

Blockchain System and Data Storage Method and Apparatus

Technical Field

The present application relates to the field of software technologies, and in particular, to a blockchain system and a data storage method and apparatus.

Background

Blockchain network, also referred to as distributed ledger network, is an Internet database characterized by decentralization, transparency, and synchronization of database records (i.e., a shared ledger) by all nodes.

In one example, a blockchain network consists of various nodes, each node holding a shared ledger. Data associated with blocks are chronologically recorded on the shared ledger (data associated with a block corresponds to a set of transactions for which consensus regarding legitimacy is reached by all nodes during a period of time). In other words, the shared ledger records a timed chain of blocks of data, hence named “blockchain.” Each node may synchronize the shared ledger and validate the authenticity of each transaction.

In addition, any node has right to suggest addition of a block of data to the shared ledger. All nodes may reach consensus regarding whether transactions corresponding to the block of data suggested to be added is legitimate and add the block of data for which a consensus regarding legitimacy is reached into the shared ledger. There are mainly two types of blockchain networks now: public blockchain networks and consortium blockchain networks.

A public blockchain network is completely decentralized and transparent to the public. Any entity (individual or organization) may become a node in the public blockchain network, which means that any entity may hold the shared ledger by becoming a node and request all nodes to reach a consensus on a block of data and record the block of data on the shared ledger.

However, since any entity may become a node in a public blockchain network, a hacker may easily intrude into the public blockchain network and attempt to control a majority of the nodes, add illegitimate blocks of data to the public blockchain (i.e., the shared ledger), and pose a threat to the security of the public blockchain network.

A consortium blockchain network is partially decentralized and is not open to the public. Only a pre-designated entity may become a node in the consortium blockchain network, while other entities are not qualified to become nodes, nor can they hold the shared ledger or participate in consensus. The consortium blockchain network may provide services to entities outside the consortium blockchain network (non-node entities). A non-node entity may interface with the consortium blockchain network and request the consortium blockchain network to reach a consensus regarding legitimacy of a transaction generated by the non-node entity.

One consortium blockchain network often only relates to one transaction field. Nodes in a

consortium blockchain network are often authoritative institutions in the field. For example, nodes in a consortium blockchain network in the financial field are often big banks and financial regulatory institutions. In addition, nodes in consortium blockchain networks for many transaction fields, such as bills, logistics, healthcare, government and administration, are also authoritative institutions in corresponding fields, respectively.

A non-node entity needs to select a corresponding consortium blockchain network according to a transaction field to which a transaction generated by the non-node entity belongs. Only after the non-node entity interfaces with a consortium blockchain network, can the non-node entity enjoy services provided by the consortium blockchain network. However, consortium blockchain networks may have different interfacing protocols. If a plurality of transactions generated by the non-node entity belong to different transaction fields, the non-node entity has to interface with a plurality of consortium blockchain networks according to different interfacing protocols, respectively, which is not convenient for the non-node entity.

Summary

Embodiments of the present application provide a blockchain system and a data storage method and apparatus, so as to provide security and convenience for blockchain networks.

To solve the above technical problems, the embodiments of the present application are implemented as follows:

a blockchain system according to the embodiments of the present application, comprising a distribution center, a non-consensus subsystem, and a plurality of consensus subsystems;

the non-consensus subsystem comprises a plurality of non-consensus nodes, and each of the consensus subsystems comprises a plurality of consensus nodes;

a non-consensus node of the plurality of non-consensus nodes sends a transaction request to the distribution center;

the distribution center receives the transaction request from the non-consensus subsystem, determines a consensus subsystem corresponding to the transaction request based on transaction data comprised in the transaction request, and forwards the transaction request to the determined consensus subsystem; and

the consensus subsystem receives the transaction request forwarded by the distribution center, and sends the transaction request to all consensus nodes in the consensus subsystem for consensus validation; if the validation is successful, generates a corresponding block according to the transaction request and stores the block into a consortium blockchain corresponding to the consensus subsystem.

A data storing method according to the embodiments of the present application, wherein a blockchain system comprises a distribution center, a non-consensus subsystem, and a plurality of consensus subsystems, the non-consensus subsystem comprises a plurality of non-consensus nodes,

and each of the consensus subsystems comprises a plurality of consensus nodes, the method comprising:

receiving, by the consensus subsystem, a transaction request forwarded by the distribution center, the transaction request comprising transaction data;

sending the transaction request to all consensus nodes in the consensus subsystem for consensus validation;

if the validation is successful, generating a corresponding block according to the transaction request and storing the block into a consortium blockchain corresponding to the consensus subsystem.

A data storing method according to the embodiments of the present application, wherein a blockchain system comprises a distribution center, a non-consensus subsystem, and a plurality of consensus subsystems, the non-consensus subsystem comprises a plurality of non-consensus nodes, and each of the consensus subsystems comprises a plurality of consensus nodes, the method comprising:

receiving, by the distribution center, a transaction request sent by the non-consensus subsystem, the transaction request comprising transaction data;

determining a consensus subsystem corresponding to the transaction request based on the transaction data comprised in the transaction request; and

forwarding the transaction request to the determined consensus subsystem, causing the consensus subsystem to perform consensus validation on the transaction request and store a block corresponding to the validated transaction request into a consortium blockchain corresponding to the consensus subsystem.

A data storing method according to the embodiments of the present application, wherein a blockchain system comprises a distribution center, a non-consensus subsystem, and a plurality of consensus subsystems, the non-consensus subsystem comprises a plurality of non-consensus nodes, and each of the consensus subsystems comprises a plurality of consensus nodes, the method comprising:

sending, by the non-consensus subsystem, a transaction request to the distribution center, the transaction request comprising transaction data, causing the distribution center to forward, based on the transaction data, the transaction request to a consensus subsystem corresponding to the transaction data.

