Practical Implementation of Smart Contracts for Payment of Digital Goods

Liliia Shumyliak[a,b], Luboš Cibák[a], Sergey Ostapov[b] , Abdel-Badeeh M. Salem [c] and Yaroslav Skrypnyk [b]

[a] *Bratislava University of Economics and Management, Furdekova 16, Bratislava, 851 04, Slovakia*
[b] *Yuriy Fedkovych Chernivtsi National University, Kotsyubynsky 2, Chernivtsi, 58012, Ukraine*
[c] *Ain Shams University, El-Khalyfa El-Mamoun Street Abbasya, Cairo , Egypt*

Abstract

Smart contracts built on blockchain technology are changing the way traditional industries and businesses operate. They allow the terms of an agreement to be automatically executed without the need for a middleman, leading to decreased administration costs, more efficient processes, and reduced risks. This article is focuses on the practical implementation of smart contracts. It discusses the benefits of using smart contracts for companies, including increased security and transparency, reduced transaction costs, and automation of contractual processes. The article also describes the key elements of a smart contract, provides a description of the implementation of smart contracts on the online market of digital goods, and presents the developed algorithm of the system operation.

Keywords 1

Marketplace, smart contract, Ether, blockchain, cryptocurrency

1. Introduction

The financial sphere, like the rest of the business, has not escaped the integration of IT technologies [1, 2, 3]. Such technologies have not only accelerated all transactions, but also made them quite cheap and instant. If earlier payments were mainly made by banks and other relevant structures only in traditional currencies, then Ether, Bitcoin and other similar cryptocurrencies, which are developing very actively, have taken the leading place of universal currency. In addition, the integration of cryptocurrency and smart contracts into online stores as a payment option for intangible goods can include increased security, reduced transaction costs, and more efficient payment processing [4, 5].

Smart contracts are self-executing agreements, the terms of the agreement between the buyer and the seller are randomly written into lines of code. When a payment is made using cryptocurrency and a smart contract, the transaction is verified and recorded in a decentralized ledger, making it secure and tamper-proof. This will reduce the risk of fraud and ensure that the terms of the agreement are automatically fulfilled.

The relevance of the development of systems supporting the payment of products through the conclusion of smart contracts is also achieved by the fact that the integration of cryptocurrency and smart contracts into online stores with the option of paying for intangible goods will provide a more universal trading solution that allows users with basic financial assets in the form of cryptocurrency to pay product without conversion into ordinary money. This will help to obtain data for conversion, which will save the client money and in the long run increase the store's user base.

IntelITSIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security, March 22–24, 2023, Khmelnytskyi, Ukraine
EMAIL: lshumylyak@gmail.com (Liliia Shumyliak); l.cibak@buem.sk (Luboš Cibák); sostapov@chnu.edu.ua (Sergey Ostapov); abmsalem@yahoo.com (Abdel-Badeeh M. Salem); yskrypnyk@chnu.edu.ua (Yaroslav Skrypnyk)
ORCID: 0000-0003-3881-7924 (Luboš Cibák); 0000-0002-4139-4152 (Sergey Ostapov); 0000-0003-0268-6539 (A.-B. M. Salem);

2. Literature review

An overview of smart contracts, including an introduction to blockchains and smart contracts, an analysis of recent technological advances, a comparison of popular smart contract platforms, and a classification of different types of smart contract programs with examples is presented in [6]. There have been several recent studies on smart contracts, including comprehensive surveys of blockchain technology and in-depth evaluations of Ethereum smart contract programming vulnerabilities [7, 8, 9], verification methods on smart contract languages, and the experiences of teaching smart contract programming. Works [10, 11, 12, 13] provides an overview of the security and privacy challenges associated with smart contracts and highlights the need for new solutions to improve their security and privacy.

In finance, blockchain technology has the potential to significantly disrupt traditional financial systems and create new opportunities for innovation. . Blockchain technology is a rapidly growing industry and offers many exciting opportunities for the future of finance [14]. A comprehensive overview of the basics of blockchain, cryptocurrencies, smart contracts and fintechs is provided in [15].

3. Analysis of similar developments and detection of "weaknesses"

The main goal of developing a digital goods web store is to implement smart contracts that allow you to trade assets, real estate, shares and any other valuable things. They work simply and without intermediaries [16, 17].

