SUCCEEDING PROGRESS AND THE ARTIFICIAL CYBERSPACE'S FUTURE IN BLOCKCHAIN

Basamma Umesh Patil[1], D V Ashoka[2]

[1] Research Scholar, Department of Computer Science and Engineering, JSS Academy of Technical Education (affiliated to VTU), Bengaluru, India; Email: bupatil25@gmail.com

[1] Assistant Professor, Department of Computer Science and Engineering, SJB Institute of Technology (affiliated to VTU), Bengaluru, India;

[2] Professor and Dean, Department of Information Science and Engineering, JSS Academy of Technical Education (affiliated to VTU), Bengaluru, India; Email: dr.dvashoka@gmail.com

Abstract:

Block chain became a buzzword in 2017, and its incorrect use resulted in a false picture, owing to societal misinformation and ignorance about the emerging technology. Since block chain is still a relatively new concept, the bulk of the details and knowledge for this analysis came from hours of online testing. Despite the availability of online databases, the details collected required to be double-checked with other outlets since some of the data was inaccurate. On the technological side of this research, books on programming in the Solidity language as well as online tutorials were looked at. The codes used in this analysis were created to illustrate the technicalities of Blockchain in an easy-to-understand manner. Data from different datasets like human aesthetics, visual images of the dance and user scores are fused together for evaluation of the dance performance. Therefore, the proposed work adopts block chain technology for fusing data. The study demonstrates that Block Chain has the ability to become a significant part of our lives by enhancing data storage and offering community more influence. Only the most important facets and characteristics of Block chain were studied during the research.

Keywords: Block chain, Bitcoin, Double check, Ethereum, Decentralized, Ledger.

DOI Number: 10.14704/nq.2022.20.9.NQ44600 Neuro Quantology 2022; 20(9):5192-5205

1. Introduction

There are many revolutionary innovations and developments that have improved our society and our lives, some of which date back thousands of years. From the development of the wheel in 3500 BC, to the Sumerians' introduction of written language in 3300 BC, to Edison's light bulb, to the more modern developments such as the machine and, without a doubt, the most powerful and popular of all time, the Internet, This game-changing technology, first released in 1990, built an integrated interactive reality where people from all over the world could interact and communicate with one another. Many people believe the Blockchain would have a similar effect in the not-too-distant future. During 2017, the term "Blockchain" was misused on many occasions, and disinformation about the issue resulted in unparalleled situations, such as an Ice Tea firm formerly known as Long Island Iced Tea Corp. that changed its brand name to "Long Blockchain Corp." and saw a 500 percent rise in its own shares, despite the fact that the company has little link to Blockchain. People were simply purchasing stocks because they were afraid of losing out on anything important. The change of name coincided with the meteoric rise of a well-known phenomenon known as Bitcoin.

The cryptocurrencies Bitcoin and Ethereum have already been linked to Blockchain, which is partially true, however it is the key explanation that everything relevant to Blockchain receives the most attention from the public. With Bitcoin hitting incredible highs in 2017, several people were searching for a simple way to make a small or large financial return and, potentially, achieve the much-desired financial independence.

Anything that hinted at proposals to use Blockchain drew media interest, even though the public had no idea what the technology was about. I've been watching the growth of Blockchain over the past couple of years, but that was before the rise of the well-known cryptocurrency Bitcoin. I decided to investigate and learn more about this subject after seeing so much curiosity in the technology, as well as some incorrect meanings and associations of the term Blockchain. The aim of this research is to thoroughly comprehend and assess if this technology can really be called one of the most significant developments of recent years, or whether it is just a major hoax that will lead nowhere.

Block Chain Technology

1.1 History of block chain

The term Blockchain is often used interchangeably for Bitcoin (BTC), the well-known cryptocurrency that hit an all-time high unit value of $19,783.06 in December 2017. Morris, David Z. (David Z. Morris, 2017) Despite this, or at least in part, this definition is inaccurate. Blockchain is the cutting-edge principle that underpins a slew of cryptocurrencies, including Bitcoin (BTC) and Ethereum (ETH) (ETH). (2017) (Tahi Gichigi) (Tahi Gichigi) (Tahi Gichigi) Blockchain is a technology that can be used with a range of services, like Bitcoin and cryptocurrency in general. Whitefield Diffie and Martin Hellman released a paper named "New Directions in Cryptography" in 1976, which was the first proposal or move towards what Blockchain is today. Whitefield Diffie and Martin E. Hellman (Whitefield Diffie and Martin E. Hellman, 1976) The very first paper released was New Direction of Cryptography, which presented the principle of public key encryption, which is still found in Blockchain technology. Nonetheless, Stuart Haber and Scott Stornetta released a whitepaper titled "How to time-stamp a digital text" in 1991, which was the first functional definition of Blockchain. W. Scott Stornetta, Stuart Haber, W. Scott Stornetta, Stuart Haber, Stuart Haber, Stuart Haber, Stuart Haber, Stuart Haber

1.2 Blockchain – What is a Block

Each block can be thought of as a container for any kind of data, including timestamps, metadata, and transactions. The data structure saved in each block is determined by the blockchain form available (from currencies to certificates or any sort of value). On Blockchain, a block is approximately 1MB in size. (2019, Block-chain). The information in the block is secured and includes information about both parties involved in the deal. The encrypted identities of sender and recipient, for example, will be included in a BTC block. After that, each block includes a hash that is calculated mathematically. The hash is important because it is different every time you use it. A virtual biometric asset, such as fingerprints or DNA, may be linked to the hash. As seen in Figure 1, each block also contains the previous block's hash code.

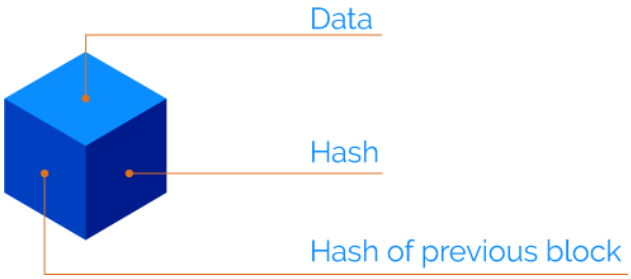

Fig. 1 Block Content.