A data storing apparatus according to the embodiments of the present application, wherein a blockchain system comprises a distribution center, a non-consensus subsystem, and a plurality of pieces of the apparatus, the non-consensus subsystem comprises a plurality of non-consensus nodes, and each piece of the apparatus comprises a plurality of consensus nodes, the apparatus comprising:

a receiving module configured to receive a transaction request forwarded by the distribution

center, the transaction request comprising transaction data;

a validation module configured to send the transaction request to all consensus nodes in the consensus subsystem for consensus validation; and

a storing module configured to, if the validation is successful, generate a corresponding block according to the transaction request and store the block into a consortium blockchain corresponding to the consensus subsystem.

A data storing apparatus according to the embodiments of the present application, wherein a blockchain system comprises the apparatus, a non-consensus subsystem, and a plurality of consensus subsystems, the non-consensus subsystem comprises a plurality of non-consensus nodes, and each of the consensus subsystems comprises a plurality of consensus nodes, the apparatus comprising:

a receiving module configured to receive a transaction request sent by the non-consensus subsystem, the transaction request comprising transaction data;

a determining module configured to determine a consensus subsystem corresponding to the transaction request based on the transaction data comprised in the transaction request; and

a forwarding module configured to forward the transaction request to the determined consensus subsystem, causing the consensus subsystem to perform consensus validation on the transaction request and store a block corresponding to the validated transaction request into a consortium blockchain corresponding to the consensus subsystem.

A data storing apparatus according to the embodiments of the present application, wherein a blockchain system comprises a distribution center, the apparatus, and a plurality of consensus subsystems, the apparatus comprises a plurality of non-consensus nodes, and each of the consensus subsystems comprises a plurality of consensus nodes, the apparatus comprising:

a sending module configured to send a transaction request to the distribution center, the transaction request comprising transaction data, causing the distribution center to forward, based on the transaction data, the transaction request to a consensus subsystem corresponding to the transaction data.

From the above-described technical solutions according to the embodiments of the present application, it can be seen that a blockchain system comprising a distribution center, a non-consensus subsystem, and a plurality of consensus subsystems is created in the embodiments of the present application, wherein the non-consensus subsystem comprises a plurality of non-consensus nodes, each of the consensus subsystems comprises a plurality of consensus nodes, and each of the consensus subsystems is equivalent to a consortium blockchain network containing the plurality of consensus nodes. This also means that the consensus subsystems can separately perform consensus validations for different transaction fields. This way, the consensus nodes in the consensus subsystems are responsible for consensus validation, and the non-consensus nodes in the

non-consensus subsystem may send a transaction request to the distribution center. The distribution center determines a consensus subsystem in a particular transaction field based on the transaction data comprised in the transaction request. Then the consensus subsystem performs consensus validation on the transaction request. According to the embodiments of the present application, on one hand, only the consensus nodes are responsible for consensus validation, and non-consensus nodes outside the consensus subsystems cannot participate in consensus validation of a consortium blockchain network. This improves the security of the blockchain network. On the other hand, the distribution center may interface with the consortium blockchain networks, and non-consensus entities (non-consensus nodes) outside the consortium blockchain network only need to interface with the distribution center, and do not need to interface with a plurality of consortium blockchain networks according to different interfacing protocols. This improves the convenience of the blockchain network.

Brief Description of the Drawings

To more clearly describe the technical solutions in the embodiments of the present application or the current technologies, the accompanying drawings to be used in the embodiments or the current technologies will be briefly described below. It is obvious that the accompanying drawings in the description below are merely some embodiments in the present application. On the basis of these accompanying drawings, other relevant drawings are obtainable by one of ordinary skill in the art without creative effort.

FIGS. 1a-b are schematic diagrams of a blockchain system according to some embodiments of the present application;

FIG. 2 is a flow chart of a data storing method according to some embodiments of the present application;

FIGS. 3-5 are schematic diagrams of three data storing apparatuses according to some embodiments of the present application.

Detailed Description

The embodiments of the present application provide a blockchain network and a data storage method and apparatus.

To enable one of ordinary skill in the art to better understand the technical solutions of the present application, the technical solutions in the embodiments of the present application will be clearly and completely described below with reference to the accompanying drawings in the embodiments of the present application. It is obvious that the described embodiments are merely some, but not all, embodiments of the present application. On the basis of the embodiments of the present application, all other embodiments obtainable by one of ordinary skill in the art without creative effort shall fall within the scope of the present application.

As described above, in exemplary applications, there are mainly two application types of

blockchain networks, i.e., public blockchain network and consortium blockchain network. There is no identity restriction for nodes in a public blockchain network, and any entity (individual or organization) may become a node in the public blockchain network to participate in consensus validation on a transaction request. Since the public blockchain network is truly decentralized and completely open, a hacker may have opportunities to control a majority of the nodes and make an illegitimate transaction request to pass validation. In addition, any entity may view all transaction data stored on the public chain, while the transaction data often involve privacy of the node and non-node entities. Even if the transaction data is encrypted, there is still a risk for the encryption to be cracked.

It should be noted that a “shared ledger” is a blockchains managed by a blockchain network, and correspondingly, a consortium blockchain network manages a consortium blockchain, and a public blockchain network manages a public blockchain.

All nodes in a consortium blockchain network are pre-designated authoritative institutions. For example, all nodes in a consortium blockchain network in the financial field may be Industrial and Commercial Bank of China, China Construction Bank, the People's Bank of China, the China Securities Regulatory Commission, the China Banking Regulatory Commission, etc. Such characteristic of the consortium blockchain network gives hackers no opportunity to participate in consensus validation or to view transaction data stored on the consortium blockchain. This way, the security of the blockchain network is greatly improved. However, since nodes in a consortium blockchain network are often authoritative institutions in a transaction field, the consortium blockchain network can only provide public validation services in the particular transaction field. A non-node entity often needs to spend a very high cost to interface with consortium blockchain networks in different transaction fields, which is very inconvenient.

