UDC 004.75/.04
DOI: 10.15587/2706-5448.2021.239019
Article type «Reports on Research Projects»

**Oksana Herasymenko,
Valeriia Bachynska**

BLOCKCHAIN TECHNOLOGY FOR ACCOUNTING AND DISTRIBUTION OF CONTRIBUTIONS FROM A CHARITABLE FOUNDATION

The object of research is software for financial accounting and distribution of funds in a non-profit charitable foundation using smart contracts of the Ethereum platform. The work is aimed at designing and implementing a software application for a charitable foundation, which allows to exclude the misuse of funds of a non-profit charitable foundation.

The paper proposes an implementation of the Ethereum smart contract for the software of a charitable foundation. In the app, users can apply for financial aid or make a charitable donation. The request for financial support is confirmed by administrators to avoid abuse by those seeking help. Anyone who has a crypto wallet can become a sponsor by transferring funds from its account to a selected request. The sponsor remains incognito when making a charitable contribution. After collecting the entire declared amount, the funds are automatically transferred to the crypto wallet of the request's owner.

A smart contract and a corresponding decentralized web application for interacting with it were experimentally deployed, and their joint work was tested. To implement the smart contract, the Solidity programming language was chosen; developed smart contract converted to bytecode using remix. The resulting bytecode is ready to be deployed on the Ethereum platform. Decentralized web application for interacting with the contract is implemented using Web3.js, Vue.js. A rough estimate of the cost of deploying a project on the Ethereum platform has been made. The deployment and operation of smart contracts and web applications comes with a certain overhead, which is most dependent on the cost of ether. However, this is a justified price to pay for the transparency of transactions and the shadowing of the turnover of funds of the charitable foundation.

The results of the research can be used as a basis for further transformation into full-fledged software with the ability to submit all reporting documents to the relevant government agencies and sponsors.

Keywords: *smart contract, blockchain technology, distributed ledger, Ethereum platform, decentralized web application, charitable foundation, Solidity, Remix, MetaMask.*

Received date: 31.03.2021
Accepted date: 13.05.2021
Published date: 23.09.2021

How to cite
Herasymenko, O., Bachynska, V. (2021). Blockchain technology for accounting and distribution of contributions from a charitable foundation. Technology Audit and Production Reserves, 5 (2 (61)), 9–14. doi: http://doi.org/10.15587/2706-5448.2021.239019

1. Introduction

Charitable activities are an integral part of public life. Usually charitable foundations are engaged in the accumulation and distribution of funds for charitable activities. Their work greatly helps to solve problems in society, improves the living conditions of citizens and saves lives. However, along with a positive impact on public life, the field of charity remains an environment of abuse by fraudsters and unscrupulous entrepreneurs, because the thirst for easy money prompts them to invent new schemes for illegal enrichment. That is why the problem of accounting for funds collected by a charitable organization and control over their distribution remains relevant over time.

The solution to this problem is especially important in Ukraine. Such fraudulent activity in Ukraine has become possible due to the lack of real methods of monitoring and reporting on the implementation of charitable activities. Other countries have introduced special mechanisms for reporting and monitoring the activities of charitable organizations. For example, in Finland, charities have an obligation to provide data on income and expenses to any applicant. In Ireland, the Public Charity Register is publicly available, which contains full information on the income and expenditure of charitable foundations. In the UK and Spain, such information is necessarily published on the websites of organizations, and in the Netherlands and Luxembourg, reports on the activities of foundations are published in special editions.

These methods could improve the charitable fraud situation in Ukraine, but they are not ideal either. Any information about income and expenses can be destroyed or fake by competitors or members of the organization. This vulnerability often casts doubt on the existence of charitable foundations and causes distrust of the population.

This problem can be solved using blockchain technology [1, 2], which provides a record of a transaction without the possibility of replacing one digit or letter. Integrity, reliability, security of information – all this will be ensured by the blockchain, which will help create a charitable foundation that can be trusted.

