



(12) **DEMANDE DE BREVET CANADIEN**
CANADIAN PATENT APPLICATION

(13) **A1**

(86) Date de dépôt PCT/PCT Filing Date: 2018/05/10
(87) Date publication PCT/PCT Publication Date: 2018/11/15
(85) Entrée phase nationale/National Entry: 2019/05/10
(86) N° demande PCT/PCT Application No.: CN 2018/086280
(87) N° publication PCT/PCT Publication No.: 2018/205971
(30) Priorité/Priority: 2017/05/12 (CN201710335973.4)

(51) Cl.Int./Int.Cl. *H04L 29/06* (2006.01)
(71) Demandeur/Applicant:
ALIBABA GROUP HOLDING LIMITED, KY
(72) Inventeur/Inventor:
QIU, HONGLIN, CN...
(74) Agent: KIRBY EADES GALE BAKER

(54) Titre : PROCEDE ET DISPOSITIF DE TRAITEMENT DE DONNEES A BASE DE CHAINE DE BLOCS
(54) Title: BLOCKCHAIN-BASED DATA PROCESSING METHOD AND DEVICE

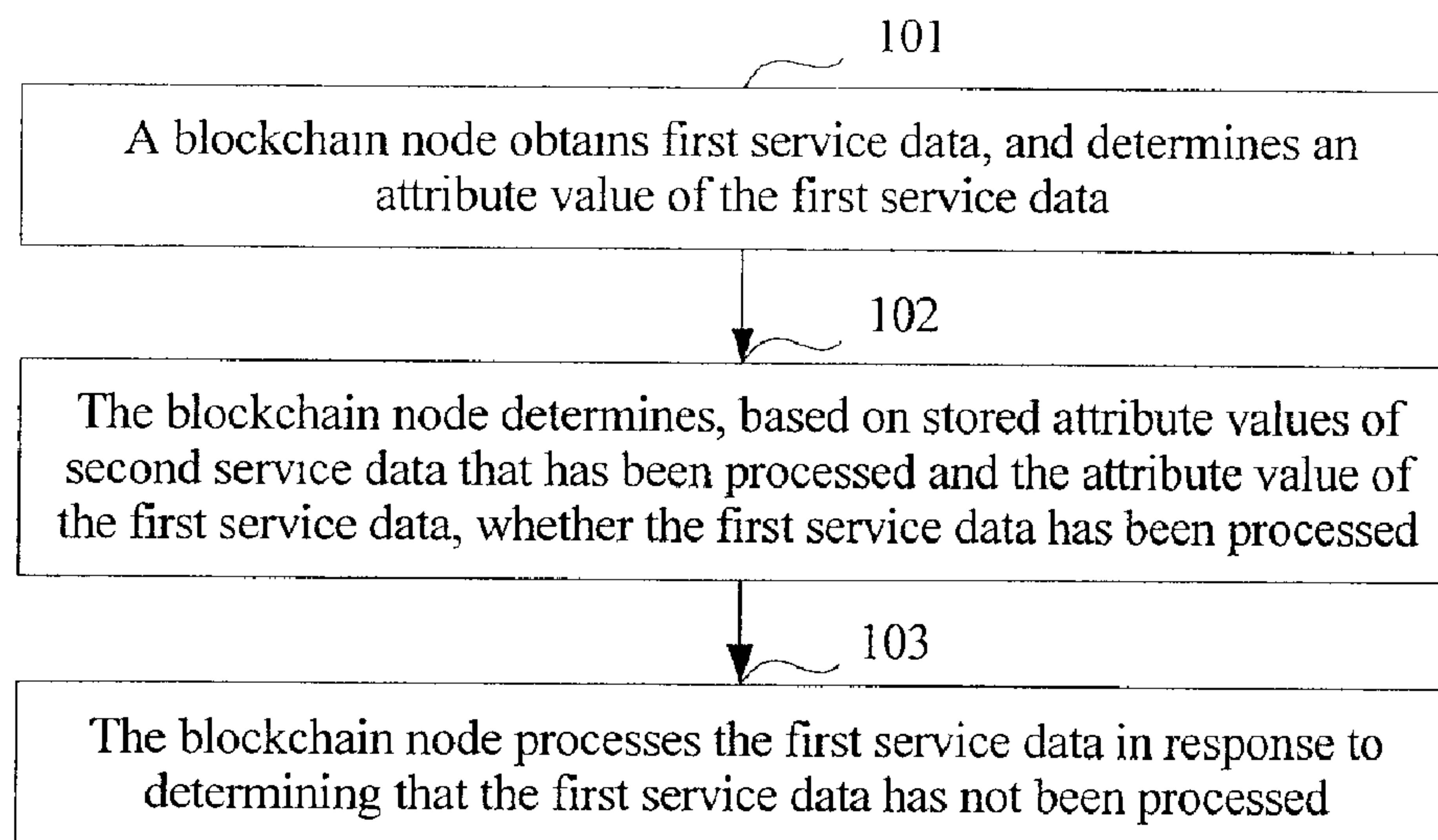


FIG. 1

(57) Abrégé/Abstract:

Disclosed in the present application is a blockchain-based data processing method and device. The method comprises: comparing an attribute value characterizing uniqueness of transaction data to be processed and an attribute value of already-accepted

(57) Abrégé(suite)/Abstract(continued):

transaction data stored by a blockchain node so as to determine whether the transaction data to be processed has been accepted; and only if not, initiating processing on the transaction data to be processed. The method of the present application can effectively prevent replay attacks, and can also prevent a problem in which, since different transaction data items have different attribute values, when one transaction data item is being processed other transaction data items cannot be processed due to locking. In this way, processing efficiency of transaction data over a blockchain network is effectively guaranteed, and throughput of the transaction data over the blockchain network is improved.

