JOIV : Int. J. Inform. Visualization, 8(3-2): IT for Global Goals: Building a Sustainable Tomorrow - November 2024 1869-1875

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Optimizing Linked List-based Smart Contract on Ethereum with IPFS for E-book Management System

Maznun Arifa Mohammadan Makhtar [a,1], Novia Admodisastro [a,2], Mohd Anuar Mat Isa [b], Daniel Hafiz Abdullah [a], Khaironi Yatim Sharif [c]

[a] *Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Selangor, Malaysia*
[b] *IExploTech Sdn. Bhd., Cyberjaya, Malaysia*
[c] *Faculty of Science and Information Technology, Universiti Teknologi Petronas, Perak, Malaysia*
Corresponding author: [1]gs65188@student.upm.edu.my; [2]novia@upm.edu.my

***Abstract*—People are now widely adopting digital assets in various applications, integrating them into almost every aspect of their lives. Electronic books, or e-books, are one of the digital assets that result from the transformation of physical reading material into the digital world. Nowadays, blockchain is used in many industries because it provides immutable and transparent records. E-book publishers may take this opportunity to adopt blockchain technology for e-book data management. However, blockchain storage is limited; thus, storing the e-book files in blockchain is not recommended. A decentralized storage system, such as InterPlanetary Files Systems (IPFS), is an alternative way to store large files like e-books. IPFS can facilitate the storage of e-book files while the metadata is stored in the blockchain. The e-book metadata should be stored in a structured way for effective search and retrieval. E-book metadata could be added, deleted, and updated occasionally. Nevertheless, some data structures often struggle with dynamic collections of records. This paper proposes a linked list-based smart contract on Ethereum that integrates with IPFS for the e-book management system. We demonstrate the implementation of a linked list smart contract for insertion, deletion, update, retrieval, and traversal of the e-book's metadata. The result shows that a linked list-based smart contract with IPFS could offer a robust solution for e-book data management. This solution provides more opportunities to explore further security and cryptography approaches toward a secure e-book management system.**

***Keywords*— Blockchain; data structures; linked list; smart contract; Ethereum; IPFS; e-book.**

Manuscript received 8 Jun. 2024; revised 17 Aug. 2024; accepted 12 Oct. 2024. Date of publication 30 Nov. 2024.

I. Introduction

The publishing industry, a cornerstone in the growth and development of knowledge, plays a pivotal role with its extensive publications such as books, magazines, journals, articles, dictionaries, and many more. It is a social responsibility to educate nations by spreading the word of knowledge through publications, which has made this industry a key player as the development of technologies is challenging. The publishing industry is also affected by technological changes. Publishers must build new business strategies and face new challenges in adapting technologies to publish reading material to satisfy customer demand. The term "e-book" was introduced to represent how books are defined in the digital world. E-books have revolutionized how people connect to their books, marking a significant shift in the reading experience.

The concept of blockchain was initially proposed by Satoshi Nakamoto in 2008 [1]. He introduced Bitcoin, a peer-to-peer electronic cash system. Bitcoin is a cryptocurrency that allows payment to be made directly without the intervention of a third party, such as a financial institution. Blockchain applications have expanded rapidly and are widely used in various fields, such as financial institutions, healthcare, manufacturing, and education. Blockchain was built on a decentralized network, which means no central authority owns it. It is managed cooperatively by nodes or a miner, which runs on a consensus mechanism to read and write the transactions into the blockchain. A study shows that approximately 40 consensus mechanisms have been developed by researchers [2]. Although there are so many consensus mechanisms, there is still a need for new mechanisms because every consensus has its limitations and cannot fit all performance parameters on the blockchain [3].

The transactions recorded in the blockchain are publicly accessible and immutable. All blocks' content cannot be removed or edited. The size of the ledger will be expanded and serve as an archival repository of the past and current repository of the present [4]. Therefore, it ensures transparency, so industries started moving to blockchain. Blockchain will mitigate all risks of having a "principle-agent" problem, which will then shape the blockchain character as "trustless" or "trust-free" [5]. Cryptographic algorithms securely protect the transactions in the blockchain. It eliminates the need for the user to disclose their identity. Blockchain can maintain high security of the transactions while users remain anonymous [6].

The Ethereum blockchain is an advanced blockchain platform built beyond the cryptocurrency introduced by Bitcoin. It was developed based on three (3) approaches: creating an entirely new blockchain, leveraging Bitcoin's scripting, and building a meta-protocol on top of Bitcoin [7]. Ethereum is a permissionless blockchain. It allows anyone to write transactions into the blockchain's block. Ethereum's enhanced capabilities can accommodate more complex applications like decentralized applications (dApps) and smart contracts.