Digital goods stores work in the same way as regular stores, except that there are no methods of delivery of products, no need for a customer address and no need to keep track of the number of products, because they are always available. Such stores can sell images, NFT tokens, keys, accounts, courses, plugins and other digital goods. Let's stop our attention on platforms that offer educational digital goods. Vseosvita, Prometheus, Udemy stores were chosen based on the criteria of popularity and positive rating.

An analysis of the proposed analogues in terms of functionality for authorization and the purchase process is presented in Table 1.

Table 1
Functional capabilities

Site Possibility of purchasing goods without authorization	Database of all users and their purchases	Non-cash payment support	Ether currency support	Possibility of purchasing goods without authorization
Vseosvita	yes	yes	no	No
Prometheus	yes	yes	no	No
Udemy	yes	yes	no	No

Attractive features of the Vseosvita store are tests for better learning of the material, as well as the ability to find a tutor. The disadvantages are that there are courses to which access should open after a certain period of time, but registration with payment of the course is already open. Therefore, the user cannot be sure of what he is buying.

The advantages of the Prometheus store are the variety of courses from professional teachers from all over the world. The shortcomings are that the site requires a real phone number for user registration and subsequent purchase of digital goods, which reduces privacy.

The benefits of the Udemy store are a large number of teachers and courses, agreements with large companies to provide free courses for their employees. The negatives are that due to free access to content creation, there is a situation where it is very difficult to find the best one among so many courses.

4. Proposed technique to reduce the risk of loss of customer privacy during online purchases

After analyzing similar products, we can conclude that all of them have such functions as cashless payment for digital goods, collection of buyer information such as surname and first name, email address, and user purchase history. These systems do not support the use of cryptocurrency to pay for digital goods, and a lot of personal information is stored in the database, which increases the risk of losing the privacy of service buyers.

To overcome the identified shortcomings, it is proposed to use Web3 and Blockchain technology to introduce maximum user privacy, a payment module for digital goods using the Ether cryptocurrency from the MetaMask wallet, a single set of products. Access to the purchased goods (for this development – courses) must be done by verifying the smart contract.

A modern SSR architectural pattern [18] was chosen for the implementation of an e-commerce website for digital goods. This approach aims to generate the full HTML on the server in response to the request (following the link), which avoids additional data requests, client-side template filling, as they are processed before the browser receives the response. Another factor that contributed to the choice of this pattern is that it integrates quite easily with the blockchain and the Web3 concept.

SSR works on the application's ability to convert HTML files on the server into a fully rendered page for the client [19]. The browser sends a request for information from the server, which instantly responds by sending the client a completely rendered page (Fig. 1).

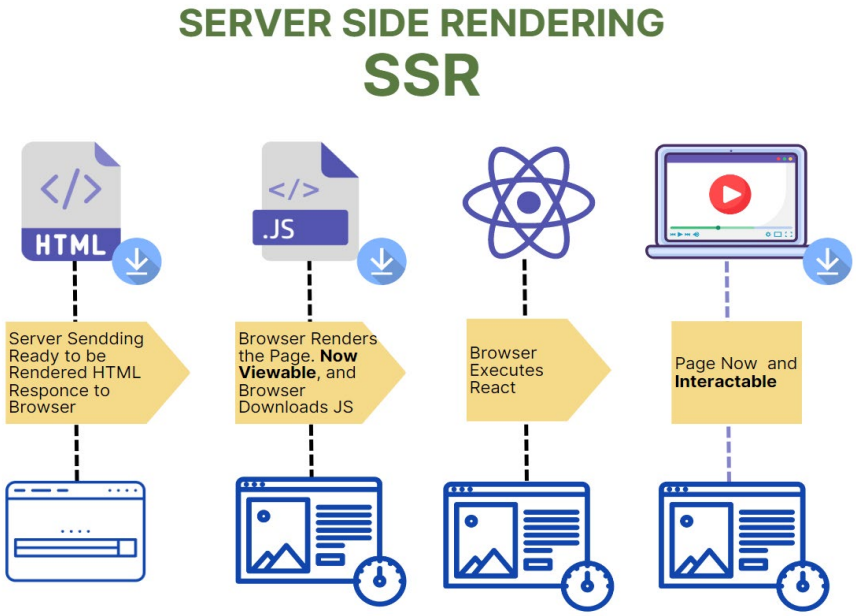

Figure 1: Generalized Model-View-Controller architecture

With SSR, users are unlikely to have to wait for CPU-intensive JavaScript to be processed before they can use the site. Even in cases where loading JavaScript files cannot be avoided, using this pattern can free up more memory for other processes.