ABSTRACT

The present application discloses a blockchain-based data processing method and device, including the following: comparing an attribute value that can represent uniqueness of service data to be processed with an attribute value, stored in a blockchain node, of service data that has been processed, to determine whether the service data to be processed has been processed; and when it is determined that the service data to be processed has not been processed, starting processing of the service data to be processed. Therefore, a replay attack can be effectively prevented. In addition, because attribute values of different service data are different, a problem that other service data cannot be processed due to the locking of one service data that is being processed can be avoided. Therefore, service data processing efficiency in a blockchain network is effectively ensured, and an overall service data throughput in the blockchain network is increased.

BLOCKCHAIN-BASED DATA PROCESSING METHOD AND DEVICE

TECHNICAL FIELD

[0001] The present application relates to the field of Internet information processing technologies and computer technologies, and in particular, to a blockchain-based data processing method and device.

BACKGROUND

[0002] A blockchain technology is also referred to as a distributed ledger technology, and is a distributed Internet database technology. A network constructed based on the blockchain technology can be referred to as a blockchain network, and the blockchain network includes network nodes (which can also be referred to as blockchain nodes, and are referred to as nodes for short in the following). Each node corresponds to at least one blockchain, and each blockchain includes at least one block. The blockchain network (or the blockchain technology) is decentralized, transparent, trustable, and cannot be tampered with. Based on these characteristics, the blockchain technology is more and more widely applied.

[0003] With the development of blockchain technology, a replay attack technology can occur in practice. The replay attack technology means that a malicious user illegally obtains service data that is being processed by a blockchain network or service data that has been processed by a blockchain network, and resends the service data to the blockchain network, so that the blockchain network reprocesses the service data. For example, assume that service data obtained by using the replay attack technology is "transferring 100 yuan from an account A to an account B". If the account A has 300 yuan, when service data sent by a user corresponding to the account A is received, the transaction data is processed, and then the account A has 200 yuan left. Because the malicious user obtains the service data by using the replay attack technology, the blockchain network also processes the service data after receiving the service data sent by the malicious user, and then the A account has 100 yuan left. It can be seen that the

replay attack technology threatens data security of the blockchain network.

[0004] To prevent a replay attack, Ethereum configures a random number for each account. When service data is generated in the account, the random number is automatically increased by 1 to obtain a value, and both the value and the service data are sent to a blockchain node. Therefore, when receiving the service data, the blockchain node can perform verification on the service data by using the value. The blockchain node processes the service data in response to determining that the service data is not a repeated transaction; otherwise, discards processing of the service data.

[0005] However, to ensure that a random number can continuously increase with generated service data, when one of two pieces of service data generated in a same account is sent to the blockchain node, the random number needs to be locked, and the random number can be unlocked when the service data is processed. In other words, when such the method is used to prevent the replay attack, the service data is processed in a serial mode, and consequently service data processing efficiency in the blockchain network is relatively low.

SUMMARY

[0006] In view of this, implementations of the present application provide a blockchain-based data processing method and device, so as to alleviate a problem of how to prevent a replay attack in a blockchain network to improve an overall throughput of blockchain transaction data processing.

[0007] The following technical solutions are used in the implementations of the present application:

[0008] An implementation of the present application provides a blockchain-based data processing method, including the following: obtaining, by a blockchain node, first service data, and determining an attribute value of the first service data, where the attribute value is used to represent uniqueness of the first service data; determining, by the blockchain node based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed; and processing, by the blockchain node, the first service data in response to determining that the first service data has not been processed.

[0009] An implementation of the present application further provides a blockchain-based data processing device, including the following: an acquisition unit, configured

to obtain first service data, and determine an attribute value of the first service data, where the attribute value is used to represent uniqueness of the first service data; a determining unit, configured to determine, based on stored attribute values of second service data that has been processed and the attribute value of the first service data, 5 whether the first service data has been processed; and a processing unit, configured to process the first service data in response to determining that the first service data has not been processed.

[0010] The previously described at least one technical solution used in the implementations of the present application can achieve the following beneficial effects: 10 [0011] In the implementations of the present application, an attribute value that can represent uniqueness of service data to be processed is compared with an attribute value, stored in a blockchain node, of service data that has been processed, to determine whether the service data to be processed has been processed; and when it is determined that the service data to be processed has not been processed, processing of the service 15 data to be processed is started. Therefore, a replay attack can be effectively prevented. In addition, because attribute values of different service data are different, a problem that other service data cannot be processed due to the locking of one service data that is being processed can be avoided. Therefore, service data processing efficiency in a blockchain network is effectively ensured, and an overall service data throughput in the 20 blockchain network is increased.

BRIEF DESCRIPTION OF DRAWINGS

[0012] The accompanying drawings described here are intended to provide a further understanding of the present application, and constitute a part of the present application. The illustrative implementations of the present application and descriptions 25 of the implementations of the present application are intended to describe the present application, and do not constitute limitations on the present application. In the accompanying drawings:

[0013] FIG. 1 is a schematic flowchart illustrating a blockchain-based data processing method, according to an implementation of the present application; 30 [0014] FIG. 2 is a schematic flowchart illustrating a blockchain-based data processing method, according to an implementation of the present application; and [0015] FIG. 3 is a schematic structural diagram illustrating a blockchain-based data

processing device, according to an implementation of the present application.

DESCRIPTION OF IMPLEMENTATIONS

[0016] To make the objectives, technical solutions, and advantages of the present application clearer, the following clearly and comprehensively describes the technical solutions of the present application with reference to specific implementations of the present application and corresponding accompanying drawings. Apparently, the described implementations are some rather than all of the implementations of the present application. All other implementations obtained by a person of ordinary skill in the art based on the implementations of the present application without creative efforts shall fall within the protection scope of the present application.

[0017] The following describes in detail the technical solutions provided in the implementations of the present application with reference to the accompanying drawings.

[0018] FIG. 1 is a schematic flowchart illustrating a blockchain-based data processing method, according to an implementation of the present application. The method can be described as follows:

[0019] Step 101: A blockchain node obtains first service data, and determines an attribute value of the first service data.

[0020] The attribute value is used to represent uniqueness of the first service data.

[0021] In this implementation of the present application, when receiving a service processing request sent by another device, the blockchain node can obtain service data from the service processing request. The service data here can be considered as the first service data.

