

Dynamic Management of Hierarchical NFTs: Efficient Splitting and Merging

Nitin Bhagat*, JongWook Bae**, Su-Hyun Lee**

*Student, Dept. of Computer Engineering, Changwon National University, Changwon, Korea

**Professor, Dept. of Computer Engineering, Changwon National University, Changwon, Korea

[Abstract]

As digital assets continue to become more complex, effective ownership management and fractional ownership have become important challenges. This paper proposes a dynamic approach to manage hierarchical NFTs through efficient splitting and merging mechanisms. The proposed model builds a parent-child relationship framework based on the ERC-6150 standard. Children inherit the roles of their parents and further split to create new children and share ownership within the same hierarchy. Conversely, through merging, users can consolidate multiple child NFTs into a single parent NFT to simplify ownership and maximize asset value. The proposed hierarchical model improves existing hierarchical NFT models to manage the tasks of splitting and merging NFTs while maintaining the parent-child relationship, providing a scalable solution for managing complex digital assets and contributing to the widespread use of blockchain technology.

▶ **Key words:** Non-Fungible Token, Hierarchical NFT, ERC-6150 Standard, Blockchain, Ethereum, Smart Contract, Splitting and Merging

[요약]

디지털 자산이 점점 더 복잡해짐에 따라 효과적인 소유권 관리와 분할 소유권이 중요한 과제가 되었다. 이 논문은 효율적인 분할 및 병합 메커니즘을 통해 계층적 NFT를 관리하는 동적인 접근 방식을 제안한다. 제안된 모델은 ERC-6150 표준을 기반으로 부모-자식 관계 프레임워크를 구축한다. 자식은 부모의 역할을 상속받고 더 나아가 분할하여 새로운 자식을 만들고 동일한 계층 내에서 소유권을 공유한다. 반대로 병합을 통해 사용자는 여러 자식 NFT를 단일 부모 NFT로 통합하여 소유권을 단순화 할 수 있다. 제안된 계층적 모델은 기존 계층적 NFT 모델을 개선하여 부모-자식 관계를 유지하면서 NFT를 분할하고 병합하는 작업을 관리하고 복잡한 디지털 자산을 관리할 수 있도록 하여, 블록체인 기술의 광범위한 사용에 기여하기 위한 확장 가능한 솔루션을 제공한다.

▶ **주제어:** 대체불가토큰(NFT), 계층적 NFT, ERC-6150 표준, 블록체인, 이더리움, 스마트계약, 분할과 병합

-
- First Author: Nitin Bhagat, Corresponding Author: Su-Hyun Lee
 - *Nitin Bhagat (bhagatnitin312@gmail.com), Dept. of Computer Engineering, Changwon National University
 - **JongWook Bae (bae@cwnu.ac.kr), Dept. of Computer Engineering, Changwon National University
 - **Su-Hyun Lee (sleep1@changwon.ac.kr), Dept. of Computer Engineering, Changwon National University
 - Received: 2025. 01. 07, Revised: 2025. 02. 10, Accepted: 2025. 02. 14.

I. Introduction

Non-fungible tokens (NFTs) have transformed the digital asset by offering a special way for ownership of digital objects, artwork, and collectibles[1, 2]. As NFTs gain its popularity across various sectors, including art, gaming, and real estate, the need for more sophisticated management solutions has become essential[3]. Hierarchical NFTs represent a significant evolution in this realm, allowing for the establishment of parent-child relationships among NFTs[4]. This structure allows better organization of assets and enables creators and collectors to manage their holdings more efficiently.

This paper aims to analyze the dynamic management of hierarchical NFTs, focused on splitting and merging through smart contracts. The splitting process allows a parent NFT to be divided into multiple child NFTs, creating a more granular ownership that meets the various user needs and market demands[5]. Conversely, the merging process enables the consolidation of child NFTs into a single parent NFT, which simplifies ownership and enhances the asset's value proposition. By using smart contract, these processes can be automated minimizing the potential human error.