To this end, the technical solution of the present application provides a blockchain system, and a distribution center is established in the system. The distribution center provides interfacing standard protocols for, on one hand, interfacing with non-node entities to accept transaction requests from the non-node entities, and on the other hand, interfacing with various consortium blockchain networks to forward the received transaction requests to corresponding consortium blockchain networks for validation according to the transaction fields corresponding to the transaction requests.

The blockchain system in the technical solution of the present application may incorporate existing consortium blockchain networks that operate independently and non-node entities in need of various consensus validation services in a unified system architecture, provide interfacing standard protocols for the consortium blockchain networks and non-node entities, and even include various validation services in the entire society in the blockchain system. All members in the society may become nodes in the blockchain system (no need to be responsible for consensus

validation), and transaction requests generated in all aspects of life and work of each member may be validated in the blockchain system.

FIG. 1a is a schematic diagram of a blockchain system according to some embodiments of the present application. As shown in FIG. 2, the blockchain system comprises a distribution center, a non-consensus subsystem, and a plurality of consensus subsystems, wherein the non-consensus subsystem may correspond to the non-node entities, and the non-node entities can serve as non-consensus nodes in the non-consensus subsystem. Alternatively, the non-consensus subsystem may also correspond to a public blockchain network, and the non-consensus nodes in the non-consensus subsystem are practically nodes in the public blockchain network, but the non-consensus nodes do not have the function to participate in consensus validation and can only submit transaction requests. On the other hand, each of the consensus subsystems is equivalent to a consortium blockchain network, and consensus nodes in each of the consensus subsystems are practically nodes in the consortium blockchain network corresponding to the consensus subsystem; transaction requests initiated by the non-consensus nodes are allocated by the distribution center in a uniform manner, and the distribution center forwards transaction requests corresponding to different transaction fields to corresponding consensus subsystems (i.e., equivalent to forwarding the transaction requests to corresponding consortium blockchain networks).

In fact, when a consensus validation system of the entire society is based on the blockchain system, the consortium blockchain networks may be deemed as service stations for providing services to each society member. For each society member, there are many transaction requests generated by the society member in social activities due to life or work, which spans a plurality of transaction fields, and the blockchain system may provide one-stop services to the society member. In addition, transaction data stored on the blockchain system also encompasses all aspects of social activities of each society member, including finance, healthcare, education, insurance, shopping, and asset liquidation of the society member, as well as fields such as administration, judiciary, enforcement, etc. The transaction data may serve as big data with very high accuracy for further construction of a credit system for the entire society.

The technical solutions of the embodiments of the present application will be described in detail with reference to the accompanying drawings.

FIG. 1a is a schematic diagram of a blockchain system according to some embodiments of the present application, comprising:

a non-consensus subsystem 101, a distribution center 102, and a plurality of consensus subsystems 103;

the non-consensus subsystem 101 comprises a plurality of non-consensus nodes, and each of the consensus subsystems comprises a plurality of consensus nodes;

a non-consensus node sends a transaction request to the distribution center 102;

the distribution center 102 receives the transaction request from the non-consensus subsystem, determines a consensus subsystem corresponding to the transaction request based on the transaction data comprised in the transaction request, and forwards the transaction request to the determined consensus subsystem 103; and

the consensus subsystem 103 receives the transaction request forwarded by the distribution center, and sends the transaction request to all consensus nodes in the consensus subsystem 103 for consensus validation; if the validation is successful, generates a corresponding block according to the transaction request and stores the block into a consortium blockchain corresponding to the consensus subsystem 103.

The consensus subsystem 103 is further configured to send the block corresponding to the transaction request to the non-consensus subsystem 101; and

the non-consensus subsystem 101 receives the block and stores the block into a public blockchain corresponding to the non-consensus subsystem 101.

The consensus subsystem 103 is further configured to generate a transaction digest corresponding to the block based on the transaction data corresponding to the block, and send the transaction digest to the non-consensus subsystem 101; and

the non-consensus subsystem 101 stores the transaction digest into the public blockchain such that the transaction digest is available for search by the non-consensus nodes.

The non-consensus subsystem 101 further comprises:

a data browser 1011 configured to receive a search request for transaction data from a non-consensus node, determine search permission of the non-consensus node according to the search request; and return, according to the search permission, transaction data corresponding to the search permission to the non-consensus node.

The data browser 1011 obtains, according to the search permission, transaction data corresponding to the search permission from the consensus subsystem 103 corresponding to the search request and returns the obtained transaction data to the non-consensus node.

In an existing blockchain network, nodes are members of the blockchain network. The nodes may participate in consensus validation on a transaction request, may also search blocks stored on the blockchain network (i.e., the shared ledger), and may further search transaction data respectively corresponding to the blocks. The transaction request comprises transaction data, and a node or non-node entity may submit a transaction request to request the blockchain network to perform consensus validation on the transaction request and to verify whether transaction data of the transaction request is legitimate.

Here, the transaction data is transaction data generated by a node or non-node entity and comprises digital signature, identifier, account address, etc. of the node or non-node entity, and further comprises matters to be verified as requested by the node or non-node entity. These matters

to be verified vary depending on different transaction fields. For example, a node A transfers 500 Yuan to a node B. To make the node B to believe that the transfer has been made, node A will submit a transaction request, and the transaction request comprises the following transaction data: account address of node A, account address of node B, and “A transfers 500 Yuan to B.” Then, nodes in the blockchain network need to verify whether 500 Yuan is deducted from the account of node A, and whether 500 Yuan from the account of node A is added into the account of B.

It should be noted that, in the field of blockchain technologies, consensus validation is performed on a transaction request by all nodes according to a consensus algorithm, and the blockchain network has right to access private information of all nodes, such as accounts, transaction records, etc., for validation. The flow of consensus validation in the embodiments of the present application is not different from an existing flow of consensus validation, and therefore, the flow of consensus validation in the embodiments of the present application will not be elaborated in the specification below.