Since each block has the hash of the previously created block, as seen in Picture 2, modifying data in a block is difficult. This is due to the fact that if a transition is made, the hash changes as well. However, this has an adverse effect on the whole chain, rendering it useless. Tania H. (2018, Tania H., Tania H., Tania H., T

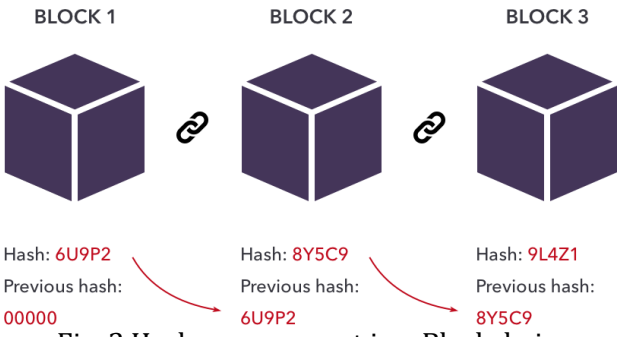

Fig. 2 Hash management in a Blockchain.

1.3 Proof of work - Simultaneous solutions to the same problem

Several devices on the network may operate together to solve the puzzle of a particular problem at the same time. This may also indicate that the mystery is being solved concurrently. As this occurs, and two or three valid solutions are

sought, the chain splits, creating the so-called "fork," as seen in Figure 3.

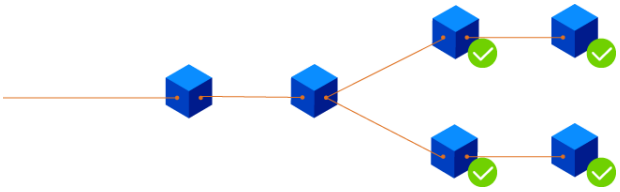

Fig. 3 Hard fork in Blockchain.

The longer of the two chains becomes the real chain as soon as a new block is attached to one of them. As seen in Figure 4, all blocks in the other chain will be discarded and will need to be checked again. The longest chain ever made, according to the official Block- chain website, is 5 consecutive blocks (Block- chain – Chain's duration, 2019).

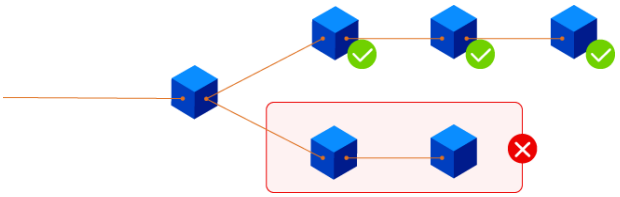

Fig. 4 Hard fork – verification.

1.4 Blockchain – Wallet

The blockchain also has a function named Wallet, which is designed specifically for cryptocurrencies. The Wallet is a piece of software that allows you to access your cryptocurrencies. Transactions are saved on the blockchain when using the wallet. In simple terms, the owner of a wallet has a blockchain wallet address to which users can transfer cryptocurrencies. The idea of a blockchain wallet is close to that of the well-known PayPal app, with the exception that cryptocurrency transfers can be managed rather than processed. To comprehend how a blockchain wallet functions, we must first comprehend the concepts of private and public keys.

1.4.1 Wallet – Private and Public Key

When a person has a Blockchain wallet, he or she is simultaneously given a private and public key (see Picture 5). Comparing the blockchain wallet to how email operates is the best way to explain what private and public keys are.

Your email address is needed if anyone wishes to give you an email. That is the public key, or the address, in the blockchain wallet. Instead, the private key is the password you use to access your email account. The public key should be exchanged with anybody and there isn't anything that can be achieved with only that knowledge apart from submitting cryptocurrencies to it. The secret key, on the other hand, should be held secure and never exchanged with others. (Maria Wachal, 2018).

BLOCKCHAIN – FIELDS WHERE IT IS CURRENTLY USED

1.5 BLOCKCHAIN – Bitcoin (BTC)

Blockchain is a very new concept, although most people are unaware of how it has been used in various fields and industries. It might come as a surprise, but Blockchain is used in a wide range of industries, from finance to cyber-security, government, and healthcare, to name a few. Without a question, Bitcoin is the most well-known use of the technology. The term "blockchain" became popular after Bitcoin reached its peak in value in 2017, particularly during that year.

Fig. 5 Wallet – Private and Public Keys.

Bitcoin was unavoidably the catalyst for widespread mainstream involvement in Blockchain technologies. Everyone has never understood why Bitcoin has become so successful, particularly in 2017, when it reached its highest unit price ever. Thousands of investors bought Bitcoin just because they heard it might be worth a lot more at any time in the future. The fact is that the vast majority of those attempting to board this train, metaphorically speaking, were either unaware of Bitcoin or had little or no knowledge of what it was. I'll confess that I was one of them when Bitcoin was first gaining popularity. This is also why my interest in Blockchain increased, and I wanted to learn more about this new technology that has the potential

to become the second most important innovation in modern history after the Internet.

Bitcoin is the only decentralised blockchain that has ever developed. The Bitcoin.org domain was registered at anonymouusspseech.com on August 18, 2008, which is a platform that allows users to register domains privately and thereby without leaving any trace. Anthony Lewis (Anthony Lewis, 2018). The white paper of Bitcoin is written by an individual or group of people under the alias Satoshi Nakamoto on October 31st of the same year. A white paper is a text that describes the characteristics of a good or service and reflects on them. White papers are typically used as publicity reports, and certain businesses, such as Microsoft, use them to sell their goods before they go on sale. "A strictly peer-to-peer version of electronic cash will enable online transfers to be transferred directly from one party to another without passing through a financial institution," according to Bitcoin's white paper. (2008, Satoshi Nakamoto).

In both the United States and the rest of the planet, 2008 was a financially disastrous year. Lehman Brothers, for example, went bankrupt with 639 billion dollars in cash and 619 billion dollars in loans, making it the largest bankruptcy in history. (2017, by Nick K. Lioudis) Satoshi made a subliminal connection to rotten financial institutions (banks) in the release of the Bitcoin white paper, offering a workaround that does not involve a third party and prevents future frauds such as the ones that afflicted millions of innocent citizens in 2008.

Satoshi states in the white paper that Bitcoin is a peer-to-peer network (P2P) that passes control to the users who are part of the network, without the need for a central entity to administer the protocol. As a result, Bitcoin is often referred to as a decentralised framework. Any activity that takes place on the network is instantly recorded on the public ledger (network) and therefore accessible to all; nothing occurs behind closed doors. Bitcoin, according to the white paper, fixes the problem of double-spending by using Blockchain technologies. As the phrase implies, double-spending refers to spending the same amount of money (coin) twice.