A smart contract is a program code deployed in Ethereum. It consists of pre-determined rules that act as an agreement to be run automatically when the conditions are met. In general, the key elements of smart contracts are 1) elimination of trusted third parties, 2) forge assistants, 3) transparency, 4) autonomous execution, 5) accuracy, and 6) paper motivation [8]. The smart contract is coded using the Solidity programming language, allowing code to be written on how the process should be managed, and it provides distributed and unchangeable records of all events that have occurred [9]. Smart contract execution is anonymous without trusted centralized administration or third parties [10]. When the smart contract is executed, the transaction will be stored inside the storage, memory, and stack. Storage is a persistent key-value store between the function calls and transactions [11].

A storage system is one of the critical components in the e-book publishing system, as the sizes of e-book files will continuously increase. However, blockchain storage is limited, and the expansion of e-book data volume will increase data storage. Thus, there is a need for an alternative way of storing extensive data in decentralized storage systems such as InterPlanetary Files Systems (IPFS), Storj, SWARM, Filecoin, and Sia. Distributed Hash Table (DHT) in IPFS efficiently coordinates the data without central governance. IPFS implemented BitSwap, a peer-to-peer file-sharing system that is similar to BitTorrent. Another essential feature of IPFS is a Merkle-Directed Acyclic Graph (DAG), which links hashed data blocks to ensure identification and tamper resistance of the stored data [12]. Using Ethereum, the publisher can optimize e-book storage management by utilizing smart contracts, IPFS, and linked list data structures.

A linked list is one of the abstract data types (ADT) in programming. It is a collection of data that are grouped using a single variable. There are three (3) types of linked lists: singly linked, doubly linked, and circular. Linked lists are the base structure for other complex data structures like binary trees, AVL trees, and bandwidth trees [13]. The linked list is implemented using two (2) entities: the data and the pointer. The data holds the actual value, and the pointer holds the address of the next value. Other types of data structures are array, stack, queue, and hash table. These data structure elements will be modified based on the operations run on them (e.g., insertion, deletion, and modification) [14]. Choosing which data structures to use in any solution depends on the operations it will perform and how frequently they will be performed. The array data structure needs a fixed length of data; thus, if the length size cannot be specified, we can use a pointer implementation on a list, which is more dynamic. Array implementation may also consume more space, as we need to allocate the space beforehand. A pointer in a linked list only uses as much space as required by the elements [15].

This paper is organized as follows: Section II describes related works, introduces proposed solutions, and describes the implementation; Section III explains the results and discusses them; and Section IV describes the conclusion and future work.

II. MATERIALS AND METHOD

In this section, we will discuss initiatives and efforts done by other researchers related to IPFS storage and linked lists.

A. IPFS Storage

Files storage is essential in any field of business that deals with extensive data, such as the medical industry. A distributed file storage model using IPFS and blockchain [16] was proposed to store patients' details and medical records in the hospital. Doctors or medical personnel can access the records by first getting permission from the patient. This is done to protect the privacy of the patient's record and prevent unauthorized persons from gaining access to the patient's details.

In the music industry, music streaming has become a trend. People subscribe to music and listen to it directly on their gadgets. Digital music copyright systems based on blockchain [17] protect against fraud by providing copyright registration, authentication, and crowdfunding. The system uses IPFS to store music audio, and the hash code generated from it will be sent to the blockchain. The Shazam algorithm is used as an additional protection layer for the audio by extracting the audio fingerprint before it is added to IPFS.

In the academic field, publishing research papers is one of the crucial activities in spreading new knowledge to people. PubChain [18] is the publication platform that uses blockchain and IPFS as its base. PubChain is a decentralized open-access journal publication platform that offers incentive schemes to stakeholders such as authors, readers, and reviewers. The author will sign the article before uploading it to IPFS, and the hash value will be stored in the blockchain with the metadata file.

Archives Management Information System by [19] explores the opportunity of applying blockchain technology to improve the efficiency and security of archives management. The system was developed using Model-View-Controller (MVC), which separates the system into three main components: the interface layer (view), the business logic layer (controller), and the data access layer (model). The architecture includes modules for archives storage, file viewing, file querying, and archive verification, which interact with the private and public blockchains. In this solution, IPFS was used for file storage. However, block size constraints in blockchain can affect system performance as the volume of data grows.

B. IPFS Storage for E-Book

The Blockchain-based Authenticity of Online Publications model [20] is designed to ensure the authenticity and integrity of the published e-book. The system is built in the Ethereum blockchain and uses smart contracts to provide traceability and authenticity to the published e-book. The system involved authors, readers, one primary publisher, and sub-publishers. IPFS was used to store e-book files because it is more cost-effective than storing them in the blockchain.

The solution proposed by [21] uses the Elliptic Curve Cryptographic (ECC) to secure the data link for the smart contract's digital assets. In addition, IPFS will be used as data storage, and the ERC-721 standard token will be used for authorization. ERC-721 is a standard used for non-fungible tokens in Ethereum, which means the asset is unique. In the previous system, the data links for digital assets in the smart contract were in plain text, which was vulnerable to attack.

