



UNIVERSITÀ DEGLI STUDI DI TRENTO

Department of Information Engineering And Computer Science

Bachelor degree in  
Computer Science

FINAL DISSERTATION

TITOLO

*Sottotitolo (alcune volte lungo - opzionale)*

Supervisor  
Agostinelli Claudio

Candidate/Student  
Vasile Adrian Bogdan Pop

Accademic Year 2018/2019

# Ringraziamenti

*...thanks to...*

# Contents

<b>Abstract</b>	<b>2</b>
<b>1 Blockchain</b>	<b>3</b>
1.1 Distributed ledger . . . . .	3
1.2 Consensus Algorithms . . . . .	3
1.2.1 Proof of Work . . . . .	4
1.2.2 Proof of Stake . . . . .	4
1.2.3 Proof of Burn . . . . .	4
1.3 Advantages and Disadvantages . . . . .	4
1.3.1 Advantages . . . . .	4
1.3.2 Disadvantages . . . . .	5
1.4 Use Cases . . . . .	6
1.4.1 Digital Identity . . . . .	6
1.4.2 Charity . . . . .	6
1.4.3 Healthcare . . . . .	6
1.4.4 Governance . . . . .	6
1.4.5 Payments . . . . .	6
<b>2 Bitcoin</b>	<b>7</b>
2.1 Infrastructure . . . . .	7
2.1.1 Incentive . . . . .	7
2.2 Transactions . . . . .	7
2.3 Wallets . . . . .	9
<b>3 Bitcoin Cash</b>	<b>9</b>
3.1 Fork . . . . .	9
3.1.1 SegWit . . . . .	10
3.1.2 Addresses . . . . .	10
3.1.3 Block mining . . . . .	10
<b>4 Package implementation in R</b>	<b>11</b>
4.1 R . . . . .	11
4.2 Feasibility study . . . . .	11
4.3 Package integration . . . . .	11
4.3.1 Commands . . . . .	13
4.4 Usage samples . . . . .	14
<b>Bibliografia</b>	<b>14</b>

# Abstract

The thesis tries to explain what a blockchain is, how it works and why it seems to have so much potential. In particular, this debate focuses on the different purposes of this technology with its functionalities and its implementations by analyzing two specific blockchains: Bitcoin (BTC) and Bitcoin Cash (BCH). The thesis also exposes which are the requirements a blockchain should have when it is used as a coin, in order to be globally handled as the leading currency. Hence, it goes deeper into the differences between the BTC and BCH analyzing the distinct approaches to satisfy the demands of a cryptocurrency. Especially, it illustrates the main reasons why the Bitcoin hard fork was reached on August 1st, 2017, giving life to the Bitcoin Cash blockchain. Once the behavior of this new technology is clear, the thesis goes further with the integration of BCH in the programming language "R". In this way, it provides a mean to work with the data stored into the Bitcoin Cash blockchain and retrieve interesting statistics out of it.

In other words, the steps to achieve the goal for this thesis will be:

- Understanding and analyzing blockchain technologies
- Studying the two most important Bitcoin blockchains BTC - Bitcoin and BCH - Bitcoin Cash
- Design a new Package in R for data reading and manipulation of the Bitcoin Cash blockchain
- Implementation of the Package in R with a description of all the possible commands developed
- Sample of the Package functionalities and its potential

# 1 Blockchain

The application of the blockchain in multiple fields is gaining more and more traction today, until to be considered by some as the new Internet. But actually, the main concept behind this revolutionary idea was described as early as 1991 in an accademic paper<sup>1</sup> published by Stuart Haver and W. Scott Stornetta. With their solution, the two scientists aim to Time-Stamp documents, in order to certify when a document was created or last changed. The system used a cryptographically secured chain of blocks to store the time-stamped documents and in 1992 Merkle trees were incorporated to the design, making it more efficient by allowing several documents to be collected into one block.[1] However, this is not enough. Indeed, to have a fully working blockchain we need to add more components.

## 1.1 Distributed ledger

As said in a report from the UK Government, "A distributed ledger is essentially an asset database that can be shared across a network of multiple sites, geographies or institutions. All participants within a network can have their own identical copy of the ledger"[6]. Due to this technology, there aren't either central administrators or centralized data storages. To have this mechanism working, we need both a Peer-to-Peer Network<sup>2</sup> and a Consensus Algorithm 1.2.