By investigating efficient splitting and merging mechanisms, our research seeks to provide valuable insights into the operational dynamics of hierarchical NFTs. Understanding these processes not only contributes to the academic discourse surrounding NFTs but also offers practical implications for developers, creators, and investors in the digital asset space. The findings of our study aim to establish a framework for managing hierarchical NFTs, highlighting the benefits of smart contracts in facilitating transactions. Through this findings, we hope our work provides a path for innovative applications of hierarchical NFTs providing more organized and efficient digital asset marketplace.

This paper focus on the dynamic method of

splitting and merging of NFTs maintaining the parent-child relationship within the system. It also explores the idea for assigning the stakes between the parent and child NFTs maintaining the hierarchical relationship. Chapter 2 describes the theoretical background, explains the existing hierarchical NFTs and application areas. Chapter 3 describes the design of the proposed scheme, and implementation of the proposed scheme is described in chapter 4. Chapter 5 provides conclusion and summary of our contributions.

II. Theoretical Background

1. Hierarchical NFT

Hierarchical non-fungible tokens (NFTs) are digital assets that build on the uniqueness and ownership characteristics of traditional NFTs to form a hierarchy of NFTs through a parent-child relationship[6]. This structure allows each NFT to exist as a child of any other NFT, enabling complex asset management and ownership division.

These hierarchical NFTs as shown in <Fig. 1> consist of the following key elements.

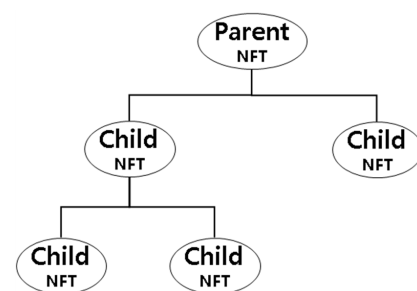


Fig. 1. Structure of Hierarchical NFTs

- 1) Parent NFT: A token that serves as the basis for other NFTs and owns the child NFT. The owner of the parent NFT has the right to issue and manage the child NFTs.
- 2) Child NFT: An NFT derived from a parent NFT, which may have specific share or attributes. Child NFTs inherit some of the characteristics of the parent NFT and can be

traded independently.

While traditional NFTs have unique ownership, hierarchical NFTs can flexibly manage ownership through parent-child relationships. This makes it easy to divide and merge assets. In addition, hierarchical NFTs introduce the concept of shares, allowing for a more detailed evaluation of the value of each NFT. In addition, they support complex asset structures, allowing for the management of various forms of digital assets (e.g., game items, artwork, etc.). While traditional NFTs are limited to a single asset, hierarchical NFTs can express the relationships between multiple assets. Through these definitions, structures, and differences, hierarchical NFTs have innovative functions in the management and trading of digital assets.

2. Existing Hierarchical NFTs

Kirillov and colleagues introduced a rental NFT protocol with advanced rewards splitting, which allows NFT owners to rent out their assets while maintaining the integrity of ownership and metadata. The protocol incorporates a novel reward-sharing mechanism based on auction dynamics, enabling market-driven determination of reward splits between owners and renters. However this method lacks for NFT rental reward splits and the reliance on informal, centralized practices[7].

D. Hawashin and colleagues proposed a scheme using composable NFTs to improve luxury food supply chain management. Their solution integrates tokenization, hierarchical NFT structure, and decentralized IPFS storage to improve traceability, transparency, and ownership. Composable NFTs establish parent-child relationships. Child NFTs, such as raw material NFTs, are embedded in packaged product NFTs, while parent NFTs, such as lot NFTs, act as containers for multiple products. This hierarchical structure ensures efficient data management and enhances integrity and collaboration in the luxury food supply chain. The proposed solution relies on composable NFTs for establishing parent child relationship, which

can be susceptible to fraudulent activities[8].

JongWook Bae and colleagues proposed a solution for hierarchical NFT using a parent-child structure designed to enhance asset management on blockchain. The parent NFT representing a basic asset creates child NFTs to distribute ownership stakes. These child NFTs possess recursive capabilities, enabling further offspring creation. This model granularizes asset ownership, facilitates complex ownership structures, and improves digital asset valuation and transparency. However, the lack of standardization for hierarchical NFTs poses compatibility challenges, as platforms and systems may struggle to recognize and interact with these NFTs without a universal protocol[9].

