Certification of digital content based on NFT technologies

Oleksii Turuta[1,†], Mariia Kozulia[2,*,†] and Vladyslav Nikitin[2,†]

[1] *Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine*
[2] *National Technical University «Kharkiv Polytechnic Institute», 2, Kyrpychova str., Kharkiv, 61002, Ukraine*

Abstract

The research focuses on creating an information and software solution for digital data certification in NFT, facilitating the purchase, sale, and transfer of NFTs. The thesis aims to develop a web resource that utilizes zero-knowledge succinct non-interactive arguments of knowledge (zkSNARKs) for securing and certifying digital objects. This approach ensures the protection of copyright for users' digital creativity on the Internet, allowing them to showcase their artwork to a broader audience and monetize their efforts .

The chosen architecture for the project is a Decentralized Application (DApp), consisting of three key components: client, server, and smart contracts on the blockchain. The software is designed to support multi-user functionality, ensuring fault tolerance, information security, and scalability. The technology stack includes Vue.js, Node.js, Solidity for smart contracts, MySQL for database management.

The developed website boasts a clear and user-friendly interface, making it accessible even to those with limited technical expertise. Thanks to this web application, users worldwide can certify digital data in NFTs, secure ownership, freely trade, sell, buy, and earn commissions on secondary market sales. An essential aspect of this solution is the use of zkSNARKs, ensuring the storage of anonymous data on the blockchain. This cryptographic technique enhances privacy and confidentiality while maintaining the integrity of the certification process. The application's design allows for easy scaling and the implementation of various blockchains as needed, providing flexibility and adaptability to different requirements.

Keywords

Certification, NFT, Blockchain, Smart-contract, zkSnarks

1. Introduction

Since the 1980s, services have been built on open protocols (for example, TCP, IP, SMTP, HTTP). This created stable conditions for the creation of an ecosystem of the Internet and web resources. After that, from the mid-2000s to the present, companies have created a second layer of their own closed protocols on top of the Internet's open protocols.

CLW-2024: Computational Linguistics Workshop at 8th International Conference on Computational Linguistics and Intelligent Systems (CoLInS-2024), April 12–13, 2024, Lviv, Ukraine
* Corresponding author.
† These authors contributed equally.
oleksii.turuta@nure.ua (O. Turuta); mariia.kozulia@khpi.edu.ua (M. Kozulia); vladyslav.nikitin@khpi.edu.ua (V. Nikitin)
0000-0002-0970-8617 (O. Turuta) ; 0000-0002-4090-8481 (M. Kozulia); 0009-0003-0506-2278 (V. Nikitin)

Web 3.0 is currently in an early stage of development where communities are encouraged and rewarded for maintaining and developing the underlying infrastructure. Decentralized Web 3.0 networks offer an alternative to the broken digital status quo. One of the many manifestations of Web 3.0 was the possibility of certifying digital content using Blockchain technology, and the certificates were called NFT [1,2].

One of the characteristics of certified digital content in NFT is full publicity in the Blockchain network. On the one hand, this is an advantage, but on the other, the question arises of how to certify documents or other content that may have hidden characteristics, for example, a certificate with a unique pass that provides access to Web3 games and others [3].

There is also the question of copyright protection for a certified digital object, how can it be protected if all properties are public and nothing prevents it from being copied? To date, there is no way to certify content with private properties, which significantly increases the risk of copyright theft of digital property [4,5].

The subject of the research is technologies for the development of a unique Web 3.0 platform to simplify the process of creation, purchase, sale and management of certified digital content in NFT. Thus, the goal is to create a unique platform for certification, purchase, sale, exchange of digital content in the form of NFT. Apply the resilience of smart contracts and provide the ability to create confidential certificates and secure copyright through the use of ZK-snarks technology and cryptographic encryption to certify content on the blockchain.

To achieve the goal, the following tasks are set:

- Analysis of technologies for certification and protection of digital content
- Development of an algorithmic description of the system
- Creation of software implementation

2. Overview of technologies used for certification and protection of digital content

The following technologies were considered and used for the formation of the digital content certification system:

1. zkSnarks (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) is a zero-knowledge protocol that can be used to prove that a subject possesses certain information without revealing it and without any interaction between the proving and verifying parties information. ZkSnarks systems can be additionally equipped with a zero-knowledge property, which ensures that the proof is done without revealing the intermediate steps. ZK-snarks consists of three algorithms: Key Generator G, Verifier P, Verifier V [6].
2. Grothl6 is a pair-based SNARK preprocessing system. Its design is aimed at minimizing the size of the proof and the cost of verification, which is achieved due to a small constant size of test prints and a fast verification check in a constant time [7].

3. ZoKrates is a tool for developing and verifying Zero Knowledge Proof (ZKP) in decentralized blockchain applications. ZoKrates allows developers to build apps that can fact-check [8].
4. Asymmetric encryption based on the RSA (Rivest–Shamir–Adleman) algorithm, which is presented in the table 1 [9].

Table 1
RSA algorithm

Public key
n is the product of two prime numbers p and q, e is the encryption key; the number is mutually prime with the Euler function НСД (e, $\varphi(n)$) $= 1$ і $e < \varphi(n)$
Closed key
$d \equiv e^{-1}(mod\ \varphi(n))$ – decryption key
Encryption
$C \equiv M^e\ (mod\ n)$
Decryption
$M \equiv C^d\ (mod\ n)$

3. System architecture, algorithmic and information support for certification of digital objects

3.1. System architecture of the software component for certification of digital objects

The architecture of Web 3.0 applications (or "DApps") is completely different from Web 2.0 applications. Decentralized applications are similar in many ways to smart contracts based on Ethereum. But still, there are significant differences between the two developments. If smart contracts are only related to financial transactions and have a limit on the number of participants at a specific moment in time, then "DApps" expands these boundaries and goes beyond the established rules - an unlimited number of users can participate in the application at the same time (although this will cause network load), and developers are not limited to the economic sector alone and invent utilities for the entertainment, music, game and other industries. The full list of programs can be found on the website, which is constantly updated by network members [7].