## 1.2 Consensus Algorithms

To clarify the benefits of this type of algorithms we must start with a dilemma as known as "The Byzantine Generals Problem", in whom, each general has its own army and that each group is situated in different locations around the city they intend to attack. The generals need to agree on either attacking or retreating. It does not matter whether they attack or retreat, as long as all generals reach consensus, i.e., agree on a common decision in order to execute it in coordination. Therefore, we may consider the following requirements:

- Each general has to decide: attack or retreat (yes or no)
- After the decision is made, it cannot be changed
- All generals have to agree on the same decision and execute it in a synchronized manner

The aforementioned communication problems are related to the fact that one general is only able to communicate with another through messages, which are forwarded by a courier. Consequently, the central challenge of the Byzantine Generals' Problem is that the messages can get somehow delayed, destroyed or lost.

In addition, even if a message is successfully delivered, one or more generals may choose (for whatever reason) to act maliciously and send a fraudulent message to confuse the other generals, leading to a total failure.

If we apply the dilemma to the context of blockchains, each general represents a network node, and the nodes need to reach consensus on the current state of the system. Putting in another way, the majority of participants within a distributed network have to agree and execute the same action in order to avoid complete failure.[11][1] There are various implemetation of the mechanism through which a blockchain network reach consensus.

---

<sup>1</sup>[https://www.anf.es/pdf/Haber\\_Stornetta.pdf](https://www.anf.es/pdf/Haber_Stornetta.pdf)

<sup>2</sup>the peers are computer systems connected to each other via the Internet, the data can be shared directly between systems on the network.

### 1.2.1 Proof of Work

Due to Bitcoin 2, the most known is the Proof of Work (PoW) algorithm, in whom is utilized for block generation. The process of generating correct proofs in order to add a block to the blockchain is known as “mining” and the individuals that participate in the mining process are known as “miners”.[14]

The PoW mining involves numerous hashing attempts, so more computational power means more trials per second. In other words, miners with a high hash rate have better chances to find a valid solution for the next block. The PoW consensus algorithm makes sure that miners are only able to validate a new block and add it to the blockchain if the distributed nodes of the network reach consensus and agree that the block hash provided by the miner is a valid proof of work.[1]

### 1.2.2 Proof of Stake

The Proof of Stake (PoS) consensus algorithm is relatively different and a new way to generate and append blocks in a blockchain. It was developed in 2011 as alternative to PoW, which requirements a massive amount of energy to make it work. In fact, according to the bitcoin energy consumption tracker at Digiconomist<sup>3</sup>, bitcoin currently consumes 66.7 terawatt-hours per year. That’s comparable to the total energy consumption of the Czech Republic, a country of 10.6 million people.

Basically, the mining process is replaced with a mechanism where blocks are validated according to the stake of the participants, also known as “validators”. The criteria used to choose the validator depends on the proof of stake system but mainly, the choice is based on the economic stake in the network of each node.[1][14]

### 1.2.3 Proof of Burn

Also the Proof of Burn (PoB) algorithm is an approach to avoid the massive waste of energy used for hashing. Indeed, with this technique, the mining process is replaced with a greener one. Where, the “miners” of the PoB coins will send coins to an unspendable address, known as “eater address”. Even this transactions are recorded on the blockchain ensuring that the coin cannot be spent again. The principle behind this consensus algorithm is that the user burning the cryptocurrency is showing long-term commitment to the coin by burning it. This is because they are taking a short-term loss in exchange for a long-term gain.[14]

## 1.3 Advantages and Disadvantages

As a result for its complexity, blockchain’s potential as a distributed form of record-keeping is almost without limit. But as every other technology, even the blockchain have its pros and cons.

### 1.3.1 Advantages

**Trustless system and small fees.** In most traditional systems, a consumer pays a bank to verify a transaction, a notary to sign a documents, or a minister to perform a marriage. Using blockchain technologies, this is no longer necessary because the distributed nodes verify the transactions through consensus algorithms.

Therefore, blockchain systems negate the risk of trusting a single person or organization and reduces the overall costs and transaction fees by cutting third-parties. For this reason, blockchain is often referred to as a “trustless” system.[1][7]

**Improved accuracy by removing human involvement in verification.**

**Transactions are secure, private and efficient.** Also in this case, by eliminating the human factor, we obtain a 24 hours a day working system, instead of institutions operating only during business hours. As a result, transactions can be completed in about ten minutes and can be considered secure after just a few hours. Actually, once a transaction is recorded, its authenticity must be verified by the blockchain network. Where, millions of nodes rush to confirm that the details of the transaction are correct.

Moreover, every block of the chain has the hash of the block before it along with its own hash, which is computed with the data contained in the block. So, when an information in a block is edited in

---

<sup>3</sup><https://digiconomist.net/bitcoin-energy-consumption>

any way, its own hash will change, but the hash in the block after it would not. This contrast makes extremely hard to change data inside the blockchain without being noticed.