3. Application Areas

Hierarchical NFTs can be used in various industries that can be divided into wholes and parts. In the field of digital art, artists can set the original work as a parent NFT and issue modified works or series as child NFTs to divide and sell shares of the work. In the case of game items, a specific character can be set as a parent NFT and level-up items or modified items can be set as child NFTs, allowing players to own and trade various types of assets. In real estate management, the entire asset can be set as a parent NFT and each floor or room can be expressed as a child NFT, allowing for detailed ownership and effective management. In addition, in the music and media fields, the entire album can be set as a parent NFT and each song can be set as a child NFT, allowing the ownership of each song to be traded separately. In the education field, educational programs or certificates can be issued as hierarchical NFTs, providing a new way to manage certificates. Finally, in the case of event tickets, the entire ticket can be set as a parent NFT and VIP seats, general seats, etc. can be issued as child NFTs, allowing the owner to flexibly trade and transfer the ticket.

The dynamic management of hierarchical NFT

for splitting and merging we proposed, initiate with the creation of a parent NFT[10]. This parent NFT serves as the basic asset, initially containing 100% of stake. The parent NFT can be split to create a new child forming a parent-child structure maintaining hierarchical relationship. The child also inherits the role of parent and further split to create a new child and sharing its ownership within the same hierarchy. This granularize ownership, establishing the proportional distribution of stake from parent to child NFTs. It also allows to merge multiple child NFTs to create a new child NFT simplifying complex ownership structure. This approach facilitates scalable and dynamic NFT management offering possibilities for ownership, valuation and asset creation.

III. Design of the Proposed Scheme

1. Structure of the Proposed Hierarchical NFT

Our proposed scheme is based on the smart contract that implements a system where a single parent NFT can be split into smaller child NFTs, and these child NFTs can later be merged back into the parent NFT.

Our contract manages the ID, metadata, price, sales status, and share ratio of the NFT using various variables and mappings. It also systematically stores detailed information about each NFT, including basic information about the NFT through the struct Nfts. This basic structure increases the reliability by inheriting ERC-6150 within the smart contract, and clearly defines the parent-child relationship of the hierarchical NFT[11].

2. Minting Scheme

The creation of parent NFTs serves as a crucial foundation for the hierarchical NFT structure. New NFTs are generated during the minting process as the top-level entities, providing the basis for the ownership and management of child NFTs.

To create a parent NFT, the process involves linking it to a root NFT and establishing the appropriate relationships. This is accomplished by setting the variables 'parentOf' and 'childrenOf'. As shown in <Fig. 2> the root (Tid_0), positioned at the top, represents the highest-level NFT in the hierarchy and serves as the parent for other NFTs. The parent (Tid_1) is a child NFT of the Root, and expression $parentOf[Tid_1] = Tid_0$ indicates that Tid_1 's parent is Tid_0 , while $childrenOf[Tid_0][0] = Tid_1$ shows that Tid_0 's first child is Tid_1 .

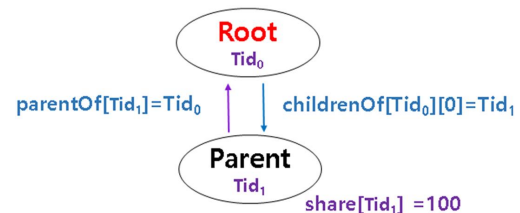


Fig. 2. The Parent NFT Minting

Additionally, the parent NFT (Tid_1) is assigned a share of 100%, as indicated by $share[Tid_1] = 100$. This allocation signifies that the NFT holds complete ownership, playing a vital role in the stake distribution of subsequently minted child NFTs.

3. Splitting Scheme

The splitting scheme involves the creation of child NFTs that derive from existing parent NFTs, establishing a clear hierarchical structure within the NFT ecosystem. During this process, each child NFT is linked to its parent NFT, facilitating the formation of parent-child relationships. This connection is managed through specific variables, such as $parentOf[Tid]$, which designates the parent NFT for each child, and $childrenOf[Tid][indx]$, which maintains an array of child NFTs associated with a particular parent. This structured approach as shown in <Fig. 3> ensures that the relationships between NFTs are well-defined and easily traceable.

