Politechnika Warszawska

WYDZIAŁ ELEKTRYCZNY

Instytut Elektrotechniki Teoretycznej
i Systemów Informacyjno-Pomiarowych
Zakład Elektrotechniki Teoretycznej
i Informatyki Stosowanej

Praca dyplomowa inżynierska

na kierunku Informatyka
w specjalności Inżynieria oprogramowania

Zdecentralizowana aplikacja do pożyczek na platformie Ethereum

Adam Kasperowicz
nr albumu 279046

promotor
dr hab. inż. Bartosz Sawicki

WARSZAWA 2018

TYTUŁ PRACY DYPLOMOWEJ

Streszczenie

Praca składa się z krótkiego wstępu jasno i wyczerpująco opisującego oraz uzasadniającego cel pracy, trzech rozdziałów (2-4) zawierających opis istniejących podobnych rozwiązań, komponentów rozpatrywanychjako kandydaci do tworzonego systemu i wreszcie zagadnień wydajności wirtualnych rozwiązań. Piąty rozdział to opis środowiska obejmujący opis konfiguracji środowiska oraz przykładowe ćwiczenia laboratoryjne. Ostatni rozdział pracy to opis możliwości dalszego rozwoju projektu.

Słowa kluczowe: praca dyplomowa, LaTeX, jakość

THESIS TITLE

Abstract

This thesis presents a novel way of using a novel algorithm to solve complex problems of filter design. In the first chapter the fundamentals of filter design are presented. The second chapter describes an original algorithm invented by the authors. Is is based on evolution strategy, but uses an original method of filter description similar to artificial neural network. In the third chapter the implementation of the algorithm in C programming language is presented. The fifth chapter contains results of tests which prove high efficiency and enormous accuracy of the program. Finally some posibilities of further development of the invented algoriths are proposed.

Keywords: thesis, LaTeX, quality

WARSZAWA, 1 lutego 2017

POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRYCZNY

OŚWIADCZENIE

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa inżynierska pt. Zdecentralizowana aplikacja do pożyczek na platformie Ethereum:

- została napisana przeze mnie samodzielnie,
- nie narusza niczyich praw autorskich,
- nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam, że przedłożona do obrony praca dyplomowa nie była wcześniej podstawą postępowania związanego z uzyskaniem dyplomu lub tytułu zawodowego w uczelni wyższej. Jestem świadom, że praca zawiera również rezultaty stanowiące własności intelektualne Politechniki Warszawskiej, które nie mogą być udostępniane innym osobom i instytucjom bez zgody Władz Wydziału Elektrycznego.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Adam Kasperowicz.....

Contents

Chapter 1

Introduction

The aim of the thesis will be a construction of a decentralized loan system using Ethereum technology. The final product should be a proof of concept of the possibilities of blockchain and smart contracts technologies. Additionally, the thesis should serve as an experiment, in which strengths and flaws of software architectures involving blockchain will be discovered.

1.1 Problem and solution

Loan business is a vital part of the modern financial world. Being one of the oldest financial mechanisms to exist it provides the consumers with required capital. Throughout the whole history it has always been an example of a centralized system. That is one central entity hoarding the money from different sources gets inquired about financing possibilities. Such entity is in power to decide upon who shall receive the funding and with what interest. It is also the burden of the entity to deal with any cases of loan repayment disobedience.

The are a few problems with this concept all of which stem from the design itself. Firstly, if one wants to capitalize on his spare money through this type of scheme he has to create an entire new company with all the legal requirements and an administrative burden. Secondly, all the processes such as storage of money, allocation of loans, interest serving, acceptance of clients etc. have to be taken care of by the workers of the loan company. Thus, costs are generated and sources of human errors and introduced. Lastly, due to the way financial services are diversified around the world it is often very troublesome or even impossible to service loans internationally.

