UDC 004 DOI https://doi.org/10.32782/1995-0519.2023.3.5

METHODS AND MEANS OF CREATING A PLATFORM BASED ON BLOCKCHAIN TECHNOLOGY

Galyna Marchuk
Senior Lecturer at the Department of Computer Science
Zhytomyr Polytechnic State University, 103 Chudnivska str., Zhytomyr, Ukraine, 10003, pzs_mgv@ztu.edu.ua
ORCID: 0000-0003-2954-1057

Vitalii Levkivskyi
Senior Lecturer at the Department of Computer Science
Zhytomyr Polytechnic State University, 103 Chudnivska str., Zhytomyr, Ukraine, 10003, levkivskyy@ztu.edu.ua
ORCID: 0000-0002-1643-0895

Alex Kuzmenko
Senior Lecturer at the Department of Computer Science
Zhytomyr Polytechnic State University, 103 Chudnivska str., Zhytomyr, Ukraine, 10003, kkn_kov@ztu.edu.ua
ORCID: 0000-0002-4937-3284

Dmytro Marchuk
Senior Lecturer at the Department of Computer Science
Zhytomyr Polytechnic State University, 103 Chudnivska str., Zhytomyr, Ukraine, 10003, kipz_mdk@ztu.edu.ua
ORCID: 0000-0001-8675-8047

Anton Levchenko
Candidate of Technical Sciences, Senior Lecturer at the Department of Software Engineering
Zhytomyr Polytechnic State University, 103 Chudnivska str., Zhytomyr, Ukraine, 10003, levchenko@ztu.edu.ua
ORCID: 0000-0002-4411-6465

Blockchain technology and the cryptocurrency market are developing and spreading quite dynamically and thus attract a lot of attention. The technology innovation of blockchain is that information about transactions is no longer stored in a centralized database, but is transferred to the computers of all network participants, who store the data locally. Bitcoin was the first application based on blockchain technology. In recent years, it has been the basis for the creation of other blockchain applications, most of which are currently being developed in the financial sector. In fact, blockchain technology is a universal way of storing and processing information in almost any field of activity.

The paper analyzes the Ethereum and Fantom platforms. It was determined that Ethereum demonstrates greater flexibility, scalability and speed, allowing the use of various electronic wallets. The platform introduced the concept of smart contracts. Programming languages for writing smart contracts were analyzed and Solidity language was selected for various use cases including decentralized finance, NFTs and wallets. In working with the blockchain, liquidity issues often appear, which have been solved and described. An analysis of methods and means of implementing blockchain technology to create a platform for processing credit and deposit operations is the result of our work.

Blockchain technology opens up a huge range of possibilities, starting with money transfers and payments, ending with smart contracts and reconciliation of documents. It is worth noting that blockchain technology has a number of questions related to insufficient study, understanding of technical implementation, and flexibility.

Key words: Blockchain, Ethereum, Fantom, Smart-contracts, crypto wallet.

Introduction. Blockchain is a special type of database. Data is added to structures called blocks. Blocks of data are organized according to certain rules in chronological order, interconnected and cryptographically protected. Most often, copies of blocks are stored on many different computers independently of each other. The term first appeared as the name of a distributed database implemented in the "Bitcoin" system. Therefore, blockchain is often referred to as transactions related to various cryptocurrencies.

Currently, there are already three generations of blockchain. The first generation appeared in 2009, which is based on a new principle of transaction reg-

istration. The release of Ethereum in 2013 marked the beginning of the second generation. The Ethereum platform introduced smart contracts that store the terms of the contract in the blockchain and allow for its automatic execution if a number of contract conditions are met. In this way, intermediaries, costs and bureaucratic costs are reduced, and the risk of third-party intervention is eliminated. The third generation started in 2017 with the Cosmos, DFINITY and EOS platforms. The third generation of blockchain technology is still in the process of development, but active work is already underway on solutions capable of revolutionizing the industry. This is the Seele [1] project, which is often called the fourth generation blockchain. The technology involves the development of large-scale industrial applications capable of simultaneously managing many processes, processing and storing huge data sets, ensuring their logical interconnection and consistency.