[0022] In addition, before triggering a consensus operation, the blockchain node can obtain a predetermined amount of service data from a plurality of pieces of stored service data. The service data here can be considered as a plurality of pieces of first service data. A specific implementation of obtaining the first service data is not limited here.

[0023] It is worthwhile to note that the blockchain node can be a processing node of service data, or can be a non-processing node of service data. The processing node can be understood as a node that directly receives the service data from another device, and the non-processing node can be understood as a node that receives the service data

from another blockchain node through broadcast. In other words, for one piece of service data, a blockchain network possibly includes one processing node and a plurality of non-processing nodes.

[0024] When obtaining the first service data, the blockchain node can determine 5 the attribute value of the first service data. The attribute value here is obtained when the service data is generated. If the service data is transaction data, when the transaction data is generated, a serial number of the transaction data is also generated, and the serial number can be used as an attribute value of the transaction data, namely, the attribute value described in this implementation of the present application. In addition, when the 10 service data is generated, the generated transaction data can be further calculated by using a predetermined algorithm, and an obtained calculation result can be considered as the attribute value of the service data. For example, the generated service data is calculated by using a hash algorithm, and an obtained hash value is the attribute value of the service data.

15 [0025] The attribute value described in this implementation of the present application includes one or more of a hash value of the service data and a serial number of the service data.

[0026] It is worthwhile to note that in this implementation of the present application, because the attribute value can uniquely identify the service data, the attribute value 20 can be used as a condition for identifying whether the service data is unique.

[0027] To ensure security of the attribute value of the service data, the attribute value can be encrypted. This is not specifically limited here.

[0028] Step 102: The blockchain node determines, based on stored attribute values 25 of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed.

[0029] In this implementation of the present application, when processing service data, the blockchain node stores an attribute value of service data that has been processed. That is, in the technical solution provided in this implementation of the present application, the blockchain node needs to maintain a database, and the database 30 stores an attribute value of service data that has been processed by the blockchain node. As such, when a replay attack occurs, whether service data to be processed is repeatedly processed can be determined by using the attribute value, stored in the database, of the service data that has been processed. In addition, a problem in the existing technology that other service data cannot be processed due to the locking of one service data that

is being processed can be avoided. Although it is slightly redundant in space, system performance and processing time can be improved. Therefore, service data processing efficiency in a blockchain network is effectively ensured, and an overall service data throughput in the blockchain network is increased.

5 [0030] Specifically, the blockchain node queries, in the stored attribute values of the second service data that has been processed, for whether there is an attribute value that is the same as the attribute value of the first service data.

[0031] The blockchain node determines, based on a query result, whether the first service data has been processed.

10 [0032] For example, the blockchain node compares the stored attribute values of the second service data that has been processed with the attribute value of the first service data, to determine whether there is an attribute value that is the same as the attribute value of the first service data in the attribute values of the second service data that has been processed.

15 [0033] If there is an attribute value that is the same as the attribute value of the first service data in the attribute values of the second service data that has been processed, it indicates that the first service data has been processed, and it can be further determined that the first service data is invalid service data, and is possibly a replay attack. If there is no attribute value that is the same as the attribute value of the first 20 service data in the attribute values of the second service data that has been processed, it indicates that the first service data has not been processed.

25 [0034] It is worthwhile to note that in a replay attack technology, when service data (which is subsequently referred to as valid service data) is intercepted or stolen, the service data is replicated, to generate other service data (which is subsequently referred to as invalid service data) that is the same as the intercepted service data. It indicates that an attribute value of the valid service data is the same as that of the invalid service data.

30 [0035] In addition, "first" and "second" in "first service data" and "second service data" described in this implementation of the present application indicate no special meaning, and are merely used to distinguish between different service data.

[0036] Step 103: The blockchain node processes the first service data in response to determining that the first service data has not been processed.

[0037] In this implementation of the present application, once determining that the first service data has not been processed, the blockchain node can process the first

service data, in other words, perform consensus processing, storage processing, etc. on the service data.

[0038] If the blockchain node is a processing node of the first service data, in response to determining that the first service data has not been processed, the 5 blockchain node can further broadcast the first service data to another blockchain node in the blockchain network by using a method in the existing technology after processing the first service data.

[0039] Specifically, the determining, by the blockchain node, that the first service data has not been processed includes the following: determining, by the blockchain 10 node, that the first service data has not been processed in response to determining that no attribute value is found to be the same as the attribute value of the first service data from the stored attribute values of the second service data that has been processed.

[0040] Preferably, in this implementation of the present application, the method further includes the following: storing, by the blockchain node, the attribute value of 15 the first service data.

[0041] As described above, the blockchain node can maintain a database (which is subsequently referred to as a processed database), and the processed database stores attribute values of service data that has been processed. The blockchain node can store the attribute value of the first service data in the processed database.

20 [0042] It is worthwhile to note that the processed database described in this implementation of the present application can be a relational database, or can be a Key-Value database. This is not specifically limited here.

[0043] Preferably, when storing the attribute value of the first service data in the processed database, the blockchain node can further determine a query index for the 25 attribute value of the first service data, and further establish a mapping relationship between the query index and the attribute value of the first service data. This helps to improve execution efficiency of step 102, and further improve service data processing performance in the entire system.

[0044] Preferably, in this implementation of the present application, the method further includes the following: discarding, by the blockchain node, processing of the 30 first service data in response to determining that the first service data has been processed.

[0045] It is worthwhile to note that the technical solution in this implementation of the present application can be triggered and executed when the blockchain node (which

refers to a processing node here) receives a service processing request, or can be executed before a consensus is triggered. A real-time timing is not specifically limited here, and can be determined based on an actual need.

[0046] If the previously described solution is executed before the consensus is triggered, it means that whether received service data has been processed needs to be identified before the consensus is started regardless of a processing node or a non-processing node, and the previously described processed database can be accessed by any blockchain node in the blockchain network, in other words, is a database that is shared in the blockchain network. Moreover, if the blockchain node is a processing node, in response to determining that the first service data has been processed, the blockchain node broadcasts information that the first service data has been processed to another blockchain node, so that the another blockchain node can also discard processing of the first service data. The processing here can be consensus processing, storage processing, etc.