A study by [22] suggested the potential use of Non-Fungible Tokens (NFT) in blockchain for the publishing industry. NFT is usually used in collectible items such as games and art, but in this model, the researchers suggested that NFT be used in the publishing industry. Instead of selling an e-book, the system allows the author to sell a separate chapter of an e-book. The model suggested using IPFS for e-book storage while the hash value will be stored in the blockchain.

C. Linked List

A study by [23] analyzed the performance difference between the array list and linked list when performing operations such as `insert,` `delete,` and `search.` The study shows that the linked list takes less time than the array list to perform insertion and deletion functions. However, the `*search*` operation in the linked list consumes more time than in the array list. The linked list will operate by moving the pointer, but the array list has to increase or decrease the array size after performing the `*insert*` and `*delete*` functions. Therefore, it is concluded that a linked list is more efficient than an array list for allocating elements.

A linked list data structure for advanced admission control has been proposed by [24] to manage reservations for network resources. The linked list was compared with a bandwidth tree and time-slotted array regarding flexibility, memory efficiency, adaptability, and performance. The result shows that the linked list allows reservations to start at any time, compared to the slotted array, which depends on the predefined time intervals. The linked list has dynamic memory, which could allocate and release the outdated nodes to recycle the memory. A slotted array has constant memory usage but is efficient over long periods, while a bandwidth tree consumes more memory than a slotted array and linked list because of the complexity of managing the nodes. This makes linked list memory more efficient. Regarding adaptability, a slotted array requires multiple entries for each reservation. Bandwidth trees can handle reservations over large intervals as linked lists, but the complexity of merging and balancing in bandwidth trees makes it less efficient. A linked list performs better than a slotted array when the number of requests is insignificant. However, when the request increases, the performance of the slotted array exceeds the linked list. In this situation, the differences are negligible.

A Data Query Method of Blockchain Based on an Index proposed by [25] aims to expand the efficiency of the data queries in blockchain. To improve user demand for data queries, two (2) index layers have been developed: the first is an Abstract-Trie layer, and the second is an Operation-Record List Layer. Abstract-Trie is a clustered index in blockchain that stores original data. Operation-Record List is a singly linked list that provides quick access to the scattered operations within the blocks. The solution was implemented and tested using Hyperledger Fabric. The result shows that the proposed method slightly increases block construction time and significantly improves query performance for both original data and operations. However, implementing the new method adds more cost to the block construction.

This section proposes a conceptual model of a blockchain-based e-book management system that utilizes IPFS and a linked list smart contract for data insertion, modification, retrieval, and deletion.

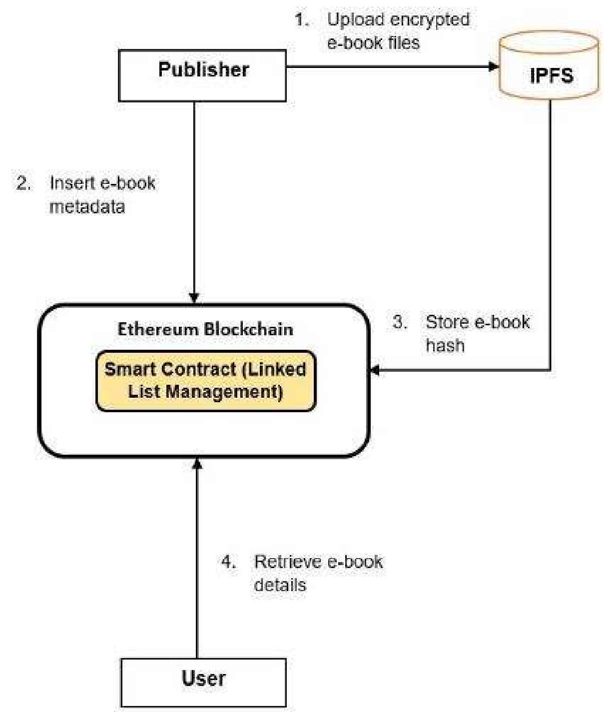

Fig. 1 Conceptual diagram of blockchain-based e-book management system

Fig. 1 illustrates the conceptual overview of the system, highlighting the entities that interact with the smart contract and related processes. Our solution focuses on the pre-sale processes: e-book uploaded into the storage system, e-book metadata and hash value updated into the blockchain, e-book details retrieved by the user, and e-book deletion. The entities involved in this process can be summarized as follows:

a. *Publisher:* an entity responsible for handling e-book publications and other processes related to them, such as editing and proofreading. The publisher creates the smart contract as an agreement to facilitate the terms and conditions for e-book publications.

b. *User:* individual who interacts with the smart contract to retrieve the details of the e-book.

c. *IPFS:* a repository system globally used in the Ethereum blockchain environment. IPFS can hold large sizes of data, and the hash value generated from IPFS will be stored in the blockchain for retrieval [26].

d. *Smart contract:* self-executed program coded based on pre-determined rules using a programming language. The smart contract will be used to store the metadata and IPFS hash in this system. Smart contract is a contract-oriented, high-level language programmed using Solidity Programming Language on a web-based Integrated Development Environment (IDE), Remix [27]. Smart contracts will handle functions such as adding, deleting, updating, and retrieving the e-book metadata.

