half of 2018: report. *Reuters*, July 2018, [Accessed 1 December 2018]. Available from: <https://www.reuters.com/article/us-crypto-currencies-ciphertrace/cryptocurrency-exchange-theft-surges-in-first-half-of-2018-report-idUSKBN1JT1Q5>
- [4] Reidy, G. and Jackman S. Technology: Hackers Steal \$60 Million From Japanese Crypto Exchange Zaif. *Bloomberg*, September 2018, [Accessed 17 November 2019]. Available from: <https://www.bloomberg.com/news/articles/2018-09-19/tech-bureau-says-6-7b-yen-in-cryptocurrency-lost-in-zaif-hack>
- [5] CoinMarketCap. Global Charts. [online], [Accessed 28 November 2018]. Available from: <https://coinmarketcap.com/charts/>
- [6] Forex Capital Markets. Bitcoin Vs Forex: Understanding The Differences. [online], [Accessed 19 November 2018]. Available from: <https://www.fxcm.com/insights/bitcoin-vs-forex-understanding-differences/>
- [7] Arbitrage.expert. Crypto Arbitrage Opportunity Expert. [online], 2018, [Accessed 19 November 2018]. Available from: <https://www.arbitrage.expert/>

## BIBLIOGRAPHY

---

- [8] Cornish, C. and Murphy, H. What next after cryptocurrency bubble bursts? *Financial Times*, August 2018, [Accessed 19 November 2018]. Available from: <https://www.ft.com/content/7ed0c3b8-a1f3-11e8-85da-eeb7a9ce36e4>
- [9] Mavrides, M. *Triangular Arbitrage in the Foreign Exchange Market: Inefficiencies, Technology, and Investment Opportunities*. Quorum Books, 1992, ISBN 9780899307183. Available from: [https://books.google.cz/books?id=1Tmcn-Z\\_\\_\\_sC](https://books.google.cz/books?id=1Tmcn-Z___sC)
- [10] Investopedia, LLC. Triangular Arbitrage. *Investopedia*, [Accessed 21 November 2018]. Available from: <https://www.investopedia.com/terms/t/triangulararbitrage.asp>
- [11] Wikipedia contributors. Triangular arbitrage — Wikipedia, The Free Encyclopedia. [online], 2018, [Accessed 7 January 2019]. Available from: [https://en.wikipedia.org/w/index.php?title=Triangular\\_arbitrage&oldid=846797058](https://en.wikipedia.org/w/index.php?title=Triangular_arbitrage&oldid=846797058)
- [12] Investopedia, LLC. Altcoin. *Investopedia*, [Accessed 3 December 2018]. Available from: <https://www.investopedia.com/terms/a/altcoin.asp>
- [13] Hileman, G.; Rauchs, M. Global Cryptocurrency Benchmarking Study. *Cambridge Centre for Alternative Finance*, volume 33, 2017. Available from: <https://cdn.crowdfundinsider.com/wp-content/uploads/2017/04/Global-Cryptocurrency-Benchmarking-Study.pdf>
- [14] Wikipedia contributors. Cryptocurrency exchange — Wikipedia, The Free Encyclopedia. [online], 2018, [Accessed 8 January 2019]. Available from: [https://en.wikipedia.org/w/index.php?title=Cryptocurrency\\_exchange&oldid=875780665](https://en.wikipedia.org/w/index.php?title=Cryptocurrency_exchange&oldid=875780665)
- [15] Sagona-Stophel, K. Bitcoin 101: How to get started with the new trend in virtual currencies. *THOMSON REUTERS*, 2013. Available from: [https://web.archive.org/web/20160813163512/http://www.trssllc.com/wp-content/uploads/2013/05/White\\_Paper\\_Bitcoin\\_101.pdf](https://web.archive.org/web/20160813163512/http://www.trssllc.com/wp-content/uploads/2013/05/White_Paper_Bitcoin_101.pdf)
- [16] Bitcoin Block Reward Halving Countdown website contributors. Bitcoin Block Reward Halving Countdown. [online], 2018, [Accessed 23 November 2018]. Available from: <https://www.bitcoinblockhalf.com/>
- [17] Bitcoin Project. Frequently Asked Questions: Who created Bitcoin? [online], [Accessed 29 November 2018]. Available from: <https://bitcoin.org/en/faq#who-created-bitcoin>