We may use a common case as an illustration to further explain this definition. When we go to the store for our weekly groceries, we pay with money, real money, at the cash register to buy the things we have in our kart and to own them. How will we do the same in a digital world? Since it is simpler to duplicate digital currency, it can be used several times. Bitcoin addresses this issue because it is backed by Blockchain technology, which, as previously stated in the chapters, maintains records of all transactions on the system and groups them into blocks that are incredibly difficult to tamper with. Of block is then hashed, and Bitcoin uses encryption to hold all of the details out of the hands of hackers.

The Genesis Block was mined at 18:15:05 GMT on January 3rd, 2009, and was the first Bitcoin block ever created. (Ben Brown, 2019) (Ben Brown, 2019) (Ben Brown, 2019). The Genesis stone, in my opinion, is a scathing critique of financial institutions' treatment of the current economic condition. The Genesis block contained a message that alluded to the headline from the British The Times newspaper on January 3rd, which read, "Chancellor on verge of second bank bailout." (2019, Sead Fadilpai)

This is unmistakably a warning about financial firms' poor management, which, rather than serving citizens, ended up destroying their lives. With Bitcoin, Nakamoto behaved like a modern Robin Hood, transferring control to ordinary citizens by not just providing financial independence but also protecting their anonymity, as personal data is not tracked in any way with Bitcoin.

Bitcoin has had its ups and downs since its first publication and transaction, putting its legitimacy and integrity in jeopardy. The idea that personal data is not recorded can seem to be a positive aspect that safeguards people's privacy, but this may be a double-edged sword. Ross Ulbricht, for example, established a website named Silk Road in 2011. Silk Road was an online auction close to eBay, but that instead of paying via PayPal, anonymous customers could make purchases with Bitcoin, which provided them with an extra layer of anonymity. As a result, the majority of Silk Road's sales were for narcotics or other illicit goods such as weapons, which Ulbricht had originally classified as banned objects among several others but was ultimately powerless to regulate. This "internet Wild West," as it was dubbed, lasted just a few years, but it was estimated that over a billion dollars were traded during that period. The FBI finally shut down the platform, and Ulbricht was charged

with a number of offences. (2018, by An- drew Norry)

BLOCKCHAIN – IBM and WALMART: A revolution in the food supply

The world of cryptocurrencies, especially Bitcoin, has undoubtedly increased the popularity of the term "blockchain." Nonetheless, over the years, Blockchain has evolved and discovered innovative and diverse ways of adaptations to old structures, revolutionising them in a very powerful manner. One of the main characteristics that distinguishes Blockchain as a transformative technology is its lack of trustworthiness. The knowledge accessible on the Blockchain is authentic; no one can alter it because of the immutability of the records. Blockchain is built on the principles of openness and transparency. There is no one in charge, which raises the level of confidence between the parties.

IBM, the world's most well-known international information management corporation, has actively invested in this technology and already provides a wide range of Block-chain-based applications to businesses. One of the most intriguing projects IBM is working on is with Walmart, another mega-corporation in the supermarket industry. The two multinationals have begun a partnership with the aim of shedding fresh eyes on food protection and revolutionising the supply chain.

Food protection has long been a big issue, and the amount of incidents of tainted food or scams has had an effect on our environment, often resulting in financial losses. The Food and Drug Administration (FDA) confirmed an outbreak linked to fresh spinaches in the United States in 2006. (FDA). Tom Karst (Tom Karst, 2016) A bacterium named E. Coli caused an epidemic that infected 199 individuals, three of whom died and 31 of whom developed kidney failure as a result of the infection. The disease afflicted 26 states in the United States, and local officials were unable to determine the source of the spinach for two weeks. Following extensive inquiries into the event, it was discovered that the root cause of this massive outbreak that resulted in deaths was one farm's harvest of one lot from only one day. When you think about it, it's a little shocking (in a bad way) that it took too long to find out where the infection came from.

Another related incident occurred in Europe in 2013, where horse meat was discovered in what was meant to be beef or pork meat. Horse meat was found mixed in with other forms of meat in large quantities. A significant danger to food protection. Any of which is due to the fact that the food supply chain is not fully transparent. An egg scandal in the United Kingdom in 2017 involved approximately 700.000 eggs and terrified the whole world. (2017) (Katie Morley)

Many of the groups participating in the new food supply chain model, such as producers, distributors, retailers, packagers, among others, have their own internal monitoring or control scheme, some of them are still on paper, and the majority of them are incompatible with one another. There is a missing connection that will enable the knowledge to be continued throughout the whole phase from beginning to end. Here, Blockchain enters the picture as a game-changing technology, designed to enable us to track down the origins of any commodity we purchase or consume. Both those concerned work together to keep the database up to date and make the material accessible to everyone in the system, ensuring that everyone is informed of every component of the product.

This revolutionary scheme, which takes advantage of both traceability and clarity, allows users to know precisely where their product is in its life cycle. Within minutes, if not seconds, information such as where this commodity comes from, how it got to the market, and when it got there is accessible. Each team, from the farmer to the retailer, is obligated to track the products they treat, and the buyer will get the details they need by simply scanning the product (Fig. 6)

Fig. 6 Walmart – Blockchain to trace food.

2. BLOCKCHAIN AND SMART CONTRACTS

4.1 Contracts that are intelligent

Smart Contracts are another aspect of the Blockchain technology. As the name implies, a smart contract is a piece of software or an algorithm that is saved on a Blockchain, most generally the Ethereum Blockchain, and implemented after those conditions are met. The IBM blog has a good explanation of the term. (2018) (Nigel Gopie) Smart Contracts may be used to trade everything important, including real estate, automobiles, and currency.

We may equate the idea of a Smart Contract to the buying of a house to simplify it. A house purchase in the real world involves many parties: the consumer, the seller, the bank, and potentially other individuals. Furthermore, the whole procedure of finalising the house acquisition requires several lengthy and complicated measures. In this scenario, smart contracts make things a lot smoother and faster. The premise behind a Smart Contract is "IF this or WHEN this certain criterion/criteria is/are reached, THEN do/process." The Smart Contract executes itself immediately until the condition or conditions are reached.

The previous example of buying a home can be converted into a Smart Contract. Without being too detailed at this stage, the Smart Contract will serve as a contract between two individuals, and the purchase will be executed and concluded Whether or Until a certain condition is reached. Before the Smart Contract is implemented on the Blockchain, the principles and requirements of the Smart Contract must be determined and settled upon; but, after this is finished, the code and rules of the Smart Contract are unchangeable. No one would be willing to change that and it is impossible to do so on the Blockchain (hard to hack as explained in the chapters before).

