## CLAIMS

What is claimed is:

1. A method for configuring a plurality of block chain-based local consensuses to run synchronically, the method comprising:

implementing an initialization of a plurality of nodes;

creating a plurality of local consensus instances each for a specific set of nodes selected from the plurality of nodes according to a service demand running on each of the plurality of local consensus instances;

for each of the plurality of local consensus instances, selecting a type of consensus algorithm; and

creating, by each of the plurality of local consensus instances, a plurality of communication channels according to the type of consensus algorithm used by the corresponding local consensus instance, each communication channel having a channel ID, wherein only channels with the same channel ID can communicate with each other.

2. The method of claim 1, wherein creating the plurality of local consensus instances each for a specific set of nodes selected from the plurality of nodes further comprises:

for each of the plurality of local consensus instances, broadcasting a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and

all the nodes in the block chain responding to each of the first transaction commands to create each of the plurality of local consensus instances.

3. The method of claim 2, wherein the first transaction command for each of the plurality of local consensus instances includes information indicating a node list of the set of nodes, information related to a local consensus identifier specified by an administrator of the block chain, and information related to a consensus algorithm used by each of the plurality of created local consensus instances.

4. The method according to claim 3, wherein the consensus algorithm comprises at least one of the following algorithms:

proof of work consensus algorithm;

proof of stake consensus algorithm;

delegated proof of stake consensus algorithm;

Practical Byzantine Fault Tolerance consensus algorithm; and/or

BFT-RAFT consensus algorithm.

5. The method according to claim 3, wherein the information related to a consensus algorithm used by each of the plurality of created local consensus instances is determined based on the service demand running on each of the local consensus instances to be created and the consensus algorithms supported by the block chain.

6. The method of claim 1, wherein the method further comprises:

a message processing thread for receiving and processing message transmitted from other nodes being initiated by each of the plurality of created local consensus instances.

7. The method according to claim 1, wherein the method further comprises:

for each of the plurality of local consensus instances, broadcasting a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and

all the nodes in the block chain responding to each of the second transaction commands to delete each of the plurality of local consensus instances.

8. The method of claim 1, wherein implementing an initialization of a plurality of nodes further comprises:

configuring network addresses and ports for the plurality of nodes;

generating node public and private keys; and

initiating and running a node block chain program.

9. A tangible computer-readable storage medium for configuring a plurality of block chain-based local consensuses to run synchronically, the storage medium comprising instructions that, when executed, cause a processor of the computer to at least:

implementing an initialization of a plurality of nodes;

creating a plurality of local consensus instances each for a specific set of nodes selected from the plurality of nodes according to a service command running on each of the plurality of local consensus instances;

for each of the plurality of local consensus instances, selecting a type of consensus algorithm; and

creating, by each of the plurality of local consensus instances, a plurality of communication channels according to the type of consensus algorithm used by the corresponding local consensus instance, each communication channel having a channel ID, wherein only channels with the same channel ID can communicate with each other.

10. The computer-readable storage medium of claim 9, wherein to create a plurality of local consensus instances each for a specific set of nodes selected from the plurality of nodes further comprises:

for each of the plurality of local consensus instances, broadcast a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and

all the nodes in the block chain responding to each of the first transaction commands to create each of the plurality of local consensus instances.

11. The computer-readable storage medium of claim 9, wherein when executed, the instructions cause the processor of the computer further to:

for each of the plurality of local consensus instances, broadcast a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and

cause all the nodes in the block chain to respond to each of the second transaction commands to delete each of the plurality of local consensus instances.

12. An apparatus for configuring a plurality of block chain-based local consensuses to run synchronically, the apparatus comprising:

an initialization module configured to implement an initialization of a plurality of nodes; and

a local consensus configuration module configured to:

create a plurality of local consensus instances each for a specific set of nodes selected from the plurality of nodes according to a service command running on each of the plurality of local consensus instances;

selecting a type of consensus algorithm for each of the plurality of local consensus instances; and

creating, by each of the plurality of local consensus instances, a plurality of communication channels according to the type of consensus algorithm used by the corresponding local consensus instance, each communication channel having a channel ID, wherein only channels with the same channel ID can communicate with each other.

13. The apparatus of claim 12, wherein the local consensus configuration module is further configured to:

for each of the plurality of local consensus instances, broadcast a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and

cause all the nodes in the block chain to respond to each of the plurality of first transaction commands to create each of the plurality of local consensus instances.

14. The apparatus of claim 13, wherein the first transaction commands for each of the plurality of local consensus instances comprises information indicating a node list of the set of nodes, information related to a local consensus identifier specified by an administrator of the block chain, and information related to a consensus algorithm used by each of the plurality of created local consensus instances.

15. The apparatus of claim 14, wherein the consensus algorithm comprises at least one of the following algorithms:

proof of work consensus algorithm;

proof of stake consensus algorithm;

delegated proof of stake consensus algorithm;

Practical Byzantine Fault Tolerance consensus algorithm; and/or

BFT-RAFT consensus algorithm.

16. The apparatus of claim 14, wherein the information related to a consensus algorithm used by each of the plurality of created local consensus instances is determined based on the service demand running on each of the local consensus instances to be created and the consensus algorithms supported by the block chain.

17. The apparatus of claim 12, wherein the local consensus configuration module is further configured to:

for each of the plurality of local consensus instances, broadcast a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and

cause all the nodes in the block chain to respond to each of the second transaction commands to delete each of the plurality of local consensus instances.