[0047] According to the technical solution in this implementation of the present application, an attribute value that can represent uniqueness of service data to be processed is compared with an attribute value, stored in a blockchain node, of service data that has been processed, to determine whether the service data to be processed has been processed; and when it is determined that the service data to be processed has not been processed, processing of the service data to be processed is started. Therefore, a replay attack can be effectively prevented. In addition, because attribute values of different service data are different, a problem that other service data cannot be processed due to the locking of one service data that is being processed can be avoided. Therefore, service data processing efficiency in a blockchain network is effectively ensured, and an overall service data throughput in the blockchain network is increased.

[0048] Based on the same inventive concept, FIG. 2 is a schematic flowchart illustrating a blockchain-based data processing method, according to an implementation of the present application. The method can be described as follows. In this implementation of the present application, an example in which execution of this solution is triggered when a blockchain node receives a service processing request is used for description.

[0049] Step 201: A blockchain node receives a service processing request, and obtains service data from the service processing request.

[0050] Step 202: The blockchain node determines an attribute value of the service data.

[0051] The attribute value here can be a serial number of the service data, a hash value of the service data, etc. This is not specifically limited.

5 [0052] Step 203: The blockchain node determines, by using the attribute value of the service data, whether the service data is service data that has been processed; and if it is determined that the service data is the service data that has been processed, performs step 204, otherwise, performs step 206.

10 [0053] In this implementation of the present application, that the blockchain node determines whether the service data is service data that has been processed includes but is not limited to the following: determining, by the blockchain node based on stored attribute values of second service data that has been processed and an attribute value of first service data, whether the first service data has been processed.

15 [0054] For a specific implementation, reference can be made to step 102 described above. Details are omitted here for simplicity.

[0055] Step 204: The blockchain node processes the service data.

[0056] Step 205: The blockchain node broadcasts the service data to another blockchain node in a blockchain network.

[0057] Step 206: The blockchain node discards the service data.

20 [0058] Based on the same inventive concept, FIG. 3 is a schematic structural diagram illustrating a blockchain-based data processing device, according to an implementation of the present application. The data processing device includes an acquisition unit 301, a determining unit 302, and a processing unit 303.

25 [0059] The acquisition unit 301 is configured to obtain first service data, and determine an attribute value of the first service data, where the attribute value is used to represent uniqueness of the first service data.

[0060] The determining unit 302 is configured to determine, based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed.

30 [0061] The processing unit 303 is configured to process the first service data in response to determining that the first service data has not been processed.

[0062] In another implementation of the present application, the data processing device further includes a discarding unit 304.

[0063] The discarding unit 304 is configured to discard processing of the first

service data in response to determining that the first service data has been processed.

[0064] In another implementation of the present application, that a determining unit 302 determines, based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed includes the following: querying, in the stored attribute values of the second service data that has been processed, for whether there is an attribute value that is the same as the attribute value of the first service data; and determining, based on a query result, whether the first service data has been processed.

[0065] In another implementation of the present application, that a processing unit 303 determines that the first service data has not been processed includes the following: determining that the first service data has not been processed in response to determining that no attribute value is found to be the same as the attribute value of the first service data from the stored attribute values of the second service data that has been processed.

[0066] In another implementation of the present application, the data processing device further includes a storage unit 305.

[0067] The storage unit 305 is configured to store the attribute value of the first service data.

[0068] In another implementation of the present application, the data processing device further includes an establishment unit 306.

[0069] The establishment unit 306 is configured to store the attribute value of the first service data in a processed database, where the processed database stores attribute values of service data that has been processed; and determine a query index of the attribute value of the first service data, and establish a mapping relationship between the query index and the attribute value of the first service data.

[0070] In another implementation of the present application, the attribute value is obtained when the service data is generated.

[0071] In another implementation of the present application, the attribute value includes one or more of a hash value of the service data and a serial number of the service data.

[0072] It is worthwhile to note that the data processing device provided in this implementation of the present application can be implemented by hardware, or can be implemented by software. This is not specifically limited here. According to the data processing device described in this implementation of the present application, an attribute value that can represent uniqueness of service data to be processed is compared

with an attribute value, stored in a blockchain node, of service data that has been processed, to determine whether the service data to be processed has been processed; and when it is determined that the service data to be processed has not been processed, processing of the service data to be processed is started. Therefore, a replay attack can

5 be effectively prevented. In addition, because attribute values of different service data are different, a problem that other service data cannot be processed due to the locking of one service data that is being processed can be avoided. Therefore, service data processing efficiency in a blockchain network is effectively ensured, and an overall service data throughput in the blockchain network is increased.

10 [0073] Based on the same inventive concept, an implementation of the present application further provides a data processing device, including at least one processor and a memory.

[0074] The memory is configured to store a program, so that the at least one processor performs the following steps: obtaining first service data, and determining an

15 attribute value of the first service data, where the attribute value is used to represent uniqueness of the first service data; determining, based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed; and processing the first service data in response to determining that the first service data has not been processed.

20 [0075] Based on the same inventive concept, an implementation of the present application further provides a computer storage medium, including a program used in conjunction with a data processing device, where the program can be used by a processor to perform the following steps: obtaining first service data, and determining an attribute value of the first service data, where the attribute value is used to represent

25 uniqueness of the first service data; determining, based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed; and processing the first service data in response to determining that the first service data has not been processed.

[0076] For specific implementations, reference can be made to the content of the

30 previously described implementations. Details are omitted here for simplicity.