Many modern frameworks, libraries, and architectures allow rendering of the same application both on the client side and on the server side.

Using SSR, we can get fairly fast rendering of FP and FCP (Fig. 2). Executing server-side logic results in reduced time-to-interactivity because only text and links are sent to the user. This approach opens up the possibilities of browser optimizations such as stream parsing of the document.

Thus, it can be concluded that this architectural template provides all the technological possibilities for building a flexible SPA in compliance with SOLID principles. In this case, this approach will ensure not only the successful implementation of the project, but also its further modification and development in accordance with changes in user requirements and information processing procedures.

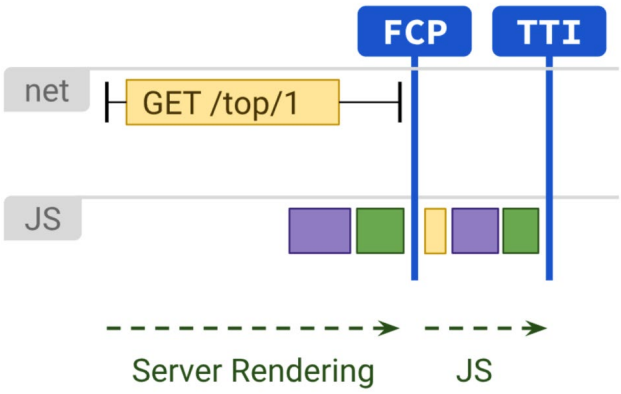

Figure 2: First Contentful Paint scheme

To create a web store, it is advisable to use: markup language for hypertext documents HTML, cascading style sheets CSS, browser programming language JavaScript, React library for the client part, Next.js framework for the server part, NPM package manager, Solidity programming language, MetaMask wallet.

5. Projecting virtual marketplace of digital goods

Virtual marketplace of digital goods created for the purpose of selling selected courses for Ether by entering into smart contracts with the user.

Users have access to the functions shown in Figure 3.

The following functional requirements were set for the system:

1. User authorization in the system: the system must present the possibility of user authorization through the MetaMask application and the definition of its appropriate role (administrator, buyer).
2. List of available courses for sale.
3. The possibility of saving information is implemented through Blockchain: the site does not have a separate database for saving information, all information about users is stored decentralized.
4. The system must support the purchase of courses by entering into smart contracts with payment through the MetaMask wallet.
5. Administrators must have access to the history of smart contracts concluded on the site.

The analysis of functional requirements made it possible to identify the following entities that will ensure the implementation of the software complex of the system. Figure 4 shows a diagram of the system classes.

The following classes can be distinguished in this figure:

- web3 – a work module that provides the necessary functions for establishing a connection to a decentralized Internet network in React components. The module contains the main functions: createWeb3State – provides information about the state of connection to the web3 network; useWeb3, useHooks, Web3Provider – return a wrapper component that contains access to web3 methods; loadProvider – a function that responds to changes in web3, checks the presence of the MetaMask application and authorizes the client; createAccountHook – a hook for changing the user in the system; createNetworkHook – a hook for changing the web3 network in which the client is located; createOwnedCourse – a hook that checks whether the course is owned by the client, createManagedCourse – a hook that allows course administration.
- CourseMarketplace – a class of working with conclusion of smart contracts for users and verification of smart contracts for system administrators. The class contains the main functions: purchaseCourse – executes the conclusion of a smart contract; stopContract, resumeContract, emergencyWithdraw, – manages the state of entering into a smart contract, goes through the stages of entering into a contract; activateCource/deactivateCource – administers course availability.
- utils – a work module that provides functions for loading existing contracts and displaying them to administrators. The module provides basic functions: loadContract – loads and creates a new course purchase contract;

- exceptions – an error handling module that covers possible errors and saves the system from a server crash.