There are a host of drawbacks to using Smart Contracts. Smart Con- tracts are, first and foremost, secure. Since the information is safe, tampering with it is difficult. If someone was to modify something in one block, the rest of the chain would have to be updated as well, as described in the chapter "Blockchain – What is a Block." Hacking the Blockchain has been almost difficult as a result of this. Another benefit of Smart Contracts is that they will save you a significant amount of time and resources. Smart Contracts eliminate the need for any of the stakeholders usually interested with a home transaction, for example. Furthermore, both paper work and time spent filling out paperwork has been eliminated, resulting in lower overall costs. Smart Contracts have an environmental effect as well, since they are paperless and everything is documented on the Blockchain.

2.1.1.1 **Decentralized Application Using Smart Contracts (DApp)**

The official documentation of Solidity online is a must-read guideline for those interested in learning how Smart Contracts are designed and implemented on the Ethereum Blockchain, as I am. Solidity is the programming language that Smart Contracts are written in. Solidity bears a striking resemblance to JavaScript. If a smart contract is used in conjunction with a frontend programme, the result is known as a Decentralized Application, or DApp. The DAPP's back-end, or context code, is a Smart Contract, which runs on the Blockchain and is therefore decentralised and accessible to anyone on the network. While we speak about deploying the DApp on the blockchain later, this point would be critical.

Understanding how a Web Application functions is a better way to learn what distinguishes a DApp from a regular Web Ap- plication. A consumer typically communicates with a Web Application via a window, and all of the details, code, and application itself are stored on a centralised server. The distinction between a Web application and a DApp can already be seen here. Although the code of a DApp is exchanged on the network and therefore unchangeable, the code of a Web App may be hacked since the whole programme is located in a single location. In terms of confidentiality, a DApp has a far higher degree of protection than a standard Web application. The fact that it is decentralised often means that corruption is far less prevalent, almost non-existent. Let's presume you want to build an online survey in which users will cast their votes. How many times have we learnt of tainted election processes, voting fraud, computer manipulation, and so on? All of this would be much more straightforward and trustworthy if it were migrated to a shared web structure where the data and procedure itself could never be modified. So, how does a DApp framework perform in practise?

Dependencies in DApp 2.1.2

From here on out, we'll get through the technical details of what DApps and Smart Contracts are and how they operate. Various books and documentations are now accessible to acquire the information needed to begin with simple coding. I used two key tools for the DApp illustration I'll be looking at in this paper: DApp University (Dapp University, 2019), a free collection of videos accessible on YouTube, and Building Games with Ethereum Smart Contracts - Intermediate Projects for Solidity Developers, a novel. Kedar Iyer and Chris Dannen (2018, Kedar Iyer and Chris Dannen)

The following is the concept behind the decentralised application: Consider the following scenario: A new season of a famous talent show is currently airing, and we are nearing the end. Different talents will perform against one another, and the winner of the Talent Show will be chosen by the audience at home. Normally, citizens can text their votes, dial a toll-free number, or use a mobile app to deliver their votes from home. Both votes are sent to a central server, which processes and manages them. The first issue is as follows: How would we be certain that the votes are legitimate? How do we be sure they can't be used against us? Each of the competitors may have enlisted the help of an unnamed firm to garner tens of thousands of votes. While it seems to be a far-fetched possibility, it is possible. The DApp's primary goal is to demonstrate how the framework should avoid both of these issues and have a more structured atmosphere.

First and foremost, we must be able to create our decentralised programme. We'll need to set up the construction framework, as well as the dependencies, in order to accomplish this. There are a variety of ways to create a DApp, but we will use the following for our application:

- Sublime Text 3.0 which is a text editor for coding. In Sublime we will code the Smart Contract using Solidity language.
- Ganache which is a software that simulates a blockchain locally in the background with fictious accounts having 100 ETH each (see Picture 7). The Ethereum are not real and are worthless on the real Ethereum block- chain. The 10 accounts offered by Ganache will represent the people that from home will express their winner. One account (address) will be one person as shown below:
- MetaMask which is a plug-in for Google Chrome. This plug-in essentially allows us to connect to the local blockchain provided by Ganache. Simply put, it turns Google Chrome into a Blockchain browser.
- Node.js which is a JavaScript runtime that includes the Google Chrome V8.
- Truffle which is the framework that will grant us the possibility to create the decentralized app on the Ethereum Blockchain. With truffle we can write the Smart Contracts, test them and eventually deploy them to the blockchain.

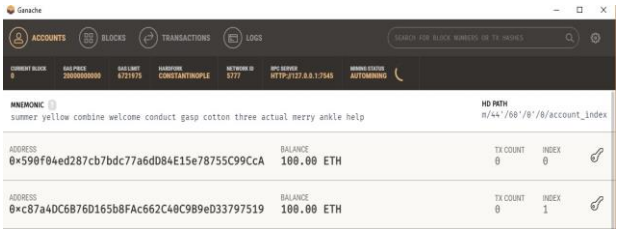

Fig. 7 Accounts in Ganache, local blockchain. Addresses are unique.

After all the dependencies are installed on the machine, the decentralized application can be created.

2.1 DApp – The Talent Show DApp

As first step, once created the folder directory for the application to develop, we can unbox the pet-shop, as shown in Picture 8. The pet-shop box is essentially a package that truffle makes available to developers to start the process more quickly.

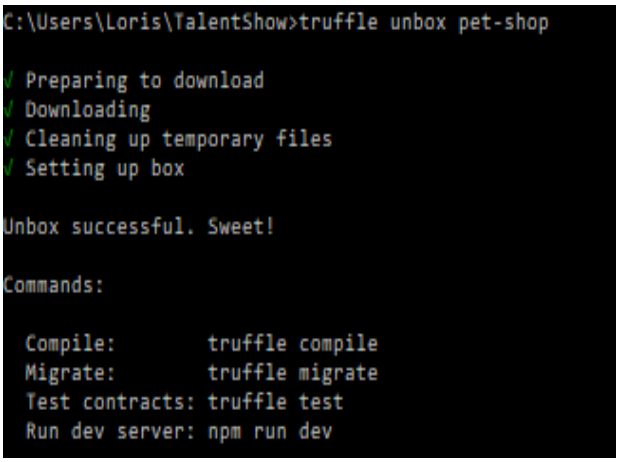

Fig. 8 Unboxing the pet-shop box into the Talent Show project directory.

