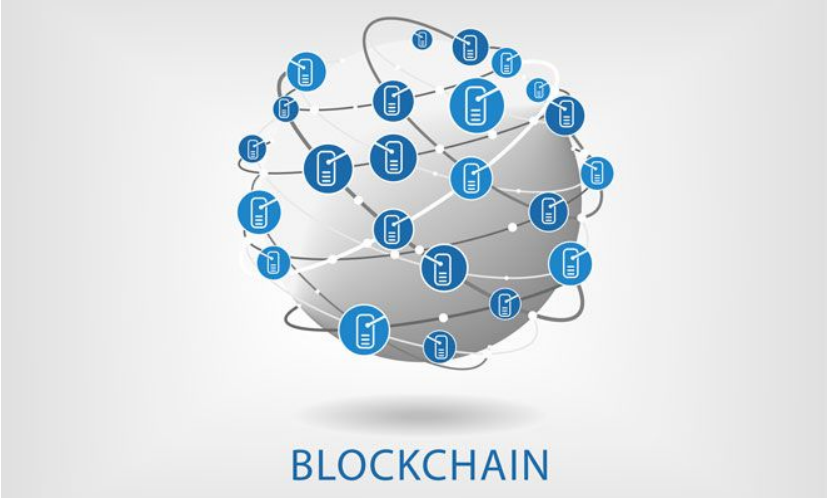
UNIVERSITY OF GRONINGEN
FACULTY OF SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTING SCIENCE

Information Systems



Group 07

AUTHORS
Frans Simanjuntak (S3038971)
Stefan Cretu(S3048438)

December 18 2017

Contents

1. Introduction

In this document is presented the solution for the Blokchain part of the assignment 3 of Information Systems course. The purpose of the assignment is to create an Ethereum smart contract that allows shareholders to make a binary decision (true or false) which later on the director can propose the final solution to shareholders.

The rest of the document is structured as following: Chapter 2 describes the theory of Blockchain, Chapter 3 describes the use case scenario, whereas Chapter 4 describes the implementation of the smart contract.

2. Blockchain

A blockchain is a digitized, decentralized, public ledger of all cryptocurrency transactions. Constantly growing as 'completed' blocks are recorded and added to it in chronological order. It allows market participants to keep track of digital currency transactions without central recordkeeping. Each node gets a copy of the blockchain, which is downloaded automatically [1].

A block is the 'current' part of a blockchain, which records some or all of the recent transactions. Once completed, a block goes into the blockchain as a permanent database. Each time a block gets completed, a new one is generated. There is a countless number of such blocks in the blockchain, connected to each other (like links in a chain) in proper linear, chronological order. Every block contains a hash of the previous block. The blockchain has complete information about different user addresses and their balances right from the genesis block to the most recently completed block. The structure of the blockchain is described in figure 1.

