OPEN ACCESS
Saudi Journal of Engineering and Technology
Abbreviated Key Title: Saudi J Eng Technol
ISSN 2415-6272 (Print) |ISSN 2415-6264 (Online)
Scholars Middle East Publishers, Dubai, United Arab Emirates
Journal homepage: https://saudijournals.com

Review Article

# The Design for Distributed Ledger Based on Main-Sub Ledger Architecture

Wenfeng Li[1*]

[1]Suzhou Shutong Digital Technology Ltd., Suzhou, Jiangsu province, China

**DOI:** 10.36348/sjet.2022.v07i03.006 | **Received:** 01.02.2022 | **Accepted:** 10.03.2022 | **Published:** 15.03.2022

***Corresponding author:** Wenfeng Li
Suzhou Shutong Digital Technology Ltd., Suzhou, Jiangsu province, China

## Abstract

Distributed ledger technology (DLT) offers new and unique advantages for information systems, but some of its features are not a good fit for many applications. We review the properties of DLT and propose a new type of architecture for DLT based on main-sub ledger. Our scheme pays more attention to data privacy, effectively relieves the pressure of data storage for nodes, thereby improves data handling capability.

**Keywords:** Distributed ledger; main-sub ledger; blockchain.



## 1. INTRODUCTION

Ledgers are as old as trading and writing. A ledger is an account book with a specific format that records all economic transactions in chronological order based on the original vouchers. Ledgers are made of a variety of materials, ranging from ancient clay tables to counting sticks and double-entry books to digital spreadsheets and databases and, more recently, distributed ledgers [1]. Each breakthrough in ledger technology leads to major financial innovations and has a profound impact on all aspects of social life. Because they are convenient and efficient, digital ledgers have quickly become the most important accounting medium since the invention of computers. Accounting computerization has now become the primary tool for accounting work.

However, digital ledgers are still centralized, which means that in order to ensure the accuracy of the accounts, both parties in the transaction must find a third party who can be trusted to keep the accounts. Computer algorithms enable distributed ledgers (DL), which are maintained collaboratively. In terms of accounting, there is no fundamental difference between distributed ledgers and traditional ledgers. However, from a technical standpoint, DL not only inherits traditional accounting philosophy, but also has incomparable advantages and capabilities over traditional ledgers as a result of its unique innovations. Following digitization, DL is regarded as the next major leap in ledger technology.

DL is essentially a database of assets that can be shared across multiple sites, geographic locations, or multi-institutional networks. A network's participants can obtain a unique, true copy of the ledger. Within minutes or seconds, any changes in the ledger will be reflected in all copies. Financial, legal, physical, or electronic assets can all be stored in this ledger. The public key cryptography and signature are used as a means of storing assets and validating transactions in DL [2]. A consensus mechanism is an essential component of a distributed ledger system because it ensures system reliability by validating all written records without the involvement of a trusted third party [3]. Records in the ledger can be updated jointly by one, several, or all participants after reaching consensus on the validity of the records, according to the network's consensus rules.

The idea of decentralized record keeping is central to DL. A ledger's primary purpose is to track and document transactions. Maintaining a certain level of privacy is recommended because a ledger transaction transfers ownership of an asset. On the other hand, proving ownership becomes possible for anyone with access to the ledger. When dealing with ledgers, it is necessary to strike a balance between privacy and transparency [4].

In this paper, we propose a new distributed ledger technology based on main-sub-ledgers architecture. It includes a main ledger and *N* sub-

ledgers, which all run in parallel in the same distributed ledger. The key contributions of our paper can be summarized below:

- Propose a new distributed ledger based on the main-sub-ledgers architecture, and standardize the data storage structure of the main ledger.
- Design the process of establishing sub-ledger and define parameters required for it. Redesign the process of generating and adding blocks in the sub-ledger.

The remaining part of the paper is structured as follows: Section 2 reviews the operation principle of DL/blockchain. Section 3 describes the proposed main-sub-ledger-based Distributed Ledger (MSDL). Section 4 discusses performance of MSDL and Section 5 provides the conclusion of the paper.