2.2 DApp – Smart Contract – TalentShow.sol

The first thing to build is going to be the Smart Contract that will be deployed on the blockchain. As shown in picture 19, a .sol file was created. TalentShow.sol is the Smart Contract for our DApp. The source code for the Smart Contract can be seen in Picture 10.

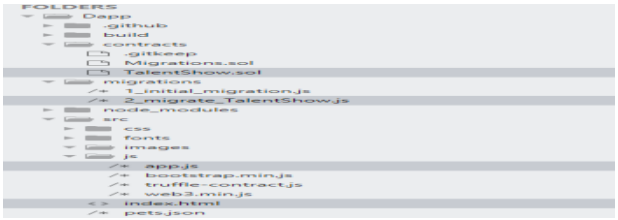

Fig. 9 Sublime Text 3– Talent Show DApp - Folders and files.

```
1  pragma solidity ^0.5.0;
2
3  contract TalentShow {
4
5      struct Talent {
6          uint id;
7          string talentName;
8          string talentCity;
9          string specialty;
10         uint numberOfVotes;
11     }
12
13     mapping(uint => Talent) public talents;
14     //Mapping of the accounts that have voted
15     mapping(address => bool) public addressesVoting;
16
17     event voteSentToBlockchain (
18         uint indexed _talentID
19     );
20
21     uint public talentsCount;
22
23     constructor () public {
24         createNewTalent("Antti", "Tampere", "Clown");
25         createNewTalent("Loris", "Nokia", "BodyBuilder");
26         createNewTalent("Joey", "Iowa", "Drummer");
27         createNewTalent("Alex", "Tokyo", "Sushi Master");
28     }
29
30     function addYourVote (uint _talentID) public {
31
32         //Check that the account in use voting has not done it already
33         require(!addressesVoting[msg.sender]);
34         //Restricting only to vote the talents available
35         require(_talentID > 0 && _talentID <= talentsCount);
36         //Setting the account as already voted, using the mapping addressesVoting
37         addressesVoting[msg.sender] = true;
38
39
40         //Adding 1 vote for the Talent Selected
41         talents[_talentID].numberOfVotes ++;
42
43         //Adding the vote to the Blockchain and registered
44         emit voteSentToBlockchain(_talentID);
45
46     }
47
48     function createNewTalent(string memory _name, string memory _city, string memory _speciality) private {
49         talentsCount++;
50         talents[talentsCount] = Talent(talentsCount, _name, _city, _speciality, 0);
51
52     }
53
54 }
```

Fig. 10 TalentShow.sol source code.

The programming language used for this contract is Solidity and more precisely version 0.5.0 as specified in row 1. The contract TalentShow is divided into 5 sections:

- A Struct
- Mapping for talents and votes
- An event that will register the vote
- A constructor
- Functions to add new talents to the list and vote

The Struct in our case is seen at row 5 throughout row 11. The Struct is essentially the “structure” of how each element/record in our system will be composed. Each talent will have an index (id), a name (talentName), a city or origin (talentCity), a specialty (specialty) and a number of votes (number of Votes). The mappings at rows 13 and 15 are meant to keep track of who are the talents registered in our system (talents) and to register which account from home has voted (addresses Voting). From row 23 to 28, we have the constructor where we essentially specify the talents available in the application. Inside the constructor we call the function createNewTalent which takes in as parameters name of the talent, the city of origin and the talent’s specialty as shown in Picture 11:

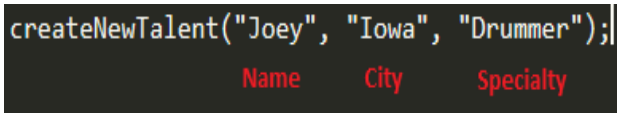

Fig. 11 Parameters passed to createNewTalent function.

Rows 48 to 52 include the createNewTalent feature, which takes the three parameters passed to it while calling the function from the function Object() { [native code] }. The talentCount counter is expanded by one at row 49. Row 21 was where the counter was set to zero. This variable's main function is to generate a specific ID that will be connected with the freshly created talent. The counter is systematically incremented by one every time the feature createNewTalent is called. The talent's ID would then be developed. The first will be number one, the second will be number two, and so on. We go on to the mapping named "talents" in row 50, which is the same as the mapping called "young talent" in row 13. The struct "Talent" is transferred to the mapping along with the required parameters: id, name, area, specialisation, and the number of votes, which will start at zero.

The other available feature of this contract is “addYourVote,” which is visible from row 30 to row 46. The main purpose of this feature is to see whether the account/address used to log into the app has already been entered in the mapping addressesVoting. This search is performed at row 33, where we demand that the voting account not already exist in the mapping addressesVoting. Using "msg.sender," you will get the account. If the account is not already mapped in our database, it ensures that this account is qualified to vote. Furthermore, we require that the talent

id chosen is currently active. At row 35, this is reviewed.

The next move is to apply the voting account to the mapping addresses Voting, so that if the same account is logging in the next time this feature is called, it would not be possible to continue, since this is verified at row 33 again. After that, at row 41, the amount of votes for the particular talent chosen by the user would be increased by one. By invoking the event "voteSentToBlockchain" at row 44, the final move is to log this event on the blockchain.

DApp – Smart Contract – App.js file

```
App = {
  web3Provider: null,
  contracts: {},
  account: '0x0',

  init: function() {
    return App.initWeb3();
  },

  initWeb3: function() {
    if (typeof web3 !== 'undefined') {
      App.web3Provider = web3.currentProvider;
      web3 = new Web3(web3.currentProvider);
    } else {
      App.web3Provider = new Web3.providers.HttpProvider('http://localhost:7545');
      web3 = new Web3(App.web3Provider);
    }
    return App.createContract();
  },

  createContract: function() {
    $.getJSON("TalentShow.json", function(TalentShow) {
      App.contracts.TalentShow = TruffleContract(TalentShow);
      App.contracts.TalentShow.setProvider(App.web3Provider);
      App.checkIfVoted();
      return App.render();
    });
  },

  checkIfVoted: function() {
      App.contracts.TalentShow.deployed().then(function(i) {
        i.voteSentToBlockchain({}, {
            fromBlock: 0,
            toBlock: 'latest'
        }).watch(function(error, event) {
            console.log("someone voted", event)
            App.render();
        });
      });
  },
```

Fig. 12 app.js source code part 1.