Thus, "DApps" are utilities that can be used to implement and in the future support CI/CD processes (Continuous Integration, or continuous integration, Continuous Delivery, or continuous delivery). For an application to be considered a "DApp", it must meet the following criteria: the application must be fully open source, run autonomously, without control.

The utility can be adapted and improved for users, but all changes must be decided based on the consensus of its participants.

To avoid any central points of failure, all application data and performance records must be cryptographically stored on a publicly accessible decentralized blockchain.

The app must use a token (Bitcoin or system-specific altcoin) that is required to access the app. Also, any contribution from miners should be rewarded with program tokens.

The application must generate tokens according to a standard cryptographic algorithm that acts as proof that the nodes are authentic (Bitcoin uses the Proof-of-Work algorithm).

For this project, the "Dapp" architecture (Fig. 1) [5, 10] was chosen, because this software is intended for multi-user work in Web 3.0, and the software should also be resistant to failures, provide the ability to scale (Fig. 2) [11].

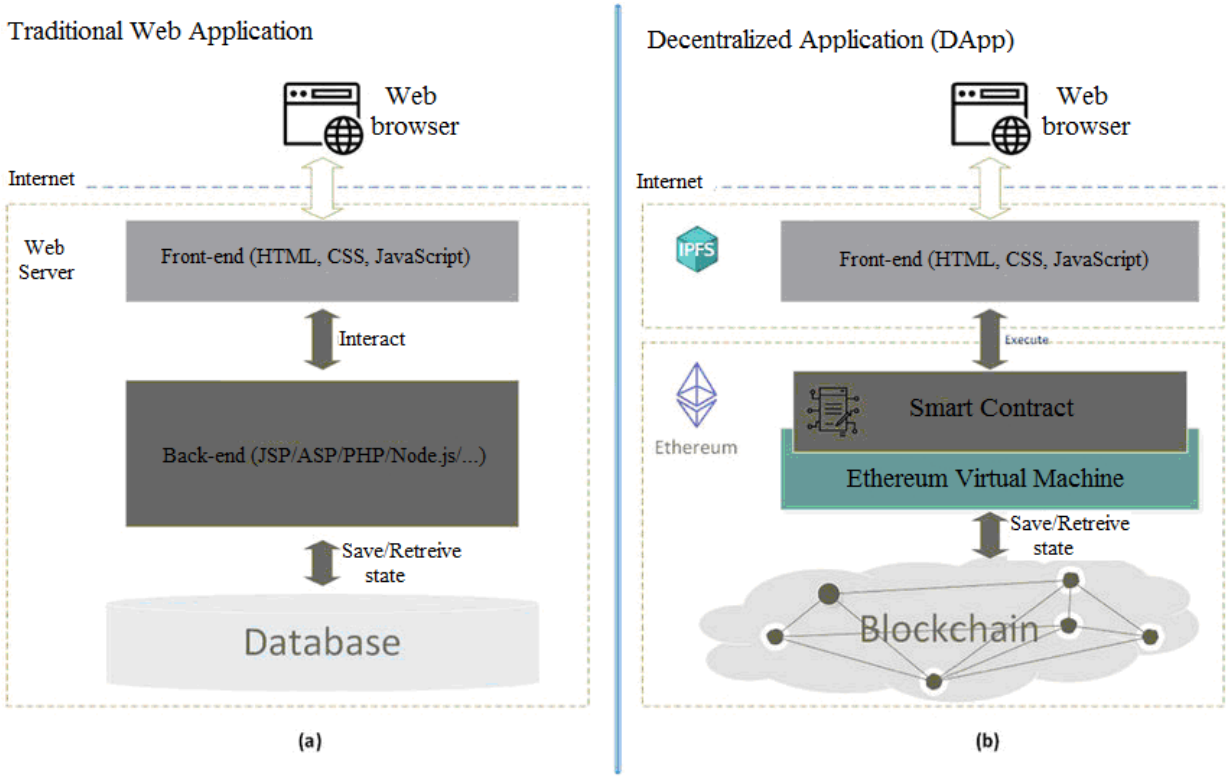

Figure 1: Comparison of client-server architecture and "DApp"