Considering that the majority of blockchains are public, it means that anyone can access the blockchain and read the data inside it with all the transaction history. So, if anyone can read see all the transactions, why is people say that blockchain transactions are anonymous?

That's why anonymity is confused with confidentiality. Indeed, every user makes public transactions with their unique code called "public key"<sup>4</sup>, which is recorded on the blockchain, rather than their personal information.[7]

**Transparency.** A big benefit to this technology is given by its open source nature. That means that the users of the blockchain network can modify the code as they see fit, so long as they have the approval from the majority of the network's nodes.

**Distributed.** Blockchain does not store any of its information in a unique node or central location. Instead, data is stored in thousands of devices on a distributed network. Whenever a block is added to the chain, every node updates its blockchain to reflect the change. Because of this, there is no single point of failure and so becomes more difficult to tamper with.[1][7]

**Stability.** Once a block is confirmed, it's extremely difficult to remove or change the data inside it. This makes blockchain a great technology for storing financial records or any other data where an audit trail is required because every change is tracked and permanently recorded on a distributed and public ledger.[1]

### 1.3.2 Disadvantages

While there are significant upsides to the blockchain, there are also serious challenges to its adoption. The most ones are political and regulatory, but there are also some technical challenges to take in account.

**Energy inefficiency.** Some blockchains, especially those using Proof of Work consensus algorithms 1.2.1, are highly inefficient. Since mining is highly competitive and there is only one winner every ten minutes, the work of every other miner is wasted. As miners are continually trying to increase their computational power, so they have a greater chance of finding a valid block hash, the resources used by the Bitcoin network has increased significantly in the last few years, and it currently consumes more energy than many countries, such as Denmark, Ireland, and Nigeria.[1]

**Scalability.** The Bitcoin is a perfect case study for the low scalability of blockchains. In fact, to add a new block on the blockchain, it takes about ten minutes and the maximum size of each block is of 2 MegaBytes. This entails that there is a limited number of transaction per block. It is estimated that the Bitcoin's blockchain can manage only seven transactions per second (TPS). Meanwhile, the Visa's actual infrastructure can process 24,000 TPS.

**51% Attacks.** The consensus algorithms that protect the blockchain has proven to be very efficient over the years (thanks to ). However, there are a few potential attacks that can be performed against blockchain networks. The most discussed is the 51% attack. Such an attack may happen if one entity manages to control more than 50% of the network, which would allow them to edit or modify the order of the transactions. Although being theoretically possible, there was never a successful 51% attack on the Bitcoin blockchain, additionally, as the network gets larger, the security increases.[1]

**Private keys.** Blockchain uses public-key cryptography to give users ownership over their blockchain data. Each public-key corresponds to a private-key which should be kept secret. Users need their private-key to access their data. If a user loses their private key, the money is effectively lost, and there is nothing they can do about it.[1]

---

<sup>4</sup>Public-key cryptography, is a cryptographic system that uses pairs of keys: public keys which may be disseminated widely, and private keys which are known only to the owner.

**Storage.** Distributed ledgers can increase very much over time. For example the Bitcoin blockchain currently requires 239<sup>5</sup> GigaByte of storage. Even if in the Bitcoin paper its said that thanks to "Moore's Law" [12] the storage should not be a problem, the current growth in blockchain size appears to be outstripping the growth in hard drives and the network risks losing nodes if the ledger becomes too large for individuals to download and store.[1]

## 1.4 Use Cases

The merit to the great attention given to the blockchain technology is to be attributed to Bitcoin. Indeed, thanks to this digital currencies (cryptocurrencies) the blockchain-based solutions have grown. So learning how this innovative technology can be applied to different scenarios is very important.

### 1.4.1 Digital Identity

The identity component in a blockchain is fulfilled through the use of cryptographic keys. Combining a public and private key creates a strong digital identity reference based on possession of informations or certifications. The public key, is distributed publicly and it is what identifies the identity, the private key is kept secret by the identity and it is used to express consent to digital interactions and to sign the data to record in the block.[4]

### 1.4.2 Charity

The number of charitable organizations which adopted cryptocurrencies as a donation method is increasing more and more. The reason is because crypto's transparency, in fact, thanks to this characteristic, every donation can be tracked and verified. So, the higher level of transparency and public accountability can ease donors' minds and encourage them to give while also reinforcing the charity's reputation for integrity. Moreover, blockchain technology reduces the fees and taxes to manage every donation by reducing the number of required intermediaries.[1]

### 1.4.3 Healthcare

Blockchain may offer significant benefits to hospitals in terms of security, interoperability and transparency. Unlike traditional databases that rely on a centralized server, the use of a distributed system allows for data exchange with higher levels of security. In addition, distributed ledgers increase the interoperability among clinics, hospitals, and other health service providers by allowing all these parties to interface with a unique storage system.

