applied sciences

Article

Digital Content Management Using Non-Fungible Tokens and the Interplanetary File System

Hoon Ko [1], **Juhee Oh** [2] **and Sung Uk Kim** [2,*]

[1] Security R&D Center, CMT Info. & Comm. Co. Ltd., #803, 37 Seongsu-ro 22-gil, Seongdong-gu, Seoul 04798, Republic of Korea; skoh21@cmtinfo.co.kr

[2] AINTCHAIN SOFT Co., Ltd., 131, Mihang-ro, Mokpo-si 58750, Jeollanam-do, Republic of Korea; itsmarvin0208@aintchain.co.kr

* Correspondence: ksubest@aintchain.co.kr

Abstract: Non-fungible tokens (NFTs) are digital identifiers containing metadata, such as token number, title, content, and image URL, and are linked to digital assets, which are characterized by the fact that, unlike conventional virtual assets, they have their own unique value and cannot be replaced. NFTs cannot be deleted or forged; therefore, they can be used to authenticate the ownership of digital assets. The metadata of the NFTs are uploaded to the interplanetary file system (IPFS), which is a distributed file system, and converted into unique content identifiers (CIDs) that are stored on the blockchain. Digital content (DC) is divided into multiple pieces; it also has its own unique value and is distributed and stored using the IPFS. This study built an NFT-based IPFS testbed and experimented with the process of generating unique values for DC divided into three groups and sharing them. The results confirmed that each DC had a unique hash value and no duplicates existed.

Keywords: non-fungible token; blockchain; digital contents; interplanetary file system; content protection

check for updates

Citation: Ko, H.; Oh, J.; Kim, S.U. Digital Content Management Using Non-Fungible Tokens and the Interplanetary File System. *Appl. Sci.* **2024**, *14*, 315. https://doi.org/10.3390/app14010315

Academic Editor: Antonio Fernández-Caballero

Received: 16 October 2023
Revised: 19 December 2023
Accepted: 26 December 2023
Published: 29 December 2023

1. Introduction

Industry 4.0 has resulted in revolutionary changes in automation, monitoring, and supply chain analysis through the implementation of smart technologies such as big data, artificial intelligence (AI), cloud computing, metaverse-based augmented reality (AR), industrial Internet of Things (IIoT), robotics, digital twins, and cybersecurity. Among them, metaverse-based digital content (DC) is used in various smart industry applications [1]. In particular, DC generated in smart industries, i.e., DC linked to the metaverse, such as digital training programs and digital biological cultivation training programs, are drawing considerable attention [2]. However, despite the creation of various types of industrial DC based on the metaverse, safeguarding against the criminal act of forging is challenging [3,4]. Conventional DC employs digital rights management (DRM) to address this issue; however, the significance of conventional digital certificates is decreasing with the rise of AI-based cybercrime [5]. DRM technology has been conventionally used to protect simple digital documents and videos, and it is not easy to apply it to AR/VR-based digital documents stored in the metaverse [6]. Additionally, cybercriminals employ AI to forge digital certificates quickly with such sophistication that distinguishing them from the real ones is difficult. Therefore, not only is it difficult to verify the authenticity of digital certificates but their verification process is also complicated, rendering the entire process complex.

This study proposes a method that employs blockchain-based non-fungible tokens (NFTs) and the interplanetary file system (IPFS) to address these challenges and enable a large amount of DC to be shared [7,8]. This paper is organized as follows: Section 2 summarizes the current security issues of DC and related works on addressing them. Section 3 describes the proposed digital content platform. Section 4 outlines the processes of the digital content management system. Section 5 describes the experiment and presents the results, and finally, Section 6 outlines the conclusions and directions for future research.

2. Related Work

2.1. Limitations of DC

NoonooTV is a piracy site that has claimed to have servers in foreign countries, including the Dominican Republic, and has allowed users to watch high-quality videos without age verification or a sign-up process. It was illegally streaming videos without the authorization of copyright holders, such as broadcasters and over-the-top (OTT) service providers. The videos received approximately 1.538 billion views in total, and the site had approximately 10 million active monthly users. Broadcasters such as MBC, KBS, JTBC, TVing, and Wave and the production studio SLL took legal action against noonooTV in early March 2023, after which the Korean National Policy Agency announced that it was launching an investigation into the website. Currently, MBC, KBS, JTBC, TVing TV, Wave, and SLL are creating the Video Copyright Protection Council to jointly file a complaint against noonooTV for copyright infringement [9]. Netflix is also filing a complaint against noonooTV through ACE, the world's largest organization fighting against illegal reproductions.

2.2. NFTs

Unlike conventional virtual assets authenticated with digital certificates, NFTs assign unique values to digital assets, making them irreplaceable and unforgeable. Furthermore, as they are stored on the blockchain, deleting or counterfeiting them is impossible [10,11]. Consequently, they can be used to authenticate the ownership of digital assets [12]. Recently, South Korean universities have awarded diplomas and prizes using NFTs. Additionally, they are being used for various purposes on campuses, and their uses are continuously expanding. NFTs offer the advantage of authenticating qualifications through their proof function, thereby enhancing transparency. Moreover, they can save time and costs because they can be easily authenticated using an electronic wallet. Furthermore, certificates can be easily issued on the Internet, which increases administrative accessibility and convenience [13].

If we define the existing problems with NFTs, we can highlight issues related to copyright and ownership, security and privacy, and technical complexity. Here are the explanations for each:

- ○ Issues with copyright and ownership: NFTs are used to prove ownership of digital content but do not fundamentally alter the ownership of the actual content [9]. For instance, purchasing an NFT of digital art does not transfer the copyright or intellectual property rights of the art itself. Additionally, the link between the original artwork and its associated NFT can be unclear, leading to potential ownership disputes. Therefore, the new ownership model brought by NFTs raises debates regarding how it aligns with existing copyright laws [14].
- ○ Security and privacy: Certain NFT platforms may possess security vulnerabilities, potentially allowing hackers to expose user information or attempt fraudulent activities, like forging transactions [15]. Moreover, due to the immutable nature of blockchain, there are concerns about user data privacy since transaction records are publicly available, risking the exposure of personal information on the blockchain [16].
- ○ Technical complexity: For users unfamiliar with blockchain and cryptocurrency technology, purchasing and trading NFTs can be complex and challenging. Technical aspects such as wallet creation, transaction fees, gas costs, etc., can pose difficulties, making entry into the NFT market challenging for the general public [17].

2.3. Use Cases

2.3.1. Case 1

On 11 March 2021, Christie's auction house in the United States sold the NFT associated with "Everydays: The first 5000 days", a digital artwork created by the digital artist Beeple, for USD 69.8 million (Figure 1a). This work was published as an NFT in the form of a JPG file created using photos that the artist had been posting online since 2007. The successful auction of this NFT significantly increased the popularity of NFT-based digital art [18].

(a) Everyday: the first 5000 days

(b) Lee Sedol vs. AlphaGo, Round 4

(c) Lebron James, NBA

Figure 1. Use cases of NFTs.

2.3.2. Case 2

In 2016, a video of a historic Go game wherein Lee Sedol defeated Google's AlphaGo computer program, developed by DeepMind Technologies, was published as an NFT on the Ethereum blockchain and auctioned (Figure 1b). It was ultimately sold for ETH 60 (approximately USD 210,000 at the time) on OPENSEA, the world's largest NFT marketplace [18].