Currently, the technology is interesting for many corporations, such as Microsoft, IBM, Blockchain Foundry, Chain, which are studying the possibilities of applying blockchain technologies in various branches: finance, manufacturing, education etc.

Problem statement. Blockchain technology has a number of advantages and disadvantages. Advantages include cost reduction, increased security, and transaction transparency. But not all operations can be simplified with the help of blockchain. Currently, the technology is not sufficiently studied. Therefore, the purpose of this study is to analyze the methods and means of implementing blockchain technology to create a platform for credit and deposit processing. A list of tasks that must be completed follows from the defined purpose:

– to analyze the most popular blockchain platforms;

– to review and analyze the programming languages for smart contracts implementation;

– to choose a crypto wallet;

– to describe problems with liquidity that may appear;

– to develop an algorithm for the platform of credit and deposit operations.

Review of the literature. A number of studies devoted to blockchain technology focus on the possibilities of its application in various areas. Ashraf Jaradat and others in their study provide an overview of the blockchain concept, its main characteristics, components, categories and architectures [2]. The authors emphasize the importance of using this technology in several areas related to data. Some aspects related to the future use of blockchain technology and artificial intelligence and implementation of digital privacy are highlighted. In the paper "A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities" a comparative study of blockchain trade-offs was presented and blockchain taxonomy and architecture were explained. The authors point to the technology's challenges, including scalability, privacy, interoperability, and power consumption. In addition, this document also highlights the future application of blockchain technology [3]. In the paper "Blockchain: Challenges and Applications", the authors review applications using Blockchain technology and the problems they face [4]. The authors of the work [5] indicate the problem of scalability for blockchain-based platforms. Xiaoqi Li and others conduct research on blockchain security threats [6]. The work offers solutions to increase security that can be used in the development of various blockchain systems.

The paper [7] presents an overview of smart contracts with blockchain support both from the technical aspect and from the aspect of use. The authors provide a taxonomy of existing solutions for smart contracts using the blockchain. Based on the results of the survey, a number of problems with smart contracts and open questions that need to be resolved in further research have been identified. Article [8] is aimed at researching the formalization of smart contracts based on blockchain technology. The results show that the most common formalization method is theorem proving, which is most often used to verify the security properties associated with smart contracts. Hewa T., Ylianttila M., Liyanage M. make a review of smart contracts using blockchain [9]. The work examines applications that have already benefited from smart contracts. The future potential of blockchain-based smart contracts is highlighted. Lin SY., Zhang L., Li J. and other authors present the model and working principle of the blockchain smart contract for the general architecture, analyze the smart contract deployment process with Ethereum, Hyperledger Fabric and EOSIO, perform a comparative analysis at the technical level [10]. The Ethereum platform supports its own cryptocurrency and smart contracts. In the work "Understanding Ethereum via Graph Analysis", Ethereum research was carried out using graph analysis. The authors characterize three main actions in Ethereum, namely money transfers, creating smart contracts and calling smart contracts [11].

Thus, blockchain technology has great potential and can replace many modern digital plat forms.

Analysis of blockchain technology implementation solutions.

Eterium vs fantom. Blockchain platforms allow developers to easily and efficiently build blockchain-based applications. However, they need to make the right choice based on their needs. The authors of the paper [12] point to the problem of a large number of platforms for blockchain development. They review and highlight 15 of the most popular open source platforms, including: Ethereum, Waves Platform, TON, NEM, Lisk.

Ethereum is one of the best blockchain development platforms in terms of usability and popularity. The platform was created in 2013. At the start of development, the main idea was to improve the algorithmization of Bitcoin and create a dApp – an online platform for interacting with the blockchain using all possible online solutions. To implement such applications, the idea of smart contracts was formed. These contracts are stored on the blockchain exactly as they were programmed at the beginning, which is a form of Stewart Haber's principle of decentralization, namely a digital signature that makes it impossible to change anything. Ethereum Virtual Machine (EVM) is used to run smart contracts in Ethereum. In turn, the Ethereum network is supported by a decentralized network of nodes, which are computers or any other devices such as servers, which in turn run virtual machines and help process and verify transactions on the network. These nodes are scattered all over the world and are managed by different groups of persons, which does not allow to control or forge any transactions in the network, but only to see the result of the work performed by the smart contract.