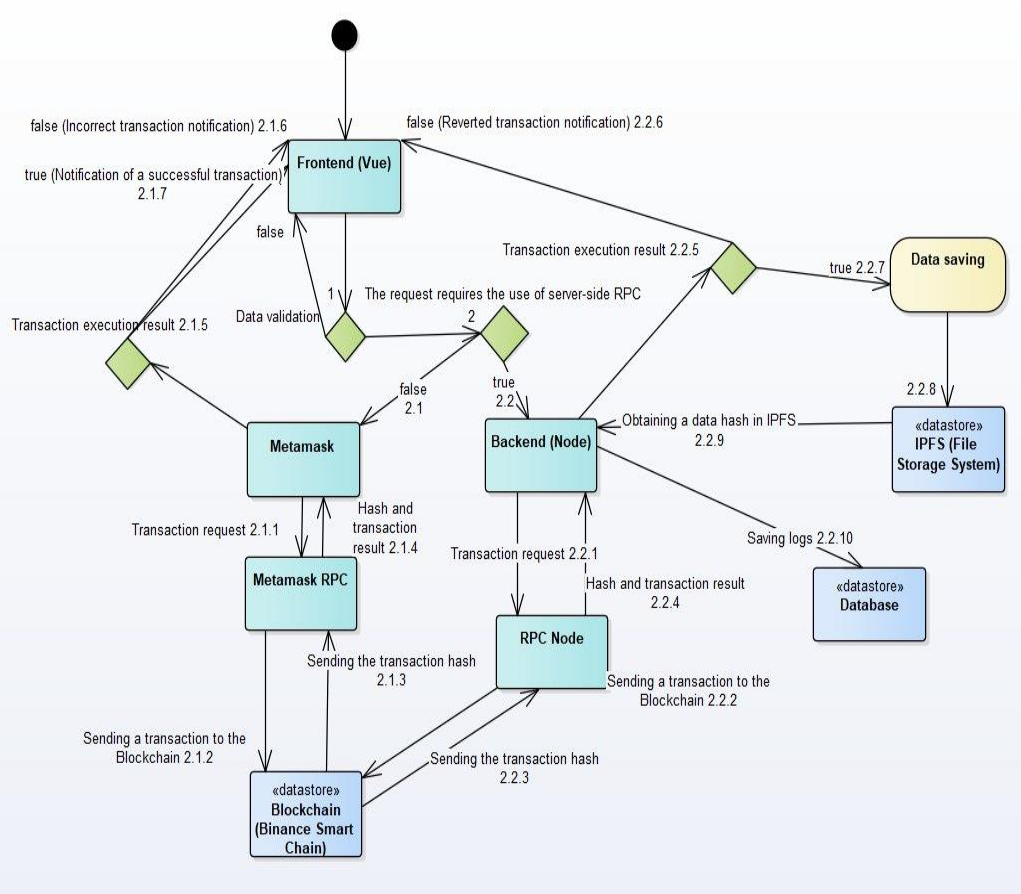

Figure 2: The general scheme of the algorithmic support of the website

This diagram depicts the sequence of interaction of components during the operation of the website. The scheme of the algorithmic support describes the general operation of the system and the interaction between the main components of the system.

3.2. Algorithmic and information support for the implementation of the NFT system

For this system, a sequence diagram has been developed, which emphasizes the temporal order of messages and is depicted as a table in which objects located along the X axis and messages arranged in time along the Y axis are presented. Sequence diagrams highlight the temporal order of messages.

The object or role that initiates the interaction is placed on the left, and the dependent objects or roles are on the right. Next, the messages sent and received by these objects are arranged chronologically from top to bottom along the vertical axis, creating a control flow association over time.

Sequence diagrams allow you to track the interaction of program components and objects over time. Consider the sequence of authorization in a web application and granting access to interact with a smart contract.

Authorization is performed using the "Metamask" browser extension. The user enters the website and clicks the Login button, after which a request appears for permission to use the

account previously selected in "Metamask" on the website. If the user has given permission to use his account (view balance, interact with smart contracts, initiate transactions), a request is made to the server, where a random nonce number of integer type is generated. After generation on the server, the nonce is sent to the user, who must sign the message in which this random number is stored. This is done in order to make sure that this user is the owner of this account. After the user has signed the message using a private key through the "Metamask" extension, the server checks the authenticity of this signature, if it is correct, the user is sent an authorization token, which is stored in the browser's local storage, otherwise - an error message (Fig. 3).

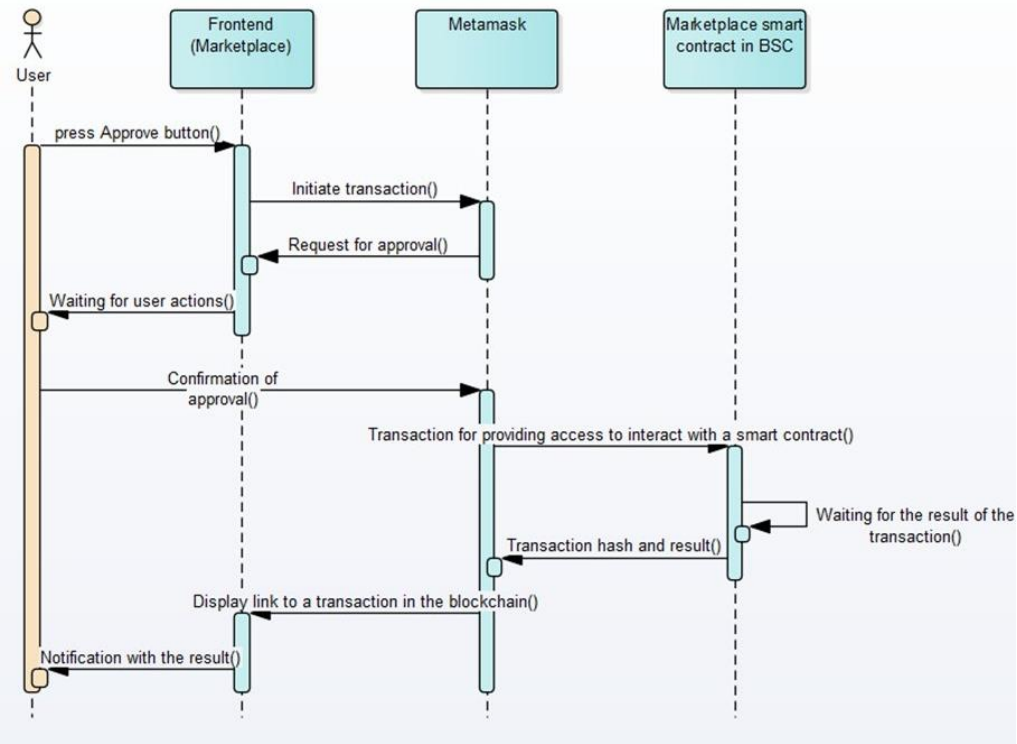

Figure 3: Constructed sequence diagram "Granting access to interact with a smart contract"