2.3.3. Case 3

The NBA Top Shot marketplace allows fans to directly own a piece of NBA history, such as Kevin Durant's three-point shot, in digital form. It made record NFT transactions of USD 600 million (approximately KRW 690 billion) in the first half of 2021. Furthermore, Daffer Labs, the developer of NBA TOPSHOT, successfully launched the cat breeding game "CryptoKit" in 2017. The keyword that runs through the utilization of NFTs is "assetization of DC" (Figure 1c) [18].

3. Digital Content Platform

3.1. System Structure and Procedure

Figure 2 shows a block diagram of an NFT-based digital certificate called BApp. The required components comprise the file system, front end, backend, blockchain network, smart contract, ERC-721 standard, IPFS, and Ethereum virtual machine (EVM) [19,20]. The front end and blockchain network components are responsible for the user interface and communicate with the blockchain through the front end's caver.js library, issue certificates, and execute inquiries. As the DC must be stored and processed for proof after issuance, it is stored on a blockchain network to ensure its integrity and processed using the IPFS and EVM [21,22]. EVM is a virtual machine that provides a secure environment for executing smart contracts on the Ethereum network, whereas IPFS is used to store DCs individually.

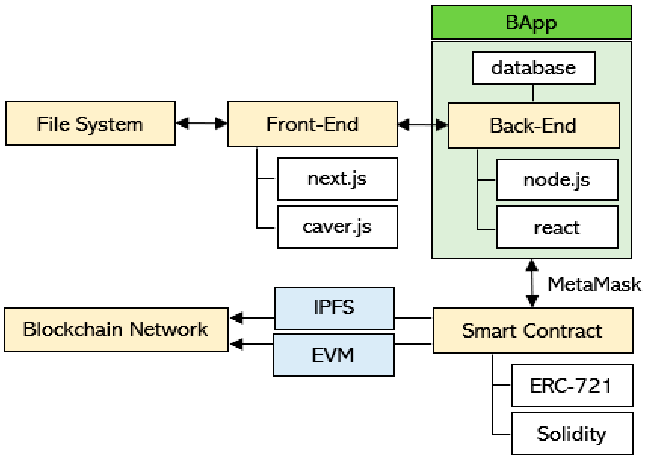


Figure 2. System structure of the DC platform.

3.2. ERC-721

NFTs that are used to authorize digital certificates are issued using the Ethereum standard ERC-721, which is represented by CryptoKit and allows the assignment of a unique identification value to each token. As only one issued token exists on the Ethereum network, the owner can be easily identified [23]. Figure 3 shows the flow of CryptoKit, which comprises a hash function, symmetric key cryptography, public key cryptography, and an insecure module (hash functions) [24].

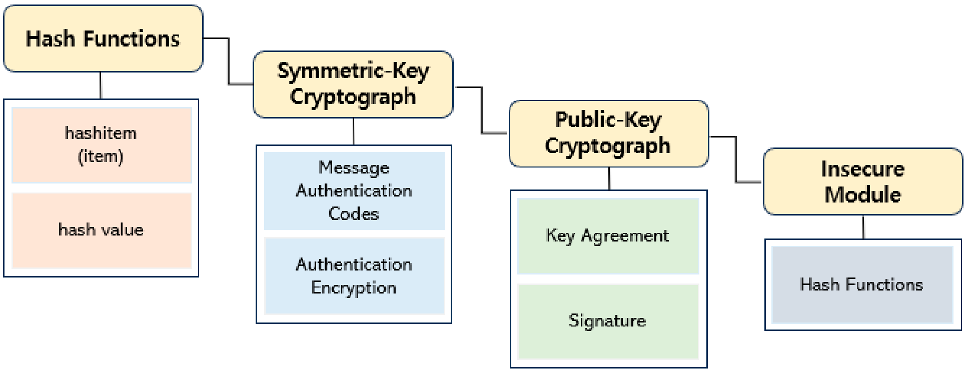


Figure 3. CryptoKit flow.

The processing sequence of CryptoKit involves opening the crytpKitTutorial.playground to check the hashItem(item:) function of the Hashable protocol and writing "the quick brown fox" [24]. The hashing algorithm is one-way with non-linear operations that generate a unique hash value, as shown in Algorithm 1. When explaining the algorithm, first, the hashItem function takes a string named item as a parameter. Then, it creates a hash function object called hasher using Hasher(). Subsequently, it hashes the item into the hasher using the hash(into:) method, which generates a hash value based on the content of the string. Next, by using hasher.finalize(), it obtains the final hash value of the hash function. Finally, this function returns the ultimate integer hash value obtained by hashing the string item.

Algorithm 1. ***//Hash Function***

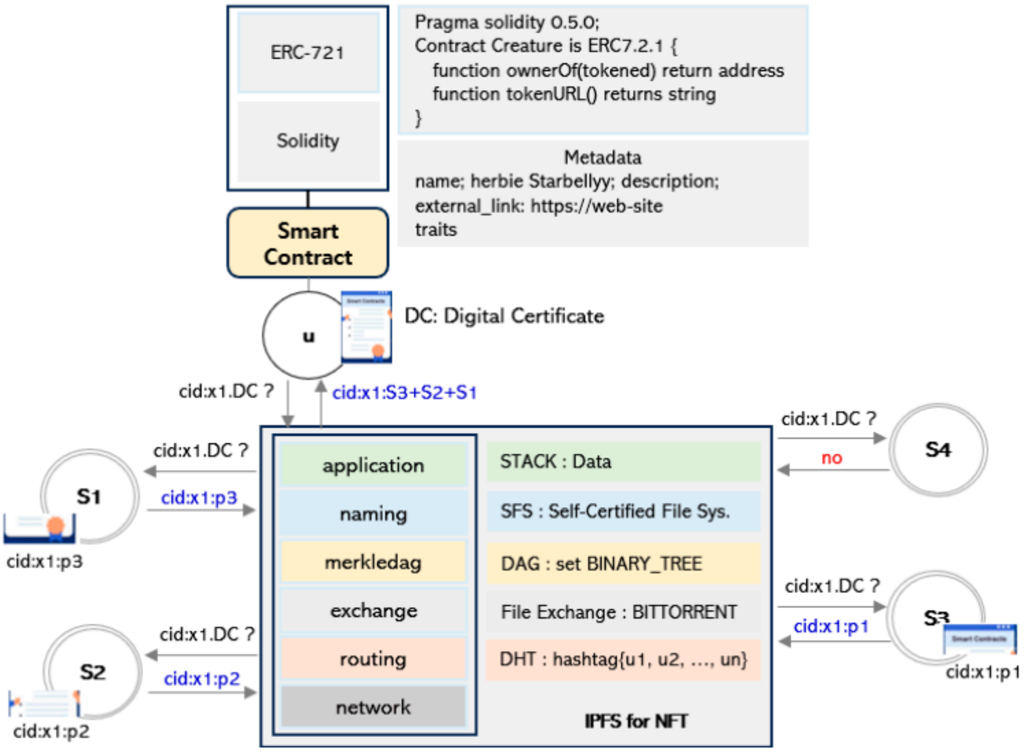
```
func hashItem(item: String) -> Int {
    var hasher = Hasher()
    item.hash(into: &hasher)
    return hasher.finalize()
    }
Let hashValue = hashItem(item: “the quick brown fox”)
```

Symmetric key cryptography comprises message authentication codes and authentication encryption, which generate the hash-based message authentication code (HMAC); this code safeguards against malicious changes by signing the message digest using a symmetric key. The HMAC uses a secret key for internal/external processing, generates an internal hash from the data (image), and obtains an inner key [25,26].

