scientific reports

Check for updates

OPEN

A lightweight scalable and dynamic blockchain-based model for storing and retrieving patient healthcare records

Seyed Rahman Soleimani Nadaf[1], Masood Niazi Torshiz[1]✉, Sayyed Majid Mazinani[2] & Seyed Reza Kamel Tabbakh[1]

Recent studies have explored blockchain technology for healthcare data storage and retrieval, addressing challenges related to scalability, trustworthiness, timely access, and patient privacy preservation. This paper presents a lightweight, scalable, and dynamic blockchain-based model, termed DHC, designed for efficient storage and retrieval of patient healthcare records. The model introduces several data structures aimed at enhancing the scalability of the blockchain network while ensuring data integrity and scalability in an off-chain environment. Patient healthcare data are stored daily in distinct proposed local dynamic blocks, which are subsequently added to a new local dynamic chain at the respective healthcare facility. These local dynamic chains periodically announce their final blocks to the blockchain network, where they are aggregated into a proposed global block. This block is verified through a two-layer, lightweight, energy-aware consensus algorithm (TLC). After validation, the global block is incorporated into the global chain. Additionally, the proposed access-control mechanism facilitates timely data access while safeguarding patient privacy. Evaluations of DHC under various scenarios demonstrate improvements in time complexity and storage efficiency, as well as higher retrieval rates for patient records. Furthermore, the TLC consensus protocol helps mitigate DDoS, Sybil, Eclipse, and fork attacks.

Keywords Blockchain, Storage and retrieval, Healthcare, Patient healthcare records, Access control, Scalability

Patient healthcare data is essential for various purposes, including diagnosis, treatment, and preventive care[1,2]. Consequently, the sharing of this data is critical for addressing the diverse needs of multiple stakeholders[3]. Traditional healthcare systems typically utilize centralized client-server architecture for storing and processing patient healthcare data. As a result, data stored at individual healthcare centers remains isolated, hindering efficient sharing with other facilities due to technical and infrastructure limitations. Thus, ensuring secure storage and timely access to patient healthcare data is paramount in healthcare systems. Several studies have used cloud computing platforms to facilitate the storage and sharing of patient healthcare data[4–6]. However, these approaches often encounter significant challenges regarding privacy, data integrity, and timely access to healthcare information[4,7,8]. Blockchain technology, recognized as a distributed system, offers a promising solution to the secure storage issues prevalent in traditional healthcare systems. In recent years, blockchain has been increasingly applied as a secure platform for data storage, effectively maintaining data integrity in a distributed environment[9–11]. By encapsulating patient healthcare data as transactions in blocks, blockchain enhances data integrity through hashing. Despite its advantages, blockchain faces scalability challenges exacerbated by the growing volume of patient healthcare data. To mitigate these issues, researchers have proposed various methods employing hashing and off-chain strategies[12–15]. In off-chain methods. patient healthcare data undergoes preprocessed before being stored on the blockchain[16]. Although off-chain methods increase on-chain scalability, off-chain scalability has not been addressed. Importantly, blockchain can maintain data integrity only for information stored on-chain, not off-chain. While off-chain strategies can enhance on-chain scalability, off-chain scalability remains unaddressed. Moreover, blockchain can maintain data integrity only for information stored on-chain, not for data stored off-chain. Furthermore, although some studies have discussed patient

[1]Department of Computer Engineering, Ma.C., Islamic Azad University, Mashhad, Iran. [2]Department of Computer Engineering, Imam Reza International University, Mashhad, Iran. ✉email: masood.niazi@iau.ac.ir

privacy preservation through smart contracts[17–19], only a few address the critical issue of timely data access[20,21]. Thus, there is a pressing need for complementary methods to address these challenges in healthcare systems.

In response, in this paper, a secure blockchain-based model is proposed for the integrated storage of patient healthcare records, designed to offer timely access to authorized users. Additionally, the model introduces lightweight structures to increase the scalability and trustworthiness of blockchain for healthcare data storage. The proposed model involves the following steps:

- A local dynamic block structure based on the Huffman tree is proposed for storing patient healthcare data. A local dynamic chain is introduced to manage these blocks in each hospital. Patient data is classified and stored in the local dynamic block structure across different wards, with updates occurring daily during the patient's hospitalization. Each newly created local dynamic block is subsequently added to the hospital's local dynamic chain. This design allows the latest local dynamic blocks to encapsulate all relevant healthcare data, enabling the local dynamic chain to remain lightweight by omitting redundant data.
- A global block structure is proposed, comprising the latest local dynamic block generated during a patient's hospitalization, which is then announced to the blockchain through the local dynamic chain. This global block not only consolidates all relevant healthcare data for the specified hospitalization period but also incorporates features that facilitate timely access. After verification through the proposed Raft-based consensus algorithm, the global block is added to a global chain.
- A mechanism is introduced to delineate user access control levels. This access control mechanism, combined with the global block structure and smart contract, enables the provision of necessary data to users while ensuring the preservation of patient privacy.

Thus, a Lightweight, Scalable, and Dynamic Blockchain-based model for Storing and Retrieving Patient Healthcare Records (DHC) is presented that ensures both lightweight scalability and data integrity during the off-chain preprocessing of healthcare data. Additionally, the model employs smart contracts and the global blockchain structure to safeguard patient privacy.

The remainder of this paper is structured as follows: The "Literature review" section examines previous works on data storage and retrieval in blockchain, the scalability of blockchain-based healthcare systems, and approaches for preserving healthcare data privacy. The "Proposed DHC model" section details the proposed structures including local dynamic block**s**, local dynamic chain, ready block**s**, ready queue, global block**s**, and global chain, along with a proposed consensus protocol and user access control mechanism that maintains patient privacy. The "Evaluation and results" section assesses the security and performance of the proposed model while the "Conclusion" section summarizes the findings and outlines directions for future research.

Literature review

Blockchain serves as a secure, distributed platform for the storage, retrieval, and sharing of data. Within this framework, data is organized as transactions in blocks, which are subsequently verified through consensus algorithms before being added to the blockchain. This process ensures data integrity through the use of hashes across distributed full nodes. In traditional blockchain systems, each piece of data is treated as a transaction and incorporated into a block. Consequently, applications that handle a high volume of transactions, such as healthcare, necessitate substantial storage capacity to accommodate the entire blockchain across full nodes. This requirement can adversely affect both the scalability and trustworthiness of the blockchain, as the number of full nodes capable of storing the complete chain decreases over time. Moreover, the transaction and block retrieval times in standard blockchains are characterized by a linear process, where the entire chain must be traversed sequentially to locate the desired transaction. Additionally, traditional blockchains lack effective mechanisms for preserving user privacy. This section examines these challenges and highlights the solutions that researchers have proposed to address them.

Data storage and retrieval in blockchain

As previously mentioned, traditional blockchain systems face significant challenges related to escalating transaction sizes and linear retrieval times. Therefore, numerous researchers have put forth solutions aimed at reducing both the required storage space and the time necessary for data retrieval[13,17,21–28]. as illustrated in Fig. 1.

As depicted in Fig. 1, a variety of strategies have been proposed to reduce the required storage space and data retrieval times in blockchain environments. These solutions can be categorized into three distinct approaches: (a) data indexing, (b) innovative data storage structures, and (c) hybrid methods for data storage and retrieval, discussed in detail below.

- Research on data indexing methods: This approach utilizes hashing techniques to decrease the required storage space of the blockchain. The hash function accepts input data of arbitrary size and generates a fixed-size output. The hash of the relevant data referred to as the data index, is stored as a transaction in a block, while the raw data is maintained off-chain. Consequently, data indexing effectively reduces the storage requirements of blockchain systems, as evidenced in various studies[17,24]. For instance, the model presented in[28] (DChain), replaces the entirety of all the raw data in a block, with a generated hash value. This hash is stored within the block, while the raw data resides in the InterPlanetary File System (IPFS). While this method considerably reduces the required storage space of the block, accessing the data necessitates interaction with IPFS.
- Research proposing new data storage structures: Although the data indexing method alleviates storage challenges, it encounters limitations regarding scalability as data volumes increase and lacks solutions for timely transaction access. Therefore, researchers have explored novel structures for blocks and chains[22,23,26,27,29].

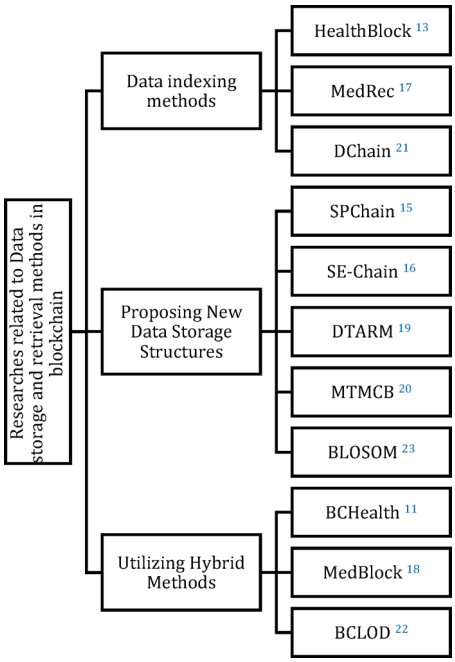

Fig. 1. Researches focused on data storage and retrieval in the area of blockchain.

These structures are tailored to the specific data applications intended for blockchain storage. For instance, block and microblock frameworks for healthcare data facilitate faster retrieval times[22]. In another method, both the username and the block name are preserved when data is stored enhancing retrieval efficiency[27]. A new structure called AB-M integrates binary and Merkle trees to improve data retrieval speeds[23]. Additionally, one methodology classifies data based on similarity and stores it within the blockchain, employing a random non-recursive binary search to significantly enhance user response times[26]. Other studies have developed blocks containing detailed lists of patient health records to expedite retrieval processes[29].

- Research utilizing hybrid methods: The advantages of data indexing and innovative storage structures include reduced storage requirements and improved data retrieval speeds. Researchers have sought to use these benefits through hybrid solutions that combine the two approaches[13,21,25]. For example, user data and access control policies are maintained across two separate chains of blocks, where the off-chain storage of access control policies diminishes the main chain's storage load containing user data[13]. Another approach involves segmenting data from various sectors into distinct blocks, classifying these blocks by sector, and creating sector-specific indexes to enhance retrieval durations[25]. In subsequent work[21], researchers clustered users to facilitate data storage, constructing a partial chain for each cluster that includes only blocks relevant to that cluster while integrating hashes of blocks from other chains to uphold data integrity. Furthermore, a full chain is established that encompasses hashes of blocks to reference those across diverse clusters. Given that most user queries are local; this method yields significant improvements in both retrieval times and storage efficiency within the blockchain.

Reducing the required storage space of blocks and the blockchain itself is essential for enhancing the scalability of blockchain systems, which will be discussed below.

Scalability of blockchain-based healthcare systems

Storing healthcare data directly on the blockchain leads to higher storage demands for both individual blocks and the overall blockchain. Increasing the length of the blockchain leads to fewer full nodes having the required storage space to store it. This growth can ultimately limit the number of full nodes capable of accommodating the required storage[30], thereby reducing the scalability of the blockchain[31]. Table 1 summarizes various studies conducted in the field of blockchain scalability for healthcare data storage.

As illustrated in Table 1, researchers have proposed a variety of methods aimed at addressing scalability challenges in blockchain-based healthcare systems[12–14,21,23,25,28]. Several studies[12–14] suggest storing only the index of original data on the blockchain instead of the data itself[12,14,28], while another study preserves only the most critical data on-chain[13]. These approaches maintain data integrity on-chain rather than relying on off-chain mechanisms. Additionally, methods have been proposed that aim to reduce computational resources by replicating older blocks across a smaller subset of full nodes[23] and by delineating the tasks assigned to different nodes[25], both of which may compromise the trustworthiness of the blockchain. Other researchers[21] construct separate chains of blocks within distinct geographic regions, increasing both scalability and trustworthiness by storing data from each region in full form, while storing data from other regions in a partial state.

No.	Research	Method	Advantages	Disadvantages
1	[14], 2017	An index is created on a web platform using a URL for each healthcare data entry, alongside access control permissions managed through smart contracts, which are stored on the blockchain.	Reduction of computational resources per node; increased scalability.	The growing volume of healthcare data is not adequately addressed, particularly regarding the integrity of data stored on the web.
2	[25], 2018	Task allocation among different nodes, where each node is responsible for a specific task.	Reduction of computational resources per node; increased scalability.	Although more nodes are participating in the network, they lack the computational resources needed to store the complete chain of blocks, which diminishes the overall trustworthiness of the blockchain.
3	[23], 2021	New blocks are replicated more frequently than old blocks across full nodes.	Reduction of computational resources per node; increased scalability.	This approach reduces the trustworthiness of the blockchain and increases retrieval time for older blocks.
4	[13], 2021	Utilizing two separate chains, with healthcare data stored on-chain and access control policies stored off-chain.	Reduction of computational resources per node; increased scalability.	The data integrity stored on the off-chain is not ensured.
5	[12], 2022	The data and its hash are initially stored off-chain, with only the hash placed on the blockchain.	Reduction of computational resources per node; increased scalability.	The integrity of off-chain data is not addressed.
6	[28], 2024	Raw data is stored on IPFS servers, with a single hash value created for all the data and included in the block.	Reduction of computational resources per node; increased scalability.	The integrity of off-chain data remains a concern.
7	[21],2025	User data is clustered, with each cluster containing only its users' data in its designated chain of blocks, while maintaining hashes of blocks created from other clusters.	Reduction of computational resources per node; increased scalability.	Global data retrieval times are not considered.

Table 1. Research on the scalability of blockchain-based healthcare systems.

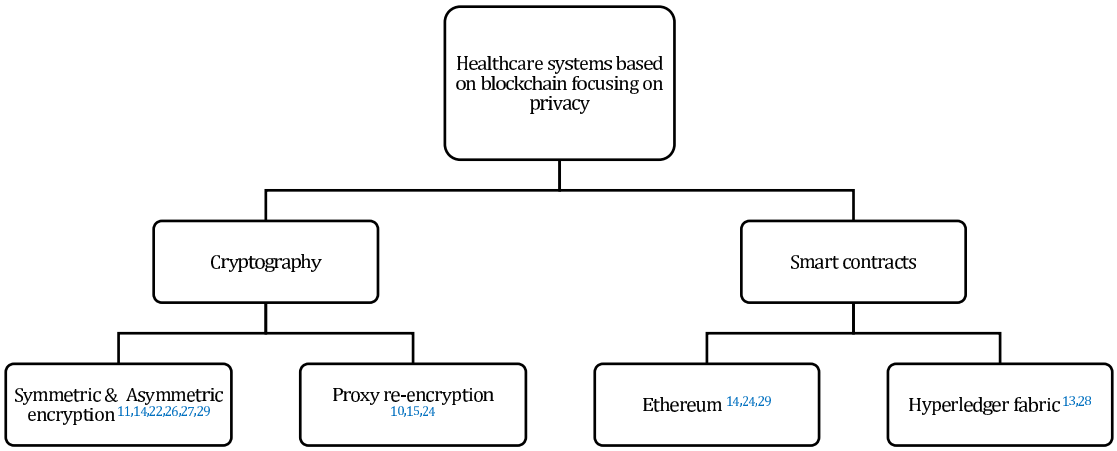

Fig. 2. Research focused on patient privacy preservation in blockchain-based healthcare systems.