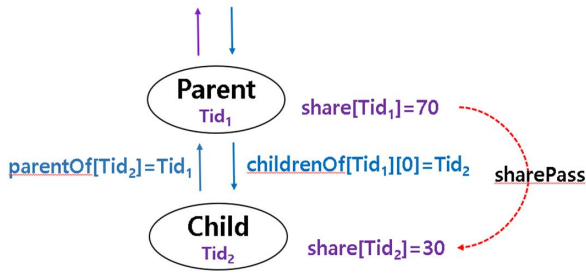


Fig. 3. A Link of Parent-Child Structure

Additionally, the splitting scheme includes the transfer of stake from the parent NFT to the child NFT. Here sharePass shown in the figure suggests a mechanism for transferring ownership shares between the parent and child NFTs. This implies that the child's ownership share, which is 30%, is derived from the parent's share. This process involves adjusting the ownership stakes as the child NFT is minted, allowing for a proportional distribution of share between the parent and child. By implementing this transfer mechanism, the system can effectively manage ownership rights and facilitate the valuation of NFTs within the hierarchical structure.

4. Merging Scheme

The merging scheme we propose refers to the process of combining multiple child NFTs into a single new parent NFT to consolidate ownership. As shown in <Fig. 4> a parent NFT is Parent₁ and its associated child NFTs are specified by Child₁, Child₂, Child₃, and Child₄.

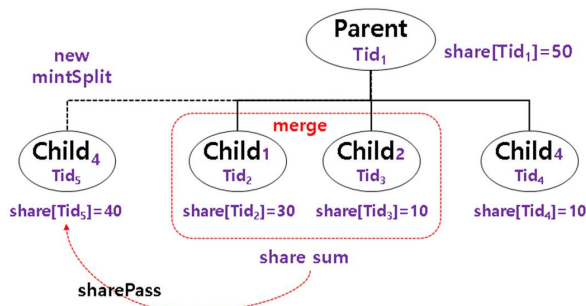


Fig. 4. Merging Child NFTs

Parent holds a share of 50%, indicating its significant ownership within the structure. The child NFTs are connected to Parent, with Child₁, Child₂, and Child₃ having shares of 30%, 10%, and 10%, respectively. Child₄, which is newly minted, has a share of 40%. SharePass signifies a mechanism for managing the distribution of ownership shares, allowing for the transfer of shares among the child NFTs and the parent NFT.

Once the shares have been summed, the specified child NFTs are burned, effectively removing them from circulation. The concept of "burn" as shown, indicates that certain shares may be removed within the hierarchy. The share amounts are summed up, and the "share sum" suggests that the total shares of the child NFTs (Child₁ and Child₂) must equal the shares allocated to Child₄. This new Child₄ NFT is then associated with the specified parent ID, completing the merging process.

5. Burn Scheme

When deleting NFTs, parent and child NFTs are treated differently to reflect their roles in the hierarchy. If the NFT being deleted has a parent ID of 0, it is identified as a parent NFT. In this case, the ownership share of the parent NFT is set to 0 before it is burned. This step ensures that the NFT's ownership is effectively nullified, preventing any lingering claims or associations after its deletion.

In contrast, if the NFT being deleted has an associated parent, its ownership share is not simply discarded. Instead, the ownership share is transferred to the parent NFT prior to the burning process. This transfer of shares reinforces the hierarchical relationship, ensuring that the parent NFT retains an accurate representation of its ownership stake even after the child NFT is removed.

IV. Implementation

1. A Smart Contract of the Proposed System

We have named the smart contract of our proposed work 'CWNFT'. Our 'CWNFT' provides a flexible asset management environment for users by providing various functions as shown in <Fig. 5>. You can issue a parent NFT using the 'mintParent' function, utilize 'mintSplit' to split the shares of the parent NFT and issue a child NFT, and apply 'mintMerge' to combine the shares of multiple child NFTs into one. Additionally, the 'sharePass' function facilitates the transfer of shares within this process.