All of the aforementioned disadvantages can be circumnavigated using the fruits of modern computer science. That is blockchain and tightly related

smart contracts. A Decentralized Application(DA) is introduced which serves as a medium between those who have the capital and those who need the capital. The DA which acts as a loan system allows anyone with any amount of Ethereum to bid a loan with an arbitrary duration and interest on a public exchange. At the same time a counterparty publishes an ask offer stating how big of a loan is required under specific duration and interest. The whole process closely resembles stock exchange. In result market forces lead to an equilibrium allowing both parties to reach their goal. The exchange itself is based on blockchain and no central server is required.

This design solves all of the issues pointed out so far. Anyone in the whole world with a connection to an internet is able to put his money to good use with few mouse clicks while maintaining anonymity. All of the operational activities are also immediately eradicated by market forces backed by smart contracts. Of course the problem of loan repayment still remains but the possible solutions of this specific mischief will be explained in the section describing a loan taker.

1.2 Blockchain

Blockchain is simply a chain of blocks. Where the function of a link is served by a hash. Each block is a record, in it's simplest form containing: cryptographic hash of the previous block, a timestamp, and some arbitrary transaction data. Blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks. By design, a blockchain is resistant to modification of any data inside of it. Once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks, which requires consensus of the network majority. Although blockchain records are not unalterable, blockchains may be considered secure by design.

To put it simply blockchain constitutes of 3 major technologies put together:

1. Peer-to-peer network: In it's simplest form blockchain is stored entirely by every user of the network. Just like torrent users spreading their data to every other user so that perfect replication of every data is accomplished. This way every user has the possibility to recreate every series of transactions that have happened on the network.
2. Assymetric cryptography: Users of blockchain communicate between themselves using their public and private keys. The algorithms used for key generation are often based on RSA or elliptic curves. This type

of communication assures us that no malicious manipulation during communication could happen.

3. Proof mechanism: The last element of the jigsaw puzzle is a solution to the double-spending problem. Solution proposed in Bitcoin is inscribed into the method blocks are created. First, a certain number off transactions started are gathered. Then, network participants start looking for 256 characters long hash which contains certain amount of zeros at the beginning. How this amount is determined will be described later. When found, a new block is created which contains the gathered transactions and is signed by the found hash. Only then are the transactions put into life. All of the contradicting transactions are also thrown out of the block. We can see that no malicious information can spread this way. The way this mechanism is implemented is also what differentiates most of the blockchain implementations present nowadays.

What is double-spending? Lets assume there are three participants in the network A, B and C. A sends two different transactions to B and C which can not happen at the same time. But B and C do not know that there are two transactions. At the moment of receiving message from A they only know of the message they received and thus happily follow with it. After some time, when all of the data propagates throughout the network, B and C learn that they have been scammed by A but can not do anything about it. A real life example could be A holding 100$ in bank and asking both B and C to buy a product for 100$. B and C will accept the payment because they lack information of the whole network. Hence the name double-spending problem.

Blockchain was invented by Satoshi Nakamoto in 2008 to serve as the public transaction ledger of the cryptocurrency bitcoin. [1] Is has since sparked an interest of thousands of software developers creating their own cryptocurrencies and developing many diverse branches of the technology. Most notably, a new financial microworld has grown around cryptocurrencies bringing fortune to many and financial ruin to even more people. What has happened to bitcoin price througout the last decade is often compared to modern tulipmania and can be clearly seen on the graph 1.1.

But financial prospects are not the aspects this thesis is concerned with. The most groundbreaking features the blockchain posses and which actually put it into the focus of modern computer science are anonymity and trustlessness.