Furthermore, blockchain systems may also give higher levels of accessibility and transparency over their own health information. But also, allows pharmaceutical organizations to increase the efficiency of their infrastructure by cutting down on the widespread problem of drug counterfeiting.[1]

### 1.4.4 Governance

The governance can be greatly improved in various sectors by the blockchain technologies. A short example can be the efficiency of taxes, indeed, because governments rely on taxpayer funds, it's important that they use their budgets wisely. Blockchain systems and smart contracts can be employed to automate tasks and workflows, which would greatly reduce time and money spent on bureaucratic processes.[1]

Another application can be the election process. With the support of a workign digital identity system, and the high level of immutability given by the blockchain, this technology could be an excellent solution for ensuring that votes cannot be tampered with. Creating the possibility to transform the secure online voting into a reality.

### 1.4.5 Payments

Thanks to Bitcoin, this is the most widespread use case of the blockchain. It starts in 2008 when Satoshi Nakamoto published the Bitcoin's white paper, where, for the first time, all the above components of a blockchain were used to solve the double-spending problem.

---

<sup>5</sup>Last update: 10/9/2019



## 2 Bitcoin

In a few words, Bitcoin is a digital form of the actual cash. But in contrast to the traditional fiat currency, which relies on a central bank controlling it, this new form of currency relies on blockchain technology<sup>1</sup> and cryptography<sup>1</sup>. As a result, this new digital cash is mainly called "cryptocurrency".

### 2.1 Infrastructure

Bitcoin, as the majority of cryptos, is based on a public permissionless blockchain architecture. Which means that anyone can join, read, write and commit, plus it is hosted on public servers, is anonymous and high resilient.

The Bitcoin's mechanism to reach consensus is the Proof of Work algorithm 1.2.1. Due to the implementation described by Satoshi Nakamoto in the white paper "The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash. For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it"[12]. It follows that the difficulty required to close a block is flexible. In fact, it changes every 2016 blocks suiting the computational power of the network. Considering that, at the desired rate of one block each 10 minutes, 2016 blocks mean that the difficulty is adjusted every two weeks.

#### 2.1.1 Incentive

To have a working PoW, miners have to spend their CPU time and electricity. So, why should they allocate their resources in Bitcoin?

The answer is: the incentive. Indeed, the first transaction of every block is a special transaction which gives a new amount of currency to the creator of the block. This, not only is a good incentive to support the network but also provides a way to initially distribute coins into circulation, since there is no central authority to issue them. However, this process is destined to end, because there is a predetermined number of coins (21 million) which can spread. To ensure that, every 210.000 blocks (corresponding to 4 years), the rewarding given to the creator of the block is halved. This leads to the second type of incentive in the Bitcoin network: transaction fees. In fact, for every transaction there is also a fee amount, which is given to the creator of the block as a reward for its effort.

### 2.2 Transactions

To figure out how a transaction works, before it is necessary to understand the concept owning a coin. Because, unlike the fiat currency, where you own a physical coin to spend, owning a bitcoin means that you own the key to unlock the coins from a past transaction and use them in the new transaction. To get a better sense of the transaction mechanism, it is recommended to read a brief extract of the Bitcoin paper:

"We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership"[12]

---

<sup>1</sup>In computer science, cryptography refers to secure information and communication techniques derived from mathematical concepts and a set of rule-based calculations called algorithms to transform messages in ways that are hard to decipher.

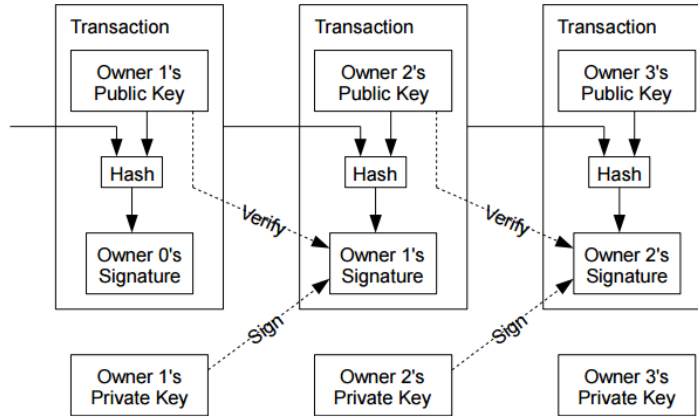


Figure 2.1: Transaction schema. From [12]

To go deeper into the design of a transaction, it is useful to imagine a double-entry ledger. In few words, every transaction has one or more "inputs"<sup>2</sup> and one or more "outputs"<sup>3</sup>. Usually outputs add up to slightly less than inputs, because the difference represents an implied "transaction fee" 2.1.1. Higher the fee, higher the probability for the transaction to be included in the next block to mine.