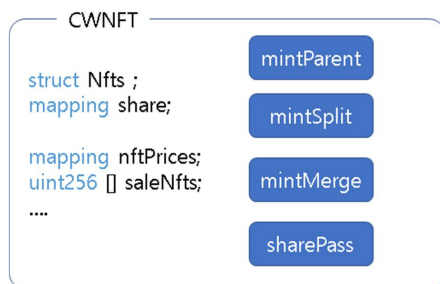


Fig. 5. Overview of Our CWNFT

Our NFT inherits the ERC721Enumerable and ERC-6150 protocols as shown in <Fig. 6>, and has a structure that effectively supports the issuance, management, and trading of NFTs. ERC721Enumerable allows you to easily query the asset list of NFT owners, and ERC-6150 is a protocol for hierarchical NFTs, allowing NFTs to form parent-child relationships.

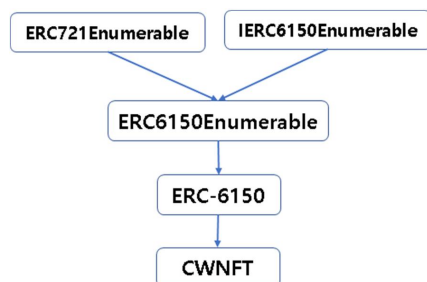


Fig. 6. Inheritance Structure of Standard Protocols

2. NFT Minting

2.1 Minting Page

As illustrated in <Fig. 7>, our platform's minting page simplifies the NFT creation process for users. Users initiate the process by selecting the digital asset they want to transform into an NFT. They enter the necessary fields, such as Title, Category, Description, and Choose File, and log in to their MetaMask wallet. When the user clicks the [Minting] button, an exception check is made to ensure that all the necessary fields are filled out correctly and triggers smart contract within our system to call mintParent() function.

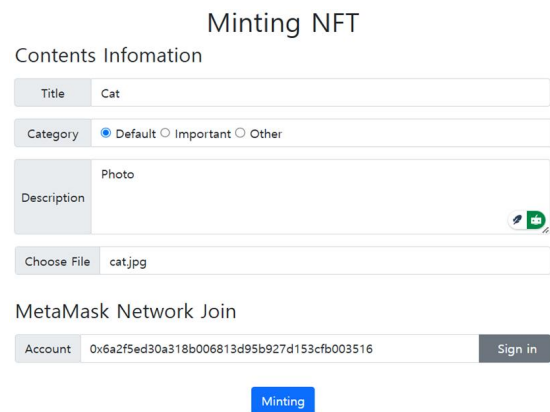


Fig. 7. NFT Minting Page Workflow

Using the _safeMintWithParent function inherited from ERC-6150, you can create a new parent NFT on the blockchain. This function mints the token and sets its parent ID to 0, indicating that it is a root NFT. The ownership share for this newly created NFT is set to 100%, signifying that the user who minted the NFT has complete ownership, allowing them to manage and control the token without any shared ownership.

```

function mintParent(string memory _tokenURI) payable
public {
    tokenIds.increment();
    uint256 tokenId = tokenIds.current();
    _safeMintWithParent(msg.sender, 0, tokenId);
    share[tokenId] = 100;
    tokenURIs[tokenId] = _tokenURI;
}
    
```

2.2 List View Page

Users can view and manage their minted NFTs on the "My NFT" page as shown in <Fig. 8>. This page lists the NFTs owned by specific user and allows them to monitor and control their NFTs within our system.

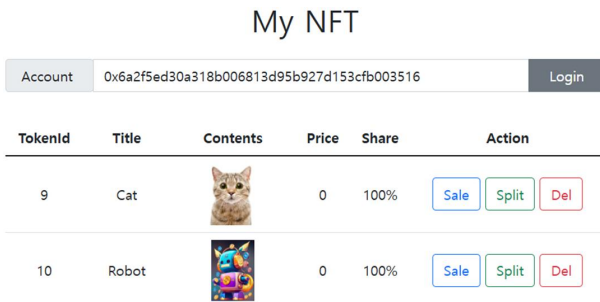


Fig. 8. List View of Minted NFTs

3. NFT Splitting

When you select an NFT on the "My NFT" page and click the [Split] button, a pop-up window as shown in <Fig. 9> will appear. Here you can input the desired ownership share. By clicking the [Add child] button, you initiate the creation of a child NFT, which enables the splitting of ownership shares.