In the next part, we will discuss every process involved in the system.

D. Upload E-book

The process begins when the publisher has prepared the final version of e-book files with metadata. Once the e-book is finalized, it will be uploaded into IPFS, a decentralized storage system working on a peer-to-peer (P2P) basis. In IPFS, a unique hash value will be generated. IPFS stores the data using a content-addressable system and distributed hash table (DHT). When the e-book file is uploaded into IPFS, the files will be chunked into smaller pieces and hashed using SHA-256 algorithm. This chunk of data will be disseminated across multiple nodes in the network [28]. Merkle DAG (Directed Acyclic Graph) links the chunks to a single root hash, ensuring data integrity and immutability. This root hash and its metadata will be stored in the blockchain. The smart contract will activate the "*addEbook*" function when new e-book metadata is inserted into the block.

E. Update E-book Metadata and Hash Value to Blockchain

After uploading the e-book files to IPFS, the publisher will create a smart contract to manage the e-book storage efficiently. The publisher will add the new e-book metadata such as title, author, price, category, and IPFS Hash. The e-book IPFS hash will be retrieved from IPFS and added to the metadata. Storing the IPFS hash into the blockchain can save storage, as the blockchains are designed to store small amounts of data efficiently. Storing and retrieving large files on blockchain will slow down the performance and scalability. The smart contract will activate the "*updateEbook*" function to update the metadata of the e-book.

F. Retrieve E-book Details

When a user enquires about the e-book details, the smart contract will access the data in the blockchain and display the e-book metadata publicly and transparently. The smart contract will call the "*readEbook"* function for e-book metadata retrieval.

G. Delete E-book Details

The "*deleteEbook*" function will be called when a request is to delete the e-book details from the blockchain. The e-book will be checked to see if it exists before the deletion process. While the conceptual diagram represents the process flow for the e-book storage system, Fig. 2 shows the entities involved in the system, how they interact with each other, and the data interaction between the entities involved.

The class diagram shows the main classes: publisher, user, e-book data struct, IPFS, and smart contract with their attributes, functions, and relationships. Publisher are the contract owner; thus, the publisher can add, modify, and delete e-books. Users can only retrieve the e-book details. The smart contract will check if the message sender is the contract owner before invoking any functions. The E-book class is a metadata class, which stores details about the e-book, including the IPFS hash value. IPFS class represents the decentralized file storage system used to store actual e-book files. It provides a function to generate a hash value. The user class denotes the end-user who interacts with the smart contract to browse and make any queries regarding the e-books. Smart contract class manages the flow related to e-book transactions. The smart contract is directly associated with all other entities: user, publisher, and IPFS. This shows the importance of smart contract in facilitating secure, transparent, and efficient e-book storage transactions and access.

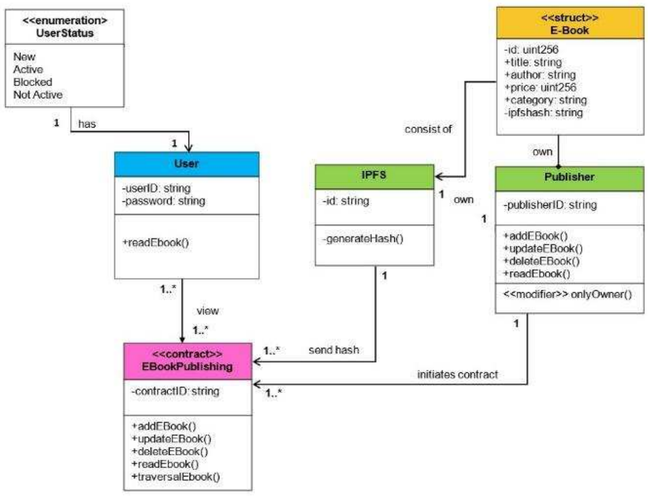

Fig. 2 Data structures represented by class diagram

We implement the proposed e-book storage model using IPFS, where each e-book is stored in IPFS storage, and the generated hash is stored in blockchain. The Elliptic Curve Digital Signature Algorithm (ECDSA) [29] was utilised for signing purposes, and the signed e-book was encrypted using Asymmetric Encryption Standard (AES) [30] before being uploaded to IPFS. Digital Signature and encryption methods were applied to strengthen the e-book's security, authenticity, and integrity. Fig. 3 shows the Content Identifier (CID) or hash value that will be generated when the signed encrypted e-book is uploaded to IPFS.

We introduce an `*eBPBM*` contract, the base contract for `*ebookstore*` smart contract, which leverages a linked list of data structures to manage e-books efficiently. The `*E-book*` struct in `*ebookstore*` contract has `*data*`, which is a string of concatenated list, `timestamp` denotes the time when the e-book is added to the blockchain, `*prev*` provides a reference to the previous e-book and `*next*' pointer referring to the next e-book. The complete solidity smart contract codes are available on GitHub under the account path name "mazarifa/Smart-Contract".