Smart-contracts provide a safe exchange of cryptocurrency, money, securities, services and goods without the participation of intermediaries. After the appearance of Smart-contracts, a mass excitement began. Various companies and large IT giants such as Microsoft have considered the idea of smart contracts as a huge step into the future. A lot of startups have also appeared. In 2017, a token exchange platform was created based on Ethereum smart contracts, and in early 2018, a separate blockchain based on a fork of public Ethereum was launched.

One of the main competitors of Ethereum today is the Fantom Foundation platform. The mission of this platform was to improve existing crypto-algorithms to create faster transaction methods and accelerate the expansion of the blockchain itself. The Fantom platform is powered by a new engine called "Lachesis", which is based on a Directed Acyclic Graph (DAG) data structure. This provides faster transaction speeds and lower fees compared to Ethereum. The Fantom platform also runs on a virtual machine and runs smart contracts just like Ethereum. In 2019, the Fantom Foundation launched the Fantom Opera Mainnet, which is the main network for the Fantom platform.

Based on the results of the analysis, the Ethereum platform demonstrates greater flexibility, scalability and speed. This platform is progressing quickly, offers great potential for growth, and is more stable today compared to the Fantom Foundation.

Smart-Contracts Solidity vs Vyper. A smart contract in a blockchain system is a program that works as a digital agreement, the receipt of which is ensured by a set of rules. These rules are defined by software code that must be followed by all network nodes.

Smart contracts in Ethereum are usually written in high-level languages and then compiled into EVM (Ethereum Virtual Machine) bytecode. Ethereum uses Solidity as its main programming language for developing smart contracts and dApps.

Table 1

Advantages and disadvantages of Solidity

Advantages	Disadvantages
Officially presented by Ethereum developers	Difficulty tracking errors
Developed in JavaScript	Big syntax difference between releases
Supports all popular Ethereum-based blockchains	No automatic stack overflow checks.
It has full documentation	No automatic checks of mathematical operations
Has a clear description of tokens and algorithms for working with them	

Solidity is a high-level programming language with JavaScript-like syntax. Solidity is the most popular and well-formed language for developing smart contracts. It has complete documentation on writing methods, a large community of developers. The advantage is that the language is similar to most existing object-oriented programming languages, which makes it easier for developers to understand. The advantages and disadvantages of Solidity are presented in Table 1.

An example of code written in Solidity is shown in Figure 1.

Fig. 1. Example of Solidity code

Vyper is the newest Blockchain programming language which is an alternative to Solidity. It is a high-level Ethereum smart contract development language designed to address security vulnerabilities. Table 2 presents the advantages and disadvantages of the Vyper language.

Table 2

Advantages and disadvantages of Vyper

Advantages	Disadvantages
Similar to Python	Incomplete documentation
Specially developed to increase the security of smart contracts	Own method of uploading contracts to the blockchain
There is no problem of automatic checks of mathematical operations	Not popular with the developer community
Possibility of audit	Much of the object-oriented functionality has been removed

The language is written in Python. Despite the high interest among developers, it is very difficult to understand. The developers continue to improve the documentation, but it does not give an understanding of all its features. Due to the fact that the language is specific, to upload contracts to the blockchain, it is necessary to use the Vyper interface, which is not very convenient. An example of code written in Vyper is shown in Figure 2.

It is possible to use both Solidity and Vyper to write smart contracts. For writing simple contracts, it is better to choose Solidity. For contracts that require security guarantees, it is better to use Vyper. Although in some cases it is possible to use both programming languages in the same project to write different contracts.