Each output, before being spent as an input in a new transaction, waits as an "Unspent Transaction Output" (UTXO). The sum of all the owned UTXOs, compose the total balance of a wallet 2.3. But, not every transaction in the blockchain produces an UTXO. Indeed, it depends on the scriptPubKey<sup>4</sup> used into the specific transaction.

**Pay To Public Key Hash (P2PKH)** Is the most common form of scriptPubKey used to send a transaction to one or multiple addresses. In this type of scriptPubKey the coins sent can be unlocked by a single private key corresponding to the public key hashed in the transaction itself. It is possible to recognize it by looking at the first character in the address used for the output. In fact, if it starts with the number "1" it is P2PKH.[2]

**Pay To Script Hash (P2SH)** Was implemented in 2012 to add more flexibility to transactions by adding a hash of a second script, the redeem script<sup>5</sup>. It can be used to process multisignature transactions, Open Assets Protocol<sup>6</sup> and to store textual data on the blockchain. The P2SH can be recognized by the number "3" at the beginning of the addresses.[2]

**Multisig** Special addresses where to spend a transaction, multiple private keys signature are required. Although P2SH multisig is now generally used for multisig transactions, this base script can be used to require multiple signatures before a UTXO can be spent. In multisig pubkey scripts, called m-of-n, m is the minimum number of signatures which must match a public key; n is the number of public keys being provided.[2]

**Null Data** This type adds arbitrary data to a probably unspendable scriptPubKey that nodes don't have to store in their UTXOs database. Consensus rules allow null data outputs up to the maximum allowed scriptPubKey size of 10,000 bytes provided.[2]

<sup>2</sup>debits against a bitcoin account.

<sup>3</sup>credits added to a bitcoin account.

<sup>4</sup>A script included in outputs which sets the conditions that must be fulfilled for those satoshis to be spent. Data for fulfilling the conditions can be provided in a signature script (another type of data generated by the spender).

<sup>5</sup>A script similar in function to a scriptPubKey. One copy of it is hashed to create a P2SH address (used in an actual scriptPubKey) and another copy is placed in the spending signature script to enforce its conditions.

<sup>6</sup>Used to create new currencies upon the Bitcoin blockchain. Called colored coins

## 2.3 Wallets

In simple words, a wallet is the mean through which is possible to manage all the owned cryptocurrency. But it can be splitted in two components, programs and files. Wallet programs create the public keys to receive the Bitcoin and uses the corresponding private key to spend them.

Wallet files, instead, stores the private keys in a file or physically on pieces of paper. Furthermore, it also gives the process used for the public-private key creation and usage, and an approach to a deterministic hierarchy key creation process in an unlinkable keys manner.

# 3 Bitcoin Cash

"Imagine you have a dollar in your wallet, and you put it aside for a while and then realise that one coin has split in two. Sounds weird? Not in the cryptocurrency world." That is an analogy with the fiat currency reported in an article of Shouth China Morning Post[10]. Bitcoin is like a software, but, due to its distributed nature, unlike all the software we know, there isn't a single entity who determines how it should be updated. As a result, to upgrade the blockchain it is necessary to be in agreement with the major part of the community. That's what happened on July 20th, 2017, when the 97% of the Bitcoin network voted to activate the "Segregated Witness" (SegWit) algorithm 3.1.1 to improve the scalability of Bitcoin. Although almost all the community agreed, some members believed that adopting SegWit without increasing the block size simply postpone the scalability problem of Bitcoin. As a result, on August 1st, 2017 there were two forks of the Bitcoin network. One for adopting SegWit and one incrementing the size of the blocks giving birth to the Bitcoin Cash blockchain.[10][9]

## 3.1 Fork

As said before, to have an upgrade in a blockchain we need the agreement of all the network, those updates are called "forks". Basically, there are two types of fork:

**Soft Fork** means that change in the blockchain protocol is backward-compatible. That means, that although some nodes of the network are outdated, they are still able to process transaction on the network, as long as they don't brake the new protocol rules. That's what happened with SegWit in August 2017.