The linked lists were used to generate the smart contract. Linked lists are an effective data structure for managing dynamic and ordered collections of records in blockchain applications. The linked list is effective for the e-book management system because the record length size cannot be fixed at the beginning of list construction. Pointers allow the newly added data to be allocated directly at the end of the list. The details of the pointers will be discussed in Section V.

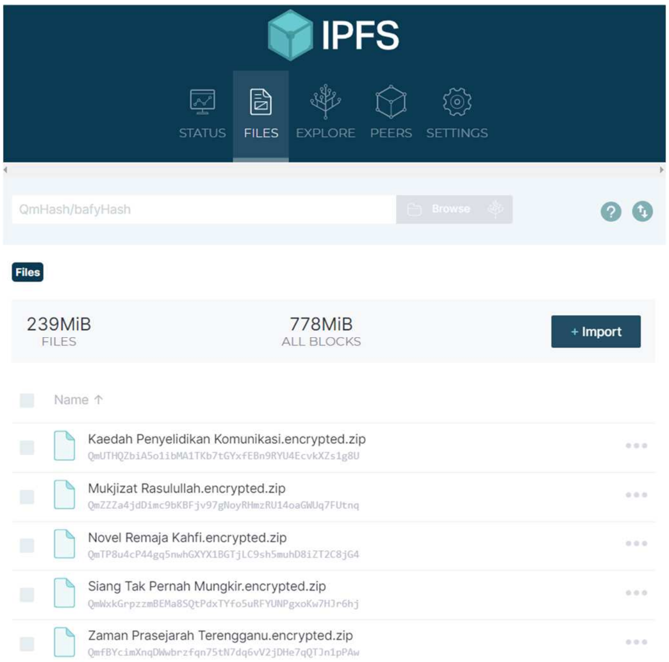

Fig. 3 E-books uploaded into IPFS

We use five (5) sets of data to demonstrate data insertion, deletion, and modification. The metadata is concatenated to the following list: title:: author:: category:: price:: ipfshash. The data strings are shown in the table below.

TABLE I
LINKED LIST DATA FOR EBPBM CONTRACT

No	Data String
1.	Kaedah Penyelidikan Komunikasi :: Che Su Mustafa :: Sains Sosial :: 28.00 :: QmUTHQZbiA5o1ibMA1TKb7tGYxfEBn9RYU4EcvkXZs1g8U
2.	Mukjizat Rasulullah :: Sayyid Muhammad Abdul Razak :: Agama :: 16.70 :: QmZZZa4jdDimc9bKBFjv97gNoyRHmzRU14oaGWUq7FUtnq
3.	Kahfi :: Rodiah Haji Said :: Sastera :: 12.00 :: QmTP8u4cP44gq5nwhGXYX1BGTjLC9sh5muhD8iZT2C8jG4
4.	Siang Tak Pernah Mungkir :: Baha Zain :: Sastera :: 8.40 :: QmWxkGrpzzmBEMa8SQtPdxTYfo5uRFYUNPgxoKw7HJr6hj
5.	Zaman Prasejarah Terengganu :: Rashid Hamat :: Sejarah :: 19.60 :: QmfBYcimXnqDWwbrzfqn75tN7dq6vV2jDHe7qQTJn1pPAw

The `eBPBM` smart contract consists of five (5) main functions; "*addEbook*", "*deleteEbook*", "*updateEbook*", "*traversalEbook*" and "*readEbook*". The first four functions can only be invoked by the contract owner, where else function "*readEbook*" can be accessed publicly by the user.

Function "*addEbook*" allows the metadata of e-books to be entered into the blockchain along with the hash value. The metadata is added as a concatenated list. When a new e-book is added, it will be appended to the end of the list. The `*next*` pointer of the current last e-book is updated to point to the new e-book, and the `*prev*` pointer of the new e-book is set to the previous last e-book. Fig. 4 shows the smart contract deployment on "*addEbook*" function.

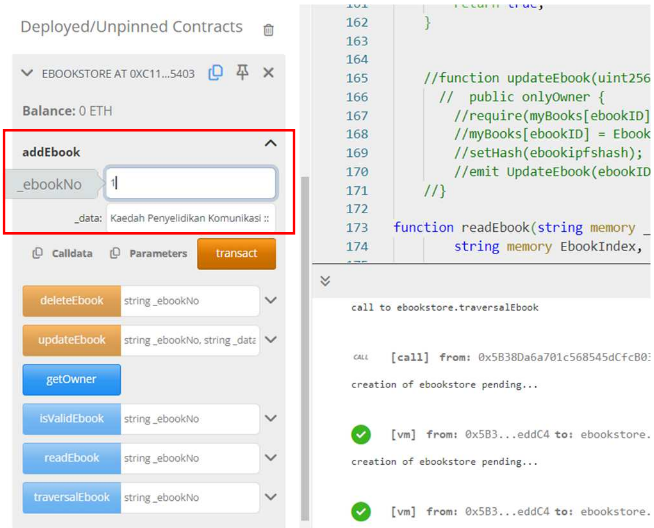

Fig. 4 Function "*addEbook*" on smart contract using Linked List

The function "*deleteEbook*" will be invoked when there is a request to delete the e-book metadata in the blockchain. It removes an e-book identified with its e-book number (`*ebookNo*`). When an e-book is deleted, the `*prev*` and `*next*` pointers of its neighboring e-books are updated to bypass the deleted e-book. This ensures the list remains intact without the deleted element. However, the deleted metadata does not delete the e-book from the IPFS.