Crypto wallet. Any transaction on the blockchain requires a wallet address, which is the user's ID. A single wallet can be an identity in different networks, so a provider is needed for the user to interact with the contract. The easiest to use is MetaMask – a cryptocurrency wallet that is popular among blockchain users. It has a fairly simple interface and does not require any verifications. It is available as a browser extension that works on Chrome, Brave, and Firefox.

Fig. 2. Code example on Vyper

MetaMask allows users to store crypto assets on Ethereum, Binance Smart Chain, Polygon and other blockchains. MetaMask does not control user's coins or limit transactions with them. Therefore, if users store their crypto assets properly and keep their private keys safe, their money will be safe. The MetaMask crypto-wallet has not had any serious incidents related to hacking, but the wallet has one significant drawback – it requires an Internet connection, which makes individual users vulnerable.

Basic aspects of the development of a platform for credit and deposit operations. During the implementation of the platform for credit and deposit operations based on blockchain technology, a problem with liquidity appeared – the evaluation of tokens. Tokens must be valued. Without this evaluation, it's just a set of hash characters. So, the first obvious solution is to create constant values inside the contract and refer to them with each transaction that occurs by running a method in the smart contract. But at the same time, in the future, in any case, a liquidity problem will appear due to a change in the price of tokens in the network itself. Also, when evaluating the price of a token, its price in fiat equivalent is used, namely, the token is converted into a dollar.

To avoid this problem during the operation of the system, it is recommended to use the Oracle tool – infrastructure algorithms that translate information outside the network into a format that is understandable for the blockchain. These algorithms work in nodes and make it possible not only to monitor liquidity, but also make it possible to implement certain event trackers, for example, during any gamifications using NFT tokens.

To solve the problem with liquidity, it can be used a substitute for a cross-chain oracle in the form of a smart contract in which rates in fiat values are set and also a number is set for correct rounding during calculations in transactions, which simply indicates the accuracy of calculations, namely the number of decimal values, after an integer parts, since all numbers in the blockchain are stored in uint256. After that, the contract is uploaded to the network.

```
contract Oracle is Ownable { mapping(address => uint256) public
rates;
mapping(address => uint256) public tokenPreccision; uint256
public preccission = 1e6;
constructor() {} function setRate(
address _token, uint256 _rate, uint256 _tokenPreccision)
external onlyOwner { rates[_token] = _rate;
tokenPreccision[_token] = _tokenPreccision;
}
function getRate(address _token) public view
returns (uint256 rate, uint256 preccissionForToken){
rate = rates[_token];
preccissionForToken = tokenPreccision[_token];
}
```

Next, it is needed to add all rates to the script that uploads the contracts to the network. After that, it is possible to receive rates from different contracts and the evaluation of each specific token in the network.

```
const oracle = await Oracle.deploy(); await
oracle.deployed(); console.log("Oracle", oracle.address);
await oracle.setRate( wbnb.address,
ethers.utils.parseUnits("300.0", 6),
ethers.utils.parseUnits("1", 18)
);
await oracle.setRate( weth.address,
ethers.utils.parseUnits("1300.0", 6),
ethers.utils.parseUnits("1", 18)
);
await oracle.setRate( wbtc.address,
ethers.utils.parseUnits("17000.0", 6),
ethers.utils.parseUnits("1", 8)
);
console.log("rates setted");
```

After these steps, any transactions can be made with the confidence that the liquidity will always be correct.

Also, for the platform to work, a contractual part of the service must be created, which will perform all calculations and store the value of credits and deposits in the blockchain. Tokens are created for this. Token is a contract in which user accumulations are stored, with the following read methods for interact-

ing with them: allowance(); balanceOf(); decimals(); name(); symbol(); totalSupply(). Write methods are also used. All of them charge a certain amount of so-called “gas” per transaction. Gas is a fee that is paid for storing contracts, using blocks (or conducting payment transactions), and for using the technology through which the transaction is carried out.