In an existing blockchain network, there is a great number of transaction requests in need of consensus validation. Therefore, consensus validation is typically performed once on a batch of transaction requests within a period of time or when the number of transaction requests as a batch reaches a number threshold, so as to improve the efficiency. Then, if the validation on this batch of transaction requests is successful, a block corresponding to this batch of transaction requests is generated and stored in the blockchain (i.e., stored in the shared ledger). Nodes may search for transaction data corresponding to a block to check whether the block has been tempered with.

In the embodiments of the present application, the consensus nodes are not different from the nodes in a consortium blockchain network in terms of functions. One consensus subsystem corresponds to a consortium blockchain network (comprising a consortium blockchain), and each consensus node corresponds to an authoritative institution for participating in consensus validation. The non-consensus nodes may be the non-node entities described in the section of “Background,” and the non-node entities may be assigned a node identity in the present system, but the non-consensus nodes may not participate in consensus validation. The non-consensus nodes may also be nodes in a public blockchain network, meaning that the non-consensus subsystem corresponds to a public blockchain network. In the application scenarios of the present application, these non-consensus nodes may not participate in consensus validation, either. In the application scenarios of the present application, consensus validation is performed by all consensus nodes in the consensus subsystems.

However, in scenarios other than the application scenarios of the present application, the non-consensus nodes may perform consensus validation. For example, the non-consensus nodes may be nodes in a bitcoin application scenario that perform consensus validation for the circulation of bitcoins according to a Proof of Work algorithm. As described above, the entire society may be

incorporated into a unified credit system on the basis of the present system. When the non-consensus subsystem corresponds to a public blockchain network, the public blockchain network just needs to interface with the present system, while the original operations of the public blockchain network will not be affected.

Furthermore, the non-consensus subsystem in the present system may further correspond to a plurality of public blockchain networks. However, in the present system, all nodes comprised in the plurality of public blockchain networks are non-consensus nodes, and the non-consensus subsystem does not care which public blockchain network these non-consensus nodes originally belong to.

In the embodiments of the present application, the distribution center provides interfacing standard protocols to the public. In one example, each consortium blockchain network may develop a client having the standard protocols built-in according to an application program interface (API) provided by the distribution center for interfacing with the distribution center and thereby becoming a consensus subsystem. In addition, any entity may interface with the distribution center and become a non-consensus node. In one example, a person or individual may install a client having the interfacing standard protocol built-in on a terminal, and then transaction requests may be submitted at any time through the client. An enterprise, in particular an enterprise that provides services to users, may interface the enterprise's application with the distribution center. When the enterprise provides a service to a user, a transaction request corresponding to the service may be submitted for a corresponding consensus subsystem to perform consensus validation.

For example, Mr. Zhang is a philanthropist who often provides financial assistance to impoverished students. Mr. Zhang is very concerned with the fate of each donation he made and whether students have truly received the donations. Then, Mr. Zhang may apply to become a non-consensus node and install a payment application with a built-in interfacing standard protocol. Every time when Mr. Zhang makes a donation, a corresponding consensus subsystem in the charity field will perform consensus validation on the donation to ensure that the donation made by Mr. Zhang is wired into an account of a designated impoverished student. Moreover, Mr. Zhang can confirm later that the transaction is not tempered with by viewing the block stored into the consortium blockchain.

For example, a client of an e-commerce platform may have a built-in interfacing standard protocol for interfacing with the distribution center. When a user buys on the e-commerce platform, the e-commerce platform requests a consensus subsystem to perform consensus validation on whether the goods purchased by the user is authentic, whether the payment made by the user is successful, etc., and provides feedback to the user.

For example, an ordinary individual may become a non-consensus node. When two non-consensus nodes conduct a transfer, one of the non-consensus nodes may initiate a transaction request to request a consortium blockchain network corresponding to the payment field to perform

consensus validation on the transfer and record a block corresponding to the transfer on a consortium blockchain.

In summary, there are abundant application scenarios under the architecture of the present system. An individual may become a non-consensus node to request validation on various events generated by the individual. An enterprise may become a non-consensus node to enhance trust that users have on the enterprise.

In the embodiments of the present application, when the non-consensus subsystem corresponds to a public blockchain network, the generated block corresponding to the transaction request may further be sent to the non-consensus subsystem, causing the non-consensus subsystem to store the received block into the public blockchain. This way, all non-consensus nodes may conveniently view time chains of transactions and do not need to request a consensus subsystem to search the blocks.

Furthermore, to further facilitate non-consensus nodes to search transaction requests that have been subject to consensus validation, transaction data corresponding to a generated block may be summarized to generate a transaction digest, and the transaction digest may be sent to the non-consensus subsystem. The non-consensus subsystem may store the transaction digest to the public blockchain. This way, the non-consensus nodes may search for the transaction digest to meet search demands in the general sense. Meanwhile, the non-consensus nodes are unable to view the completed transaction data, which prevents users with unlawful intent from using certain private data. By sending the block and transaction digest to the non-consensus subsystem for storage, it is ensured that the non-consensus subsystem does not face a risk of being invaded, while the openness of the blockchain network is achieved.

In the embodiments of the present application, the non-consensus subsystem may further comprise a data browser. The data browser has a function to provide data search and permission management capabilities for the non-consensus nodes. As described above, the block and transaction digest are sent to the non-consensus subsystem for storage. Under normal circumstances, the non-consensus nodes may learn about information such as whether consensus validation is successful, what matters have been verified, etc., by searching for the block and transaction digest stored on the public chain. In some cases, if a non-consensus node suspects that a block is tempered with, the suspicion can be confirmed only by searching for the transaction data corresponding to the block. In other cases, if a non-consensus node needs to prove the node's credit to other non-consensus nodes, the non-consensus node also needs to present some detailed transaction data to other non-consensus nodes. However, it would lead to a high risk in security if non-consensus nodes are allowed to directly view all transaction data stored on a consortium blockchain. Therefore, the data browser may perform search permission management on the non-consensus nodes.