Data privacy preservation in blockchain-based healthcare systems

Traditional blockchains do not adequately address the challenge of data privacy. Consequently, researchers have introduced several methods to safeguard healthcare data privacy[13,17,18,21,22,30,32–35]. as depicted in Fig. 2.

As shown in Fig. 2, several strategies utilize cryptographic techniques and smart contracts to protect patient privacy. For example, MediBchain[32] and BCHealth[13] encrypt patient healthcare data, ensuring that patients retain ownership of their information. Ancile[30] employs Ethereum smart contracts and proxy re-encryption to manage access to patient data, while SPChain[22] utilizes proxy re-encryption to uphold patient ownership of data. MedChain[18] integrates time-based Ethereum smart contracts for privacy preservation. Additionally, researchers from BCLOD[21] use smart contracts and asymmetric cryptography to establish access control for users and network nodes. The RBAC-HDE framework[35] implements data access control and privacy through Ethereum smart contracts and asymmetric cryptography. MedChain[18] also adopts the ElGamal distributed blinding re-encryption scheme[36] to manage transactions and access to electronic medical records. In MCPS[33], hospitals encrypt electronic health records (EHR) and their hashes using the hospital's public key storing them on the blockchain. Lastly, frameworks presented in[17,34] use smart contracts and Hyperledger Fabric private channels to maintain privacy, whereas BlockMedCare[12] employs proxy re-encryption to ensure data privacy.

Proposed DHC model

In this section, the DHC model is presented. This proposed model features innovative structures for blocks and chains that enhance the scalability, trustworthiness, and lightweight of the blockchain. These structures significantly reduce the storage requirements for both off-chain and on-chain data while ensuring the integrity of healthcare information across both chains. Furthermore, the model includes solutions to expedite healthcare data retrieval and uphold privacy through the use of smart contracts and proposed block structures.

The DHC model encompasses various components, including hospitals, wards, databases, local dynamic blocks, global blocks, ready blocks, local dynamic chains, a ready queue, a global chain, and smart contracts. Roles for patients and physicians are integral to the model, as depicted in Fig. 3.

As illustrated in Fig. 3, the proposed model consists of multiple hospitals where patients may visit different wards. To facilitate the organized and compact storage of patient healthcare data, a new Local Dynamic Block (LDB) structure based on a Huffman Tree (HT) is proposed. Each ward is assigned a specific weight in the HT to store its patients' data, resulting in the creation of a Wards Huffman Tree (WHT) (1). This WHT is established to store the healthcare data of each patient visiting the hospital (2), followed by the creation of the corresponding LDB (3). The WHT undergoes daily updates as the patient seeks care in various wards (4), thereby generating a new LDB (5).

Additionally, the model introduces Local Dynamic Chain (LDCh) and Global Chain (GCh) to reduce the required storage space for both off-chain and on-chain data. Each hospital maintains an LDCh that comprises the patients' LDBs generated daily (6). In the LDCh. the most recent LDB includes the updated WHT associated with a patient's healthcare data, in ensuring that each patient's most recent LDB contains a comprehensive record of their healthcare information during the relevant period. This design eliminates the need to store all LDBs while maintaining data integrity by retaining only the latest blocks for each patient.

Subsequently, each hospital's LDCh communicates the most recent patient LDB for a desired period to the blockchain network for integration into the proposed GCh (7). This block is transformed into a Ready Block (RB), which contains the hash of patient's WHT (HWHT) (8), while the raw data is stored on local IPFS servers within the corresponding cluster (9). The RB is placed in the Ready Queue (RQ) (10) and is verified through a two-layer consensus algorithm (TLC) based on Raft (11). Once validated, the block is incorporated into the GCh as a Global Block (GB) (12), which integrates the patient's health data.

Furthermore, an access control mechanism is proposed to preserve patient privacy using the smart contracts (13), the database, and the GB structure. This mechanism not only reduces the retrieval time for patient healthcare data but also delineates access permissions for users. Thus, the DHC model features the proposed structures of LDB, LDCh, RB, RQ, GB, and GCh, along with a novel access control mechanism for storing and retrieving patient healthcare data, which are elaborated upon below.

Proposed local dynamic block (LDB) and local dynamic chain (LDCh) structures

The traditional block structure, primarily based on Merkle trees, encounters scalability issues when tasked with managing the extensive volume of patient data. The DHC model presents a new LDB structure founded on the WHT to address these challenges by reducing storage requirements and enhancing data retrieval speed. Initially, healthcare data from different hospital wards is organized within the WHT, leading to the creation of an LDB. Subsequently, each day of the patient's hospitalization prompts an update to the existing WHT and the addition of a new LDB to the hospital's LDCh. Consequently, at any given time, the patient's most recent LDB encapsulates their complete healthcare data for that period. The subsequent subsections will elaborate on the proposed structures of WHT, LDB, and LDCh.

Proposed wards Huffman tree (WHT) structure

The WHT is derived from Huffman coding, a type of optimal prefix code utilized in information theory for data compression[37]. To construct the WHT using this coding approach, a Huffman Tree (HT) is first developed, where patient healthcare data is systematically organized based on the wards that patients visit. The process of creating the HT involves several steps. Initially, a set is formed containing multiple elements, each assigned a specific weight. The two elements with the lowest weight are selected, and their sum is added as a new element to the set. After this, the selected elements are removed from the set. This procedure is repeated by continuously selecting the two elements with the lowest weights from the remaining set and merging them into a new element until the root of the HT is established. An illustration of this process using three weights—W1 = 10. W2 = 20. and W3 = 15— is shown in Fig. 4[37].

As depicted in Fig. 4, the two elements with the lowest weights, W1 = 10, and W3 = 15, are selected to form Sub-HT1. Their combined weight, 25, is then added as a new element. Next, the element W2 = 20 is merged with Sub-HT1 to form the complete HT.

In this paper, the WHT is proposed as a modification of the HT to effectively store patient healthcare data across different wards. Upon a patient's visit to the hospital. the DHC constructs a WHT for the patient based on the weights assigned to various wards. Each day's patient healthcare data is systematically organized within the WHT according to the respective wards visited. The proposed WHT structure assigns greater weights to more important wards, positioning them at higher levels within the tree to facilitate faster data retrieval. The steps for constructing a patient-specific WHT are illustrated in Fig. 5.

As shown in Fig. 5, the process begins with each hospital determining the weights for its wards, which are then stored in a table. Addresses for each ward are also created using a public key to further optimize healthcare data retrieval times. The wards are subsequently sorted in ascending order based on their weights. The two wards with the lowest weights, W1 = 10 and W4 = 10, are selected and designated as the leaves of Sub-WHT1. Following this, the total weight of these two wards is combined with the weight of the third ward, W3 = 15, to form Sub-WHT2. In the final step, the weight of the second ward, W2 = 20, is merged with Sub-WHT2 to create the complete WHT. The root of the WHT, referred to as the WHT Root (WHR), contains the count of wards visited by the patient (Pi). Finally, the patient's healthcare data (Pi's data) is stored in the corresponding relevant wards within the WHT.

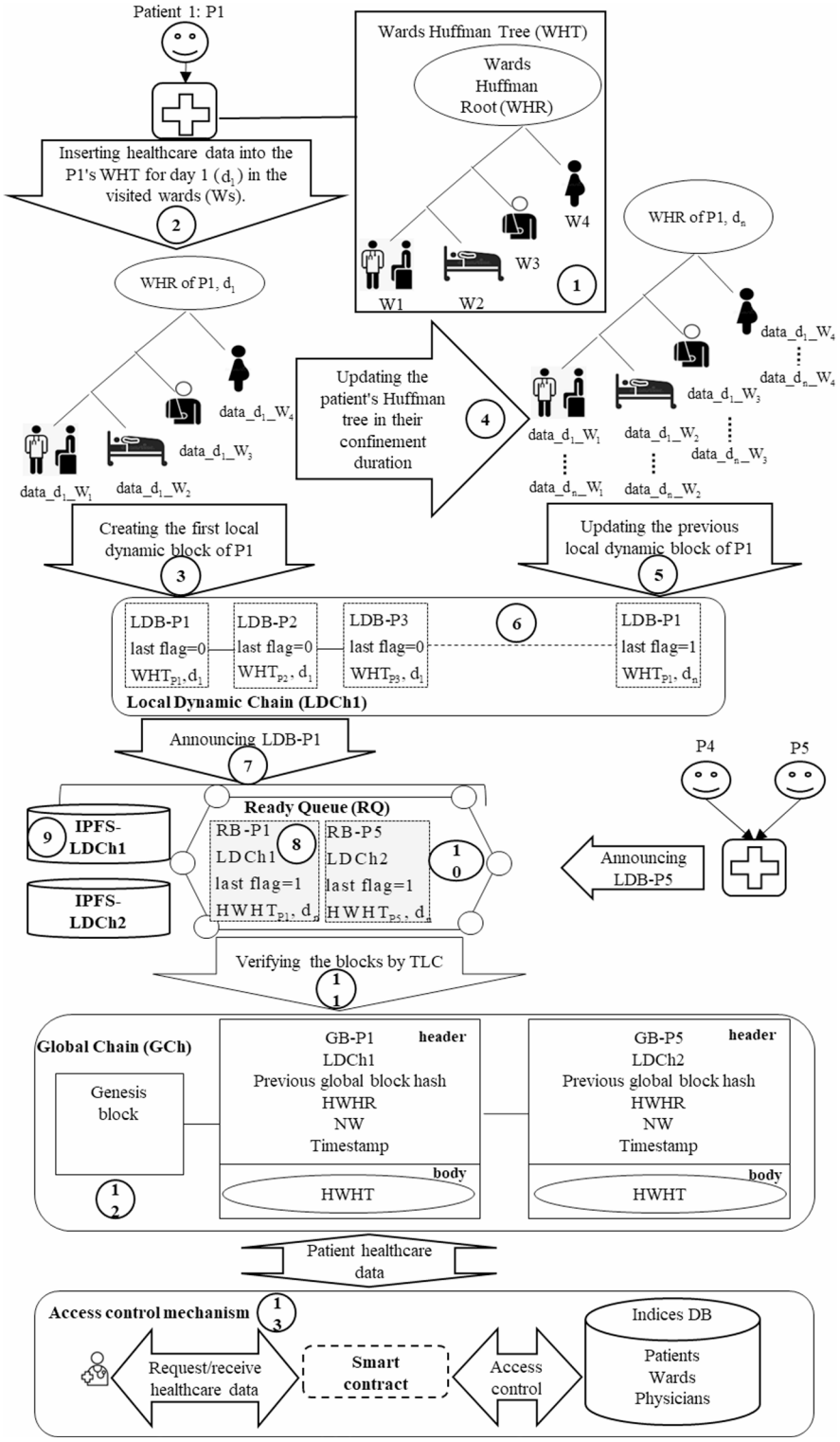

Fig. 3. Components of the proposed model and their interactions.

Proposed local dynamic block (LDB) structure

In the context of DHC, this paper introduces a novel LDB structure aimed at locally storing data for each hospital, thereby enhancing blockchain scalability. Each patient in the hospital is associated with an LDB based on the WHT, which is updated daily throughout their hospitalization. The most recent LDB for a patient encompasses all healthcare data pertinent to the specified hospitalization period. The proposed LDB structure consists of two components: a header and a body. The header includes the following elements: patient ID (Pi), the hash of the previous LDB, WHR, a timestamp, a last flag, and a threshold value. Here, Pi refers to the unique identification

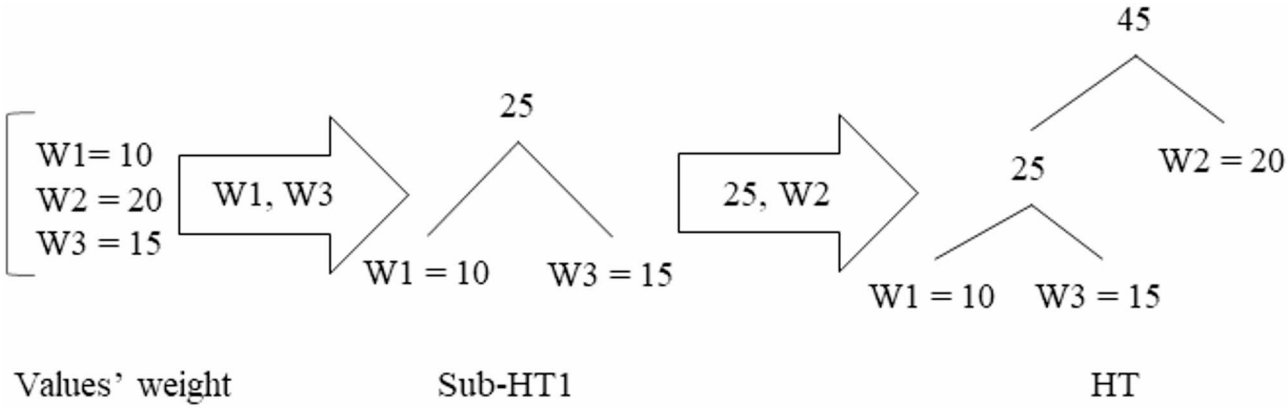

Fig. 4. Steps to build the HT.

Ward	Weight	Public key
W1	10	A
W2	20	B
W3	15	C
W4	10	D

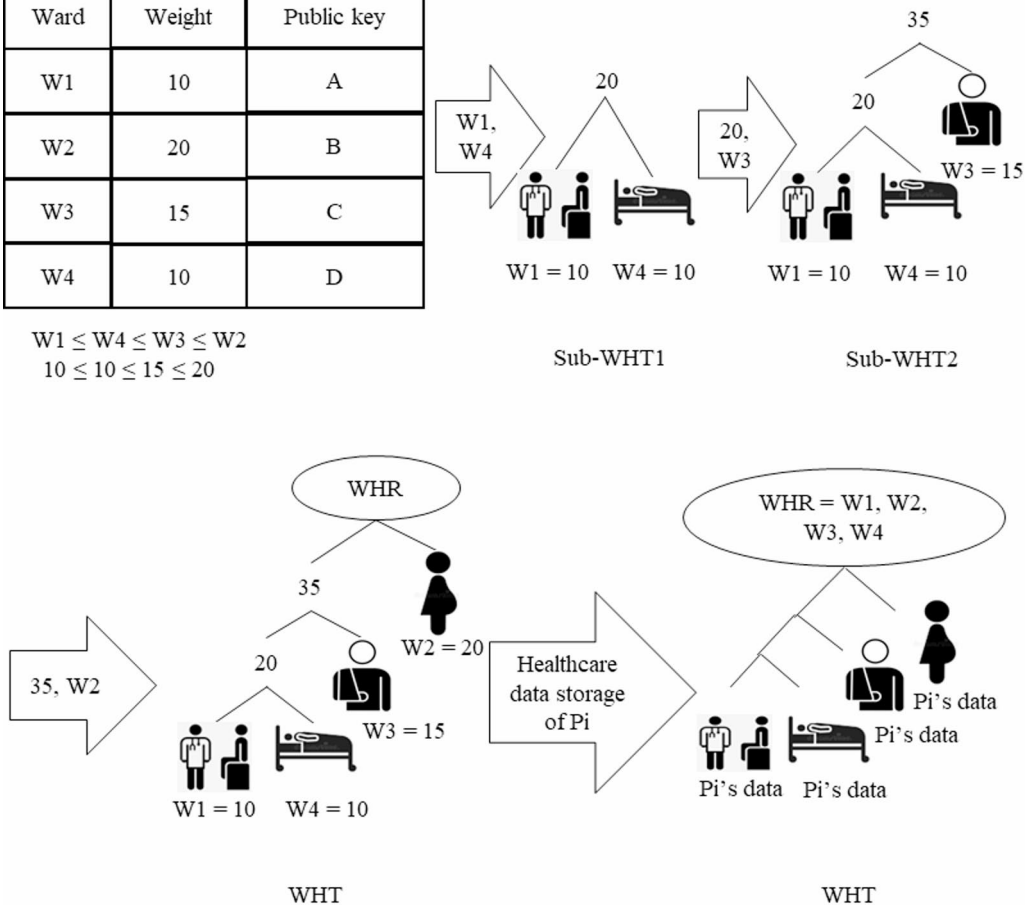

Fig. 5. Steps to build a patient's WHT.

of the patient, while the previous LDB hash contains the hash of the most recent LDB within the Local Dynamic Chain (LDCh), as further elaborated in the section titled "Proposed Local Dynamic Chain (LDCh) structure".

