Vrije Universiteit Amsterdam

Research Paper

# What is blockchain? How is it going to affect Business?

*Luc Severeijns*

supervised by
Prof.Dr. Sandjai Bhulai

November 6, 2017

# Executive Summary

This research looks at blockchain technology and especially blockchain in the business sector. First, an introduction to blockchain is given to understand the technology, and then the business possibilities are discussed. The first section is dedicated to describing the technology itself step-by-step. The second part is describing which blockchains exist, and how to distinguish these. The last part contains examples of use cases of blockchain. In this research, blockchain is dismantled and categorized for business.

# Contents

# Preface

This paper is written for newbies in blockchain who would like to understand how blockchain works, IT-specialists who want to go further in depth and decision makers who would like to know more about the possibilities of blockchain. I would like to thank my supervisor S. Bhulai for guiding me during this research.

## 1 Introduction

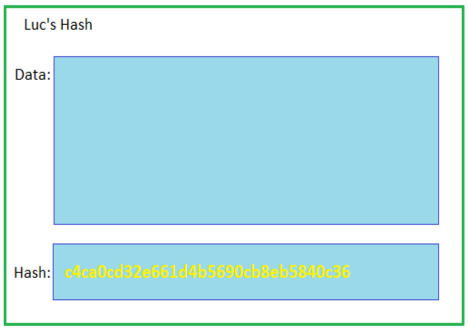
The word blockchain is used very often these days. Public society is excited about this new technology, and they are not wrong to be. It is essential to have a good understanding of blockchain and its features as it is going to change the business. There has been a lot of research on the security of blockchain, see, e.g., the papers [1][2]. This paper is to inform about the business side to blockchain. Blockchain will be explained, different types of blockchain will be discussed and use cases. A small new topic is introduced: perishability. Blockchain is a technology that stores value but what happens when we store a perishable product like music rights. These rights perish after some time and can the blockchain technology handle this loss in value?

# 2 What is blockchain?

A blockchain stores value, for instance, money. A famous example of a money-blockchain is Bitcoin. Most people put their money in the bank nowadays because they trust the bank with their savings. Although most banks have a small chance of bankruptcy, there is still a possibility every day that they can tell their clients that they do not have their money anymore. What the bank does is storing how much money there is in a bank account, this is exactly what a blockchain does. So why do we still take this risk, why do we not cut out the middle man and bear no risk? This sounds like a perfect scenario, but before one would be willing to do that, he/she will want to understand the blockchain technology, which I will address below.

## 2.1 Hashes

To understand blockchain first we will need to understand what a hash is. To get a better understanding, illustrations will be used. Let us look at a hash:

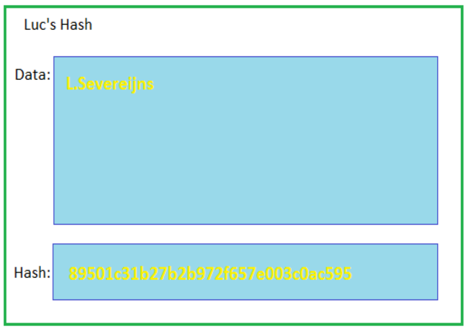


Luc's Hash

Data:

Hash: c4ca0cd32e661d4b5690cb8eb5840c36

(a) Empty databox



Luc's Hash

Data: L.Severeijns

Hash: 89501c31b27b2b972f657e003c0ac595

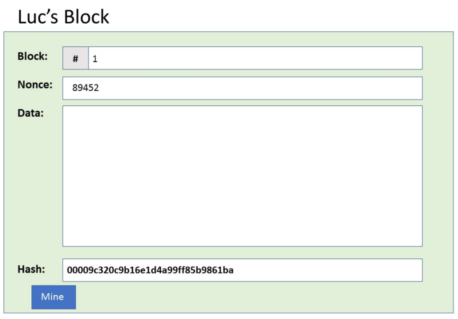
(b) databox with name

In Figure (a) and (b) we can see a hash, a hash looks like a series of random characters, but it represents a digital fingerprint of some digital data. For now, it is the digital fingerprint of what is typed in the Databox. Whenever the input in the Databox is changed, the hash will change as well. As we can see in Figure (b) the hash of "L.Severeijns" is: 89501c31b27b2b972f657e003c0ac595. If we erase and retype "L.Severeijns" it will be the same hash. So the hash: 89501c31b27b2b972f657e003c0ac595 is a digital fingerprint of the name L.Severeijns. There are different generators of hashes; for this paper, I used the MD5 hash. [4]

The databox is unlimited and can be filled with anything, however, the hash will always have the same length of characters. It is impossible to guess the hash of a specific input due to the randomness. We are going to extend this idea of a hash into something called a "block".

## 2.2 Blocks

A block is like a hash, but the databox is split into three parts: Block, Nonce, and data.



Luc's Block

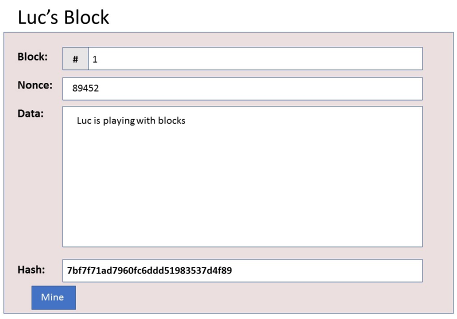
Block: # 1

Nonce: 89452

Data:

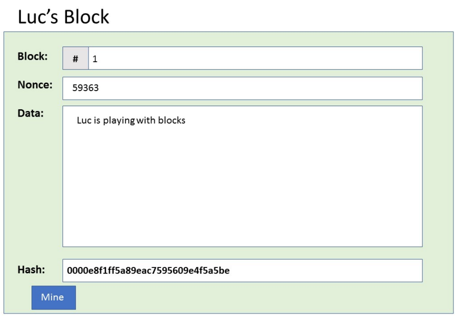
Hash: 00009c320c9b16e1d4a99ff85b9861ba

Block indicates the number of a block, in Figure 2 it is block 1. The nonce will be explained further on. The databox is similar to the databox of the hash. The hash of this block is different than the previous one although the databox is empty. This is due to that Luc's block is slightly different than the Luc's hash before, and therefore they cannot have the same fingerprint/hash. The hash of this block starts with four zeros: <u>0000</u>9c320c9b16e1d4a99ff85b9861ba. This is a relatively unusual hash as most hashes will not start with four zeros. Because this block does start with four zeros, we are going to call it "signed", every block starting with <u>0000</u> is signed, this is chosen arbitrarily. Signed means valid, so a block needs to start with 0000 to be valid. If we change Block, nonce or Data the hash is going to change. The probability that the block will be signed (hash starting with 0000) is very small. This is illustrated below:


Luc's Block

Block: # 1
Nonce: 89452
Data: Luc is playing with blocks
Hash: 7bf7f71ad7960fc6ddd51983537d4f89

Data and hash are changed, since the hash does not start with four zeros this block is not signed. This is the part where the nonce plays an important role. By changing the nonce, the hash will change as well. The idea of changing the nonce is to try to change a non-signed block into a signed block. When the Mine button in Luc's Block is pressed, the nonce is going to loop through all numbers until it hashes out to a block that starts with 0000. A hash that starts with 0000 is always possible to create if we find a correct nonce by looping through all numbers. In the first case at the beginning of the Blocks section, the nonce was 89452. 89452 is not the only number which hashes out to a signed block. There are many more, but this number came up the first. This process is called mining. When we mine the last block (invalid block):

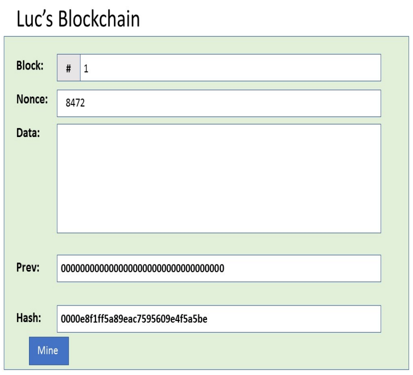

Luc's Block

Block: # 1
Nonce: 59363
Data: Luc is playing with blocks
Hash: 0000e8f1ff5a89eac7595609e4f5a5be

The nonce is changed to 59363, and the hash starts with 0000 again. The result is that the block is signed. So if this is a block, a blockchain will be a chain of blocks.

