BLOCKCHAIN-BASED NFT INTEROPERABILITY USING SMART CONTRACTS IN WEB3.0

Arya Abdul Azis
Departemen Teknik Komputer
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
5024201069@student.its.ac.id

Mochamad Hariadi
Departemen Teknik Komputer
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
mochar@te.its.ac.id

Reza Fuad Rachmadi
Departemen Teknik Komputer
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia 60111
fuad@its.ac.id

***Abstrak*—The emergence of Non-Fungible Tokens (NFTs) as a key component of the digital economy has sparked significant interest in their potential to revolutionize various industries, from art and entertainment to digital identity management. A fundamental aspect of maximizing the utility of NFTs involves ensuring their seamless operation across different blockchain platforms, which is addressed by the concept of interoperability. This thesis presents a comprehensive study and implementation of blockchain-based NFT interoperability using smart contracts within the framework of Web3.0. The research primarily focuses on developing a smart contract architecture that not only supports the standard features of NFTs but also facilitates their cross-chain interactions. By leveraging the Ethereum blockchain and utilizing the ERC-721 standard, this work establishes a robust protocol for NFT creation, transaction, and management that ensures secure, transparent, and efficient interoperability. Key aspects include the design of a decentralized application (DApp) that interacts with smart contracts to mint, manage, and transfer NFTs across blockchain boundaries. The implementation demonstrates the practicality and efficiency of the proposed system in a controlled testnet environment. Future work might explore scaling solutions, enhanced security features, and the integration of additional blockchain platforms to extend the reach and applicability of interoperable NFTs in the expanding universe of Web3.0.**

***Keywords*—Non-Fungible Tokens (NFTs), Blockchain Interoperability, Smart Contracts, Web3.0, Ethereum**

I. INTRODUCTION

In recent years, NFTs have come into the spotlight for both industry and academia. Data indicates that the daily transaction volume of the NFT market reaches approximately $4.592 billion USD, while the total daily transaction volume of the crypto market is around $341.017 billion USD. Non-Fungible Tokens (NFTs) are digital assets that represent objects like art, collectibles, and in-game items. These assets are traded over the internet, mostly with cryptocurrencies, and are typically embedded in smart contracts on a blockchain. NFTs are unique, making them non-interchangeable with similar objects, which makes them ideal for uniquely identifying something or someone. Although NFTs promise a significant impact on the current decentralized market and future business opportunities, the technology is still in its infancy. There are several challenges that need to be carefully addressed, and many significant opportunities that need to be seized.

NFTs, backed by blockchain technology and smart contracts, offer tremendous opportunities for the creative industry, though their presence has disrupted the market. However, NFTs also have limitations stemming from the underlying blockchain technology. Blockchain itself ensures trust within its distributed system by relying on computers (often called "miners") to solve complex mathematical problems. One major challenge in the evolution of blockchain technology is interoperability. Although blockchains provide robust and reliable solutions, the various types and variants of blockchains currently in existence often struggle to interact and communicate effectively. This implies that smart contracts or NFTs developed on one blockchain may not be compatible or recognized by another type of blockchain. This limitation restricts the blockchain's ability to grow and integrate with larger industry systems, such as finance, healthcare, or international business.

Blockchain and Smart Contracts are crucial foundations in realizing the vision of Web3.0. Blockchain provides transparency, security, and trust by storing data on a decentralized network, while Smart Contracts enable the automation of transactions and agreements in the digital world without intermediaries. The combination of these technologies could lay the foundation for a new era in the digital world, where interactions are more secure, transparent, and seamless.

The discussion in this paper begins with a presentation on other research (Section II). It then continues with an explanation of the architecture of the system developed (Section III). Based on this, we show the results of testing and analysis (Section **??**). Finally, conclusions from the research conducted are presented (Section V).

II. RELATED RESEARCH

Several other studies have been conducted, as formulated by Wang et al. [1], which suggest that NFTs can transform the digital or virtual asset market. The results of this research provide an in-depth understanding of NFT technology, its potential, and the challenges it faces. Additionally, research

results from Zheng et al. [2] have yielded knowledge in the form of a review on the latest smart contract technology, challenges in various aspects of creation, deployment, execution, and resolution of smart contracts, comparisons of several major smart contract platforms, and reviews on smart contract and blockchain technology. Cited from research previously written by Malik et al. [3], the study examines blockchain technology's impact on the creative industries, such as music, graphic design, gaming, and software. In this journal, they highlight how NFTs (non-fungible tokens) and smart contracts offer exciting opportunities for the creative industry. Although this technology has created significant excitement in the market, amidst the hype, real value emerges for the industry. Traditionally, creators in the creative industry often had to rely on powerful intermediaries to distribute and monetize their creations. However, with the advent of NFTs and smart contracts, creators can now be closer to their consumers/buyers of content. Additionally, this journal also explores the market share and "transaction costs" creators face when distributing their creative content and how smart contracts and NFTs can change market dynamics by reducing these costs.