- Anonymity: In the simplest implementations of blockchain such as bitcoin it is not possible to find any information of the user participating

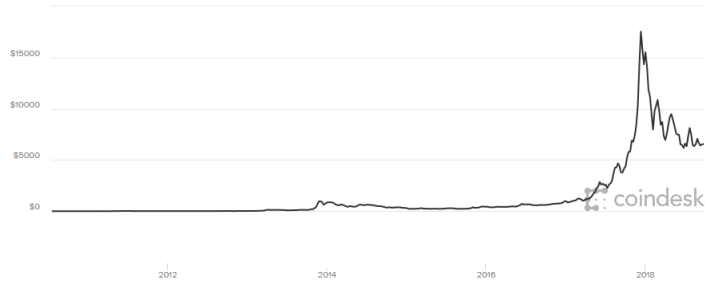

Figure 1.1: Bitcoin price over years

in the network. That is a simple consequence of relying purely on assymetric cryptography and no other requirement for allowing to interact with the network other than having a key pair. This simple feature has amazing implications in the real world. Every one in the world with internet connection can actually use blockchain. Coupled with the fact that cryptocurrencies can be used for value storage a way of escaping unstable national currencies has appeared for every human being. Recent example of that can be seen in Venezuela .[2] There are also people who have found ways to use the anonymity in an illegal way. Silk road, the greatest illegal drug webshop accessible by TOR network has experienced a renaissance every since bitcoin payments have been implemented. [3]

- Trustless: All of the currencies in our world are based on trust. The paper money has only a fraction of the value we assign to it. The numbers in our bank accounts have virtually no value. Currencies are simply are major agreement between all of the people on the world. In this world central parties are required such as banks which hold record of all transactions. We trust banks that when we ask them whether a person willing to buy from us has enough money, a true information will be sent to us. The double spending problem is solved this way. But blockchain has it's own mechanisms for this and thus allows us to omit the central party. We call the blockchain a trustless system because there is no trust required when interacting with other parties on the blockchain. Everything is taken care of by the computers, algorithms and protocols. The most grand example is Bitcoin itself which serves as a currency and a bank for all of the participants.

The element of blockchain technology that needs explanation is creation of a new block. As described earlier, in Bitcoin it is achieved by computing

hashes of random numbers until a hash with certain amount of zeros at the beginning is reached. The whole process is commonly referred to as mining. This amount of zeros is called *the difficulty*. The purpose of *the difficulty* is to simply hold the rate of blocks being created constant. That is lower the difficulty when there are few miners and the transactions would be processed very slowly or increase the difficulty when everyone starts using supercomputers for mining. Currently, the rate is adjusted every 2016 mined blocks. More exact information about this topic can be found on bitcoin wiki.[4] One important question that is left to be answered is: why do miners should bother with mining? Especially, why do people mine transactions which are not theirs? The answer is *reward* and *fees*. Currently, every mined block grants the miner who has done it some amount of bitcoin, that is a *reward* for mining. Additionally, people simply willing to use Bitcoin for transaction purposes can attach fee to their transactions so that miners will be more likely to mine those transactions first and receive the fee.

But looking for specific hashes is only one way of controlling block creation rate. PrimeCoin is especially worth noting as its method of mining is by finding prime numbers. Mining itself has also become a major part of computer hardware world due to computing power is has accumulated. Mining hashes is nowadays done only by GPUs due to parallelism benefits. There has been even created a special hardware called *ASIC* which exist solely for the purpose of mining hashes.

After the subject of mining has been explained we encounter another problem. We know that after a block has been mined it has to be propagated throughout the network as to be actually accepted by the network. What happens when in a large network two blocks are mined at the same time and both are able to propagate only through a part of the network before they collide? Then at least the majority of the miners (that is 51% of all computing power) has to decide which block to link to the blockchain and which to throw out. It is very important that a solid method of choosing the block by majority is used in a blockchain implementation because otherwise a malicious party could take over the network in some way or another by forcing his blocks with his transactions. It it also possible that the declined block will not be thrown away but will start living his own life in a sequence of events that are known as *forks*. What follows is that one blockchain is split into two smaller ones which live from then on their own lives. Most of the time forks happen because a new version of the blockchain implementation is being deployed and not everyone wants this new version. This situation has occured with Bitcoin multiple times and the whole subject is worth thesis on its own.

Possible topics to elaborate:

- Comparing blockchain to sql and nosql? (trustlessness, no scalability)
- What coins are there? (Bitcoin, Ethereum)
- When to use blockchain? (use cases)
- How to take over the network?
- Scalability solutions?