[0077] In the 1990s, improvement of a technology can be clearly distinguished between hardware improvement (for example, improvement on a circuit structure such as a diode, a transistor, or a switch) and software improvement (improvement on a method procedure). However, with the development of technologies, improvement of

many method procedures can be considered as direct improvement of a hardware circuit structure. Designers almost all program an improved method procedure to a hardware circuit, to obtain a corresponding hardware circuit structure. Therefore, it cannot be said that improvement of a method procedure cannot be implemented by using a hardware entity module. For example, a programmable logic device (PLD) (for example, a field programmable gate array (FPGA)) is such an integrated circuit. A logical function of the programmable logic device is determined by component programming executed by a user. The designers perform voluntary programming to "integrate" a digital system into a single PLD without requiring a chip manufacturer to design and produce a dedicated integrated circuit chip. In addition, instead of manually producing an integrated circuit chip, the programming is mostly implemented by "logic compiler" software, which is similar to a software compiler used during program development. Original code before compiling is also written in a specific programming language, which is referred to as a hardware description language (HDL), and there is more than one type 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, a Lola, MyHDL, PALASM, and RHDL (Ruby Hardware Description Language), etc. Currently, VHDL (Very-High-Speed Integrated Circuit Hardware Description Language) and Verilog are most commonly used. A person skilled in the art should also understand that a method procedure only needs to be logically programmed, and programmed to the integrated circuit by using the previous hardware description languages so that a hardware circuit that implements the logical method procedure can be easily obtained.

25 [0078] A controller can be implemented by using any appropriate method. For example, the controller can be a microprocessor or a processor, or a computer-readable medium, a logic gate, a switch, an application-specific integrated circuit (ASIC), a programmable logic controller, or an embedded microprocessor that stores computer readable program code (such as software or firmware) that can be executed by the microprocessor or the processor. Examples of the controller include but are not limited to the following microprocessors: ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20, and Silicone Labs C8051F320. The memory controller can also be implemented as a part of the control logic of the memory. A person skilled in the art also knows that a controller can be implemented by using pure computer-readable

program code, and the steps in the method can be logically programmed to enable the controller to further implement same functions in forms of a logic gate, a switch, an application-specific integrated circuit, a programmable logic controller, an embedded microcontroller, etc. Therefore, the controller can be considered as a hardware component, and an apparatus that is included in the controller and that is configured to implement various functions can also be considered as a structure in the hardware component. Alternatively, an apparatus configured to implement various functions can be considered as both a software module for implementing the method and a structure in the hardware component.

5 [0079] The system, apparatus, module, or unit described in the described implementations can be implemented by a computer chip or an entity, or implemented by a product with a certain function. A typical implementation device is a computer. The computer can be, for example, a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media player, a

10 navigation device, an email device, a game console, a tablet computer, or a wearable device, or a combination of any of these devices.

15 [0080] For ease of description, the described apparatus is described by dividing functions into various units. Certainly, when the present application is implemented, the functions of the units can be implemented in one or more pieces of software and/or hardware.

20 [0081] A person skilled in the art should understand that the implementations of the present disclosure can be provided as a method, a system, or a computer program product. Therefore, the present disclosure can use a form of hardware only implementations, software only implementations, or implementations with a

25 combination of software and hardware. In addition, the present disclosure can use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a disk memory, a CD-ROM, and an optical memory) that include computer-usable program code.

[0082] The present disclosure is described with reference to the flowcharts and/or block diagrams of the method, the device (system), and the computer program product according to the implementations of the present disclosure. It should be understood that computer program instructions can be used to implement each process and/or each block in the flowcharts and/or the block diagrams and a combination of a process and/or a block in the flowcharts and/or the block diagrams. These computer program

30

instructions can be provided for a general-purpose computer, a dedicated computer, an embedded processor, or a processor of any other programmable data processing device to generate a machine so that the instructions executed by a computer or a processor of any other programmable data processing device generate an apparatus for 5 implementing a specific function in one or more processes in the flowcharts or in one or more blocks in the block diagrams.

[0083] These computer program instructions can be stored in a computer readable memory that can instruct the computer or any other programmable data processing device to work in a specific method, so that the instructions stored in the computer 10 readable memory generate an artifact that includes an instruction apparatus. The instruction apparatus implements a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0084] These computer program instructions can be loaded to a computer or another programmable data processing device, so that a series of operations and steps are 15 performed on the computer or the another programmable device, generating computer-implemented processing. Therefore, the instructions executed on the computer or the another programmable device provide steps for implementing a specific function in one or more processes in the flowcharts or in one or more blocks in the block diagrams.

[0085] In a typical configuration, the computing device includes one or more 20 processors (CPU), one or more input/output interfaces, one or more network interfaces, and one or more memories.

[0086] The memory can include a non-persistent memory, a random access memory (RAM) and/or a nonvolatile memory in a computer readable medium, for example, a 25 read-only memory (ROM) or a flash memory (flash RAM). The memory is an example of a computer readable medium

[0087] The computer readable medium includes persistent, non-persistent, movable, and unmovable media that can implement information storage by using any method or technology. Information can be a computer readable instruction, a data structure, a program module, or other data. A computer storage medium includes but is not limited 30 to a parameter random access memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), a random access memory (RAM) of another type, a read-only memory (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or another memory technology, a compact disc read-only memory (CD-ROM), a digital versatile disc (DVD) or another

optical storage, a magnetic tape, a magnetic disk storage, another magnetic storage device, or any other non-transmission media that can be used to store information that can be accessed by the computing device. Based on the definition in the present specification, the computer readable medium does not include transitory computer
5 readable medium (transitory media), for example, a modulated data signal and carrier.

[0088] It is worthwhile to further note that the term "include", "contain", or their any other variant is intended to cover a non-exclusive inclusion, so that a process, a method, merchandise, or a device that includes a list of elements not only includes those elements but also includes other elements which are not expressly listed, or further
10 includes elements inherent to such process, method, merchandise, or device. An element preceded by "includes a ..." does not, without more constraints, preclude the existence of additional identical elements in the process, method, merchandise, or device that includes the element.

[0089] The present application can be described in the general context of computer executable instructions executed by a computer, for example, a program module. Generally, the program module includes a routine, a program, an object, a component, a data structure, etc. for executing a specific task or implementing a specific abstract data type. The present application can also be practiced in distributed computing environments. In the distributed computing environments, tasks are performed by
20 remote processing devices connected through a communications network. In a distributed computing environment, the program module can be located in both local and remote computer storage media including storage devices.