## 2.3 Blockchains

A blockchain is a chain of blocks; these blocks need to be linked for the blockchain to work. A linked blockchain is shown below:
Block 1 contains the same elements as before, but now there is an extra feature: Prev, previous. The Prev of block 1 is all zeros. When we look at block 2 and 3 we see a different Prev. The Prev of block 3 is the hash of block 2 and the Prev of block 2 is the hash of block 1. Prev is the hash of the previous block. Prev is needed so that each block points back to the block before it. At the first block, the Prev is some fake number as the first block does not point back to another block. Let us look at changing the data in the blocks. When we change the data in block 3, the hash


Luc’s Blockchain

Block: # 1

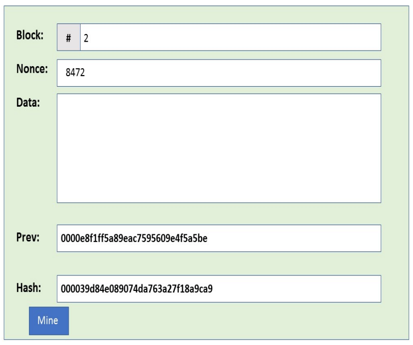
Nonce: 8472

Data:

Prev: 00000000000000000000000000000000

Hash: 0000e8f1ff5a89eac7595609e4f5a5be

(a) Block1


Block: # 2

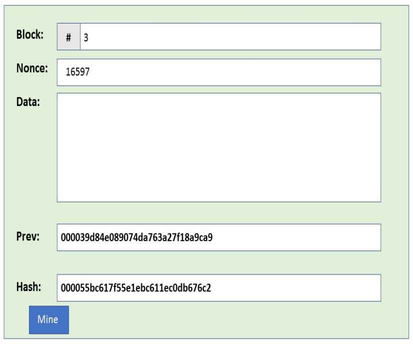
Nonce: 8472

Data:

Prev: 0000e8f1ff5a89eac7595609e4f5a5be

Hash: 000039d84e089074da763a27f18a9ca9

(b) Block2


Block: # 3

Nonce: 16597

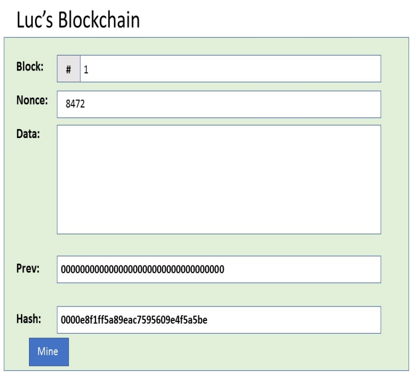
Data:

Prev: 000039d84e089074da763a27f18a9ca9

Hash: 000055bc617f55e1ebc611ec0db676c2

(c) Block 3

will change, and the block will be invalidated. On the next page is shown what happens when we change the data in block 2 in Luc’s Blockchain:


Luc’s Blockchain

Block: # 1

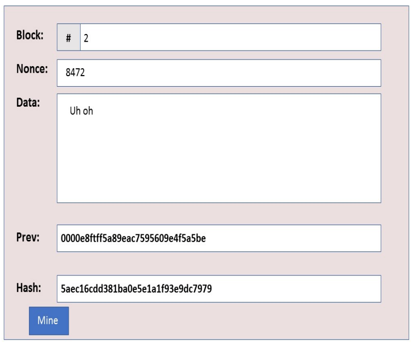
Nonce: 8472

Data:

Prev: 000000000000000000000000000000000

Hash: 0000e8f1ff5a89eac7595609e4f5a5be

(a) Block1


Block: # 2

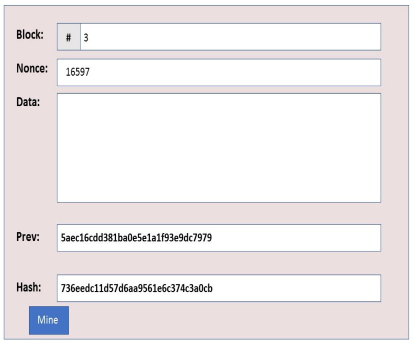
Nonce: 8472

Data: Uh oh

Prev: 0000e8ftff5a89eac7595609e4f5a5be

Hash: 5aec16cdd381ba0e5e1a1f93e9dc7979

(b) Block2


Block: # 3

Nonce: 16597

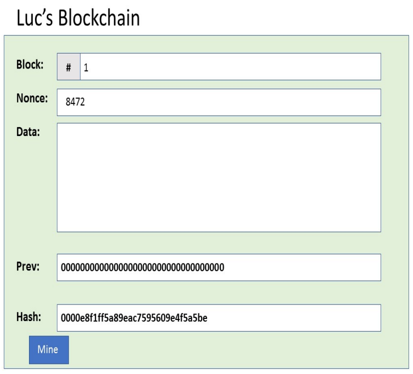
Data:

Prev: 5aec16cdd381ba0e5e1a1f93e9dc7979

Hash: 736eedc11d57d6aa9561e6c374c3a0cb

(c) Block 3

The hash of block 2 is changed by changing the data in block 2. This hash gets copied up to block 3 which changes the hash of block 3. Block 2 and 3 both become invalid and so it breaks both blocks when changing the data of block 2. We can go back as far as we want in the past blocks and break a block, all the blocks since that block will be broken as well. If we want to change a previous block permanently in the blockchain, we will need to remine it. If we remine block 2 at this moment, block 2 will hash out but block 3 will still be broken.


Luc's Blockchain

Block: # 1

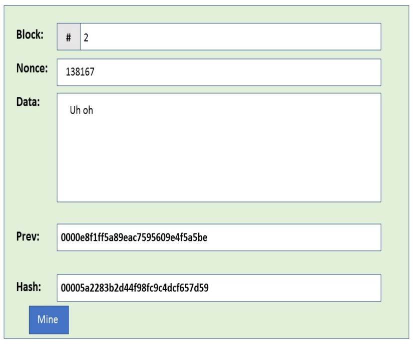
Nonce: 8472

Data:

Prev: 00000000000000000000000000000000

Hash: 0000e8f1ff5a89eac7595609e4f5a5be

(a) Block1


Block: # 2

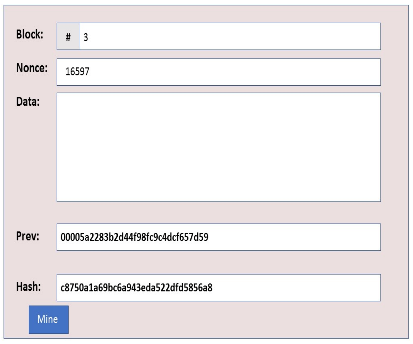
Nonce: 138167

Data: Uh oh

Prev: 0000e8f1ff5a89eac7595609e4f5a5be

Hash: 00005a2283b2d44f98fc9c4dcf657d59

(b) Block2


Block: # 3

Nonce: 16597

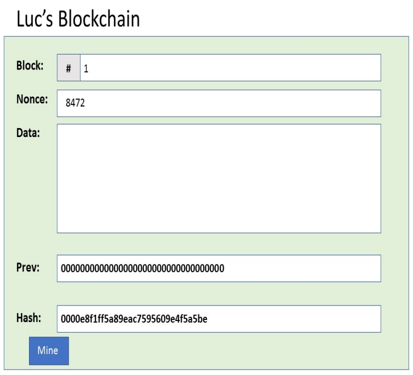
Data:

Prev: 00005a2283b2d44f98fc9c4dcf657d59

Hash: c8750a1a69bc6a943eda522dfd5856a8

(c) Block 3

The Prev of block 3 changed and so the hash does not start with 0000. If we want the blockchain to be valid, we need to remine block 3 as well. By changing the last block of the blockchain, we only need to remine the last block, changing any other block in the blockchain results into remining that block and all the blocks after the chosen block. The further we go back in the blockchain the harder it is to make a change. This is how a blockchain is resisting change. We will remine block 3 to make it valid.


Luc's Blockchain

Block: # 1

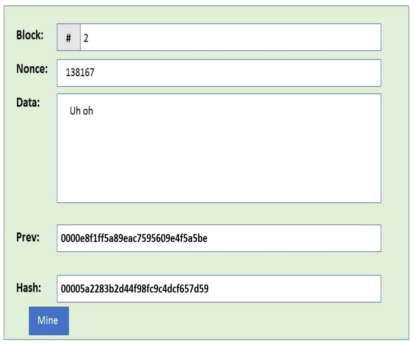
Nonce: 8472

Data:

Prev: 00000000000000000000000000000000

Hash: 0000e8f1ff5a89eac7595609e4f5a5be

(a) Block1


Block: # 2

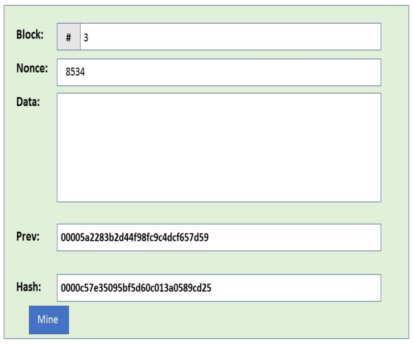
Nonce: 138167

Data: Uh oh

Prev: 0000e8f1ff5a89eac7595609e4f5a5be

Hash: 00005a2283b2d44f98fc9c4dcf657d59

(b) Block2


Block: # 3

Nonce: 8534

Data:

Prev: 00005a2283b2d44f98fc9c4dcf657d59

Hash: 0000c57e35095bf5d60c013a0589cd25

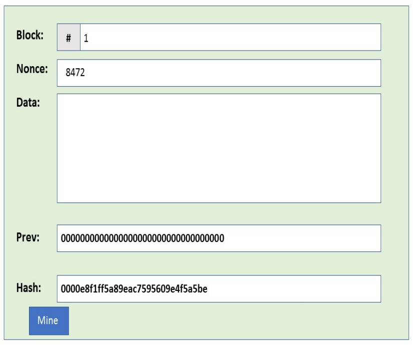
(c) Block 3

The blockchain is now valid, but there is a difference in data between this blockchain and the blockchain at the beginning of Section 3.3. How do we know that the blockchain has been remined?