1.3 Ethereum

Ethereum is one of many implementations of blockchain technology. It differs from the many other ones in that it is as a next step in the evolution of blockchain. Learning from Bitcoin mistakes and trying to create a flawless general-purpose blockchain implementation, Vitalik Buterin together with other programmers have published a paper outlying desing for Ethereum. [5] The new cryptocurrency has been officially released in year 2015.

Owing to a plethora of new features, among which smart contracts are most notable, Ethereum has managed to become one of the most widely known and used cryptocurrencies in the world. It was the first to acquire support of real-world corporations. There are also actual companies serving products based on Ethereum. Being one of the cryptocurrencies it has also became a big part of the financial cryptoworld and has gone trough phases of big volatility, which can be seen on the graph 1.2

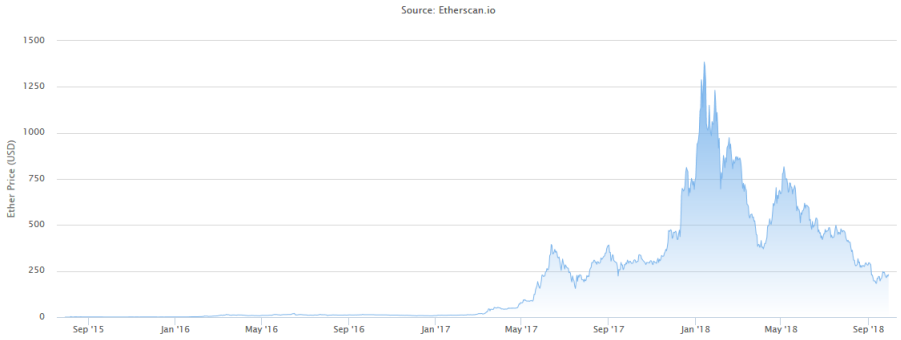

Figure 1.2: Ethereum price over years

The are a couple factors which make Ethereum unique and can be seen as an improvement over Bitcoin.

- Faster and more scalable than Bitcoin: Every cryptocurrenct has scalability issues which stem from the fact that theoretically every network

participant should posses the whole blockchain. But Ethereum team has gone one step ahead and started creating tweaks for this problem. While estimated peak performance of Bitcoin is that of seven transactions per second, Ethereum has managed to reach performance of twelve transactions per second. The most important fact tough is that constant research is being carried out which should find the solution for the scalability problem. So far the only discovered methods for solving this issue consist of sacrificing decentralization and thus security of the network.

- Proof-of-Stake: As explained earlier, when two conflicting blocks meet on Bitcoin blockchain, the majority of miners have to decide which block to accept. This way of solving conflicts is called Proof-of-Work. The problem with this approach of using miners in general is that a lot of power is being used by the hardware around the world to power the different blockchains. The amount of power being used is apparently comparable to energy consumption of some smaller countries. [6] Ethereum team is willing to change that by using Proof-of-Stake mechanism. This way the whole notion of mining is being eradicated. To create a new block a group of network participants is being asked to confirm transactions. The group has to be big enough so that the amount of coins they hold on specific blockchain sums up to 51% of all the coins present on this blockchain. What follows is that only a fraction of the energy will be consumed and the network is still secure because a person willing to take over the network would have to buy 51% of the coins on this blockchain. If we assume that market forces are working properly then the intruder probably will not have enough resources to carry out the attack. Nevertheless, it can be seen that we switch from one problem to another and users of Ethereum should be wary of that.

- Smart contracts: The single most important notion which differentiates Ethereum from all the other blockchain implementations is that of a smart contract. The general idea is that Ethereum blockchain has an Ethereum Virtual Machine(EVM) built into it. Which actually serves similar purpose as Java Virtual Machine for Java. If we put some lines of code of a language called Solidity into the transaction part of a block, the code will be executed after confirmation of this block. What smart contracts, EVM and solidity actually are will be explained in the next section. The most important fact is that thanks to this invention we are actually able to store not only static data in the blockchain but also code which will be executed. Thus, we acquire a fool-proof method of

storing and carrying out procedures.