- 
- [18] Wikipedia contributors. Bitcoin — Wikipedia, The Free Encyclopedia. [online], 2019, [Accessed 6 January 2019]. Available from: <https://en.wikipedia.org/w/index.php?title=Bitcoin&oldid=876832085>
- [19] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System [online]. 2008. Available from: <https://bitcoin.org/bitcoin.pdf>
- [20] Bitcoin Project. Bitcoin Developer Guide: Block Chain. [online], [Accessed 19 November 2018]. Available from: <https://bitcoin.org/en/developer-guide#block-chain>
- [21] The Economist Newspaper Limited. Blockchains: The great chain of being sure about things. *The Economist*, October 2015, [Accessed 9 December 2018]. Available from: <https://www.economist.com/briefing/2015/10/31/the-great-chain-of-being-sure-about-things>
- [22] Wikipedia contributors. Blockchain — Wikipedia, The Free Encyclopedia. [online], 2019, [Accessed 8 January 2019]. Available from: <https://en.wikipedia.org/w/index.php?title=Blockchain&oldid=877275529>
- [23] Bitcoin Project. Glossary: Genesis Block, Block 0. [online], [Accessed 2 December 2018]. Available from: <https://bitcoin.org/en/glossary/genesis-block>
- [24] Bitcoin community contributors. Genesis block — Bitcoin Wiki. [online], 2018, [Accessed 11 December 2018]. Available from: [https://en.bitcoin.it/wiki/Genesis\\_block](https://en.bitcoin.it/wiki/Genesis_block)
- [25] Bitcoin community contributors. Block — Bitcoin Wiki. [online], 2018, [Accessed 11 December 2018]. Available from: <https://en.bitcoin.it/wiki/Block>
- [26] Bitcoin community contributors. Protocol documentation — Bitcoin Wiki. [online], 2018, [Accessed 11 December 2018]. Available from: [https://en.bitcoin.it/wiki/Protocol\\_documentation#tx](https://en.bitcoin.it/wiki/Protocol_documentation#tx)
- [27] Lombrozo, E.; Lau, J.; et al. Segregated Witness (Consensus layer). [online], December 2015. Available from: <https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki>
- [28] Bitcoin Project. Bitcoin Developer Guide: Wallets. [online], [Accessed 21 November 2018]. Available from: <https://bitcoin.org/en/developer-guide#wallets>
- [29] Bitcoin community contributors. Network — Bitcoin Wiki. [online], 2018, [Accessed 10 December 2018]. Available from: <https://en.bitcoin.it/wiki/Network>

- [30] Eskandari, S.; Clark, J.; et al. A First Look at the Usability of Bitcoin Key Management. *arXiv preprint arXiv:1802.04351*, February 2018. Available from: <https://arxiv.org/abs/1802.04351>
- [31] Litke, P.; Stewart, J. Cryptocurrency-Stealing Malware Landscape. February 2014. Available from: <https://www.secureworks.com/research/cryptocurrency-stealing-malware-landscape>
- [32] ecurrencyhodler. The Advantages of Hardware Wallets. *Medium*, September 2017, [Accessed 21 December 2018]. Available from: <https://medium.com/the-litecoin-school-of-crypto/6-the-advantages-of-hardware-wallets-2d54d40b4298>
- [33] SatoshiLabs. Security:Security philosophy. [online], [Accessed 10 December 2018]. Available from: [https://wiki.trezor.io/Security:Security\\_philosophy](https://wiki.trezor.io/Security:Security_philosophy)
- [34] Bitcoin community contributors. Seed phrase — Bitcoin Wiki. [online], 2018, [Accessed 11 December 2018]. Available from: [https://en.bitcoin.it/wiki/Seed\\_phrase](https://en.bitcoin.it/wiki/Seed_phrase)
- [35] Bitcoin Project. Bitcoin Developer Guide: P2P Network. [online], [Accessed 10 December 2018]. Available from: <https://bitcoin.org/en/developer-guide#p2p-network>
- [36] Bitcoin Project. Running A Full Node. [online], [Accessed 29 November 2018]. Available from: <https://bitcoin.org/en/full-node>
- [37] Bitcoin Project. Bitcoin Developer Guide: Mining. [online], [Accessed 10 December 2018]. Available from: <https://bitcoin.org/en/developer-guide#mining>
- [38] Bitcoin Project. Glossary: nBits, Target Threshold. [online], [Accessed 11 December 2018]. Available from: <https://bitcoin.org/en/glossary/nbits>
- [39] Bitcoin community contributors. Target — Bitcoin Wiki. [online], 2016, [Accessed 4 December 2018]. Available from: <https://en.bitcoin.it/wiki/Target>
- [40] Investopedia, LLC. Target Hash. *Investopedia*, [Accessed 29 November 2018]. Available from: <https://www.investopedia.com/terms/t/target-hash.asp>
- [41] Jakobsson, M.; Juels, A. Proofs of Work and Bread Pudding Protocols(Extended Abstract). In *Secure Information Networks*, Springer, 1999, pp. 258–272. Available from: [https://link.springer.com/chapter/10.1007%2F978-0-387-35568-9\\_18](https://link.springer.com/chapter/10.1007%2F978-0-387-35568-9_18)

