

CLAIMS

1. A method for handling a blockchain service, the method comprising:
 - receiving, by a first application of a consensus node, a service request sent by a client,
 - 5 wherein the consensus node comprises the first application, a second application, and a database;
 - determining to-be-sent service data corresponding to the service request;
 - storing, by the first application, the to-be-sent service data in the database;
 - determining based on an identifier of the to-be-sent service data that the to-be-sent service
 - 10 data satisfies a predetermined condition that defines a data volume corresponding to the consensus node to be retrieved within a variable time interval that increases with reduced service volume and decreases with increased service volume;
 - in response to determining that the to-be-sent service data satisfies the predetermined condition, fetching, by the second application or the first application, the to-be-sent service
 - 15 data from the database;
 - fetching, by the second application, consensus data from the database, wherein the consensus data is determined, by a third application of a second consensus node, based on service data in a sent state in a second database that was stored, by the third application of the
 - second consensus node, in the second database, the consensus data being fetched, by a fourth
 - 20 application of the second consensus node, from the second database, the consensus data being sent, by the fourth application of the second consensus node, to consensus nodes other than the
 - second consensus node, the consensus data being received, by the first application of the consensus node, and stored in the database;
 - performing a consensus verification on the consensus data based on the to-be-sent service
 - 25 data in the to be sent state; and
 - sending the to-be-sent service data to a second consensus node for storage.

2. The method according to claim 1, wherein before storing, by the first application, the to-be-sent service data in the database, the method further comprises:
 - 30 determining, by the first application, that a state of the to-be-sent service data is a to be sent state; and
 - storing the to-be-sent service data and the to be sent state of the to-be-sent service data in

the database.

3. The method according to claim 1, wherein satisfying the predetermined condition comprises:

5 when a specified moment arrives, determining that the predetermined condition is satisfied.

4. The method according to claim 1, wherein fetching the to-be-sent service data from the database comprises:

10 fetching a specified amount of to-be-sent service data from the database.

5. The method according to claim 1, further comprising:
updating a state of sent service data in the database to a sent state.

15 6. The method according to claim 1, further comprising:
~~receiving, by the first application, consensus data sent by the second consensus node; or~~
receiving, by the first application, a verification result sent by the second consensus node,
and using the verification result as the consensus data.

20 ~~7. The method according to claim 6, further comprising:
fetching, by the second application, the consensus data from the database and service data
in the to be sent state in the database.~~

25 ~~8. The method according to claim 7, further comprising:
performing a consensus verification on the consensus data based on the to be sent service
data in the to be sent state, and determining the verification result.~~

30 ~~9. The method according to claim 8, wherein the second consensus node comprises a
the third application, a the fourth application, and a the second database.~~

~~10. The method according to claim 9, wherein before receiving, by the first application,
consensus data, the method further comprises:~~

~~determining, by the third application, the consensus data based on service data in a sent
state in the second database, and storing the consensus data in the second database; and~~

~~fetching, by the fourth application, the consensus data from the second database, and sending the consensus data to consensus nodes other than the second consensus node.~~

118. The method according to claim 81, wherein performing the consensus verification is
5 based on a hash value and a content of the to-be-sent service data.

129. The method according to claim 81, wherein the consensus verification comprises a Practical Byzantine Fault Tolerance algorithm.

10 1310. The method according to claim 1, wherein the to-be-sent service data comprises at least one of a service attribute, a service identifier, a data version, a time stamp, service information, and a state identifier.

1411. An apparatus for handling a blockchain service, the apparatus comprising a plurality
15 of modules configured to perform the method of any one of claims 1 to 1310.