III. ARCHITECTURE

A. Arsitektur Sistem

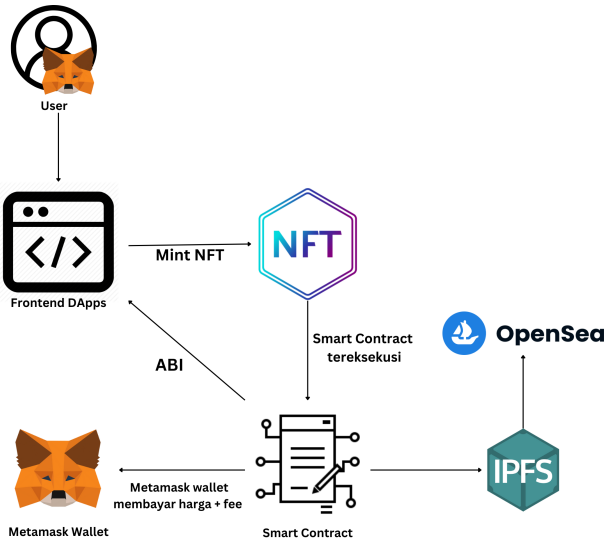

Figure 1. Desain sistem

In the development of this smart contract system to function as intended by the author for completing this thesis, the author has designed a system architecture. In this system architecture, there is a role for the user who can become the owner of a token, and the user is the party with limited access to a token within a specified time. There is also a front end for the minting process of NFT tokens. To access the front end, both the owner and the user must connect using a Metamask wallet to interact.

On the front end, it connects with the smart contract when a user performs Minting, where the process involves the user uploading a digital asset they own to IPFS through a backend application, which then returns a Content Identifier (CID). A CID is a file address within IPFS used to access that file. The obtained CID is then uploaded to the blockchain network and becomes a token. Data about the available NFTs can be directly accessed by the front end through the smart contract on the blockchain.

B. Proses Minting

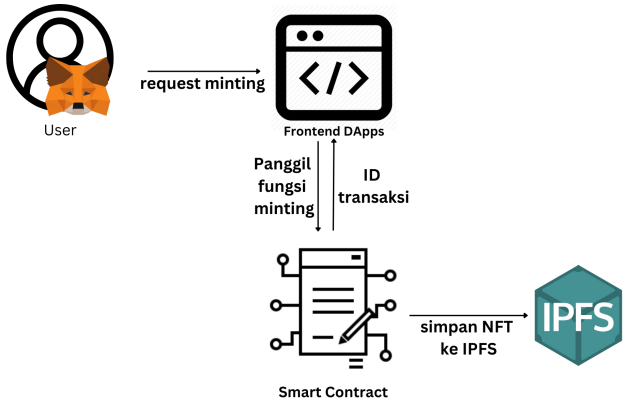

Figure 2. Desain sistem

The minting process is the initial step where a new NFT is created on a platform. During this process, various important details about the NFT must be defined, including its name, description, category, available supply, and the visual asset representing the NFT, which could be a 2D image, 3D model, or video. The initial price, the collection that includes the NFT, and other attributes must also be determined. Once the minting process is complete, the data from the newly created NFT is stored in the InterPlanetary File System (IPFS), which allows for decentralized storage and access to data. This storage ensures that all related information, such as ownership recorded in the NFT's attributes, can be accessed permanently and securely.