3.3. IPFS

IPFS is a shared distributed file system that enables sharing files across multiple computers (Figure 4). NFTs are issued using smart contracts based on the ERC-721 standard and the Solidity programming language. The token metadata are uploaded to the IPFS distributed file system and converted into content identifier (CID) values [27,28]. Data management can be made more efficient by storing the CID values on the blockchain, which also enables the decentralization of web services as data are not stored on a central server. NFTs comprise metadata such as the token number, title, content, and image URL, which are linked to the NFT on the blockchain. The higher the amount of information stored in the token, the higher the amount of information stored on the blockchain, which can render the operation inefficient. Thus, to address this limitation, the NFT metadata are efficiently stored on the blockchain using the IPFS [29].

content address -> /ipfs/QmahZg6Ju3iM5nU3Ry2X2Kx4rFdxUHa3DP9biDP29yFGYT



```
Pragma solidity 0.5.0;
Contract Creature is ERC7.2.1 {
   function ownerOf(tokened) return address
   function tokenURL() returns string
}
```

Metadata
name; herbie Starbellyy; description;
external_link: https://web-site
traits

Figure 4. Processing sequence of an IPFS-based DC call.

4. Digital Content Management (DCM) System

IPFS is a new protocol that addresses the limitations of the existing hypertext transfer protocol (HTTP). Figure 4 shows the processing of an IPFS-based DC call. The processing sequence after the content address is addressed as follows: DHT/routing->BITTORRENT/File exchange->DAG/merkledag->SFS/naming->application. Herein, IPFS is invoked through the content hash instead of the traditional method of addressing the domain name.

NFT-based DCMs employ the tamper-proof characteristic of IPFS files to the DC procedure on the blockchain and check for changes in the DC metadata, as shown in Algorithm 2. If we explain the procedure in Algorithm 2, the digital content (DC) is comprised of blocks (b), each having three components, such as b1, b2, b3. Utilizing these three elements (b1, b2, b3), a hash function (H) is employed to generate a single value, represented as p{p1, p2, p3}. The 'generate' function maps the digital contents to each block, where, for instance, p1 is mapped to b1 and p2 is mapped to b2. The interplanetary file system (IPFS) is responsible for storing digital content and includes users (u) and servers, such as S1, S2, S3, Sn. Each server (S1, S2, S3, Sn) currently holds subsets of digital content, and lastly, S4 does not hold any content.

Algorithm 2. ***//Procedure of DCM***

```
DC: Digital Contents
b: block = {$b_1$, $b_2$, $b_3$}
p{$p_1$, $p_2$, $p_3$} = H($b_1$, $b_2$, $b_3$)
generate = ($p_1$,$b_1$)($p_2$,$b_2$)($p_3$,$b_2$)
IPFS = {u, $S_1$, $S_2$, $S_3$, $S_n$}$_{DC}$
$S_1$ = {$p_1$, $p_2$, $p_3$}
$S_2$ = {$p_1$, $p_2$, $p_3$}
$S_3$ = {$p_1$, $p_2$, $p_3$}
$S_4$ = {}
```

First, IPFS allows node u to request a DC sent via a smart contract using the systems connected to the IPFS. If the system sends a DC that is not stored at u, the node constructs the complete DC by assembling the partially received pieces of information. The DC, which comprises multiple blocks, is processed block-by-block using a hash and then assigned a unique name. After storing the names of all files in a database, duplicate files are eliminated, and the version information of each file is tracked. Subsequently, each node (u, S1, S2, S3, S4) stores only the required files, and the indexing information is used to determine the file stored at each node. To find a file on the network, the file name is first searched; thereafter, the node that has the file stored is contacted.

4.1. Distributed Hash Table (DHT)

DHTs comprise six APIs, namely findPeer, findProvs, get, provide, put, and query, that manage the routing process and DHTs, as shown in Table 1. A P2P network can be realized without a centralized server by allowing nodes participating in the network to manage their own hash tables.

Table 1. APIs of DHTs.

API Name	Function
dht.findPeer	Find multi-addresses using the PeerID
dht.findProvs	Find peers to provide a specific value (CID)
dht.get	Query the routing system for a given key
dht.provide	Alarm the network with the given values
dht.put	Write a key or value pair to the routing system
dht.query	Find the closest PeerID to the given PeerID or CID

Rather than using a centralized system, DHTs use hash tables to locate files by mapping the name of each node to a value (Figure 5). The DHT operation determines network efficiency and how nodes enter or leave the network and register new content. Therefore, it reduces the network load and allows searching the network content quickly and accurately, depending on the configuration.

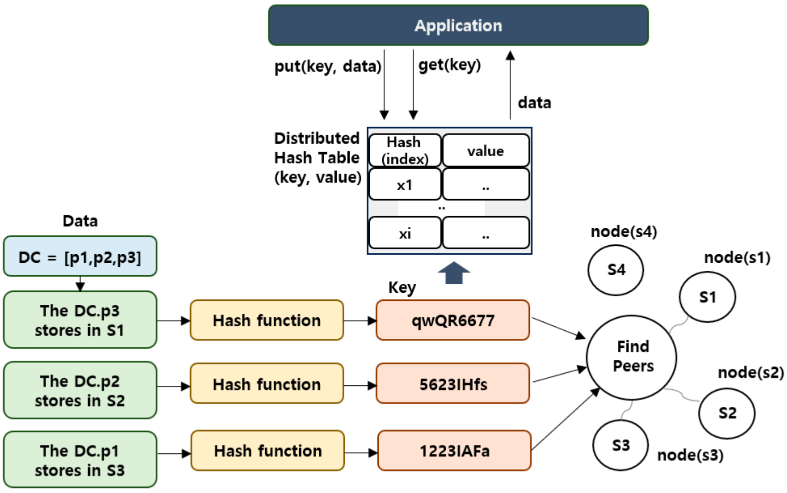


Figure 5. Hash function in a DHT.

Explaining the DHT algorithm defined in Algorithm 3, initially, 'contents1 (DC.p1): S1' denotes a portion of digital contents allocated to server S1, while 'contents2 (DC.p2): S2' and 'contents3 (DC.p3): S3' represent different segments of digital contents assigned to S2 and S3, respectively. 'n' represents the total number of servers, and the distributed hash table (DHT) contains pairs of keys and values. Here, 'key' signifies the content name, and 'value' represents the IP address. 'peers(DATABASE)' refers to querying peers regarding the database. 'query' performs a query on the database, and if the database matches a specific key, it returns that database. In cases of inserting values into the database, 'insert(key, value) by peers into database' indicates peers inserting key–value pairs into the database. Within these operations, the message count is O(log(N)), and state changes are O(log(N)). Here, 'O(log(N))' represents the algorithm's time complexity, indicating that the execution time of operations is logarithmically proportional to the database size (N).

