

BUSINESS VERIFICATION METHOD AND APPARATUS

Claim of Priority

This application claims priority to Chinese Patent Application No.
5 201710096987.5 filed on February 22, 2017, the entire contents of which are hereby
incorporated by reference.

Technical field

The present application relates to the field of computer technologies, and in
10 particular, to a business verification method and apparatus.

Background art

The block chain technology is also referred to as a distributed account book
technology. Data stored in a block chain has characteristics such as tamper-resistance
15 and decentralization. Therefore, the block chain technology provides a safer data storage
environment for people and makes data storage more convenient for people.

At present, when receiving a business request sent by a terminal, a block chain node
may store the business request in a business memory of its own. At the same time, the
block chain node may broadcast the business request to other block chain nodes in the
20 whole consensus network, such that the other block chain nodes, after receiving the
business request, store the business request in respective corresponding business
memories.

The block chain node will then gain a set number of business requests from the
business memory of its own, package these business requests into a pre-processed block,
25 and broadcast the pre-processed block to other block chain nodes in the whole consensus
network to reach a consensus, so as to determine whether the business requests need to
be stored in the block chain as blocks.

In an actual application, in a process that a block chain node in a consortium chain
broadcasts a received business request to other block chain nodes, some other block
chain nodes in the whole consensus network may not receive the business request
30 broadcast by the block chain node due to impact of factors such as a network failure. In
other words, with respect to business requests stored in a business memory
corresponding to one block chain node, business memories corresponding to other block

chain nodes may be in lack of a part of the business requests. The block chain nodes in lack of a part of the business requests may greatly affect a consensus verification result of the whole consensus network to some extent.

For example, it is assumed that the whole consensus network has three block chain nodes: A, B, and C. The block chain node A stores five business requests: #1, #2, #3, #4, and #5. The block chain node B stores four business requests: #1, #2, #3, and #4. The block chain node C stores three business requests: #1, #2, and #3. The business requests are all stored in business memories corresponding to the block chain nodes. When the block chain node A packages the five business requests #1, #2, #3, #4, and #5 into a pre-processed block, and broadcasts the pre-processed block to other two block chain nodes to carry out consensus verification on the five business requests, as the block chain nodes B and C are both in lack of a part of the five business requests, the block chain nodes B and C may directly consider the pre-processing block including the five business requests as failed in the consensus verification. In this case, as more than half of the block chain nodes in the whole consensus network consider the pre-processed block as failed in the consensus verification, the five business requests included in the pre-processed block will not be able to pass the consensus verification of the whole consensus network. The five business requests will then not be recorded in block chains of the whole consensus network.

The other block chain nodes consider the pre-processed block as failed in the consensus verification only because a part of the business requests in the pre-processed block is not stored in the business memories corresponding to the other block chain nodes, rather than that a part of the business requests in the pre-processed block is illegally tampered. Therefore, the probability that the pre-processed block cannot pass the consensus verification of the whole consensus network will be greatly increased. However, the business requests included in the pre-processed block may not have any problems actually. As a result, a normal business request is very likely to be failed in the consensus verification of the block chain nodes in the prior art, thereby affecting the business processing accuracy of the whole block chain business.

30

Summary of the Invention

Embodiments of the present application provide a business verification method to solve the problem of low business processing accuracy of a block chain business in the prior art.

5 The embodiments of the present application provide a business verification method, including:

receiving, by a first block chain node, a business request sent by a terminal;

storing the business request in a business memory corresponding to the first block chain node, and broadcasting the business request to second block chain nodes, such that
10 the second block chain nodes store the business request in respective corresponding business memories; and

gaining at least one business request from the business memory, packaging the gained at least one business request into a pre-processed block, and broadcasting the pre-processed block to the second block chain nodes, such that when determining that
15 the business memory corresponding thereto does not include a part of the business request in the pre-processed block, each of the second block chain nodes acquires the part of the business request from another block chain node, and carries out consensus verification on the pre-processed block by using the part of the business request and a business request stored in the business memory of its own.

20 The embodiments of the present application provide a business verification apparatus to solve the problem of low business processing accuracy of a block chain business in the prior art.

The embodiments of the present application provide a business verification apparatus, including:

25 a receiving module configured to receive a business request sent by a terminal;

a storage module configured to store the business request in a business memory corresponding to the apparatus, and broadcast the business request to second block chain nodes, such that the second block chain nodes store the business request in respective corresponding business memories; and

30 a request gaining module configured to gain at least one business request from the business memory, package the gained at least one business request into a pre-processed block, and broadcast the pre-processed block to the second block chain nodes, such that when determining that the business memory corresponding thereto does not include a

part of the business request in the pre-processed block, each of the second block chain nodes acquires the part of the business request from another block chain node, and carries out consensus verification on the pre-processed block by using the part of the business request and a business request stored in the business memory of its own.

5 The embodiments of the present application provide a business verification method to solve the problem of low business processing accuracy of a block chain business in the prior art.

The embodiments of the present application provide a business verification method, including:

10 receiving, by a second block chain node, a business request broadcast by a first block chain node;

storing the business request in a business memory corresponding to the second block chain node;

receiving a pre-processed block that is broadcast by the first block chain node and
15 includes at least one business request, and acquiring a part of the business request from another block chain node when it is determined that the business memory corresponding to the second block chain node does not include the part of the business request in the pre-processed block; and

carrying out consensus verification on the pre-processed block by using the part of
20 the business request and a business request stored in the business memory corresponding to the second block chain node.

The embodiments of the present application provide a business verification apparatus to solve the problem of low business processing accuracy of a block chain business in the prior art.

25 The embodiments of the present application provide a business verification apparatus, including:

a request receiving module configured to receive a business request broadcast by a first block chain node;

a request storage module configured to store the business request in a business
30 memory corresponding to the apparatus;

a receiving module configured to receive a pre-processed block that is broadcast by the first block chain node and includes at least one business request, and acquire a part of the business request from another block chain node when it is determined that the

business memory corresponding thereto does not include the part of the business request in the pre-processed block; and

a verification module configured to carry out consensus verification on the pre-processed block by using the part of the business request and a business request stored
5 in the business memory corresponding thereto.

