DOI 10.34185/1562-9945-1-132-2021-11

UDC 004.415.2

I.V. Ponomarev

FEATURES OF CREATING A VOTING SYSTEM USING THE ETHEREUM BLOCKCHAIN PLATFORM

Abstract. Today, blockchain applications are being developed for a wide variety of areas of activity - from trade and advertising to logistics and social networks. Building an application using a ready-made blockchain on one of the specialized platforms is the most efficient way to develop. The development of a decentralized voting system on one of the most functional blockchain platforms Ethereum with a developed infrastructure for creating smart contracts is being considered.
Keywords: blockchain, platform, smart contract, peer-to-peer network, transaction, mining, Ethereum, Solidity, Geth, Remix, Web3.

Formulation of the problem. When creating an online voting system, it is important that the system is accessible to voters and observers, reliable, protected from attacks and falsifications, and at the same time, hides information about how a particular voter voted.

Centralized voting systems have a single central node for storing, counting votes and issuing voting results, which is the main point of failure of the entire system.

Therefore, for reliable operation, a decentralized system based on blockchain technology without a single master node, with control distributed among many nodes, is needed.

The creation of such a system is possible in two ways:

- development of your blockchain system from scratch,
- using existing blockchain platforms based on smart contracts.

The first method is very laborious, since it is necessary to implement all the basic elements of support for the performance of a decentralized system: the creation of blocks, the consensus mechanism, mining algorithms, key generation, encryption, data transfer between nodes [1].

The second method allows creating business logic of an application using smart contracts on the basis of a ready-made blockchain platform infrastructure.

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

Purpose of the research. Consideration should be given to creating a reliable voting system based on the open Ethereum platform. During the development process, the following steps should be performed:

1) writing a smart contract to implement the voting functionality;

2) development of an interface for voting and viewing the results;

3) deployment of a private blockchain network on the Ethereum platform: creating network nodes and setting up their interaction with each other;

4) compilation and implementation of the contract into a private blockchain network;

5) testing the performance of the developed system from various network nodes.

Main part. The use of blockchain for voting provides the following benefits:

• Transparency of the process. Ability to control the voting process - any participant can deploy a node with a full copy of all data and analyze it.

• Protection of results. Voting results cannot be falsified. You can always check how many votes were issued at the beginning of the vote, how they were distributed among wallets and at what time the transactions were carried out.

• Anonymity. A public and private key pair is created for a voter on the local machine, and no one knows that a particular wallet belongs to this voter. Thus, no one can know how a given participant voted.

• Speed of data processing. Decentralization allows you to see the overall voting results at each network node.

The open platform Ethereum represents a single decentralized virtual machine in a peer-to-peer network and allows transactions of any complexity level. Smart contracts allow you to register transactions with any asset in a distributed ledger, protecting transactions through hash sums on the blockchain. A smart contract is executed on every node in the network [2].

To develop smart contracts in Ethereum, the built-in programming language Solidity is used, in which you can create contracts with arbitrary conditions, transaction formats and state change functions.

One of the most convenient tools for writing and compiling smart contracts for the Ethereum platform is the Remix IDE. It includes a built-in debugger with blockchain emulation mode in which you can run contracts and a testing environment. After compilation and testing, contracts are executed on the Ethereum virtual machine [3].

An example of a smart contract for voting.

```
pragma solidity >=0.4.0 <0.6.0;
contract Voting {
  mapping (bytes32 => uint256) public votesReceived;
  bytes32[] private voteAnswers;
  string public voteQuestion;
  constructor(string question, bytes32[] voteAnswers) public {
    voteQuestion = question;
    voteAnswers = voteAnswers;
  }
  function totalVotes(bytes32 variant) constant public re-
turns(uint256) {
    require(valid (variant));
    return votesReceived[variant];
  }
  function vote(bytes32 variant) public {
    require(valid(variant));
    votesReceived[variant] += 1;
  }
  function valid(bytes32 variant) constant public returns (bool)
{
    for(uint i = 0; i < voteAnswers.length; i++) {
      if (voteAnswers[i] == variant) {
        return true;
      }
    }
    return false;
  }
  function getAnswers() public constant returns (bytes32[]){
      return voteAnswers;
  }
}
```

The contract has a constructor that takes 2 parameters: a string of type string, in which the question of the vote is passed, and an array of type bytes32, in which the choices (candidates) are passed. bytes32 is a byte array of a fixed size of 32 bytes. Fixed length bytes can be used in function arguments to transfer data or return data by contract.