## 2.4 Distributed blockchain

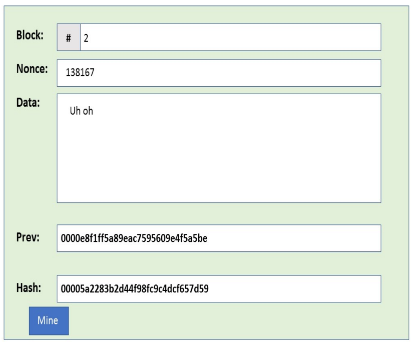
Distributed blockchain looks exactly like the previous blockchain but now next to Luc's Blockchain we will have two other blockchains, and we will call it Don's Blockchain and Milou's Blockchain. Both blockchains are identical to Luc's Blockchain, so we have three identical blockchains:

Luc’s Blockchain



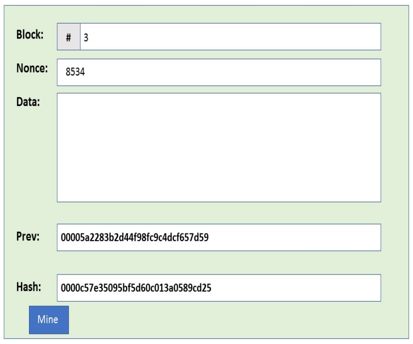
Block: # 1
Nonce: 8472
Data:
Prev: 0000000000000000000000000000000000
Hash: 0000e8f1ff5a89eac7595609e4f5a5be

(a) Block1 Luc



Block: # 2
Nonce: 138167
Data: Uh oh
Prev: 0000e8f1ff5a89eac7595609e4f5a5be
Hash: 00005a2283b2d44f98fc9c4dcf657d59

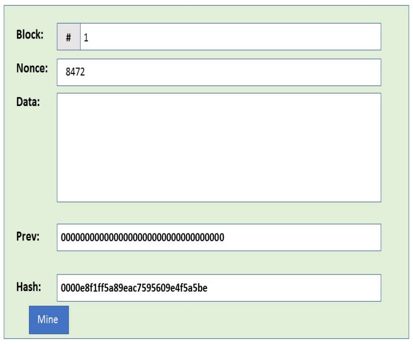
(b) Block2 Luc



Block: # 3
Nonce: 8534
Data:
Prev: 00005a2283b2d44f98fc9c4dcf657d59
Hash: 0000c57e35095bf5d60c013a0589cd25

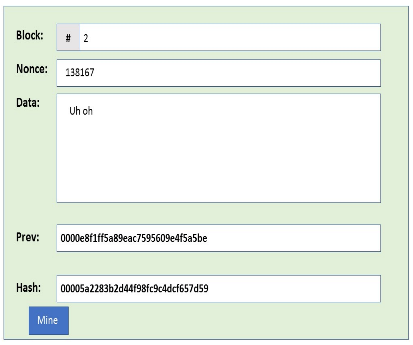
(c) Block 3 Luc

Luc’s Blockchain



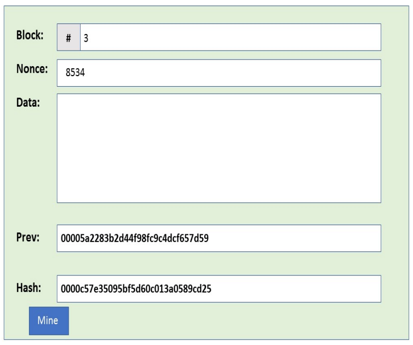
Block: # 1
Nonce: 8472
Data:
Prev: 0000000000000000000000000000000000
Hash: 0000e8f1ff5a89eac7595609e4f5a5be

(d) Block1 Don



Block: # 2
Nonce: 138167
Data: Uh oh
Prev: 0000e8f1ff5a89eac7595609e4f5a5be
Hash: 00005a2283b2d44f98fc9c4dcf657d59

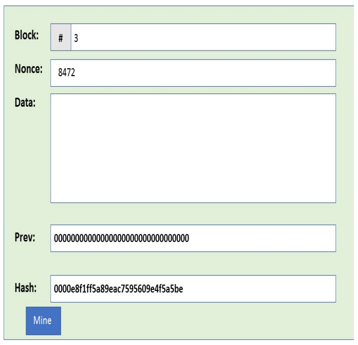
(e) Block2 Don



Block: # 3
Nonce: 8534
Data:
Prev: 00005a2283b2d44f98fc9c4dcf657d59
Hash: 0000c57e35095bf5d60c013a0589cd25

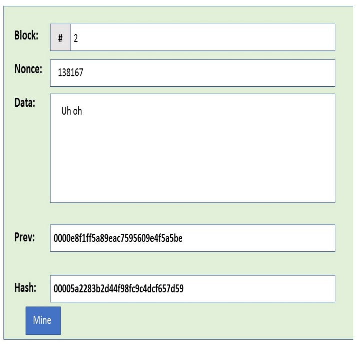
(f) Block 3 Don

Milou’s Blockchain



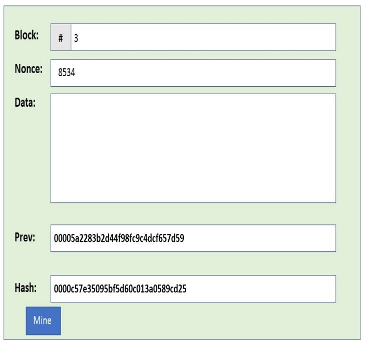
Block: # 3
Nonce: 8472
Data:
Prev: 00000000000000000000000000000000
Hash: 0000e8f1ff5a89eac7595609e4f5a5be

(g) Block1 Milou



Block: # 2
Nonce: 138167
Data: Uh oh
Prev: 0000e8f1ff5a89eac7595609e4f5a5be
Hash: 00005a2283b2d44f98fc9c4dcf657d59

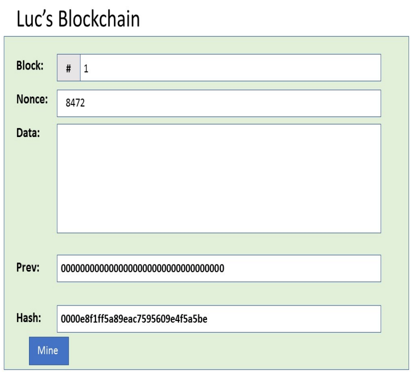
(h) Block2 Milou



Block: # 3
Nonce: 8534
Data:
Prev: 00005a2283b2d44f98fc9c4dcf657d59
Hash: 0000c57e35095bf5d60c013a0589cd25

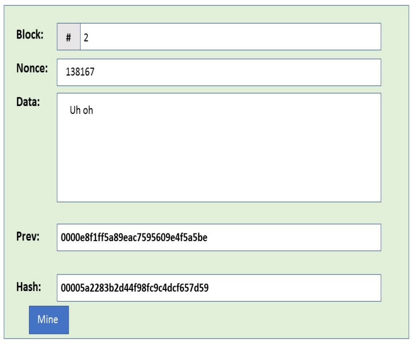
(i) Block3 Milou

We are going to change the data of Don's Blockchain and then remine it again. All three the blockchains will be valid as illustrated on the next page:

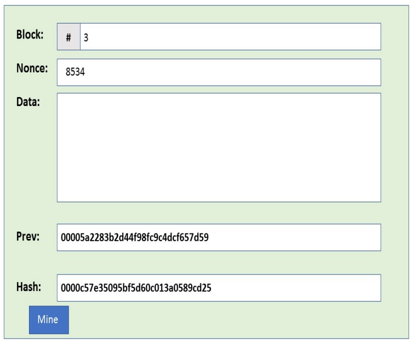

Luc's Blockchain

Block: # 1
Nonce: 8472
Data:
Prev: 0000000000000000000000000000000000
Hash: 0000e8f1ff5a89eac7595609e4f5a5be

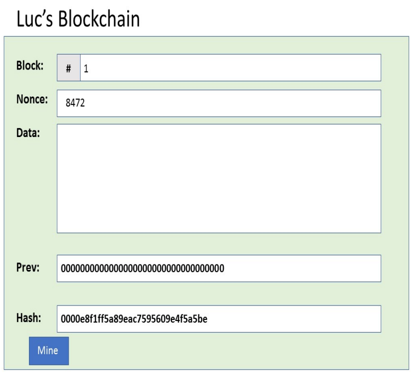
(a) Block1 Luc


Block: # 2
Nonce: 138167
Data: Uh oh
Prev: 0000e8f1ff5a89eac7595609e4f5a5be
Hash: 00005a2283b2d44f98fc9c4dcf657d59

(b) Block2 Luc

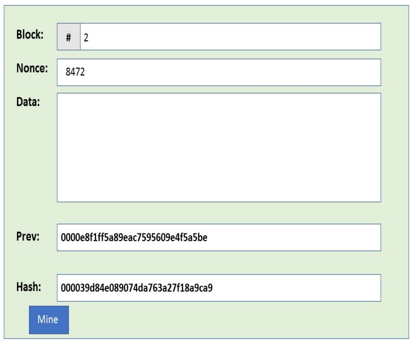

Block: # 3
Nonce: 8534
Data:
Prev: 00005a2283b2d44f98fc9c4dcf657d59
Hash: 0000c57e35095bf5d60c013a0589cd25

(c) Block 3 Luc

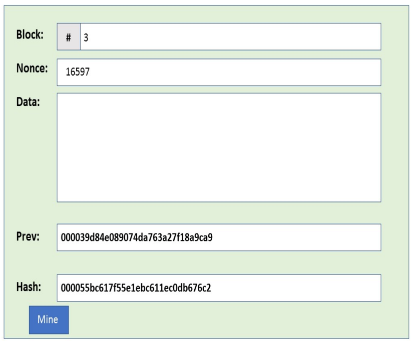

Luc's Blockchain

Block: # 1
Nonce: 8472
Data:
Prev: 0000000000000000000000000000000000
Hash: 0000e8f1ff5a89eac7595609e4f5a5be