The architecture of the blockchain system in the technical solutions according to the present

application is flexible. The data manager may be implemented not in the non-consensus subsystem, but in the distribution center or in parallel to the distribution center, the non-consensus subsystem, and the consensus subsystems. In short, the data browser may provide data search services to and manage search permission for non-consensus nodes regardless of the location of the data browser in the system.

In one example, the search permission of a non-consensus node may be determined in the following manner: for each non-consensus node, determining a type of the non-consensus node; and according to the type of the non-consensus node, assigning search permission to the non-consensus node.

Here, the type of the non-consensus node may be individual, enterprise, regulatory agency, etc., or may be different credit levels, such as high credit, medium credit, low credit, etc. In short, the type of non-consensus nodes may be decided according to actual situations, which is not limited in the present application.

For example, search permission for an enterprise-type non-consensus node may be transaction data generated by all users served by the enterprise; search permission for a regular individual may be transaction data only related to the individual; and search permission for a regulatory agency may be all transaction data.

In the embodiments of the present application, a search request for transaction data sent by a non-consensus node may carry a block, indicating that the non-consensus node wants to search for transaction data corresponding to the block; alternatively, the search request may also carry an identifier of the non-consensus node, indicating that the non-consensus node wants to search for transaction data within the search permission of the non-consensus node.

When the data browser receives the search request for transaction data sent by the non-consensus node, the data browser verifies the search permission of the non-consensus node. When the non-consensus node does not have corresponding search permission (e.g., the non-consensus node does not have permission to search for transaction data corresponding to a block), the data browser rejects the search request. If the non-consensus node has corresponding search permission, the data browser may obtain transaction data corresponding to the search permission from the consensus subsystem corresponding to the search request according to the search permission and return the obtained transaction data to the non-consensus node.

Furthermore, the data browser may determine what transaction data may be searched by the non-consensus node according to the determined search permission, and then obtain the transaction data from a corresponding consensus subsystem; or may directly send the determined search permission to the corresponding consensus subsystem for the consensus subsystem to return corresponding transaction data according to the search permission.

In the blockchain system shown in FIG. 1a, a blockchain system comprising a distribution

center, a non-consensus subsystem, and a plurality of consensus subsystems is created, wherein the non-consensus subsystem comprises a plurality of non-consensus nodes, each of the consensus subsystems comprises a plurality of consensus nodes, and each of the consensus subsystems corresponds to a consortium blockchain network containing the plurality of consensus nodes. As a result, the consensus subsystems can perform consensus validations in different transaction fields. This way, the consensus nodes in the consensus subsystems are responsible for consensus validation, and the non-consensus nodes in the non-consensus subsystem may send a transaction request to the distribution center. The distribution center determines a consensus subsystem in a particular transaction field based on the transaction data comprised in the transaction request. Then, the consensus subsystem performs consensus validation on the transaction request. According to the embodiments of the present application, on one hand, only the consensus nodes are responsible for consensus validation, and non-consensus nodes outside the consensus subsystems cannot participate in consensus validation by a consortium blockchain network, thereby improving the security of the blockchain network; on the other hand, the distribution center may interface with the consortium blockchain networks, and non-consensus entities (non-consensus nodes) outside the consortium blockchain network only need to interface with the distribution center, and do not need to interface with a plurality of consortium blockchain networks according to different interfacing protocols, thereby improving the convenience of the blockchain network.

In addition, it should be stressed that the functions of the distribution center may be limited only to determining a corresponding consensus subsystem according to a transaction request or may serve as an agent of data interaction between the consensus subsystem and the non-consensus subsystem. In other words, the consensus subsystem and the non-consensus subsystem may perform data interaction (e.g., sending a block, a transaction digest, transaction data, etc.) without passing through the distribution center but based on a particular routing protocol, as shown in FIG. 1b.

FIG. 2 is a flow chart of a data storing method according to some embodiments of the present application, comprising the following steps:

S201: sending, by the non-consensus subsystem, a transaction request to the distribution center.

These embodiments of the present application are based on the same invention concept as the blockchain system shown in FIG. 1a. Explanations of the blockchain system shown in FIG. 1a may be referenced for explanations of relevant concepts.

In the embodiments of the present application, it may be a non-consensus node in the non-consensus subsystem that sends the transaction request to the distribution center.

S202: determining, by the distribution center, a consensus subsystem corresponding to the transaction request.

S203: forwarding, by the distribution center, the transaction request to the determined

consensus subsystem.

The distribution center may receive transactions sent by a plurality of non-consensus nodes for a number of times, and for each transaction request, forward the transaction request to a corresponding consensus subsystem for consensus validation.

The distribution center may serve as an agent of data interaction between the consensus subsystem and the non-consensus subsystem (e.g., sending a block, a transaction digest, transaction data, etc. in subsequent steps), or may not serve as an agent but let the consensus subsystem and the non-consensus subsystem perform data interaction directly.

S204: performing, by the consensus subsystem, consensus validation on the transaction request.

The consensus subsystem performs consensus validation on the transaction request, which is practically to send the transaction request to all consensus nodes comprised in the consensus subsystem to perform consensus validation. A consensus algorithm based on which the consensus nodes perform consensus validation may be the Byzantine Fault Tolerance algorithm or other consensus algorithms, which is not limited.

S205: after the validation is successful, the consensus subsystem generates a corresponding block according to the transaction request, stores the block in a consortium blockchain corresponding to the consensus subsystem, and generates a transaction digest corresponding to the block.

After the validation is successful, the consensus subsystem generates a corresponding block according to the transaction request. In practice, the corresponding block is generated according to a batch of transaction requests that comprise the transaction request, which has been explained above and will not be elaborated.

S206: sending, by the consensus subsystem, the block corresponding to the transaction request and the transaction digest to the non-consensus subsystem.

S207: storing, by the non-consensus subsystem, the block into a corresponding public blockchain.

S208: sending, by the non-consensus subsystem, an obtaining request for the transaction data to the consensus subsystem.