Function vote adds a vote for a specific candidate (variant). Function totalVotes returns the number of votes for a specific candidate (variant). They use function require (bool condition), which checks the condition and if the condition is false, abort and return the state change. Function getAnswers returns a list of voting results.

To connect nodes to the blockchain, the implementation of the Ethereum Geth protocol is used, which is downloaded as a client to all network nodes. A private

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

blockchain network with its unique first block and network identifier is being deployed. The initial genesis block is created based on the genesis.json configuration file. The required number of network nodes is created, mining is started. The contract is compiled and deployed on the private blockchain network.

To work with a smart contract, you need to access it through a web interface, which includes web pages with forms for entering variable values and passing them to functions. For interaction between the browser and the blockchain, the Web3.js library is required, which allows working with the nodes of the Ethereum network using the Remote Procedure Call (RPC) and HTTP [3] protocols. With its help, event handlers are created that implement the corresponding functions of the smart contract (Fig. 1).

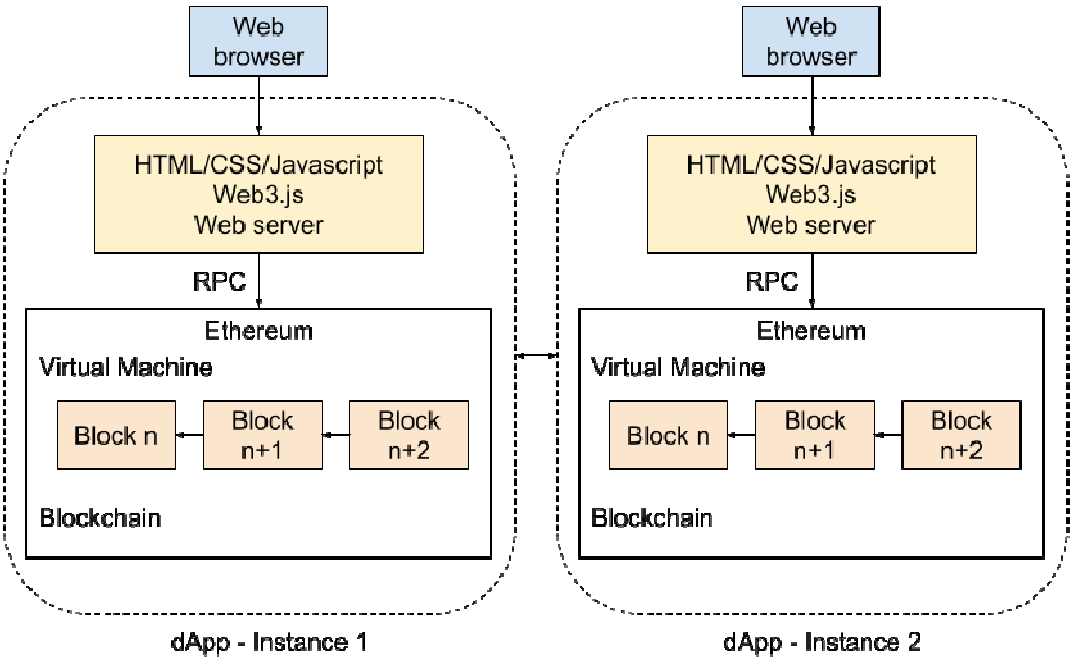

Figure 1 – Decentralized application architecture
on the Ethereum blockchain

For the convenience of interacting with the contract, an interface has been developed for voting and viewing the results. The voter is authorized, the connection to the network node is made, the identifier of the smart contract is specified, the account containing this contract is unlocked, the data for its operation is loaded. After a voter selects one of the candidates on the form, information in byte32 format is sent for processing to the smart contract of the network node. A transaction is created with a call to the contract, mining is started and after its completion, the contract is executed, the results of its work are saved and displayed on the web page.