The at least one above technical solution used by the embodiments of the present application can achieve the following beneficial effects:

In the embodiments of the present application, when a second block chain node finds, after receiving a pre-processed block which is broadcast by a first block chain node and
10 includes business requests, that a business memory corresponding thereto does not include a part of the business requests in the pre-processed block, the second block chain node does not directly consider the pre-processed block as failed in local consensus verification. Instead, the second block chain node may acquire the missing business requests from other block chain nodes in the whole consensus network, and carry out
15 consensus verification on the business requests included in the pre-processed block by using the acquired business requests and business requests stored in the business memory of its own. In this way, adverse effects on consensus verification of business requests caused by a network failure are greatly reduced, thereby improving the business processing accuracy of the whole block chain business.

20

Brief Description of the Drawings

The accompanying drawings described herein are used to provide further understanding of the present application, and construct a part of the present application. Exemplary embodiments of the present application and illustrations thereof are used to
25 explain the present application, but are not intended to form any improper limitation on the present application. In the accompanying drawings:

FIG. 1 is a schematic diagram of a business efficiency process according to an embodiment of the present application;

FIG. 2 is a schematic diagram of storing, by block chain nodes, received business requests in business memories corresponding thereto respectively by using preset distributed middleware according to an embodiment of the present application;
30

FIG. 3 is a schematic diagram of determining a to-be-verified total characteristic value according to an embodiment of the present application;

FIG. 4 is a schematic diagram of establishing a consensus by a consensus network for a pre-processed block sent by a first block chain node according to an embodiment of the present application;

FIG. 5 is a schematic diagram of a business verification apparatus according to an embodiment of the present application; and

FIG. 6 is a schematic diagram of another business verification apparatus according to an embodiment of the present application.

Detailed Description

At present, a process of carrying out business processing by a block chain node is substantially as follows: after a terminal sends a business request to a block chain node, the block chain node may send the received business request to other block chain nodes by broadcasting. The other block chain nodes may store the received business request in business memories corresponding thereto. Definitely, the block chain node sending the business request to other block chain nodes may also store the business request in a business memory of its own.

In a consensus network formed by block chain nodes, each block chain node has a right to initiate a consensus request to other block chain nodes. The block chain node may arrange a set number of business requests stored in the business memory of its own in a certain order, to obtain a business request queue, and generate a Hash value for the business request queue. The block chain node may then package the business request queue and the Hash into a pre-processed block, and send the pre-processed block to other block chain nodes by broadcasting, so as to carry out consensus verification.

During the consensus verification, after receiving the pre-processed block, the other block chain nodes will verify the legality of asymmetric signatures of the business requests included in the pre-processed block. That is, the block chain node may parse the business requests included in the pre-processed block according to a public key (or private key, depending on whether a private key or a public key is used when the business requests are encrypted) possessed by its own, to verify whether the business requests are legal business requests.

In addition, upon receiving a business request sent by a terminal, the block chain node may broadcast the business request to other block chain nodes. Therefore, the business memories corresponding to the block chain nodes generally all store the

business requests received by the whole consensus network. On this basis, after receiving the pre-processed block, the other block chain nodes may verify Hash integrity of the business requests in the pre-processed block. That is, the block chain node may search the business memory of its own for the business requests included in the pre-processed block, and arrange the found business requests according to an arrangement order of the business requests in the pre-processed block, to obtain a business request queue. The block chain node may then generate a Hash value for the business request queue, and compare the obtained Hash value with the Hash value included in the pre-processed block, so as to determine whether content of the business requests in the pre-processed block has been tampered illegally.

According to the asymmetric signature legality verification and the Hash integrity verification carried out for the pre-processed block, each of the block chain nodes will obtain a verification result of its own about whether the pre-processed block as a whole is legal, and broadcast the verification result obtained by itself to other block chain nodes via broadcasting.

According to verification results sent by the other block chain nodes for the pre-processed block and the verification result obtained by itself, each of the block chain nodes will obtain an integrated verification result, about whether the pre-processed block passes the verifications, of the block chain nodes in the whole consensus network, and further broadcast the obtained integrated verification result to the other block chain nodes via broadcasting.

After receiving the integrated verification results broadcast to each other, each of the block chain nodes in the consensus network will further judge whether most of the integrated verification results obtained by the block chain nodes in the consensus network indicate that the verification is successful. If yes, the business requests in the pre-processed block are stored in a block chain of its own as blocks; or if no, the business requests in the pre-processed block are refused.

After storing the business requests in the pre-processed block into a block chain of its own as blocks, each of the block chain nodes may release the business requests included in the pre-processed block from the business memory of its own, such that the business memory after releasing can continue to store business requests received by the block chain node.

However, in the prior art, in the process that the block chain node broadcasts the received business request to other block chain nodes, some of the other block chain nodes may not receive the business request broadcast by the block chain node due to impact of factors such as a network failure. As a result, when the block chain node
5 broadcasts the pre-processed block including the set number of business requests to other block chain nodes for consensus verification subsequently, because corresponding business memories of some block chain nodes are in lack of some of the business requests in the pre-processed block, these block chain nodes may directly consider that the pre-processed block is failed in local consensus verifications of the block chain
10 nodes. Therefore, the probability that the pre-processed block is failed in the consensus verification of the whole consensus network is greatly increased, thereby affecting the business processing accuracy of the whole block chain business.

Further, in an actual application, if the business request is failed in the consensus verification of the whole consensus network, the block chain node will return a message
15 indicating a processing failure of the business request to the user terminal. Therefore, the above situation may further bring about great inconvenience for the user.

To effectively solve the above problem, in the present application, when a second block chain node finds, after receiving a pre-processed block which is broadcast by a first block chain node and includes business requests, that a business memory
20 corresponding thereto does not include a part of the business requests in the pre-processed block, the second block chain node may acquire the missing business requests from other block chain nodes in the whole consensus network, and carry out consensus verification on the business requests included in the pre-processed block by using the acquired business requests and business requests stored in the business memory of its
25 own. In this way, adverse effects on consensus verification of business requests caused by a network failure are greatly reduced, thereby improving the business processing accuracy of the whole block chain business.