## 2. Distributed Ledger Technology/Blockchain

### 2.1 Structure of DLT/Blockchain

Distributed ledgers are a form of technology used to distribute, exchange, or store data among users via public or private networks [5]. It can be simply seen as a distributed database with certain specific properties. The blockchain is the most widely used data structure for distributed ledger [6]. The decentralized, replicated data synchronized among separate network nodes, which may be geographically dispersed, is a key feature of a blockchain-based system. Blockchains employ cryptographic and algorithmic methods to record and synchronize data across a network in an immutable manner. Consensus protocols are designed so that when the consensus is reached, all nodes in the network have an identical copy of the distributed ledger records [7, 8].

The blockchain's root is called as genesis block, which is the first block in the chain. It is the common origin of all blocks and contains the information that is generally known to all nodes. A blockchain begins with a genesis block, followed by subsequent blocks. As shown in the Fig 1: **The structure of blockchain**, each block's structure consists of a block header and a block body. The block header contains a previous block's hash, nonce, and timestamp, as well as the Merkel root. Each current block is linked to the previous block by using the previous block's hash as a chain [9]. Depending on the blockchain's requirements, the block body contains a list of transactions as well as some additional data. For immutability, all transactions should be hashed using Merkel hash which is derived from the Merkel algorithm. According to the calculation sequence of the Merkel tree in Fig 2: **Block header format**, a Merkel root finally is obtained by hashing all transactions and it is eventually appended in the block header.

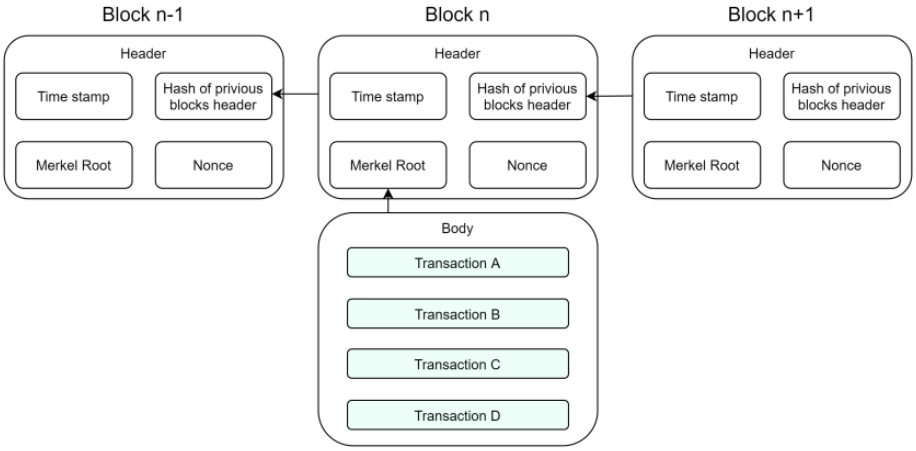


**Fig 1: The structure of blockchain**

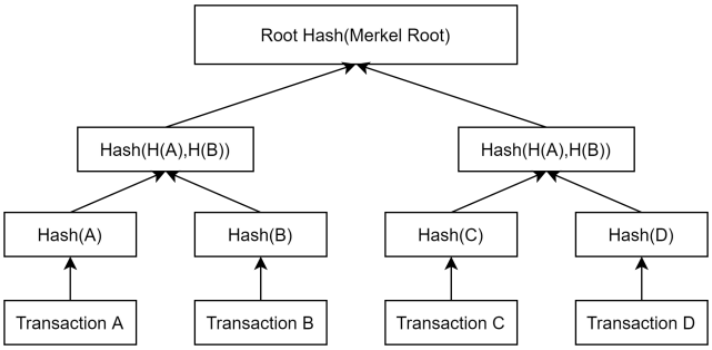


**Fig 2: Block header format**

### 2.2 Operating process

New additions to the database are initiated by one of the nodes, who creates a new "block" of data, for example containing several transaction records. Information about this new data block is then shared across the network, and all network participants collectively determine the block's validity according to a pre-defined consensus algorithm. Only after validation, all participants add the new block to their respective ledgers. Through this mechanism each change to the ledger is replicated across the network and each network member has a full, identical copy of the entire ledger at any point in time [10].