- 
- [42] Bitcoin community contributors. Confirmation — Bitcoin Wiki. [online], 2018, [Accessed 4 December 2018]. Available from: <https://en.bitcoin.it/wiki/Confirmation>
- [43] Payward, Inc. How long do digital assets/cryptocurrency deposits take? [online], [Accessed 4 December 2018]. Available from: <https://support.kraken.com/hc/en-us/articles/203325283-How-long-do-digital-assets-cryptocurrency-deposits-take->
- [44] Bitcoin Project. Bitcoin Developer Guide: Proof Of Work. [online], [Accessed 10 December 2018]. Available from: <https://bitcoin.org/en/developer-guide#term-target>
- [45] Bitcoin community contributors. Controlled supply — Bitcoin Wiki. [online], 2018, [Accessed 4 December 2018]. Available from: [https://en.bitcoin.it/wiki/Controlled\\_supply](https://en.bitcoin.it/wiki/Controlled_supply)
- [46] Investopedia, LLC. Block Reward. *Investopedia*, [Accessed 5 December 2018]. Available from: <https://www.investopedia.com/terms/b/block-reward.asp>
- [47] Bitcoin community contributors. Units — Bitcoin Wiki. [online], 2017, [Accessed 4 December 2018]. Available from: <https://en.bitcoin.it/wiki/Units>
- [48] Nakamoto, S.; The Bitcoin Core developers. Bitcoin Core integration/staging tree. *GitHub repository*, [Accessed 4 December 2018]. Available from: <https://github.com/bitcoin/bitcoin/blob/master/src/amount.h>
- [49] Antonopoulos, A. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. O’Reilly Media, 2014, ISBN 9781491902646. Available from: <https://books.google.cz/books?id=IXmrBQAAQBAJ>
- [50] Croman, K.; Decker, C.; et al. On Scaling Decentralized Blockchains. In *International Conference on Financial Cryptography and Data Security*, Springer, August 2016, pp. 106–125. Available from: [https://link.springer.com/chapter/10.1007/978-3-662-53357-4\\_8](https://link.springer.com/chapter/10.1007/978-3-662-53357-4_8)
- [51] Wikipedia contributors. Bitcoin scalability problem — Wikipedia, The Free Encyclopedia. [online], 2018, [Accessed 8 January 2019]. Available from: [https://en.wikipedia.org/w/index.php?title=Bitcoin\\_scalability\\_problem&oldid=875143107](https://en.wikipedia.org/w/index.php?title=Bitcoin_scalability_problem&oldid=875143107)
- [52] O’Dwyer, K. J.; Malone, D. Bitcoin Mining and its Energy Footprint. June 2014. Available from: <https://digital-library.theiet.org/content/conferences/10.1049/cp.2014.0699>

## BIBLIOGRAPHY

---

- [53] CoinMarketCap. All Cryptocurrencies. [online], [Accessed 15 December 2018]. Available from: <https://coinmarketcap.com/all/views/all/>
- [54] Wiki, L. Comparison between Litecoin and Bitcoin — Litecoin Wiki. [online], 2018, [Accessed 8 January 2019]. Available from: [https://litecoin.info/index.php?title=Comparison\\_between\\_Litecoin\\_and\\_Bitcoin&oldid=18](https://litecoin.info/index.php?title=Comparison_between_Litecoin_and_Bitcoin&oldid=18)
- [55] Jason Fernando. Bitcoin Vs. Litecoin: What's The Difference? *Investopedia*, [Accessed 20 December 2018]. Available from: <https://www.investopedia.com/articles/investing/042015/bitcoin-vs-litecoin-whats-difference.asp>
- [56] Harm, J.; Obregon, J.; et al. Ethereum vs. Bitcoin. *Creighton University, undated manuscript, retrieved*, volume 1, 2017. Available from: [https://www.economist.com/sites/default/files/creighton\\_university\\_kraken\\_case\\_study.pdf](https://www.economist.com/sites/default/files/creighton_university_kraken_case_study.pdf)
- [57] Oscar Williams-Grut. WALKTHROUGH: How traders 'pump and dump' cryptocurrencies. *Business Insider*, November 2017, [Accessed 13 December 2018]. Available from: <https://www.businessinsider.com/how-traders-pump-and-dump-cryptocurrencies-2017-11/>
- [58] Wikipedia contributors. Pump and dump — Wikipedia, The Free Encyclopedia. [online], 2018, [Accessed 8 January 2019]. Available from: [https://en.wikipedia.org/w/index.php?title=Pump\\_and\\_dump&oldid=866774359](https://en.wikipedia.org/w/index.php?title=Pump_and_dump&oldid=866774359)
- [59] Donna Rivera. 'Pump And Dump' Hits Cryptocurrency Market. *Investopedia*, [Accessed 21 December 2018]. Available from: <https://www.investopedia.com/news/pump-and-dump-hits-cryptocurrency-market/>
- [60] CoinMarketCap. Top 100 Cryptocurrencies by Market Capitalization. [online], [Accessed 30 November 2018]. Available from: <https://coinmarketcap.com/>
- [61] Statista. The 100 largest companies in the world by market value in 2018 (in billion U.S. dollars). [online], May 2018, [Accessed 28 November 2018]. Available from: <https://www.statista.com/statistics/263264/top-companies-in-the-world-by-market-value/>
- [62] Janze, C. Are Cryptocurrencies Criminals Best Friends? Examining the Co-Evolution of Bitcoin and Darknet Markets. September 2017. Available from: <https://aisel.aisnet.org/amcis2017/InformationSystems/Presentations/2/>