In order to make those skilled in the art better understand the technical solutions in the present application, the technical solutions in the embodiments of the present
30 application will be described clearly and completely through the accompanying drawings in the embodiments of the present application. Apparently, the described embodiments are merely some of embodiments of the present application, rather than all the embodiments. Based on the embodiments of the present application, all other

embodiments derived by those of ordinary skill in the art without any creative effort shall fall within the protection scope of the present application.

FIG. 1 is a schematic diagram of a business efficiency process according to an embodiment of the present application, specifically including the following steps:

5 S101: A first block chain node receives a business request sent by a terminal.

In the embodiment of the present application, during business processing, a user may fill corresponding business processing content in a user terminal held by the user. The terminal will generate a corresponding business request according to the business processing content filled by the user, and send the business request to a first block chain node in a whole consensus network. The terminal mentioned here may be a device such as a computer or a smart phone. Definitely, the user may also send the business request to the first block chain node by using a client terminal installed in the terminal. That is, the user fills corresponding business processing content in an interface presented in the client terminal on the terminal. The client terminal generates a corresponding business request according to the business processing content filled by the user in the interface, and further sends the business request to the first block chain node by using the terminal.

It should be noted that, in an actual application, the whole consensus network includes multiple block chain nodes. The first block chain node mentioned in the embodiment of the present application refers to a block chain node that receives the business request of the terminal. Block chain nodes other than the first block chain node may be referred to as second block chain nodes in the embodiment of the present application. The first block chain node and the second block chain node are relative concepts. That is, the block chain node that receives the business request from the terminal is the first block chain node, and the block chain node that receives the business request broadcast by the first block chain node is referred to as the second block chain node. The block chain nodes in the consensus network may all receive the business request sent by the terminal, and therefore, all the block chain nodes may be the first block chain nodes or second block chain nodes substantially. Whether a block chain node is the first block chain node or the second block chain node depends on where the received business request is from.

Definitely, during the consensus verification, whether a block chain node is the first block chain node or the second block chain node may also be determined according to a node that initiates the consensus verification. That is, a consensus verification initiator

that broadcasts a pre-processed block including at least one business request to the whole consensus network may be the first block chain node, and a block chain node that receives the pre-processed block may be referred to as the second block chain node.

5 S102: The business request is stored in a business memory corresponding to the first block chain node, and the business request is broadcast to second block chain nodes, such that the second block chain nodes store the business request in respective corresponding business memories.

During the consensus verification, the first block chain node needs to gain a part of business requests from the business memory corresponding thereto, package the part of
10 business requests into a pre-processed block, and broadcast the pre-processed block to the second block chain nodes in the whole consensus network. After receiving the pre-processed block including the part of business requests, the second block chain nodes need to carry out consensus verification on the part of business requests in the pre-processed block according to business requests included in their respective
15 corresponding business memories and matching with the part of business requests. The business requests included in the respective corresponding business memories of the second block chain nodes and matching with the part of business requests need to be acquired from the first block chain node.

On this basis, in the embodiment of the present application, after receiving the
20 business request sent by the terminal, the first block chain node may store the business request in the business memory corresponding thereto. At the same time, the first block chain node may send the business request to the second block chain nodes in the whole consensus network by broadcasting, such that the second block chain nodes store the business request in their respective corresponding business memories.

25 In the prior art, the first block chain node generally stores the received business request in a cache of its own. That is, in the prior art, the business memory is a cache of the block chain node. The cache has a limited storage space. Therefore, when the storage space of the cache is occupied completely, the first block chain node cannot continue to receive a business request sent by the terminal. Only after a part of business requests in
30 the cache passes the consensus verification carried out by all the block chain nodes in the whole consensus network, can storage space occupied by this part of business requests be utilized to continue to store the business request sent by the terminal.

Therefore, in the prior art, the storage space of the cache greatly limits the business processing efficiency of the block chain business.

To effectively solve the problem in the prior art, in the embodiment of the present application, operation and maintenance staff of block chain nodes may set business memories in a database form for the block chain nodes respectively. That is, each block chain node may correspondingly have a business memory in a database form. In other words, the business memory mentioned in the embodiment of the present application is a database used for storing business requests. In this way, after receiving the business request sent by the terminal, the first block chain node may store the business request in the business memory corresponding to the first block chain node, and carry out consensus verification on the business request stored in the business memory in a subsequent procedure.

The storage space of the business memory in a database form is much greater than the storage space of the cache. Therefore, when the first block chain node carries out the consensus verification on a part of the business requests in the business memory by using the whole consensus network, the first block chain node may still continue to receive business requests sent by the terminal. That is, it is unnecessary to utilize the storage space occupied by the part of business requests passing the consensus verification to receive business requests sent by the terminal. Compared with the prior art, the first block chain node greatly meets the requirement of constantly growing business volume of the block chain business, and the business processing efficiency of the block chain business is improved.

Further, in the prior art, as the block chain nodes in the whole consensus network all store business requests in caches of their own (that is, the business memories in the prior art are caches), when a block chain node is down or has other failures, the business requests stored in the cache of its own will disappear. However, in the embodiment of the present application, the business requests are stored in the database-form business memories corresponding to the block chain nodes. Therefore, even if a block chain node is down or has other failures, the business requests stored in the database-form business memory will not disappear, thereby further guaranteeing the security of the business requests.

In the embodiment of the present application, data transmission between the block chain nodes and the business memories in the whole consensus network may be

implemented by using preset distributed middleware. That is, after receiving the business request sent by the terminal, the first block chain node may send the business request to the distributed middleware. The distributed middleware may send, according to a node identifier of the first block chain node, the business request to the business memory corresponding to the first block chain node for storage. Likewise, after receiving the business request broadcast by the first block chain node, the second block chain node may send the business request to the distributed middleware. The distributed middleware may also send, according to a node identifier of the second block chain node, the business request to the business memory corresponding to the second block chain node for storage, as shown in FIG. 2.

FIG. 2 is a schematic diagram of storing, by block chain nodes, received business requests in business memories corresponding thereto respectively by using preset distributed middleware according to an embodiment of the present application.

