DOI 10.34185/1562-9945-1-126-2020-11
UDC 004.415.2

I.V. Ponomarev

DEVELOPMENT OF A DECENTRALIZED VOTING APPLICATION USING BLOCKCHAIN TECHNOLOGY

Abstract. To date, blockchain technologies have gained general interest as a new approach to creating secure storage and data processing systems. This approach can be applied in many industries and opens up new opportunities to increase transparency and improve the performance of distributed systems. When voting, it is important to have a guarantee that no one will be able to manipulate the data and that the information is accessible to all. The development of a decentralized voting application using blockchain technology is being considered.
Keywords: blockchain, block, transaction, mining, SHA-256, JSON, Proof-of-Work, WebSocket, Dapps, peer-to-peer.

Formulation of the problem. The problem of automation and translation of the voting process online is relevant worldwide. The main disadvantages of traditional voting are the inconvenience associated with the time spent waiting in lines, when voting, and when changing the place of voting. In this regard, there is not a high turnout. Using online voting allows you to vote without being tied to a polling station, without leaving your home and away. The main task is to ensure the impossibility of falsification of the voting process, vote substitution, incapacitation of the system itself. It also requires high reliability of storage and calculation of voting results, transmission of its results and final output online. Blockchain technology, as a matter of fact, ensures the transparency and reliability of storing decentralized, unchanged data.

Purpose of the research. It is necessary to analyze the capabilities of distributed registry technology that can be used for the voting system.

The following requirements must be met when developing an application:

SSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

1. Design a blockchain storage model that will store the necessary user information and votes.

2. Develop an algorithm for mining and linking blocks.

3. Provide the ability to authorize and register users in the blockchain.

4. Organize the ability to create and hold votes, as well as calculate and display their results.

5. Implement a communication system between nodes [2].

Using the results of the analysis, create a decentralized voting application (Dapps) using the .NET Core framework using C# programming language.

Main part. Blockchain is called a continuous chain of blocks containing information, constructed by certain rules (Fig. 1).

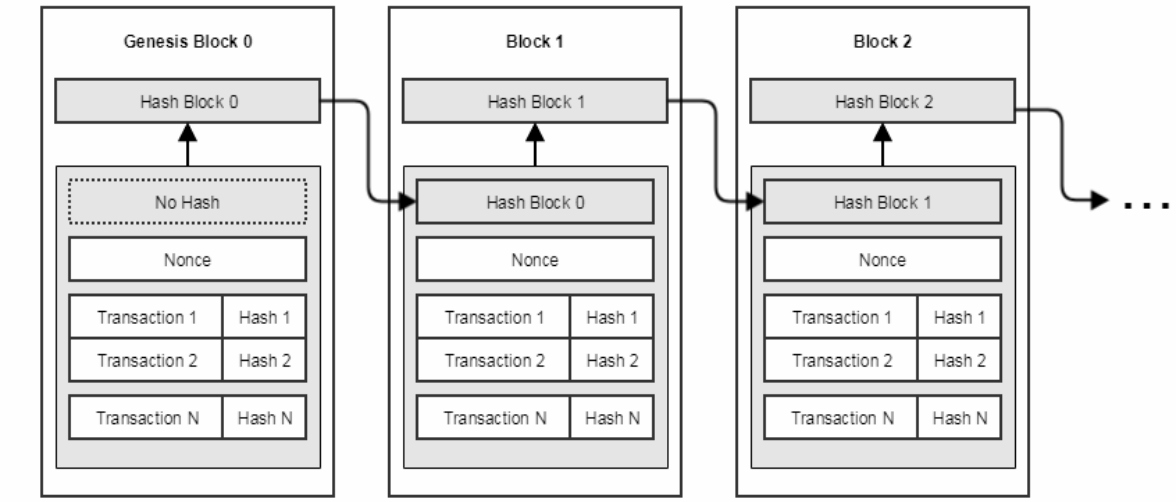

Figure 1 – Block chain in blockchain network

Most often, copies of block chains are stored on multiple computers of peers, independently of one another. All members of the network are divided into two categories: regular users creating new records, and miners creating blocks.