The last flag can take on values of zero and one, a value of one indicates that the patient has been discharged, while a value of zero signifies that the patient remains hospitalized. The WHR value, outlined in the section "Proposed Wards Huffman Tree (WHT) structure" encompasses the wards to which the patient has been assigned. To support scalability, a threshold value is defined; this value indicates the maximum number of LDBs that may be generated for an individual patient during a specified period. The threshold is determined according to the average number of hospitalizations for patients in a given geographical location and initially set to zero. the threshold increases by one for each LDB generated daily. Once the number of LDBs for a patient

reaches this threshold, the locally generated LDBs must be permanently stored. To ensure the permanent storage of a patient's LDBs for a particular period, the final LDB created during that timeframe is transmitted to the blockchain network. This procedure guarantees the long-term preservation of the patient's healthcare data, as this LDB contains the most up-to-date WHT.

Concurrently, the establishment of the threshold serves to regulate the scalability of the LDB by limiting the maximum number of levels within the WHT. As mentioned in the "Proposed Wards Huffman Tree (WHT) structure" section, the body of the LDB houses the WHT, which is updated daily throughout the designated period. Subsequently, when the last LDB for the ongoing period is dispatched to the blockchain for permanent storage, a new WHT is generated for the initial LDB in the next period, which will also undergo daily updates during the patient's hospitalization. Figure 6 illustrates the process of creating an LDB on the initial day of a patient's hospitalization.

As shown in Fig. 6, on the first day, a WHT is constructed for patient Pi according to the weights of wards (W1, W2, and W3) that they visit. The patient's healthcare data for that day (data_d1_Wi) was meticulously assigned to the corresponding leaves of the WHT. This initial WHT is integrated into the body of the LDB for patient Pi's first day, and the header includes patient ID, previous LDB hash, WHR, timestamp, last flag, and threshold value. A last flag value of zero signifies that the patient has not yet been discharged, and their subsequent LDB will be updated on the following day.

Update process of LDB: Over the course of the hospitalization period, the patient's healthcare data is consistently updated for each day of within the patient's WHT for each day, leading to the creation of a new LDB. Fig. 7 details the procedure for updating the LDB on the second day.

As depicted in Fig. 7, the WHT for the second day is revised based on the weights of the various wards (W1, W2, W3, and W4), the healthcare data for the second day (data_d2_Wz), and the WHT of the previous day. Following this, an LDB is generated for the second day of the patient's hospitalization. Importantly, on the second day, the last flag is set to one, indicating that this constitutes the final LDB for patient Pi's duration of hospitalization.

Proposed local dynamic chain (LDCh) structure

Most blockchain-based healthcare models utilize off-chain storage to reduce the storage requirements for the chain of blocks. While this approach alleviates some storage issues, it often compromises data integrity and faces scalability challenges regarding local storage capacity. To address these concerns, LDCh is proposed, which ensures data integrity while optimizing storage scalability. Each hospital will implement an LDCh to store the LDBs of the patients who have received care within its facilities. This chain temporarily retains each patient's LDBs throughout a designated period until the final LDB for that period is permanently recorded on the blockchain. The last LDB of a patient for any given period includes all their healthcare data from prior LDBs in that timeframe, which contributes to the chain's lightweight and scalable nature, allowing it to be stored locally on fog nodes. Figure 8 illustrates the LDChs of two hospitals, showcasing the patients' LDBs and the corresponding block announcements for permanent storage in the blockchain network.

As depicted in Fig. 8, the LDChs of two hospitals (LDCh1 and LDCh2) consist of LDBs associated with various patients Pi. Both LDCh1 and LDCh2 forward their blocks to the Requests Queue (RQ) for permanent storage in the form of Ready Blocks (RBs). An LDB for patient Pi (LDB-Pi) may transition to the RB state under two conditions: (a) the last flag is set to one, indicating discharge, or (b) the threshold has reached its maximum value. On the first day, an LDB-Pi for each patient (P1, P2, P3, P4, and P5) is created in both LDCh1 and LDCh2, with the last flags set to zero, meaning none are converted to RBs (1), (2). However, on the second day, an LDB-P1 is created in LDCh1 with the last flag set to one, indicating discharge. The LDCh1 is then included in

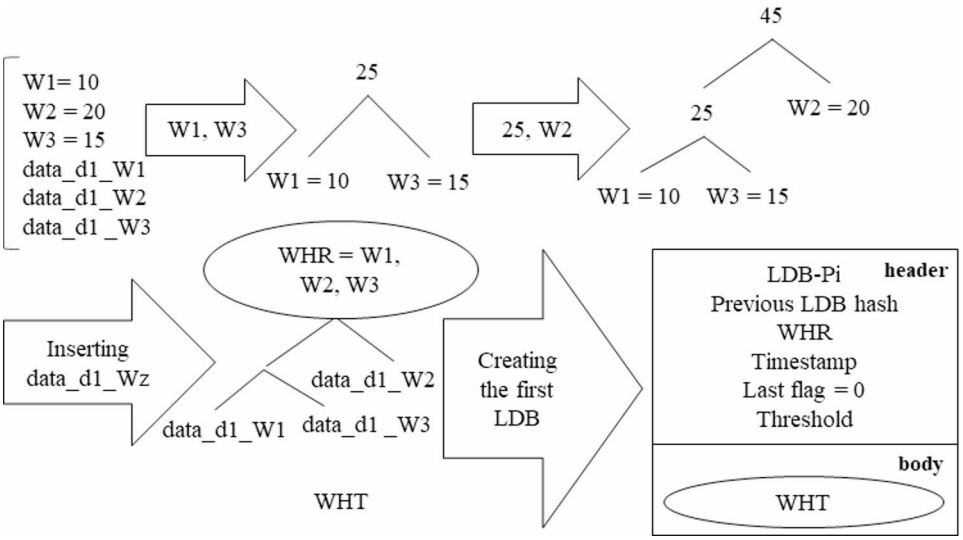

Fig. 6. Steps for creating the LDB on the first day of the patient's hospitalization.

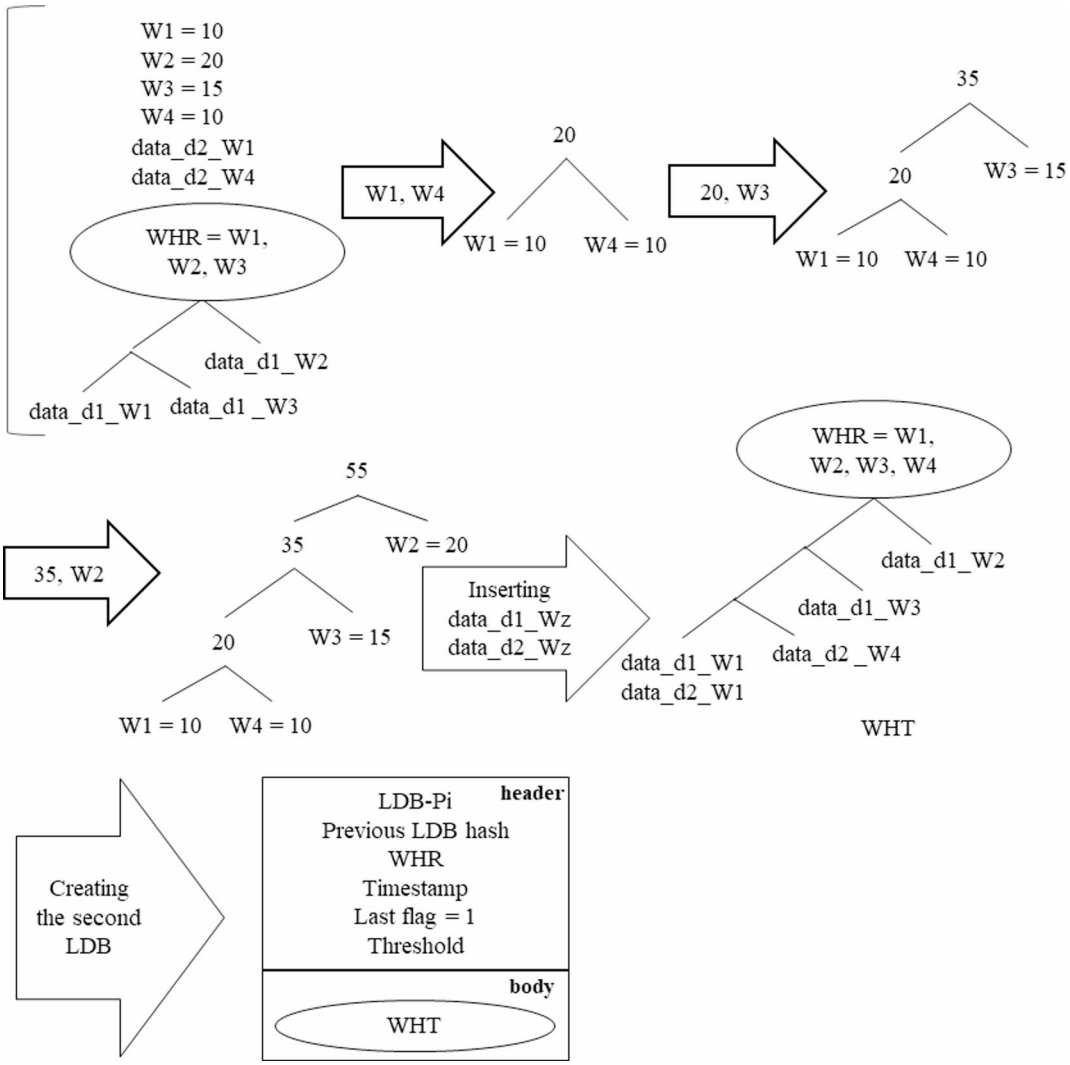

Fig. 7. Update of the patient's LDB on the second day of hospitalization.

its header and submitted to the RQ as the RB for patient P1 (RB-P1) (3). In contrast, on the third day, LDB-P4 in LDCh2 reaches the maximum threshold value, prompting LDCh2 to announce it as RB for patient P4 (RB-P4) (4). The RQ accepts the requests from LDCh1 and LDCh2 and places them in a First-In-First-Out (FIFO) queue for validation (5). The approved blocks are subsequently integrated into the Global Chain (GCh) as Global Blocks (GBs) as detailed in the section on "Proposed Global Block (GB) and Global Chain (GCh) structures".

Once the LDB-P4 on the third day has been converted to the RB-P4 due to exceeding the maximum threshold, additional LDBs will still be generated for patient P4 in LDCh2. For the fourth day, only the healthcare data from that day—marking the beginning of the next period—will be utilized to create the WHT. The WHT for the fourth day will not encompass data from previous days, as this data was previously with the initial RB-P4 for permanent blockchain storage. The new WHT will be updated until a second RB-P4 is announced to the RQ, enabling the formation of subsequent LDBs for patient P4 in the following days of the second period. The WHT is the fourth day not the previous days, this data was sent to RB-P4 for permanent storage. The new WHT is updated until a second RB-P4 is announced to the RQ. After the second RB-P4 is announced, a new WHT will be developed for the creation of LDB-P4 in the third period. This iterative process will continue until the final RB-P4 is submitted to the RQ. This method preserves the scalability of the LDB and LDCh by generating new WHTs that exclude previously stored healthcare data.

- Creating ready blocks (RBs): As previously discussed in the "Proposed Local Dynamic Chain (LDCh) structure" section, during each specified period, the last LDB of a patient is converted into RB status for permanent storage on the blockchain. The LDBs created for a patient encapsulate comprehensive healthcare data from the relevant period, requiring substantial storage capacity. Hence, to maintain the RB scalability, a proposed RB block structure based on LDB is introduced, illustrated in Fig. 9.

