

디지털 자산을 위한 사용자 친화적인 NFT 플랫폼

भागत नितिन^o, 배종욱^{*}, 이수현^{*}

^o창원대학교 컴퓨터공학과,

^{*}창원대학교 컴퓨터공학과

e-mail: bhagatnitin312@gmail.com^o, bae@cwnu.ac.kr^{*}, sleep1@cwnu.ac.kr^{*}

A user friendly NFT platform for Digital Assets

Nitin Bhagat^o, Jong-Wook Bae^{*}, Su-Hyun Lee^{*}

^oDept. of Computer Engineering, Changwon National University,

^{*}Dept. of Computer Engineering, Changwon National University

● 요약 ●

Blockchain technology has paved the way for the rise of non-fungible tokens(NFTs) in recent years. NFTs enable the unique ownership of digital assets and harness the power of blockchain's transparency and decentralization. However, existing NFT platforms often pose barriers to entry for the public due to their technical complexity and high issuance costs. To address the problem, this paper proposes a user-friendly NFT platform with a simplified issuance process. A home page is created to allow easy NFT issuance for anyone. These advancements are expected to foster the growth of the NFT market and facilitate the development of new business models.

키워드: Non-Fungible Tokens, Klaytn blockchain, smart contract, metadata, IPFS, KIP-17

I. Introduction

The rise of Non-Fungible Tokens has revolutionized the digital asset landscape, providing a new paradigm for owning and trading unique digital assets[1]. However, the current NFT ecosystem often lacks adequate platforms and tools for creators and collectors to manage and track their NFT holdings.

That's where our platform comes in. In this research, we additionally discover that the Klaytn blockchain offers a infrastructure for NFT creation, ensuring the authenticity, uniqueness, and immutability of digital assets.

Therefore, this research is focused on an integrated platform where anyone can easily mint and use NFTs. in order to implement, we create a smart contract that can mint based on KIP-17. We also implement a user friendly interface and server-side processing on our server. These advancements are expected to foster the growth of the NFT market and facilitate the development of new business models.

II. Theoretical Background

Blockchain is a distributed ledger technology that stores and manages data across multiple servers (nodes) without a central server. Each node stores the same information, and when a new transaction (or data) occurs, the data is verified, and all nodes are updated equally[2].

Smart contracts are automated agreements that operate on the blockchain and execute transactions according to predefined promises when certain conditions are met. This allows them to codify the intentions of the contracting parties and expedite transactions according to predefined rules[3].

Non-fungible tokens represent a typical blockchain-based digital asset. NFTs have a unique value for each token and are non-exchangeable. This NFT structure plays an important role in clarifying the ownership of digital assets.

KIP-17 is a token standard on the Klaytn blockchain that defines the implementation of NFTs[4]. It is a protocol for the creation, management, and transfer of unique digital assets, ensuring interoperability and compatibility within the Klaytn ecosystem.

III. The proposed NFT platform

The overall block diagram of our system is shown in Figure 1. The user can upload their digital content on our platform's upload page by filling in the relevant details. Our server then internally uploads the content to the IPFS site, and the IPFS server generates a unique CID based on a hash value for the content. This CID becomes the address of the uploaded content. A metadata file is created by our server, based on the content's CID and the details provided by the user. This metadata file is also uploaded to IPFS, which generates a separate CID for the metadata.

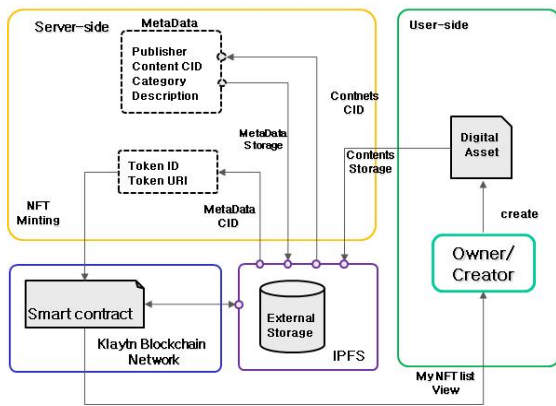


Fig. 1. Block diagram of NFT

After uploading their digital content, the user can log in to their Kaikas wallet and click on the Mint button. This action triggers the smart contract incorporated in our system, and it mints an NFT using the issued CID as the token URI. Once the minting process is complete, the user can view and manage all the NFTs they have minted on the "MyNFT" page. This feature allows users to track and manage all their NFTs within our system.

IV. Implementation of NFT platform

The following source code is part of the Solidity source for our smart contract. To satisfy the Klaytn-based KIP-17 protocol, we inherit KIP17Enumerable to create a smart contract, and create a Counters type to increment the token ID by one for each mint[5]. We also store the CID of the metadata in _tokenURIs. The tokenURI() function reads the stored CIDs, and the mintNFT() function performs the minting. The getNftTokens() function retrieves the minted data per account.

```
contract QKNFT is KIP17Enumerable {
    Counters.Counter private _tokenIds;
    mapping(uint =>string) public tokenURIs ;

    constructor() KIP17("CWNU_NFT", "CWNU"){ }
    function tokenURI(uint tokenId) { }
    struct NFTs { uint256 id; string uri; }
    function mintNFT(string memory _tokenURI) { }
    function getNfts(address _owner) { }
}
```

Next, generates the details of the content as metadata, and uploads it to the IPFS (INFURA) site through the following source code. Here, we can use it only when we register the project ID (pId) and secret key (pSecret) for IPFS in the user account of INFURA[6].

```
function upload(buf, tag){
    const auth = "Basic " + btoa(pId + ":" + pSecret);
    const ipfs = window.IpfsApi({host:'ipfs.infura.io',port:
    5001, protocol:'https', headers:{authorization: auth},});
    hash = "";
    ipfs.add(buf, (err, result) => {
        if (err) { console.error(err); return; }
        hash = result[0].hash;
    });
    return hash;
}
```

When the upload is complete, the CID of the uploaded data is displayed at the tag location, as shown in figure 2, and this becomes the tokenURI when minting.