Immediate result of this invention are what's called Decentralized Applications(DAs) and Decentralized Autonomous Organizations(DAOs). If we call a typical blockchain a trustless database then we could call Ethereum blockchain coupled with solidity code a trustless application. Referring to a bank example mentioned earlier which can be replaced by blockchain for storing money but nothing more, we can see that now we are able to replace a banking entity also in such aspects as periodic payments or loans. Actually any operation which required intermediary can be now replaced by a piece of code which will be used by both side of the contract. The only requirement is that the deal has to use Ethereum and has to be carried out on the Etehereum platform. Fulfilment of those tasks can be achieved by construction of a Decentralized Application.

If we go one step further we notice that is actually possible to create another cryptocurrency on top of Ethereum. Just as Ethereum is backed by a Proof-of-Work or a Proof-of-Stake we can create our own coins which will be backed by smart contracts, which in turn are backed by the Ethereum mechanisms. Those coins are most often called tokens. Finally, if a DA is created which facilitates it's own token we end up with a self living entity which trades its own money with it's own set of rules. That is what we call DAO.

As for the moment of writing this work there are almost two thousand DApps (more popular naming of a DA) serving purposes ranging from video games through social interactions up to financials. [7] Additionally, almost seven hundred tokens are being lively supported. [8]

Lastly, it is worthwhile to elaborate on the topic of cryptocurrency creation using smart contracts. The process goes by the name of Internal Coin Offering(ICO) which is taken straight from Initial Public Offering(IPO). IPO is simply what happens when company starts listing itself on the stock exchange. ICOs pursue similar position as they mostly go hand in hand with a deployment of a fresh DAO. The coin offerings have become a vital part of the cryptocurrencies world as they have managed to earn millions of dollars and have become a new way for companies around the world to gain funding. Another sign of blockchain technology binding itself stronger to the world of real life usage.

1.4 Smart contracts

Smart contracts are files of code written in programming language called Solidity which are stored on blockchain and executed by Ethereum Virtual

Machine(EVM). The specifications of the EVM will not be examined in this work as the material reaches outside of the aim of this thesis. Most of the information can found easily in the Ethereum yellow paper. [9] The notion of EVM will be touched upon in the last chapter where specific methods of code compilation will be described.

What needs explanation are the characteristics of Solidity language. It is an object-oriented, high-level language for implementing smart contracts. It's syntax has been influenced by C++, Python and JavaScript. Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features. Additionally, it is turing complete. It is very important to notice that blockchain serves overall a similar function to that of a database. But databases in general offer us SQL or SQL-like languages which are not turing complete without considerable extensions. Solidity in turn offers us a different perspective. We are able to experience a language which deals with directly with data storage but offers us the possibility to solve most computational problems at the same time. Of course SQL languages are declarative and operate on different types of data structures thus serve other purposes. But we can see that a new frontier has opened in the data storage world.

It is also important to understand the implications of programs running on blockchain. As every transaction done on blockchain has to be stored on blockchain it is also true that programs running on blockchain have to save their every change of the state on the blockchain. Thus the actions have to be confirmed by other users of the network and propagated. It is now possible to create program by a malicious user or by a bug which would spam the whole network and thus make it unusable. For example, an infinite loop in a smart contract would need constant confirmations and would not allow others to do anything with the network. To circumnavigate this problem a notion of *gas* is introduced. Every smart contract put on the blockchain has to have some amount of Ether assigned to it. Every line of code run on this contract consumes gas. Different operations cost different amount of gas. For example, read operations cost no gas but sending ether from contract to somewhere else costs a lot of gas. The gas amounts can be found easily on the internet. [10] The amount of gas respective operations consume is fixed. But the price of every gas unit denominated in Ether is subject to change. To run a line of code the contract has to pay product of amount of gas used and current gas price. The gas price is shaped by many factors but it is mostly set by miners. As a result the infinite loops would eventually run out of funds to consume and would stop. Moreover, usage of gas and not simply Ether lets those two values be decoupled. Thus a double increase of Ether value will not directly affect the gas price. Still, the gas price can change and

thus we can see that big smart contracts are discouraged as they can become expensive to use.