In Fig. 9, a hash value is generated for each ward's healthcare data (1), which is then included in the WHT as the ward transaction (2). The new HWHT is incorporated into the body of the RB body (3), along with values for

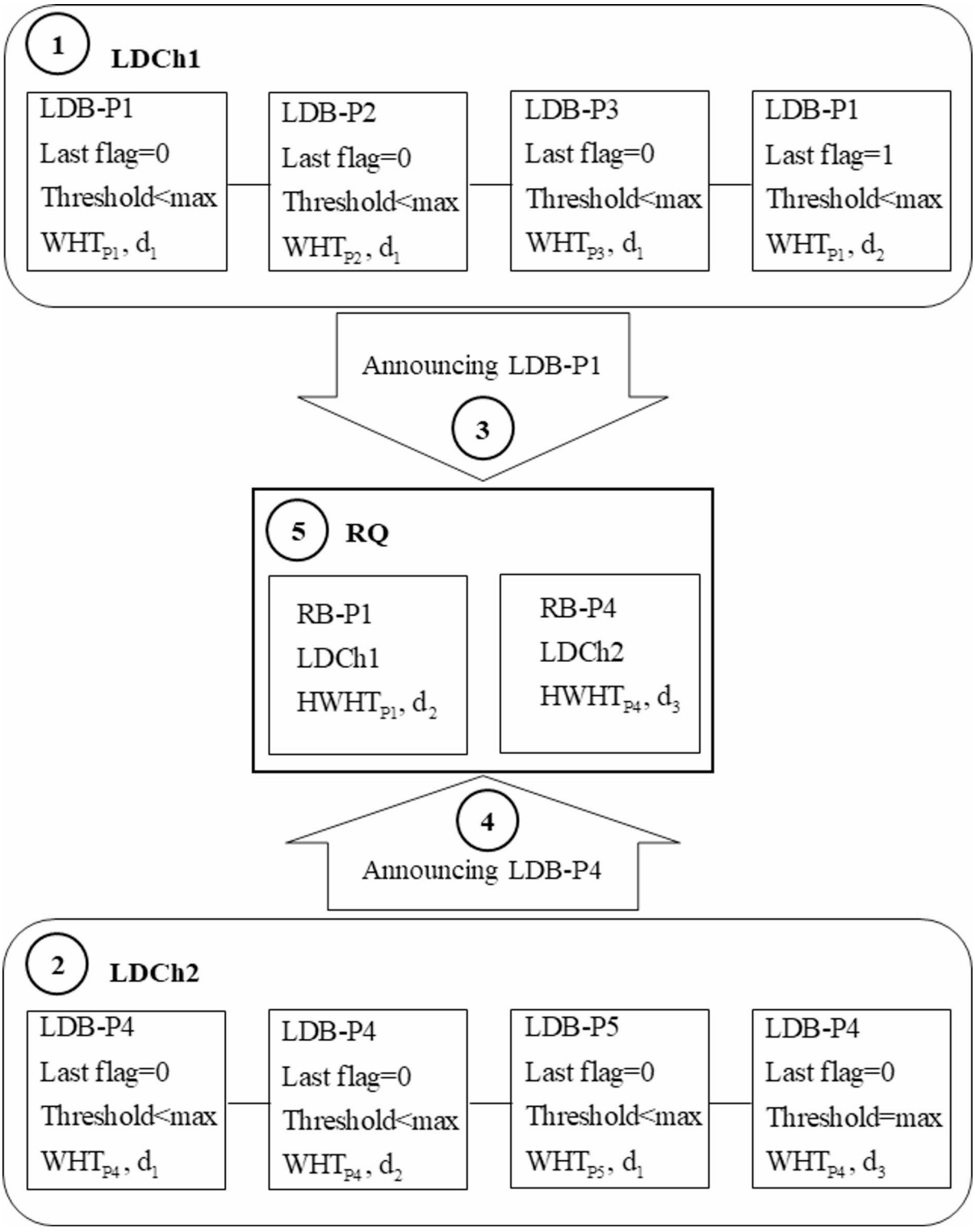

Fig. 8. Proposed LDCh structure and the block announcement for permanent storage on the blockchain.

RB-Pi, LDChj, the HWHT's root (HWHR), the number of wards visited by the patient (NW) and timestamp in the RB header (4). Here, the LDChj indicates the LDCh number from which the block was created. Additionally, IPFS servers are employed by each hospital to store the ward's healthcare data (5).

- LDCh trimming: To manage storage demands, many researchers resort to off-chain solutions to alleviate the on-chain storage burden of healthcare data. However, such local off-chain storage often encounters scalability issues. In this paper, the LDCh is proposed as an off-chain solution, implementing a trimming process to

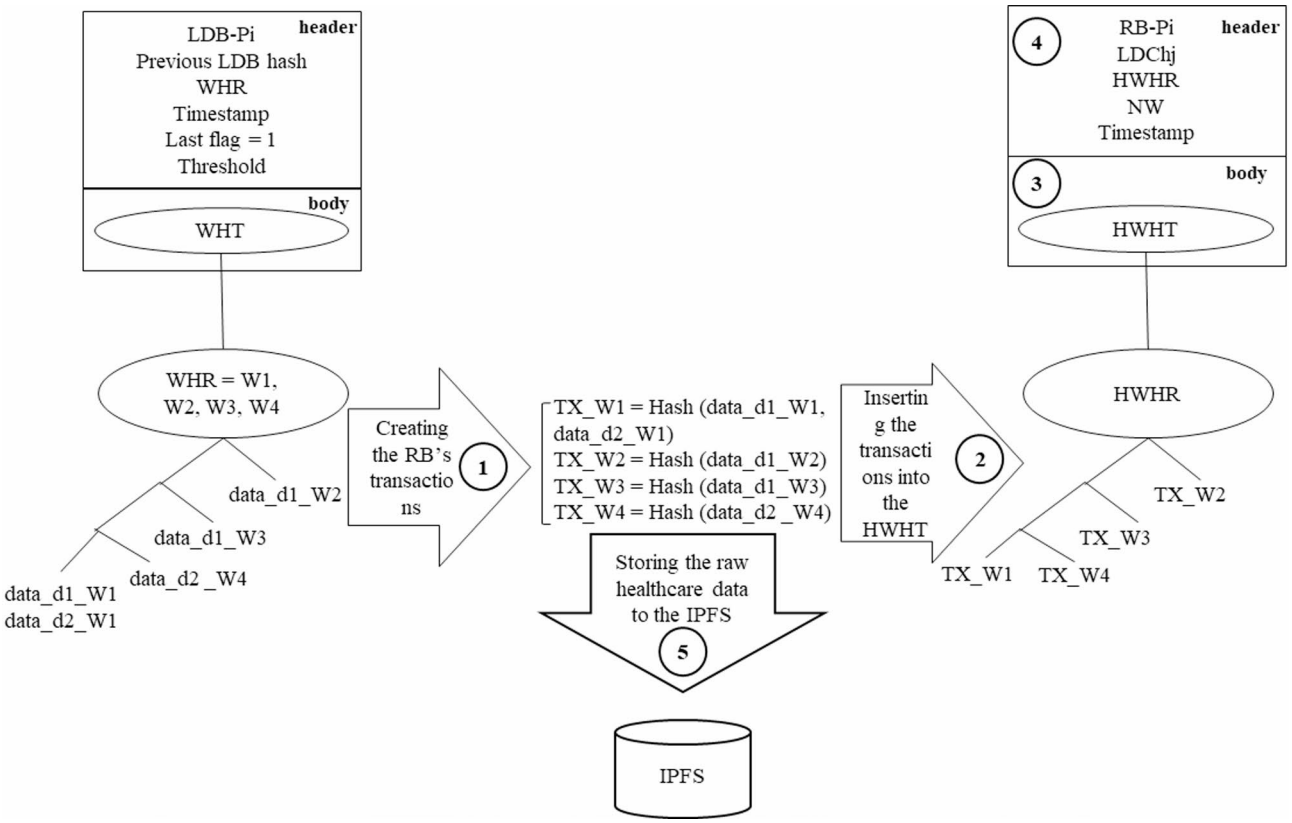

Fig. 9. Proposed RB structure and storage of raw healthcare data in IPFS.

remove outdated blocks from the on-chain storage in each time frame to bolster scalability. The LDCh trimming process unfolds as follows:

a) Patient LDBs are identified as Candidate Blocks (CBs) if their corresponding RBs have been submitted to the RQ and integrated into the blockchain as delineated in the section on "Proposed Global Block (GB) and Global Chain (GCh) structures". All blocks not classified as candidates are labeled Non-Candidate Blocks (NCBs).
b) The LDCh scans from the left until the first NCB is identified. The six blocks preceding the first NCB, along with all subsequent blocks—regardless of whether they are CBs or NCBs—remain in the chain, while the other CBs are purged from the LDCh. This trimming procedure ensures block integrity, borrowing principles from Bitcoin, which maintains the integrity based on its preceding six blocks[38]. Therefore, in LDCh trimming, the six blocks before the first NCB are preserved in the chain. The LDCh trimming process is illustrated in Fig. 10.

As demonstrated in Fig. 10, during the timeframe T1, the LDCh generates fifteen blocks for patients P1, P2, P3, P4, P5, P6, and P7. LDB-P1, LDB-P2, LDB-P3, and LDB-P4 are formulated on the second day, while LDB-P5 is created on the fourth day, all transitioning to RB-P1, RB-P2, RB-P3, RB-P4, and RB-P5 for submission to the RQ for inclusion in the GCh.

These blocks and their associated LDBs created for these patients in previous days are classified as CBs. Conversely, LDB-P6 and LDB-P7 have not yet produced any blocks in a ready state, categorizing them as NCBs. Upon scanning from the left side of DLCh, LDB-P6 emerges as the first NCB (FNCB). Consequently, the six preceding blocks, even if they are CBs, cannot be deleted and will remain within the LDCh. Additionally, LDB-P5 and LDB-P7, which follow FNCB, are retained in the LDCh, regardless of their status as CBs or NCBs. By eliminating other CBs, the length of the LDCh is effectively reduced from fifteen to ten. Therefore, the trimming of the LDCh in each timeframe promotes off-chain scalability, contrasting with other methods proposed in existing research that struggle with adequate storage space.

Proposed global block (GB) and global chain (GCh) structures

In blockchain-based healthcare models, researchers typically utilize on-chain storage for the permanent retention of blocks. This on-chain framework is often complemented by off-chain solutions to enhance the scalability and manage the storage demands of healthcare data. Consequently, simultaneous access to both on-chain and off-

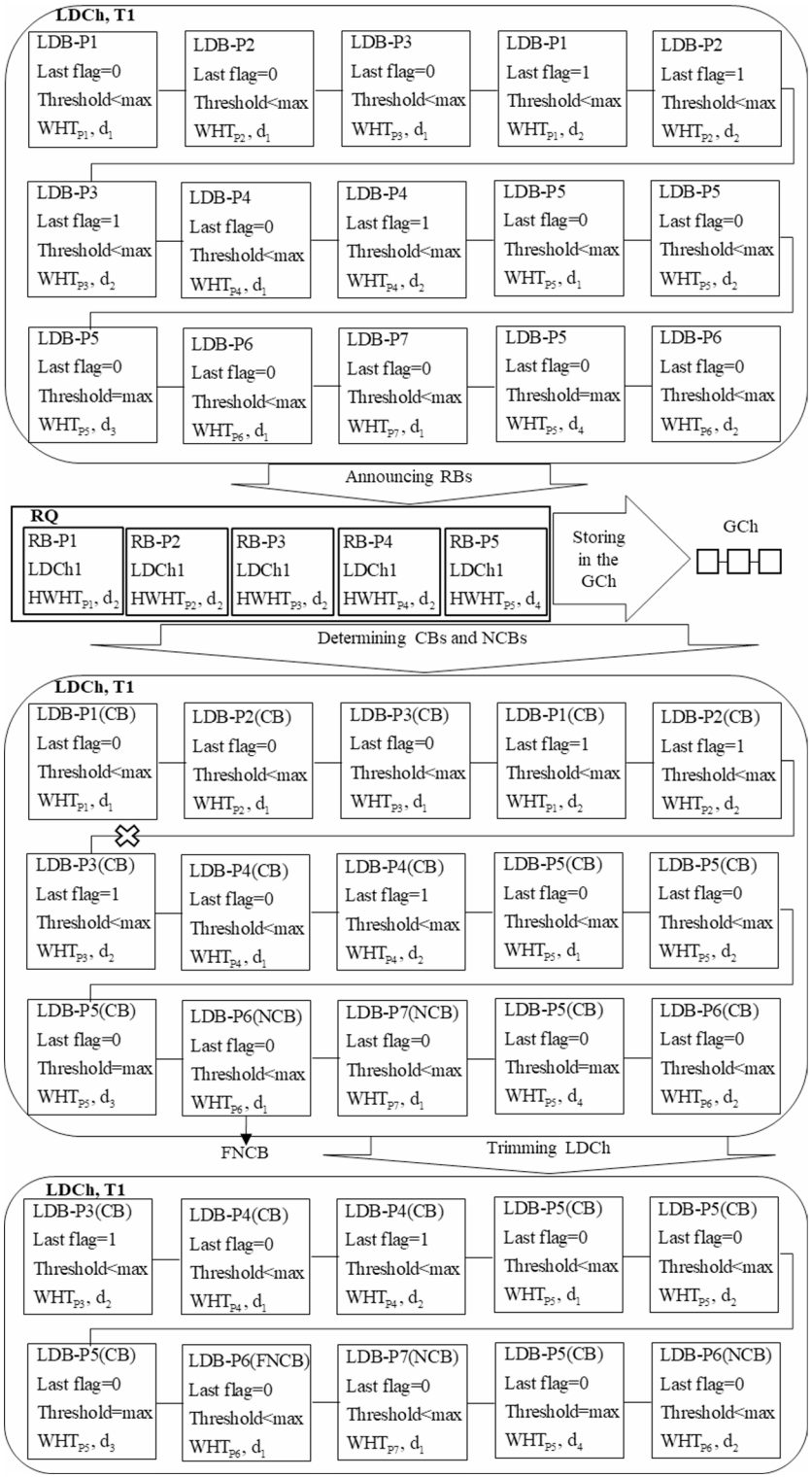

Fig. 10. Presentation of the LDCh trimming process.

chain data is necessary for effective data retrieval. This paper proposes a Global Chain (GCh) as an on-chain solution that stores global blocks (GBs) containing comprehensive patient healthcare data over specific time periods. Although the GCh uses off-chain storage for healthcare data, retrieval is solely accomplished through access to the GCh itself, thereby reducing data retrieval times. The architectural framework of the proposed GB and GCh is illustrated in Fig. 11.

As depicted in Fig. 11, once the LDB-Pi is transformed into an RB-Pi, it is transmitted to the blockchain network for verification. Upon successful verification, the RB-Pi is incorporated into the GCh as a GB

corresponding to the patient Pi (GB-Pi). The GB structure comprises two components: a header and a body. The header includes the patient identification number (Pi), the LDCh number of the hospital where the RB-Pi was created (LDChj), the previous GB hash, the HWHT root (HWHR), the number of wards visited by the patient (NW), the discharge timestamp, and Count, which is an ordered pair (block num, status) that links all GBs related to the patient during their hospital stay.