- 
- [63] Ito, T.; Yamada, K.; et al. Free Lunch! Arbitrage Opportunities in the Foreign Exchange Markets. Technical report, National Bureau of Economic Research, November 2012. Available from: <https://www.nber.org/papers/w18541>
- [64] Gradojevic, N.; Gencay, R.; et al. Robust Prediction of Triangular Currency Arbitrage with Liquidity and Realized Risk Measures: A New Wavelet-Based Ultra-High-Frequency Analysis. *SSRN Electronic Journal*, February 2017. Available from: <https://ssrn.com/abstract=3018815>
- [65] Sarkissian, J. Spread, volatility, and volume relationship in financial markets and market making profit optimization. *arXiv preprint arXiv:1606.07381*, 2016. Available from: <https://arxiv.org/abs/1606.07381>
- [66] Payward, Inc. What are Maker and Taker fees? [online], [Accessed 10 December 2018]. Available from: <https://support.kraken.com/hc/en-us/articles/360000526126-What-are-Maker-and-Taker-fees->
- [67] Fielding, R. T.; Taylor, R. N. *Architectural Styles and the Design of Network-based Software Architectures*. Dissertation thesis, University of California, Irvine, USA, 2000. Available from: [https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding\\_dissertation.pdf](https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf)
- [68] Fette, I.; Melnikov, A. RFC 6455: The WebSocket Protocol. Technical report, Internet Engineering Task Force (IETF), December 2011. Available from: <https://tools.ietf.org/html/rfc6455>
- [69] BLOCKCHAIN TRANSPARENCY INSTITUTE. December 2018: Rankings. [online], December 2018, [Accessed 25 December 2018]. Available from: <https://www.blockchaintransparency.org/december-2018-rankings>
- [70] Coinbase. Locations and trading pairs. [online], 2018, [Accessed 19 December 2018]. Available from: <https://support.pro.coinbase.com/customer/en/portal/articles/2945308-locations-and-trading-pairs>
- [71] Coinbase. Fee Structure. [online], 2018, [Accessed 15 December 2018]. Available from: <https://pro.coinbase.com/fees>
- [72] Coinbase. Coinbase Pro trader and developer documentation. [online], [Accessed 15 December 2018]. Available from: <https://docs.pro.coinbase.com/>

## BIBLIOGRAPHY

---

- [73] Confirno Ltd. Tradeview. [online], [Accessed 16 December 2018]. Available from: <https://coinmate.io/trade>
- [74] Confirno Ltd. Fees. [online], [Accessed 15 December 2018]. Available from: <https://coinmate.io/fees>
- [75] Confirno Ltd. CoinMate.io Trading Portal API. [online], [Accessed 15 December 2018]. Available from: <https://coinmate.docs.apiary.io/>
- [76] Pusher Ltd. Channels. [online], 2019, [Accessed 15 December 2018]. Available from: <https://pusher.com/docs>
- [77] BITTREX, INC. Markets. [online], 2019, [Accessed 15 December 2018]. Available from: <https://international.bittrex.com/home/markets>
- [78] Bittrex Support. BITTREX SERVICE FEES AND WITHDRAWAL LIMITATIONS. [online], December 2017, [Accessed 15 December 2018]. Available from: <https://bittrex.zendesk.com/hc/en-us/articles/115003684371>
- [79] Bittrex Support. Developer's Guide - API. [online], January 2018, [Accessed 15 December 2018]. Available from: <https://bittrex.zendesk.com/hc/en-us/articles/115003723911>
- [80] Bittrex Developers. API Reference. [online], [Accessed 15 December 2018]. Available from: <https://bittrex.github.io/api/v1-1>
- [81] Kashmir Hill. The Bitcoin Economy's 'Backbone' Is Bitstamp, An Exchange Run By Two Young Slovenians. *Forbes*, June 2014, [Accessed 16 December 2018]. Available from: <https://www.forbes.com/sites/kashmirhill/2014/06/26/bitcoin-bitstamp>
- [82] Bitstamp Ltd. Tradeview. [online], 2019, [Accessed 16 December 2018]. Available from: <https://www.bitstamp.net/market/tradeview/>
- [83] Bitstamp Ltd. UNIFIED FEE SCHEDULE. [online], 2019, [Accessed 16 December 2018]. Available from: [https://www.bitstamp.net/fee\\_schedule/](https://www.bitstamp.net/fee_schedule/)
- [84] Bitstamp Ltd. API. [online], 2019, [Accessed 16 December 2018]. Available from: <https://www.bitstamp.net/api/>
- [85] Clare Baldwin. Bitcoin worth \$72 million stolen from Bitfinex exchange in Hong Kong. *Reuters*, August 2016, [Accessed 13 December 2018]. Available from: <https://www.reuters.com/article/us-bitfinex-hacked-hongkong/bitcoin-worth-72-million-stolen-from-bitfinex-exchange-in-hong-kong-idUSKCN10EOKP>