The process to add a new block explained as follows [11]:

Step 1: A node starts a transaction by first creating and then digitally signing it with its private key (created via cryptography).

Step 2: A transaction can represent various actions in a Blockchain. A new block to represent that transaction or set of transactions is then created.

Step 3: A new transaction is broadcasted to all participating nodes to validate that transaction based on predefined scripts. Normally, blockchain need multiple nodes to verify a transaction.

Step 4: Special nodes called miner nodes are responsible to validate new transaction or block and store it onto distributed ledger. Miners get into a competition to resolve a cryptographic hash algorithm oriented complex mathematical problem or puzzle. The solution to this problem, known as Proof of Work (PoW), is a evidence that miner utilized significant computing efforts. Miners may entitled with some incentive for mining that could be either in form of cryptocurrency or transaction charges.

Step 5: Once a transaction is validated, it is appended in a block, then new instance of blockchain is again propagated into network to provide the latest information about a block. On this step, a transaction is getting its first confirmation.

Step 6: This latest block is then stored on distributed ledger and subsequent blocks links with this block via a hash pointer. At this point, the transaction receives its second confirmation and the block gets its first confirmation. Whenever a new block is being created its associated transactions get reconfirmations. Normally, a network requires six confirmations for considering a transaction to be final.

## 3. Main-sub-ledger-based Distributed Ledger (MSDL)

### 3.1 System architecture of MSDL

However, this has a significant impact on the distributed ledger's data processing capabilities.

The key features of DLT/Blockchain, as distinct from other relational databases, are associated with its distributed nature. In DLT/Blockchain, multiple copies of the ledger are held by different parties, with data added by consensus and without the need for a third party [12]. As a result, DLT/Blockchain can provide gains in efficiency, trust and data reconciliation across all ledger participants.

The MSDL consists of two type of ledgers, the main-ledger and the sub-ledger, as well as the corresponding management tools. The data storage organization of the main-ledger and the sub-ledgers adopts the blockchain as shown in Fig 3: **System architecture diagram of MSDL**, thereby forming a main-sub-ledger system architecture, in which a main-ledger and *N* sub-ledgers run in parallel.

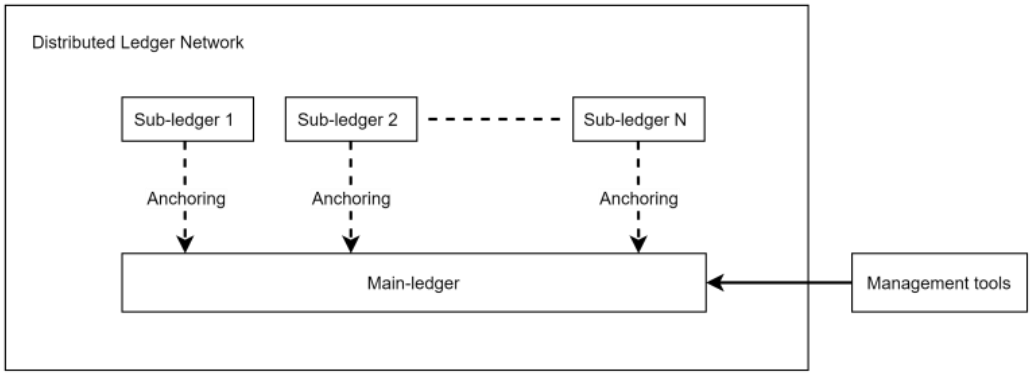


**Fig 3: System architecture diagram of MSDL**

The main-ledger is automatically created when the network is initialized, and all nodes participate in the data consensus of the main ledger. In addition to storing necessary system global information and related data, main-ledger should also contain:

- Sub-ledger list
- Sub-ledger block header information
- The mapping relationship between creator and sub-ledger
- Mapping relationship between network nodes and sub-ledgers

- Sub-ledger status information

Sub-ledgers are established on demand by nodes using system functions. The blocks generated by each sub-ledger are anchored to the main-ledger in as transactions. Each node in the network can establish sub-ledger based on its own business requirements and broadcast it to all nodes via the P2P network. Through the consensus mechanism, it is finally updated to main-ledger saved by each node, ensuring that the main-ledger saved by all nodes is consistent. Each node must store the main ledger as well as all sub-ledger established by itself.

The administration tool is a kind of command-line based client tool. In addition to providing regular distributed ledger management and viewing functions, it also has a sub-ledger established function. Users with management authority of distributed ledger can submit request to establish a sub-ledger to any node in the network through system management tools according to own business requirements.

### 3.2 The process of establish sub-ledger

The steps to establish sub-ledger are as follows:

Step 1: Specify the parameter group, including the unique name of sub-ledger *Name*, the business description of sub-ledger *Desc* , the ID set of node participating in sub-ledger *NodeIDs*, and the consensus algorithm *Cons*;

Step 2: According to the input parameters, system management tool assembles the *command*{*Name* || *Desc* || *NodeIDs* || *Cons*} to create sub-ledger, and generates digital signature *Sign* of *command* by using private key $K_{pri}$. Then, sends *message*{*command* || $K_{pub}$ || *Sign*} to any connectable node server *NodeA* in the network;

Step 3: *NodeA* extracts the public key $K_{pub}$ from the received *message*, then runs the account generation algorithm $f(K_{pub})$ to obtain the client account $Account_{client}$ that initiated the command request, and verifies whether the client account $Account_{client}$ is the same as the pre-configured administrator account $Account_{admin}$ in the network. If the account verification is successful, *NodeA* continues to verify signature *Sign.*

Step 4: The distributed ledger ensures the correct execution of the *command* in the network through the consensus algorithm and the P2P network broadcasting mechanism. When a node in the network executes the command to create a sub-ledger, it must first confirm that it belongs to the node identification set *NodeIDs*. Then, node initializes the sub-ledger instance, allocates computing and storage resources to it. Finally, the details of the sub-ledger is stored in the persistent medium with the *Name* as index.

The administrator account $Account_{admin}$ is obtained by the account generation algorithm $f(K_{pub})$, and configured on all node servers in the network when the distributed ledger is initialized. Table 1 displays the details of algorithm $f(K_{pub})$.

**Table 1: The account generation algorithm $f(K_{pub})$**

| |
|---|
| Intial: Generating key pair($K_{pub}$,$K_{pri}$), giving Version prefix identifier $V$(16 bits), Hash algorithm $H1$, $H2$(256 bits) and encoding algorithm *Base58.* |
| Start:<br>1. $D \leftarrow H1(K_{pub})$<br>2. $Msg \leftarrow V+D$<br>3. Checksum $Sum \leftarrow$the lower 224bits of $H2(Msg)$<br>4. $S1 \leftarrow V+D+Sum$<br>5. $Account_{admin} \leftarrow Base58(S1)$<br>End. |

### 3.3 Broadcast transactions and add blocks

Because a more complex main-sub-ledger structure is employed in MSDL, the propagation of transactions in the network and the generation strategy of blocks have been improved.

#### 1) Broadcast transactions

The distributed ledger access program can submit transaction requests to the sub-ledger in MSDL. In this case, a node will specify a sub-ledger as target ledger when submitting a transaction request. After receiving a transaction request, the node server obtains the *NodeIDs* of the target sub-ledger from the back-end storage of the main-ledger by indexing on the sub-ledger name. And then, the transaction request will be broadcast to the node in *NodeIDs.* If the sub-ledger name is not specified when the transaction is initiated, the main-ledger will be defaulted to the target ledger, and the transaction request will be broadcast to all nodes in the entire network.