The first registered fund using cryptocurrency was BitGive, an organization of American K. Gallipi, founded in 2013. There are dozens of similar projects in operation now. The most famous of them: GiveTrack, Medic Mobile, Water Project, Electronic Frontier Foundation, Code to Inspire, Common Collection, SENS. An example of a charitable blockchain project in Ukraine is the work of the Ukrainian Philanthropic Exchange and the NEM.io Foundation to collect donations to fulfill the New Year's wishes of children from the cancer center.

Nevertheless, the use of blockchain technology for financial accounting is a relatively new technological solution, therefore, it is relevant to study the possibilities of various blockchain platforms for the implementation of this task.

Thus, *the object of research* is software for financial accounting and distribution of funds in a non-profit charitable foundation using smart contracts of the Ethereum platform. And *the aim of research* is to design and implement a software application for a charitable foundation, which is designed to exclude the misuse of funds of a non-profit charitable foundation.

2. Methods of research

In this study, let's experimentally analyze the development of software (software) for organizing the collection and distribution of funds of a non-profit charitable foundation using smart contracts. The software uses the Ethereum blockchain platform [3] as a distributed ledger platform [4, 5].

The blockchain concept underlying the creation of the virtual cryptocurrency Bitcoin [6] turned out to be extremely useful in various areas, in particular, financial [7, 8], because distributed ledger technologies are most effective where there is a high level of transaction costs. Blockchain, as a chain of distributed ledgers of transactions, offers a fundamentally new decentralized approach to the formation and management of distributed data. The use of distributed ledgers, reliably protected by cryptography, allows to minimize the risks associated with dishonest, corrupt or criminal behavior of subjects.

In recent years, the introduction of smart contracts [9] into projects in the financial sector has gained popularity. A smart contract is a computer protocol designed to verify or enforce a contract, allowing to abandon the paper format for concluding transactions. The work of such contracts is provided by blockchain technology and a distributed ledger. It is the registry that ensures data integrity control, and also excludes the substitution, deletion of transactions or modification of their content «retroactively», which guarantees the absence of disagreements between the parties to the contract on the fulfillment of obligations.

At its core, a smart contract is a computer code, a record of sequential functions, which determines the parameters and logic of the fulfillment of the contractual obligations of the two parties. In 2014, Ethereum experts designed a special programming language Solidity to write the code of smart contracts. It has many similarities with JavaScript and is characterized by the ability to implement any computational function. The smart contract code is translated into the bytecode of the Ethereum virtual machine, which enforces the contract. As the terms of the contract are fulfilled, computers and blockchain nodes update the ledger.

In this study, the Ethereum platform was used for accounting and distribution of funds. It is one of the most developed in the field of smart contracts and has a wide range of functionality. The developed software consists of several components. Its core is several smart contracts, users interact with them through a web application.

In the app, users can apply for financial aid or make a charitable donation. For ease of understanding, let's call the former clients, the latter – sponsors, and any user can act both as a sponsor and as a client. To apply for assistance, the client fills out a form where it indicates the necessary information, attaches the relevant documents. These applications are sent to the e-mail of the charitable foundation. Administrators are responsible for regulating the operation of the application. They check the correctness of filling out the application, the accuracy of the documents and publish applications. If all documents are correct, then the application is published in the application and the fundraising process begins. Sponsors are identified only by their crypto wallet number. The basic domain model for this development is shown in Fig. 1.

The sponsor can find the application that interests him from the list of applications in the application and provide financial assistance. Funds are transferred to the account of the application until the entire required amount is collected. After all the money is collected, it is automatically sent to the client's wallet. The data flow diagram of this software is shown in Fig. 2.