However, it is important to define an optimal nearness metric, maintain a minimum hop count, and define a minimum hash table for optimal DHT routing and fast processing. Hence, efficient generates O(log(N) messages for each lookup, and scalable generates O(log(N) states.

Algorithm 3. *//DHT API Algorithm*

```
define contents1 (DC.p1): S1
define contents2 (DC.p2): S2
define contents3 (DC.p3): S3
$n$ = the total number of servers

DHT <- pair(key, value);
      key <- content name
      value <- IP address
query <-peers(DATABASE)
if database == key
      returns (database)
insert (key, value) by peers into database
messages == O(log(N))
state == O(log(N))
```

4.2. Bit Torrent (File Exchange)

The IPFS is a peer-to-peer file exchange protocol implemented on a distributed network as it shows in Algorithm 4. In BitTorrent, a single file is divided into multiple pieces, and each node informs the other nodes about the pieces it has and requests from them the ones it requires. Therefore, numerous sessions are generated to exchange information between nodes, and the download speed increases as the number of sessions increases (Figure 6).

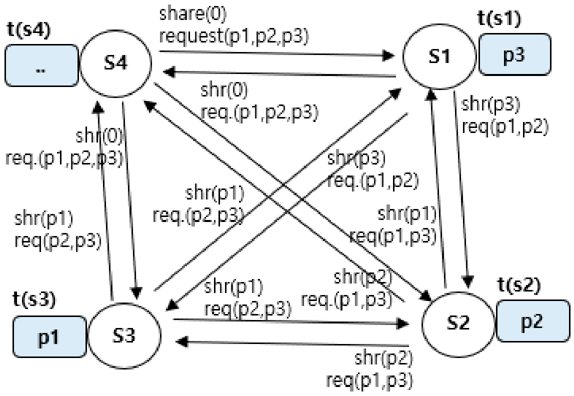


Figure 6. File sharing among nodes in BitTorrent.

Algorithm 4. *//File Share & Exchange*

```
p = {p1, p2, p3}
S1 = {p3}
S2 = {p2}
S3 = {p1}
S4 = {}

(S2, S3, S4)_share(p3) <- S1
(S2, S3, S4).request(p1, p2) <- S1
session_establishment (S1-S2, S1-S3, S1-S4)

(S1, S3, S4)_share(p2) <- S2
(S1, S3, S4).request(p1, p3) <- S2
session_establishment (S2-S1, S2-S3, S2-S4)

(S1, S2, S4)_share(p1) <- S3
(S1, S2, S4).request(p1, p3) <- S3
session_establishment (S3-S1, S3-S2, S3-S4)

(S1, S2, S3)_share() <- S4
(S1, S2, S3).request(p1, p2, p3) <- S4
session_establishment (S4-S1, S4-S2, S4-S2)
```

4.3. Merkle Directed Acyclic Graphs (DAGs)

The IPFS employs Merkle DAGs, wherein each node has a unique hash representing its content. This hash is used to identify an object or node and serves as a representation of the data location (Figure 7).

The Merkle DAG comprises three important features: content addressing, tamper resistance, and deduplication.

- Content addressing: All content has self-organizing links, and their integrity is verified through their multihash checksums.
- Tamper resistance: The self-integrity of the content is checked using a checksum. In cases of forgery, the hash value of the Merkle root is changed; thus, the integrity is automatically checked.

- Deduplication: Content cannot be duplicated in a Merkle DAG because identical content will have the same hash value, which is not allowed.

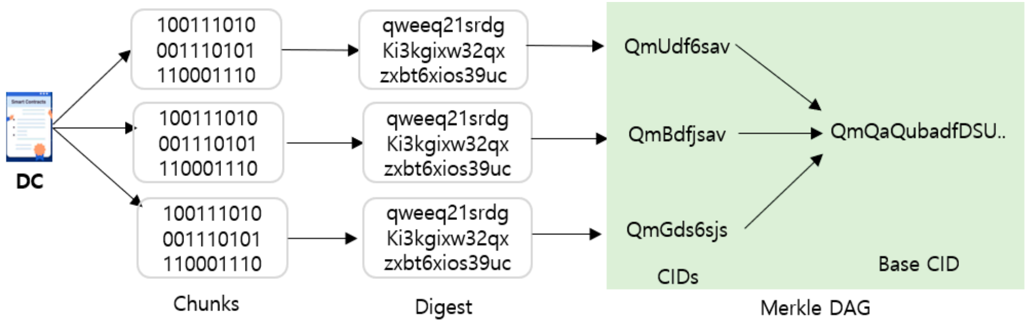


Figure 7. Structure of a Merkle DAG.

4.4. Self-Certifying File System (SFS)

The SFS is the underlying technology for enforcing the interplanetary name system (IPNS) and follows the syntax /sfs/location:HostID. The user verifies that the server matches the address based on the public key of the server (Figure 8).

- Location: server address
- HostID: hash_value(public _key(from Server), location)

The IPNS used in SFS creates a Merkle DAG with filenames comprising the hashed values of files. Every file has a permanent, immutable name; however, sometimes a mutable name is required. Hence, the IPNS can be used to generate mutable names in the IPFS.

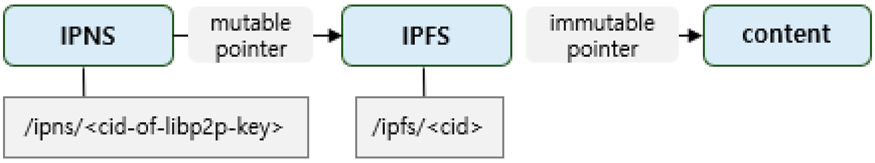


Figure 8. Content path.

The IPNS records can point to mutable or immutable paths; however, the IPFS can only handle immutable paths because the CIDs contain multihashes.

Ipfs = immutable *pointer => content

ipns = **pointer => content

5. Experiment

The IPFS structure consists of multiple nodes, each organized on a peer-to-peer network, wherein identical DCs have the same CID. It requests the required DC from multiple nodes to obtain a file and involves the following steps:

Step 1: Unique identification via content addressing.

- Locates a file using the CID, which is the hash of the file, and connects directly to the nodes where the content is stored to obtain the file.

Step 2: Content linking via DAGs.

- The IPFS splits the DC into multiple blocks and stores them as DAGs.
- When DC is uploaded using the DAG builder, it is separated into three blocks: DC1, DC2, and DC3.
- Each DC is created as a single file using the BitTorrent method, and each DC block generates its own CID using a hash value.

Step 3: Content discovery via DHTs.

- To find a file, each node uses the DHT to determine the node containing the desired file.

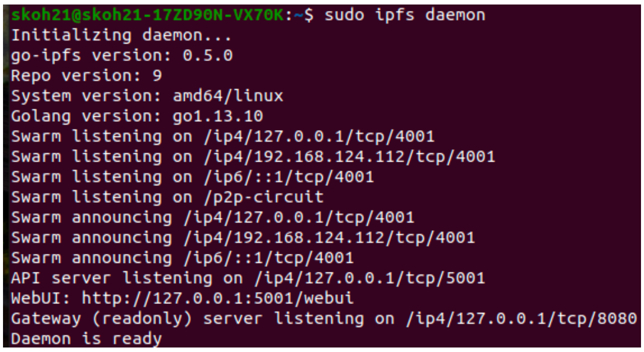
- A query is sent to the DHT using the library peer-to-peer (libp2p) framework. Once the file is located, it connects to the nodes through the Bitswap module.
- The blocks are acquired using the want list (list of desired blocks).