- 
- [86] iFinex Inc. Statistics. [online], [Accessed 17 December 2018]. Available from: <https://www.bitfinex.com/stats>
- [87] iFinex Inc. Fees Schedule. [online], [Accessed 17 December 2018]. Available from: <https://www.bitfinex.com/fees>
- [88] iFinex Inc. The Bitfinex APIs. [online], [Accessed 17 December 2018]. Available from: <https://docs.bitfinex.com/docs>
- [89] Yuji Nakamura. Cryptocurrencies: World’s Biggest Cryptocurrency Exchange Is Heading to Malta. *Bloomberg*, March 2018, [Accessed 17 December 2018]. Available from: <https://www.bloomberg.com/news/articles/2018-03-23/the-world-s-biggest-cryptocurrency-exchange-is-moving-to-malta>
- [90] Finex.cz. Binance recenze – návod, poplatky. [online]. [in Czech language], October 2018, [Accessed 17 December 2018]. Available from: <https://finex.cz/recenze/binance/>
- [91] Binance Holdings Ltd. Binance Exchange: Whitepaper. [online], 2017. Available from: [https://www.binance.com/resources/ico/Binance\\_WhitePaper\\_en.pdf](https://www.binance.com/resources/ico/Binance_WhitePaper_en.pdf)
- [92] Binance.com. Binance Blog: Binance Monthly Review — Month 17. [online], 2018, [Accessed 16 December 2018]. Available from: <https://www.binance.com/en/blog/281368813841768448>
- [93] Binance.com. Deposit Fiat currency. [online], [Accessed 18 December 2018]. Available from: <https://support.binance.com/hc/en-us/articles/115003783671-Deposit-Fiat-currency>
- [94] Jon Russell. Crypto giant Binance looks to the future with fiat trading and a decentralized exchange. *TechCrunch*, 2018, [Accessed 17 December 2018]. Available from: <https://techcrunch.com/2018/09/25/binance-looks-to-the-future/>
- [95] Binance.com. Fee Schedule. [online], [Accessed 18 December 2018]. Available from: <https://www.binance.com/en/fee/schedule>
- [96] Binance. Official Documentation for the Binance APIs and Streams. *GitHub repository*, 2018, [Accessed 18 December 2018]. Available from: <https://github.com/binance-exchange/binance-official-api-docs>
- [97] Payward, Inc. We are a team: LEARN A LITTLE ABOUT US. [online], [Accessed 18 December 2018]. Available from: <https://www.kraken.com/en-us/about>

## BIBLIOGRAPHY

---

- [98] Jamie Holmes. Kraken: An Overview of One of Europe’s Top Bitcoin Exchanges. *Bitcoin Magazine*, September 2018, [Accessed 18 December 2018]. Available from: <https://bitcoinmagazine.com/articles/kraken-overview-one-europes-top-bitcoin-exchanges/>
- [99] Payward, Inc. Fiat currencies available on Kraken. [online], [Accessed 16 December 2018]. Available from: <https://support.kraken.com/hc/en-us/articles/360000382363>
- [100] Payward, Inc. Currency pairs available for trading on Kraken. [online], [Accessed 16 December 2018]. Available from: <https://support.kraken.com/hc/en-us/articles/201893658>
- [101] Payward, Inc. FEE SCHEDULE. [online], [Accessed 18 December 2018]. Available from: <https://www.kraken.com/help/fees>
- [102] Payward, Inc. API: Learn about our API. [online], [Accessed 18 December 2018]. Available from: <https://www.kraken.com/en-us/help/api>
- [103] Payward, Inc. Does Kraken offer a Live Streaming API / WebSocket? [online], [Accessed 16 December 2018]. Available from: <https://support.kraken.com/hc/en-us/articles/360000906063-Does-Kraken-offer-a-Live-Streaming-API-WebSocket->
- [104] Payward, Inc. What is the API call rate limit? [online], [Accessed 16 December 2018]. Available from: <https://support.kraken.com/hc/en-us/articles/206548367-What-is-the-API-call-rate-limit->
- [105] Payward, Inc. Kraken WebSockets Public API Version 0.1.1. [online], [Accessed 5 March 2019]. Available from: <https://www.kraken.com/features/websocket-api>
- [106] Zheping Huang. World’s third largest cryptocurrency exchange Huobi creates Communist Party committee. *South China Morning Post*, November 2018, [Accessed 26 December 2018]. Available from: <https://www.scmp.com/tech/start-ups/article/2173875/worlds-third-largest-cryptocurrency-exchange-huobi-creates-communist>
- [107] BLOCKCHAIN TRANSPARENCY INSTITUTE. December 2018: Exchange Volumes Report. [online], December 2018, [Accessed 25 December 2018]. Available from: <https://www.blockchaintransparency.org/>
- [108] Huobi Global. Exchange: Markets. [online], [Accessed 17 December 2018]. Available from: <https://www.hbg.com/en-us/markets/>
- [109] Huobi Global. Fee: Trading Fee. [online], [Accessed 19 December 2018]. Available from: <https://www.hbg.com/en-us/about/fee/>