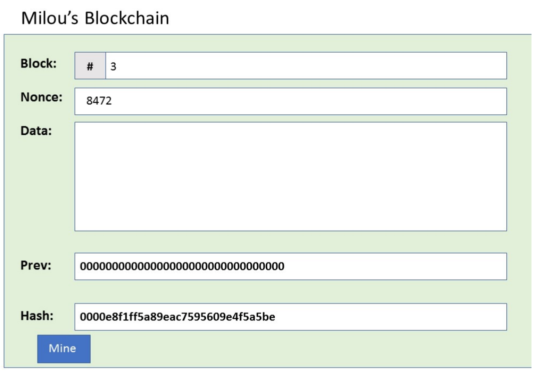
(d) Block1 Don


Block: # 2
Nonce: 8472
Data:
Prev: 0000e8f1ff5a89eac7595609e4f5a5be
Hash: 000039d84e089074da763a27f18a9ca9

(e) Block2 Don


Block: # 3
Nonce: 16597
Data:
Prev: 000039d84e089074da763a27f18a9ca9
Hash: 000055bc617f55e1ebc611ec0db676c2

(f) Block 3 Don



Milou’s Blockchain

Block: # 3

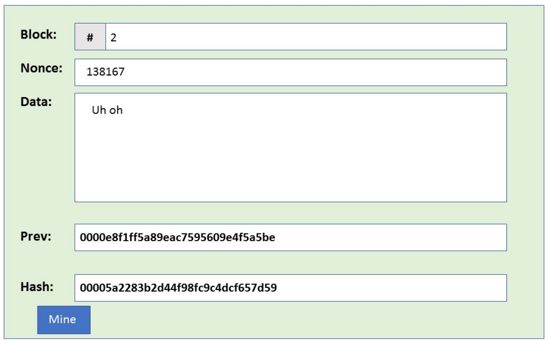
Nonce: 8472

Data:

Prev: 00000000000000000000000000000000

Hash: 0000e8f1ff5a89eac7595609e4f5a5be

(a) Block1 Milou



Block: # 2

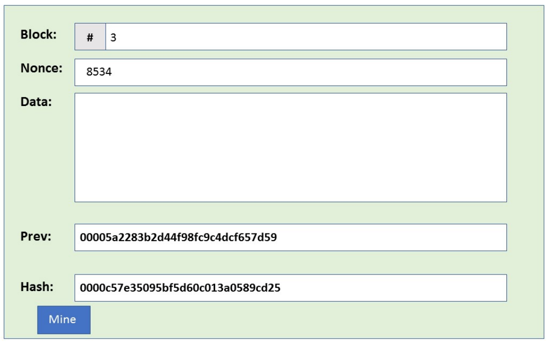
Nonce: 138167

Data: Uh oh

Prev: 0000e8f1ff5a89eac7595609e4f5a5be

Hash: 00005a2283b2d44f98fc9c4dcf657d59

(b) Block2 Milou



Block: # 3

Nonce: 8534

Data:

Prev: 00005a2283b2d44f98fc9c4dcf657d59

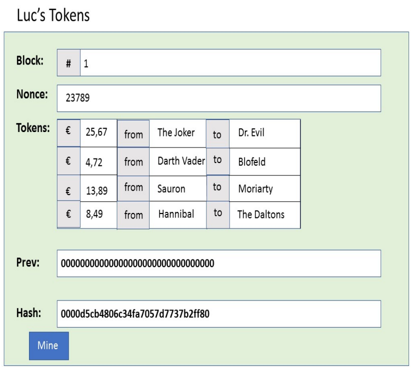
Hash: 0000c57e35095bf5d60c013a0589cd25

(c) Block3 Milou

All three blockchains are valid, so how do we know which blockchain has been changed. Therefore we need to take a look at the last hashes of the blockchain. Luc and Milou’s blockchain both have the same last hash but Don’s last hash is different. By inspection, we see that Don’s blockchain has been changed. Blockchain works a little like a democracy in this case two of the three blockchains (Luc and Milou) say: **000057c35095bf5d60c013a0589cd25** and only one blockchain (Don) says: **000055bc617f55e1ebc611ec0db676c2**. Two votes are more than one, so Don’s blockchain is wrong. In this example, we used three distributed blockchains, in other blockchains there will be much more "peers" than three who check the blockchain which makes the blockchain verification a lot more reliable. No matter how many blocks there are in a blockchain, the peers only need to check the last hash to see if a blockchain is altered.

This is all there is to the blockchain technology, but in these examples blockchain is not useful. So far what was typed in the databox is irrelevant information, what we want in the databox are tokens. In the next subsections, I will discuss relevant blockchains.

## 2.5 Tokens


Luc's Tokens

Block: # 1

Nonce: 23789

Tokens:

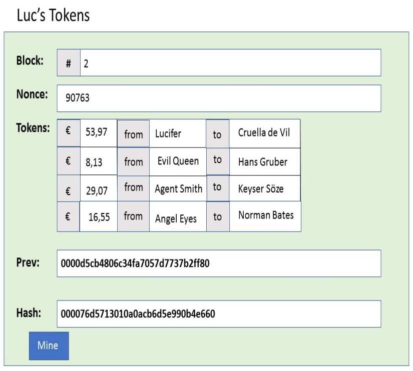
| | | | | | |
|---|---|---|---|---|---|
| € | 25,67 | from | The Joker | to | Dr. Evil |
| € | 4,72 | from | Darth Vader | to | Blofeld |
| € | 13,89 | from | Sauron | to | Moriarty |
| € | 8,49 | from | Hannibal | to | The Daltons |

Prev: 0000000000000000000000000000000

Hash: 0000d5cb4806c34fa7057d7737b2ff80

Mine

(a) Block1


Luc's Tokens

Block: # 2

Nonce: 90763

Tokens:

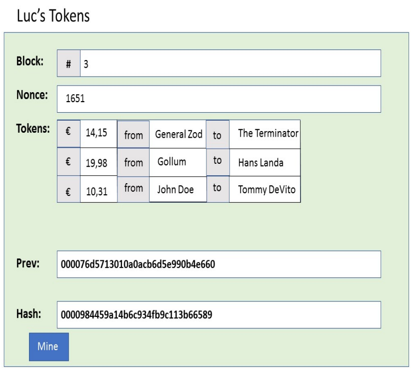
| | | | | | |
|---|---|---|---|---|---|
| € | 53,97 | from | Lucifer | to | Cruella de Vil |
| € | 8,13 | from | Evil Queen | to | Hans Gruber |
| € | 29,07 | from | Agent Smith | to | Keyser Söze |
| € | 16,55 | from | Angel Eyes | to | Norman Bates |

Prev: 0000d5cb4806c34fa7057d7737b2ff80

Hash: 000076d5713010a0acb6d5e990b4e660

Mine

(b) Block2


Luc's Tokens

Block: # 3

Nonce: 1651

Tokens:

| | | | | | |
|---|---|---|---|---|---|
| € | 14,15 | from | General Zod | to | The Terminator |
| € | 19,98 | from | Gollum | to | Hans Landa |
| € | 10,31 | from | John Doe | to | Tommy DeVito |

Prev: 000076d5713010a0acb6d5e990b4e660

Hash: 0000984459a14b6c934fb9c113b66589

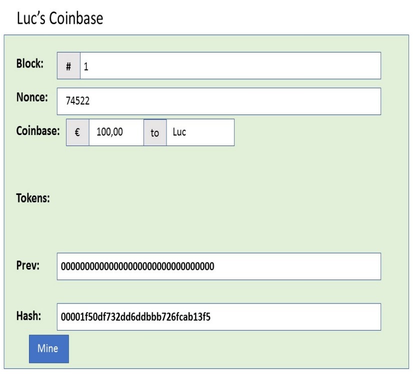
Mine

(c) Block 3

Instead of typing some strings in the databox, there are now transactions. What was called data is now called tokens, and these tokens are in euros. The first transaction of Block 1 is The Joker gives €25.67 to Dr. Evil. It does not matter how many transactions there are in a block. This blockchain with transactions is also distributed as is described in the previous section. So when we change a transaction in these blocks, the blockchain will become invalid and we would have to remine it. The most important feature of this money blockchain is that it does not lose track of the transactions, if a block is changed in the blockchain we will notice. The blockchain is resisting modification; this is a good reasoning for using blockchain to remember tokens. It is worth mentioning that we do not check whether The Joker has €25.67, the only thing we are seeing is The Joker gives €25.67 to Dr. Evil. The blockchain does not remember bank account balances; it remembers money movements. So how do we check whether the Joker has enough money to make this transaction? In this blockchain, we do not know since our first block does not give any information on this matter.