For each transaction, the sender sets the gas limit and price. To independently calculate the total cost of the commission for a transaction in ether, it is necessary to multiply the gas limit by its price. For example, if the gas limit is 50,000 units and the gas price is 20 Gwei, it means that the sender is willing to spend 0.001 ether to complete the transaction. Gas calculation formula: Gas limit * gas price = maximum transaction fee. For example: 50,000*20 gwei=0.001 ether.

After gas payment, one of the following methods is called: approve(), decreaseAllowance and increaseAllowance(), mint() and mintTo(), transfer() and transferFrom(). This call belongs to the wallet that initiated the transaction. After writing the token, it needs to be uploaded to the network and checked for methods and their operation. Since there are already templates for writing tokens correctly, it’s only necessary to specify the number of decimal values and the name of the token.

After uploading the token to the blockchain, it can be checked for its presence by the hash address, which will be automatically created by the blockchain. Next, immediately after performing all the operations with uploading tokens, it is necessary to download the Oracle, which will store the value of the rate of each of the tokens in the project. After that, it can be used tokens, that is buy a certain amount of tokens or send them to some wallet, to get their rates. However, without logic, these are just numbers that have some liquidity but are not involved in the blockchain.

To get a really working dApp, it’s necessary to describe the logic of the contract, which will contain the processing of deposit and credit operations. This contract inherits the methods of the token contract, but at the same time contains methods that are responsible for the logic of credit or deposit tokens and the system for credit repayment and receiving rewards for investing cryptoassets.

The scheme of contract interactions in the network is presented in Figure 3.

Contracts interact with each other through a chain, and the BorrowerContract uses methods to access the tokens for pricing, which in turn requests the Oracle, and the Oracle requests the network for information.

In order to be able to call any method in the blockchain when interacting with contracts, Web3.js is used – a set of libraries that allow JavaScript language developers to interact with Ethereum nodes using HTTP, IPC or WebSocket protocols. This tool makes it possible to use JSON RPC to contact a specific node of the Ethereum network or something similar to it and receive a response from the contract.

Every function is called using the provider. MetaMask was used to implement the platform. Interaction with it is carried out with the "@usedapp/core" library, which contains all the necessary methods for activating the provider in the project. Once a connection to MetaMask has been received in the local

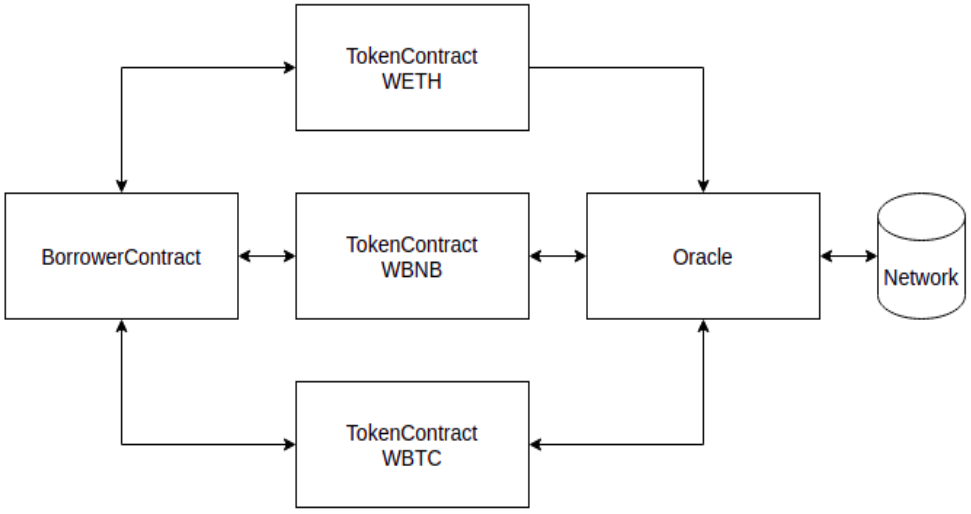

Fig. 3. Scheme of contract interaction

repository, the RPC access link is stored, allowing any requests to be made to the blockchain.

Results. As a result of the conducted research, the algorithm of blockchain operation of the lending and deposit platform was worked out.