Utilizing the Count attribute allows for efficient access to each GB associated with a patient in an O (1) time complexity, through recursive referencing using the block num of the most recent GB-Pi. The block num

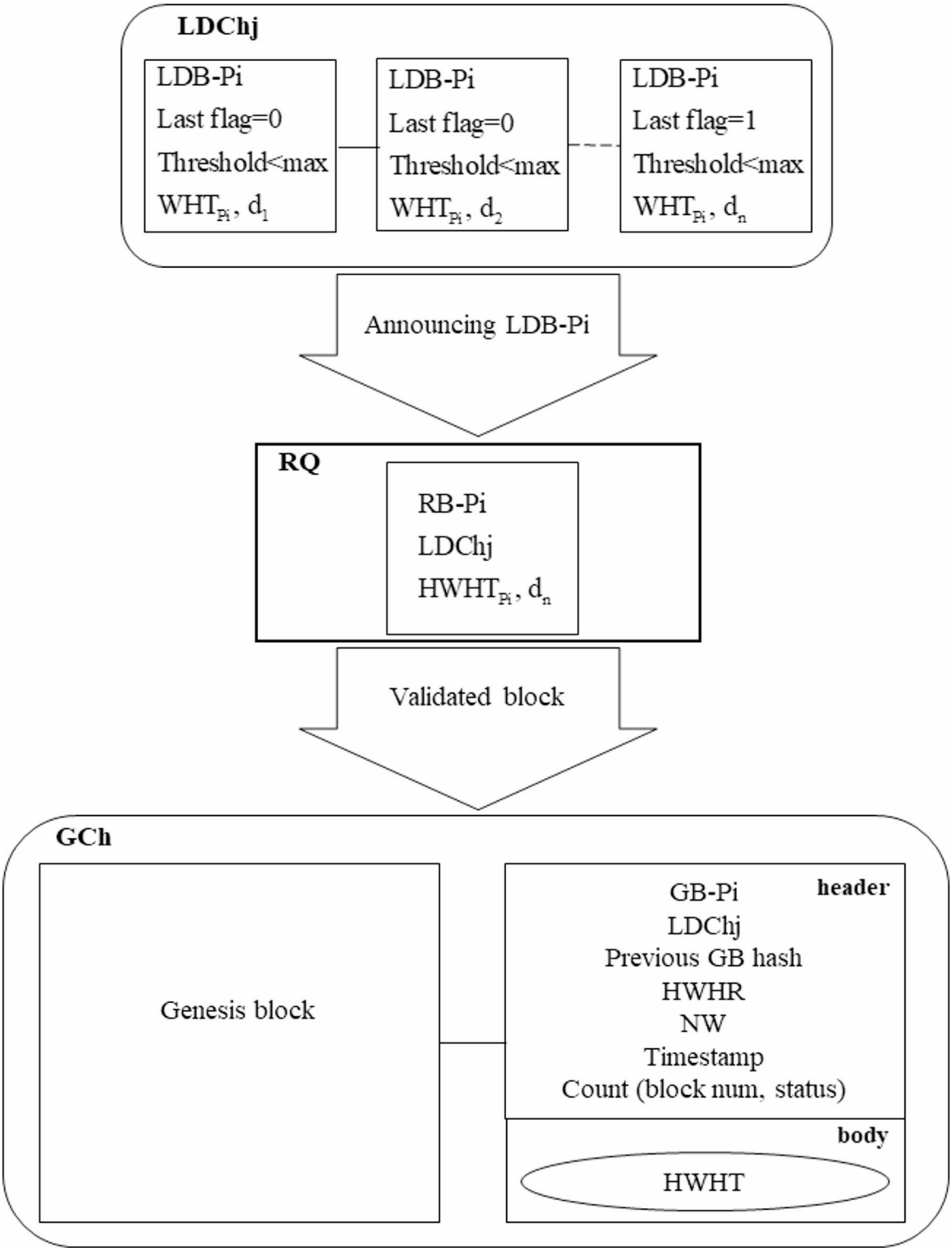

Fig. 11. The proposed GB and GCh structures.

reflects the sequence of the last GB-Pi added to the GCh. If the status value in the last GB-Pi equals one, it indicates the conclusion of the Pi's hospitalization, allowing for full access to their healthcare data from that period. Conversely, if the status value indicates that the patient's treatment is ongoing; their healthcare data will be appended to the GCh upon completion. Detailed procedures for retrieving healthcare data from the GCh using the Count attribute will be outlined in the "Proposed access control mechanism" section.

The body of the GB comprises the latest updated hash values of the WHT leaves (HWHT) for the patient during the specified period. The GBs from various hospitals are incorporated into the proposed GCh, which is stored across distributed nodes. The process of creating a patient transaction and its subsequent inclusion into the GCh is illustrated in Fig. 12.

Proposed two-layer consensus (TLC) protocol

This paper presents a two-layer consensus protocol designed to increase throughput, reduce execution time, and reduce energy consumption. The proposed consensus protocol verified blocks through two distinct layers, local and global. In each hospital, LDBs are verified locally, while GBs are verified globally across all hospitals.

During each consensus round in the local layer, each hospital identifies several trustworthy fog nodes that have recently been authenticated by the hospital's internal network to serve as participating nodes in the consensus process. These trustworthy nodes then vote to elect a leader using the Raft protocol[39] and verify the generated LDBs locally. Once an LDB is transformed into an RB, the global consensus layer is activated to validate the RB. In this layer, the leader nodes from the local consensus layer are recognized as trustworthy nodes and introduced to the blockchain network. These nodes engage in a voting process within the global consensus layer, also utilizing the Raft protocol to select a global leader, followed by the verification of the RB. The validated RB is subsequently added to the GCh as a GB.

The adoption of a two-layer consensus protocol enables local verification of LDBs by nodes within each hospital, which increases the throughput and reduces the execution time. Additionally, each hospital incorporates several trustworthy nodes from its private network to participate in the consensus, thereby increasing the overall trustworthiness of the model. Furthermore, since LDBs are verified locally and independently in each hospital—with only a limited number submitted for global verification—both energy consumption and execution time are reduced. The pseudocode for the two-layer consensus model is illustrated in Fig. 13.

Proposed access control mechanism

Ensuring patient privacy is a critical challenge in blockchain-based healthcare data storage and retrieval models, a concern that traditional blockchains fail to sufficiently address. Additionally, the data retrieval time in traditional blockchains is linear, exhibiting O(n) complexity, which negatively impacts efficiency in healthcare contexts. Many blockchain-based models employ complementary solutions, such as smart contracts (SCs), to protect user privacy and reduce data retrieval times. Consequently, this paper presents an access control mechanism aimed at safeguarding patient privacy while enhancing the efficiency of healthcare data retrieval. The proposed mechanism utilizes the GB structure, SCs, an indices database (IDB), and local IPFS servers to facilitate timely access to patient healthcare data while preserving their privacy. The steps involved are detailed as follows:

```
Input: healthcare records, patients, hospitals, RQ, GCh.
Create a LDChj for each hospital:
    Measure patients' healthcare records:
        Create a WHT for each patient:
            Update WHT by the patient's healthcare records for each day.
            Create a LDB:
                Insert the WHT to the LDB's body.
                Insert patient ID, previous LDB hash, WHR, timestamp, last flag, and threshold to the LDB's header.
                Verify the LDB.
                Add the validated LDB to the LDChj.
            If last flag = 1 or threshold = max:
                Convert the LDB to a RB:
                    Insert associated LDChj to its header
                Send the RB to the RQ.
                Verify the RB.
                Create a GB:
                    Insert Pi, LDChj, previous GB hash, HWHR, NW, Timestamp, Count (block num, status) to the header.
                    Insert HWHT to the body.
                    Verify the GB.
                    Add the validated GB to GCh.
            If threshold = max:
                Repeat 4-20 steps.
            If time frame = T:
                Trim the LDChj:
                    Determine CBs and NCBs.
                    Determine FNCB.
                    Preserve 6 blocks previous of FNCB and all blocks after that.
                    Drop other CBs.
Outputs: trimmed LDChs, GBs, GCh.
```

Fig. 12. Pseudocode for creating LDB, LDCh, GB, and GCh.

```
Inputs: LDBs, LDChs, the GCh, network nodes, hospitals.
Network nodes are authenticated in each hospital.
In each consensus round:
    Hospitals select some their authenticated nodes as trustworthy nodes.
    The trustworthy nodes in each hospital participate in the first layer of consensus: // In this step, the first layer of consensus is applied.
        The trustworthy nodes select their leader using Raft and other nodes remain as follower nodes.
        The follower nodes verify a LDB using Raft and send their response to their leader.
        The leader node announces the validated LDB to the network and adds it to the associated LDCh.
    If the LDB is an RB: // In this step, the second layer of consensus is applied.
        The RB is converted to a GB.
        The leader nodes in all hospitals are selected as participating consensus nodes in the second layer:
            The nodes using Raft select the super leader and other nodes remain as super follower nodes.
            The super follower nodes verify the GB using Raft and send their response to the super leader node.
            The super leader node announces the validated GB to the network and adds it to the GCh.
Outputs: the validated LDBs, the validated GBs, LDChs, the GCh.
```

Fig. 13. Pseudocode for the two-layer consensus protocol.

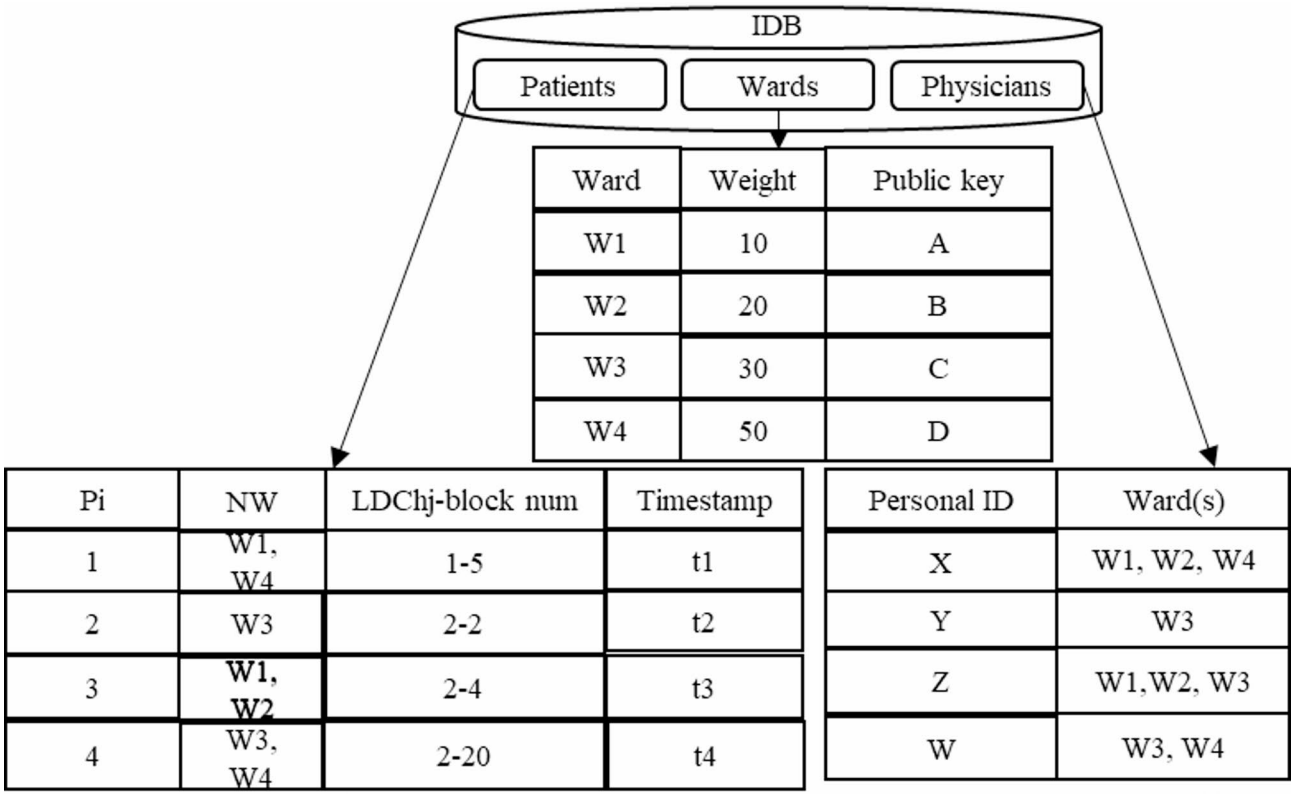

Fig. 14. The IDB structure.

First step: creating a indices database

The first step in the proposed access control mechanism involves establishing an IDB to organize healthcare data. The IDB includes tables of patients, physicians, and wards, as displayed in Fig. 14.

As illustrated in Fig. 14, the IDB comprises three tables for storing data related to wards, patients, and physicians, described below:

- Patients table: This table records the hospitalization history of patients, designated by the identifier (Pi), with the NW indicating the wards in which each patient received treatment. Given that a patient may be hospitalized in multiple facilities, leading to the creation of several GBs in the GCh during each hospitalization period, the table also includes Timestamp and LDChj-block num attributes. These features facilitate more efficient healthcare data time retrieval. The LDChj-block num denotes the most recent GB allocated to the patient during their hospitalization at each hospital, ensuring that as new GBs are created, the patients table is updated accordingly. By utilizing the LDChj-block num in the patients table and the Count (block num, status) in the proposed GB structure, patient healthcare data can be retrieved in O(1) time, compared to O(n) retrieval time in traditional blockchains.

- Wards table: Initially, a private-public key pair is generated for each ward by the SC. Subsequently, a weighting is assigned to each ward based on its priority in the treatment process; this information, along with the ward's public key, is recorded in the ward's table.
- Physicians table: This table outlines the access control level of physicians concerning the wards. It includes identifiers for physicians, represented by personal IDs, along with the number of wards to which they have access.

Second step: determining access control level of physicians

- Following the creation of the IDB, the proposed access control mechanism establishes the access control level of physicians to healthcare data to protect patient privacy through the following procedure: Physicians are first authenticated by the SC. Each physician then generates a signature script using the public keys of the wards they are authorized to access, as outlined in Eq. (1).

$$sign(hash(pubkey_{W1}), hash(pubkey_{W2}), ..., hash(pubkey_{Wn})) \quad (1)$$

Physicians insert the hashed public key of the accessible wards into Eq. (1) and sign the script with their private key. They then submit this signed script to the SC, requesting access to the healthcare data of patient Pi.

- The SC references the patients table to obtain the Timestamp and LDChj-block num values associated with the GB of patient Pi and retrieve the corresponding blocks from the GCh.
- The desired healthcare data from the relevant ward is then fetched by the SC using the HWHT and transmitted to the physicians.

The process for physicians to access patient healthcare data is outlined in Fig. 15.

In Fig. 15, to access the healthcare data of patient P1, a physician with personal ID = Z follows these steps: (1) First, they create a digital signature according to their authorized wards; (2) they send the signature to the SC; (3) the SC retrieves the GB(s) linked to patient P_1 using the LDCh_j-block num and Timestamp in the patients table; (4) the GB body contains a HWHT for rapid access to the relevant ward data; (5) the SC retrieves the requested data from the local IPFS according to the LDCh_j value. This mechanism ensures patients privacy by enforcing access control levels of physicians and only references the GCh to fetch healthcare data, thereby reducing retrieval times. The pseudocode of the proposed access control mechanism is depicted in Fig. 16.

Retrieving patient healthcare data from the GCh

As described in the section on "Proposed Global Block (GB) and Global Chain (GCh) structures" in a DHC system. although off-chain storage is utilized, data retrieval is executed solely through the on-chain GCh. This approach contrasts with other blockchain-based data storage and retrieval models that use both off-chain and on-chain mechanisms for data retrieval. The process of healthcare data retrieval from the GCh is illustrated in Fig. 17.