## 2.6 Coinbase

In blockchain, there is a specific type of transaction it is called a "base" transaction. The transactions we have seen before were movements of euros in the blockchain. These are different transactions since they only move money around. Base transactions pump extra euros into the blockchain. If the blockchain does not contain any money, other transactions should be stopped as well. The final extra feature to our blockchain, coinbase:


Luc's Coinbase

Block: # 1

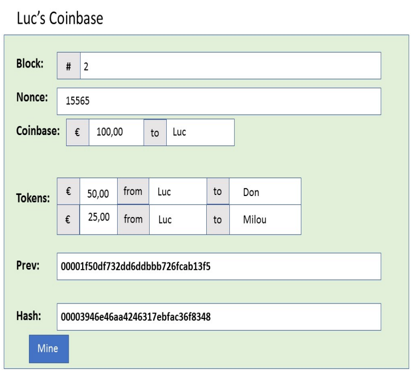
Nonce: 74522

Coinbase: € 100,00 to Luc

Tokens:

Prev: 00000000000000000000000000000000

Hash: 00001f50df732dd6ddbbb726fcab13f5


Luc's Coinbase

Block: # 2

Nonce: 15565

Coinbase: € 100,00 to Luc

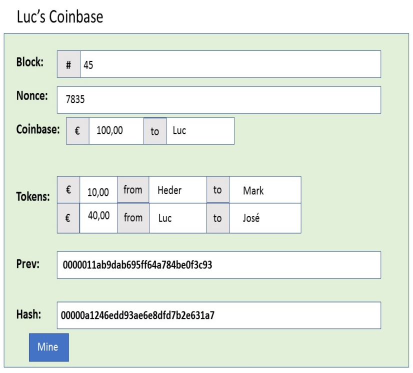
Tokens:

| | | | | | |
|---|---|---|---|---|---|
| € | 50,00 | from | Luc | to | Don |
| € | 25,00 | from | Luc | to | Milou |

Prev: 00001f50df732dd6ddbbb726fcab13f5

Hash: 00003946e46aa4246317ebfac36f8348

In the first block, there is a coinbase transaction, 100 euro is given to Luc. There are no transactions in block 1 because there was no money to move around yet. In block 2 another coinbase transaction occurs, and again Luc receives 100 euro (great technology). In this block, there is money to move around, and we have two transactions. Luc gives €50 and €25 to Don and Milou respectively. All transactions are from Luc because it is the only person with any money. We can check whether Luc has €50 to send. In the first block, we can see that Luc has indeed 100 euros from the coinbase transaction. There is a small check if all transactions do not exceed €100, so that we will not invent money out of thin air. This can only be done with a base transaction which can be in any block. Let us take an example a little bit further in the blockchain:


Luc's Coinbase

Block: # 45

Nonce: 7835

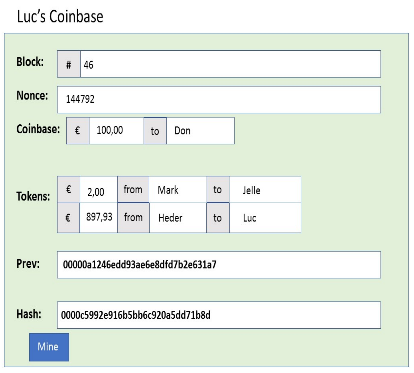
Coinbase: € 100,00 to Luc

Tokens:

| | | | | | |
|---|---|---|---|---|---|
| € | 10,00 | from | Heder | to | Mark |
| € | 40,00 | from | Luc | to | José |

Prev: 0000011ab9dab695ff64a784be0f3c93

Hash: 00000a1246edd93ae6e8dfd7b2e631a7


Luc's Coinbase

Block: # 46

Nonce: 144792

Coinbase: € 100,00 to Don

Tokens:

| | | | | | |
|---|---|---|---|---|---|
| € | 2,00 | from | Mark | to | Jelle |
| € | 897,93 | from | Heder | to | Luc |

Prev: 00000a1246edd93ae6e8dfd7b2e631a7

Hash: 0000c5992e916b5bb6c920a5dd71b8d

In block 46 we notice that Mark gives €2 to Jelle. When we go back one block, we see that Heder gave Mark 10 euros, so that Mark can give Jelle €2 in block 46. The Prev allows us to find the previous block and find out whether or not a transaction can occur.

This is a basic blockchain where we are running a currency on top of it. Because the blockchain is distributed to many peers any modification in this blockchain will be noticed, this is the main reason why blockchain is safe. It is a very efficient way to agree on what happened in the past; this will be discussed in the Mining section.

## 2.7 Genesis Block

The genesis block is the first block of the blockchain. This is the only block without a previous block. This begs the question if it less safe than the other blocks since it cannot be checked. In the Genesis block of bitcoin, there is a coinbase transaction of 50 bitcoins to an address. Probably this is the developer (Satoshi Nakamoto) who started the first transaction. The genesis block is easy to find on the internet and open to everybody who is interested. If he had credited himself with 1.000.000 bitcoins, it would be open as well, but this would have diluted the value of the bitcoin. The genesis block is basic and therefore easy to inspect. The genesis block is a start, and the Nakamoto started with 50 bitcoins. The next block has the genesis block as previous, and so the blockchain begins. Note that if the genesis block is altered by hacking, the whole blockchain would be invalid and this hacked version will be removed.

## 2.8 Consensus and mining

Every blockchain has a consensus model, how the blocks are validated. A consensus model exists out of three parts, find something that:

1. is hard to do
2. is easy to verify
3. enforces a linear history

These are the 3 key elements to any consensus model. Due to the demand that a block must be signed, block validation becomes a hard thing to do. It is easy to verify, we can check by inspecting the hash. The linear history is enforced by the chain itself.

There are different types of blockchain, at first we need to decide if a blockchain is anonymous, or permissioned. I will discuss these types further in a later section. If the blockchain is anonymous then we need some proof that the validation of a block has been done and is correct. In a permissioned blockchain, we trust the validators and therefore a proof is not necessary.

Miners are the peers of the blockchain and validate the blocks in the blockchain. Every time a new block comes in, miners are competing with each other to be the first to validate the block. Every miner validates the new block, and makes a hash. If one would win this game, they will receive a small payment. In the Bitcoin blockchain whoever wins the validation game receives transaction fees and newly issued bitcoins. The number of issued bitcoins is determined by which block is validated. The further the block is in the blockchain the fewer number of new bitcoins one will receive. In the future, there will only be transaction fees.

**CAP Theorem** The CAP theorem states that any distributed system cannot have Consistency, Availability and Partition tolerance simultaneously. This theorem is also known as Brewer's theorem; Eric Brewer introduced it as a conjecture in 1998; In 2002 it was proved as a theorem by Seth Gilbert and Nancy Lynch. Consistency is a property that ensures that all nodes in a distributed system have the last copy of the distributed ledger. Availability means that the system is up, accessible for use, and is accepting incoming requests and responding with data without any failures as and when required. Partition tolerance ensures that if a group of nodes fails the distributed system still continues to operate correctly. [3] Gilbert and Lynch proved that these three properties together cannot exist in a distributed system. If we check the CAP theorem against Blockchain, it seems that this new blockchain technology uses the three the properties together, however this is not the case. Blockchain technology favors availability and partition tolerance over consistency. Consistency is achieved in blockchain but not simultaneously with availability and partition tolerance. Consistency is achieved over time. This is called eventual consistency, multiple nodes validate blocks over time and consistency is achieved as a result. In bitcoin mining was introduced to achieve consistency. Consistency is achieved by using a consensus algorithm. Some consensus algorithms will be described in a further paragraph.

**Fault tolerance** Fault tolerance is an important aspect of Blockchain. Whenever a node crashes or exhibits malicious or inconsistent behavior, the blockchain should not be affected by this. To achieve fault tolerance, replication is used. This is a commonly used method to achieve fault tolerance. Consistency is achieved using consensus algorithms to ensure that all nodes have the same distributed ledger. This is called state machine replication. Blockchain is a method to achieve state machine replication.

**The problem of the byzantine generals** The first development of successful and practical consensus mechanisms happened last century. In 1982 the problem of the byzantine generals was introduced: There is a group of army generals who are leading different parts of the Byzantine army. They have all have to collaborate to seize cities. To seize a city, all generals should attack at the same time. To communicate the generals have one messenger. The problem is that one or more generals can be traitors and can communicate misleading messages. The generals need to find a feasible mechanism that allows them to safely seize a city even when there are traitorous generals present. Translating this problem to distributed systems: generals are the nodes of the system, traitorous generals are the malicious nodes, and the messenger is the communication channel. Castro and Liskov solved this problem 1999 with their Practical Byzantine Fault Tolerance algorithm (PBFT). The first practical implementation is the consensus algorithm of bitcoin: Proof of Work.

**Types of consensus** Consensus is a distributed computing concept that has been used in blockchain in order to provide a means of agreeing to a single version of truth by all peers on the blockchain network. There are two main categories of consensus mechanism:

- Byzantine fault tolerance-based, which reaches consensus by rounds of votes of the nodes. This is explained in the first paragraph of this section (Consensus and Mining).
- Leader-based, which reaches consensus by electing one leader who proposes the final value: The election procedure is done for every new block, and every node can compete in the election. The election is a lottery, and a leader is picked by chance.

Below the most used consensus algorithms are described.

**proof of work** Since it is easy to validate a block and make a hash of a new block, mining can be very rapid. If Bitcoin does not slow this mining process down, many blocks will be generated, and all bitcoins would be mined in no time. Therefore Bitcoin devised two characteristics. Only one new block is issued in every 10 minutes and Bitcoin requires the miners to have a proof of work (PoW). PoW shows that a miner used some amount of computing power. This is where signed blocks come in, the PoW we used in the sections above is the requirement that every block starts with 0000. A miner must provide a hash that starts with 0000 otherwise we will not acknowledge his work. As we recall this was done by computing the right nonce, this is what all miners essentially are doing. They race each other to be the first to the correct nonce.

Next to the bad ecological impact mining has on the environment. There is a danger to this approach as the reward for mining reduces in time, more miners will think it is not worth the work anymore. This results in a mining market with only few players left. If a player can get a 51% of the "mining market", this player can alter blocks and validate them without being stopped as the market is controlled by him/her. This is called a 51% attack.

**proof of stake** Proof of Stake (PoS) serves a similar function to the proof of work which underpins the security, but has significant advantages in terms of security and energy efficiency. PoS is not about mining; it is about validating. Blocks still need to be created by someone; this someone is chosen by a specific Proof of Stake algorithm. This selection process must have some kind of randomness, or at least distribute voting shares properly. Take the example of Ethereum, Ethereum is a blockchain currency like bitcoin. To become a validator in Ethereum, one must deposit some amount of ether (ethereum tokens) which results in a share in the Ethereum network. The higher the amount some validator put into the network, the higher the validator's share percentage will be. This share percentage is used as a collateral to vouch for a block. In PoW the chain is valid because lots of work is behind it, while in PoS the chain with the highest collateral is trusted.