To start working with the platform, the user needs to connect an electronic wallet. After connecting, the number of network tokens will be displayed to evaluate whether the user has enough tokens to pay for gas in the network. It also displays information about the wallet and a link to the scanner to check the transactions that have taken place for this wallet.

On the platform, it is possible to get the balances of crypto assets stored in the wallet. The platform works with Ethereum Coin, Binance Coin and Bitcoin.

When using the platform, it is possible to choose actions: credit or deposit. If the user wants to deposit tokens, he or she must first choose a token from the list of proposed ones. After selecting a token, the user's available balance and a field for entering the deposit amount are displayed. The required amount must be entered in the field for entering the number of tokens. And after that, the MetaMask plugin is run, in which the user gives consent that the contract will write off crypto-assets in the token from the wallet. As a result of the transaction, the user is needed to confirm the withdrawal of the token and its addition to the deposit. After the transaction is confirmed, the invested pool balance and rewards for the share in the pool are displayed.

Conclusions. Blockchain technology can be applied to any types and spheres of human activity. The paper studies the work of blockchain technology.

An overview of the popular blockchain platforms Ethereum and Fantom was made. The preference was given to the Ethereum platform. An analysis of smart contract development languages for these platforms was carried out. The Solidity language was chosen for the implementation of smart contracts. Since the system cannot work without a crypto wallet, it is suggested to use the MetaMask plugin. During the implementation of the platform, liquidity problems appeared, which were resolved and described. A contractual part of the service was also created, which will perform all calculations and store the values of credits and deposits.

Currently, there are still many open questions that require further research and analysis, the solution of which will allow to create more efficient platforms and ecosystems with the benefits of using blockchain.

REFERENCES

1. Seele, 2023. URL: https://seele.pro.

2. A. Jaradat, O. Ali, A. AlAhmad, Blockchain Technology: A Fundamental Overview, Springer Singapore, Singapore, 2022, pp. 1–24. URL: https://doi.org/10.1007/978-981-16-6301-7_1.

3. A. A. Monrat, O. Schelén, K. Andersson, A survey of blockchain from the perspectives of applications, challenges, and opportunities, IEEE Access 7 (2019) 117134–117151. doi:10.1109/ACCESS.2019.2936094.

4. P. Tasatanattakool, C. Techapanupreeda, Blockchain: Challenges and applications, in: 2018 International Conference on Information Networking (ICOIN), 2018, pp. 473–475. doi:10.1109/ICOIN.2018.8343163.

5. H. Wang, Z. Zheng, S. Xie, H.-N. Dai, X. Chen, Blockchain challenges and opportunities: a survey, International Journal of Web and Grid Services 14 (2018) 352 – 375. doi:10.1504/ IJWGS.2018.10016848.

6. X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future Generation Computer Systems 107 (2020) 841–853. URL: https:// www.sciencedirect.com/science/article/pii/S0167739X17318332. doi:https://doi.org/ 10.1016/j.future.2017.08.020.

7. S. Khan, F. Loukil, C. Ghedira-Guegan, et al., Blockchain smart contracts: Applications, chal lenges, and future trends, Peer-to-Peer Networking and Applications 14 (2021) 2901–2925. URL: https://doi.org/10.1007/s12083-021-01127-0.

8. A. Singh, R. M. Parizi, Q. Zhang, K.-K. R. Choo, A. Dehghantanha, Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities, Comput ers Security 88 (2020) 101654. URL: https://www.sciencedirect.com/science/article/pii/S0167404818310927. doi:https://doi.org/10.1016/j.cose.2019.101654.

9. T. Hewa, M. Ylianttila, M. Liyanage, Survey on blockchain based smart contracts: Applica tions, opportunities and challenges, Journal of Network and Computer Applications 177 (2021) 102857. URL: https://www.sciencedirect.com/science/article/pii/S1084804520303234. doi:https://doi.org/10.1016/j.jnca.2020.102857.