By taking a transaction business as an example, when a user needs to carry out a transfer business, the user may select a transfer object in a terminal held by the user, and enter a transfer amount. The terminal will generate a corresponding transaction request according to content entered by the user, and send the transaction request to a first block chain node.

After receiving the transaction request (i.e., the business request) sent by the terminal, the first block chain node may send the transaction request to preset distributed middleware, such that the preset distributed middleware can store, according to a node identifier of the first block chain node, the transaction request in a business memory corresponding to the first block chain node.

When receiving the transaction request, the first block chain node may then send the transaction request to other block chain nodes, i.e., second block chain nodes in a whole consensus network by broadcasting. After receiving the transaction request, the second block chain nodes may also send the transaction request to the preset distributed middleware, such that the preset distributed middleware stores, according to respective node identifiers of the second block chain nodes, the transaction request in respective business memories corresponding to the second block chain nodes.

It should be noted that, when receiving the business request sent by the first block chain node, the second block chain node may also send the business request to other block chain nodes in the whole consensus network by broadcasting. For a normal and

legal business request, it is actually expected by the whole consensus network that the business request can pass consensus verification carried by the block chain nodes. Therefore, it is actually expected by the whole consensus network that the business request can exist in the business memories corresponding to the block chain nodes
5 before the consensus verification is carried out.

However, in an actual application, network communication between the block chain nodes generally has failures such as a network outage and network jitter. If the business request is only broadcast by the first block chain node while other block chain nodes (i.e., the second block chain nodes) do not broadcast the business request again, when a
10 failure occurs in network communication between the first block chain node and one or more second block chain nodes, the one or more second block chain nodes will be unable to receive the business request, thereby affecting the consensus verification for the business request in a subsequent procedure.

To prevent this situation as much as possible, in the embodiment of the present application, after receiving the business request sent by the first block chain node, the
15 second block chain node may further broadcast the business request to other block chain nodes in the whole consensus network by broadcasting. When receiving the business request, the other block chain nodes may first judge whether the business request has been previously received; if yes, the other block chain nodes ignore the business request;
20 if no, the other block chain nodes store the business request in business memories corresponding thereto by using the preset distributed middleware.

For example, in FIG. 2, when a failure occurs in network communication between the first block chain node and a second block chain node 3, the second block chain node 3 can still receive the transaction request by using a second block chain node 2 and a
25 second block chain node 4. Therefore, it is guaranteed that the transaction request will be stored in the business memories of the block chain nodes in the whole consensus network as far as possible when the transaction request is a normal and legal transaction request.

In the embodiment of the present application, in a process of storing the business request, the first block chain node may first determine a business type corresponding to
30 request, the first block chain node may first determine a business type corresponding to the business request, and rank the business request and previously received business requests according to a preset priority sequence of business types.

In an actual application, different businesses require different delays for business processing. For example, a transaction-type business generally has a relatively high requirement on business processing delay. That is, it is expected that the whole consensus network can finish processing the business quickly. A public benefit type
5 business has a relatively low requirement on business processing delay. That is, the business will not be greatly affected even if the whole consensus network processes the business after a long time.

On this basis, in the embodiment of the present application, when storing the business request in the business memory corresponding to the first block chain node, the
10 first block chain node may rank the business request in the business memory according to a priority sequence of businesses, thereby obtaining a business queue including the business request. In the business queue, business requests having high requirements on delay have high ranks, while business requests having low requirements on delay have low ranks. A specific ranking manner is determined according to the priority sequence
15 of business types preset by the operation and maintenance staff.

It should be noted that, in the embodiment of the present application, in addition to determining an arrangement sequence of business requests in a business queue according to the priority sequence of the business types, the first block chain node may also comprehensively determine the arrangement sequence of the business requests in
20 the business memory according to storage time of the business requests in the business memory. That is, when the storage time of a business request in the business memory is too long, the business request may be lifted to the front of the whole business queue even if the business request has a low requirement on delay.

S103: At least one business request is gained from the business memory, the gained
25 at least one business request is packaged into a pre-processed block, and the pre-processed block is broadcast to the second block chain nodes, such that when determining that the business memory corresponding thereto does not include a part of the business request in the pre-processed block, each of the second block chain nodes acquires the part of the business request from another block chain node, and carries out
30 consensus verification on the pre-processed block by using the part of the business request and a business request stored in the business memory of its own.

In the embodiment of the present application, it is necessary for the first block chain node to establish a consensus for the business request stored in the business memory

corresponding thereto by using the whole consensus network, to finish processing the business request. Therefore, the first block chain node may gain at least one business request from the business memory corresponding thereto, and then establish a consensus for the at least one business request by using the whole consensus network in a subsequent procedure.

The first block chain node may gain business requests with a business type higher than a set priority from the business memory. For example, by using a business type as a boundary, the first block chain node may gain business requests of business types having priorities higher than the business type from the business memory.

Definitely, the first block chain node may also gain a set number of business requests from the business memory. For example, when the business requests are stored in the business memory in the form of the above business queue, the first block chain node may gain a set number of business requests from the business queue. Further, if the set number is represented by N, the first block chain node may gain first N business requests from the business queue.

In addition to gaining the business requests according to the set number, the first block chain node may also gain the business requests according to another standard. For example, the first block chain node may gain business requests having storage time exceeding a preset time length in the business memory. Alternatively, when establishing a consensus for the business requests by using the whole consensus network, the first block chain node may select a business, and gain business requests corresponding to the business from the business memory. The first block chain node may select the business randomly, or select the business according to a certain sequence. Definitely, the first block chain node may further gain business requests according to other standards, which are not described in detail here.

After the first block chain node gains the set number of business requests, sub-characteristic values corresponding to the business requests are determined respectively according to a preset characteristic value determination rule. For example, when the preset characteristic value determination rule is a Hash algorithm, the first block chain node may determine sub-Hash values corresponding to the business requests respectively. When the preset characteristic value determination rule is a Message Digest Algorithm (MD5), the first block chain node may determine sub-MD5 values corresponding to the business requests respectively.

After the first block chain node determines the sub-characteristic values corresponding to the business requests, a to-be-verified characteristic value uniquely corresponding to the business requests may be determined according to the determined characteristic values and the arrangement sequence of the business requests in the business memory.