**Hard Fork** is the opposite. An obsolete node cannot perform any transaction on the new protocol. According to the situation, hard fork can either be planned or controversial. In a planned fork, participants are going to upgrade their software voluntarily, leaving the old version behind. The non-update nodes, are going to mine on the old blockchain, which will be used by very few participants. In a controversial fork instead, there is a disagreement within the community about the upgrade, so, the old blockchain is splitted in two new incompatible ones.

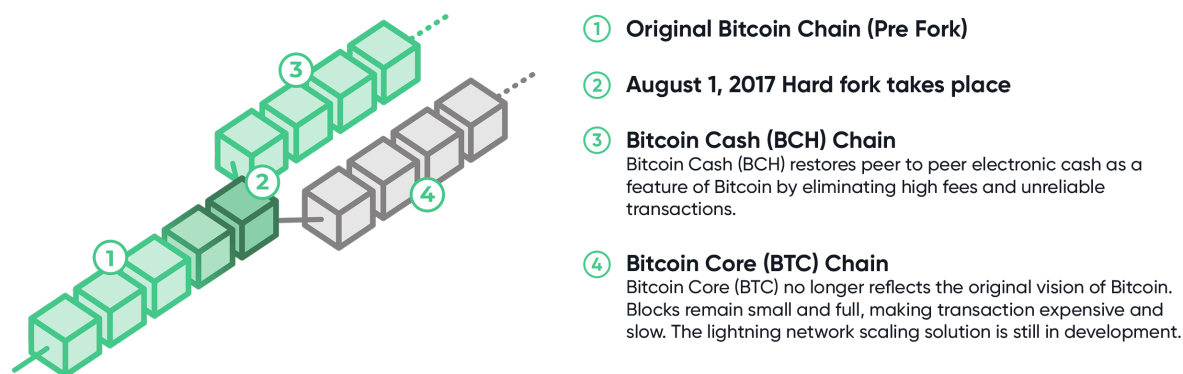


Figure 3.1: A simple schema of the controversial hard fork of Bitcoin Cash. From [3]

Both will have their own network, and as it happened to Bitcoin Cash on August 2017, will be developed in the way the participants believe is the best. Here, is where comes to help the analogy made at the beginning, because, all the coins owned before the fork, will be splitted. So, the same amount of coins as before will be owned in both the blockchains.[1]

### 3.1.1 SegWit

Segregated Witness is the major discrepancy between Bitcoin and Bitcoin Cash. In Fact, unlike BCH, where to increase the number of transactions per second (TPS) the block size was only incremented to 8 MegaByte, with SegWit the upgrade was more complicated. The transaction is splitted in two segments, the first, which is saved on the blockchain, contains the information about the sender and the receiver. Meanwhile, the second part containing the scripts and the signature remain at the bottom in a separate structure. As a result, the amount of information per transaction is less and makes possible to add more transaction in each block.

This is only one of the multiple benefits taken by the adoption of SegWit algorithm<sup>1</sup>.

### 3.1.2 Addresses

Due to the nature of Bitcoin Cash, its addresses are generated in exactly the same manner as the Bitcoin ones, so it is very difficult to recognize the source of a specific address. To facilitate its use and decrease the probability of being confused when someone tries to read it, was introduced the new Cashaddr Address Format. In few words, its only a new type of encoding which will display the address with a different pattern. In addition, every new Cashaddr corresponds to an old address, so if necessary, also the outdated addresses can be used.

### 3.1.3 Block mining

With the fork on August 1st, 2017, Bitcoin Cash not only changes the block size, but implements also the "Emergency Difficulty Adjustment" (EDA) algorithm. The motivation was to have mean through convince miners to expend their resources in BCH. In fact, at the beginning, even the price of Bitcoin Cash was significantly lower than the price of Bitcoin, due to the fork process, the difficulty for mining was the same. So, it was less convenience to mine in the BCH blockchain. EDA resolves this problem by reducing the difficulty for block number  $t$  by 20% if the time difference between the  $(t - 6)^{th}$  block and  $(t - 12)^{th}$  block was greater than 12 hours. This system was active from 1st August 2017, until 12th November 2017. Indeed, the next day, a hard fork was made to implement the new "Difficulty Adjustment Algorithm" (DAA), which seeks to accomplish the following objectives:

- Adjust difficulty to hash rate to target a mean block interval of 600 seconds.
- Avoid sudden changes in difficulty when hash rate is fairly stable.
- Adjust difficulty rapidly when hash rate changes rapidly.
- Avoid oscillations from feedback between hash rate and difficulty.
- Be resilient to attacks such as timestamp manipulation.