When interacting with a smart contract, it is first necessary for the user to grant access for this, that is, to send a transaction that confirms the granting of access. The system initiates the transaction, the "Metamask" window appears with a prepared transaction, which the user needs to confirm. After confirmation, the transaction is sent to the blockchain, after which the user receives a message depending on the result of the transaction.

4. Use of the application by users

4.1. Use of the application by the user

For the client's work with the software, Figures 4–11 show the main screen forms and the principles of working with them.

Figure 4 shows the screen form used for authorization. After clicking the Sign Up button, the Metamask extension window appears (where the user previously logged into his account). The client side sends a request to generate a random number. The server returns a random number and waits for a signature from the user, this is done in order to verify that the user is the owner of this account. After receiving the signature, the server verifies it, if the signature is valid for the given account, then returns a JWT authorization token. The user is redirected to his own profile.

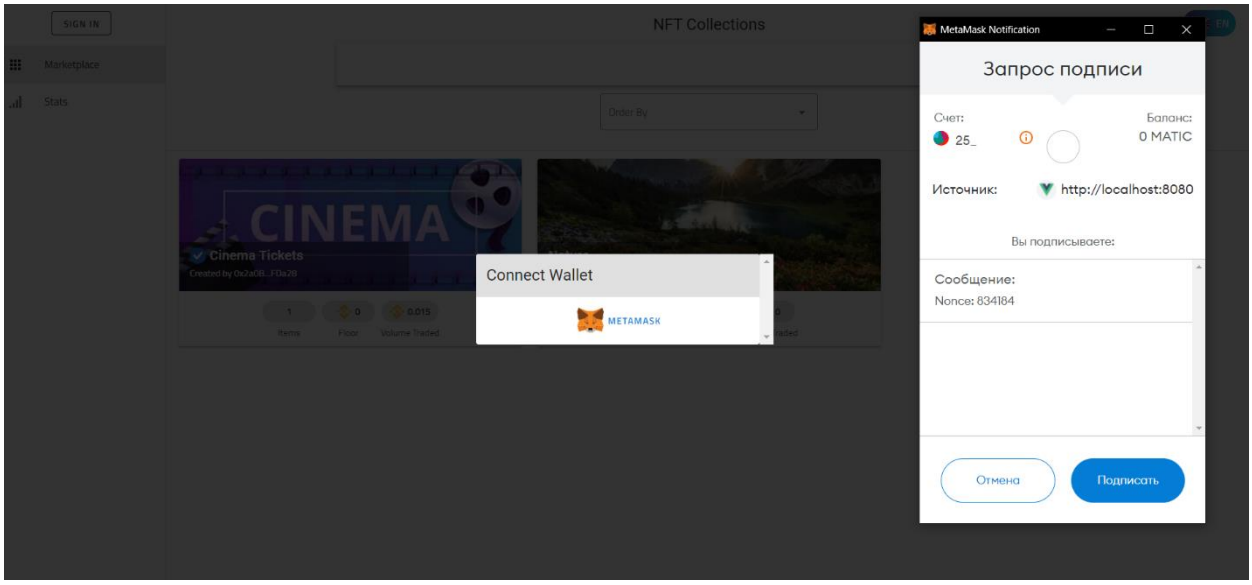

Figure 4: Screen form of user authorization through "Metamask"

The screen form responsible for displaying the user's profile, at the top of the screen, the profile avatar, nickname and address of the Metamask crypto wallet are provided in Figure 5. In the profile, sections are available with collections created by the user, certified digital data in NFT, NFT for sale, favorites NFT and transactions. A navigation menu is displayed on the left, which allows you to change the pages of the application. In the upper right corner there is a menu that allows you to change the language on the site. The section with user collections allows you to manage these collections, namely: edit information, place the collection on the market (make it possible to buy and sell NFT of this collection), check and collect funds earned from the purchase and sale commission.