The app.js is written in JavaScript and it is the code that interacts with the smart contract TalentShow.sol and the HTML page index.html (Picture 12).

```
  render: function() {
    var talentShowInstance;
    var loader = $("#loader");
    var content = $("#content");
    var votedAlready = $("#voted");

    loader.show();
    content.hide();
    votedAlready.hide();

    // Load account data
    web3.eth.getCoinbase(function(err, account) {
      if (err === null) {
        App.account = account;
        $("#accountAddress").html("currently logged as: " + account);
      }
    });

    // Load contract data
    App.contracts.TalentShow.deployed().then(function(i) {
      talentShowInstance = i;
      return talentShowInstance.talentsCount();
    }).then(function(talentsCount) {
      var talentsResults = $("#talentsResults");
      talentsResults.empty();

      var talentsSelect = $("#talentsSelect");
      talentsSelect.empty();

      for (var i = 1; i <= talentsCount; i++) {
        talentShowInstance.talents(i).then(function(talent) {
          var id = talent[0];
          var name = talent[1];
          var city = talent[2];
          var specialty = talent[3];
          var votes = talent[4];


          var talentsCollection = "<tr><td style= 'text-align: center;'>" + id + "</td><td style= '
          talentsResults.append(talentsCollection);

          var possibleTalent = "<option value='" + id + "' >" + name + "</ option>"
          talentsSelect.append(possibleTalent);

        });
      }
      return talentShowInstance.addressesVoting(App.account);
    }).then(function(hasVoted){
      if(hasVoted){
          $('form').hide();
          votedAlready.show();
      }
      loader.hide();
      content.show();
    }).catch(function(error) {
      console.warn(error);
```

Fig. 13 app.js source code – render function.

The first element of the app.js file is the source code seen above. The contract TalentShow is generated in this piece of code, which is achieved inside the function initWeb3 where the link to localhost is formed and the function createContract is invoked as the final stage, row 18. Aside from creating a truffle contract, another feature checkIfVoted is called at row 26 inside the function createContract. The aim of this feature is to record the event of voting by the currently logged-in account. The programme is then loaded by calling the render method after this has been verified (Fig. 13).

The render function's job is to load the page, or programme in this case, and make it available in the window. We obtain the logged-in address in the render, which will then be shown on the screen as the currently logged-in user. A simple table is loaded in the render function, to which the talents specified in the function Object() { [native code] } of the TalentShow.sol file are attached. Rows 85 and 86 demonstrate this. The id and name of the talent are both filled into a drop down column, which users can use to pick and vote for the talent. The logged-in address is also tested against the mapping addressesVoting (row 93) in the render, and if the account is detected, the software will show a message stating that it is not possible to vote again.

The add Your Vote (Picture 14) feature collects the id of the talent chosen from the drop-down list (talentsSelect) and then calls the "addYourVote" function in the TalentShow contract. The software will reload the table and hide the vote button and drop-down list after the vote has been counted.

2.3 DApp – Smart Contract – Index.html file

The next step is to create the user interface that will show the results of our Talent Show and give the possibility to the users to interact with it (Fig. 15).

```
107    addYourVote: function() {
108        var talentID = $('#talentsSelect').val();
109        App.contracts.TalentShow.deployed().then(function(i){
110          return i.addYourVote(talentID, { from: App.account });
111        }).then(function(checkResult){
112        $("#content").hide();
113        $("#loader").show();
114        $("#voted").show();
115        }).catch(function(err) {
116          console.error(err);
117        });
118    }
119  };
120
121  $(function() {
122    $(window).load(function() {
123      App.init();
124    });
125  });
```

Fig. 14. app.js source code – addYourVote function.

The index.html file outlines a simple user interface with a drop-down list where the talents that can selected, a submit button to send the vote and a table that will display the current situation of the votes. One thing to highlight in this code is the fact that when the event of submitting a vote is triggered, the function addYourVote in the app.js file in then called. This can be seen in row 23 where the "onSubmit" event triggers the call of the function

Fig. 15. index.html source code.

2.4 Smart Contract – Migration file

Finally, the migration file (Picture 16). This file is necessary if we want to deploy the contract to the blockchain. In this DApp, the file is found under the migrations folder.

```
1  var TalentShow = artifacts.require("./TalentShow.sol");
2
3  module.exports = function(deployer) {
4    deployer.deploy(TalentShow);
5  };
```

Fig. 16 2_migrate_TalentShow.js source code.

With all the files ready, the TalenShow Dapp can be deployed to the local Block- chain. To migrate a contract to the network, we can use the command prompt and issue the command "truffle migrate –reset" (Picture 17). Every time that this command runs, it costs Ethereum. Every time we write on the Blockchain, this requires some funds, in this case Ethereum.

```
C:\Users\Loris\Dapp>truffle migrate --reset

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.


Starting migrations...
======================
> Network name:    'development'
> Network id:      5777
> Block gas limit: 6721975


1_initial_migration.js
======================

   Replacing 'Migrations'
   ----------------------
   > transaction hash:    0xd5e33893e685b3de5d7ab5382b937fd48e0afd1e427b0ba90c79262126d235fd
   > Blocks: 0            Seconds: 0
   > contract address:    0x448F81A86c58ECcDF9DcAf744576b91fC26cf849
   > account:             0x590f04ed287cb7bdc77a6dD84E15e78755C99CcA
   > balance:             99.90071098
   > gas used:            284908
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00569816 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00569816 ETH


2_migrate_TalentShow.js
=======================

   Replacing 'TalentShow'
   ----------------------
   > transaction hash:    0x796108780024e9515a34bfbe408cf945f5864e17e6e189139902f2b3bcf5cdd3
   > Blocks: 0            Seconds: 0
   > contract address:    0x34A5003d5E762A93D5D06ddA853810A7D354b7aE
   > account:             0x590f04ed287cb7bdc77a6dD84E15e78755C99CcA
   > balance:             99.8821957
   > gas used:            883730
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.0176746 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.0176746 ETH
```

Fig. 17 Migration to Blockchain.

The TalentShow.sol contract has successfully been migrated to the local Blockchain, as seen in Figure 17. The expense of migrating the Smart Contract is shown in colour. One question arises as a result of this. Either time we try to post a Smart Contract on the Blockchain, we must ensure that it has been thoroughly checked and is free of bugs. Apart from the conversion expenses, we must bear in mind that all changes to the source code and subsequent migration of the contract would delete and restore the original copy on the Blockchain. Holding our application as an example, if we had deployed the application and the voting was already underway when we discovered an error, migrating the contract again would delete our votes and reset it to 0.