Just by this fact we can see that programming in Solidity is specific and requires careful considerations in many areas. Most notably those areas would be:

- Gas optimization: Besides cost of running code there also come costs of storing data. Because every variable or data structure used in a smart contract also comes with a cost. As a result we could notice a similarity to the old days of programming when computing power and more importantly storage space were scarce. A Solidity programmer also has to use a lot of tricks to keep his code small and fast. Unfortunately, the history of software development has shown that such practices often end up in a code that looks badly and is unmaintainable. Remedy for possible future catastrophes could be usage of open-source libraries which hold optimal fault-proof solutions, future developments in language which would abstract the gas optimization away and extensive testing.
- Security: Ethereum is all about money and the end purpose of Solidity is automation of money handling processes. An immediate consequence of those facts is that Ethereum world is a breeding ground of malicious users which will use every possible loop-hole in the system to steal the biggest possible amount of funds possible and then perish behind the veil of anonymity. The problem with stealing on blockchain is that once it is done it is done. If we are not able to connect the public address of the hacker with his real world identity then there is nothing we can really do. Moreover, the gas optimization practices stimulate creation of unsecure code. Fortunately, the issue has been noticed by the community and there are plenty of materials allowing for education in those matters. The documentation of Solidity being a good starting point. [11]
- Bugs: Solidity is one of the most costly language in the world when it comes to bugs. One bad mathematical operation could create a completely wrong address of transaction receiver and thus result in a transfer of millions of dollars to a dead account. It has happened before and is still happening. Testing is one of the most vital aspects of Solidity programming and an aim of 200% test coverage should not be treated as a joke. It is also a good practice to use private test blockchain networks where funds can be created out of thin air. Those networks can be used for integration testing and such. Ethereum itself offers

three world-wide networks besides the main one which only purpose is letting user test their contracts and letting Ethereum developers test new solutions. These networks are named Ropsten, RinkeBy and Kovan.

It is obligatory for the language to allow for a communication with an outside world. That is to allow for external parties to read the blockchain data and interact with the blockchain state. The task is achieved by a Remote Procedure Call(RPC) communication as well as by Inter Process Communication(IPC). In this work the RPC variant is being used and explained due to it's versatility. The messages used are simply JSONs. The whole API reference can be found on ethereum wiki. [12] Our client has to connect to an Ethereum node which serves and appropriate end-point through HTTP. As the API is quite simple a ready implementation of the communicator can be found in most of the widely used programming languages. The general standard of RPC messaging used by Ethereum goes by the name web3. As this project used Java, the library web3j was used. Still, web3.js for JavaScript or web3.py for Python could have been used as well.

Chapter 2

Current state of the technology

2.1 ETHLend

2.2 Kamboo Finance

2.3 DAI

2.4 Unchained Capital

2.5 SALT Lending

2.6 Othera

2.7 BitBond

2.8 BTCPOP

2.9 Credible Friends

Chapter 3

Software design

3.1 Loans

One loan offer on an exchange will consist of following parameters:

- **Basis**: Size of the loan denominated in Ethereum.
- **Duration**: The time after which the whole loan should be paid back together with interest. Depicted per predefined amount of days.
- **Interest**: Percentage of the loan basis which has to be additionally paid by loan taker. Calculated per predefined amount of days.
- **Collateral**: Information whether the loan is backed by a third party and has a low probability of going default. For example loan taker with no evidence to back his repayment probability will have only access to loans with collateral parameter being equal to None. At the same time loan taker with his loan being backed by a special bank agreement whereas there is a deposit with amount equal to the loan basis and interest incurred which is connected to a smart contract will have access to loans with collateral being equal to Yes.