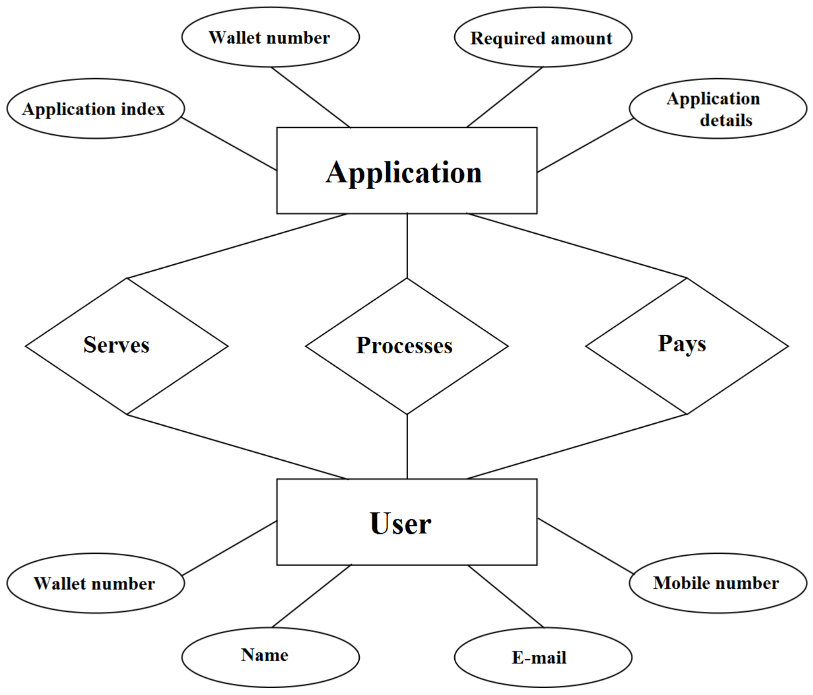

Fig. 1. Entity-relationship diagram for the subject area «Charitable Foundation»

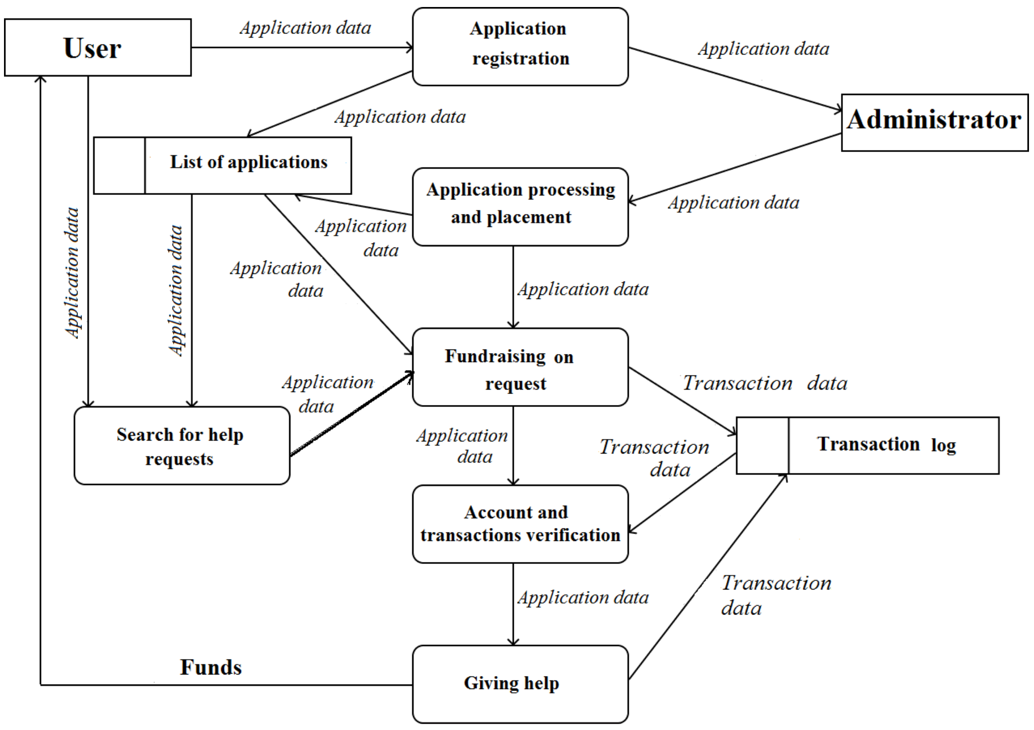

Fig. 2. Diagram of data flows of creation and processing of requests

The writing of smart contracts in this project was carried out in the Solidity language as the base for the Ethereum platform. The most common development environment for smart contracts in Solidity is the Remix IDE [10]. To use Remix, it is not necessary to perform any installation or configuration steps. Remix IDE is an online smart contract deployment application.