The key benefits of PoS is that there is less computing power used, therefore, less electricity and the reduction of risk of a 51% attack. It is far more expensive to own 51% of a currency than 51% of the computing power. Even if someone obtains 51% of a certain currency, everyone could stop using this currency, making it worthless.

A variation on PoS is proof of burn. To become a miner in a PoB consensus model, one first has to "burn" some amount of currency in the blockchain. What is meant by burning coins is sending the coins to a wallet/address where they cannot be redeemed. In return, mining rights are rewarded. Burning more coins leads to a higher percentage in being assigned the next block. It is like PoS but the deposit is lost. Other variations on this are that the deposit is returned after a certain period.

**Delegated Proof of Stake** Delegated Proof of Stake (DPOS) is an innovation over standard PoS whereby each node that has a stake in the system can delegate the validation of a transaction to other nodes by voting. In DPOS every node can vote for a delegate. These delegates validate the blocks and are awarded transaction fees. There is a set number of delegates slots, and the node with the most votes for a slot wins the delegate slot. This change makes it possible for nodes with little stake to have more say about the ledger. The downside is that the ledger is less distributed which results in more insecurity. This is used in the bitshares blockchain.

**Proof of importance** Proof of importance resembles the PoS but in PoI also relies on the usage and movement of tokens by the user. These two extra properties establish a level of trust and importance of a user. The level of trust and importance plays a role in becoming the lottery winner. This is used in Nemcoin.

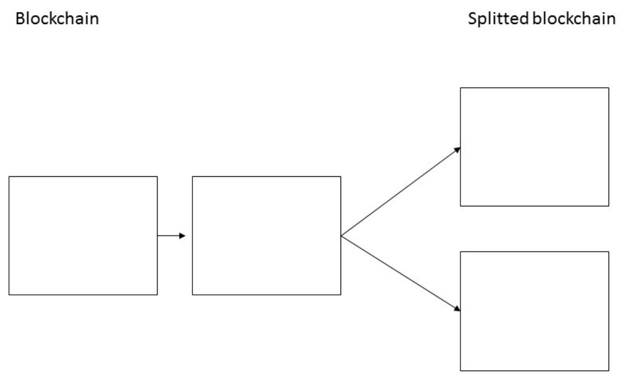
**Federated consensus or federated Byzantine consensus** The nodes in these systems select a group of publicly trusted peers and let them validate blocks. The vote of the trusted group does not have to be unanimous. The majority wins. This is used in the stellar consensus protocol.

**Reputation-based mechanisms** A leader is elected on the basis of the reputation it has built over time on the network. This can be based on the voting from other members.

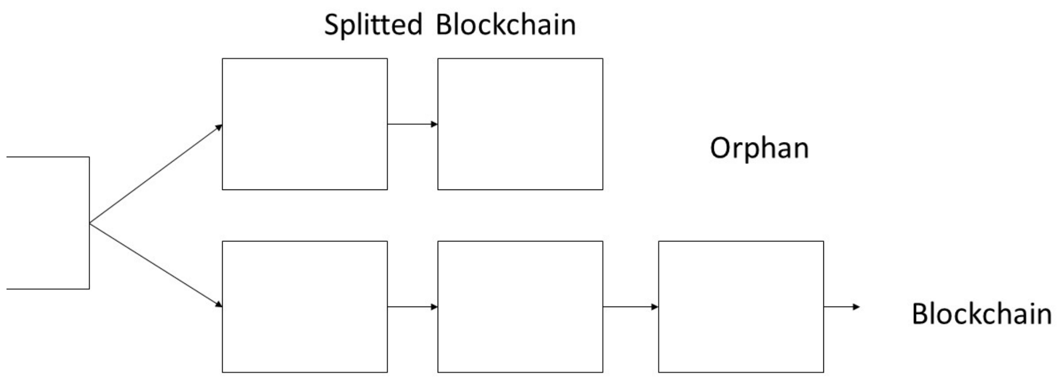
## 2.9 Forks

A fork is when a blockchain splits in two blockchains (the babies). This can be a temporary or permanent split. Temporary splits happen all the time in the blockchain itself, permanent splits happen when the rules of the blockchain are altered. Altering of the rules can be for a good reason, for example, to upgrade the blockchain. A split in the blockchain is not desired because the babies are weaker than their mother, relative to mining power and clients.

Temporary splits in the blockchain happen frequently, usually when a miner mines a block and sends it to validation it is the only block that has to get validated at that exact moment. However, it can happen that two miners mine a block at the same time. Both blocks will be assigned to the blockchain (if valid) on top of each other.



Miners always choose the last block of the blockchain to start mining. In this case, there are two blocks at the end of the chain. Miners are working at both sides to try and add blocks. It could happen that both sides again finish a block at the same time than the chain remains split. When one of the chains becomes larger than the other one, all miners are going to work from there since miners always start at the last block. The other chain is left behind and is called an orphan.



All blockchains have a set of rules which defines the blockchain. Rules such as, transaction formats, block formats, built in limits or bytecode. Changing the rules has consequences and could lead a blockchain to fork permanently. When the rules change not all nodes will be able to participate or do not want to participate. Imagine a software update is released, most people choose to update their software, but some people choose to keep the software the way it was. If the new and the old version are compatible, the division of groups will not be a big problem, but if these versions are not compatible, then it is very likely that there are going to be two camps. The same thing happens with blockchains, the division in two groups leads to a fork and the chain splits. As long as both chains are supported, both chains will continue. It is hard to update a blockchain (change the rules) since a fork in the blockchain is not desired. There are two main methods how to tackle this problem, a hard fork where the new rules replace the old rules, a soft fork where new rules are added, but the old rules remain intact.

**hard fork** Hard fork is a change that is not compatible with non-upgraded software. The old rules become invalid under the new rules. A deadline is picked, and after this deadline, the rules will be changed. Every node has the chance to upgrade their software in the meantime. If every node would upgrade then the new rules are adopted by everyone, and the blockchain would go on without losing users. In blockchains with a small number of users, this is possible since it is easier to communicate and explain the change. In blockchains with a big number of users, this is much harder. Some users will not agree with a new set of rules, others may not be able to upgrade their software, and some will not be paying attention when the new rules are announced. These groups could choose to stick with the old rules and move forward with the "old" blockchain, leading the chain to fork in a chain with the new rules and a chain with old rules. A hard fork is an efficient and clean way to upgrade since it provides a "fresh start" but a hard fork is more sensitive to a fork in the chain than a soft fork.

**soft fork** Soft fork is a compatible change, it adds rules and does not remove existing ones. This is called a soft fork because not all nodes have to upgrade their software. Only the users in the consensus mechanism must be upgraded. For proof-of-work blockchains, this means the miners; for other blockchains, this means the block signers (the users who approve the blocks). Other nodes will not have to be upgraded since the old rules are still valid. To implement the change, block signers will stop working for a moment. This is for their own benefit since it is not sure if the change is going to be adopted well. When the change is not adopted well, this could lead to a permanent fork. The soft fork method is a slow upgrade method relative to the hard fork method, so a split is less likely to happen. A soft fork can be a good choice if a blockchain has many distributed users. When a blockchain has little users and centralized a hard fork is a good possibility.

Neither hard forks nor soft forks are intended to fork the blockchain. Both are ways to upgrade the blockchain and could lead to a fork. Forks happen when big parties are divided over what should happen with the blockchain in the future. If they cannot agree both will continue with their blockchain vision and users can choose which to follow. When the decisive parties are unanimous, a fork is unlikely to happen.

Recently bitcoin announced to change the block format to keep up with the popularity of the coin. Bitcoin customers can have to wait for three days to be confirmed and get their money. This is caused by the rule that a block has a maximum of 1MB and only one block can be processed every 10 minutes. This rule was set to shield bitcoin against DDoS attacks. Due to this limit, the system slows down. The developers proposed to cancel this 1MB rule, but the mining community does not agree since it would be unfeasible to mine blocks. This difference in vision seemed to lead to a fork in the bitcoin blockchain. However, the developers and miners came to an agreement to revamp the blockchain software. This allows bitcoin to increase the processing limit to 2MB[5].

# 3 Types of blockchain

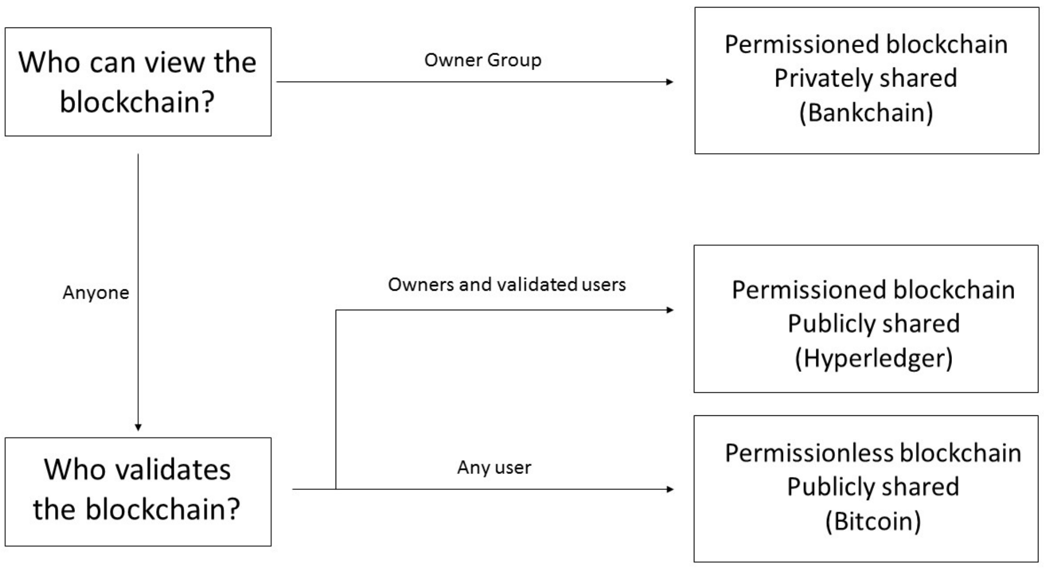
We briefly discussed two types of blockchains in this paper: anonymous and permissioned. We will rephrase anonymous as permissionless, so we have permissioned and permissionless blockchains. In which category a blockchain falls, is classified by consensus. Consensus is by whom the blockchain is validated. Validation can be done by private users, permissioned blockchains, or every user, permissionless blockchains.