- 
- [110] Huobi Global. Huobi-API. *GitHub repository*, [Accessed 18 December 2018]. Available from: [https://github.com/huobiapi/API\\_Docs\\_en/wiki](https://github.com/huobiapi/API_Docs_en/wiki)
- [111] Bloomberg L.P. Capital Markets: Company Overview of OKEX Technology Company Limited. [online], 2018, [Accessed 17 December 2018]. Available from: <https://www.bloomberg.com/research/stocks/private/snapshot.asp?privcapId=547129182>
- [112] OKEx.com. OKEx Accelerates Expansion to Malta. [online], 2018, [Accessed 17 December 2018]. Available from: <https://support.okex.com/hc/en-us/articles/360006882871-OKEx-Accelerates-Expansion-to-Malta>
- [113] OKEx.com. Charts. [online], [Accessed 17 December 2018]. Available from: <https://www.okex.com/marketList>
- [114] OKEx.com. OTC Trading. [online], [Accessed 17 December 2018]. Available from: <https://www.okex.com/otcTrade>
- [115] OKEx.com. Trading Fees. [online], [Accessed 19 December 2018]. Available from: <https://www.okex.com/pages/products/fees.html>
- [116] OKEx.com. OKEx developer documentation. [online], [Accessed 19 December 2018]. Available from: <https://www.okex.com/docs/en/>
- [117] OKEx. API docs OKEx. *GitHub repository*, 2018, [Accessed 19 December 2018]. Available from: <https://github.com/okcoin-okex/API-docs-OKEx.com/tree/master/API-For-Spot-EN>
- [118] DigiFinex Support. Introduction of DigiFinex exchange. [online], [Accessed 18 December 2018]. Available from: <https://digifinex.zendesk.com/hc/en-us/articles/360009411033-Introduction-of-DigiFinex-exchange>
- [119] DIGIFINEX LIMITED. DigiFinex Launch Maker & Taker Transaction Fee Structure with Different VIP Level. [online], November 2018, [Accessed 20 December 2018]. Available from: <https://www.digifinex.com/notice/en-ww/71>
- [120] DigiFinex. DigiFinex API Documentation. *GitHub repository*, 2018, [Accessed 20 December 2018]. Available from: [https://github.com/DigiFinex/api/blob/master/api\\_doc\\_en.md](https://github.com/DigiFinex/api/blob/master/api_doc_en.md)
- [121] Hamilton, J. Blackbird Bitcoin Arbitrage: a long/short market-neutral strategy. *GitHub repository*, [Accessed 20 February 2019]. Available from: <https://github.com/butor/blackbird>

## BIBLIOGRAPHY

---

- [122] PeanutPower. CryptoArbitrageTrader. *GitHub repository*, [Accessed 20 February 2019]. Available from: <https://github.com/PeanutPower/CryptoArbitrageTrader>
- [123] Jeff Vernon. Triangle Arbitrage Bot. [online], [Accessed 3 March 2019]. Available from: <https://help.haasonline.com/docs/v/2.x/custom-bots/triangle-arbitrage-bot>
- [124] Siebler, T. Triangular Arbitrage - Binance. *GitHub repository*, [Accessed 20 February 2019]. Available from: <https://github.com/tiagosiebler/TriangularArbitrage>
- [125] FX Empire Editorial Board. How to Read Foreign Exchange Rates? [online], [Accessed 24 April 2019]. Available from: <https://www.fxempire.com/education/article/how-to-read-foreign-exchange-rates-548324>
- [126] Muyao Shen. CoinMarketCap Says Data Glitch Boosted Its Crypto Price Numbers. *CoinDesk*, August 2018, [Accessed 1 January 2019]. Available from: <https://www.coindesk.com/coinmarketcap-says-data-glitch-boosted-its-crypto-price-numbers>
- [127] Marco Paez. OVER \$6 BILLION IN DAILY TRADING VOLUME FAKED ACROSS TOP 100 EXCHANGES. *Bitcoinist*, August 2018, [Accessed 1 January 2019]. Available from: <https://bitcoinist.com/6-billion-volume-faked-coinmarketcap/>
- [128] Carylyne Chan. CoinMarketCap stands for data transparency and clarity. [online], July 2018, [Accessed 28 December 2018]. Available from: <https://blog.coinmarketcap.com/2018/07/19/coinmarketcap-stands-for-data-transparency-and-clarity/>
- [129] Aibly Realtime. Long Polling – Concepts and Considerations. [online], [Accessed 28 April 2019]. Available from: <https://www.aibly.io/concepts/long-polling>
- [130] Hickson, I. Server-Sent Events. W3C recommendation, W3C, Feb. 2015, <http://www.w3.org/TR/2015/REC-eventsource-20150203/>.
- [131] Bharathvaj Ganesan. Polling vs SSE vs WebSocket – How to choose the right one. [online], [Accessed 28 April 2019]. Available from: <https://codeburst.io/polling-vs-sse-vs-websocket-how-to-choose-the-right-one-1859e4e13bd9>
- [132] PubNub Inc. What is WebSocket? [online], [Accessed 28 April 2019]. Available from: <https://www.pubnub.com/learn/glossary/what-is-websocket/>

- [133] Cryptonews.com team. The Beginners Guide to Bittrex: Complete Review 2019. [online], [Accessed 3 May 2019]. Available from: <https://cryptonews.com/reviews/bittrex/>
- [134] Adolph Obasogie. Bittrex vs Poloniex: Which is a Better Digital Currency Exchange? [online], [Accessed 29 December 2018]. Available from: <https://www.coinreview.com/bittrex-vs-poloniex/>
- [135] Pečev, A. Cryptocurrency Triangular Arbitrage Opportunities. *GitHub repository*, [Accessed 7 May 2019]. Available from: <https://github.com/adameen/triangular-arbitrage-opportunities>