10. S. Lin, L. Zhang, J. e. a. Li, A survey of application research based on blockchain smart contract, Wireless Networks 28 (2022) 635–690. URL: https://doi.org/10.1007/s11276-021-02874-x.

11. T. Chen, Z. Li, Y. Zhu, J. Chen, X. Luo, J. C.-S. Lui, X. Lin, X. Zhang, Understanding ethereum via graph analysis, ACM Trans. Internet Technol. 20 (2020). URL: https://doi.org/10.1145/3381036.

12. C. Connors, D. Sarkar, Review of most popular open-source platforms for developing blockchains, in: 2022 Fourth International Conference on Blockchain Computing and Applications (BCCA), 2022, pp. 20–26. doi:10.1109/BCCA55292.2022.9922396.

МЕТОДИ ТА ЗАСОБИ СТВОРЕННЯ ПЛАТФОРМИ НА БАЗІ ТЕХНОЛОГІЇ БЛОКЧЕЙН

Галина Марчук
старший викладач кафедри комп'ютерних наук
Державний університет «Житомирська політехніка», вул. Чуднівська, 103, Житомир, Україна, 10005, pzs_mgv@ztu.edu.ua
ORCID: 0000-0003-2954-1057

Віталій Левківський
старший викладач кафедри комп'ютерних наук
Державний університет «Житомирська політехніка», вул. Чуднівська, 103, Житомир, Україна, 10005, levkivskyy@ztu.edu.ua
ORCID: 0000-0002-1643-0895

Олександр Кузьменко
старший викладач кафедри комп'ютерних наук
Державний університет «Житомирська політехніка», вул. Чуднівська, 103, Житомир, Україна, 10005, kkn_kov@ztu.edu.ua
ORCID: 0000-0002-4937-3284

Дмитро Марчук
старший викладач кафедри комп'ютерних наук
Державний університет «Житомирська політехніка», вул. Чуднівська, 103, Житомир, Україна, 10005, kipz_mdk@ztu.edu.ua
ORCID: 0000-0001-8675-8047

Антон Левченко
кандидат технічних наук, старший викладач кафедри інженерії програмного забезпечення
Державний університет «Житомирська політехніка», вул. Чуднівська, 103, Житомир, Україна, 10005, levchenko@ztu.edu.ua
ORCID: 0000-0002-4411-6465

Технологія блокчейн і ринок криптовалюти демонструють динамічний розвиток і привертають пильну увагу. Нововведення цієї технології полягає у тому, що інформація про транзакції більше не зберігається у централізованій базі даних, а передається на комп'ютери всіх учасників мережі, які зберігають дані локально. Першим додатком на базі технології блокчейн став біткоїн для однойменної криптовалюти. В останні роки він послужив основою для створення інших блокчейн-додатків, більшість із яких нині розробляється у фінансовій сфері. Фактично технологія блокчейн є універсальним способом зберігання й обробки інформації практично в будь-якій сфері діяльності.

У роботі проведено аналіз платформ Ethereum та Fantom. Визначено, що Ethereum демонструє більшу гнучкість, масштабованість і швидкість, даєзмогу використовувати різні електронні гаманці. Платформа надає у вільному доступі документацію для розробників. Ethereum увела поняття смарт-контрактів. Проведено аналіз мов програмування для написання смарт-контрактів та вибрано Solidity, яка підходить для різних варіантів використання, включаючи децентралізовані фінанси, NFT та гаманці. У роботі з блокчейном часто виникають проблеми з ліквідністю, які було вирішено та описано. Результатом даної роботи є аналіз методів та засобів реалізації технології блокчейн для створення платформи для кредитування і депозитів.

Технологія блокчейн відкриває величезний спектр можливостей, починаючи з грошових переказів і платежів і закінчуючи смарт-контрактами і зберіганням документів. Але варто відзначити, що технологія блокчейн має низку тонкощів, пов'язаних і з недостатньою вивченістю, розумінням технічної реалізації, гнучкістю.

Ключові слова: Blockchain, Ethereum, Fantom, Smart-contracts, криптогаманець.

Стаття надійшла 14.06.2023