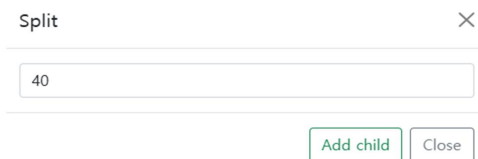


Fig. 9. Pop-Up Window for NFT Splitting

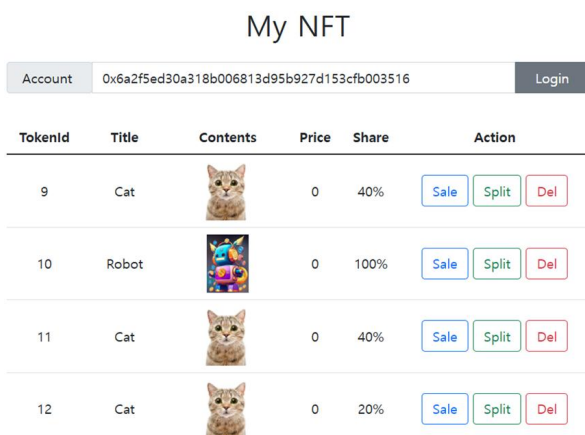


Fig. 10. Example of Splitting Ownership Shares in Cat NFT

<Fig. 10> shows an example of minting two child NFTs in Cat NFT and splitting the ownership shares into 40% and 20% respectively. Token IDs 9, 11, 12 are all Cat NFTs with the ownership shares of 40%, 40% and 20% respectively. Token ID 10 is Robot NFT with the ownership shares of 100%.

This is achieved using the mintSplit() function. The process inherits the _safeMintWithParent() function from ERC-6150 to mint the child NFT and connect it to the parent NFT with the parent ID.

```
function mintSplit(uint parentId, uint8 _share) payable public
{
    tokenIds.increment();
    uint256 tokenId = tokenIds.current();
    _safeMintWithParent(msg.sender, parentId, tokenId);
    sharePass(parentId, tokenId, _share);
    tokenURIs[tokenId] = tokenURI(parentId);
}
```

The transfer of parent's share to the child's NFT is performed by sharePass() function. This function reduces the share specified in the parent's share variable and adds it to the child's share variable.

```
function sharePass(uint256 to, uint256 from, uint8 _share)
public {
    share[to] -= _share ;
    share[from] += _share;
}
```

4. NFT Merging

4.1 Detail Contents Page

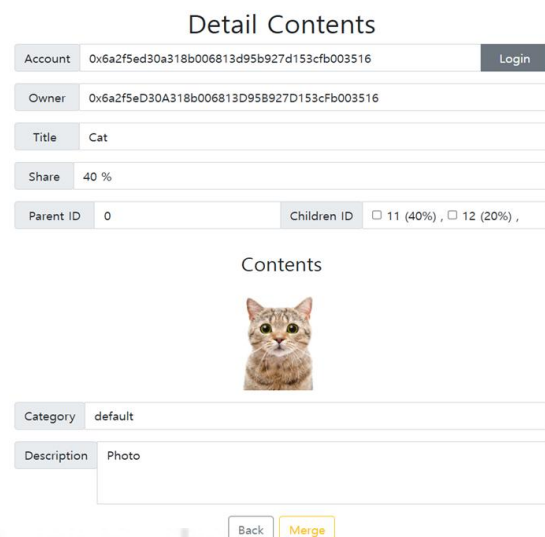


Fig. 11. Detail Contents Page for Cat NFT

When you select an NFT on the 'My NFT' page, "Detail Contents" page opens. <Fig. 11> shows the Detail Contents page containing detailed information about the Cat NFT. The share field, showing 40%, indicates the current share of the selected NFT. A Parent ID of '0' indicates that the selected NFT is a root NFT. The Children ID field displays the child NFTs of the current NFT, with token IDs 11 and 12 having respective shares of 40% and 20%.