## Acronyms

<b>ADA</b>	Cardano
<b>API</b>	Application Programming Interface
<b>ASIC</b>	Application-Specific Integrated Circuit
<b>BCH</b>	Bitcoin Cash
<b>BNB</b>	Binance Coin
<b>BSV</b>	Bitcoin SV
<b>BTC</b>	Bitcoin
<b>CPU</b>	Central Processing Unit
<b>CSS</b>	Cascading Style Sheets
<b>CZK</b>	Czech Koruna
<b>DFT</b>	DigiFinex Token
<b>EOS</b>	EOS (cryptocurrency)
<b>ETC</b>	Ethereum Classic
<b>ETH</b>	Ethereum
<b>EUR</b>	Euro
<b>EXP</b>	Expanse
<b>FIX</b>	Financial Information eXchange
<b>GDAX</b>	Global Digital Asset Exchange
<b>GUI</b>	Graphical User Interface

## A. ACRONYMS

---

**HT** Huobi Token

**HTML** Hypertext Markup Language

**HTTP** Hypertext Transfer Protocol

**HUSD** Huobi Stable Coin

**IP** Internet Protocol

**JPY** Japanese Yen

**JSON** JavaScript Object Notation

**LTC** Litecoin

**MB** Megabyte

**ODM** Object-Document Mapping

**OKB** OKEx Utility Token

**ORM** Object-Relational Mapping

**OS** Operating System

**OSI** Open Systems Interconnection

**OTC** Over-The-Counter

**RDBMS** Relational Database Management System

**REST** Representational State Transfer

**SCSS** Sassy Cascading Style Sheets

**SHA** Secure Hashing Algorithm

**SSE** Server-Sent Events

**SV** Satoshi Vision

**TCP** Transmission Control Protocol

**TUSD** TrueUSD

**UI** User Interface

**URL** Uniform Resource Locator

**USB** Universal Serial Bus

**USD** United States Dollar



---

**USDT** Tether

**W3C** World Wide Web Consortium

**WWW** World Wide Web

**XLM** Stellar

**XMR** Monero

**XRP** Ripple

**XZC** Zcoin

**ZRX** 0x (protocol)



## GUI Snapshots

All the GUI snapshots are to be found on the next pages in order to be visualised in their full sizes.