To deploy a smart contract as part of the study, it is necessary to create a local blockchain. For this, Ganache was chosen [11]. Ganache is used to configure the local Ethereum blockchain to deploy contracts, develop applications, and run tests. The program allows to perform all the actions that could be performed with the main blockchain. After creating a local blockchain, the user receives ten accounts, each of which has 100 ETH (ether cryptocurrency) in the account. With these accounts, it is possible to deploy and interact with the contract in Remix. Also in Ganache it is possible to track all transactions made.

In addition to Ganache and Remix, MetaMask was used in the study. MetaMask is a Google Chrome browser extension, an Ethereum wallet that allows web applications to interact with the Ethereum blockchain without running a full node. It can be used to store and send cryptocurrencies, as well as gain access to decentralized Ethereum applications. To use it, it is necessary to import your Ganache account into MetaMask. Then it is possible to carry out transactions.

When creating a web application, JavaScript, HTML, the Vue.js framework, and the Vuetify library were used. In order to connect the web application with the Ethereum blockchain platform, the API of the special Web3.js library is used. Web3.js is a set of libraries that allow to interact with a local or remote Ethereum host using an HTTP or RPC connection [12]. To use Web3, it is necessary first install the Node.js environment. The Web3.js library communicates with the local host using RPC. This library allows to work with a smart contract and do: deploying a contract, tracking contract events, converting a currency unit, sending ETH to a contract, getting a link to a contract, estimating the amount of gas to call a method [12].

In this project, the contract address and the Application Binary Interface (ABI) of the contract are used to communicate between the contract and the application. The contract ABI defines the interfaces of the smart contract. It describes how to interact with a smart contract. Fig. 3 shows a diagram of the interaction of the application with the blockchain.

During the research, two smart contracts were created in one sol file: CharitableFund and Request. CharitableFund acts as a container for all created requests – Request contracts. That is, this contract contains the function of adding a new order and a function for displaying all orders.

The functions for creating a new order and displaying all orders of the CharitableFund smart contract are shown in Fig. 4.

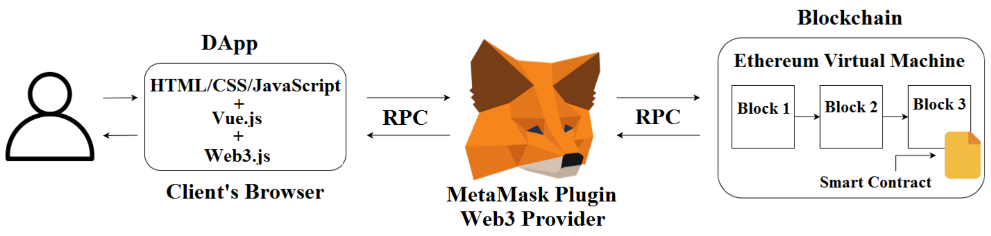

Fig. 3. Scheme of interaction of the application with the blockchain

```
function startRequest(
 address payable owner,
 string calldata title,
 string calldata description,
 string calldata link,
 uint durationInDays,
 uint amountToRaise
 ) external {
 uint raiseUntil = now.add(durationInDays.mul(1 days));
 Request newRequest = new Request(msg.sender, owner, title,
 description, link, raiseUntil, amountToRaise);
 requests.push(newRequest);
 emit RequestStarted(
 address(newRequest),
 msg.sender,
 owner,
 title,
 description,
 link,
 raiseUntil,
 amountToRaise
 );
 }
 function returnAllRequests() external view returns(Request[] memory){
 return requests;
 }
```

Fig. 4. Functions for creating a new order and displaying all orders of the CharitableFund smart contract