[0090] The implementations in the present specification are all described in a progressive way. For the same or similar parts of the implementations, refer to the
25 implementations. Each implementation focuses on a difference from other implementations. Particularly, a system implementation is basically similar to a method implementation, and therefore is described briefly. For related parts, refer to partial descriptions of the method implementation.

[0091] The previous descriptions are merely implementations of the present application, and are not intended to limit the present application. A person skilled in the art can make various modifications and changes to the present application. Any modification, equivalent replacement, improvement, etc. made within the spirit and principle of the present application shall fall within the scope of the claims of the present application.
30

CLAIMS

What is claimed is:

1. A blockchain-based data processing method, comprising the following:
 - obtaining, by a blockchain node, first service data;
 - 5 determining an attribute value of the first service data, wherein the attribute value is used to represent uniqueness of the first service data;
 - determining, by the blockchain node based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed; and
 - 10 processing, by the blockchain node, the first service data in response to determining that the first service data has not been processed.
2. The data processing method according to claim 1, wherein the method further comprises the following:
 - 15 discarding, by the blockchain node, processing of the first service data in response to determining that the first service data has been processed.
3. The data processing method according to claim 1, wherein the determining, by the blockchain node based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed comprises the following:
 - 20 querying, by the blockchain node in the stored attribute values of the second service data that has been processed, for whether there is an attribute value that is the same as the attribute value of the first service data; and
 - 25 determining, by the blockchain node based on a query result, whether the first service data has been processed.
4. The data processing method according to claim 3, wherein the determining, by the blockchain node, that the first service data has not been processed comprises the following:
 - 30 determining, by the blockchain node, that the first service data has not been processed in response to determining that no attribute value is found to be the same as

the attribute value of the first service data from the stored attribute values of the second service data that has been processed.

5. The data processing method according to claim 1, wherein the method further comprises the following:

storing, by the blockchain node, the attribute value of the first service data.

6. The data processing method according to claim 1, wherein the method further comprises the following:

10 storing, by the blockchain node, the attribute value of the first service data in a processed database, wherein the processed database stores attribute values of service data that has been processed;

determining, by the blockchain node, a query index of the attribute value of the first service data; and

15 establishing a mapping relationship between the query index and the attribute value of the first service data.

7. The data processing method according to any one of claims 1 to 6, wherein the attribute value is obtained when the service data is generated.

20

8. The data processing method according to any one of claims 1 to 6, wherein the attribute value comprises one or more of a hash value of the service data and a serial number of the service data.

25

9. A blockchain-based data processing device, comprising the following:

an acquisition unit, configured to obtain first service data, and determine an attribute value of the first service data, wherein the attribute value is used to represent uniqueness of the first service data;

30

a determining unit, configured to determine, based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed; and

a processing unit, configured to process the first service data in response to determining that the first service data has not been processed.

10. The data processing device according to claim 9, wherein the data processing device further comprises a discarding unit, wherein

the discarding unit is configured to discard processing of the first service data in response to determining that the first service data has been processed.

5

11. The data processing device according to claim 9, wherein that a determining unit determines, based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed comprises the following:

10 querying, in the stored attribute values of the second service data that has been processed, for whether there is an attribute value that is the same as the attribute value of the first service data; and

determining, based on a query result, whether the first service data has been processed.

15

12. The data processing device according to claim 11, wherein that a processing unit determines that the first service data has not been processed comprises the following:

20 determining that the first service data has not been processed in response to determining that no attribute value is found to be the same as the attribute value of the first service data from the stored attribute values of the second service data that has been processed.

25 13. The data processing device according to claim 9, wherein the data processing device further comprises a storage unit, wherein

the storage unit is configured to store the attribute value of the first service data.

14. The data processing device according to claim 9, wherein the data processing device further comprises an establishment unit, wherein

30 the establishment unit is configured to store the attribute value of the first service data in a processed database, wherein the processed database stores attribute values of service data that has been processed; and

determine a query index of the attribute value of the first service data; and

establish a mapping relationship between the query index and the attribute value

of the first service data.

15. The data processing device according to any one of claims 9 to 14, wherein the attribute value is obtained when the service data is generated.

5

16. The data processing device according to any one of claims 9 to 14, wherein the attribute value comprises one or more of a hash value of the service data and a serial number of the service data.

10 17. A data processing device, comprising at least one processor and a memory, wherein the memory is configured to store a program, so that the at least one processor performs the following steps:

obtaining first service data;

15 determining an attribute value of the first service data, wherein the attribute value is used to represent uniqueness of the first service data;

determining, based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed; and

20 processing the first service data in response to determining that the first service data has not been processed.

18. A computer storage medium, comprising a program used in conjunction with a data processing device, wherein the program can be used by a processor to perform the following steps:

25 obtaining first service data;

determining an attribute value of the first service data, wherein the attribute value is used to represent uniqueness of the first service data;

determining, based on stored attribute values of second service data that has been processed and the attribute value of the first service data, whether the first service data has been processed; and

30 processing the first service data in response to determining that the first service data has not been processed.