The miner checks the records created by ordinary users, form blocks from them, and then sends them to other nodes in the network. Blockchain members have access to other computers on the network so that they can share data. Each user checks the correctness of the new data. If they are reliable, it stores them and transmits them further through the network nodes [1].

In the network, the miner also acts as a network node, that is, it must store the entire chain of blocks, and interact with other nodes in the network.

Block binding technology ensures that data is invariable, as you change the information in a block to look again for the hash sums of the given complexity of all subsequent blocks. Therefore, in order to make changes to the blockchain, it is necessary to have control of at least 51% of all nodes in the system and to have large processing capacity to guess the new hash sums of all subsequent blocks. Due to this, blockchain networks are characterized by extreme data security.

All modern blockchain systems are owned the following important properties.

• Transparency. Access to all event history - voting results, deals and other records - is always open to all members of the system.

• Decentralization. The transaction history is stored by each member, not the main server.

• Anonymity. To work in a blockchain you do not need to reveal your identity.

• Security. No one will forge and replace the information recorded in the blockchain. You can be sure that it is reliable.

• Equality. The blockchain has no administrators or custodians of information, and all participants have the same status and capabilities.

Development of a decentralized voting application.

Blockchain storage model. All the information in the form of linked blocks will be stored in a file with the JSON extension. To store transaction data in a block, we create a Transaction class. It contains all the necessary automatic properties that will store information about the survey questions, answer options, selected answer and the user. A block class is created to store transaction information, information required for mining, and to add a block to a blockchain repository. This class contains properties that store the block index, author name, hash value of the previous block, date of creation of the current block, and object of the Transaction class. To interact with the blockchain miner, a Blockchain class has been created that contains methods for writing and reading blockchain information, which is a JSON file.

At the beginning of the work you need to create your own genesis block, the hash sum of which will be based on the following blocks.

SSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Development of a mining algorithm. At the stage of adding a new block to the existing block chain, there is a mining process that is implemented using the SHA-256 hashing algorithm. For blockchain mining, there is a mine() method.

```
public Block mine(Block block)
 {
  List<Block> items = readJson();
  block.index = items.Last().index + 1;
  using (SHA256 mySHA256 = SHA256.Create())
  {
    block.previousHash = ByteArrayToString(
  mySHA256.ComputeHash(SerializeByte.SerializeToByteArray(
    items.Last())));
    int count = 0;
    while (true)
     {
      block.random = count;
      count++;
      byte[] i = mySHA256.ComputeHash(
            SerializeByte.SerializeToByteArray(block));
      if(isLess(ByteArrayToString(i), complexity))
       {
          Console.WriteLine(ByteArrayToString(i));
        Console.WriteLine("Miner found the right hash");
        Console.WriteLine("Iterations: " + count);
        break;
       }
    }
  }
  return block;
}
```

In order to set the previous block hash and the corresponding index in the new block, the mine method first obtains the index and hash of the last block in the blockchain. For successful block mining, it is necessary that the hash sum of the block is less than the specified complexity, only then the block is added to the blockchain. The random field gradually increases in the loop. Because the SHA-256 algorithm is extremely sensitive, any change in the random field significantly changes the hash of the block.

This means that there is nothing left for the miner to select such a value of the random field at which the hash sum will correspond to a given complexity. The miner, who first found the appropriate hash sum, tells the other nodes that the block has been successfully added to the blockchain.

User registration. At the registration stage, the miner receives a model with the registration data, and begins the mining process. Once the appropriate number for the random field has been selected, the registration block is added to the blockchain, thus registration is considered successful.

Organization of communication. To create decentralized decentralized blockchain and Proof-of-Work, you need to have a link between the miners. WebSocket is a two-way communication protocol for real-time messaging. The messages act as a JSON string that contains a transaction type in the header.