The Request contract is formed using the Charitable-Fund contract and handles all the methods that can be performed on requests. It contains the functions of transferring funds to the account of the application, checking the status of the application, paying out funds to the client, viewing the details of the application. The contribute() function is called when a sponsor tries to fund a request, and adds only the amount of funds that has already been raised. This function also calls the checkIfFundingComplete() function, which will check the status of the order (see the code in Fig. 5).

```
function contribute() external inState(State.Fundraising) payable {
 require(msg.sender != ownerRequest);
 contributions[msg.sender] = contributions[msg.sender].add(msg.value);
 currentBalance = currentBalance.add(msg.value);
 emit FundingReceived(msg.sender, msg.value, currentBalance);
 checkIfFundingCompleteOrExpired();
 }
 function checkIfFundingCompleteOrExpired() public {
 if (currentBalance >= amountGoal) {
 state = State.Successful;
 payOut();
 } else if (now > raiseBy) {
 state = State.Expired;
 }
 completeAt = now;
 }
```

Fig. 5. Contribute() function

Each project will first be in a fundraising state and will change the status of the request using the check-IfFundingCompleteOrExpired() function. Every time someone transfers funds on request, the status will be checked and changed depending on the conditions. For example, if the required amount has already been collected, or the deadline for fundraising has been exceeded. If the status of the application changes to «Successful» – the applicant can already receive its funds. The payOut() function ensures the transfer of funds to the owner (client) of the order, if the required amount has been collected (Fig. 6).

Fig. 7 shows the code for the Web3.js file that allows to load the Web3 instance to initialize the MetaMask extensions that will be needed to interact with the smart contract. After checking the browser in support of interaction with decentralized applications, a request for access to the Ethereum wallet. If the browser does not support the decentralized application or the request to access the account is denied, the user will not be able to open the application.

```
function payOut() internal inState(State.Successful) returns (bool) {
 uint256 totalRaised = currentBalance;
 currentBalance = 0;
 if (ownerRequest.send(totalRaised))
 {
 emit CreatorPaid(ownerRequest);
 return true;
 } else {
 currentBalance = totalRaised;
 state = State.Successful;
 }
 return false;
 }
```

Fig. 6. PayOut() function

```
import Web3 from 'web3';
if (window.ethereum) {
 window.web3 = new Web3(ethereum);
 try {
 // Запит на отримання доступу до акаунту, при необхідності
 ethereum.enable();
 } catch (error) {
 // Користувачу відмовлено у доступі до облікового запису
 }
} else if (window.web3) {
 // Застарілі DApp браузери
 window.web3 = new Web3(web3.currentProvider);
} else {
 // неDApp браузери
 console.log('Non-Ethereum browser detected. You should consider trying
MetaMask!');
}
export default web3;
```

Fig. 7. Application initialization code

For two written contracts, two files with the *.js extension are created, in which their metadata will be recorded, namely the ABI and the address for the main contract (Charitablefund) and the ABI for the Request contract. The program imports the Web3 library, writes the metadata constants, and creates a new instance of the contract with all the methods and events recorded in the contract at the specified address. It is these files that the application will refer to when executing the functions of a smart contract. The application interface and functionality are contained in a single App.vue file. To perform any operation, the application refers to the contract. When creating a request, the application calls the startRequest() function, which interacts with the startRequest() function of the smart contract (Fig. 8).

In order for sponsors to have the opportunity to pay for the application, the fundRequest () function was created (Fig. 9). It provides the operation of the «Fund» button. The sponsor enters the required amount and transfers funds upon request. The fundRequest() function uses the contract's contribute() method.

Deploying a decentralized application also implies deploying a smart contract. It will not be possible to deploy a smart contract without an Ethereum account. So, the first stage of deployment is creating an Ethereum account, that is, a wallet. The easiest way to create a cryptocurrency wallet is to install the MetaMask browser extension. After installing the extension, it is necessary to register.