## 3.1 Permissioned blockchains

Permissioned blockchains can only be validated by owners and validated users. Only trusted validators participate in consensus. These trusted validators are not anonymous, so a proof is not necessary anymore. An example of a trusted validator is the government. Permissioned blockchains can be distributed but do not have to be. If it is not distributed, then the blockchain is said to be private. A private permissioned blockchain is centralized under an organization which controls the right to view and send transactions. An example of this is a bankchain.
A permissioned blockchain is said to be public when everyone can view the blockchain. A public permissioned blockchain is distributed but still controlled by the owners and validated users. An example of this is called a hyperledger which will be discussed in the section: "Blockchain in Business".

## 3.2 Permissionless blockchains

In permissionless blockchains, every user participates in the consensus procedure. Since every user is anonymous in these chains, it is necessary to have proof that ensures us that a new block is valid in the chain, this is all discussed in the "What is blockchain" section.
A summary diagram of all the blockchain types:


Who can view the blockchain?
Owner Group
Permissioned blockchain Privately shared (Bankchain)
Anyone
Owners and validated users
Permissioned blockchain Publicly shared (Hyperledger)
Who validates the blockchain?
Any user
Permissionless blockchain Publicly shared (Bitcoin)

There are three main types of blockchains as we can see above. When to use which type is essential in using blockchain technology. In the next section, we will discuss the business side of blockchain.

# 4 Blockchain in Business

There are different types of blockchain, each with a different purpose. Companies know which clients they do business with. Companies are unlikely to do business with clients who are anonymous. Most blockchains in business will be permissioned blockchains; this is why the bigger part of this section we will discuss permissioned blockchains in business.

## 4.1 What is innovative about blockchain?

The most innovating part of blockchains, is the possibility to have a distributed ledger/blockchain which can be viewed and updated by enabled users. This innovation makes it possible to link businesses to each other. Let us take an example in the music right business.

Every time a song is played on the radio/spotify, the owner of the song receives a small payment. The rights to a song can be bought/sold which happens frequently. "Dutch Music" (it is made up) is trying to find every owner of every song that is played and pays them. Dutch Music has a large database of many owners of a song but not all owners. Every day they have to call several notaries to find out if there has been a switch in the music rights and make the necessary changes in their database. During the day, they will encounter songs that they will not have in their database. In these cases, they need to call another company in the same business, for example, "Swedish Music". Swedish Music might be able to help them, and otherwise Dutch Music will call the next. Like Dutch Music and Swedish Music there are a lot of other companies with incomplete databases. As we recall, blockchain technology makes it possible to have one distributed ledger/chain that can be viewed and updated by enabled users. If all "Music" companies would collaborate and store their data in a blockchain that contains the music rights of every song it would resolve a lot of unnecessary work such as double tracking and locating the right data. It would be great if they also linked the database of the notaries to this chain in order to keep all data updated. In this example the chain would be publicly shared, the validators of the blockchain would be the "Music" companies and notaries, so this is a public permissioned blockchain.

Blockchain in business is all about distributing data in order to resolve redundant work and speeding up processes.

## 4.2 Advantages of different blockchains

There are three types of blockchain, permissionless, public permissioned and private permissioned. Each chain has different properties and different advantages. In this section, we will discuss the advantages of each type.

**Permissionless blockchains** permissionless blockchains are public which means that they are accessible to anyone. It is not uncommon for a public chain to hide the identity of all associated participants (Bitcoin). This openness comes with advantages such as the ability to resist hacking or capital controls from oppressive regimes. The public distribution of the chain makes sure that every participant can see all account balances and the movement of all transactions. This ensures security. This is a new manner of security, and many people are still uncertain about how secure this approach actually is. There are numerous well-documented papers on security of blockchains. This paper will not go in the same depth about security of blockchains. However, we will mention that the most known permissionless blockchain, Bitcoin, has never been hacked in its eight-year existence. This is quite an achievement as there is a payday of 40 billion euros.
An other advantage of using a public blockchain is that they protect the users of an application from the developers, establishing rules that even the developers have no authority to change. The last reason to choose a public blockchain is that anyone can build on it and it benefits from the same network effects Internet does.

**Private permissioned blockchains** Private blockchains are not accessible to anyone, only users who are enabled by the owners of the blockchain can view the ledger/chain. Private blockchains are secured by user rights and passwords/secrets. These chains are mostly used between parties who trust each other not to mess with the contents of the chain. Private chains are less secure than public chains since a hacker can find out the login data and once he is in he can change the ledger/chain. Private blockchains can work great for small groups of people who deal with sensitive data. The fewer people who know about the chain, the safer it is. Some benefits of private blockchains are: privacy, better performance, faster evolvement and cost effective. More privacy in private blockchains is due to the difference in the consensus construction. Private chains give a better performance because the blocks are validated after each other. This salvages a lot of redundant computing power (in public chains everybody tries to validate the same block). In private blockchains, there is an owner who controls the set of rules. This makes it possible to adapt these rules quicker and lead to a better performance of the chain. Public chains have a set of rules as well, but there is no owner who can adapt these rules, making it harder to evolve over time. At last private chains are more cost effective as only a few users have to validate the blocks instead of every user. Permissioned blockchains are liked because they are easier and faster to update, they can be optimized for specific purposes, and it is easier to comply with both existing regulations and user expectations.

**Public permissioned blockchains** Public permissioned blockchains are a mixture of the accessibility of permissionless blockchains and the managing of private blockchains. Public permissioned blockchains are distributed between enterprises that are interested in that blockchain. Public permissioned blockchains are usually collaboration blockchains used by different enterprises at the same time. Private permissioned blockchains are used within an enterprise itself.

**Future prospects** While user rights management secures private chains, a mixture of cryptography and economic incentives secures public chains. Different organizations and users have varying goals for their networks. Public and private blockchains both have their properties and businesses can choose which to implement. Because of this diversity, it is unlikely for one method to prevail against the other. The real question we need to ask ourselves before starting a blockchain is: what problem are we solving and establishing by using a blockchain. Is a public, private, or mixture solution better than a centralized database or using a time stamp server + public key cryptography?

For many enterprises, a permissioned blockchain can meet business requirements that are impossible to meet with a permissionless blockchain. Enterprises are not fond of anonymity, the clients they have are known. Companies like to set their own rules which they can manage themselves or with associates. Therefore, it is likely that permissioned blockchains will be used more often in business than permissionless blockchains.

## 4.3 Smart contracts

One of the other innovations of blockchain is smart contracts. A smart contract is a contract which is put in one of the blocks of the blockchain. A smart contract is the same as any other contract only smart contracts can be executed by the blockchain itself. An example of a smart contract is a transaction. Before a transaction occurs first the balance of the debtor is checked, if there are enough funds the transaction occurs. If not, then the transaction does not occur and the block is invalidated. In this case, the agreements of the contract are whether the debtor has enough funds. The obligation of the contract is the transaction from debtor to creditor. Smart contracts automatically enforce the obligations of the contract when the agreements are met. They can be executed without human interaction which saves time and work.

In the transaction example above a lot of time and work is saved by smart contracts. The example above is transferring money done by banks at the moment. Now, when a transaction from A to B occurs, the bank checks if A has enough money to make the transaction. Then the bank debits A's account and credits B's account. The bank updates their ledger, and we all had to trust the bank. As we have seen due to smart contracts Blockchain can do the exact same thing as the bank. An other advantage of blockchain is that we do not have to trust this single-entity third party anymore (banks, clearing houses, etc.). There is no need for trust anymore since we can verify all transactions using the distributed ledger. This is why public blockchain is called "Trustless". Smart contracts can replace any existing contract as long as we can put it in the blockchain. There are many applications possible, and we will discuss a couple:

Business is usually done between two parties, say Company A agrees to do a project for Company B. They agree on a contract to pay half of the fee when half of the work is done and the other half of the fee when the project is finished. It is in the interest of company A to obtain the first half of the fee as soon as possible. For company B it is important that the project is done properly and are therefore cautious with transferring their money. The result of this difference in expectations is a silly game where A argues that the half of the project is already reached and B argues that they expected more in this stage of the project. This game is played a lot in business and it a waste of time and resources. A smart contract ends this game by automatically enforcing the obligations whenever the agreements of a contract are reached. In this case, the rules of the project are set in a smart contract which cannot be bend anymore since it is recorded in the blockchain. Due to the distribution of the blockchain the rules of the contract can easily be obtained by everyone with a copy.