Step 4: Verification.

- The arriving blocks are hashed to obtain their CIDs, which are verified as the requested CIDs.
- The content delivery process is verified by checking whether the CID values match.

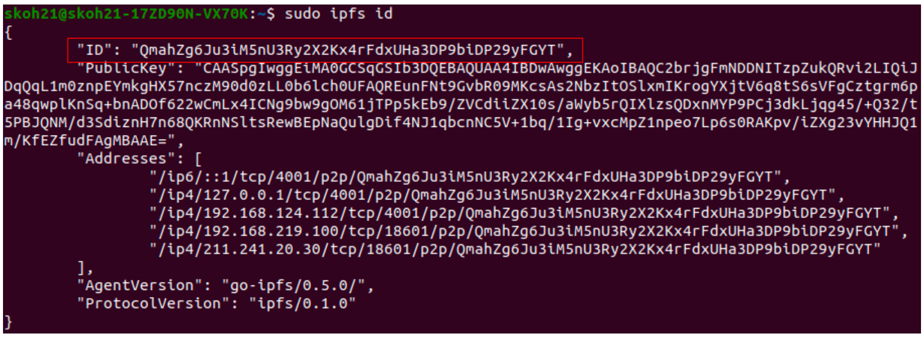
5.1. IPFS

For the experiments, both the server and node were implemented on Ubuntu 18.20. It employed the characteristics of the IPFS, and the experiments were conducted using the go-ipfs package. The component is organized as a peer-to-peer (P2P) network with nodes in the network. All identical files had the same CID, and the files were obtained by requesting them from multiple nodes where they were stored. First, the IPFS daemon (sudo ipfs daemon) was run, as shown in Figure 9, and the CID was verified using the "ipfs ID" value (Figure 10).



```
skoh21@skoh21-17ZD90N-VX70K:~$ sudo ipfs daemon
Initializing daemon...
go-ipfs version: 0.5.0
Repo version: 9
System version: amd64/linux
Golang version: go1.13.10
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip4/192.168.124.112/tcp/4001
Swarm listening on /ip6/::1/tcp/4001
Swarm listening on /p2p-circuit
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip4/192.168.124.112/tcp/4001
Swarm announcing /ip6/::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
WebUI: http://127.0.0.1:5001/webui
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready
```

Figure 9. Running the IPFS daemon.



```
skoh21@skoh21-17ZD90N-VX70K:~$ sudo ipfs id
{
        "ID": "QmahZg6Ju3iM5nU3Ry2X2Kx4rFdxUHa3DP9biDP29yFGYT",
        "PublicKey": "CAASpgIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC2brjgFmNDDNITzpZukQRvi2LIQiJ
DqQqL1m0znpEYmkgHX57nczM90d0zLL0b6lch0UFAQREunFNt9GvbR09MKcsAs2NbzItOSlxmIKrogYXjtV6q8tS6sVFgCztgrm6p
a48qwplKnSq+bnADOf622wCmLx4ICNg9bw9gOM61jTPp5kEb9/ZVCdiiZX10s/aWyb5rQIXlzsQDxnMYP9PCj3dkLjqg45/+Q32/t
5PBJQNM/d3SdiznH7n68QKRnNSltsRewBEpNaQulgDif4NJ1qbcnNC5V+1bq/1Ig+vxcMpZ1npeo7Lp6s0RAKpv/iZXg23vYHHJQ1
m/KfEZfudFAgMBAAE=",
        "Addresses": [
                "/ip6/::1/tcp/4001/p2p/QmahZg6Ju3iM5nU3Ry2X2Kx4rFdxUHa3DP9biDP29yFGYT",
                "/ip4/127.0.0.1/tcp/4001/p2p/QmahZg6Ju3iM5nU3Ry2X2Kx4rFdxUHa3DP9biDP29yFGYT",
                "/ip4/192.168.124.112/tcp/4001/p2p/QmahZg6Ju3iM5nU3Ry2X2Kx4rFdxUHa3DP9biDP29yFGYT",
                "/ip4/192.168.219.100/tcp/18601/p2p/QmahZg6Ju3iM5nU3Ry2X2Kx4rFdxUHa3DP9biDP29yFGYT",
                "/ip4/211.241.20.30/tcp/18601/p2p/QmahZg6Ju3iM5nU3Ry2X2Kx4rFdxUHa3DP9biDP29yFGYT"
        ],
        "AgentVersion": "go-ipfs/0.5.0/",
        "ProtocolVersion": "ipfs/0.1.0"
}
```

Figure 10. IPFS IDs obtained.

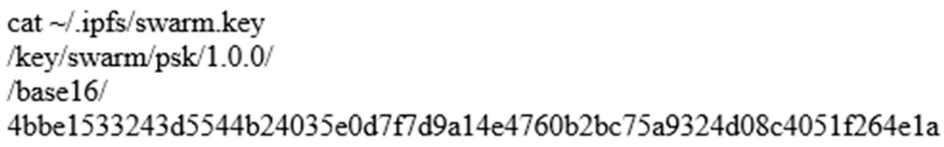
The ID of each node was verified using the same method. As shown in Figure 10, the IDs of nodes 1, 2, 3, and 4 were as follows:

Node 1: QmahZg6Ju3iM5nU3Ry2X2Kx4rFdxUHa3DP9biDP29yFGYT
Node 2: QmWFSjvfttNUbzuW1MmhTfuS3nB74zZucdnkrSJv4EDgsw
Node 3: QmRqHonzHNQDjnEw3UexhqKEKCQyY5mUH9DueAKipcuSbj
Node 4: QmiM5afZaAfRat5yX2W1ucdnzZH3aAagZGeqGAaat4A4ax

5.2. Grouping in a Private Network

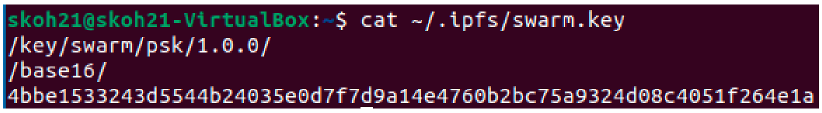
To join a group for content sharing, a swarm key is required, which is a private key shared among peers to create a private network, i.e., to form a group by owning the same swarm key. In this study, four nodes were used to create a group. Thus, all four nodes shared the same swarm key (node 1).

As shown in Figures 11 and 12, nodes 1 and 2 share the same key. Moreover, nodes 3 and 4 also share the same key.



```
cat ~/.ipfs/swarm.key
/key/swarm/psk/1.0.0/
/base16/
4bbe1533243d5544b24035e0d7f7d9a14e4760b2bc75a9324d08c4051f264e1a
```

Figure 11. Swarm key generation at node 1.



```
skoh21@skoh21-VirtualBox:~$ cat ~/.ipfs/swarm.key
/key/swarm/psk/1.0.0/
/base16/
4bbe1533243d5544b24035e0d7f7d9a14e4760b2bc75a9324d08c4051f264e1a
```

Figure 12. Swarm key of node 2.