To retrieve healthcare data, it is necessary for the SC to access the patient's GB(s) and then present the required data to users according to the proposed access control mechanism. The patients table stores the latest GB(s) for each patient during their hospitalization period, identified by the LDChj-block num. As shown in Fig. 17, the SC first refers to the patients table to obtain the LDChj-block nums associated with patients P1, P2, and P3, subsequently retrieving their healthcare data as follows:

- Retrieving GB(s) for patient P1: In the patients table, the LDChj-block num for P1 is noted as 1–5, indicating the patient was hospitalized at a facility with LDChj is 1 and the last GB recorded for this hospitalization period in the GCh is block num 5. The SC first accesses block num 5 in the GCh with an O (1) time complexity. From this block, it retrieves the Count value (3,1), which indicates two things: The status value of 1 signifies that block num 5 is the final block for P1 in this hospitalization period, meaning no further blocks will be added as the patient has been discharged. There exists at least one additional block for this patient in the GCh, specifically block num 3. The SC subsequently retrieves block num 3 with O(1) time complexity, which yields a Count value (1, 0) indicating that block num 1 is another record for P_1. The SC also accesses block num 1 in O(1) time, obtaining a Count value (0, 0). Since the block num equals 0, this indicates it is the first block of P1 recorded in the GCh for the specified hospitalization period. Therefore, the search for GBs of patient P1 concludes, retrieving all relevant blocks with O(1) efficiency.
- Retrieving GB(s) of patient P2: The SC retrieves the LDChj-block num for P2, which is 2–2. This indicates that the block(s) of the patient was introduced to the GCh by the hospital with LDChj = 2. The SC retrieves this block in O(1) time, yielding a Count value (0,1), suggesting that this is the final block for P2 during their hospitalization since the status is 1. Block num "0" indicates that there are no additional blocks for this patient during the hospitalization period, which concludes the retrieval process.
- Retrieving GB(s) of patient P3: The LDChj-block num for P3 is recorded as 2–4, which the SC retrieves from the GCh in O(1) time. The resulting Count value (0,0) indicates that P3 has not yet been discharged, and further blocks are anticipated to be added to the GCh in the future. The block num being 0 signifies that this is the initial block for P3 in this particular hospitalization period, effectively concluding the search, as there are currently no other blocks for P3 in the GCh.

After successfully retrieving the relevant GB(s) for each patient, the SC proceeds to access the necessary ward data using the physicians' signature script.

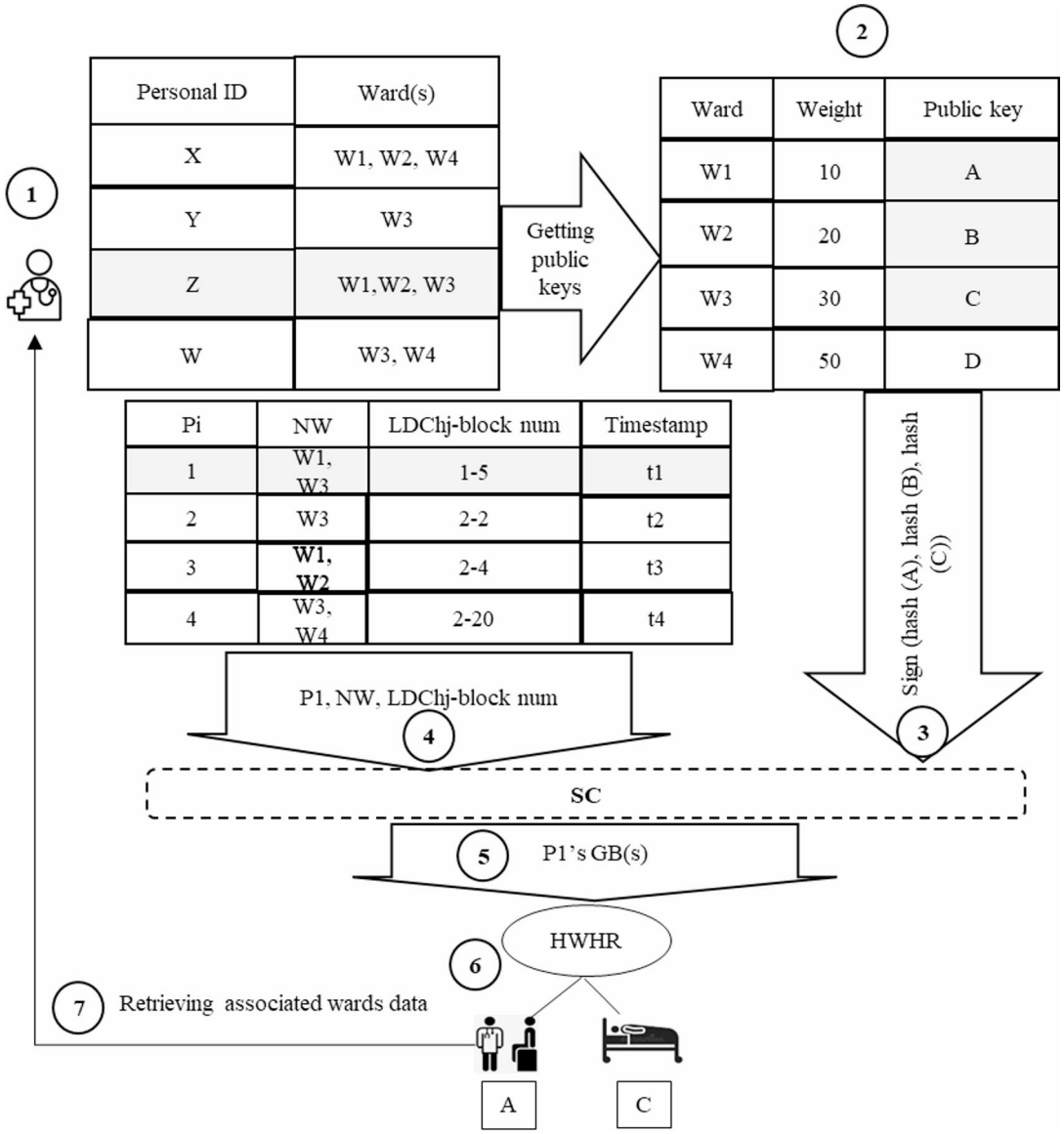

Fig. 15. Steps for retrieving patient healthcare data for physician.

```
Input: the SC, the GCh.
Authenticate physicians by the SC.
Generate a pair of private-public keys for each ward.
Create a patients table include Pi, NW, and LDChj-block num.
Create a wards table include weight, public key.
Create a physicians table include personal ID and ward(s).
Create a signature script by a physician:
    | Sign public key(s) of ward(s) which accesses to them.
Send the signature script to the SC.
Retrieve LDChj-block num from the patients table by the SC.
    | Retrieve the desired GB(s) from the GCh.
    | Retrieve the desired ward(s) using HWHT by the SC.
    | Retrieve the raw healthcare data from local IPFS by the SC using LDChj.
Outputs: the desired healthcare data.
```

Fig. 16. Pseudocode of the proposed access control mechanism.

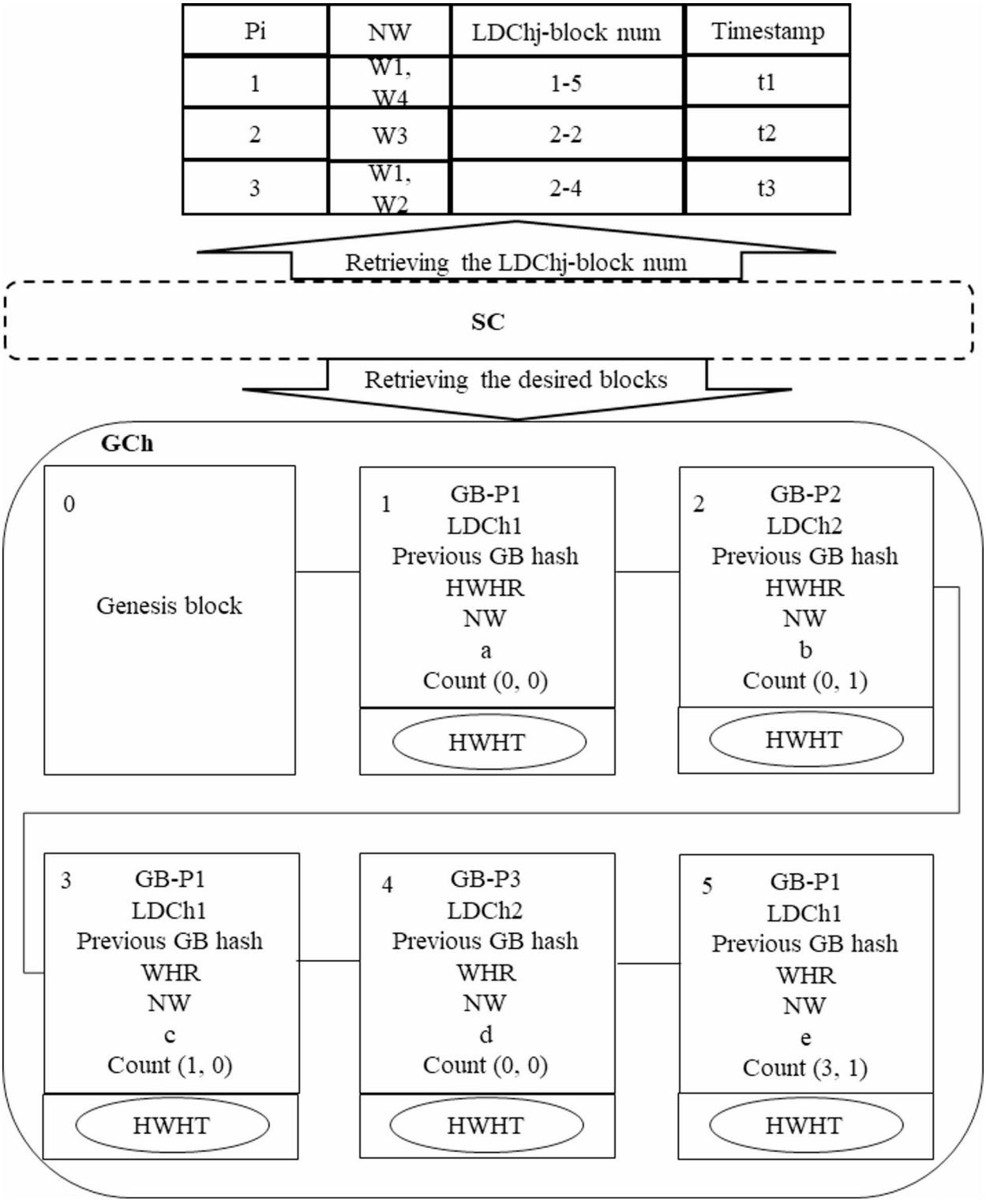

Fig. 17. Retrieving healthcare data from GCh using the proposed access control mechanism.

Evaluation and results

This section discusses a security analysis and performance evaluation of the proposed DHC model.

Security criteria

The DHC model incorporates the following security criteria:

- Authentication: In the proposed model, authentication for patients and physicians at each hospital is facilitated through the SC. Additionally, each ward is authenticated using the SC, which generates a private-public key pair for it. Information regarding patients, physicians, and wards is stored in the IDB. Furthermore, specific fog nodes in each hospital are authenticated as validator nodes used during the consensus process.
- Authorization: The SC is the sole entity with access to the GCh and is responsible for retrieving GBs. It subsequently presents patient healthcare data in each ward utilizing the GB and the physician's signature script. This signature script, as detailed in the "Proposed access control mechanism" section. includes the hashed public key of the wards to which the physician has authorized access.
- Integrity: Patient data integrity ensured both on-chain and off-chain. Off-chain data integrity is maintained at the time of creating an LDCh by incorporating the hash of each LDB in the header of the subsequent block. When trimming an LDCh, at least six blocks preceding the first RB are preserved to maintain data integrity, following principles established in Bitcoin. On-chain, the integrity of patient data is reinforced by including the hash of the previous GB in the header of each GB in the GCh.
- Confidentiality and privacy: Patient data retrieval is conducted using the proposed signature script tailored for physicians who are authenticated and registered in the IDB, and who possess access rights to the relevant ward. In the proposed access control mechanism described in the "Proposed access control mechanism" section. each physician generates a signature script based on their access control level for different wards. The SC then presents the patient data under this generated signature script.
- Data ownership and access control: In the proposed model, hospitals retain ownership of their patient data, with access to healthcare data granted to physicians through the proposed access control mechanism. Each physician first creates a signature script based on the wards they are permitted to access, after which the SC presents healthcare data aligned with their access control levels.
- Traceability: Each LDB and GB includes a Patient ID (Pi) and WHR in the header to denote the associated patient and their referred wards. Additionally, to identify the LDCh in which a block was created, the LDChj value is included in the header of each GB. Each GB also features the Count attribute, as discussed in the "Proposed Global Block (GB) and Global Chain (GCh) structures" section. which tracks the patient's GBs by initializing the last GB number at a specific Timestamp in the LDChj.
- Availability: Each hospital temporarily stores patients' LDBs in its LDCh, ensuring data integrity by including the hash of each block in the corresponding subsequent block. GBs within the GCh are also hashed and linked together, maintaining redundancy in distributed nodes.
- Scalability: The implementation of the proposed WHT structure for LDB and GB creation, as described in both the "Proposed Local Dynamic Block (LDB) and Local Dynamic Chain (LDCh) structures" and the "Proposed Global Block (GB) and Global Chain (GCh) structures" sections. significantly reduces the storage requirements for both LDCh and GCh. Additionally, trimming further decreases the storage space needed for LDCh and boosts its scalability. The proposed LDCh structure reduces the storage needs of the GCh by ensuring that only select portions of the LDBs in the LDCh are permanently stored in the GCh, thereby enhancing the scalability of the model and reducing storage demands on distributed nodes.
- Trustworthiness: The scalability of distributed nodes is enhanced through the use of the proposed WHT, LDB, GB, LDCh, and GCh structures, facilitating greater node participation in storing GBs and, subsequently, increasing the overall trustworthiness of the blockchain.

Table 2 Compares the DHC with other blockchain-based storage and retrieval models[21,28] regarding these security criteria.

As detailed in Table 2, the DHC model offers solutions that uphold security criteria relating to privacy, ownership, traceability, and scalability. Notably, the BCLOD model[21] lacks provisions to maintain ownership, and the data stored in DChain[28] is not traceable.

Security attacks

The DHC model is designed to mitigate several types of security attacks through the following mechanisms:

- Distributed denial of service (DDoS): As a consortium blockchain, the DHC model employs an SC to authenticate patients, wards, fog nodes, and LDCh present in each hospital. Consequently, only transactions from authenticated patients are included in the LDBs. Moreover, RQ processes only LDBs from authenticated LDChs before converting them into RBs. In addition, the TLC, as detailed in the "Proposed Two-Layer Consensus (TLC) protocol" section. accepts block validation results from fog nodes authenticated by the SC. This layered approach effectively prevents DDoS attacks targeting the transaction pool, RQ, and block verification processes.
- Eclipse: The DHC model presents a consortium blockchain wherein each fog node in a hospital is authenticated by the SC. As a result, fog nodes in the first layer of TLC explained in the "Proposed Two-Layer Consensus (TLC) Protocol" section. are restricted to verifying LDBs that reside in the same hospital. Moreover, in the

No.	Research	Privacy	Data ownership	Traceability	Scalability
1	DChain[28], 2024	√	√	-	√
2	BCLOD[21], 2025	√	-	√	√
3	DHC	√	√	√	√

Table 2. Comparison of DHC with previous research considering security criteria.

second layer of the proposed consensus, only a single fog node from each hospital is designated as a verifier. These nodes, identified as trustworthy, are elected as leaders based on their performance in the Raft protocol voting process conducted in the first layer. This design effectively mitigates the risk of Eclipse attacks across both layers of the proposed consensus protocol.