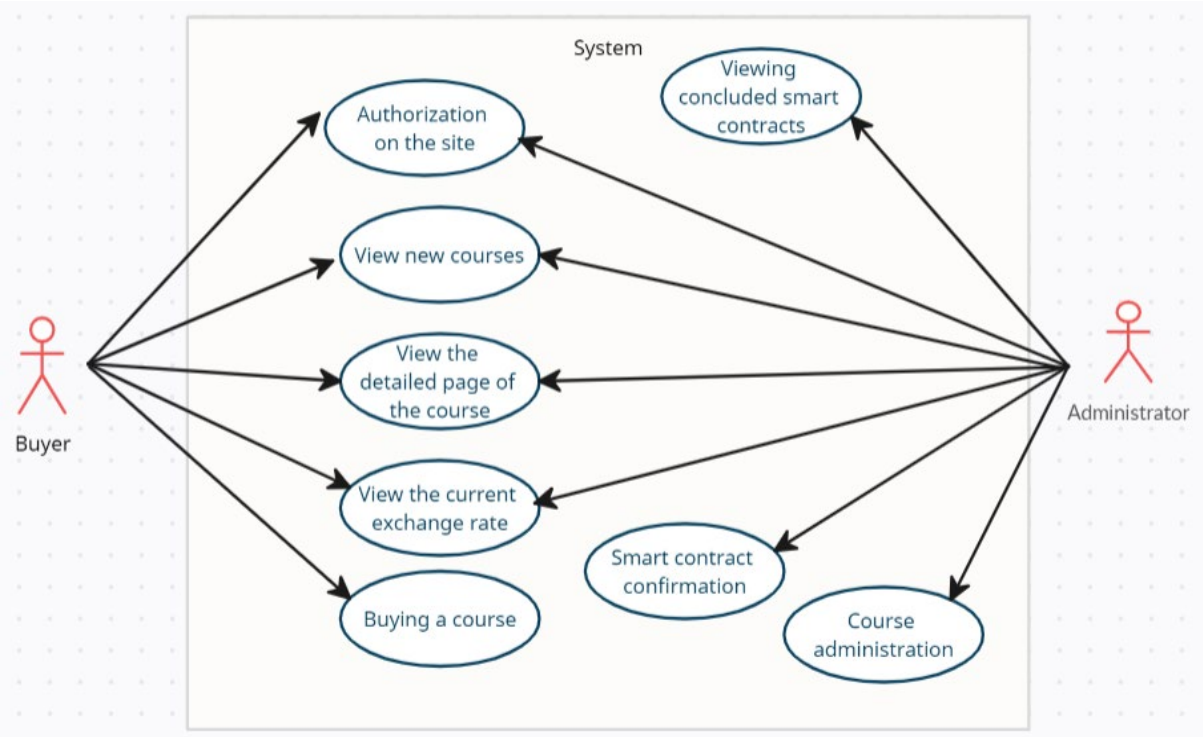

Figure 3: Diagram of precedents of system operation

Thus, this system implements the functionality of reviewing courses, concluding smart contracts, reviewing concluded contracts and purchased courses using the methods of the web3, utils, exceptions and CourseMarketplace modules.

The Blockchain scheme was used as a database. Blockchain technology was originally developed for use in cryptocurrency, but since then its potential uses have expanded to include a wide range of applications, including as databases.

One of the main advantages of using a blockchain as a database is its security. Because the database is decentralized and maintained by a network of users, there is no single point of failure that an attacker could exploit. This makes it ideal for storing sensitive information that needs to be protected from unauthorized access or tampering. Another advantage of the blockchain database is its transparency. All network users can see the data stored on the blockchain, and the ledger is open for review. This can help build trust among users, because transparency is important in such systems as online stores.

Blockchain technology allows us to store all the data necessary for the site in a decentralized manner. These are transaction metadata that are stored when entering into smart contracts. So, using the CourseMarketplace class and its courseHasOwner and coursesHasOwner methods, we can identify the person who has access to this course through its web3 contract, which contains the id of this course and the corresponding contract for it.

6. Development of algorithms for system operation

For modeling business processes, technological processes, an activity diagram has been developed (Fig.5).

After loading the main page of the online store, the user must log in, after that the system will identify him as a user or administrator. After the authorization process, the following actions can be selected: view courses, purchase courses and watch videos in purchased courses for regular users, and administer courses with view of executed smart contracts for administrators.