The second example we will discuss is in the mortgage sector. Usually when we apply for a mortgage, we have to go to the bank and provide them with our personal information. The bank is going to review our information and decides whether to offer us a mortgage and how high the mortgage will be. These steps can take a couple of weeks. The time-consuming step in this process is the reviewing. The reviewing itself is not time consuming but the human interaction around the reviewing is. When a request for a mortgage comes in an employee checks the record and gives it a score. After the scoring, the employee sends it to the next station. The next station is not sure whether the request is already scored and checks the same record again just to be sure. These steps are repeated at every station. This is a lot of waste of time and resources. It would be much easier to put the record checking in a smart contract, which can check a record faster than an employee, and keep a distributed ledger of mortgages throughout the company. If we put these steps in smart contracts, we can go to the bank, identify ourselves and in five minutes we would have the appropriate mortgage.

The great benefit of smart contracts is the automatic enforcement. A lot of time is wasted on getting in touch, double work, arguing about rules and waiting. In the Netherlands, we have to go to city hall to pick up our drivers license. Due to long waiting times, it can be two hours before it is our turn. At the employee's desk we need to pay and the employee checks our photo if it is allowed on our drivers license. With some bad luck the photo is disapproved, and we will have to come back and wait again. A better approach for city hall would be having a vending machine with driver licenses where people can just pay a small amount, and in two minutes they have their drivers license. This sounds crazy but with smart contracts this is possible and even online, so going to city hall is history. To see what this would look like we first need to establish the requirements of a drivers license. The requirements for a drivers license are passed exam, approved picture and a small payment. So the agreements of the smart contract will be passed exam, approved picture, small payment. City hall should make a blockchain with this smart contract in it. For even better performance they could link it to the blockchain of CBR (the drivers exam company) which provides them with all passed people. Now obtaining a drivers license would be two minutes behind a computer. We would simply upload a photo, which can be approved or disapproved in 5 seconds by an algorithm, and make a small payment and the drivers license will drop in our account.

Smart contracts are faster, cheaper and more fair than traditional contracts. We can cut out a lot of time and cost by reducing a lot of unnecessary work done by employees. Smart contracts not only define the rules and penalties around an agreement in the same way that a traditional contract does, but also automatically enforce those obligations.

# 5 Blockchain and ...

This section is some sort of encyclopedia of blockchain in business for decision makers who are interested in the blockchain possibilities on some use cases. consensus/permissioned/public.

## 5.1 mobile

Will there be mobile blockchain in the future? It is possible to run a blockchain on a smartphone. A smartphone is a small computer which is handy for being accessible and updated. However, a blockchain asks for much computing power, which a smart phone does not have, making it unpractical to run a blockchain on a smartphone. Even if there are smartphones created with more computing power blockchain will still consume the battery of the smartphone fast since blockchains are frequently updated. It is unlikely that a mobile blockchain will become feasible in the near future.

## 5.2 Internet of things

The internet of things is about things connected to other things via the internet. A thing is a device with an on/off switch. Take for example the inhouse-regulation for lamps. It is possible to turn on/off the lights in a house with a smartphone. In this case, the smartphone is connected to the inhouse-regulation device. A different example is the meters in the Dikes which continually pass along the height of the water. We should keep in mind that blockchain is about value transferring so how to connect it? IoT works with a centralized model at the moment. The number of devices will grow rapidly, generating many transactions and increasing computational requirements. Other bottlenecks for IoT will be the servers that can no longer take data traffic from compromised devices. The nodes in a centralized model are less connected to each other than in a blockchain model where every node is connected. Blockchain technology will enable the creation of secure mesh networks. Every node of the network is registered on the blockchain, devices can identify and authenticate each other without the need of any human. This blockchain-network can also cope with many devices without being slowed down. Such a blockchain is used within enterprises to change their centralized models. These chains will be private permissioned blockchains. However, device identification and intercommunication are secured by a permissionless blockchain that holds the unique identity of each participating node in the network.

## 5.3 Supply chain management

Blockchain technology and SCM can be combined well. SCM faces the problem that the supply chain is not transparent. Supply chains are networks that exist of many stages and many geographical locations. This makes it hard to trace events and investigate incidents. For customers, this means that they cannot verify or validate the true value of the products and services. This results in an inaccurate reflection of the costs of production. Blockchain enhances the transparency and security of SCM. In the distributed ledger we can implement a simple application such as registering the transfer of goods on the ledger that would identify the parties involved. Other features that can be included are the date, price, quality, and state of the product and location. Blockchain technology makes it easier to trace back the events in the past. Accountability and security will no longer be a problem. This change in SCM will lead to dynamic demand chains in place of rigid supply chains, resulting in more efficient resource use for all. This resolves of disclosure and accountability. Usually, one party in the supply chain updates the chain, and they have to inform every user of the supply chain, this takes much time and is error prone. Many supply chains have to be updated often. In a blockchain, this can be done faster and without any mistakes. It seems like permissioned public blockchains are going to rule SCM in the future. These chains are regulated by the companies but distributed at large scale.

## 5.4 Housing market

Everything that has a value can be put into a blockchain. Houses have value, and therefore there can be a blockchain of the housing market. This blockchain could work great since we can cut out brokers and use the blockchain instead. For this to work, a platform should be made where home owners can put their houses for sale in a smart contract, if the buyer meets all the agreements of the smart contract, the owning rights to the house will immediately be transferred. Everybody who wants to buy/sell a house should be allowed to have a copy of the blockchain, so this is a public chain. This blockchain must also be a permissioned blockchain. If this chain would be a permissionless chain, anonymous buyers and sellers can exchange houses, and nobody will know to whom the houses belong (except for the owner). The government will not accept this, and therefore this chain should be permissioned.

## 5.5 Perishability

Perishable products are not a part of blockchains yet. It is an interesting topic since there are a lot of perishable products. When we take a look at the Bitcoin blockchain, there is a base transaction which provides a way for new coins to enter the chain but there is no "empty" or "remove" transaction. In the Bitcoin blockchain, it is impossible to extract bitcoins from the chain. For Bitcoin, it is not necessary to remove bitcoins from the system so let us say that we will develop a public permissionless blockchain for music rights. In this chain, all the transactions of music rights are embedded. The expiration dates of the rights are kept by the government. The government should have the power to remove the music rights from the chain as soon as they expire. In a public permissionless blockchain, this is not possible since the users are protected from changes by the developers of the chain. In a permissionless blockchain, there cannot be one entity with the power to remove value out of the chain. For perishable products to work, we need to use a permissioned blockchain in which we agreed in advance that the government has the power to remove rights as soon as they expire. This will likely slow down the transaction process as not every user can validate the blocks anymore. We can extend this idea to currency of a country. If we are going to make a blockchain currency the legal currency of a country, we will need to have an entity such as the government who can freeze assets of criminals. This has the same impact on the chain as the perishable products. So if there is going to be a blockchain currency nationally or internationally accepted, there will likely be rules and therefore this must be a permissioned blockchain. Permissionless blockchain currency is restricted in its use by default.

## 5.6 Sports

Blockchain can be used for sports as well. Smart contracts can be used instead of traditional contracts between players and clubs. We can cut the job of negotiators and let the smart contracts take care of the negotiations. Agents of players with a big network are redundant as the ledger is distributed between all clubs. In an ideal world, tickets for seats in the stadiums can be put into a permissionless blockchain. However, there are some hooligans who have a stadium ban and should be checked for. Therefore, the blockchain cannot be anonymous. All of these examples are public blockchains, these chains are used to collaborate and should be distributed between the collaborating parties.

## 5.7 Law

Blockchains are becoming more and more relevant in business. Smart contracts will develop quickly which is going to result in different expectations due to misunderstanding. There should be rules assigned what to include in smart contracts. There should be contractual facts, for instance, a soccer player can be bought by clubs on the market if they pay enough money. There should also be human preference embedded; a soccer player should get a say in where he wants to play. If the player does not want to play in the Netherlands, then clubs in the Netherlands should not be able to execute this smart contract. As we know little of civil rights and commitment law, we want to impose on other researchers who do have the knowledge to investigate this matter further.
There is no double checking in a smart contract, when the agreements are met, the contract is automatically enforced. Further research should investigate the contents of smart contracts.

# 6 Results

Most use cases of blockchain in business are public permissioned blockchains. Blockchain technology enables us to share a distributed ledger and promotes collaboration. Public permissioned blockchains are shared distributed ledgers between collaborating enterprises. These blockchains can contain private data, are cost effective, evolve fast and have a good performance. Private permissioned blockchains are more used internally in enterprises to speed up their centralized models. Permissionless blockchains are faster than permissioned blockchains, but in business they will not be much used due to the anonymity of users, private data of enterprises and lack of control. Permissionless blockchains are great where openness is an important factor but is limited in usage if there are rules and an entity which determines these rules is involved.

## 7 Discussion

Few research is done about blockchain in business. Most research is done on the security of blockchain. This paper is not based on previous research but on the understanding of blockchain and creativity. Most examples in the paper are described in general but for it to work in business more research should be done into the details. This paper provides guidelines into the new blockchain technology and does not provide hand ready-made blockchains. This paper aims to inform everyone on blockchain and show what the possibilities are in the business sector.

## References

[1] LaTeX: *Mastering bitcoin, unlocking digital cryptocurrencies*, Andreas M. Antonopoulos, 1st edition, December 2014.

[2] LaTeX: *Mastering bitcoin, unlocking digital cryptocurrencies*, Melanie Swan, 1st edition, February 2015.

[3] LaTeX: *Mastering bitcoin, unlocking digital cryptocurrencies*, Imran Bashir, 1st edition, March 2017.

[4] http://www.md5hashgenerator.com/

[5] http://www.itpro.co.uk/cloud/29095/blockchain-upgrade-helps-avert-split-in-bitcoin