The "*updateEbook*" function allows the owner to modify the e-book details. The e-book will be identified using its unique e-book number (`*_ebookNo*`). The function will check if the provided e-book number (`*ebookNo*`) exists in the mapping (`*mapBooks*`). If the e-book does not exist, the function will return a `*false*` value and terminate. This prevents the function from attempting to update a non-existent e-book. If the e-book exists, the `*data*` field in the `Ebook` struct will be updated to the new value. The function also updates the `*timestamp*` field to the current block. This provides a record of when the e-book was last modified.

Another important function in the linked list is "*traversalEbook*" which allows navigating through the linked list of e-books. This function will retrieve detailed information about a specific e-book, including its details, timestamp, and pointers to the previous and next e-books in the linked list. This function facilitates traversing the linked list both forward and backward, which is crucial for managing and navigating through the collection of e-books. Users can start searching from any e-book and navigate sequentially through the list.

Table II summarises the smart contract structure with a linked list for the `*ebookstore*` smart contract. The table outlines various elements of the contract, including state variables, structs, modifiers, and functions with their descriptions.

TABLE II
SMART CONTRACT STRUCTURE

Element Type	Name	Description
Contract	`eBPBM`	Base contract for basic access control functionality
	`ebookstore`	The primary contract for managing e-books
State Variables	`owner`	Stores the owner of the contract
	`mapBooks`	Mapping from e-book number (string) to `Ebook` struct for storing e-book
Modifier	`onlyOwner`	Restricts function access only to the contract owner
Struct	`Ebook`	Defines the structure of an e-book with data timestamp, previous and next fields
Function`*eBPBM*`	`getOwner`	Returns the owner's address
	`constructor`	Sets the deployer as the owner
	`concat5strPadding`	Concatenates five (5) strings with "::" as a separator
	`addEbook`	Adds new e-book to the linked list
	`readEbook`	Read the e-book details
	`updateEbook`	To modify the details of the e-book
	`deleteEbook`	Deletes an e-book from the linked list and updates pointers accordingly
	`traversalEbook`	Traverses the linked list to get e-book details
	`isValidEbook`	Check if the e-book exists

III. RESULTS AND DISCUSSION

The implementation of `*ebookstore*` smart contract, integrated with IPFS and linked list data structures, can effectively manage e-book records. Using the `*concat5StrPadding*` function, the metadata of the e-book can be effectively concatenated. Pointers in the `*Ebook*` struct (`*prev*` and `*next*`) allow traversal of the list in forward and backward navigation.

When an e-book is added, its timestamp, metadata, and pointers are stored in the blockchain, and the e-book file is uploaded into IPFS. The IPFS hash is stored as part of the e-book's metadata in the blockchain. Storing the e-book files in IPFS will reduce on-chain storage costs and enhance stability.

Updating an e-book involves modifying its metadata and updating the timestamp. The linked list structure will remain, with no changes to the pointers. If the modification involves new files updated to IPFS, a new hash will be generated, which can be updated in the blockchain.

Deleting an e-book will update the `*prev*` and `*next*` pointer of the next and previous e-book. The e-book will be successfully deleted without changing the order. The e-book files in IPFS will not be deleted if the metadata has been removed from the blockchain.

Traversal of the linked list provides quick access to the e-book records. The `*traversalEbook*` function allows the traversal of the linked list by enabling forward and backward navigation through the e-book records. Fig. 5 shows the usability of the linked list in the `*ebookstore*` function in the smart contract. When the e-book with the `*ebookNo*` 2 is retrieved from the smart contract, the metadata of the e-book will be displayed in the concatenated list: title :: author :: category :: price :: IPFS hash. The list will show the previous e-book added with the `*ebookNo*` 1 and the next e-book added with the data `*ebookNo*` 3.