C. Metode Yang Digunakan

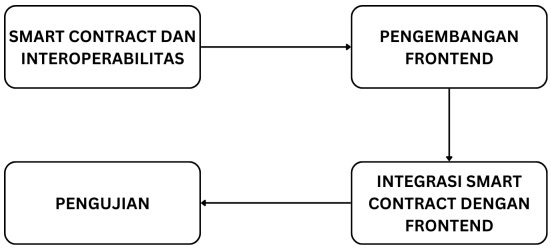

Figure 3. Metodologi Pengerjaan

1) Smart Contract dan Interoperabilitas: In this stage, the author develops a smart contract system used for creating interoperable tokens using the Solidity programming language. This smart contract system acts as a bridge to process requests generated from user interactions on the frontend to the

blockchain network. Upon compiling the smart contract, an Application Binary Interface (ABI) is generated, which can be used on the frontend for encoding and decoding data, allowing the frontend application to send precise instructions to the smart contract and understand the data returned by it. The smart contract developed by the author includes several core functions for the developed token.

2) Pengembangan Frontend: To integrate a React application with the Ethereum blockchain, we utilize two primary libraries: ethers.js and web3.js. These libraries provide the necessary functionality to interact with the Ethereum blockchain, though they approach it slightly differently. Ethers.js is known for its minimalistic API and ease of use, which is particularly suitable for projects with simpler needs and a focus on reading and writing data to the blockchain. This library offers powerful functions for interacting with smart contracts, such as sending transactions, reading contract statuses, and handling event notifications.

3) Integrasi Smart Contract dengan Frontend: This stage enables applications built with React.js to interact directly with smart contracts deployed on the Ethereum network. To achieve this integration, we use the ethers.js library, which provides a clean and user-friendly interface for communicating with Ethereum.

Once the smart contract is developed and deployed, the ABI (Application Binary Interface) of the contract is used to construct a contract instance within the React application. The ABI allows the frontend to understand what functions are available in the smart contract, including the variables and data types used. With this information, ethers.js can invoke these functions, such as safeMint or transferOwnership, according to the logic defined in the contract.

4) Pengujian: This stage includes several steps for testing the smart contract, namely unit testing and integration testing.

- Unit Testing This testing focuses on validating each component individually. In the context of a React.js frontend, this means testing components separately to ensure that they behave as expected. Unit testing is also conducted on the smart contract functions to verify their business logic, such as minting or token transfer functions. This testing is typically performed using the React framework for frontend and Hardhat for smart contracts.
- Integration Testing Following unit testing, the next step is integration testing, which ensures that all components in the application work well when combined. In the context of smart contract integration, this involves testing the interaction between the React.js frontend and the smart contract through ethers.js or web3.js. Integration testing helps detect issues in the data flow between the frontend and blockchain, including transaction validation and correct state updates.

IV. RESULTS AND DISCUSSION

A. Results

The overall result of the developed system is a web interface, Non-Fungible Token (NFT), and interoperable smart contract. The following sections detail each result of the developed system.

1) Web Interface: The initial web interface displays a list of NFT collections available for minting. The NFT can initially be owned by the address that mints it first, and ownership can be transferred to another address. At the initial dashboard, users must press the *Connect Wallet* button, which will then connect to the Metamask Wallet account. From this connection, users obtain an address that can be used to mint NFT tokens.

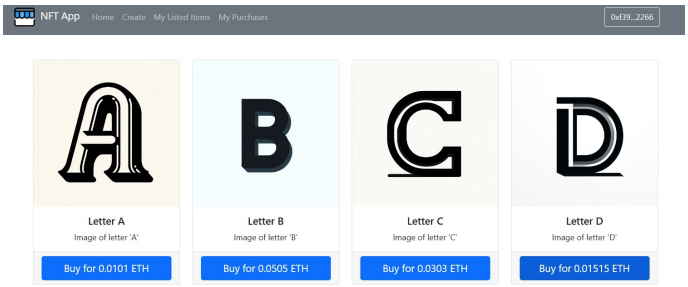

Figure 4. View of the web interface

2) Smart Contract: In the development of blockchain-based applications, integrating the smart contract with user interfaces such as React.js becomes crucial. The smart contract built using Solidity can be deployed on Ethereum testnets, such as the Sepolia Testnet, which provides an environment similar to the Ethereum mainnet but without requiring high transaction fees.

ABI, or Application Binary Interface, is a means that allows functions within an Ethereum smart contract to communicate with external applications, including user interfaces built with frameworks like React.js. ABI acts as a translation layer that outlines how to call functions in the smart contract, the types of parameters it accepts, the expected output types, and the state nature of those functions. ABI is typically generated automatically by the Solidity compiler as part of the smart contract compilation process and is stored in JSON format. Whenever the frontend application sends a transaction or query to the blockchain, it uses the ABI to encode the call data into a format that the Ethereum Virtual Machine (EVM) can understand. Then, when data is returned from the smart contract, ABI is used to decode the response so the React.js application can understand and process it.