4.2 Merging Child NFTs


When you select multiple checkbox in the "Children ID" field and click the [Merge] button, the system triggers the mintMerge() function, allowing the selected NFTs to be merged into a new child NFT under the same parent. This function combines multiple child NFTs into a single new parent NFT to consolidate ownership.

```
function mintMerge(uint256 parentId, uint256[] memory
_tokenIds) payable public returns (uint256) {
    uint8 totalShare = 0;
    for (uint256 i = 0; i < _tokenIds.length; i++) {
        uint256 tokenId = _tokenIds[i];
        totalShare += share[tokenId];
        burn(tokenId);
    }
    uint256 newTokenId = mintSplit(parentId, totalShare);
    return newTokenId;
}
```

<Fig. 12> illustrates an example of merging child IDs 11 and 12 of Cat NFT into one Child ID 13 resulting in a combined share of 60%.

Title	Cat
Share	40 %
Parent ID	0
Children ID	<input type="checkbox"/> 13 (60%) ,

Contents



Category	default
----------	---------

Fig. 12. Example of Merging Child IDs in Cat NFT

The total ownership shares of the merged child NFTs are stored in a variable, and the child NFTs are burned. If the parent ID is 0, NFT is a root and its ownership share is set to 0 before it is burned. In contrast, if the NFT has a parent, its ownership share is transferred to the parent NFT prior to being burned.

The burn() function removes an NFTs from circulation and transfers the ownership share based on its parent ID. It then calls _safeBurn() function to remove the NFT from the blockchain.

```
function burn(uint256 _tid) public {
    uint256 pid = parentOf(_tid);
    if (pid == 0)
        share[_tid] = 0 ;
    else
        sharePass( _tid, pid, share[_tid]);
    _safeBurn(_tid);
}
```

V. Conclusions

In this paper, we have proposed a solution for dynamic management of hierarchical NFTs for splitting and merging that initiates with the creation of parent NFT. This parent NFT servers as the basic asset, initially holding 100% of stake. The parent NFT is able to split and create a new child forming a parent-child structure maintaining the hierarchical relationship inherited from ERC-6150. By utilizing a clear framework for parent and child relationships, our system addresses the complexities of ownership that often arise in the current NFT landscape.

The existing NFT ecosystems frequently encounter challenges regarding flexibility and ownership management, particularly when users hold multiple assets. Our proposed splitting and merging features provide solution to this challenge. The child inherits the role of parent and further split to create a new child and sharing its ownership within the same hierarchy. Conversely the merging allows users to consolidate multiple

child NFTs into a single parent NFT, streamlining ownership and maximizing asset value. This promotes a more granular approach to ownership distribution that caters to diverse user needs.

The existing research work proposed by A. Kirillov's rental NFT protocol[7] primarily focuses on reward splitting. However, it lacks a standardized fractional ownership mechanism, rendering it unsuitable for hierarchical asset management. In contrast, our proposed splitting mechanism enables precise control over NFT shares while preserving ownership integrity. Hawashin's composable NFT-based solution[8] for supply chain management establishes parent-child relationships. This is susceptible to fraudulent activities due to inadequate verification mechanisms. Our approach addresses this vulnerability by implementing a structured ownership framework, which enhances transparency and prevents unauthorized modifications to assets. Bae's hierarchical NFT model[9] does improve asset traceability, yet the absence of a standard protocol for managing NFT hierarchies limits its compatibility. Our approach offers a more robust and adaptable solution for managing NFT ownership and distribution.

Moreover, our approach significantly contributes to the transparent and efficient valuation of digital assets. By clearly delineating ownership and management structures, our hierarchical model maximizes the advantages of blockchain technology, ensuring that users can navigate their NFT portfolios with confidence.

In conclusion, the dynamic management of hierarchical NFT model we propose offers a promising solution to the challenges of ownership management and fractional ownership in the digital asset landscape. By enabling a robust parent-child relationship combined with innovative splitting and merging functionalities, we anticipate a future where digital assets can be owned and traded more flexibly and transparently. In future, we aim to address potential challenges related to the

complexity and cost of implementing such a system, as well as the need for standardized protocols for splitting and merging of hierarchical NFTs.

ACKNOWLEDGEMENT

This research was supported by Changwon National University in 2023~2024.