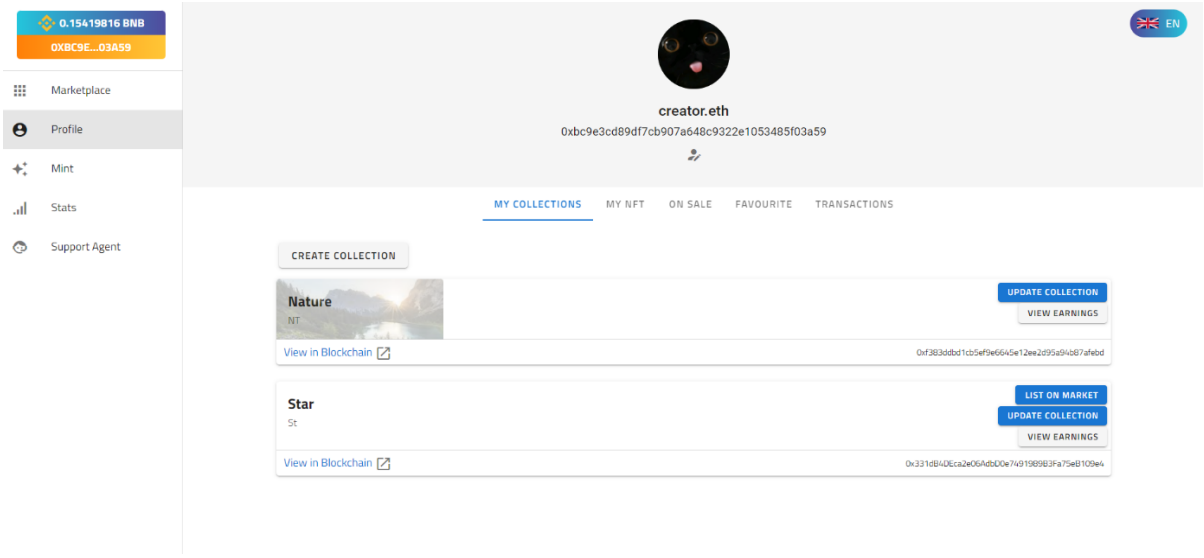

Figure 5: Screen form of the user profile and the user-created collections section

The screen form provided in Figure 6 depicts the certified digital data in the NFT owned by the user. User can offer NFT for sale, transfer, add to favorites. Each NFT contains information with data that has been certified.

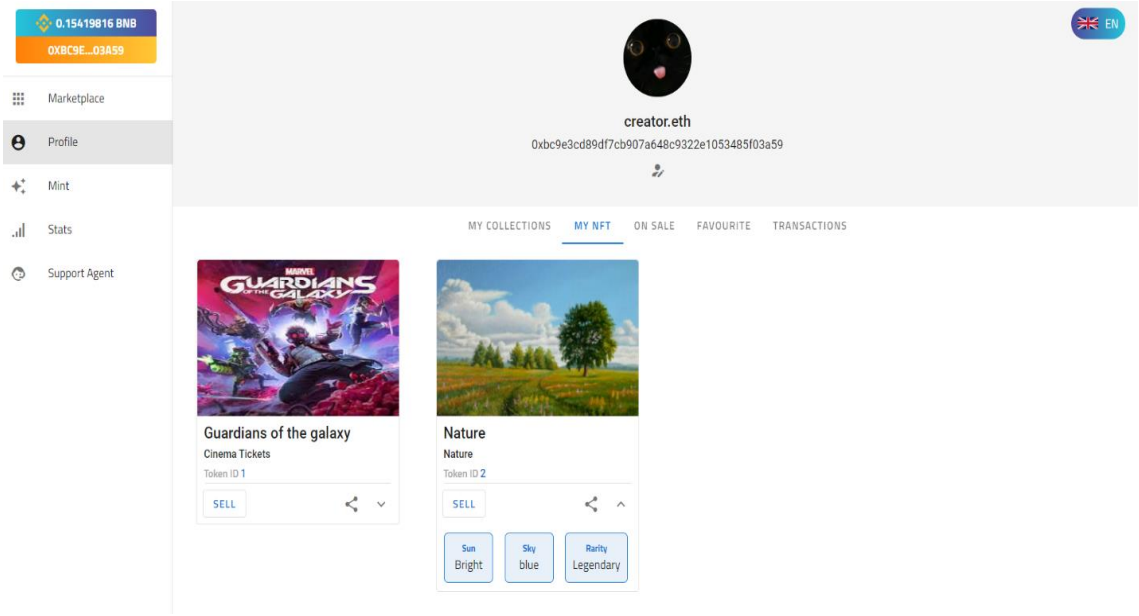

Figure 6: Screen view of certified digital data by the user

The screenshot shown in Figure 7 shows the certified digital data in NFT owned by the user and offered for sale. The user can cancel the sale by clicking the Cancel button.