With an Ethereum account, it is possible to work with a smart contract. The code needs to be compiled first. After successful compilation, it is necessary to deploy the contract. All this is done in the Remix environment using a previously created wallet. The smart contract itself will be deployed to the blockchain at each node.

```
startRequest() {
 this.newRequest.isLoading = true;
 crowdfundInstance.methods.startRequest(
 this.newRequest.owner,
 this.newRequest.title,
 this.newRequest.description,
 this.newRequest.link,
 this.newRequest.duration,
 web3.utils.toWei(this.newRequest.amountGoal, 'ether'),
 ).send({
 from: this.account,
 }).then((res) => {
 const requestInfo = res.events.RequestStarted.returnValues;
 requestInfo.isLoading = false;
 requestInfo.currentAmount = 0;
 requestInfo.currentState = 0;
 requestInfo.contract = crowdfundRequest(requestInfo.contractAddress);
 this.startRequestDialog = false;
 this.newRequest = { isLoading: false };
 });
```

Fig. 8. StartRequest() function

```
fundRequest(index) {
 if (!this.requestData[index].fundAmount) {
 return;
 }
 const requestContract = this.requestData[index].contract;
 this.requestData[index].isLoading = true;
 requestContract.methods.contribute().send({
 from: this.account,
 value: web3.utils.toWei(this.requestData[index].fundAmount, 'ether'),
 }).then((res) => {
 const newTotal =
parseInt(res.events.FundingReceived.returnValues.currentTotal, 10);
 const requestGoal = parseInt(this.requestData[index].goalAmount, 10);
 this.requestData[index].currentAmount = newTotal;
 this.requestData[index].isLoading = false;
 if (newTotal >= requestGoal) {
 this.requestData[index].currentState = 2;
 }
 });
 }
```

Fig. 9. FundRequest() function

To run a web application, it is necessary to install Node.js and the Web3.js library on the local machine where the application will be deployed. When deploying a charity application on a real blockchain, it is possible to link to Etherscan in order to show the transactions that occur within the application created by the charity fund application looks like a one-page site. On the website, the client has the opportunity to apply for help. After verification of the data by the administrators, the application is published in the application and the sponsor can transfer funds to the account. The client will receive financial assistance automatically when the required amount is collected.

3. Research results and discussion

In order to make an estimate of the cost of deploying and implementing an application that uses blockchain technology, it is necessary to estimate the cost of all development processes and support for its operation, namely:
– writing a smart contract;
– application development;
– hosting cost;
– cost of the domain;
– cost of using the resources of the blockchain platform;
– salary for application administrators;
– advertising to promote the project.

Considering that the smart contract and the application program were created as part of the study, there is no need to take into account the cost of these stages of work. Further development support and bug fixes will not count as well. The assessment was carried out for deployment on the resources of hosting providers in Ukraine.

The minimum cost of hosting, according to tariff plans that satisfy the conditions necessary for deploying the application, is about 10–15 USD per month. Domain prices range from 5 USD to 55 USD per year.

The cost of deploying a smart contract is 0.038 ETH, which is about 105 USD. For the placement of one request, that is, the deployment of one Request contract, the application uses on average 0.03 ETH and this is about 85 USD. However, this price will fluctuate depending on the value of the ETH cryptocurrency.

The cost of salaries for the fund's employees and the cost of advertising for the application were not estimated, since they depend on the market conditions at the time of deployment and should be additionally estimated at the time of making a decision on deployment. The cost of supporting and deploying a project is not unambiguous, because the cost of some processes can vary significantly over time. Nevertheless, even according to preliminary estimates, it can be said that for a non-profit organization in Ukraine, such amounts to support the project may turn out to be very significant.

The use of blockchain is advisable, if necessary, to carry out distributed control of assets and liabilities, the status of which dynamically changes depending on the occurrence of certain conditions, and the control of the state of assets is carried out by the most competing entities. The implementation of accounting and distribution of charitable foundation funds using distributed ledgers and smart contracts can be a significant factor in avoiding abuse and violations of legislation in this area. The advantages of such charitable foundations over conventional ones include:
– transparency of the distributed ledger system;
– no geographic restrictions (a distributed registry makes it possible to transfer funds from any country);
– the person who made the donation can verify the honesty of the distribution of funds;
– reducing the administrative costs of charitable organizations;
– increase in the speed of transactions (due to the absence of intermediaries);
– convenient accounting of assets.