After sharing the swarm key, the nodes connected to each node are checked. The following example shows the information of the nodes connected to node 3 obtained using the ipfs bootstrap list command.

```
skoh21@skoh21-VirtualBox:~/.ipfs$ ipfs bootstrap list
/dnsaddr/bootstrap.libp2p.io/p2p/QmNnooDu7bfjPFoTZYxMN
LWUQJyrVwtbZg5gBMjTezGAJN
/dnsaddr/boot
strap.libp2p.io/p2p/QmQCU2EcMqAqQPR2i9bChDtGNJchTbq5TbXJJ16u19uLTa
/ip4/104.131.131.82/tcp/4001/p2p/QmaC
pDMGvV2BGHeYERUEnRQAwe3N8SzbUtfsmvsqQLuvuJ
/ip4/104.131.131.82/udp/4001/quic/p2p/QmaC
pDMGvV2BGHeYERUEnRQAwe3N8SzbUtfsmvsqQLuvuJ
```

However, the nodes with which the swarm key is shared are not listed. Thus, the ipfs config address.API/gateway command is used to register them in the bootstrap.

```
skoh21@skoh21-VirtualBox:~/.ipfs$ ipfs config Address.API /ip4/10.0.2.15/tcp/5001
skoh21@skoh21-VirtualBox:~/.ipfs$ ipfs config Address.Gateway /ip4/10.0.2.15/tcp/8080
```

The following information is registered on node 2 after all commands have been executed:

```
skoh21@skoh21-VirtualBox:~$ ipfs id
{
    "ID": "QmWFSjvfttNUbzuW1MmhTfuS3nB74zZucdnkrSJv4EDgsw",
    "PublicKey":
"CAASpgIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoI
BAQD6bIaT8/t13lCkJFZsQNoaCWyTOfqrFBImzSTRGDiaivhBhXwA
OMKakUr8q+HvQlI39nxuB/s/qaZjrniZJvPqN
dEzRs4zjVcjXBkfmWBdYL8FCU5ZhWv9RDThn44ThjI1/xv2pkw0JCAeUEaF14z44
Dk5XlCKBIKb+TFdVMIYs0axFhsLt5bm1wTmBUrQw/sYocH
dfEy49GZVD9B6EezBVu7shbePrN2csROnxTOQK3TAA+GLsIN
RoVCH93YkosxhmhbfxlkppFThc1LwDuqj50GpDAYem8oBCQRK
miFj4AYHT0Asz3FIAb0Pyuus8Kla/FiHTQgZZ1eBbvqIUqsVAgMBAAE=",
    "Addresses": [
        "/ip4/127.0.0.1/tcp/4001/p2p/QmWFSjvfttNUbzuW1MmhT
fuS3nB74zZucdnkrSJv4EDgsw",
        "/ip4/10.0.2.15/tcp/4001/p2p/QmWFSjvfttNUbzuW1MmhT
fuS3nB74zZucdnkrSJv4EDgsw",
        "/ip6/::1/tcp/4001/p2p/QmWFSjvfttNUbzuW1MmhT
fuS3nB74zZucdnkrSJv4EDgsw"
   ],
   "AgentVersion": "go-ipfs/0.5.0/",
   "ProtocolVersion": "ipfs/0.1.0"
}
```

Thereafter, the ipfs bootstrap add command is used to add node 2 to node 3.

```
skoh21@skoh21-VirtualBox:~/.ipfs$ ipfs bootstrap
add/ip4/10.0.2.15/tcp/4001/ipfs/QmWFSjvfttNUbzuW1MmhT
fuS3nB74zZucdnkrSJv4EDgsw
added /ip4/10.0.2.15/tcp/4001/ipfs/QmWFSjvfttNUbzuW1MmhT
fuS3nB74zZucdnkrSJv4EDgsw
```

Finally, node 2 is registered on node 3. Thereafter, the ipfs add dc# command is used to add a file. Once it is completed, all four nodes have the DC1, DC2, and DC3 blocks.

```
skoh21@skoh21-VirtualBox:~$ ipfs bootstrap list
/ip4/10.0.2.15/tcp/4001/p2p/QmWFSjvfttNUbzuW1MmhT
fuS3nB74zZucdnkrSJv4EDgsw
skoh21@skoh21-VirtualBox:~$ ipfs add dc1
added QmWFSjvfttNUbzuW1MmhTfuS3nB74zZucdnkrSJv4EDgsw dc1
```

5.3. Discussion

Certain characteristics of NFTs facilitate straightforward asset creation, enabling seamless verification of ownership rights through uncomplicated processes. Their inherent advantages encompass resilience against forgery, intricate traceability, and the facilitation of fractional transactions. In terms of preventing forgery, NFTs mitigate the depreciation of digital assets by rendering the replication of critical data—such as origin, ownership details, and transaction chronicles—challenging. As for traceability, the transparent nature of blockchain data permits anyone to authenticate the origin, issuance time, quantity, and ownership history of NFTs. Ownership authentication and traceability are ensured through decentralized blockchain storage, facilitating easy recovery in case of loss. Additionally, NFTs introduce the concept of fractional ownership, converting illiquid assets into liquid ones by acknowledging partial ownership of assets. Table 2 presents an analysis of NFTs from a security standpoint, outlining threats such as spoofing, tampering, repudiation, information disclosure, and denial-of-service (DoS) attacks.

Table 2 demonstrates how it has provided clues to address existing issues following this study. In response to threats from tampering with data external to the existing blockchain, the proposed method enables buyers to share both original and hashed data, allowing for early detection of manipulation. Moreover, the issue of combining hashed data with attackers' addresses has been resolved through the utilization of multi-signature usage. In addressing concerns related to transaction exploitation, they have employed privacy-preserving smart contract policies.

Spoofing poses a risk wherein attackers exploit authentication vulnerabilities, potentially pilfering users' private keys. However, the safeguarding of NFT private keys can be achieved through the secure utilization of smart contracts and cold wallets. In scenarios where an NFT's blockchain data are stored externally, there exists a susceptibility to tampering by malicious entities. Yet, sharing both the original and hashed data with the buyer allows verification through hash comparison, thus detecting any tampering attempts. Mitigating the threat of repudiation attacks—wherein attackers combine hashed data with their own address—can be accomplished through the implementation of a multi-signature method. Additionally, the security threat posed by information disclosure, arising from the exploitation of hashes and transactions by attackers, can be mitigated by adopting privacy-centric smart contracts. Consequently, addressing the issue of NFT aging errors stemming from design flaws can be tackled by rigorously verifying the smart contracts in use.

Table 2. NFT security threats and solutions.

Type	Potential Security Threat	Solution
Spoofing	Abuse authentication vulnerabilities/steal private keys	Employ NFT-enabled smart contracts/cold wallets
Tampering	Manipulate data outside the blockchain	Share both original and hashed data with buyers
Repudiation	Combining the hash data with the attacker's address	Use multiple signatures
Information Disclosure	Attackers exploit hashes and transactions	Apply privacy-protecting smart contracts
Evaluation of Privilege	NFT property errors caused by design issues	Validation of smart contracts

To achieve scalability and time efficiency in research focusing on managing digital content using non-fungible tokens (NFTs) and the interplanetary file system (IPFS), there are several elements required in terms of technical specifications and infrastructure, as shown in Table 3.