The deployment process is carried out where the smart contract will be deployed to several blockchain networks, but for this testing, only the Sepolia testnet is used. During the deployment process, a gas or fee will be charged, which can be paid using Ethereum in the Metamask wallet. Details of this transaction can be seen on Etherscan.

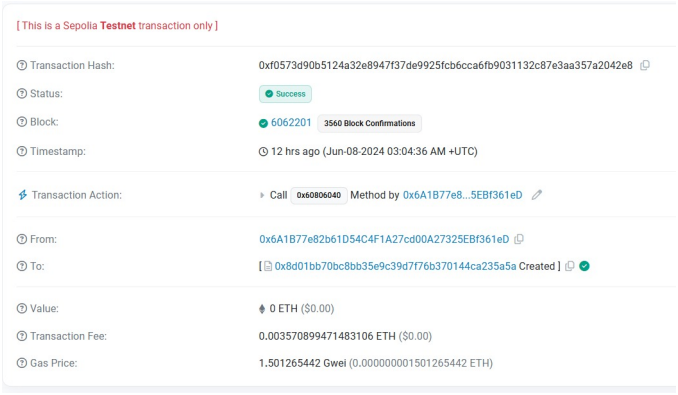

Figure 5. Transaction details on Etherscan

3) Non-Fungible Token (NFT): Non-Fungible Tokens (NFT) also undergo several stages before they can be stored in The Interplanetary File System (IPFS) and then be published on OpenSea test net. In this thesis work, Pinata is used as a platform to upload NFT photos and JavaScript Object Notation (JSON) data so that it can be minted as a Uniform Resource Identifier (URI) on the smart contract. This is necessary so that the minted NFT has an image that can be viewed on OpenSea.

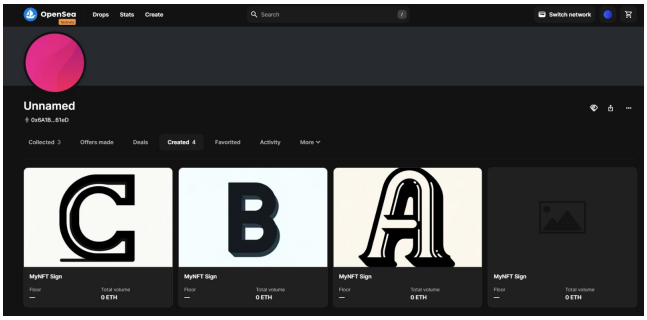

Figure 6. NFT uploaded on Opensea

B. Testing Features of Smart Contract Interoperability

The testing expectations from the smart contract system include the following:

- Users can mint NFT tokens, after which the ownership of these NFTs can be viewed on the test net platform OpenSea.
- Users from different Networks can communicate with each other, and the ownership of the NFT tokens can move from user A on network A' to user B on network B'.

Here is the data regarding the wallet used in testing:

The following are the steps of testing the smart contract system that will be conducted:

- User A will mint an NFT first

Table I
VOLUME EDV DAN ESV ANTARA GROUND TRUTH DAN PREDIKSI

Account	Address	Network
A	0x6A1B77e 82b61D54C4 F1A27cd00A27 325EBf361eD	Sepolia Ethereum Testnet
B	0xD066d6576 D9485Eb2 c2a41BB8B52EcE 17a0557d6	BNB Chain Testnet

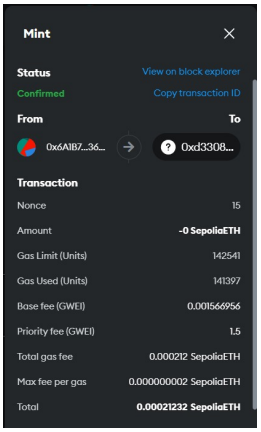

Figure 7. Minting transaction history on Metamask Wallet

- The details on Etherscan can also be seen in the image below, showing details like the transaction hash, block, and the type of token, which is ERC-721.