REFERENCES

- [1] N. Bhagat, J. W. Bae, and S. H. Lee, "A user friendly NFT platform for Digital Assets," Proceedings of the Korean Society of Computer Information Conference, pp. 447-450, 2023.
- [2] J. S. Bellagarda and A. M. Abu-Mahfouz, "Connect2nft: a web-based, blockchain enabled nft application with the aim of reducing fraud and ensuring authenticated social, non-human verified digital identity," Mathematics, vol. 10, no. 21, Oct. 2022. DOI: 10.3390/math10213934
- [3] J. Seifoddini, "A multi-criteria approach to rating Metaverse games," Journal of Metaverse, vol. 2, no. 2, pp. 42-55, Dec. 2022. DOI: 10.57019/jmv.1053778
- [4] S. Gebreab, K. Salah, R. Jayaraman, et al, "Trusted Traceability and Certification of Refurbished Medical Devices Using Dynamic Composable NFTs," IEEE Access, vol. 11, pp. 30373-30389, Mar. 2023. DOI: 10.1109/ACCESS.2023.3261555
- [5] M. Babel, V. Gramlich, M. F. Körner, et al, "Enabling end-to-end digital carbon emission tracing with shielded NFTs," Energy Informatics, vol. 5, no. 1 pp. 22, Sep. 2022. DOI: 10.1186/s42162-022-00199-3
- [6] G. N. Vivekananda, M. A. Jarwar, M. M. Jaber, et al, "Effective two-tier tokenization for intelligent transportation supply chain systems using hybrid optimized query expansion," Multimedia Tools and Applications, Dec. 2022. DOI: 10.1007/s11042-022-14317-6
- [7] A. Kirillov, A. Rahman, and A. Aggarwal, "Rental NFT protocol with advanced rewards splitting," 2022 IEEE International Conference on Big Data, pp. 3201-3206, Dec. 2022. DOI: 10.1109/BigData55s660.2022.10020887
- [8] D. Hawashin, K. Salah, and R. Jayaraman, "Using Composable NFTs for Trading and Managing Expensive Packaged Products in the Food Industry," IEEE Access, vol. 11, pp. 10587-10603, 2023. DOI: 10.1109/ACCESS.2023.3241226
- [9] J. W. Bae, N. Bhagat, and S. H. Lee, "Hierarchical NFT using

Parent-Child Structure," Journal of The Korea Society of Computer and Information, vol. 29, no. 2, pp. 127-136, Feb. 2024. DOI: 10.9708/jksci.2024.29.02.127

- [10] D. Hawashin, M. Nemer, K. Salah, et al, "Blockchain and NFT-based traceability and certification for UAV parts in manufacturing," Journal of Industrial Information Integration, vol 39, May 2024. DOI: 10.1016/j.jii.2024.100597
- [11] K. Lee, msfew, Kartin, and qizhou, ERC-6150: Hierarchical NFTs, <https://eips.ethereum.org/EIPS/eip-6150>. accessed 2024-12-31

Authors



Nitin Bhagat received the B.E and M.E degrees in Computer Engineering from Pokhara University, Nepal, in 2004 and 2017, respectively. He joined the faculty of Computer Engineering as a Lecturer at

Purbanchal University, Biratnagar, Nepal, in 2009. He is currently a Research scholar in the blockchain technology at Changwon National University in Korea. He is interested in blockchain technology, NFT web platform design and evaluation.



JongWook Bae received the B.S. in Computer Science from Changwon University, Korea in 2001, followed by his Master's and Doctoral degrees in the same department in 2005 and 2012, respectively.

Dr. Bae currently is a lecturer in the Department of Computer Science at Changwon National University. He is interested in image processing and blockchain.



Su-Hyun Lee received the B.S. in Computer Science from Kwangwoon University, Korea in 1987. He received the M.S. and Ph.D. degrees in Computer Science from Korea Advanced Institute of Science and

Technology(KAIST), Korea, in 1989, 1994, respectively. Dr. Lee is a Professor in the Department of Computer Engineering, Changwon National University since 1996. He is interested in computer algorithm, programming languages, compiler, and blockchain.