- Sybil: Fog nodes authenticated by the SC are allocated only one voting score in each consensus round, thereby preventing Sybil attacks in both the first and second layers of TLC, as discussed in the "Proposed Two-Layer Consensus (TLC) Protocol" section.
- Forks: As outlined in the "Proposed Local Dynamic Chain (LDCh) structure" section. once LDBs are created in each hospital and subsequently, converted into RBs, they are transferred to RQ for permanent storage. The RQ accepts only RBs originating from LDChs previously authenticated by the SC. RQ then sequentially selects RBs from this authenticated queue and submits them to the blockchain network for verification. This sequential processing ensures that no two blocks are verified simultaneously, thus effectively preventing the occurrence of forks in the DHC model.

Table 3 Compares the DHC with other blockchain-based storage and retrieval models considering various attacks.

As indicated in Table 3, the BCLOD model[21], functioning as a consortium blockchain network, is capable of preventing DDoS, Eclipse, Sybil, and Forks attacks through the implementation of clustering nodes in each region and appropriate access control mechanisms. Similarly, the DHC effectively prevents all aforementioned attacks according to the structural designs outlined in the "Proposed Two-Layer Consensus (TLC) Protocol" section. In contrast. the DChain model presented in[28] lacks solutions to safeguard against any of these security threats.

Performance analysis

This section analyzes the time complexity, space complexity, and retrieval rate of healthcare data in the DHC model.

Retrieval rate of patient healthcare records

The retrieval rate in the proposed model refers to the quantity of patient healthcare records accessed via the blockchain. As outlined in the "Proposed Local Dynamic Block (LDB) and Local Dynamic Chain (LDCh) structures" section, each GB comprises transactions that include hashes of all healthcare data associated with a specific ward over a defined period. Assume the threshold value in each LDB is designated as d. The number of patient healthcare data stored in each ward over d days is determined using Eq. (2):

$$1 \leqslant retrieval\_rate\_EHRs \leqslant d \tag{2}$$

According to Eq. (2), if at least one healthcare record is present in a ward, then that ward is represented in the WHT. Consequently, healthcare records can be retrieved by accessing the corresponding transactions related to that ward. If a ward stores one healthcare record for each day of hospitalization of a patient, up to d records may be retrieved from the relevant transaction of that ward. Table 4 compares the retrieval rate of the DHC model with other blockchain-based data storage and retrieval models.

As illustrated in Table 4, the retrieval rate in DHC ranges from one to d patient healthcare records. In contrast, both the BCLOD model[21] and the DChain model[28] exhibit a retrieval rate of only one, akin to that of the traditional blockchain (TChain)[40].

Bandwidth

Assuming that the number of transactions in a GB is 1044, the bandwidth required for transfer is 67,070 bytes. This requirement is also compared with other blockchain-based storage and retrieval models in Table 5.

As shown in Table 5, TChain[40] requires the most bandwidth for block transfer because it stores both raw data and transactions in the block. Conversely, BCLOD[21] transmits only partial blocks over the network, with linked blocks stored locally on the cluster's partial chain. Each partial block contains only the body hash and header of the linked block, thus minimizing bandwidth requirements. DChain[28] requires minimal bandwidth for block transfers by storing transactions in IPFS and allocating just one address for each block. Furthermore, each block in DHC solely contains hashes of transactions, with the raw data stored on local IPFS servers, resulting in bandwidth consumption approximately 12 times lower than that of TChain[40]. Although the required bandwidth to transfer a block in DChain[28] and BCLOD[21] is significantly lower than that of DHC, each transaction in DHC addresses at least one and at most d healthcare records, as indicated in the "Retrieval rate of patient healthcare records" section. whereas each transaction in DChain[28] and BCLOD[21] includes only one data record.

No.	Research	DDoS	Eclipse	Sybil	Forks
1	DChain[28], 2024	-	-	-	-
2	BCLOD[21], 2025	√	√	√	√
3	DHC	√	√	√	√

Table 3. Comparison of DHC with previous research considering security attacks.

No.	Research	Retrieval rate of healthcare records
1	TChain[40], 2024	1
2	DChain[28], 2024	1
3	BCLOD[21], 2025	1
4	DHC	A minimum of 1 A maximum of *d*

Table 4. Comparison of the retrieval rate of patient healthcare records in the proposed model and other blockchain-based data storage and retrieval models.

No.	Research	Block size (Bytes)	Multiplication factor
1	TChain[40], 2024	825,098	1
2	DChain[28], 2024	300	2,750
3	BCLOD[21], 2025	286	2,884
4	DHC	67,070	12

Table 5. Comparison of the required bandwidth to transmit a block in DHC and other models.

No.	Research	Global throughput	Multiplication factor (global)	Local throughput	Multiplication factor (local)
1	TChain[40], 2024	1,044	1	-	-
2	DChain[28], 2024	21,733	2,750	-	-
3	BCLOD[21], 2025	1,044 *n*	*N*	1,044 *n*	1
4	DHC	A minimum of 1,044 A maximum of 1,044 *d*	A minimum of 1 A maximum of *d*	A minimum of 1,044 *n* A maximum of 1,044 *dn*	A minimum of 1 A maximum of *d*

Table 6. Comparison of throughput of the proposed model and other models.

Throughput

In the DHC model, each LDCh independently verifies patient healthcare data and stores this information in its corresponding LDB. Once blocks are validated and converted to a ready state, they are verified for inclusion in the GCh and subsequently form GBs. Considering that each transaction contains a minimum of one and a maximum of d patient healthcare records, as specified in Eq. (2), the quantity of healthcare records stored in an LDB with transaction size T can be calculated using Eq. (3):

$$T \leqslant NumberofEHRofLDB \leqslant Td \tag{3}$$

As shown in Eq. (3), the number of healthcare records stored in each LDB depends on the number of days (d) that the healthcare data is stored. Assuming that there are n LDChs in the environment that independently and simultaneously verify blocks, then the local throughput can be derived using Eq. (4):

$$Tn \leqslant Throughput \leqslant Tdn \tag{4}$$

As demonstrated in Eq. (4), throughput in the DHC model for T transactions is at least n times and at most dn times the number of transactions that can be stored. Table 6 compares the throughput of the proposed model with other models[21,28,40].

As noted in Table 6, TChain[40] includes 1,044 patient healthcare records per block, resulting in the verification of only 1,044 records in each consensus round, making its throughput the lowest among the models compared[21,28]. In contrast, DChian[28], utilizes IPFS servers to include a larger number of healthcare records per block, thereby increasing its throughput relative to TChain[40]. BCLOD[21] enhances efficiency by conducting consensus rounds simultaneously across different clusters, each verifying distinct blocks. If each block contains 1,044 transactions and there are n clusters, the throughput of BCLOD[21] would be n times that of TChain[40]. In the case of DHC, if each block similarly contains 1,044 transactions, where each transaction corresponds to one patient healthcare record, and the verification of LDBs in LDChs occurs simultaneously, the local throughput would range from at least 1,044 n to a maximum of 1,044 dn as per Eq. (4). Assuming the number of clusters in BCLOD[21] matches the number of LDChs in DHC, the local throughput for DHC could be at most d times the throughput of BCLOD[21]. However, since DHC verifies only a single GB globally in each time frame, its global throughput is at least 1 and at most d times that of TChain[40].

Storage space

In the DHC model, as outlined in the "Proposed Local Dynamic Block (LDB) and Local Dynamic Chain (LDCh) structures" section. a WHT is initially created for each patient based on the wards to which they have been

referred. Subsequently, the patient's healthcare records are documented in the appropriate wards, and an LDB is generated for each day of the patient's hospitalization. If the last flag equals 1 or the threshold reaches the maximum value, the final LDB created during the specified period transitions into an RB state. A transaction is generated for all patient healthcare records in each ward and these transactions are incorporated into the RB. Therefore, the total number of transactions for each RB depends on the number of wards, and each transaction addresses at least one and at most d patient healthcare records according to Eq. (2), where d represents the threshold value. Table 7 presents a comparison of the required storage space for $d = 4$ days in the DHC and other models.

As illustrated in Table 7, under the assumption that one LDB is created for each day and the number of transactions is 1,044, then the required storage space in the GB for maintaining patient healthcare records over four days amounts to 67,070 bytes. Furthermore, if we assume the BCLOD model[21] utilizes four clusters, then every four days, each cluster produces one linked block containing 1,044 transactions along with three partial blocks. Consequently, the required storage space for each partial chain is calculated to be 67,928 bytes. While the required storage space for partial chains over $d = 4$ days is slightly greater than that needed for DHC, such space requirements in BCLOD[21] will escalate with an increase in d and the number of clusters. In contrast, the required storage space for GB in the DHC remains constant. Additionally, while the required storage space in DChain[28] is significantly smaller than that of DHC, this model incurs a higher implementation cost due to its reliance on global IPFS servers.

Performance evaluation

The proposed DHC model was evaluated using Python 3.9.5 on a Windows 10, operating system with a 64bit, 6 GB of RAM, and an Intel Core i5 processor. The evaluation utilized COVID-19 death rate data provided by the World Health Organization (WHO)[41].

Retrieval rate of patient healthcare records

The retrieval rate for patient healthcare records depends on the duration for which the data is stored in the LDB as detailed in the "Retrieval rate of patient healthcare records" section. Here, the scalability of retrieval rates is compared for records contained in blocks that have 1,044 transactions, using different threshold values, in both the proposed DHC model and BCLOD[21]. The results of this evaluation results are illustrated in Fig. 18.

As shown in Fig. 18, the maximum retrieval rate for patient healthcare records per transaction in DHC blocks corresponds to the threshold value, attributable to the use of the WHT in block construction. In contrast, The BCLOD[21] retrieves only one healthcare record per transaction since it, along with many blockchain-based storage and retrieval models, utilizes a Merkle tree for data storage. Hence, increasing the threshold value increases the scalability of the retrieval rate for patient healthcare records.

Storage space

The healthcare data presented by the WHO were stored using both the DHC and BCLOD models[21,28]. Assuming that each block contains 1,044 transactions, the required storage space on IPFS servers for varying numbers of blocks in both DHC and BCLOD[21] is depicted in Fig. 19.

As illustrated in Fig. 19, the DHC model requires less storage space on IPFS servers compared to the BCLOD model[21]. This difference arises because the DHC utilizes only local IPFS servers to store raw data in each hospital. In contrast, BCLOD[21] employs both local and global IPFS servers, storing raw data on local IPFS servers while storing the hashes of each piece of raw data along with their respective cluster numbers on global IPFS servers. Thus, the required storage space for BCLOD[21] is marginally higher than that of DHC.

Execution time

As mentioned in the "Proposed Two-Layer Consensus (TLC) Protocol" section. TLC verifies blocks in two stages: local and global, employing the Raft consensus algorithm. Figure 20 compares the execution time of TLC with other consensus protocols for blocks containing various transaction numbers of transactions.

In Fig. 20, is evident that the execution time of the Green-PoW protocol[43] is the highest among the consensus protocols evaluated, due to its reliance on puzzle-solving. Conversely, the PoR protocol[42] has the shortest execution time because it utilizes random nodes for block verification. Furthermore, the PoAct protocol[44], which is based on the Delegated Proof of Stake (DPoS) model[45], involves fewer participating nodes in the consensus process than both BCLOD[21] and TLC, resulting in reduced execution times. The execution time for block verification in BCLOD[21] and TLC is lower than that of Green-PoW[43] as they both employ the Raft protocol[39]. Since block verification in TLC is performed globally, its execution time is slightly longer than the verification

No.	Research	Required storage space for chain of blocks (Bytes)	Multiplication factor
1	TChain[40], 2024	3,300,392	1
2	DChain[28], 2024	1,200	2,753
3	BCLOD[21], 2025	67,928	48
4	DHC	67,070	49

Table 7. Comparison of the required storage space of the block in DHC and other models.

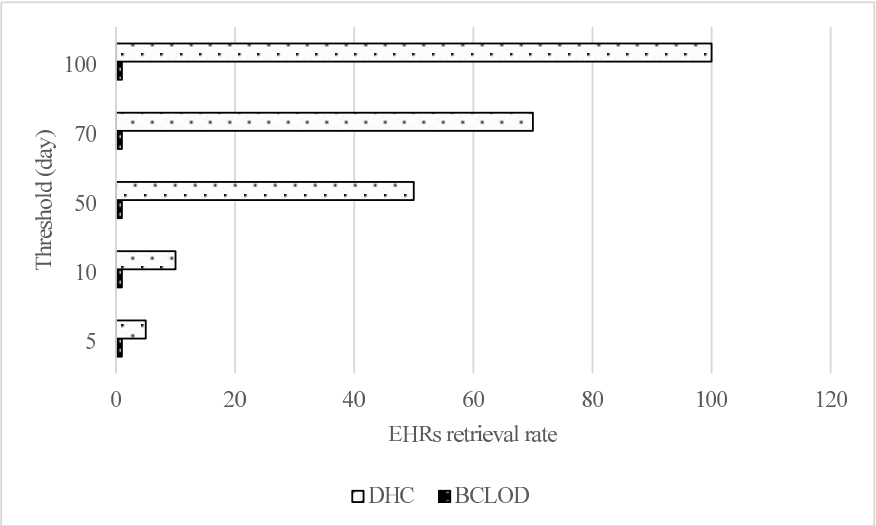

Fig. 18. Comparison of the retrieval rates of patient healthcare records in DHC and BCLOD[21].

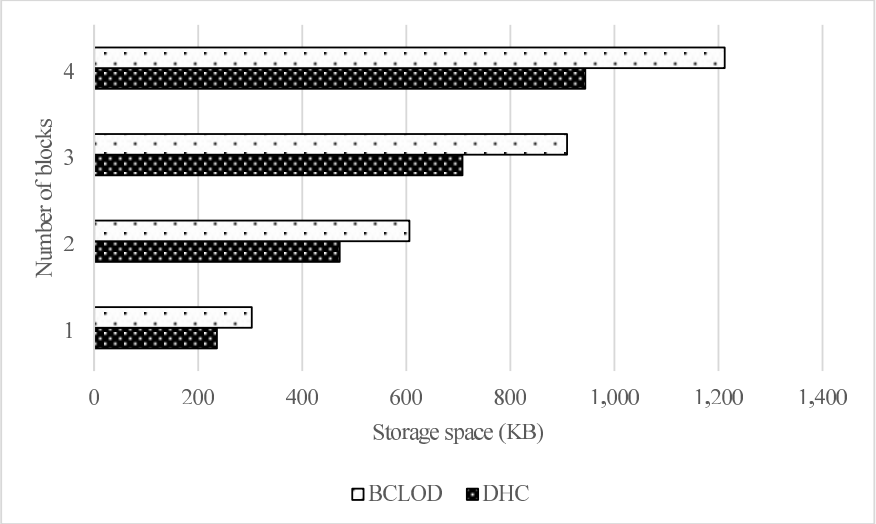

Fig. 19. Comparison of required storage space for IPFS in DHC and BCLOD[21].

time in BCLOD[21]. Additionally, the execution time of block creation in DHC and BCLOD[21] is compared across different numbers of healthcare records, as presented in Fig. 21.