In the embodiments of the present application, it may be the non-consensus subsystem that sends the obtaining request to the consensus subsystem, and in one example, it may be the data browser comprised in the non-consensus subsystem that sends the obtaining request to the consensus subsystem.

When the obtaining request comprises search permission of a non-consensus node, the consensus subsystem may determine, according to the search permission, transaction data corresponding to the search permission from transaction data stored in the non-consensus

subsystem, wherein the search permission is determined by the non-consensus subsystem according to a search request for the transaction data sent by the non-consensus node.

When the obtaining request comprises a list of identifiers of transaction data, the consensus subsystem may return corresponding transaction data to the non-consensus subsystem according to the list of identifiers.

S209: returning transaction data corresponding to the obtaining request to the non-consensus subsystem.

With the data storing method shown in FIG. 2, non-consensus nodes outside the consensus subsystem may be prevented from freely searching for transaction data, while transaction requests submitted by the non-consensus nodes may be allocated in a unified manner, thereby improving the convenience of the blockchain network.

Based on the data storing method shown in FIG. 2, the embodiments of the present application further provide a corresponding data storing apparatus. As shown in FIG. 3, a blockchain system comprises a distribution center, a non-consensus subsystem, and a plurality of pieces of the apparatus, wherein the non-consensus subsystem comprises a plurality of non-consensus nodes, and each piece of the apparatus comprises a plurality of consensus nodes, the apparatus comprising:

a receiving module 301 configured to receive a transaction request forwarded by the distribution center, the transaction request comprising transaction data;

a validation module 302 configured to send the transaction request to all consensus nodes in the consensus subsystem for consensus validation; and

a storing module 303 configured to, if the validation is successful, generate a corresponding block according to the transaction request and store the block into a consortium blockchain corresponding to the consensus subsystem.

The apparatus further comprises: a sending module 304 configured to, if the validation is successful, send the block corresponding to the transaction request to the non-consensus subsystem, causing the non-consensus subsystem to store the block into a public blockchain corresponding to the non-consensus subsystem.

The apparatus further comprises: a generating module 305 configured to, if the consensus nodes reach a consensus that the transaction request is legitimate, generate a transaction digest corresponding to the block based on the transaction data corresponding to the block, and send the transaction digest to the non-consensus subsystem, causing the non-consensus subsystem to store the transaction digest into the public blockchain such that the transaction digest is available for search by the non-consensus nodes.

The apparatus further comprises: a transaction data management module 306 configured to receive an obtaining request for transaction data from the non-consensus subsystem, and return, according to the obtaining request, transaction data corresponding to the obtaining request to the

non-consensus subsystem.

The obtaining request comprises search permission of a non-consensus node, and the search permission is determined by the non-consensus subsystem according to a search request for the transaction data sent by the non-consensus node; and

the transaction data management module 306 determines, according to the search permission, transaction data corresponding to the search permission from transaction data stored in the non-consensus subsystem.

Based on the data storing method shown in FIG. 2, the embodiments of the present application further provide a corresponding data storing apparatus. As shown in FIG. 4, a blockchain system comprises the apparatus, a non-consensus subsystem, and a plurality of consensus subsystems, the non-consensus subsystem comprises a plurality of non-consensus nodes, and each of the consensus subsystems comprises a plurality of consensus nodes, the apparatus comprising:

a receiving module 401 configured to receive a transaction request sent by the non-consensus subsystem, the transaction request comprising transaction data;

a determining module 402 configured to determine a consensus subsystem corresponding to the transaction request based on the transaction data comprised in the transaction request; and

a forwarding module 403 configured to forward the transaction request to the determined consensus subsystem, causing the consensus subsystem to perform consensus validation on the transaction request, and store a block corresponding to the validated transaction request into a consortium blockchain corresponding to the consensus subsystem.

Based on the data storing method shown in FIG. 2, the embodiments of the present application further provide a corresponding data storing apparatus. As shown in FIG. 5, a blockchain system comprises a distribution center, the apparatus, and a plurality of consensus subsystems, the apparatus comprises a plurality of non-consensus nodes, and each of the consensus subsystems comprises a plurality of consensus nodes, the apparatus comprising:

a sending module 501 configured to send a transaction request to the distribution center, the transaction request comprising transaction data, causing the distribution center to forward, based on the transaction data, the transaction request to a consensus subsystem corresponding to the transaction data.

The apparatus further comprises: a first storing module 502 configured to receive a block from the consensus subsystem and store the block into a public blockchain corresponding to the non-consensus subsystem.

The apparatus further comprises: a second storing module 503 configured to receive a transaction digest corresponding to the block from the consensus subsystem, and store the transaction digest into the public blockchain such that the transaction digest is available for search by the non-consensus nodes.

The apparatus further comprises: a search module 504 configured to receive a search request for transaction data from a non-consensus node, determine search permission of the non-consensus node according to the search request, and return, according to the search permission, transaction data corresponding to the search permission to the non-consensus node.

The search permission of a non-consensus node may be determined in the following manner:
for each non-consensus node, determining a type of the non-consensus node; and
according to the type of the non-consensus node, assigning search permission to the non-consensus node.

The search module 504 obtains, according to the search permission, transaction data corresponding to the search permission from the consensus subsystem corresponding to the search request and returns the obtained transaction data to the non-consensus node.