The to-be-verified characteristic value is uniquely corresponding to the business requests as a whole. That is, when content of a business request among the business requests is changed, the to-be-verified characteristic value will be changed. A manner of determining the to-be-verified characteristic value by the first block chain node is as shown in FIG. 3.

FIG. 3 is a schematic diagram of determining a to-be-verified characteristic value according to an embodiment of the present application.

In FIG. 3, the characteristic value determination rule used by the first block chain node is the Hash algorithm. Assume that the first block chain node gains a set number (which is 4) of business requests from the business memory corresponding thereto. Arrangement of the four business requests in a business queue is as shown in FIG. 3. After determining four Hash values corresponding to the four business requests respectively, the first block chain node may place the four Hash values on four leaf nodes of a Merkle tree from left to right according to ranks of the four business requests in the business queue, and determine non-leaf nodes and a root node of the Merkle tree accordingly. The first block chain node may then determine the root node Hash7 of the Merkle tree as a to-be-verified characteristic value uniquely corresponding to the four business requests.

It should be noted that the method of determining the to-be-verified characteristic value described above is not the only method. The first block chain node may also carry out determination in other manners, as long as it is guaranteed that the to-be-verified characteristic value is uniquely corresponding to the business requests in a certain sequence.

After determining the to-be-verified characteristic value uniquely corresponding to the business requests (that is, the at least one business request gained from the business memory), the first block chain node may package the to-be-verified characteristic value and the business requests into a pre-processed block, the pre-processed block including the business requests and the to-be-verified characteristic value. At the same time, the

business requests in the pre-processed block are arranged according to the arrangement sequence of the business requests in the business memory.

The first block chain node may send the determined pre-processed block to other block chain nodes (i.e., the second block chain nodes) in the whole consensus network by broadcasting, and then the whole consensus network establishes a consensus on the business requests included in the pre-processed block, as shown in FIG. 4.

FIG. 4 is a schematic diagram of establishing a consensus by a consensus network for a pre-processed block sent by a first block chain node according to an embodiment of the present application.

In FIG. 4, the first block chain node may broadcast the pre-processed block to the second block chain nodes in the whole consensus network. For each of the second block chain nodes, when receiving the pre-processed block sent by the first block chain node, the second block chain node may parse the pre-processed block, to determine the business requests and the to-be-verified characteristic value that are included in the pre-processed block.

As for each of the second block chain nodes, after obtaining the business requests from the pre-processed block by parsing, the second block chain node then needs to carry out asymmetric signature legality verification on the business requests obtained by parsing, to verify whether the business requests are all legal business requests.

Specifically, when sending a business request to a block chain node, the terminal may generally encrypt (sign) the business request by using a possessed private key (which definitely may also be a public key). Therefore, when carrying out the asymmetric signature legality verification on the business requests included in the pre-processed block, the second block chain node needs to parse the business requests by using a public key (or, when the terminal carries out encryption by using the public key, the second block chain node parses the business requests by using a possessed private key), and verifies content obtained through parsing.

For example, when carrying out the asymmetric signature legality verification on a transaction request (i.e., a business request) in the pre-processed block, the second block chain node may decrypt the transaction request by using the public key possessed by its own, so as to obtain account addresses of both transaction parties involved in the transaction request, thereby verifying whether the account addresses of the both transaction parties are legal. When it is determined that the account addresses of the both

transaction parties involved in the transaction request are legal accounts, and the amount of money stored in an account of a transaction initiator is greater than or equal to a transfer amount involved in the transaction request, it is determined that the transaction request passes the asymmetric signature legality verification; otherwise, it is determined
5 that the transaction request is failed in the asymmetric signature legality verification.

After determining that the business requests included in the pre-processed block all pass the asymmetric signature legality verification, the second block chain node may further determine sub-characteristic values corresponding to the business requests by using a preset characteristic value determination rule. The characteristic value
10 determination rule used by the second block chain node is the same as that used by the first block chain node.

After determining the sub-characteristic values corresponding to the business requests, the second block chain node may determine a characteristic value uniquely corresponding to the business requests as a whole according to an arrangement sequence
15 of the business requests in the pre-processed block and the sub-characteristic values. The second block chain node then compares the characteristic value with the to-be-verified characteristic value in the pre-processed block. When the two characteristic values are the same, it can be determined that content of the business requests on which a consensus needs to be established by the first block chain node is not changed, i.e., it
20 is determined that the business requests pass the Hash integrity verification.

After the second block chain nodes carry out the asymmetric signature legality verification and the Hash integrity verification on the pre-processed block according to the above method, local verification results for the pre-processed block may be obtained respectively (only when the business requests in the pre-processed block all pass the
25 asymmetric signature legality verification and the Hash integrity verification, can the local verification result of the pre-processed block indicate a success; the local verification result of the pre-processed block indicates a failure once either of the verifications is unsuccessful). Each of the second block chain nodes may then send the respective obtained verification result to other block chain nodes in the whole consensus
30 network by broadcasting, so as to enter a consensus verification procedure of the whole consensus network. After receiving the verification results broadcast to each other, each of the block chain nodes in the whole consensus network may obtain an integrated verification result, about whether the business requests included in the pre-processed

block passes the verifications, of the block chain nodes in the whole consensus network according to the received verification results and the verification result obtained by itself. The obtained integrated verification result is then further broadcast to the other block chain nodes in the whole consensus network.

5 After receiving the integrated verification results broadcast to each other, each of the block chain nodes in the consensus network may further judge whether most of the integrated verification results obtained by the block chain nodes in the consensus network indicate that the verification is successful. If yes, the business requests included in the pre-processed block are written into a block for storage, and the block is further
10 written into a block chain, in which the block is stored, according to a time sequence; or if no, the business requests are refused.

 The consensus verification procedure of the whole consensus network described above is a general consensus verification procedure. In the embodiment of the present application, a procedure of carrying out consensus verification on a set number of
15 business requests by the whole consensus network may further involve a complicated consensus algorithm, such as a Practical Byzantine Fault Tolerance (PBFT) algorithm, a consistency algorithm (Raft), and a Paxos algorithm. The procedure involving the consensus algorithm in the embodiment of the present application is the same as that in the prior art, and will not be described in detail here.