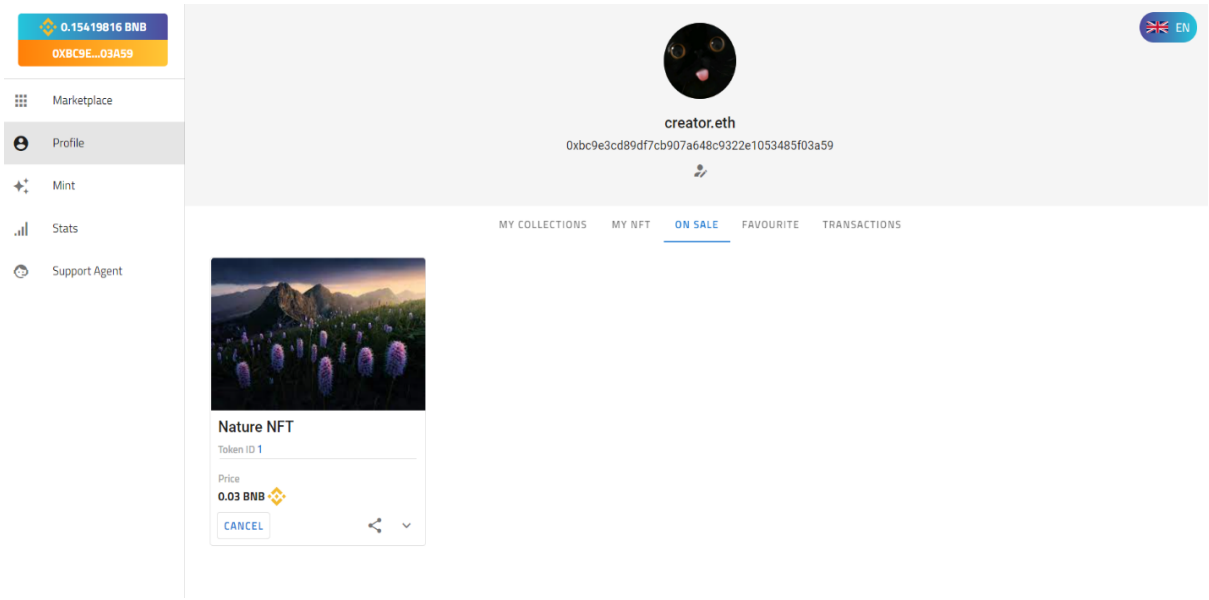

Figure 7: Screen view of certified digital data for sale by the user

Figure 8 provides a screen form depicting the editing of user data, namely the ability to change the nickname and avatar.

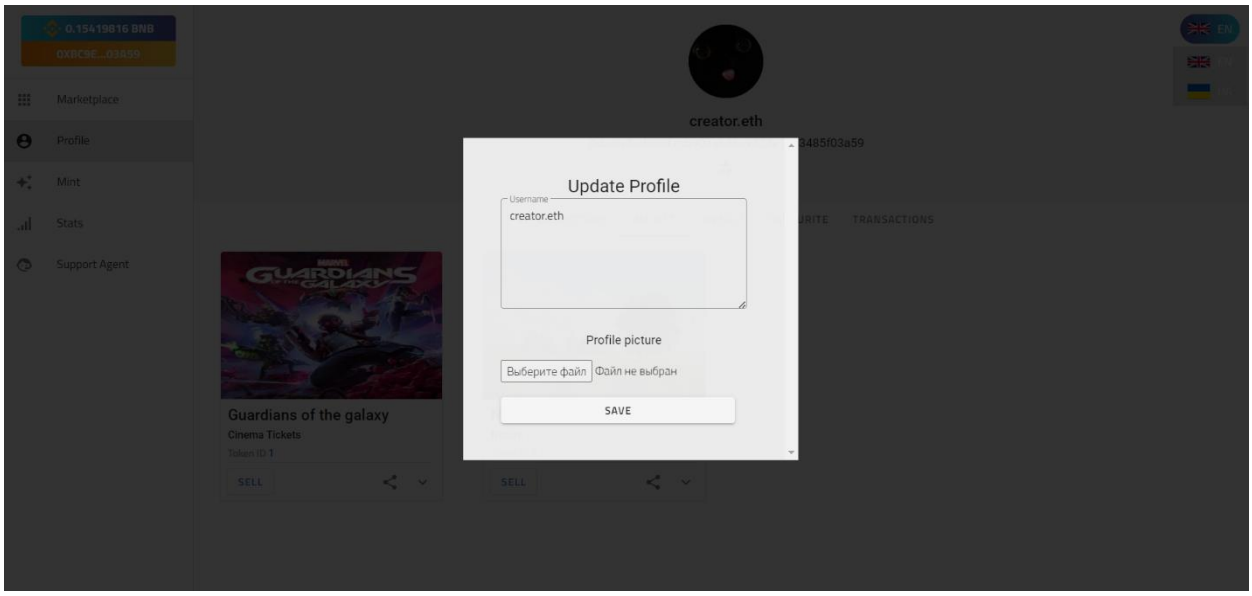

Figure 8: User data editing screen

The screen form in Figure 9 depicts the digital data certification form in NFT. First, the user uploads a file, it can be a picture, a video, a piece of music or a photo. Then fills in the fields with the name for the certificate and its description. After that, he chooses one of his own collections, to which the certificate will belong. The last step is to add properties to the certificate (this step is optional) and click the create button. After a short period of time, he

receives a notification with the results of the completed certification. The certificate appears among the certificates created by the user.

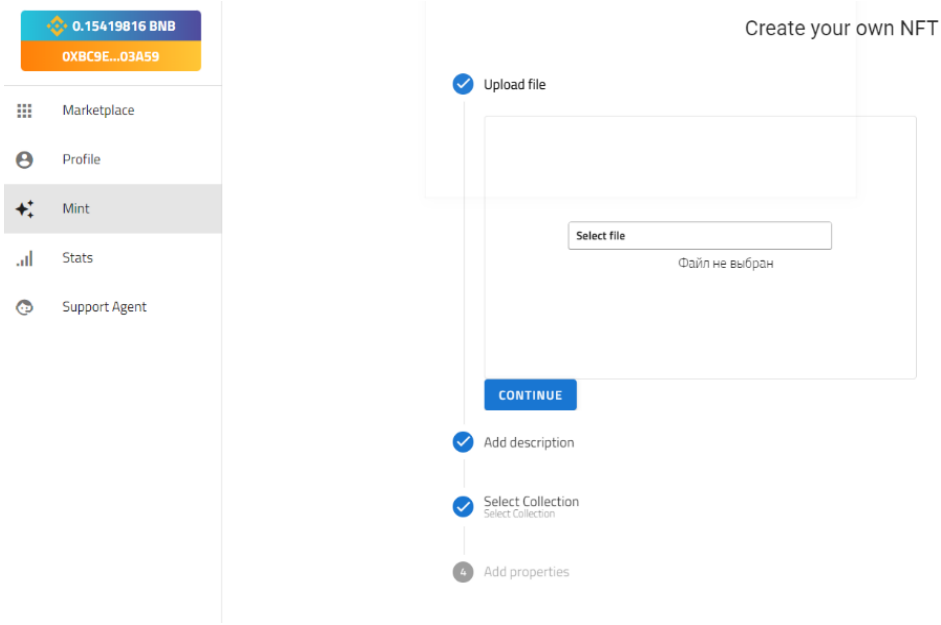

Figure 9: Screen form of digital data certification by the user

The Marketplace page (Fig. 10) displays collections of certified digital data that are placed on the marketplace, that is, NFTs of data collections can be placed for purchase and sale. It is possible to search and sort collections by various indicators. After clicking on a collection, a page with certified digital data of this collection will be opened.