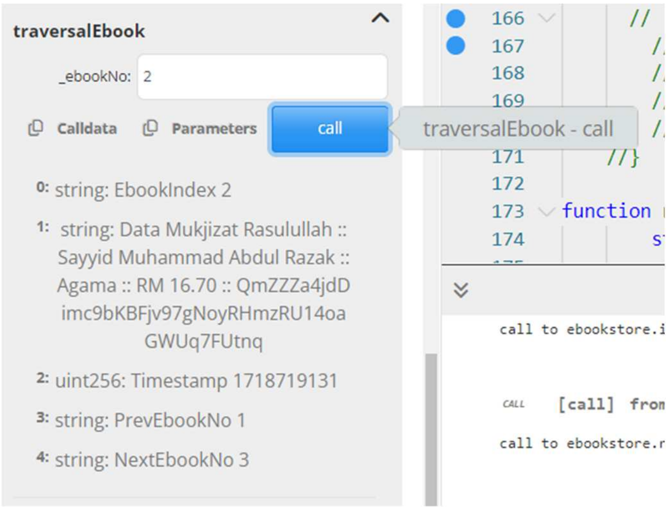

Fig. 5 Function "*traversalEbook*"

IV. CONCLUSION

Through this study, we demonstrate that IPFS can offer an alternative solution for limited storage in blockchain. IPFS provides a secure file storage system by applying SHA-256 hash algorithm and Merkle-DAG data structuring method to produce a single root hash. This solution utilized a doubly linked list, facilitating forward and backward navigation through the e-book records using the `*prev*` and `*next*` pointers. A linked list-based smart contract integrated with IPFS could offer a robust solution for decentralized content management. The linked list structure allows for efficient insertion, modification, and deletion of records, thus preserving the chronological order of e-books. This implementation optimizes the storage and enhances the usability and accessibility of e-books in the Ethereum blockchain. `eBPBM`smart contract in blockchain could enhance the capability of e-book publishers to trace e-book transactions and any changes associated with it without the need for intermediaries. These transactions are immutable and will be automatically executed and recorded. All transactions are transparent. Therefore, it is reliable. Nevertheless, the traditional contract needs human intervention and manual authorization, which is time-consuming. The recorded e-book details and transactions may be manipulated as they are not transparent.

However, based on the proposed solutions, much can still be done to enhance the e-book management system. Future development can be done by improving the query and search capabilities to allow faster search and retrieval of e-books. This can be achieved by implementing advanced query and search functions by title, author, and category. Hash verification can be applied in the smart contract to ensure the authenticity of the e-book. Additional cryptographic algorithms can be implemented to ensure secure transactions. This solution can also be extended to other digital content, such as images, videos, and journal articles. Another recommendation would be user interface development, which would include front-end integration for easy accessibility to the user.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System." [Online]. Available: www.bitcoin.org

[2] A. K. Yadav, K. Singh, A. H. Amin, L. Almutairi, T. R. Alsenani, and A. Ahmadian, "A comparative study on consensus mechanism with security threats and future scopes: Blockchain," *Comput Commun*, vol. 201, pp. 102–115, Mar. 2023, doi: 10.1016/j.comcom.2023.01.018.

[3] N. Sana and S. Uk-Jin Lee, "Why the consensus mechanism is needed in blockchain technology?," in *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*, 2020, pp. 92–99.

[4] L. Alashaikh, "Blockchain-Based Software Systems: Taxonomy Development," in *Proceedings - 2021 IEEE International Conference on Blockchain, Blockchain 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 491–498. doi:10.1109/Blockchain53845.2021.00075.

[5] P. De Filippi, M. Mannan, and W. Reijers, "Blockchain as a confidence machine: The problem of trust & challenges of governance," *Technol Soc*, vol. 62, Aug. 2020, doi:10.1016/j.techsoc.2020.101284.

[6] C. Tuteja, N. Saxena, P. Johri, and V. R. Vadi, "Blockchain Technology: A case study of its Decentralized Use," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, COM-IT-CON 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 771–775. doi: 10.1109/COM-IT-CON54601.2022.9850511.

[7] V. Buterin, "A Next Generation Smart Contract & Decentralized Application Platform."

[8] T. Hewa, M. Ylianttila, and M. Liyanage, "Survey on blockchain based smart contracts: Applications, opportunities and challenges," Mar. 01, 2021, *Academic Press*. doi: 10.1016/j.jnca.2020.102857.

[9] H. Guo and X. Yu, "A survey on blockchain technology and its security," *Blockchain: Research and Applications*, vol. 3, no. 2, Jun. 2022, doi: 10.1016/j.bcra.2022.100067.

[10] S. Nurgaida Yutia, R. Z. Fathiana, and Z. Fajriyah, "Blockchain-based Smart Contract for Decentralized Marketplace," *JOIV : International Journal on Informatics Visualization*, vol. 6, no. 2–2, p. 475, 2022, doi:10.30630/joiv.6.2-2.1121.

[11] F. Corradini, A. Marcelletti, A. Morichetta, and B. Re, "A Data Extraction Methodology for Ethereum Smart Contracts," in *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events, PerCom Workshops 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 524–529. doi:10.1109/PerComWorkshops59983.2024.10502604.

[12] H. S. Huang, T. S. Chang, and J. Y. Wu, "A secure file sharing system based on IPFS and blockchain," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jul. 2020, pp. 96–100. doi: 10.1145/3409934.3409948.