Issuing NFTs

Content details

Publisher	Changwon
Category	<input checked="" type="radio"/> basic <input type="radio"/> important <input type="radio"/> etc
Description	Cat NFT
Choose File	cat.jpg <input type="button" value="upload"/>
token URI	QmNpR3q4eF1ybYxe4kk39BvwwM3pzbvFPzPeCY1SwwZns29
Metadata content:	<pre>{ "name": "Changwon", "image": "https://ipfs.io/ipfs/QmNpR3q4eF1ybYxe4kk39BvwwM3pzbvFPzPeCY1SwwZns29", "description": "Cat NFT", "attributes": [{"trait_type": "category", "value": "default"}]}</pre>

Fig. 2. Content and metadata upload result

Next, 'mintNFT' function in the smart contract mints the NFT with a unique token ID and associates it with the provided

tokenURI. It increments the token ID counter, assigns the tokenURI to the corresponding token ID, mints the NFT for the caller, and returns the token ID.

```
function mintNFT(string memory _tokenURI
                payable public returns(uint256) {
    _tokenIds.increment();
    uint256 tokenId = _tokenIds.current();
    tokenURIs[tokenId] = _tokenURI;
    _mint(msg.sender, tokenId);
    return tokenId;
}
```

Next, the 'getNFTs' function retrieves a list of NFTs owned by owner. It iterates through the tokens owned by the owner address, fetching their ID and URI. This information is stored in an array and returned, allowing read-only access to the NFT token details. so gas free.

```
function getNFTs(address _owner) view virtual
    public returns (NFTs[] memory) {
    uint256 length = balanceOf(_owner);
    NFTs[] memory nfts = new NFTs[](length);

    for(uint256 i = 0; i < length; i++) {
        uint256 id = tokenOfOwnerByIndex(_owner, i);
        string memory uri = tokenURI(id);
        nfts[i] = NFTs(id, uri);
    }
    return nfts;
}
```

Upon clicking the "My NFT" menu on the our platform, a comprehensive list of non-fungible tokens (NFTs) that have been minted by the corresponding user is readily presented, as illustrated in Figure 3.

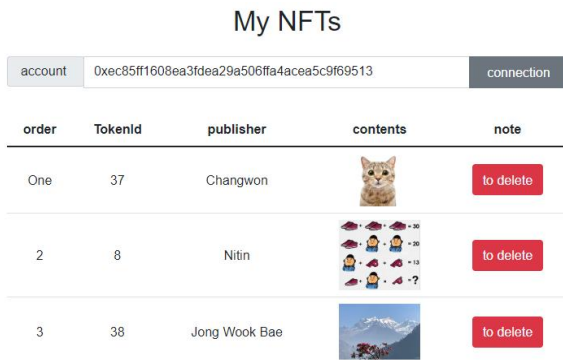


Fig. 3. Listed of created NFTs

the NFTs created on our platform are listed in Opensea testnet website by connecting through Kaikas wallet[7]. The user can view all the NFTs created by each individuals on our platform through OpenSea. Users can also set price, for their individual NFTs they have created in our platform, in Opensea website for sale as shown in figure 4.

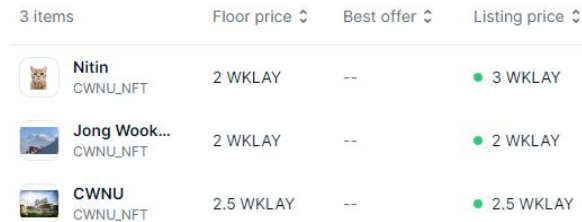


Fig. 4. List of NFTs for sale

V. Conclusions

In conclusion, this research has provided a detailed explanation of the process of creating an NFT on the Klaytn blockchain. Once the user is connected to our platform through Kaikas wallet, our system provides a user-friendly platform for uploading and minting NFTs. By utilizing the IPFS for secure storage of digital content and metadata, our platform ensure the uniqueness and accessibility of NFTs. Based on KIP-17, the smart contract we created utilizes the generated CID as a token URI to enable the issuance process. Moreover, our platform simplifies user experience by offering a convenient way to track and manage all minted NFTs in one location. Our platform enhances user convenience by allowing them to track and manage their minted NFTs at one stop. Additionally, our platform enables greater trading opportunities since the minted NFTs are compatible with other marketplaces such as OpenSea, allowing users to trade their NFTs easily in a more extensive market.

In the future, we plan to complete the NFT platform by creating a form to register to sell NFTs on our platform and a form to view all NFTs for sell.

REFERENCES

- [1] Kugler, L. (2021). Non-fungible tokens and the future of art. Communications of the ACM, 64(9), 19-20.
- [2] Ethereum Whitepaper, <https://ethereum.org/en/whitepaper>, accessed 2023.06.
- [3] Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., & Bani-Hani, A. (2021). Blockchain smart contracts:

Applications, challenges, and future trends. Peer-to-peer
Networking and Applications, 14, 2901-2925.

- [4] Klaytn API service, <https://www.klaytnapi.com/en/resource/openapi/kip17/reference/overview/> accessed 2023.06.
- [5] Kaikas Docs, <https://docs.kaikas.io/>, accessed 2023.6.
- [6] INFURA, <https://www.infura.io/product/ipfs/>, accessed on 2023.06.
- [7] OpenSea, <https://testnets.opensea.io/>, accessed on 2023.06.