The process of buying and selling loans will have the same characteristics as the one seen on the stock exchange.

That is loan ask whose size exceeds size of one respective loan bid will automatically cover the second identical loan bid. For example loan taker A asks for a loan of 1 ETH. There are two identical offers by loan provider B and C both equal to 0.5 ETH. The matching will automatically buy for A both loans from B and C.

Moreover, if loan taker asks for a loan of interest higher than the lowest present on the market he will be sold the loan with the lowest interest. For

example loan taker asks for a loan of interest 2%. There are loans on the market being bid with both interest of 2% and 1%. The loan of 1% will be sold to the loan taker.

In terms of repayments the loan will function as an amortized loans. That is, the repayments will be constant with principal amount being paid back increasing and the interest amount decreasing. The formula for computing the installments looks as follows:

$$C = \frac{rP}{1 - \frac{1}{(1+r)^n}}$$

Where C is monthly installment, r is interest rate per month, P is principal amount/basis and n is a duration of a loan in periods equal to predefined amount of days.

3.2 Types of users

The project provides for three types of possible actors. They are as follows:

- **Loan providers**: Any user with spare Ethereum and access to the application. Such person bids his loan offer and once the loan is sold automatic processes backed by smart contracts care for repayments.
- **Loan takers**: User who has fulfilled required information inquires and can be assured of that at least one of the methods of repayment can be exercised on him. The methods are:
  1. Regular individual refilling of his repayment account.
  2. Usage of a collateral deposit from a third party.
  3. Following legal procedures by bailiff of a country where the loan taker is located.
- **Authorities**: Third parties which for example serve the role of a collateral deposit or bailiff chasing the loan taker. This user has the possibility to refill the account of the loan takers repayment account.

3.3 Processes

3.4 Requirements

The complete system should provide following functions:

1. Loan providers and loan takers should be able to:
    - check and transfer funds to and from their accounts using external wallet based on Ethereum network.
    - specify the following parameters in their loan bids and asks: Basis, Duration, Interest, whether it has collateral.
    - check their open bids and asks as well as the whole public exchange.
    - cancel all of their open bids and asks.
2. Loan takers should be able to:
    - specify legal informations which could be used by bailiff if repayment did not happen. At the same time the information should remain confidential so long there is no need to use them.
    - bind his account with an account of a third party. The additional account will always hold a sum required to repay the whole loan when needed.
    - repay the loan earlier.
3. Every user should be able to:
    - access the application having only internet connection.
    - maintain his anonymity while using the system.

3.5 Architecture

3.6 Visuals

Chapter 4

Software implementation

4.1 Ethereum network

What is Geth? What is ganache-cli?

4.2 Solidity

How to compile? What is truffle? Math without decimals and negative numbers? How to compute exponential?

Safety

How to transfer funds? What is condition-computation-action? What are public operators?

Tests

Js tests? Solidity tests?

4.3 Java

How to compile?

Bibliography

[1] Satoshi Nakamoto, „Bitcoin: A Peer-to-Peer Electronic Cash System", https://bitcoin.org/bitcoin.pdf

[2] „How Venezuela Came to Be One of the Biggest Markets for Crypto in the World", https://cointelegraph.com/news/how-venezuela-came-to-be-one-of-the-biggest-markets-for-crypto-in-the-worl

[3] „The Effect Of Silk Road On Bitcoin And Tor", https://silkroaddrugs.org/the-effect-of-silk-road-on-bitcoin-and-tor/

[4] https://en.bitcoin.it/wiki/Difficulty

[5] https://github.com/ethereum/wiki/wiki/White-Paper

[6] https://digiconomist.net/bitcoin-energy-consumption

[7] https://www.stateofthedapps.com/

[8] https://etherscan.io/tokens

[9] https://gavwood.com/paper.pdf

[10] https://github.com/djrtwo/evm-opcode-gas-costs

[11] https://solidity.readthedocs.io/en/v0.4.24/security-considerations.html

[12] https://github.com/ethereum/wiki/wiki/JSON-RPC