1/3

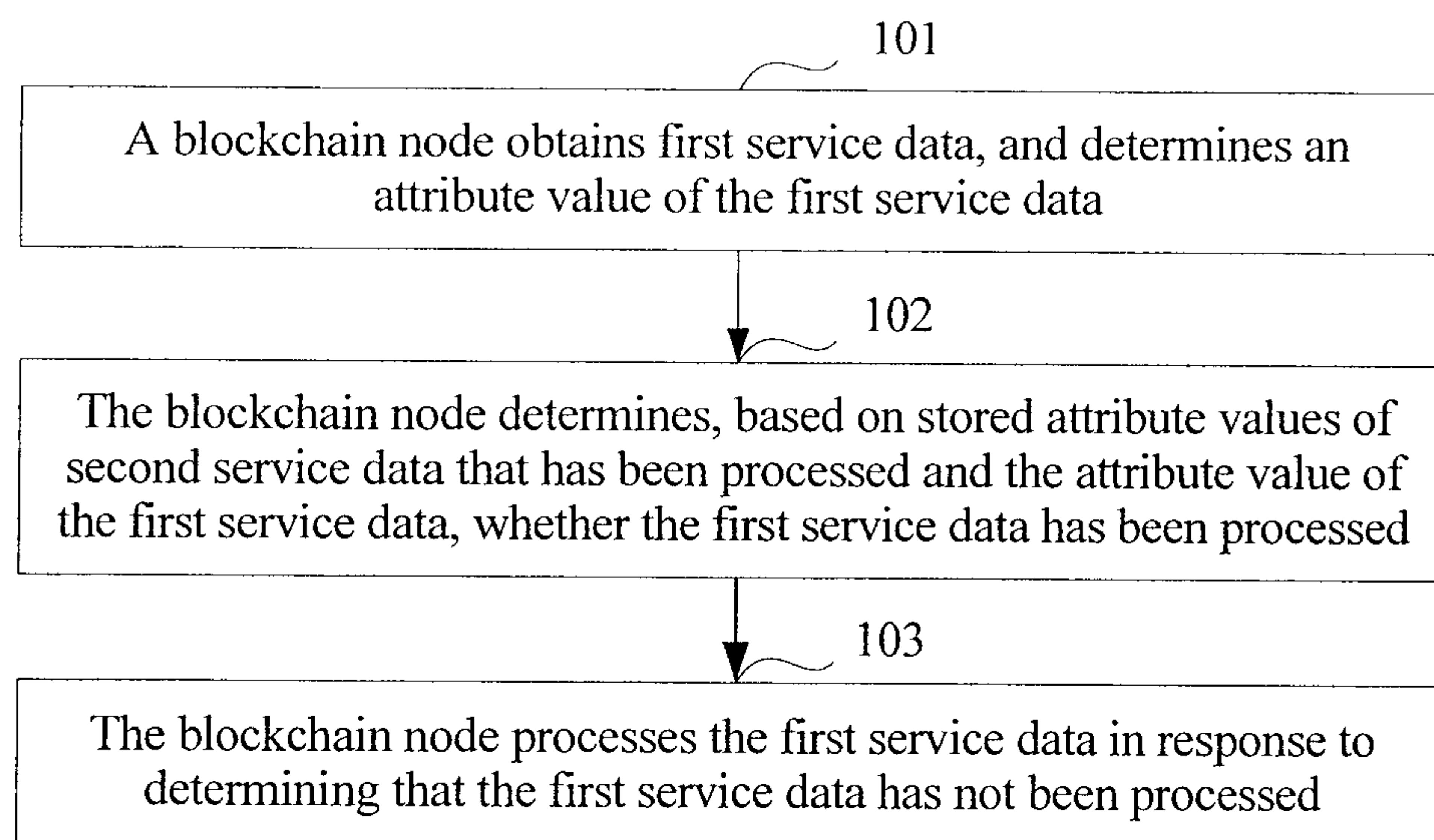


FIG. 1

2/3

201

A blockchain node receives a service processing request, and obtains service data from the service processing request

202

The blockchain node determines an attribute value of the service data

203

The blockchain node determines, by using the attribute value of the service data, whether the service data is service data that has been processed

Yes

No

The blockchain node processes the service data

204

The blockchain node broadcasts the service data to another blockchain node in a blockchain network