# Realtime triangular arbitrage opportunities



Figure B.1: GUI snapshot of the *Real-time data page*

## Historical triangular arbitrage data

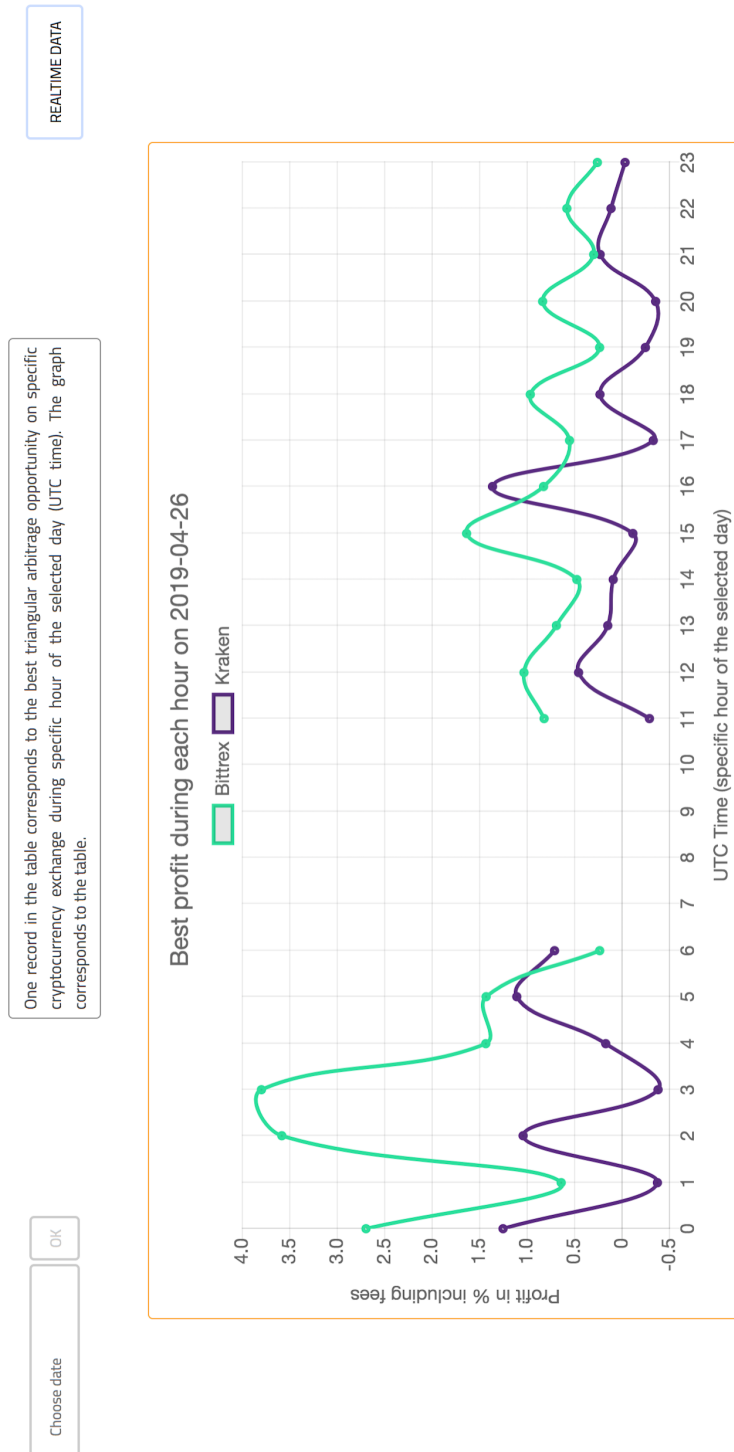


Figure B.2: GUI snapshot of the *Historical data page* with the chart

## Historical triangular arbitrage data

Choose date

OK

REALTIME DATA

One record in the table corresponds to the best triangular arbitrage opportunity on specific cryptocurrency exchange during specific hour of the selected day (UTC time). The graph corresponds to the table.

Time	Profit (%) incl. fees	Triangle	Exchange	Pair #1	Pair #2	Pair #3
2019-04-26 00:44:57 UTC	2.6921	GNO-ETH-BTC	Bittrex	ETH-GNO	BTC-ETH	BTC-GNO
2019-04-26 00:43:05 UTC	1.2483	XTZ-XETH-ZUSD	Kraken	XTZ-XETH	XETH-ZUSD	XTZ-ZUSD
2019-04-26 01:20:49 UTC	0.6955	SOLVE-ETH-BTC	Bittrex	ETH-SOLVE	BTC-ETH	BTC-SOLVE
2019-04-26 01:25:39 UTC	-0.378	XLTC-ZEUR-XXBT	Kraken	XLTC-ZEUR	XXBT-ZEUR	XLTC-XXBT
2019-04-26 02:10:09 UTC	3.5792	RVN-USDT-BTC	Bittrex	USDT-RVN	USDT-BTC	BTC-RVN
2019-04-26 02:48:59 UTC	1.037	XTZ-ZUSD-XETH	Kraken	XTZ-ZUSD	XETH-ZUSD	XTZ-XETH
2019-04-26 03:45:15 UTC	3.7937	RVN-USDT-BTC	Bittrex	USDT-RVN	USDT-BTC	BTC-RVN
2019-04-26 03:50:41 UTC	-0.3847	XXBT-XETH-ZCAD	Kraken	XETH-XXBT	XETH-ZCAD	XXBT-ZCAD
2019-04-26 04:50:56 UTC	1.4302	TRX-USD-USDT	Bittrex	USD-TRX	USD-USDT	USD-TRX
2019-04-26 04:08:11 UTC	0.1665	XTZ-XXBT-ZEUR	Kraken	XTZ-XXBT	XXBT-ZEUR	XTZ-ZEUR
2019-04-26 05:56:27 UTC	1.4265	ENJ-USDT-BTC	Bittrex	USDT-ENJ	USDT-BTC	BTC-ENJ

Figure B.3: GUI snapshot of the *Historical data page* with the table

---

## User's Manual

Since the application is open-source and the subject of future work, the installation of the environment is described step by step on a Unix-based system. In order to install the implemented Ruby on Rails application with all the functionality and in development environment, follow these instructions:

1. Download (or fork) the project from the public repository:  
`https://github.com/adameen/triangular-arbitrage-opportunities`
2. Unzip the downloaded project and navigate to the project directory:  
`cd triangular-arbitrage-opportunities-master`
3. Install Ruby version 2.6.1. The detailed description is here:  
`https://www.ruby-lang.org/en/documentation/installation/`
4. Install Ruby on Rails framework version 5.2.3. The detailed description is here:  
`https://gorails.com/setup/`
5. Install Bundler. The detailed description is here:  
`https://bundler.io/`
6. Install the Gemfile dependencies: `bundle install`
7. Install a JavaScript run-time library in case you do not have it installed already. E.g. Node.js may be installed – the detailed description is here:  
`https://nodejs.org/en/download/`
8. Run pending Migrations:  
`bin/rails db:migrate RAILS_ENV=development`

In order to run the server, follow these instructions:

1. Launch the server: `bin/rails server`
2. Connect (e.g. via web browser) to: `localhost:3000`





---

## Contents of enclosed CD

```
readme.txt ..... the file with CD contents description
| thesis_assignment.pdf ..... the thesis assignment in PDF format
| text ..... the thesis text directory
| | DP_Pecev_Adam_2019.pdf ..... the thesis text in PDF format
| | src ..... the thesis source files directory
| | | DP_Pecev_Adam_2019.zip ..... the thesis source files
| | | application.zip ..... the application source files
```