For authorization, the user must install the MetaMask browser application. After that, provide access to the site's wallet and confirm registration.

For course administration, the administrator must go to the administration section, select the required product from among the available ones, and activate or deactivate the product to conclude new contracts.

Figure 4: Diagram of modules and classes

When purchasing a product, the user selects a course, then gets acquainted with the price and fills out the purchase form. The site generates a new smart contract for signature and sets the Ether price at the given exchange rate. The user reads the contract in the MetaMask application and signs it. The currency is blocked for the time necessary to implement the contract. After a successful purchase, the user has access to view his purchased course.

When purchasing a product, the web3 and utils modules interact. The main methods used in this process are loadContract – loads and creates a new contract for the purchase of a course, Web3Provider – access to web3 methods, purchaseCourse – executes the conclusion of a smart contract.

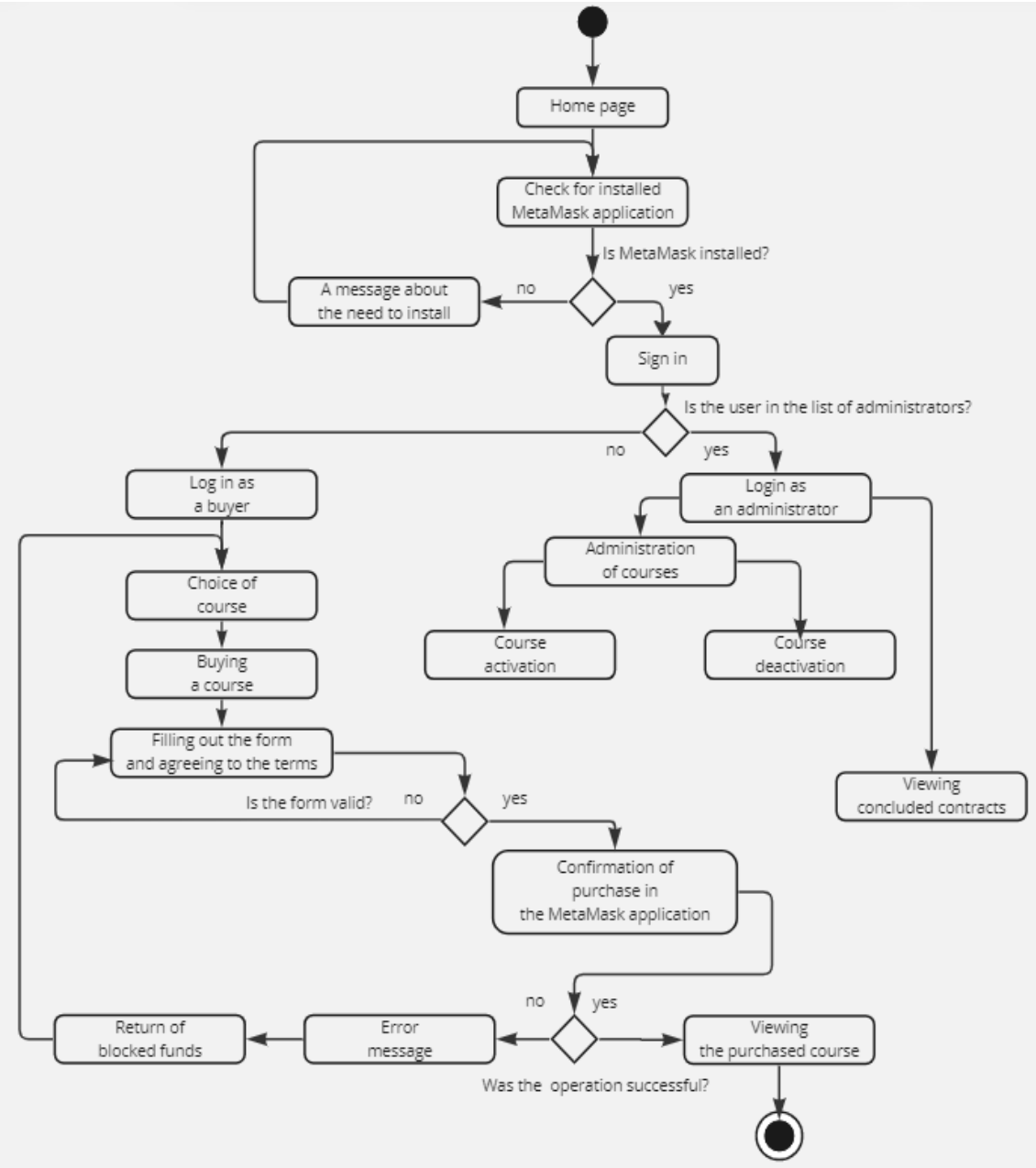

Figure 5: Activity Diagram

To implement the online store, the CourseMarketplace class was written – for communication with the Blockchain system through a web3-type network and the conclusion of smart contracts. This class was written using the Selenium programming language and used to generate a configuration file by compiling the class in the Remix IDE.

To establish communication with the web3 network, the useWeb3 JavaScript module was implemented, as well as the functions used by this module during authorization, purchase and verification of rights for actions on the part of the client.

In this way, we considered the algorithms of the main processes of the online store and the interaction of the system modules during the course purchase process.

The Digital Goods Web Store is a JavaScript-powered site with configuration files that specify the necessary npm packages, Solidity contract settings [20], and the required Node.js version. Before starting the system, it is necessary to change the crypto wallet data in the configuration files and install npm packages. The procedure in this case is as follows:

1. Creating an up-to-date CourseMarketplace.json file. This file contains settings for generating smart contracts. To generate a new file, it is necessary to compile the CourseMarketplace.sol contract

in IDE Remix, specifying the recipient of funds for the sale of goods. After that, replace the previous file with the newly generated one.

2. Editing the contents of the useAccount.js file. This file contains the settings for the list of store administrators. To add or remove administrators, you need to edit the adminAdresses object, adding the admin wallet id to it.

7. Results and discussions