205

The blockchain node discards the service data

206

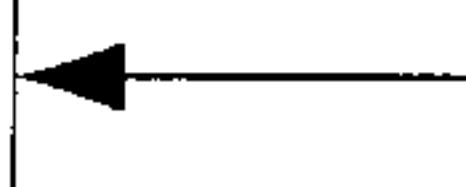


FIG. 2

3/3

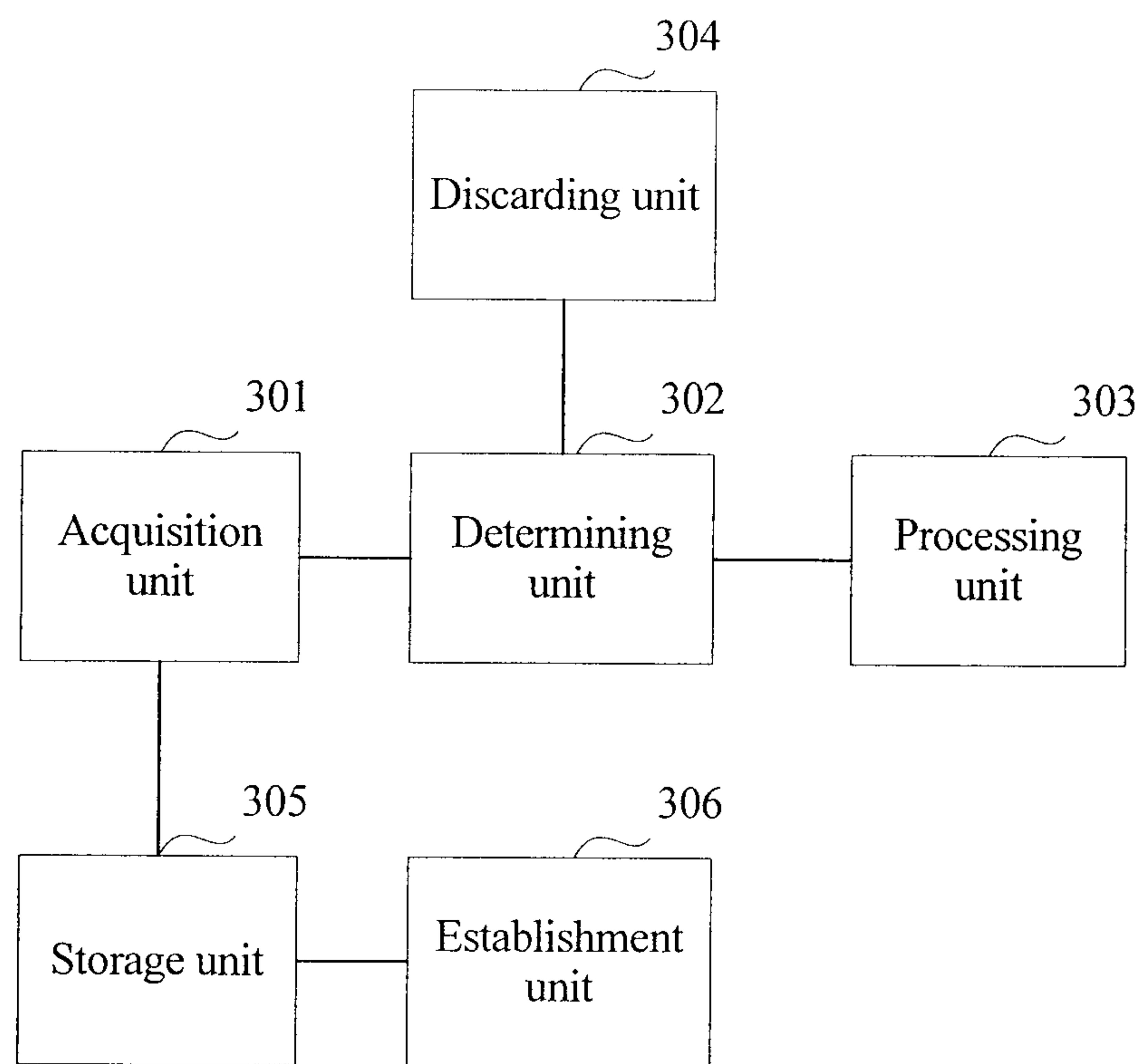


FIG. 3

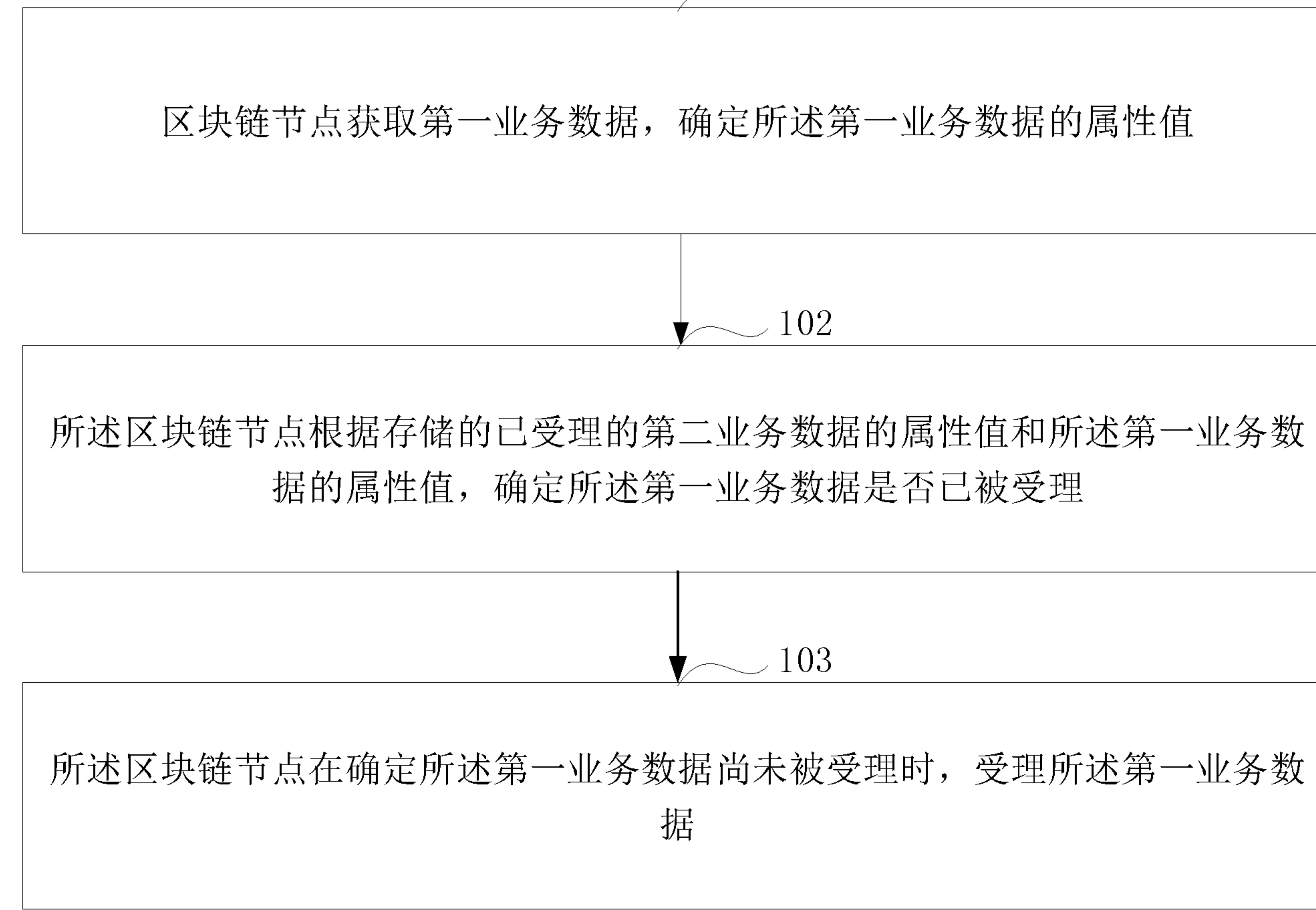


图 1

- 101 A BLOCKCHAIN NODE OBTAINS FIRST TRANSACTION DATA, AND DETERMINES AN ATTRIBUTE VALUE OF THE FIRST TRANSACTION DATA
- 102 THE BLOCKCHAIN NODE DETERMINES, ACCORDING TO A STORED ATTRIBUTE VALUE OF A PROCESSED SECOND TRANSACTION DATA AND THE ATTRIBUTE VALUE OF THE FIRST TRANSACTION DATA, WHETHER THE FIRST TRANSACTION DATA HAS BEEN PROCESSED. IF NOT, THE BLOCKCHAIN NODE PROCESSES THE FIRST TRANSACTION DATA
- 103