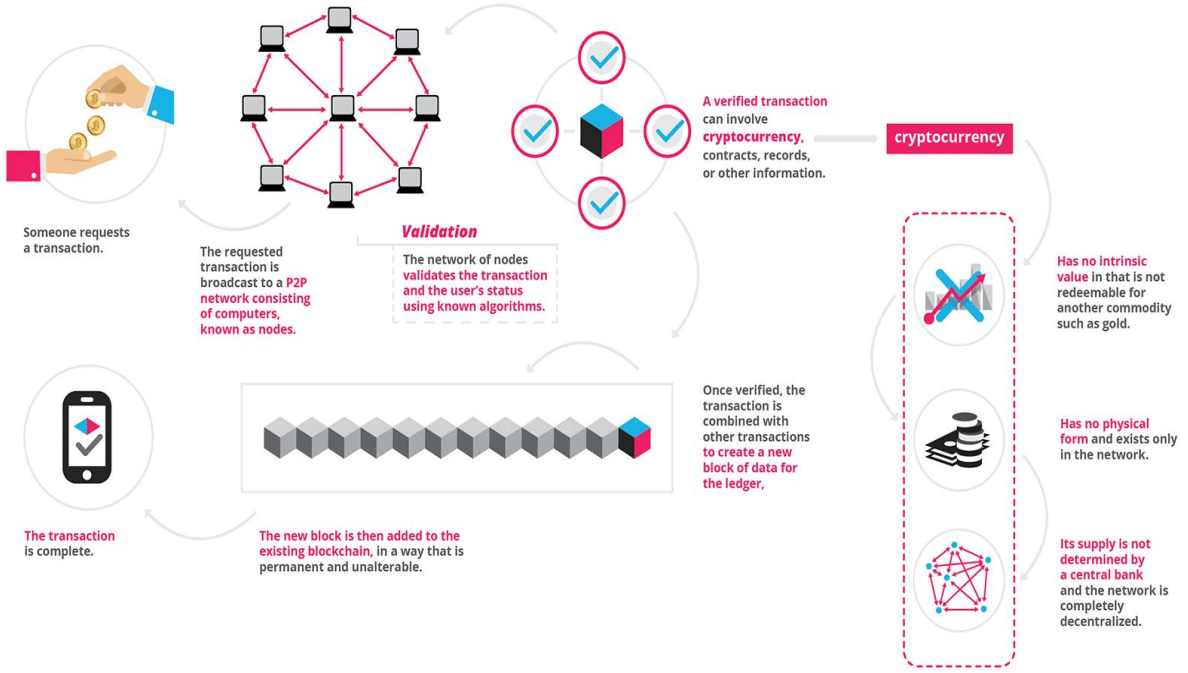


Figure 1. The structure of blockchain

The blockchain was designed so these transactions are immutable, meaning they cannot be deleted. The blocks are added through cryptography, ensuring that they remain meddle-proof: The data can be distributed, but not copied. However, the ever-growing size of the blockchain is considered by some to be a problem, creating issues of storage and synchronization.

Blockchain and Bitcoin

The blockchain is, perhaps, the main technological innovation of Bitcoin. Bitcoin isn't regulated by a central authority. Instead, its users dictate and validate transactions when one person pays another for goods or services, eliminating the need for a third party to process or store payments. The completed transaction is publicly recorded into blocks and eventually into the blockchain, where it's verified and relayed by other Bitcoin users. On average, a new block is appended to the blockchain every 10 minutes, through mining [2].

Based on the Bitcoin protocol, the blockchain database is shared by all nodes participating in a system. Upon joining the network, each connected computer receives a copy of the blockchain, which has records and stands as proof of every transaction ever executed. Thus, it can provide insight about facts like how much value belonged to a particular address at any point in the past. Blockchain.info provides access to the entire Bitcoin blockchain.

3. Usecase

In this chapter it is aimed at explaining the scenario of the requirements presented in chapter 1. The descriptions of the use case is given below:

1. The director will be the one to upload the contract. He should thereafter be recognised as the director because he was the one to upload the contract.
2. The director would like the ability to upload any number of questions (which require a true or false response to). The director will upload each question, one at a time
3. The director would like the ability to add and remove shareholders from being able to vote and being able to see results for approved decisions at any point.
4. Each question can be answered by different number of shareholders
5. The director would like the ability to add and remove shareholders from being able to vote and being able to see results for approved decisions at any point.
6. Each shareholder may only vote for each decision once
7. The director should be able to close the voting process for a specific question. The majority result should then be able to be seen by all shareholders

4. Implementation

In this chapter it is aimed at explaining the implementation of the requirements presented in chapter 3. The programming language used for this project is Solidity which was implemented on Node JS. Below are explained all implemented functions, one by one.

1. Create three types (struct) named Shareholder, Voter, and Question.
    - **Shareholder** type represents an employee that can be assigned to answer question and can see the final decision after voting done. It contains one attribute called addr which describes the address of shareholder in a blockchain.
    - **Voter** represents a shareholder that will answer question which was assigned by director. Voter contains three attributes named addr, voted, and answer. Addr is the address of shareholder in blockchain, voted is a status whether shareholder already answered a question or not, and answer is the answer of voter for corresponding question.
    - **Question** is a type which represents the question. It contains four attributes namely id, description, isopen, and list of voters. Id describes the id of the question, description describes the question description, isopen describes a status whether the question is still open or close, and voters is the list of voters who will answer the question.
2. Define a public counter called **questionCounter** that keeps track of question number
3. Define a variable called **director** that owned the contract
4. Define a map called **questions** that stores list of questions
5. Define a map called **shareholders** that stores list of shareholders
6. Create a constructor that accepts list of shareholder addresses as parameter. Then iterate through addresses and store it in shareholder map. Set the msg.sender as director since he is the one that owns the contract.
7. Create a modifier called **OnlyDirector** which can be used to set restrictions on particular functions. In this case, no one can access the function except director.
8. Create a function called **createQuestion** that allows director to create a question
9. Create a function called **addVoterIntoQuestion** that allows director to add voter into question
10. Create a function called **removeVoterFromQuestion** that allows director to remove a voter from question
11. Create a function called **closeVoting** that allows director to close voting
12. Create a function called **vote** that allows voter to answer a question
13. Create a function called **majorityDecision** that allows director to see the result anytime and allow the shareholder to see the result after voting is closed.

The implementation of this contract can be found in file Survey.sol

5. Reference

[1] https://www.investopedia.com/terms/b/blockchain.asp accesed on December 18 2017
[2]https://blockgeeks.com/wp-content/uploads/2016/09/infographics0517-01-1.png accesed on December 18 2017