Table 3. Requirements of scalability and time efficiency.

Issue	Contents
IPFS Nodes and Network Infrastructure	- IPFS is a distributed file system that identifies data via hashes and stores them across a network. - Adequate nodes are required for data retrieval based on these hashes, necessitating a robust network infrastructure for high availability and data integrity.
Smart Contracts and Blockchain Platforms	- Utilizing a blockchain platform is essential for creating and managing NFTs. - This involves executing smart contracts to guarantee the uniqueness and ownership of NFTs, requiring consideration of security, scalability, and transaction processing capabilities.
Data Security and Encryption	- To uphold digital content security, encryption technologies are crucial. - Establishing secure storage and transmission protocols through encryption standards and robust key management systems is necessary.
Scalable Databases and Storage	- Managing and storing large volumes of files and metadata requires scalable databases and storage solutions. - Efficiently handling distributed storage, replication, and backups of data is essential.
Distributed Computing Resources	- Processing and analyzing data necessitate distributed computing resources. - Cloud computing enables the distribution of large-scale tasks, enhancing performance through parallel processing. - These technical elements ensure the functionality and stability of managing digital content using NFTs and IPFS. - Designing and implementing systems considering aspects like network establishment and maintenance, data security and integrity, as well as distributed processing and storage are critical for constructing a scalable and efficient digital content management system.

These technical elements ensure the functionality and stability of managing digital content using NFTs and the IPFS. Designing and implementing systems considering aspects like network establishment and maintenance, data security and integrity, as well as distributed processing and storage are critical for constructing a scalable and efficient digital content management system.

5.4. Future Works

5.4.1. Future Research Prospect

The Fourth Industrial Revolution refers to the transformative changes driven by the advancement and innovation of digital technologies. It signifies a revolutionary shift across industries and societies through the convergence and interaction of diverse technologies. In the context of research, the progression related to the Fourth Industrial Revolution can evolve in the directions given in Table 4.

Table 4. Future research prospects.

Issue	Contents
Integration of Digital Assets and Blockchain Technology	- The Fourth Industrial Revolution introduces new concepts of digital assets. Blockchain-based digital assets like NFTs play significant roles in various sectors such as art, media, gaming, real estate, and more. - The fusion of blockchain and NFTs can propose new business models and innovative methods for content management.
Utilization of Decentralized Technologies	- The Fourth Industrial Revolution emphasizes the importance of decentralized technologies. Systems like the interplanetary file system (IPFS) offer distributed storage methods, enhancing data reliability and security by avoiding centralization. - Such technologies are expected to provide new opportunities for trustworthy data management and sharing.
Integration with Artificial Intelligence (AI)	- The Fourth Industrial Revolution emphasizes the interaction with AI. - Content management systems based on NFTs and blockchain can utilize AI for content recommendation, analysis, security reinforcement, and innovation in various areas.
Integration with Global Networks	- The Fourth Industrial Revolution stresses global connectivity. - Hence, research can focus on confirming stability and efficiency within global networks, exploring potential applications in various regions and cultures.

The Fourth Industrial Revolution can contribute to the creation of new digital asset innovations and decentralized data management systems by integrating technologies like NFTs and the IPFS. Research exploring the potential development and industrial applicability of these technologies can accelerate the ripple effects of the Fourth Industrial Revolution.

5.4.2. Potential Future Improvements

Some potential future improvements could be considered, as given in Table 5.

These approaches aim to enhance scalability and reduce the time complexity of systems using NFTs and the IPNS. Considering these improvements can significantly advance the content of the paper and the system's overall development.

Table 5. Potential future improvements.

Issue	Contents
Efficient Distributed Storage and Access	- Research and develop more efficient data distribution and storage methods using NFT and IPNS. - This could maintain data integrity while reducing the algorithm's time complexity. Designing efficient data partitioning and storage methods could help achieve this.
Algorithm Optimization	- Research optimization of existing algorithms and processes to decrease time complexity. - Devising efficient algorithms for data access and management could enhance the overall system performance.
Enhancement of Smart Contract Functionality	- Explore expanding the functionalities of smart contracts utilizing NFT and IPNS. - Adding more features to manage and access data could improve the system's scalability.
Improvements in Security and Privacy	- Focus on enhancing data security and privacy in systems using NFT and IPNS. - Research methods to strengthen encryption techniques, access control, and secure data transmission and storage.
Improving Network Scalability	- Consider connecting NFT and IPNS to more users and enhancing scalability for large-scale traffic. - Researching efficient network structures and data processing methods could enhance the overall system performance.

5.4.3. Algorithm Optimization Approaches

There are two optimization methods for this research: IPFS storage and management optimization and smart contract optimization. First, IPFS storage and management optimization involve maximizing the effectiveness of the interplanetary file system (IPFS), a distributed file system used for storing and managing files. By optimizing data replication and caching mechanisms, it is possible to enhance data access speeds. Additionally, smart contract optimization involves streamlining the code of smart contracts associated with NFT transactions to reduce execution costs.

And the results of scalability and time complexity affect how well a system can accommodate many users and handle data. Good scalability supports numerous users and enables swift transactions, while limited scalability can lead to performance degradation and a decline in user experience. Lower time complexity signifies faster transactions and efficient data processing, thereby enhancing the user experience and the system's utility. These outcomes have implications for the practical deployment and usability of systems across various domains, such as art markets, academic research repositories, and gaming industries, necessitating the need to design and utilize systems with these considerations in mind.

5.4.4. Response Time, Scalability, Time Complexity Analysis

The Table 6 shows the result of the average response time, Table 7 is for the result of the load condition. And Table 8 shows the Input Message Size(N) and Execution Time (ms).

- Response time: Response time measures the duration taken by the system to receive a request, process it, and return a response. Through experiments, requests are sent to the system, and the response times for these requests are measured to determine the average, maximum, and minimum response times. This allows us to understand the distribution of response times for each type of request.
- Scalability: Scalability refers to the system's ability to maintain performance while handling increasing loads. Through experiments, the system's response times are

measured under various load conditions, analyzing performance changes with increasing load. By comparing response times across scenarios like low, moderate, and high loads, the system's scalability is assessed.

- Time complexity: Time complexity represents the time taken by a specific algorithm based on the input size. By implementing the algorithm and varying the input size, one can measure the execution time. These data are presented in graphs or tables to display the algorithm's execution time over time, allowing us to understand the relationship between input size and the algorithm's time-based performance.

Table 6. The result of the average response time.

Avg. Response Time (ms)	Max. Response Time (ms)	Min. Response Time (ms)
30	70	20

Table 7. The result of the load condition.

Load Condition	Avg. Response Time (ms)	Max. Throughput (Requests/min)
Low load	20	800
Moderate Load	40	600
High Load	80	400

Table 8. Input Message Size(N) and Execution Time (ms).