As shown in Fig. 21, the execution time for block creation in the DHC for 20, 80, 200, and 400 patient healthcare records exceeds that of BCLOD[21]. Assuming the number of transactions per block in DHC is set at 4, then 20, 80, 200, and 400 patient healthcare records can be stored in a single block for threshold values of 5, 20, 50, and 100 respectively. In contrast, BCLOD[21] requires 5, 20, 50, and 100 blocks, each containing 4 transactions, to accommodate the same number of patient healthcare records. Although the block creation time in DHC is longer than in BCLOD, the BCLOD approach requires the creation of more blocks to store the same number of patient healthcare records. This results in increased communication overhead and greater storage space requirements.

Conclusion

This paper presents a lightweight, scalable, and dynamic blockchain-based model for storing and retrieving patient healthcare records, referred to as the DHC model. The Local Dynamic Block (LDB) and Local Dynamic Chain (LDCh) structures are introduced to manage these records off-chain, allowing for efficient on-chain storage. These structures support the development of a lightweight and scalable off-chain solution while ensuring data integrity. In DHC, each hospital maintains its patients' healthcare data within a proposed LDB, which is organized based on the Huffman tree and ward assignments. Each patient's data are updated in the LDB according to the wards visited daily and are subsequently added to the LDCh, where the most recent LDB reflects

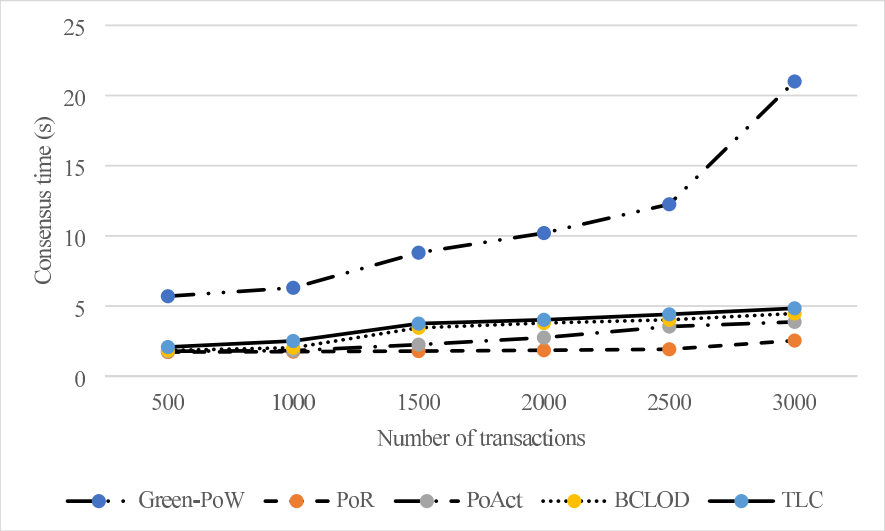

Fig. 20. Comparison of block verification execution times in TLC and other consensus protocols[21,42–44].

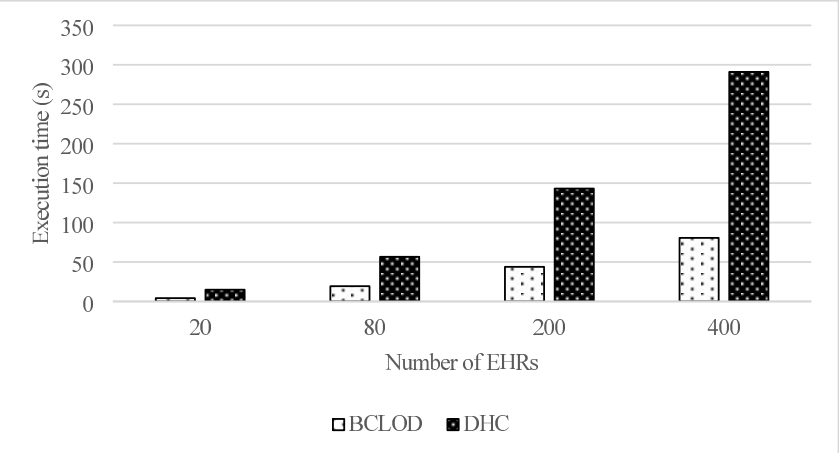

Fig. 21. Comparison of block creation execution times in DHC and BCLOD[21] for varying numbers of healthcare records.

all healthcare data throughout the hospitalization period. This approach enables the LDCh to demonstrate the integrity of hospital healthcare data during preprocessing by retaining only the latest LDBs for each patient. Raw data are stored on local IPFS servers of the respective cluster, and their hashes are stored as a proposed ready block, which is introduced by the corresponding LDCh for on-chain permanent storage. This block is then placed in the proposed ready queue to be converted into a Global Block (GB) and verified using a two-layer consensus algorithm (TLC) based on Raft. Upon verification, the GB is integrated into the proposed global chain. The proposed GB structure includes data from the most recent LDB for each patient, along with essential information for tracking the patient and the hospital where care was provided. Consequently, the GB aggregates healthcare transactions from multiple hospitals. The model also employs smart contracts to safeguard patient privacy, enabling data presentation consistent with user access control levels through the proposed signature script and the GB. Evaluations indicate that the Huffman-tree–driven block structuring yields substantial improvements in time and space complexity, as well as retrieval rates for patient healthcare records, validating the scalability and efficiency of the proposed model. Moreover, the TLC consensus protocol helps mitigate DDoS, Sybil, Eclipse, and fork attacks. Future work will explore game-theoretic and machine-learning approaches to optimize the Huffman-tree construction and preprocessing of healthcare data, as well as extensions to open healthcare data initiatives.

Data availability

https://www.who.int/data/sets/global-excess-deaths-associated-with-covid-19-modelled-estimates.

Received: 15 April 2025; Accepted: 10 November 2025
Published online: 24 December 2025

References

1. Lytras, M. D. & Papadopoulou, P. *Applying Big Data Analytics in Bioinformatics and Medicine.* (IGI Global, 2017).
2. Khan, A. A. et al. A cost-effective approach using generative AI and gamification to enhance biomedical treatment and real-time biosensor monitoring. *Sci. Rep.* **15**, 17305 (2025).
3. Lytras, M. D., Papadopoulou, P. & Sarirete, A. Smart Healthcare: emerging technologies, best practices, and sustainable policies. *Innovation health informatics* https://doi.org/10.1016/B978-0-12-819043-2.00001-0 (2020).
4. Kiourtis, A., Mavrogiorgou, A., Symvoulidis, C., Tsigkounis, C. & Kyriazis, D. Indexing of cloud stored electronic health records for consented third party accessing. In: *28th Conference of Open Innovations Association (FRUCT).* 158–166 (IEEE, 2021).
5. Li, X., Lu, Y., Fu, X. & Qi, Y. Building the internet of things platform for smart maternal healthcare services with wearable devices and cloud computing. *Future Generation Comput. Syst.* **118**, 282–296 (2021).
6. Yan, S., He, L., Seo, J. & Lin, M. Concurrent healthcare data processing and storage framework using deep-learning in distributed cloud computing environment. *IEEE Trans. Industr Inf.* **17**, 2794–2801 (2020).
7. Altowaijri, S. M. An architecture to improve the security of cloud computing in the healthcare sector. *Smart Infrastructure Applications: Found. Smart. Cities Soc.* 249–266 (2020).
8. Chinnasamy, P. & Deepalakshmi, P. HCAC-EHR: hybrid cryptographic access control for secure EHR retrieval in healthcare cloud. *J. Ambient Intell. Humaniz. Comput.* **13**, 1001–1019 (2022).
9. Mayer, A. H., da Costa, C. A. & Righi, R. Da R. Electronic health records in a blockchain: A systematic review. *Health Inf. J.* **26**, 1273–1288 (2020).
10. Nasab, S. S. F., Bahrepour, D. & Tabbakh, S. R. K. A review on secure data storage and data sharing technics in blockchain-based IoT healthcare systems. In: *12th International Conference on Computer and Knowledge Engineering (ICCKE).* 428–433 (IEEE, 2022).
11. Fateminasab, S. S., Memarian, S., Tabbakh, S. R. K. & Romero-Ternero, M. C. A Review on Open Data Storage and Retrieval Techniques in Blockchain-based Applications. In: *10th International Conference on Web Research (ICWR).* 297–302 (IEEE, 2024).
12. Azbeg, K., Ouchetto, O., Andaloussi, S. J. & BlockMedCare A healthcare system based on IoT, blockchain and IPFS for data management security. *Egypt. Inf. J.* **23**, 329–343 (2022).
13. Hossein, K. M., Esmaeili, M. E., Dargahi, T., Khonsari, A. & Conti, M. BCHealth: A novel blockchain-based privacy-preserving architecture for IoT healthcare applications. *Comput. Commun.* **180**, 31–47 (2021).
14. Culver, K. *Blockchain Technologies: A whitepaper discussing how the claims process can be improved.* (2016)
15. Khan, A. A. et al. A lightweight scalable hybrid authentication framework for internet of medical things (IoMT) using blockchain hyperledger consortium network with edge computing. *Sci. Rep.* **15**, 19856 (2025).
16. Khan, A. A. et al. BDLT-IoMT—a novel architecture: SVM machine learning for robust and secure data processing in internet of medical things with blockchain cybersecurity. *J. Supercomput.* **81**, 271 (2025).
17. Zaabar, B., Cheikhrouhou, O., Jamil, F., Ammi, M. & Abid, M. HealthBlock: A secure blockchain-based healthcare data management system. *Comput. Netw.* **200**, 108500 (2021).
18. Daraghmi, E. Y., Daraghmi, Y. A., Yuan, S. M. & MedChain A design of blockchain-based system for medical records access and permissions management. *IEEE access.* **7**, 164595–164613 (2019).
19. Khan, A. A. et al. Quantum computing empowering blockchain technology with post quantum resistant cryptography for multimedia data privacy preservation in cloud-enabled public auditing platforms. *J. Cloud Comput.* **14**, 43 (2025).
20. Dhasarathan, C., Shanmugam, M., Khapre, S. P., Shukla, A. K. & Shankar, A. Blockchain-enabled decentralized reliable smart industrial internet of things (BCIIoT). In Innovations in the Industrial Internet of Things (IIoT) and Smart Factory. 192–204 (IGI Global Scientific Publishing, 2021).
21. Fateminasab, S. S., Bahrepour, D. & Tabbakh, S. R. K. A novel blockchain-based clustering model for linked open data storage and retrieval. *Sci. Rep.* **15**, 5931 (2025).
22. Zou, R., Lv, X., Zhao, J. & SPChain Blockchain-based medical data sharing and privacy-preserving eHealth system. *Inf. Process. Manag.* **58**, 102604 (2021).
23. Jia, D. Y., Xin, J. C., Wang, Z. Q., Lei, H. & Wang, G. R. SE-chain: a scalable storage and efficient retrieval model for blockchain. *J. Comput. Sci. Technol.* **36**, 693–706 (2021).
24. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A. & Medrec Using blockchain for medical data access and permission management. In: *2nd international conference on open and big data (OBD).* 25–30 (IEEE, 2016).
25. Fan, K. et al. Efficient and secure medical data sharing via blockchain. *J. Med. Syst.* **42**, 1–11 (2018).
26. Amudha, G. Dilated transaction access and retrieval: improving the information retrieval of blockchain-assimilated internet of things transactions. *Wirel. Pers. Commun.* **127**, 85–105 (2022).
27. Tu, J., Zhang, J., Chen, S., Weise, T. & Zou, L. An improved retrieval method for multi-transaction mode consortium blockchain. *Electron. (Basel).* **9**, 296 (2020).
28. Mahmud, M., Sohan, M. S. H., Reno, S., Sikder, M. A. B. & Hossain, F. S. Advancements in scalability of blockchain infrastructure through IPFS and dual blockchain methodology. *J. Supercomput.* **80**, 8383–8405 (2024).
29. Johari, R., Kumar, V., Gupta, K. & Vidyarthi, D. P. BLOSOM: blockchain technology for security of medical records. *Ict Express.* **8**, 56–60 (2022).
30. Dagher, G. G., Mohler, J., Milojkovic, M., Marella, P. B. & Ancile Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. *Sustain. Cities Soc.* **39**, 283–297 (2018).
31. Amadeo, M., Campolo, C., Molinaro, A. & Mitton, N. Named data networking: A natural design for data collection in wireless sensor networks. In *In 2013 IFIP wireless days (WD)* 1–6 (IEEE, 2013).
32. Al Omar, A., Rahman, M. S., Basu, A., Kiyomoto, S. & Medibchain A blockchain based privacy preserving platform for healthcare data. In *Security, Privacy, and Anonymity in Computation, Communication, and Storage: SpaCCS 2017 International Workshops, Guangzhou, China, December 12–15* Proceedings. **10** 534–543 (Springer, 2017).
33. Cheng, X. et al. Springer,. Blockchain-based secure authentication scheme for medical data sharing. In: *International Conference of Pioneering Computer Scientists, Engineers and Educators.* 396–411 (2019).
34. Choudhury, O., Fairoza, N., Sylla, I. & Das, A. A blockchain framework for managing and monitoring data in multi-site clinical trials. arXiv preprint at https://arXiv.org/abs/1902.03975 (2019).
35. Akkaoui, R., Hei, X., Guo, C. & Cheng, W. R. B. A. C. H. D. E. On the design of a role-based access control with smart contract for healthcare data exchange. In: *IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW).* 1–2 (IEEE, 2019).
36. Zhou, L., Marsh, M. A., Schneider, F. B. & Redz, A. Distributed blinding for distributed elgamal re-encryption. In: *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05).* 824 (IEEE, 2005).
37. A Huffman, D. A method for the construction of minimum-redundancy codes. *Proc. IRE.* **40**, 1098–1101 (1952).
38. King, S., Nadal, S. & Ppcoin Peer-to-peer crypto-currency with proof-of-stake. *self-published Paper.* (2012).
39. Ongaro, D. & Ousterhout, J. The raft consensus algorithm. *Lecture Notes* **190**, 2022 (2015).
40. Attention Required! | Cloudflare. https://www.blockchain.com/explorer/charts.

41. Global excess deaths associated with COVID-19, January 2020 - December 2021. https://www.who.int/data/stories/global-excess-deaths-associated-with-covid-19-january-2020-december-2021 (2021).
42. Xie, Q., Dong, F., Feng, X. & HLOChain,. HLOChain: A hierarchical blockchain framework with lightweight consensus and optimized storage for IoT. *Secur. Commun. Net.* **2023**, 3412200 (2023).
43. Lasla, N., Al-Sahan, L., Abdallah, M. & Younis, M. Green-PoW: an energy-efficient blockchain Proof-of-Work consensus algorithm. *Comput. Netw.* **214**, 109118 (2022).
44. Fateminasab, S. S., Bahrepour, D. & Tabbakh, S. R. K. A fair non-collateral consensus protocol based on Merkle tree for hierarchical IoT blockchain. *Sci. Rep.* **15**, 3645 (2025).
45. Saad, S. M. S. & Radzi, R. Z. R. M. Comparative review of the blockchain consensus algorithm between proof of stake (pos) and delegated proof of stake (dpos). *Int. J. Innov. Comput.* https://doi.org/10.11113/ijic.v10n2.272 (2020).

Author contributions
All authors reviewed the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to M.N.T.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.