#### 2) Add blocks in sub-ledger

As an independent existence, the sub-ledger also has the entire life cycle of distributed ledger technology. When the conditions for creating a new block are triggered, the pending transaction requests will be packaged to generate a new block $Block_{new}$ by node *Leader* which obtains the right to create block, according to the preset priority order and block constraint restrictions.

*Leader* signs with its private key and sends it to other nodes in *NodeIDs*. After reaching a consensus on $Block_{new}$, each node adds $Block_{new}$ to their own sub-ledger to ensure the consistency of the sub-ledger of each node in *NodeIDs.*

Finally, the block header information of $Block_{new}$ is anchored to the main-ledger in the form of transactions, which enhances the immutability and trustworthiness of the sub-ledger.

**4. Evaluation**

The design of the MSDL shows that each sub-ledger has an independent consensus mechanism, and the block header information of each block generated by sub-ledgers is anchored to the main-ledger with a specific transaction type. Thus, the sub-ledger has greater trustworthiness and immutability than DL. The system architecture of MSDL realizes the isolation of data on the physical level. It provides a guarantee for the privacy and security of business data, which is lacking in DL. Meanwhile, the sub-ledgers are logically independent of each other, and run in parallel in the network without interference. These bring significantly higher capacity of overall data processing to MSDL.

## 5. CONCLUSION

We introduced a new type of distributed ledger technology based on main-sub-ledger architecture, which pays more attention to the privacy of data. Theoretically, our scheme has higher overall data processing capability, while providing better data trustworthiness and immutability. It undoubtedly provides a reference for the development of distributed ledger technology. As a future work, we will build ledger platform for our scheme and provide detailed performance analysis of our scheme.

## REFERENCES


1. Burkhardt, D., Werling, M., & Lasi, H. (2018, June). Distributed ledger. In *2018 IEEE international conference on engineering, technology and innovation (ICE/ITMC)* (pp. 1-9). IEEE.
2. Kakavand, H., Kost De Sevres, N., & Chilton, B. (2017). The blockchain revolution: An analysis of regulation and technology related to distributed ledger technologies. *Available at SSRN 2849251*.
3. Ballandies, M. C., Dapp, M. M., & Pournaras, E. (2021). Decrypting distributed ledger design—taxonomy, classification and blockchain community evaluation. *Cluster Computing*, 1-22.
4. Drescher, D., & Kinoshita, L. A. (2020). Blockchain basics [M]. Ascent Audio, 2020.
5. Farahani, B., Firouzi, F., & Luecking, M. (2021). The convergence of IoT and distributed ledger technologies (DLT): Opportunities, challenges, and solutions. *Journal of Network and Computer Applications*, *177*, 102936.
6. Kuhn, R., Yaga, D., & Voas, J. (2019). Rethinking distributed ledger technology. *Computer*, *52*(2), 68-72.
7. Liu, X., Farahani, B., & Firouzi, F. (2020). Distributed ledger technology, in *Intelligent Internet of Things*. Springer, pp. 393–431.
8. Firouzi, F., Chakrabarty, K., & Nassif, S. (Eds.). (2020). *Intelligent internet of things: From device to fog and cloud*. Springer Nature.
9. Shrestha, R., Bajracharya, R., Shrestha, A. P., & Nam, S. Y. (2020). A new type of blockchain for secure message exchange in VANET. *Digital communications and networks*, *6*(2), 177-186.
10. Natarajan, H., Krause, S., & Gradstein, H. (2017). Distributed ledger technology and blockchain.
11. Kaur, M., & Gupta, S. (2021). Blockchain Technology for Convergence: An Overview, Applications, and Challenges. *Blockchain and AI Technology in the Industrial Internet of Things*, 1-17.
12. Deshpande, A., Stewart, K., Lepetit, L., & Gunashekar, S. (2017). Distributed Ledger Technologies/Blockchain: Challenges, opportunities and the prospects for standards. *Overview report The British Standards Institution (BSI)*, *40*, 40.