Input Message Size (N)	Execution Time (ms)
100	5
500	15
1000	30

6. Conclusions

NFTs contain metadata, such as token number, title, content, and image URL, and are characterized as having a unique value distinct from the digital asset, making them irreplaceable. As NFTs are stored on the blockchain, they cannot be deleted or forged after they are created. Therefore, they can be used to authenticate original content and prove the ownership of digital assets. Hence, they are being increasingly used to specify the ownership and copyright of DC. The IPFS, which has the characteristics of a shared distributed file system, is used to employ these features of NFTs. The IPFS issues NFTs using a smart contract based on the ERC-721 standard and the Solidity programming language. Thereafter, the token metadata are uploaded to the IPFS distributed file system and converted into a CID value, which is stored on the blockchain. Storing the CID value on the blockchain enables more efficient data management and the decentralization of web services as the data are not stored on a central server. This study described and experimented with an NFT-based DCM method that can manage and share DC across four nodes. The experiment results showed that the DC defines a unique CID value and also confirmed that the content location can be identified using the hash value instead of web link information. Moreover, the interconnection and sharing of DC in an NFT-based IPFS were confirmed; however, the experiment was conducted using a testbed. Therefore, in future work, we will focus on proving that the same results can be obtained on a global network. Furthermore, based on the findings of this study, we will also conduct experiments with and verify the proposed method across various domains.

Author Contributions: H.K.: conceptualization of this study and methodology; J.O.: analysis of results with mathematics; S.U.K.: project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Juhee Oh and Sung Uk Kim are employed by the AINTCHAIN SOFT Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Hwang, H.J.; Park, Y.J. Blockchain-Metaverse-NFT using Web 3.0 platform should be paid attention to virtual assets. *Donga Ilbo*, 28 June 2022.
2. Hwang, Y.A.; Han, H.Y. Metaverse Trends and Educational Suggestions: Focusing on Keyword Network Analysis. *Future Educ. Res.* **2022**, *12*, 51–69. [CrossRef]
3. Go, D.W. An efficient regulatory system that reflects the characteristics of cryptographic assets is needed. *Newspaper*, 2 November 2022.
4. Lee, I.B. *The Future of Metaverse That Has Already Begun-Another World Created by NFTs and Virtual Reality*; Thousand Trees Forest: Seoul, Republic of Korea, 2021.
5. Hong, K.H. *NFT Future Class-New Opportunity to be Created by the Digital Economy Ecosystem*; Korea Economic Daily: Seoul, Republic of Korea, 2022.
6. Hyun, S.J. Issuance of NFTs and discussion of copyright issues. *Bus. Law* **2022**, *32*, 433–463.
7. Taherdoost, H. Non-Fungible Tokens (NFT): A Systematic Review. *Information* **2022**, *14*, 26. [CrossRef]
8. Wang, Q.; Li, R.; Wang, Q.; Chen, S. Non-fungible token (NFT): Overview, evaluation, opportunities and challenges. *arXiv* **2021**, arXiv:2105.07447.
9. Wilson, K.B.; Karg, A.; Ghaderi, H. Prospecting non-fungible tokens in the digital economy: Stakeholders and ecosystem, risk and opportunity. *Bus. Horiz.* **2022**, *65*, 657–670. [CrossRef]
10. Rehman, W.; e Zainab, H.; Imran, J.; Bawany, N.Z. NFTs: Applications and challenges. In Proceedings of the 2021 22nd International Arab Conference on Information Technology (ACIT), Muscat, Oman, 21–23 December 2021; pp. 1–7.
11. Maouchi, Y.; Charfeddine, L.; El Montasser, G. Understanding digital bubbles amidst the COVID-19 pandemic: Evidence from DeFi and NFTs. *Finance Res. Lett.* **2022**, *47*, 102584. [CrossRef] [PubMed]
12. Sung, S.R.; Hoeffer, R.; McLaughlin, S. *NFT Revolution—The Birth of a New Economic Ecosystem that Crosses Reality and the Metaverse*; The Quest: Seoul, Republic of Korea, 2021.
13. Yoon, H.J. Hot NFT-how much do you know? *Magazine Hankyung*, 21 March 2022.
14. Çağlayan Aksoy, P.; Özkan Üner, Z. NFTs and copyright: Challenges and opportunities. *J. Intellect. Prop. Law Pract.* **2021**, *16*, 1115–1126. [CrossRef]
15. Mukhopadhyay, M. Golden brush and evolving canvas—Navigating the digital art and Non-fungible tokens. *J. Inf. Technol. Teach. Cases* **2023**. [CrossRef]
16. Bhujel, S.; Rahulamathavan, Y. A survey: Security, transparency, and scalability issues of nft's and its marketplaces. *Sensors* **2022**, *22*, 8833. [CrossRef] [PubMed]
17. Gupta, Y.; Kumar, J.; Reifers, D.A. Identifying security risks in NFT platforms. *arXiv* **2022**, arXiv:2204.01487.
18. Kugler, L. Non-fungible tokens and the future of art. *Commun. ACM* **2021**, *64*, 19–20. [CrossRef]
19. Casale-Brunet, S.; Ribeca, P.; Doyle, P.; Mattavelli, M. Networks of Ethereum Non-Fungible Tokens: A graph-based analysis of the ERC-721 ecosystem. In Proceedings of the 2021 IEEE International Conference on Blockchain (Blockchain), Melbourne, Australia, 6–8 December 2021; pp. 188–195.
20. Bao, H.; Roubaud, D. Recent Development in Fintech: Non-Fungible Token. *FinTech* **2021**, *1*, 44–46. [CrossRef]
21. Daniel, E.; Tschorsch, F. IPFS and Friends: A Qualitative Comparison of Next Generation Peer-to-Peer Data Networks. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 31–52. [CrossRef]
22. Dwivedi, S.K.; Amin, R.; Vollala, S. Smart contract and ipfs-based trustworthy secure data storage and device authentication scheme in fog computing environment. *Peer Peer Netw. Appl.* **2023**, *16*, 1–21. [CrossRef]
23. Yue, Y.; Li, X.; Zhang, D.; Wang, S. How cryptocurrency affects economy? A network analysis using bibliometric methods. *Int. Rev. Financ. Anal.* **2021**, *77*, 101869. [CrossRef]
24. Kodeco. Introducing CryptoKit. Available online: https://www.kodeco.com/10846296-introducing-cryptokit (accessed on 27 July 2020).

25. Jonáš, J. Economic Consequences of Cryptocurrencies and Associated Decentralized Systems. Bachelor's Thesis, Masaryk University, Brno, Czech Republic, 2015.
26. Sarkodie, S.A.; Ahmed, M.Y.; Owusu, P.A. COVID-19 pandemic improves market signals of cryptocurrencies–evidence from Bitcoin, Bitcoin Cash, Ethereum, and Litecoin. *Finance Res. Lett.* **2022**, *44*, 102049. [CrossRef] [PubMed]
27. Kumar, S.; Bharti, A.K.; Amin, R. Decentralized secure storage of medical records using Blockchain and IPFS: A comparative analysis with future directions. *Secur. Priv.* **2021**, *4*, e162. [CrossRef]
28. Athanere, S.; Thakur, R. Blockchain based hierarchical semi-decentralized approach using IPFS for secure and efficient data sharing. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 1523–1534. [CrossRef]
29. Kang, P.; Yang, W.; Zheng, J. Blockchain private file storage-sharing method based on IPFS. *Sensors* **2022**, *22*, 5100. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.