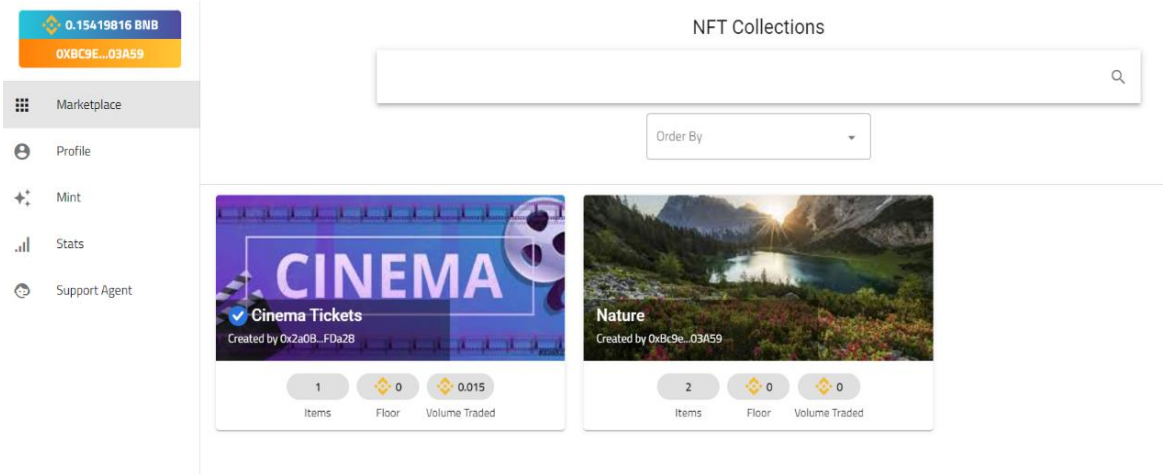

Figure 10: Screen view of collections of certified digital data that are displayed on the marketplace

Figure 11 shows a screen form that displays a certain collection page. The Items section displays NFTs available for purchase, the Activity displays transactions for that collection. On the right is a drop down menu for sorting NFTs.

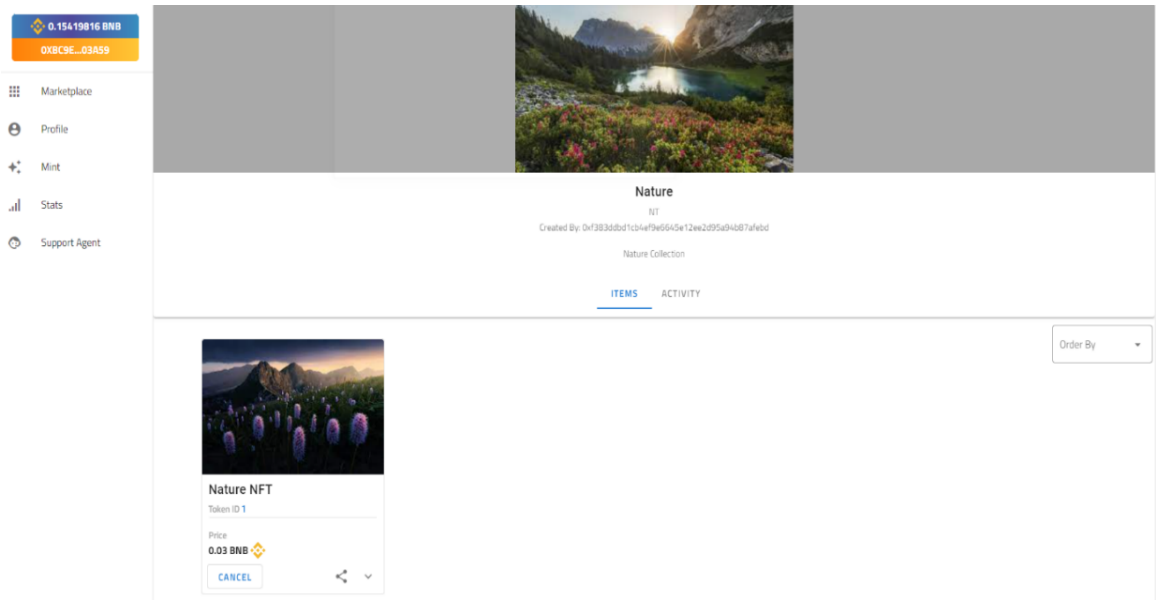

Figure 11: Screen view of certified digital data of a certain collection put up for sale

The Stats page in the on-screen form contains a list of the best collections in the marketplace, in which the ability to sort by various indicators is implemented.

A form has also been created that allows the user to create a collection for future certified digital data, after filling out the form the user clicks the Submit button.

4.2. Use of the application by a support agent

The support agent has access to additional functionality and functionality available to users. Figure 12 shows a screen form that displays the support agent menu.