20 After a block chain node (the block chain node mentioned here may be the first block chain node or the second block chain node) stores the business requests in the block chain as blocks, storage space occupied by the business requests in respective business memories may be released, and the business requests are transferred to a database used for storing historical business requests.

25 It should be noted that the second block chain nodes may further broadcast the business request of the first block chain node. However, some block chain nodes in the whole consensus network may still be unable to receive the business request effectively due to influences of the network condition. Therefore, during the consensus stage, when a second block chain node does not find, from the business memory corresponding
30 thereto, a part of the business request in the pre-processed block, the second block chain node may send a query message for acquiring this part of the business request to another block chain node by using a preset distributed middleware. After receiving the query message, the another block chain node may determine whether a business memory

corresponding thereto includes this part of the business request; if yes, the another block chain node returns a reply message to the second block chain node; if no, the another block chain node does not return the reply message to the second block chain node.

After receiving the reply message, the second block chain node may acquire, by
5 using the preset distributed middleware, this part of the business request from the business memory corresponding to the block chain node sending the reply message. The second block chain node may then carry out the asymmetric signature legality verification on this part of the business request. When determining that this part of the business request passes the asymmetric signature legality verification, the second block
10 chain node stores this part of the business request in the business memory corresponding thereto. The second block chain node may store this part of the business request in the business memory corresponding thereto according to an arrangement sequence of the business requests in the pre-processed block. When determining that this part of the business request does not pass the asymmetric signature legality verification, the second
15 block chain node does not store this part of the business request, and determines that the pre-processed block sent by the first block chain node is failed in the local (i.e., the second block chain node) consensus verification.

After receiving this part of the business request from the another block chain node, if the second block chain node still receives this part of the business request from other
20 block chain nodes, the second block chain node may ignore this part of the business request received subsequently, and it is unnecessary to carry out the asymmetric signature legality verification on and store this part of the business request received subsequently.

In the embodiment of the present application, the whole consensus network may be
25 a consensus network of a consortium chain, and the block chain nodes may be block chain nodes in the consortium chain. In the embodiment of the present application, the first block chain node may be a leader node in a consortium chain consensus algorithm, and the second block chain node may be a non-leader node in the consortium chain consensus algorithm.

30 It can be seen from the above method that, when a second block chain node finds, after receiving business requests broadcast by a first block chain node, that a business memory corresponding thereto does not include some of the business requests, the second block chain node does not directly consider the business requests as failed in

local consensus verification. Instead, the second block chain node may acquire the missing business requests from other block chain nodes in the whole consensus network, and carry out consensus verification on the business requests received from the first block chain node by using the acquired business requests and business requests stored
5 in the business memory of its own. In this way, adverse effects on consensus verification of business requests caused by a network failure are greatly reduced, thereby improving the business processing accuracy of the whole block chain business.

Further, in the embodiment of the present application, the business memory, for storing business requests, of each block chain node exists as a database. Compared with
10 the prior art in which each block chain node stores business requests by using a respective cache, the business memory in the form of a database provided in the embodiment of the present application greatly improves the storage capability for business requests. Moreover, when a block chain node carries out consensus verification on a part of the business requests in the business memory by using the whole consensus
15 network, the block chain node can still continue to receive business requests sent by the terminal. That is, it is unnecessary to utilize the storage space occupied by the part of the business requests passing the consensus verification to receive business requests sent by the terminal, and the business processing efficiency of the block chain business is further improved.

20 The business verification method provided in the embodiment of the present application is described in the foregoing, and based on the same idea, the embodiments of the present application further provide two types of business verification apparatuses, as shown in FIG. 5 and FIG. 6.

FIG. 5 is a schematic diagram of a business verification apparatus according to an
25 embodiment of the present application, specifically including:

- a receiving module 501 configured to receive a business request sent by a terminal;
- a storage module 502 configured to store the business request in a business memory corresponding to the apparatus, and broadcast the business request to second block chain nodes, such that the second block chain nodes store the business request in respective
30 corresponding business memories; and

- a request gaining module 503 configured to gain at least one business request from the business memory, package the gained at least one business request into a pre-processed block, and broadcast the pre-processed block to the second block chain nodes,

such that when determining that the business memory corresponding thereto does not include a part of the business request in the pre-processed block, each of the second block chain nodes acquires the part of the business request from another block chain node, and carries out consensus verification on the pre-processed block by using the part
5 of the business request and a business request stored in the business memory of its own.

The business memory is a database that stores business requests.

The storage module 502 is configured to store the business request in the business memory by using preset distributed middleware.

The request gaining module 503 is configured to gain, from the business memory, a
10 set number of business requests with a business type higher than a set priority.

The storage module 502 is configured to store the business request in the business memory according to the business type of the business request and a preset priority sequence of business types.

The apparatus is a leader node in a consortium chain consensus algorithm, and the
15 second block chain node is a non-leader node in the consortium chain consensus algorithm.

FIG. 6 is a schematic diagram of another business verification apparatus according to an embodiment of the present application, specifically including:

a request receiving module 601 configured to receive a business request broadcast
20 by a first block chain node;

a request storage module 602 configured to store the business request in a business memory corresponding to the apparatus;

a receiving module 603 configured to receive a pre-processed block that is broadcast by the first block chain node and includes at least one business request, and acquire a
25 part of the business request from another block chain node when it is determined that the business memory corresponding thereto does not include the part of the business request in the pre-processed block; and

a verification module 604 configured to carry out consensus verification on the pre-processed block by using the part of the business request and a business request stored
30 in the business memory corresponding thereto.

The receiving module 603 is configured to send a query message for acquiring the part of the business request to another second block chain node or the first block chain node when it is determined that the business memory does not include the part of the

business request in the pre-processed block; receive a reply message returned by the another second block chain node or the first block chain node, the reply message indicating that a business memory corresponding to the another second block chain node or the first block chain node sending the reply message stores the part of the business request; and acquire the part of the business request from the business memory corresponding to the second block chain node or the first block chain node sending the reply message.