As the miner, in case of successful addition of the block, has to inform the other nodes of the network, it is necessary to implement listening of messages from other nodes. When a message is received, it is deserialized into a Block object, and the writeBlockFromOther method is invoked, which finds the hash of the block, and if it matches the specified complexity, adds the block to the blockchain. The requestRegister method, after receiving a list of all the IP addresses of the miner, gradually addresses each of them with a request. After that, the miners begin the process of selecting the appropriate hash amount to add the transaction to the block.

Creation and holding of votes. The RegisterVotingModel class contains properties that store questions, answer options, and user logins. At the polling stage, the miner receives the RegisterVotingModel model and begins the block mining process, after which the vote is considered to be registered.

The VotingController class gives the user the ability to create new votes. The requestRegisterVoting method sends requests to miners. Miners start searching for blocks with registered polls and then serialize them to JSON and return them to the server. The server deserializes the response into a list of Voting type items. The Voting class contains properties that store the poll number, questions, and answer options as a list of Choise objects.

```
public class Voting
    {
        public int Id { get; set; }
        public string Question { get; set; }
        public List<Choise> Choises { get; set; }
    }
```


SSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

```
public class Choise
{
    public int count { get; set; }
    public string Answer { get; set; }
    public Choise(int count, string answer)
    {
        this.count = count;
        this.Answer = answer;
    }
}
```

In order to allow users to answer the survey, the Answer method in the VotingController class was created. This method takes as an argument a model that contains a user-selected answer and other information to uniquely identify the vote. The method sends queries to the miners and waits for answers. Miners begin to search for blocks of answers for this user to verify that he or she has not yet participated in this survey. After that, the mining begins and if the block is successfully added, the miner returns the confirmation. If more than 50% of the miner returned the confirmation, then the user response is considered accepted.

The user does not necessarily need to be a miner to vote. It can be trusted by other miners who ensure the reliability and consistency of data on the network.

Conclusions. Due to decentralization, voting results depend not on a single center, but equally on all nodes of the network, since all network members are equal and keep a complete transaction history. Block binding technology ensures that data is invariable, as you change the information in a block to look again for the hash sums of the given complexity of all subsequent blocks. Blockchain provides both security and anonymity to voting, and can help increase the number of voting participants.

REFERENCES

1. Imran Bashir. Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained, 2nd Edition. – Packt Publishing, 2018. – 656 p.

2. Narayan Prusty. Building Blockchain Projects: Building decentralized Blockchain applications with Ethereum and Solidity. — Packt Publishing, 2017. — 268 p.

3. I. Ponomarev. The possibilities of using the blockchain technology in the online voting system - Promising directions of modern electronics, information and computer systems: IV All-Ukrainian Scientific-Practical Conference MEICS-2019. Dnipro, November 27-29, 2019 - p. 88-89.

Received 23.01.2020.
Accepted 27.01.2020.

Розробка децентралізованого застосунка голосування з використанням технології блокчейн

Розглядаються такі основні особливості блокчейна, як дублювання, незмінюваність та відкритість даних. Пропонується децентралізований додаток онлайн-голосування, розроблений на базі цієї архітектури.

Разработка децентрализованного приложений голосования с использованием технологии блокчейн

Рассматриваются следующие основные особенности блокчейна, как дублирование, несменяемость и открытость данных. Предлагается децентрализованный приложение онлайн-голосования, разработанный на базе этой архитектуры.

Пономарев Игорь Владимирович - доцент, к.т.н., доцент кафедры ЭВМ Днепропетровского национального университета им. О. Гончара.

Пономарьов Ігор Володимирович – доцент, к.т.н., доцент кафедри ЕОМ Дніпропетровського національного університету ім. О. Гончара.

Ponomarev Igor - candidate of technical sciences, associate professor of the department of electronic computers of the faculty of physics electronics and computer systems of the Oles Honchar Dnipro National University.

SSN 1562-9945 (Print)
ISSN 2707-7977 (Online)