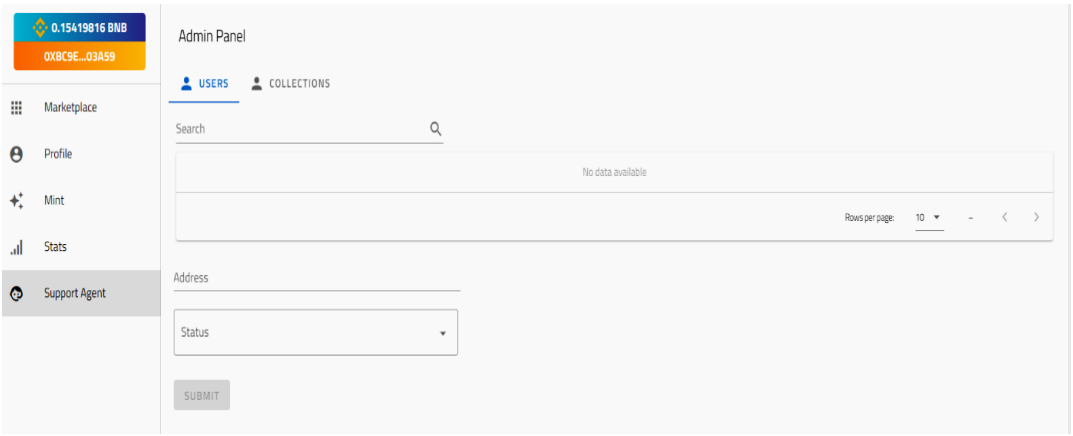

Figure 12: Screen form of the support agent

An agent can view and change the verification status for a collection or user. Verification status means that the collection owner or collection is known and enjoyed by a large number of users. In order to prevent fraudsters from passing off similar collections as originals, the possibility of verification was introduced. If a collection or user is verified, a blue tick appears next to the name or nicknames to confirm this.

5. Conclusions

On the basis of the analysis of the subject area and the considered methods of solving the problem of digital content certification, conclusions were made about effective solutions to the problem are demonstrated with the help of zkSnarks, Grothl6, ZoKrates, Asymmetric encryption based on the RSA (see table 1).

Algorithmic approaches have been developed that characterize the operation of the considered methods (see Fig. 2). To describe the operation of the system, UML diagrams are created that demonstrate possible cases and the operation of the main processes of the implemented software (see Fig. 3).

The developed application can be used as an initial image certification system (see Fig. 4-11). The software solution used a set of technologies – HTML5, SCSS, MySQL Vue.js, Node.js – that are an integral part of web application development.

Digital data certification in NFT significantly lowers the barriers to entry for certified digital data markets with a wide audience for artists, collectors, developers and ordinary people. There is no need for intermediaries to deal with. There is no need for the huge hassles and upfront investments required, for example, to set up production and distribution of a set of several thousand physical collectibles.

References

[1] Zhao, X., & Si, Y. W.. "NFTCert: NFT-based certificates with online payment gateway" IEEE International Conference on Blockchain (Blockchain). IEEE, (2021): p. 538-543. doi: 10.48550/arXiv.2202.09511.

[2] Khati, P., Shrestha, A. K., & Vassileva, J. "Student certificate sharing system using blockchain and nfts" In International Congress on Blockchain and Applications. Cham: Springer Nature Switzerland (2023):pp. 61-70. doi: 10.48550/arXiv.2310.20036.

[3] Dos Santos, R. B., Torrisi, N. M., & Pantoni, R. P. "Third party certification of agri-food supply chain using smart contracts and blockchain tokens" Sensors, 21(16), (2021): 5307. doi: 10.3390/s21165307.

[4] Bamakan, S. M. H., Nezhadsistani, N., Bodaghi, O., & Qu, Q. A decentralized framework for patents and intellectual property as nft in blockchain networks (2021). doi: 10.21203/rs.3.rs-951089/v1.

[5] Madine, M., Salah, K., Jayaraman, R., Battah, A., Hasan, H., & Yaqoob, I. "Blockchain and NFTs for time-bound access and monetization of private data" IEEE Access, 10 (2022): 94186-94202. doi: 10.3390/su151712870.

[6] Chen, T., Lu, H., Kunpittaya, T., & Luo, A. A review of zk-snarks. arXiv preprint arXiv:2202.06877. (2022). doi: 10.48550/arXiv.2202.06877.

[7] Tran Anh Minh Theoretical and practical introduction to ZK-SNARKs and ZK-STARKs Brno, Spring (2022): p. 146.
[8] JENG, Ya-wen; HSIEH, Yung-chen; WU, Ja-Ling. "Step-by-step guidelines for making smart contract smarter" IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA). IEEE, (2019): pp. 25-32. doi:10.1109/SOCA.2019.00012.
[9] Kadhim, S. A., Yas, R. M., Azize Abdual Rahman, S. A., & Abd, S. K. "Developing a new encryption algorithm for images transmitted through WSN systems" Eastern-European Journal of Enterprise Technologies, 124(9) (2023). doi:10.15587/1729-4061.2023.285261.
[10] Farnaghi, Mahdi & Mansourian, Ali. Blockchain, an enabling technology for transparent and accountable decentralized public participatory GIS. Cities. p. 105. (2020). doi: 10.1016/j.cities.2020.102850.
[11] V.V. Nikitin, M.M. Kozulia "Sertyfikatsiia tsyfrovoho kontentu u NFT" Tezy dop. XVI mizhnar. naukovo-prakt. konf. mahistrantiv ta aspirantiv. Kharkiv : NTU «KhPI», (2022): pp. 115-116.