In the embodiment of the present application, a first block chain node packages at least one business request gained from a business memory of the first block chain node into a pre-processed block, and broadcasts the pre-processed block to second block chain nodes. If finding that a business memory corresponding thereto does not include a part of the business request in the pre-processed block, a second block chain node may acquire the part of the business request from another block chain node, and carries out consensus verification on the business request included the pre-processed block by using the part of the business request and a business request stored in the business memory of its own. When a second block chain node finds, after receiving a pre-processed block broadcast by a first block chain node, that a business memory corresponding thereto does not include a part of the business requests in the pre-processed block, the second block chain node does not directly consider the pre-processed block as failed in local consensus verification. Instead, the second block chain node may acquire the missing business requests from other block chain nodes in the whole consensus network, and carry out consensus verification on the pre-processed block by using the acquired business requests and business requests stored in the business memory of its own. In this way, adverse effects on consensus verification of business requests caused by a network failure are greatly reduced, thereby improving the business processing accuracy of the whole block chain business.

In the 1990s, an improvement on a technology may be obviously distinguished as an improvement on hardware (for example, an improvement on a circuit structure such as a diode, a transistor, and a switch) or an improvement on software (an improvement on a method procedure). However, with the development of technologies, improvements of many method procedures at present may be considered as direct improvements on hardware circuit structures. Almost all designers program the improved method procedures into hardware circuits to obtain corresponding hardware circuit structures.

Therefore, it is improper to assume that the 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 whose logic functions are determined by devices programmed by a user. Designers program by themselves to "integrate" a digital system into a piece of PLD, without the need to ask a chip manufacturer to design and manufacture a dedicated integrated circuit chip. Moreover, at present, the programming is mostly implemented by using logic compiler software, instead of manually manufacturing an integrated circuit chip. The logic compiler software is similar to a software compiler used for developing and writing a program, and original code before compiling also needs to be written by using a specific programming language, which is referred to as a Hardware Description Language (HDL). There are many types of HDLs, such as Advanced Boolean Expression Language (ABEL), Altera Hardware Description Language (AHDL), Confluence, Cornell University Programming Language (CUPL), HDCal, Java Hardware Description Language (JHDL), Lava, Lola, MyHDL, PALASM, and Ruby Hardware Description Language (RHDL), among which Very-High-Speed Integrated Circuit Hardware Description Language (VHDL) and Verilog are most commonly used now. Those skilled in the art also should know that a hardware circuit for implementing the logic method procedure may be easily obtained by slightly logically programming the method procedure using the above several hardware description languages and programming it into an integrated circuit.

A controller may be implemented in any suitable manner. For example, the controller may be in the form of, for example, a microprocessor or a processor and a computer readable medium storing computer readable program code (for example, software or firmware) executable by the (micro)processor, a logic gate, a switch, an Application Specific Integrated Circuit (ASIC), a programmable logic controller, and an embedded micro-controller. Examples of the controller include, but are not limited to, the following micro-controllers: ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20, and Silicone Labs C8051F320. A memory controller may also be implemented as a part of control logic of a memory. Those skilled in the art also know that the controller may be implemented by using pure computer readable program code, and in addition, the method steps may be logically programmed to enable the controller to implement the same function in a form of a logic gate, a switch, an application specific integrated circuit,

a programmable logic controller and an embedded microcontroller. Therefore, this type of controller may be considered as a hardware component, and apparatuses included therein for implementing various functions may also be considered as structures inside the hardware component. Or, the apparatuses used for implementing various functions
5 may even be considered as both software modules for implementing the method and structures inside the hardware component.

The system, apparatus, module or unit illustrated in the above embodiments may be specifically implemented by using a computer chip or an entity, or a product having a certain function. A typical implementation device is a computer. Specifically, the
10 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 media player, a navigation device, an email device, a game console, a tablet computer, a wearable device, or a combination of any of these devices.

For ease of description, when the apparatus is described, it is divided into various
15 units in terms of functions for respective description. Definitely, when the present application is implemented, functions of the units may be implemented in the same or multiple pieces of software and/or hardware.

Those skilled 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.
20 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 a computer program product implemented on one or more computer usable storage media (including, but not limited to, a magnetic disk memory, a CD-ROM, an optical memory, and the like) including
25 computer usable program code.

The present application is described with reference to flowcharts and/or block diagrams according to 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
30 the flowcharts and/or block diagrams and combinations 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 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 implementing a specified function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

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

 These computer program instructions may also be loaded onto a computer or another programmable data processing device, such that a series of operation steps are performed on the computer or another programmable device, thereby generating computer-implemented processing. Therefore, the instructions executed on the
15 computer or another programmable device provide steps for implementing a specified function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

 In a typical configuration, a computation device includes one or more central processing units (CPUs), an I/O interface, a network interface, and a memory.

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

 The computer readable medium includes non-volatile and volatile media as well as
25 movable and non-movable media, and can implement information storage by means of any method or technology. Information may be a computer readable instruction, a data structure, and a module of a program or other data. A storage medium of a computer includes, for example, but is not limited to, a phase change memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), other
30 types of RAMs, a ROM, an electrically erasable programmable read-only memory (EEPROM), a flash memory or other memory technologies, a compact disk read-only memory (CD-ROM), a digital versatile disc (DVD) or other optical storages, a cassette tape, a magnetic tape/magnetic disk storage or other magnetic storage devices, or any

other non-transmission medium, and can be used to store information accessed by the computing device. According to the definition of this text, the computer readable medium does not include transitory media, such as a modulated data signal and a carrier.

It should be further noted that the term "include", "comprise" or other variations thereof are intended to cover non-exclusive inclusion, so that a process, method, commodity or device including a series of elements not only includes the elements, but also includes other elements not clearly listed, or further includes inherent elements of the process, method, commodity or device. In a case without any more limitations, an element defined by "including a/an..." does not exclude that the process, method, commodity or device including the element further has other identical elements.

Those skilled 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, a CD-ROM, an optical memory and the like) including computer usable program code.