Analysis of the developments of Udemy, Vseosvita, Prometheus showed that it is expedient to implement the following functions for this type of web-based platform: the use of Web3 and Blockchain technology to introduce maximum user privacy, a payment module for digital goods using the Ether cryptocurrency from the MetaMask wallet, a single set of products, access to purchased courses by verifying the smart contract.

The proposed development uses a blockchain database for secure, decentralized and tamper-proof entries.

Each block in the chain contains a unique set of data, and data integrity is maintained using cryptographic techniques that ensure blocks cannot be altered once added to the chain.

In order to ensure the possibility of remote work and independence from the user's operating system, this software complex requires the presence of a server and a user's computer. If necessary, the user's computer can also act as a server.

The developed program will be used on IBM PC/AT compatible computers. This subsystem works under the following operating systems: Windows 7, 10, 2012 Server and 2016 Server. It is also possible to install under *nix systems, such as Ubuntu, Fedora, RedHat.

Considering the fact that the subsystem does not require a significant amount of hardware resources for its operation, the necessary computer configuration will depend mainly on the choice of operating system.

Thus, the developed software code does not require specialized hardware, additional settings and tools for deployment other than standard for most modern sites.

8. Conclusion

In summary, the integration of cryptocurrency and smart contracts into online stores as a payment option for intangible goods has the potential to provide increased security, reduced transaction costs, and more efficient payment processing, making it a relevant and important development in the field of online commerce. Using cryptocurrency and smart contracts for payments is that can greatly reduce transaction costs. Since there is no need for intermediaries such as banks, the fees associated with traditional payment methods are eliminated. Additionally, cryptocurrency transactions are processed almost instantly, making the payment process faster and more efficient. In the case of intangible goods, such as digital products and services, the use of cryptocurrency and smart contracts can provide a more streamlined and secure payment process. This can help increase trust and confidence in online transactions, leading to more widespread adoption of these payment methods.

The proposed algorithm of the site's functioning and the determined order of interaction of classes and modules during the execution of the software code can be used in the development of an e-commerce website for digital goods. Ensuring high user confidentiality is achieved by connecting to the Web3 decentralized Internet network and using Blockchain technologies to save only the most necessary data for identifying the user and his purchased goods, and the proposed architecture will allow to further expand the functionality of the system in accordance with new requirements.

9. Acknowledgments

The work was funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I03-03-V01-00085.