Now that the programme has been effectively moved to the Blockchain, the only thing left is to get lite-server running in the background in order to communicate with the DApp. Lite-server is a locally installed web server that helps users to evaluate Single Page Apps. (2018, John Papa)

From the command line, we can run lite-server with the command npm run dev (Picture 18).

Ganache, the local blockchain emulator, must be open in addition to lite-server running in the context. The DApp will automatically open on the predefined window Chrome at the URL localhost:3000 once the lite-server has been started. We'd still need to use Google Chrome to log into MetaMask and import one of the accounts accessible in Ganache.

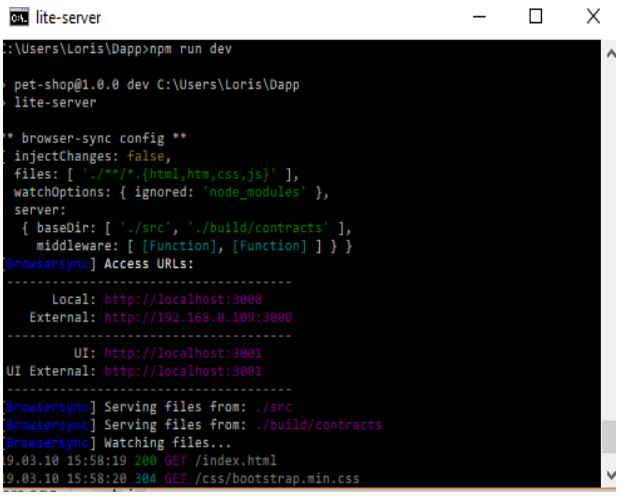

Fig. 18 Lite-server running on local machine.

For the TalentShow DApp project, there will be mainly 4 files that are necessary, as shown in Picture 19:

- A Smart Contract file that will have the extension .sol
- A Migration file that will be invoked when deploying the Smart Contract on the blockchain
- A JavaScript file that will handle the frontend of the application
- A HTML file that will be essentially the application interactive user inter- face (UI)

For the example shown in the following screenshots, the account having the ad- dress 0x3365603866D350ecF2baC7EfF02221517cA46491 was used (Picture 8). This address is the public key of the address. To import one of the accounts to MetaMask we need to use the private key of this account. This value can be obtained easily by pressing the key icon in Ganache. A pop-up containing the private key for that account will appear.

Fig. 19 Private key of 0x3365603866D350ecF2baC7EfF02221517cA46491.

This private key is then used to import, in essence to log in this account in the browser Chrome as active address. The public key then should be visible and displayed by the DApp as the current logged in account.

2.5 Smart Contract – The TalentShow User Interface

With the address logged in to MetaMask, the TalentShow is ready to be tested. A list of talents available in the competition will be displayed and a drop down containing the names of the same talents will be shown as well (Picture 20).

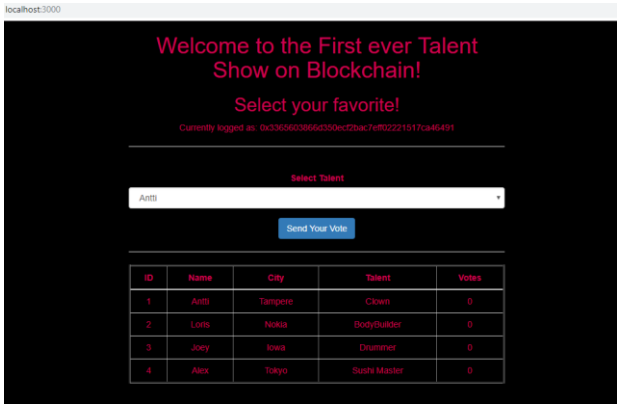

Fig. 20 DApp UI – Before Voting.

Note that the top of the screen, the address entered through MetaMask is displayed. This account has never voted before (all talents have 0 votes) and is then granted to express its own vote. The talent to be voted can be selected from the drop-down list available on top of the table (Picture 21).

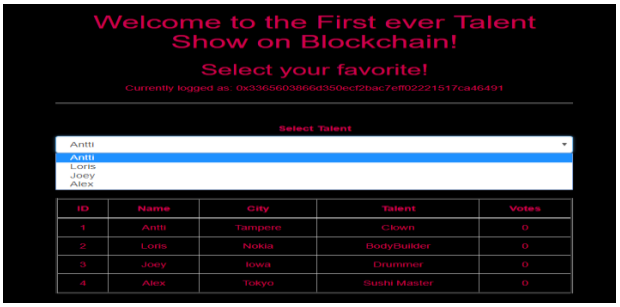

Fig. 21 DApp UI – Selecting the Talent.

By selecting the talent from the drop-down list, the user is then able to vote for it by pressing the "Send Your Vote" button. This event will then be captured by the back-end code that will first register the vote for the selected talent and also memorize the account that just gave its vote and therefore change the UI accordingly by removing the button and showing a message that each user is granted just one vote only (Fig. 22).

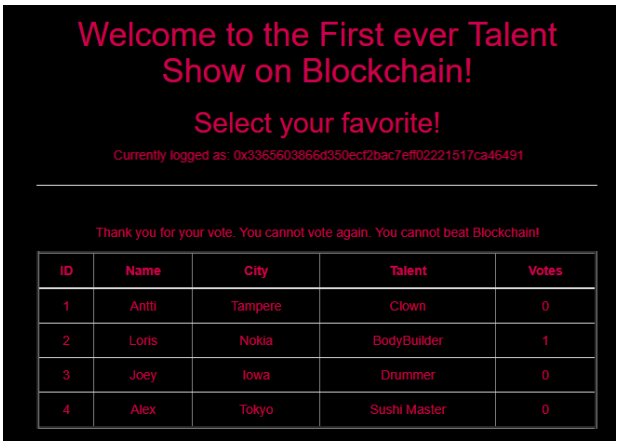

Fig. 22 DApp UI – After Voted – UI not allowing to vote anymore.

Once the transaction is checked by the Block-chain network, a user's operation of voicing a vote turns into a writing action on the Blockchain, which entails an expense. As a result, if the voting addresses do not have any funds, the transaction would be refused by the network.