In the 1990s, an improvement to a technology can be obviously differentiated into a hardware improvement (e.g., an improvement to a circuit structure, such as a diode, a transistor, a switch, etc.) or a software improvement (an improvement to a flow of a method). With the technological development, however, many current improvements to method flows may be deemed as direct improvements to hardware circuit structures. Designers almost always obtain a corresponding hardware circuit structure by programming an improved method flow into a hardware circuit. Therefore, it cannot be concluded that an improvement to a method flow cannot be realized with a hardware module. For example, Programmable Logic Device (PLD) (e.g., Field Programmable Gate Array (FPGA)) is such integrated circuit that the integrated circuit's logic functions are determined by a user through programming the device. A designer programs on his/her own to "integrate" a digital system onto one piece of PLD, who does not need to ask a chip manufacturer to design and manufacture a dedicated IC chip. At present, moreover, this type of programming has mostly been implemented through "logic compiler" software, rather than manually manufacturing the IC chips. The logic compiler software is similar to a software compiler used for program development and writing, while a particular programming language must be used for writing source codes prior to compiling, which is referred to as a Hardware Description Language (HDL). There is not just one, but many types of HDL, such as ABEL (Advanced Boolean Expression Language), AHDL (Altera Hardware Description Language), Confluence, CUPL (Cornell University Programming Language), HDCal, JHDL (Java Hardware Description Language), Lava, Lola, MyHDL, PALASM, RHDL (Ruby Hardware Description Language), etc. The most commonly used right now includes VHDL (Very-High-Speed Integrated Circuit Hardware Description Language) and Verilog. One of ordinary skill in the art should also be aware that it would be very easy to obtain a hardware circuit to implement a logic method flow by using the above HDLs to carry out slight logic programming on the method flow and program the method flow into an IC.

A controller may be implemented in any proper manner. For example, a controller may be in,

for example, a form of a microprocessor or processor, as well as a computer readable medium that stores computer readable program codes (e.g., software or firmware) capable of being executed by the (micro)processor, a logic gate, a switch, an Application Specific Integrated Circuit (ASIC), a programmable logic controller, and an embedded microcontroller. Examples of the controller include, but are not limited to, the following microcontrollers: ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20, and Silicone Labs C8051F320. A memory controller may further be implemented as a part of a control logic of a memory. One of ordinary skill in the art should also be aware that, in addition to that a controller is implemented in a manner of pure computer readable program codes, it is totally feasible to perform logic programming on steps of a method to enable a controller to implement the same functions in a form of a logic gate, a switch, an ASIC, a programmable logic controller, and an embedded microcontroller. Therefore, such controller can be deemed as a hardware part, while devices comprised in the controller and configured to achieve various functions may also be deemed as a structure inside the hardware part. Alternatively, devices configured to achieve various functions may even be deemed as both software modules to implement a method and a structure inside a hardware part.

Technical carriers involved in the payment in the embodiments of the present application may comprise, for example, Near Field Communication (NFC), WIFI, 3G/4G/5G, POS terminal card swiping technologies, QR code scanning technologies, barcode scanning technologies, Bluetooth, IR, Short Message Service (SMS), Multimedia Message Service (MMS), etc.

The system, apparatus, module, or unit described in the above embodiments may be implemented by a computer chip or entity, or implemented by a product having a function. A typical implementation device is a computer. In one example, a computer may be, for example, a personal computer, a laptop computer, a cellular phone, a camera phone, a smart phone, a personal digital assistant, a medium player, a navigation device, an email device, a game console, a tablet computer, a wearable device, or a combination of any devices in these devices.

For convenience of description, the above device is divided into various units according to functions for description. Functions of the units may be implemented in one or more pieces of software and/or hardware when the present application is implemented.

One of ordinary skill in the art should understand that the embodiments of the present invention may be provided as a method, a system, or a computer program product. Therefore, the present invention may be implemented as a complete hardware embodiment, a complete software embodiment, or an embodiment combining software and hardware. Moreover, the present invention may be in the form of a computer program product implemented on one or more computer usable storage media (including, but not limited to, a magnetic disk memory, CD-ROM, an optical memory, etc.) comprising computer usable program codes.

The present invention is described with reference to flowcharts and/or block diagrams of the

method, device (system), and computer program product according to the embodiments of the present invention. It should be understood that a computer program instruction may be used to implement each process and/or block in the flowcharts and/or block diagrams and a combination of processes and/or blocks in the flowcharts and/or block diagrams. These computer program instructions may be provided for a general-purpose computer, a special-purpose computer, an embedded processor, or a processor of other programmable data processing devices to generate a machine, causing the instructions executed by a computer or a processor of other programmable data processing devices to generate an apparatus for implementing a function specified in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

These computer program instructions may also be stored in a computer readable memory that can instruct a computer or other programmable data processing devices to work in a particular manner, causing the instructions stored in the computer readable memory to generate a manufactured article that includes an instruction apparatus. The instruction apparatus implements a function specified in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

These computer program instructions may also be loaded onto a computer or other programmable data processing devices, causing a series of operational steps to be performed on the computer or other programmable devices, thereby generating computer-implemented processing. Therefore, the instructions executed on the computer or other programmable devices provide steps for implementing a function specified in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

In a typical configuration, the computation device includes one or more processors (CPUs), input/output interfaces, network interfaces, and a memory.

The memory may include computer readable media, such as a volatile memory, a Random Access Memory (RAM), and/or a non-volatile memory, e.g., a Read-Only Memory (ROM) or a flash RAM. The memory is an example of a computer readable medium.

Computer readable media include non-volatile, volatile, mobile, and immobile media, which can implement information storage through any method or technology. The information may be computer readable instructions, data structures, program modules, or other data. Examples of storage media of computers include, but are not limited to, Phase-change Random Access Memories (PRAMs), Static Random Access Memories (SRAMs), Dynamic Random Access Memories (DRAMs), other types of Random Access Memories (RAMs), Read-Only Memories (ROMs), Electrically Erasable Programmable Read-Only Memories (EEPROMs), flash memories or other memory technologies, Compact Disk Read-Only Memories (CD-ROMs), Digital Versatile Discs (DVDs) or other optical memories, cassettes, cassette and disk memories or other magnetic memory devices, or any other non-transmission media, which can be used for storing information

accessible to a computation device. According to the definitions in the specification, the computer readable media do not include transitory media, such as modulated data signals and carriers.

It should be further noted that the terms of “including,” “comprising,” or any other variants of the terms are intended to encompass a non-exclusive inclusion, causing a process, method, commodity, or device comprising a series of elements to not only comprise these elements, but also comprise other elements that are not clearly listed, or further comprise elements that are inherent to the process, method, commodity, or device. When there is no further restriction, elements defined by the statement “comprising one...” does not exclude that a process, method, commodity, or device comprising the above elements further comprises additional identical elements.