However, the Ethereum blockchain platform as a creation platform for deploying smart contracts has its drawbacks. In particular:
– high cost of using the resources of this platform for non-profit organizations;
– low transaction processing speed. This is because each node is executing a smart contract in real time;
– platform skips imperfect projects, so there is a risk of launching a contract with an error;
– lack of clear documentation on working with the platform, as it is still actively developing;
– lack of specialists in the development of smart contracts.

However, even taking into account these shortcomings, the use of smart contracts for accounting and distribution of funds of a charitable foundation can become a starting point in de-obscuring this area and increasing public confidence in charitable projects.

4. Conclusions

During the study, two smart contracts and a decentralized web application were obtained, which have the minimum basic functionality for the accumulation and distribution of charitable foundation funds using the Ethereum blockchain platform. The main contract is deployed once, while the fundraising request contract must be deployed for each person seeking charity separately. Estimating the cost of deploying the proposed solution has shown that the overhead of using it can be significant for non-profit organizations and does not make sense when collecting insignificant amounts. Nevertheless, it is quite acceptable to use this solution on the Ethereum platform to collect significant amounts of money, because it minimizes the risks of misuse of funds by a charitable foundation.

References

1. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. *2017 IEEE International Congress on Big Data (BigData Congress)*. Honolulu, 557–564. doi: http://doi.org/10.1109/bigdatacongress.2017.85
2. B. Rawat, D., Chaudhary, V., Doku, R. (2020). Blockchain Technology: Emerging Applications and Use Cases for Secure and Trustworthy Smart Systems. *Journal of Cybersecurity and Privacy, 1 (1),* 4–18. doi: http://doi.org/10.3390/jcp1010002
3. *What is Ethereum? The foundation for our digital future*. Ethereum Foundation. Available at: https://ethereum.org/en/what-is-ethereum/
4. Buterin, V. *A Next Generation Smart Contract & Decentralized Application Platform*. Ethereum white paper. Available at: https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf
5. Buterin, V. *Ethereum: Platform Review. Opportunities and Challenges for Private and Consortium Blockchains*. Available at: http://www.smallake.kr/wp-content/uploads/2016/06/314477721-Ethereum-Platform-Review-Opportunities-and-Challenges-for-Private-and-Consortium-Blockchains.pdf
6. Satoshi, N. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Available at: http://bitcoin.org/bitcoin.pdf
7. Budman, M., Hurley, B., Khan, A., Gangopadhyay, H. (2019). *Deloitte's 2019 Global Blockchain Survey. Blockchain gets down to business*. Available at: https://www2.deloitte.com/content/dam/Deloitte/se/Documents/risk/DI_2019-global-blockchain-survey.pdf
8. Tapscott, D., Tapscott, A. (2016). *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World*. Portfolio, 365.
9. Hu, B., Zhang, Z., Liu, J., Liu, Y., Yin, J., Lu, R., Lin, X. (2021). A comprehensive survey on smart contract construction and execution: paradigms, tools, and systems. *Patterns, 2 (2),* 100179. doi: http://doi.org/10.1016/j.patter.2020.100179
10. Ethereum Foundation. (2021). *Remix – Ethereum IDE*. Available at: https://remix.ethereum.org
11. The Truffle Suite Team. (2021). *Ganache: one click blockchain*. Available at: https://www.trufflesuite.com/ganache
12. Ethereum Foundation (2021). *Solidity*. Available at: https://docs.soliditylang.org/en/v0.8.6/

✉***Oksana Herasymenko,*** *PhD, Associate Professor, Department of Network and Internet Technologies, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, e-mail: oksgerasymenko@gmail.com, ORCID: https://orcid.org/0000-0001-6804-2125*

Valeriia Bachynska, *Department of Information Systems and Technologies, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine, ORCID: https://orcid.org/0000-0001-9271-8105*

✉*Corresponding author*