The present application may be described in a common context of a computer executable instruction executed by a computer, for example, a program module. Generally, the program module includes a routine, a program, an object, an assembly, a data structure, and the like used for executing a specific task or implementing a specific abstract data type. The present application may also be implemented in distributed computing environments, and in the distributed computer environments, a task is executed by using remote processing devices connected through a communications network. In the distributed computer environment, the program module may be located in local and remote computer storage media including a storage device.

The embodiments in the specification are described progressively, identical or similar parts of the embodiments may be obtained with reference to each other, and each embodiment emphasizes a part different from other embodiments. Especially, the system embodiment is basically similar to the method embodiment, so it is described simply, and for related parts, reference may be made to the description of the parts in the method embodiment.

The above description is merely embodiments of the present application, and are not intended to limit the present application. For those skilled in the art, the present application may have various modifications and variations. Any modification, equivalent replacement, improvement or the like made without departing from the spirit
5 and principle of the present application should all fall within the scope of claims of the present application.

CLAIMS

1. A business verification method, comprising:

receiving, by a first block chain node in a whole consensus network, a business request sent by a terminal;

5 storing, at the first block chain node, the business request in a business memory corresponding to the first block chain node, and broadcasting the business request to second block chain nodes in the whole consensus network;

receiving, at some second block chain nodes, the business request broadcast by the first block chain node, wherein at least one of the second block chain nodes in the whole consensus network does not receive the business request broadcast by the first block chain node due to a network failure between the at least one of the second block chain nodes and the first block chain node;

storing the business request in respective corresponding business memories of the second block chain nodes;

15 gaining, at the first block chain node, at least one business request from the business memory of the first block chain node;

packaging, at the first block chain node, the gained at least one business request into a pre-processed block and broadcasting the pre-processed block of the at least one business request to the second block chain nodes for consensus verification;

20 receiving, by the second block chain nodes, the pre-processed block of the at least one business request;

carrying out consensus verification, at the second block chain nodes on the pre-processed block of the at least one business request according to business requests included in respective corresponding business memories and matching with the at least one business request from the business memory of the first block chain node;

25 in response to a second block chain node finding that a business memory corresponding thereto does not include a part of the at least one business request in the pre-processed block, wherein the second block chain node does not receive the part of the at least one business request due to the network failure, determining, by the second block chain node, that the pre-processed block of the at least one business request in local consensus verification is not directly failed;

acquiring, by the second block chain node, the part of the at least one business

request that is not included in the business memory corresponding thereto from other block chain nodes in the whole consensus network; and

carrying out consensus verification, at the second block chain node on the at least one business request included in the pre-processed block by using the acquired
5 part of the at least one business request and business requests stored in the business memory of the second block chain node.

2. The method of claim 1, wherein the business memories of the first block chain node and the second block chain nodes are set in form of a database that stores business requests, and wherein the business requests stored in a business
10 memory do not disappear in case of failure of a corresponding block chain node.

3. The method of claim 2, wherein the step of storing the business request in the business memory corresponding to the first block chain node specifically comprises:

sending the business request sent from the terminal to a preset distributed
15 middleware for data transmission between the block chain nodes and the business memories in the whole consensus network; and

storing the business request in the business memory corresponding to the first block chain node, according to a node identifier of the first block chain node, by using a preset distributed middleware.

4. The method of any one of claims 1 to 3, wherein the step of gaining at least one business request from the business memory specifically comprises:

gaining, from the business memory, a set number of business requests with a business type higher than a set priority.

5. The method of claim 4, wherein the step of storing the business request
25 in a business memory corresponding to the first block chain node specifically comprises:

ranking, by the first block chain node, the business request and previously received business requests according to a preset priority sequence of business types.

6. The method of claim 1, wherein the first block chain node is a leader
30 node in a consortium chain consensus algorithm, and the second block chain node is a non-leader node in the consortium chain consensus algorithm.

7. The method of claim 1, wherein the step of acquiring, by the second block chain node, the part of the at least one business request that is not included in the business memory corresponding thereto from the other block chain nodes in the whole consensus network comprises:
- 5 sending a query message for acquiring the part of the at least one business request to another second block chain node or the first block chain node when it is determined that the business memory of the second block chain node does not comprise the part of the at least one business request in the pre-processed block;
- 10 receiving a reply message returned by the another second block chain node or the first block chain node, the reply message indicating that a business memory corresponding to the another second block chain node or the first block chain node sending the reply message stores the part of the at least one business request; and
- acquiring the part of the at least one business request from the business memory corresponding to the second block chain node or the first block chain node
- 15 sending the reply message.
8. The method of claim 1, further comprising:
- after receiving the business request sent by the first block chain node, the second block chain node further broadcasting the business request to other block chain nodes in the whole consensus network;
- 20 at the other block chain nodes when receiving the business request, judging whether the business request has been previously received;
- if yes, the other block chain nodes ignore the business request; and
- if no, the other block chain nodes store the business request in business memories corresponding thereto by using a preset distributed middleware.
- 25 9. The method of claim 8, further comprising:
- during the consensus verification, after receiving the pre-processed block, verifying, by the second block chain nodes, legality of asymmetric signatures of the business requests included in the pre-processed block.
10. The method of claim 9, further comprising:
- 30 according to the asymmetric signature legality verification and a Hash integrity verification carried out for the pre-processed block, obtaining by each of the block chain nodes a verification result of its own about whether the pre-processed block as a

whole is legal;

broadcasting, by each of the block chain nodes, the verification result obtained by itself to other block chain nodes; and

5 according to verification results sent by the other block chain nodes for the pre-processed block and the verification result obtained by itself, each of the block chain nodes, obtaining an integrated verification result, about whether the pre-processed block passes the verifications, of the block chain nodes in the whole consensus network,

10 further broadcasting the obtained integrated verification result to the other block chain nodes;

after receiving the integrated verification results broadcast to each other, judging, by each of the block chain nodes in the consensus network, whether most of the integrated verification results obtained by the block chain nodes in the consensus network indicate that the verification is successful;

15 if yes, storing the business requests in the pre-processed block in a block chain of its own as blocks; and

if no, refusing the business requests in the pre-processed block.

11. A business verification apparatus comprising a plurality of modules configured to perform the method of any one of the claims 1 to 10.

20