10. References

[1] Y. Xu, The Strategy of How to Deeply Integrate Technology and Finance in the Internet Environment, J Environ Public Health (2022) 1-13. doi: 10.1155/2022/5018160.

[2] Samar Ali, Dana Alali, Haitham Nobanee, The Integration of Technology into Financial Services. (2022) 1-31.

[3] D. Bisht, R. Singh, A. Gehlot, S.V. Akram, A. Singh, E. Caro Montero, N. Priyadarshi, B. Twala, Imperative Role of Integrating Digitalization in the Firms Finance: A Technological Perspective, Electronics 11.3252, (2022). doi.org/10.3390/electronics11193252.

[4] I.A. Zeidy, The Role of Financial Technology (FINTECH), Changing Financial Industry and Increasing Efficiency in the Economy, COMESA, 2022.

[5] A. Geddes, T. Schmidt, Integrating finance into the multi-level perspective: Technology niche-finance regime interactions and financial policy interventions, Research Policy 49.6 (2020). https://doi.org/10.1016/j.respol.2020.103985.

[6] Z. Zheng, S. Xie, H. Dai, W.Chen, Xiangping Chen, Jian Weng, Muhammad Imra, An overview on smart contracts: Challenges, advances and platforms, Future Generation Computer Systems, 105 (2020) 475-491. doi.org/10.48550/arXiv.1912.10370.

[7] H. Atzei, M. Bartoletti, T. Cimoli, A Survey of Attacks on Ethereum Smart Contracts, General Relativity and Quantum Cosmology, (2017). doi.org/10.48550/arXiv.1706.06639

[8] R. Pise, S.Patil, A Survey on Smart Contract Vulnerabilities and Safeguards in Blockchain, International Journal of Intelligent Systems and Applications in Engineering, 10.3 (2022) 01–16.

[9] A. López Vivar, A. Turégano Castedo, A. Sandoval Orozco, L. García Villalba, An Analysis of Smart Contracts Security Threats Alongside Existing Solutions , Entropy, 22.2 (2020). doi.org/10.3390/e22020203.

[10] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, F. -Y. Wang, Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49.11 (2019) 2266-2277. doi: 10.1109/TSMC.2019.2895123.

[11] S. Moin, A. Karim, Z. Safdar, K. Safdar, E. Ahmed, M. Imran, Securing iots in distributed blockchain: Analysis, requirements and open issues, Future Gener. Comput. Syst. 100 (2019) 325–343. doi.org/10.1016/j.future.2019.05.023.

[12] A. Dhakal, X. Cui, Blockchain and Smart Contracts for Internet of Things: A Systematic Literature Review, (2018).

[13] J.Huang, K. Zhou, A. Xiong, D. Li, Smart Contract Vulnerability Detection Model Based on Multi-Task Learning, Sensors *,* 22.*5* (2022). https://doi.org/10.3390/s22051829

[14] G. Sladic, B. Milosavljević, S. Nikolic, D. Sladić, A. Radulović, A Blockchain Solution for Securing Real Property Transactions: A Case Study for Serbia, ISPRS International Journal of Geo-Information, 10 (2021). doi.org/10.3390/ijgi10010035.

[15] Y. Xu, H. Chong, M. Chi, A Review of Smart Contracts Applications in Various Industries: A Procurement Perspective, Advances in Civil Engineering, (2021). doi.org/10.1155/2021/5530755.

[16] G. Tereshchenko, I. Kyrychenko and N. Sharonova, Problems and Prospects of Practical Application of Blockchain Information Technology in Smart-Contracts, EasyChair preprint, 2019.

[17] S. Bock, Blockchain: Bitcoin, Ethereum, Smart Contracts, Cryptocurrencies and Everything about the Fintech Explained , CreateSpace Independent Publishing Platform, 2017.

[18] S. Connolly,Comparing Create React App vs. Next.js performance differences, 2022. URL: https://blog.logrocket.com/create-react-app-vs-next-js-performance-differences/.

[19] HEAVY AI, Server-Side Rendering, 2022. URL: https://www.heavy.ai/technical-glossary/server-side-rendering.

[20] Contract Metadata, Ethereum, 2021. URL: https://docs.soliditylang.org/en/v0.8.17/metadata.html/