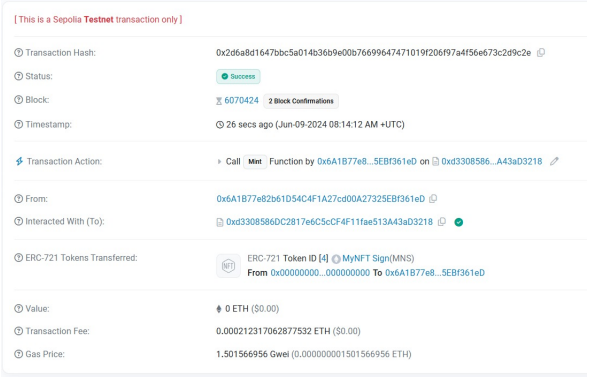

Figure 8. Transaction details on Etherscan

- Then, on Etherscan, the details of the NFT that has been minted can be seen as in the following image

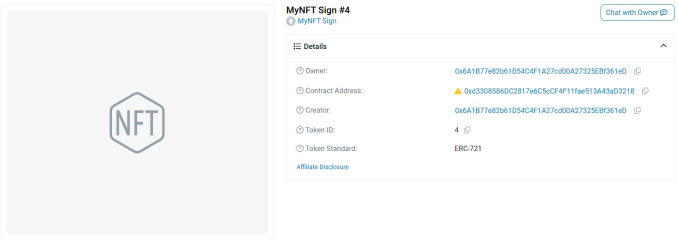

Figure 9. NFT Details on Etherscan

- After the process is successful, the NFT that has been minted can be seen on the platform OpenSea. In this test case, the minted NFT is "Letter D".

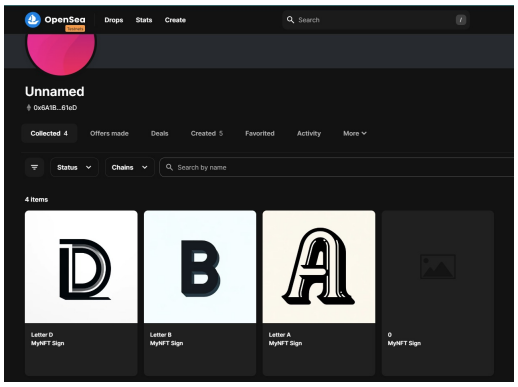

Figure 10. NFT on OpenSea

- After the NFT is created, the next step is to lock the NFT to be sent to user B on network B'. The function of locking is to ensure that the token data is not changed during the transfer process, maintaining the trust and authenticity of the NFT data. It is also useful to signal to all related parties (users, smart contracts on other networks) that the token is undergoing a transfer process, and operations on the token should be suspended until the process is completed.

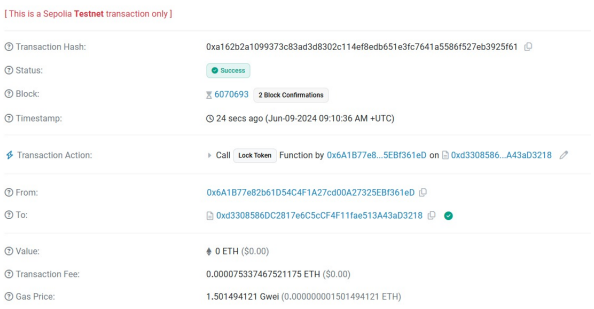

Figure 11. Lock Token function details on Etherscan

- After completing the lock token process, then the bridge transfer function can be executed. This function facilitates the secure transfer of NFTs between blockchains by ensuring that the NFT is locked during the transfer process and providing transparent visibility of the transfer event through recorded events. This is a key component in building interoperable applications that allow digital assets to move across blockchain ecosystems.

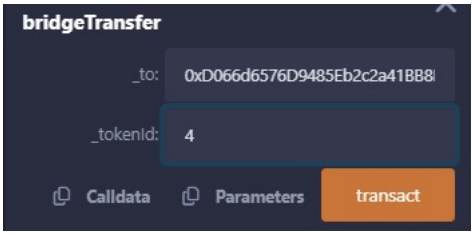

Figure 12. Bridge transfer parameters

- In the image 12, there are two parameters: "`_to`", used to receive the parameter address of user B, and "`_tokenId`" to receive arguments from the NFT token. After the transaction is executed, it will be followed by payment on the Metamask Wallet.

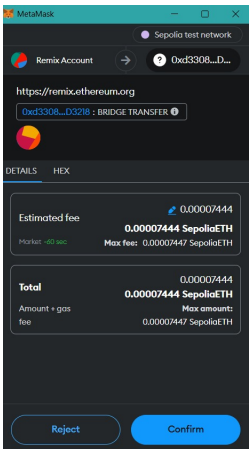

Figure 13. Payment on Metamask Wallet

- After the transaction is successful, the record of the transaction is stored on the blockchain and can be viewed on Etherscan as shown in the image below.