One of ordinary skill in the art should understand that the embodiments of the present application may be provided as a method, a system, or a computer program product. Therefore, the present application may be implemented as a complete hardware embodiment, a complete software embodiment, or an embodiment combining software and hardware. Moreover, the present application may be in the form of a computer program product implemented on one or more computer usable storage media (including, but not limited to, a magnetic disk memory, CD-ROM, an optical memory, etc.) comprising computer usable program codes.

The present application may be described in a regular context of a computer executable instruction that is executed by a computer, such as a program module. Generally, the program module comprises a routine, a program, an object, a component, a data structure, etc. for executing a particular task or implementing a particular abstract data type. The present application may also be practiced in distributed computing environments. In these distributed computing environments, remote processing devices connected via communication networks carry out tasks. In the distributed computing environments, a program module may be located in local and remote computer storage media, including storage devices.

The embodiments in this specification are described in a progressive manner with each embodiment focused on differences from other embodiments, and the embodiments may be mutually referenced for identical or similar parts. In particular, the system embodiment is described in a relatively simple manner, as the system embodiment is substantially similar to the method embodiment. The description of the method embodiment may be referenced for the related parts.

The above-described is only embodiments of the present application, which are not used to limit the present application. To one of ordinary skill in the art, the present application may have various modifications and changes. Any modification, equivalent substitution, or improvement made within the spirit and principle of the present application shall be encompassed by the claims of the present application.

CLAIMS

1. A method comprising:

receiving a transaction request from a first node, the transaction request comprising transaction data;

determining, based on the transaction data, a blockchain network corresponding to the transaction request from a plurality of blockchain networks, wherein the first node is excluded from a consensus process associated with the determined blockchain network; and

forwarding the transaction request to the determined blockchain network.

2. The method of Claim 1, further comprising:

receiving, from the determined blockchain network, a block generated based on consensus validation on the transaction request; and

forwarding the block to the first node.

3. The method of Claim 1, wherein:

the determined blockchain network comprises a consortium blockchain network;

the consortium blockchain network comprises a plurality of nodes; and

the first node is not among the plurality of nodes of the consortium blockchain.

4. The method of Claim 1, further comprising:

performing, by the determined blockchain network, the consensus validation on the transaction request;

generating, by the determined blockchain network, the block associated with the transaction request; and

adding, by the determined blockchain network, the generated block to a blockchain associated with the determined blockchain network.

5. The method of Claim 1, further comprising:

storing, by the first node, information associated with the block to a public blockchain.

6. The method of Claim 1, further comprising:

receiving a transaction digest corresponding to the block from the determined

blockchain network; and

forwarding the transaction digest to the first node.

7. The method of Claim 6, further comprising:

storing, by the first node, information associated with the transaction digest to a public blockchain.

8. The method of Claim 1, further comprising:

receiving a search request for transaction data from the first node;

determining a search permission associated with the first node based on the search request;

obtaining, from the determined blockchain network, transaction data corresponding to the search permission; and

sending the obtained transaction data to the first node.

9. The method of Claim 8, wherein the determining a search permission associated with the first node comprises:

determining a type of the first node; and

assigning the search permission to the first node based on the determined type.

10. The method of Claim 8, wherein the search permission associated with the first node comprises:

permission to access transaction data associated with the first node;

permission to access transaction data associated with a plurality of second nodes, wherein the second nodes respectively correspond to a plurality of users of an entity corresponding to the first node; or

permission to access all transaction data associated with the determined blockchain network.

11. A system comprising a processor and a non-transitory computer-readable storage medium storing instructions executable by the processor to cause the system to perform operations comprising:

receiving a transaction request from a first node, the transaction request comprising transaction data;

determining, based on the transaction data, a blockchain network corresponding to the transaction request from a plurality of blockchain networks, wherein the first node is excluded from a consensus process associated with the determined blockchain network; and
forwarding the transaction request to the determined blockchain network.

12. The system of Claim 11, wherein the operations further comprise:

receiving, from the determined blockchain network, a block generated based on consensus validation on the transaction request; and
forwarding the block to the first node.

13. The system of Claim 11, wherein:

the determined blockchain network comprises a consortium blockchain network;
the consortium blockchain network comprises a plurality of nodes; and
the first node is not among the plurality of nodes of the consortium blockchain.

14. The system of Claim 11, wherein the operations further comprise:

performing, by the determined blockchain network, the consensus validation on the transaction request;
generating, by the determined blockchain network, the block associated with the transaction request; and
adding, by the determined blockchain network, the generated block to a blockchain associated with the determined blockchain network.

15. The system of Claim 11, wherein the operations further comprise:

storing, by the first node, information associated with the block to a public blockchain.

16. The system of Claim 11, wherein the operations further comprise:

receiving a transaction digest corresponding to the block from the determined blockchain network; and
forwarding the transaction digest to the first node.

17. The system of Claim 16, wherein the operations further comprise:
storing, by the first node, information associated with the transaction digest to a public blockchain.

18. The system of Claim 11, wherein the operations further comprise:
receiving a search request for transaction data from the first node;
determining a search permission associated with the first node based on the search request;
obtaining, from the determined blockchain network, transaction data corresponding to the search permission; and
sending the obtained transaction data to the first node.

19. The system of Claim 18, wherein the determining a search permission associated with the first node comprises:
determining a type of the first node; and
assigning the search permission to the first node based on the determined type.

20. A non-transitory computer-readable storage medium, configured with instructions executable by one or more processors to cause the one or more processors to perform operations comprising:
receiving a transaction request from a first node, the transaction request comprising transaction data;
determining, based on the transaction data, a blockchain network corresponding to the transaction request from a plurality of blockchain networks, wherein the first node is excluded from a consensus process associated with the determined blockchain network; and
forwarding the transaction request to the determined blockchain network.