To fulfill all these requirements, the new algorithm adjusts the difficulty with each block, taking into account the amount of work done and the elapsed time of the previous 144 blocks.[5][13]

---

<sup>1</sup>To read about all the benefits of this algorithm read here: <https://bitcoincore.org/en/2016/01/26/segwit-benefits/>

## 4 Package implementation in R

Once understood what is a Bitcoin Cash, how works the technology behind it and the reasons why it was created. Its time to start and read some data from the blockchain in order to put into practice what we have learned in the above chapters. Moreover, after the data is imported, its a good practice to analyze the data in such a way as to get usefull information out of it. To do so, the R programming language will be used.

### 4.1 R

R is an Open Source software for statistical computing(linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering) and graphics. One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed.

R is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R which makes it easy for users to follow the algorithmic choices made. But, for computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Furthermore, thanks to its Open Source nature, it can be easily extended via "Packages" even written by the user himself.[8]

### 4.2 Feasibility study

The goal for this project is to extend the R language with a package which allow us to read and analyze the information into the blockchain Bitcoin Cash.

To make it in the best way, firstly was done a research of the existing implementations, unfortunately, whitout any result. But, it brings us two completely differnt ways to realize the package.

The first one was to turn the used device into a node of the Bitcoin Cash network. This lead user to download all the blockchain and make all the queries locally. As a result, the queries will be very fast and all the data of the blockchain can be manipulated as you desire. But, on the other hand, storing all the blockchain requires a lot of space on you device (around 143 GigaByte on 13/9/2019). So making a package of that size doesn't seem to be the best option.

For theese reasons, the second way was choosen. Although its slowest implementation, it avoids the need to become a node of the network, making the installation of the package very simple and fast. That was obtained by relying on a service provider (SP).

### 4.3 Package integration

Once decided the more convenient option to adopt, it was necessary to do another research phase. In fact, it is pretty difficult to find a SP which not only implemented an "Application Programming Interface" (API) where you can make queries an have the wanted data as a response. But also makes the public, so anyone can use it. Additionally, the service providers was choosen also according to the amount of data provided, the format of the response and the number of queries you are allowed to make. At the end, the picks were:

- Blockdozer Explorer
- Blockchain

The motivation for these two options, is the fact that they are complementary. The first one, allows an illimitate number of queries returning essential information. The second one instead, has a limit of 30 queries per minute, but returns more and better formatted data.

Let's start with the technical implementation by saying that to maintain all the R project, was used the RStudio editor.  
First of all, it is necessary to create and manage a new package in R. For this purpose, some other package are required :

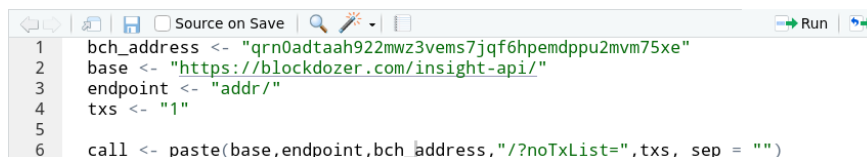
- "devtools" - To manage the project and the source code of the new package.
- "roxygenize2" - To handle all the documentation of the package.

Then its necessary to install other two package to handle the data and the connection with the service providers.

- "httr" - To perform correctly the http requests to the service providers and receive the linked response
- "jsonlite" - To manipulate and store the received data, in order to have a good dataset for the future analysis

Now we are ready to perform a simple query to an API. Let's say that we want to know some information about a specific Bitcoin Cash address. After choosing the appropriate SP, we have to read their documentation and see how to correctly build the http request. In this case the pattern is:

*GET /api/addr/{paymentAddress}[?noTxList&from&to&returnLegacyAddresses]*



```

1 bch_address <- "qrn0adtaah922mwz3vems7jqf6hpemdppu2mvm75xe"
2 base <- "https://blockdozer.com/insight-api/"
3 endpoint <- "addr/"
4 txs <- "1"
5
6 call <- paste(base,endpoint,bch_address,"/?noTxList=",txs, sep = "")

```

Figure 4.1: Example of the building of a http request

With a few lines code, we can than construct a possible request for our information. As a result we can have:

*https://blockdozer.com/insight-api/addr/qrn0adtaah922mwz3vems7jqf6hpemdppu2mvm75xe/?noTxList=1*

As said before, the response will be in a json format and will be handled thanks to the jsonlite package. But due to the multiple ways to manipulate the data, its suggested to go on the GitHub repository<sup>1</sup> and take a look to the source code.



```

addrStr: "qrn0adtaah922mwz3vems7jqf6hpemdppu2mvm75xe"
balance: 0
balanceSat: 0
totalReceived: 0.34323506
totalReceivedSat: 34323506
totalSent: 0.34323506
totalSentSat: 34323506
unconfirmedBalance: 0
unconfirmedBalanceSat: 0
unconfirmedTxApperances: 0
txApperances: 9504

```

Figure 4.2: Example json response

<sup>1</sup>[github.com/PopBogdan97/bCashReader](https://github.com/PopBogdan97/bCashReader)

### 4.3.1 Commands

Due to the multiple data formats provided by the Service Providers and the different information that can be retrieved from the Bitcoin Cash blockchain, we can subdivide all the implemented commands in four macro categories.