[13] F.-M. Costea, C.-B. Chirila, and V.-I. Creţu, "Auto-Generative Learning Objects for Learning Linked Lists Concepts," in *International Symposium on Electronics and Telecommunications (ISETC)*, 2020, pp. 1–4. doi: 10.1109/isetc50328.2020.9301136.

[14] R. Ravishankar, S. Kulkarni, S. Peri, and G. Sharma, "Distributing Context-Aware Shared Memory Data Structures: A Case Study on Singly-Linked Lists," Apr. 2024, [Online]. Available: http://arxiv.org/abs/2404.10151

[15] A. V Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*, vol. 175. Boston: Addison-wesley, 1983.

[16] P. A. Lobo and V. Sarasvathi, "Distributed File Storage Model using IPFS and Blockchain," in *2021 2nd Global Conference for Advancement in Technology, GCAT 2021*, Institute of Electrical and Electronics Engineers Inc., Oct. 2021. doi:10.1109/gcat52182.2021.9587537.

[17] Y. Jiang and J. Zhou, "Digital Music Copyright Protection System Based on Blockchain," in *2022 4th International Academic Exchange Conference on Science and Technology Innovation, IAECST 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 962–966. doi: 10.1109/IAECST57965.2022.10062214.

[18] T. Wang, S. Chang Liew, and S. Zhang, "PubChain: A decentralized open-access publication platform with participants incentivized by blockchain technology," in *2020 International Symposium on Networks, Computers and Communications, ISNCC 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020. doi:10.1109/ISNCC49221.2020.9297213.

[19] Q. Liping, "Design of Archives Management Information System Based on Blockchain Technology," in *Proceedings - 2022 2nd International Signal Processing, Communications and Engineering Management Conference, ISPCEM 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 66–72. doi:10.1109/ISPCEM57418.2022.00020.

[20] N. Nizamuddin, H. R. Hasan, and K. Salah, "IPFS-blockchain-based authenticity of online publications," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2018, pp. 199–212. doi: 10.1007/978-3-319-94478-4_14.

[21] M. I. S. Ayasy and A. M. Barmawi, "Protecting Author Royalty of Digital Assets Using Blockchain and Elliptic Curve Cryptography," Institute of Electrical and Electronics Engineers (IEEE), Jan. 2022, pp. 86–92. doi: 10.1109/gecost55694.2022.10010412.

[22] M. Rahrouh, W. Alayash, and M. Ghanem, "The Potential Application Of NFT in the Publishing Industry; Opportunities and Challenges," in *2022 International Arab Conference on Information Technology (ACIT)*, IEEE, Nov. 2022, pp. 1–5. doi:10.1109/acit57182.2022.9994159.

[23] K. R. Devi, "Analysis of Arraylist and Linked list," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 5, pp. 1566–1570, May 2019, doi: 10.26438/ijcse/v7i5.15661570.

[24] Q. Xiong, C. Wu, J. Xing, L. Wu, and H. Zhang, "A Linked-List Data Structure for Advance Reservation Admission Control *," 2005.

[25] M. Liu, H. Wang, and F. Yang, "An Efficient Data Query Method of Blockchain Based on Index," in *2021 7th International Conference on Computer and Communications, ICCC 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1539–1544. doi:10.1109/ICCC54389.2021.9674708.

[26] J. Anthal, S. Choudhary, and R. Shettiyar, "Decentralizing File Sharing: The Potential of Blockchain and IPFS," in *2023 International Conference on Advancement in Computation and Computer Technologies, InCACCT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 773–777. doi:10.1109/InCACCT57535.2023.10141817.

[27] S. Kumar Singh, V. Tiwari, and V. Rao Vadi, "Smart Contract Using Solidity (Remix-Ethereum IDE)," *International Journal of Advanced Research in Computer and Communication Engineering ISO*, vol. 3297, no. 2, 2007, doi: 10.17148/IJARCCE.2023.12253.

[28] M. M. Arer, P. M. Dhulavvagol, and S. G. Totad, "Efficient Big Data Storage and Retrieval in Distributed Architecture using Blockchain and IPFS," in *2022 IEEE 7th International conference for Convergence in Technology, I2CT 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi:10.1109/I2CT54291.2022.9824566.

[29] C. M. Nalayini, Jeevaakatiravan, P. V. Imogen, and J. M. Sahana, "A Study on Digital Signature in Blockchain Technology," in *Proceedings of the 3rd International Conference on Artificial Intelligence and Smart Energy, ICAIS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 398–403. doi:10.1109/ICAIS56108.2023.10073680.

[30] M. Fartitchou, H. El Marraki, L. Lafkir, A. Azzouz, K. El Makkaoui, and Z. El Allali, "Public-Key Cryptography behind Blockchain Security," in *Proceedings of the 5th International Conference on Networking, Information Systems and Security: Envisage Intelligent Systems in 5G/6G-Based Interconnected Digital Worlds, NISS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi:10.1109/NISS55057.2022.10085236.