Thus, a decentralized application has been created, ready to be placed on the real blockchain of the Ethereum network. However, for each use of the computing resources of the platform when executing a smart contract, a commission will be charged, which goes to the platform miners who ensure the operation of the network.

Conclusions. The features of the Ethereum open blockchain platform and its tools for developing a smart contract and an interface for a voting system are considered. The sequential stages of actions are described: creating nodes and combining them into a network, installing the necessary tools on them, creating and debugging a contract, introducing a smart contract into a blockchain network, creating a user interface and interacting with a blockchain.

REFERENCES

1. I.V. Ponomarev Development of a decentralized voting application using blockchain technology – System technologies: Regional interuniversity compendium of scientific works. Issue 1(126). – Dnipro, 2020, - P. 104-110.

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

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

Received 25.01.2021.
Accepted 29.01.2021.

Особливості створення системи голосування з використанням блокчейн-платформи Ethereum

Система онлайн-голосування повинна бути доступною і надійною, захищеною та анонімною. Всі ці якості забезпечує децентралізована система заснована на технології блокчейн без єдиного головного вузла, з управлінням розподіленим між багатьма вузлами.

Можливо розробити свою блокчейн-систему з нуля або використовувати існуючу блокчейн-платформу. Перший спосіб дуже трудомісткий, так як необхідно реалізовувати всі базові елементи підтримки працездатності децентралізованої системи: створення блоків, механізм консенсусу, алгоритми майнінгу, генерації ключів, шифрування, передачі даних між вузлами. Другий спосіб дозволяє на базі готової інфраструктури блокчейн-платформи створювати бізнес-логіку додатка за допомогою смарт-контрактів.

Розглядається розробка децентралізованої системи голосування на одній з найбільш функціональних блокчейн-платформ Ethereum з розвиненою інфраструктурою для створення смарт-контрактів.

Розбираються особливості відкритої платформи Ethereum, її інструментарії для розробки смарт-контракту та інтерфейсу для системи голосування. Описуються послі-

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

довні етапи дій: створення вузлів і об'єднання їх в мережу, установка на вузлах необхідних інструментальних програм, створення і налагодження контракту, впровадження смарт-контракту в блокчейн-мережу, створення інтерфейсу користувача і взаємодія з блокчейном.

Створено децентралізований додаток онлайн-голосування готовий для розміщення в реальній блокчейн-мережі Ethereum.

Особенности создания системы голосования с использованием блокчейн-платформы Ethereum

Система онлайн-голосования должна быть доступной и надежной, защищенной и анонимной. Все эти качества обеспечивает децентрализованная система основана на технологии блокчейн без единого главного узла, с управлением распределенной между многими узлами.

Возможно разработать свою блокчейн-систему с нуля или использовать существующую блокчейн-платформу. Первый способ очень трудоемкий, так как необходимо реализовывать все базовые элементы поддержания работоспособности децентрализованной системы: создание блоков, механизм консенсуса, алгоритмы Майнинг, генерации ключей, шифрование, передачи данных между узлами. Второй способ позволяет на базе готовой инфраструктуры блокчейн-платформы создавать бизнес-логику приложения с помощью смарт-контрактов.

Рассматривается разработка децентрализованной системы голосования на одном из наи-более функциональных блокчейн-платформ Ethereum с развитой инфраструктурой для создания смарт-контрактов.

Разбираются особенности открытой платформы Ethereum, ее инструментарии для разработки смарт-контракта и интерфейса для системы голосования. Описываются после-довательно этапы действий: создание узлов и объединение их в сеть, установка на узлах необходи-ных инструментальных программ, создания и отладки контракта, внедрение смарт-контракта в блокчейн-сеть, создание интерфейса пользователя и взаимодействие с блокчейном.

Создан децентрализованный приложение онлайн-голосования готов для размещения в реальной блокчейн сети Ethereum.

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

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

Ponomarev Igor Volodimirovich - 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.