#### Address Commands

Function	Description
addr_balance(hash string)	Get the balance of a Bitcoin Cash address.
addr_history(hash string)	Get the general data (history) of a Bitcoin Cash address.
addr_totalReceived(hash string)	Get the total amount of Bitcoin Cash received by the address.
addr_totalSent(hash string)	Get the total amount of Bitcoin Cash sent by the address.
addr_txApperances(hash string)	Get the total amount of transactions made by the address.
addr_txs(hash string)	Get the all the transactions made by the address.
addr_unconfirmedBalance(hash string)	Get the unconfirmed balance of a Bitcoin Cash address.
addr_unconfirmedTransactions(hash string)	Get the total amount of unconfirmed transactions made by the address.
addr_utxo(hash string)	Get the unspent transactions (UTXO) of an address.

#### Block Commands

Function	Description
block_byDate(hash string)	Get all the blocks in a specific Date with the format yyyy-MM-dd.
block_byDateCount("yyyy-MM-dd" string)	Get the block count in a specific Date with the format yyyy-MM-dd.
block_byHash(hash string)	Get the information of a specific Bitcoin Cash block (without the transactions).
block_byHashRaw(hash string)	Get the raw information of a specific Bitcoin Cash block.
block_byHashTxS(hash string)	Get the transaction hashes of a specific Bitcoin Cash block.
block_byHeight(height integer)	Get the informations contained in a specific Bitcoin Cash block.

## Transaction Commands

Function	Description
txs_byHash(hash string)	Get the all the information of a specific transaction <sup>2</sup> .
txs_byHashIn("yyyy-MM-dd" string)	Get the all the information of a specific transaction inputs <sup>2</sup> .
txs_byHashOut(hash string)	Get the all the information of a specific transaction outputs <sup>2</sup> .

## Chain Commands

Function	Description
chain_currency()	Get the Bitcoin Cash current value..
chain_stats()	Get the Bitcoin Cash current blockchain stats <sup>3</sup> .

## 4.4 Usage samples

repository

---

<sup>2</sup>To see the description of the values retrieved visit: [https://github.com/Blockchair/Blockchair.Support/blob/master/API\\_DOCUMENTATION\\_EN.md#link\\_chainstats](https://github.com/Blockchair/Blockchair.Support/blob/master/API_DOCUMENTATION_EN.md#link_chainstats) in Dashboard/transaction

<sup>3</sup>To see the description of the values retrieved visit: [https://github.com/Blockchair/Blockchair.Support/blob/master/API\\_DOCUMENTATION\\_EN.md#link\\_chainstats](https://github.com/Blockchair/Blockchair.Support/blob/master/API_DOCUMENTATION_EN.md#link_chainstats) in Dashboard/stats.



# Bibliography

- [1] Binance Academy. [binance.vision/blockchain](https://binance.vision/blockchain).
- [2] bitcoin. [bitcoin.org/en](https://bitcoin.org/en).
- [3] Bitcoin.com. [bitcoin.com](https://bitcoin.com).
- [4] Coindesk. [coindesk.com/information](https://coindesk.com/information).
- [5] Bitcoin ABC. Difficulty adjustment algorithm update. *bitcoinabc.org*, November 2017.
- [6] UK Government Office for Science. Distributed ledger technology: beyond block chain. *gov.uk/government/science*, 2016.
- [7] Luke Fortney. Blockchain explained. *Investopedia*, June 2019.
- [8] The R Foundation. What is r? *r-project.org*, 2019.
- [9] Matteo Gatti. Bitcoin cash (bch): a brief analysis of the bitcoin fork. *The Cryptonomist*, July 2019.
- [10] Zheping Huang. Bitcoin cash “hard fork”: everything you need to know about the latest cryptocurrency civil war. *Shouth China Morning Post*, November 2018.
- [11] Marshall Pease Leslie Lamport, Robert Shostak. *Distributed Systems: concepts and Design*. ACM New York, NY, USA, 1982.
- [12] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *bitcoin.org*, 2008.
- [13] Jimmy Song. Bitcoin cash difficulty adjustments. *medium.com*, August 2017.
- [14] CoinBundle Team. Consensus algorithms. *Medium*, September 2018.