If we were on a real Blockchain, after the transaction is validated, the data will hopefully be shared through all nodes on the network, with everyone having a copy of the data. The only way to vote again for this local DApp is to update the account logged in MetaMask and reload the tab, but this is obviously not feasible in the actual Blockchain since each user has only one account.

Data fusion from different datasets like human aesthitic evaluation data, scores calculated from the image processing using visual images of the dance performance and scores given by the users using the DApp are fused together (fig23). Therefore, the proposed work adopts block chain technology for securing the fused data.

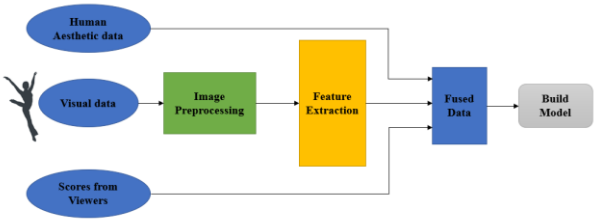

Fig. 23 Data Fusion from different datasets

Smart Contracts are the most innovative and successful means of implementing blockchain technologies. The decentralised implementation seen above is only the tip of the iceberg in terms of what can be done for Smart Contracts and how Blockchain technology functions.

The idea that the Blockchain is a decentralised framework of data saved around the network is, in my mind, the most intriguing part of the Blockchain. Nowadays, one of the more pressing issues is data protection and ethics, especially when dealing with sensitive information. Smart Contracts are not vulnerable to corruption or cyber-attacks on the scheme.

Another benefit that I see is the ability to reduce expenses by growing the number of stakeholders participating and allowing individuals more leverage. Not only does it save money, but it often has a lower environmental effect since less paper work is required. Is it possible for Smart Contracts to fully bypass today's regulatory procedures all around the world? At the moment, perhaps not. Smart Contract technology is not yet widely used, but I believe it will be implemented in many industries in the future, including real estate, banking, health care, and many others.

5. Conclusion

Blockchain technology is undeniably a cutting-edge approach that offers a slew of advantages. Despite the negative connotations that have overshadowed Blockchain and, in particular, Bitcoin in recent years, it is clear that, when used correctly, Blockchain has many benefits than drawbacks. One of the most important

characteristics of this technology, as this study demonstrates, is that it has high data protection. Without risk of being compromised or losing data, blockchain may be used to handle every kind of data, even classified information. The fact that blockchain data is secure and transparent is unquestionably beneficial, yet it often has unintended consequences. As good as it is, it becomes a challenge when it is used for illegal purposes, such as downloading pornography, paedophilia, narcotics, and so forth, because the data cannot be erased or updated.

It would have been beneficial to investigate in this analysis what options and strategies are available when an unauthorised blockchain is set up, as well as what can be done to prevent it. Estonia's adoption of this technology and migration to the distributed ledger of nearly any government and administrative service is a clear success tale of how good this technology can work and what it can achieve. When it comes to this technology, security is the most important consideration. Today's environment is being more digitalized, but data breaches and major digital controversies are unfortunately becoming more common, as shown by the recent Facebook event. Nobody allows their sensitive information to be used and misused without their permission. Smart Contracts and the Ethereum Blockchain have the potential to completely transform the way voting processes are conducted to have a truly trusted framework.

Another advantage of Blockchain, which was not addressed in detail in this report, is that it has resulted in the creation of new employment, such as Blockchain developers and organisations inside established companies such as IBM or new start-ups that see tremendous promise in this new technology. Since Blockchain would be a part of our lives at some stage in the future, courses about it might be provided for free, similar to what is happening in Finland for Artificial Intelligence. Here, data from different datasets like human aesthetics, visual images of the dance and user scores are fused together for evaluation of the dance performance using block chain technology for fusing data.

REFERENCES

[1] Alex de Vries. 2018. Bitcoin's Growing Energy Problem. Read on 14.03.2019. https://www.cell.com/joule/fulltext/S2542-4351(18)30177-6

[2] Alex Treece. 2017. 21 & Done: Will Bitcoin Survive the Tragedy of the Com- mons?. Read on 15.02.2019. https://cryptofundamental.com/21-done-will-bitcoin-survive-the-tragedy-of-the-commons-271a6cd21b04

[3] Andrew Norry. 2018. The History of Silk Road: A Tale of Drugs, Extortion & Bitcoin. Read on 13.03.2019. https://blockonomi.com/history-of-silk-road/

[4] Andrew Tar. 2018. Proof-of-Work, Explained. Read on 22.01.2019. https://cointelegraph.com /explained /proof-of-work-explained.

[5] Antony Lewis. 2018. The Basics of Bitcoins and Blockchains: An Introduction to Cryptocurrencies and the Technology that Powers Them

[6] Ben Brown. 2019. Genesis Block: The First Bitcoin Block Was Mined Ten Years Ago Today. Read on 12.03.2019. https://blockexplorer.com/news/genesis-block-first-bitcoin-block-mined-ten- years-ago-today/

[7] Best Bitcoin Alternative. 2018. Read on 20.02.2019. https://bestbitcoinalternative.com/resources/crypto-wallet/ Blockchain. Average Block Size. Read on 15.01.2019. https://www.blockchain.com/charts/avg-block-size?

[8] Blockchain. Chain's Length. Read on 12.02.2019. https://www.blockchain.com/ru/charts/n-orphaned-blocks?timespan=all Blockchain. Hashrate Distribution. Read on 11.02.2019. https:// www. Blockchain .com/pools?

[9] Cedric Kiama Wachira. 2018. SHA-256 algorithm: How Hashing works in Bitcoins Blockchain. Read on 16.01.2019 https://medium.com/coinmonks/sha-256-algorithim-how-hashing-works-in- bitcoins-blockchain-a4315eb3df05

[10] Christopher Helman. 2018. Bitcoin Mining Uses as Much Power as Ireland. Here's Why That's Not a Problem. Read on 07.02.2019. https://www.forbes.com/sites/christopherhelman/2018/01/16/bitcoin-mining- uses-as-much-power-as-ireland-and-why-thats-not-a-problem/#65b154eb4589

[11] CoinMarketCap. 2019. All Cryptocurrencies. https://coinmarketcap.com/all/views/all/
[12] Damien McGuinness. 2017. How a cyber attack transformed Estonia. Read on 08.04.2019. https://www.bbc.com/news/39655415
[13] Daniel McLaughlin. 2007. Violence in Tallinn as Soviet statue removed. Read on 08.04.2019. https://www.irishtimes.com/news/violence-in-tallinn-as-soviet-statue-removed-1.1295460