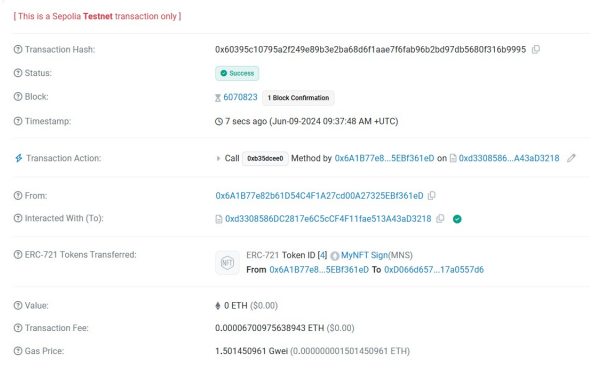

Figure 14. Details on Etherscan

- On Etherscan, details of the NFT can also be checked for ownership. As seen from the image 9, the owner of NFT #4 is the address of user A on the Sepolia Ethereum Testnet, then in the image below, after successfully performing the bridge transfer, the ownership changes to user B on the BNB Chain Testnet.

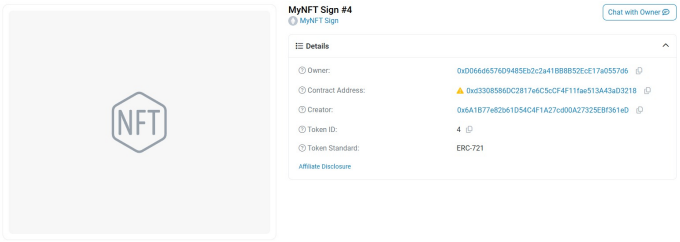

Figure 15. NFT details on Etherscan after bridge transfer

- On the OpenSea platform, the NFT sent to the address of user B can also be seen.

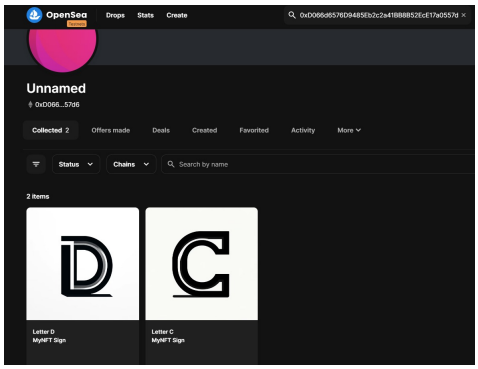

Figure 16. NFT on OpenSea at the address of B

V. Conclusion

In this research, we have explored and implemented the concept of blockchain-based NFT interoperability using smart contracts within the context of the Metaverse and Web3.0. This study has successfully demonstrated the potential and effectiveness of blockchain technology, particularly Ethereum, in supporting the creation and management of NFTs that can operate across various tokens.

1) Conclusions from this research are:

- Smart Contract Implementation: The designed smart contracts have successfully facilitated the creation, transactions, and verification of NFTs securely and efficiently. By using the ERC-721 standard, we have implemented unique NFTs that can be traced to their origin, which is essential in the Metaverse and Web3.0 ecosystems.
- The developed NFTs show high interoperability across various Metaverse platforms. This was achieved through the use of consistent protocols and APIs, allowing the NFTs to be used in various applications and games within the Web3.0 ecosystem without substantial modifications.

2) : Recommendations for further development To enhance the interoperability and efficiency of NFT usage in the Metaverse, it is recommended that future research focus on developing more universal protocols for cross-chain integration. Additionally, further research is needed to refine security mechanisms that can protect user privacy while maintaining the transparency and reliability of the blockchain system.

References

[1] Q. Wang, R. Li, Q. Wang, and S. Chen, "Non-fungible token (nft): Overview, evaluation, opportunities and challenges." *NFT's*, 2021.

[2] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. hen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms. future generation computer systems," *An overview on smart contracts: Challenges, advances and platforms. Future Generation Computer Systems*, 2020.

[3] N. Malik, Y. M. Wei, G. Appel, and L. Luo, "Blockchain technology for creative industries: Current state and research opportunities," *International Journal of Research in Marketing*, 2023.