

Claims

1. A method of operating a distributed database in a data communication network, the network comprising at least one, transmitting, first node and at least one second node;
- 5 the first node comprising:
- a memory in which at least some data records of the distributed database is stored;
- a set of at least one, predetermined and pre-set, state machine;
- a filter connected to the memory, to filter predetermined content data out of a
- 10 data record of the database, and to provide filtered data identifying a current state
- of the filtered data record, and
- an integrated circuit processor arranged to run state transition software to
- manipulate data records of the distributed database, which integrated circuit
- processor is connected to the memory and all the state machines of the set;
- the second node comprising:
- 15 a register memory in which a register is stored with data which identifies whether
- or not the current state of the data record is allowed to be modified or not;
- a set of verification state machines, each state machine of the first node being
- equal to a verification state machines, such that the set of verification state
- machines comprises, or consists of, equals for all the state machines of the set of
- 20 the first node, and
- an integrated circuit processor arranged to run verification software to verify an
- intended state transition against a predetermined set of at least one verification
- rule defined by a selected verification state machine selected out of the set of
- verification state machines;
- 25 the method comprising:
- the integrated circuit processor of the first node running the state transition software, the
- running comprising:
- inputting a selected data record into a selected state machine selected out of the
- set of state machines and thereby generating the intended state transition of the
- 30 selected data record from a current state of the selected data record to a new state
- of the selected data record in accordance with rules and constraints of the selected
- state machine;
- the integrated circuit processor of the first node passing the data record in the
- current state through the filter to obtain state identification data representing an

- identifier unique for the current state of the selected data record, with a content of the selected data record filtered out of the identifier by the filter;
calculating verification data with a one-way function of which the input variables comprise the current state and the new state;
- 5 transmitting to the second node a verification request message without the input variables, the verification request message containing: the state identification data, state machine identification data representing an identification of the selected state machine, and verification data;
- 10 receiving at the second node the verification request message, and in response thereto the integrated circuit processor in the second node running the verification software to verify the intended state transition for compliance with the rules of the selected state machine, the running comprising:
- determining the verification state machine equal to the selected state machine using the state machine identification data;
- 15 verifying with the verification data whether or not the intended state transition meets the rules and constraints of the equal verification state machine; and if the state transition meets the rules and constraints of the equal verification state machine, outputting a confirmation message to the transmitting first node; and the method further comprising, when the confirmation message is outputted:
- 20 the first node in response to receiving the confirmation message: accepting the intended state transition and adjusting the data record according to the intended state transition.
2. A method of manipulating data records in a distributed database by a first node in a network
- 25 comprising said first node and at least one second node, the first node comprising:
- a memory in which at least some data records of the distributed database is stored;
- a set of at least one, predetermined and pre-set, state machine;
- a filter connected to the memory, to filter predetermined content data out of a data record of the database, and to provide filtered data identifying a current state of the filtered data
- 30 record, and
- an integrated circuit processor arranged to run state transition software to manipulate data records of the distributed database, which integrated circuit processor is connected to the memory and all the state machines of the set;
- the method comprising:

the integrated circuit processor running the state transition software, the running comprising:

5 inputting a selected data record into a selected state machine and thereby
 generating an intended state transition of the selected data record from a current
 state of the selected data record to a new state of the selected data record in
 accordance with rules and constraints of the selected state machine;
 the integrated circuit processor of the first node passing the data record in the
 current state through the filter to obtain the state identification data representing
 an identifier of the current state of the selected data record with a content of the
10 selected data record filtered out of the identifier by the filter;
 calculating verification data with a one-way function of which the input variables
 comprise the current state and the new state,
 transmitting to the second node a verification request message without the input
 variables, the verification request message containing: the state identification data,
15 state machine identification data representing an identification of the selected
 state machine, and the verification data;

the method further comprising:

 in response to receiving a confirmation message from the second node, the integrated
 circuit processor executing instructions to accept the intended state transition and
20 adjusting in the memory the data record according to the accepted state transition, or else
 rejecting the intended state transition and maintaining the data record unchanged.

3. A method of manipulating data records in a distributed database by a recipient first node in a
network comprising at least two first nodes and at least one second node, the first nodes
25 comprising a transmitting first node and the recipient first node, the transmitting node and the
 recipient node having stored thereon corresponding data records of the distributed database;

the recipient first node comprising:

 a memory in which at least some of the data records of the distributed database are
 stored,
30 a set of at least one, predetermined and pre-set, state machine, and
 an integrated circuit processor arranged to run software to manipulate the data
 records, the software including instructions to verify an intended state transition of
 a data record against a set of predetermined rules and constraints defined by a

selected state machine selected from the set, the integrated circuit processor connected to the memory and all the state machines of the set;

the method comprising:

- 5 the recipient first node receiving an intended transition message from the transmitting first node, the intended transition message informing the recipient node of an intended state transition to a selected data record, and indicating a selected state machine to be used; the integrated circuit processor of the recipient first node running the software, and verifying whether the intended state transition meets the rules and constraints of the selected state machine;
- 10 if the state transition meets the rules and constraints: outputting an approval message to the transmitting first node and/or the second node, and else outputting an error message to the transmitting first node and/or the second node; and
- if a confirmation message is received from the second node, then, in response to receiving the conformation message, the integrated circuit processor executing instructions to accept
- 15 the intended state transition and adjusting in the memory the data record according to the accepted state transition, and else rejecting the intended state transition and maintaining the data record unchanged.

4. A method of checking changes to data records of a distributed database in a first node of a network, the method performed by a second node of the network,
- 20

the second node comprising:

- a register memory in which a register is stored with data which identifies whether or not a current state of a data record is allowed to be modified;
- 25 a set of verification state machines, each state machine of the first node being equal to a verification state machines, such that the set of verification state machines comprises, or consists of, equals for all the state machines of the set of the first node, and
- an integrated circuit processor arranged to run verification software to verify an intended state transition of the data record from the current state to a new state
- 30 against a predetermined set of at least one verification rule defined by a verification state machine;

the method comprising:

receiving at the second node a verification request message without the contents of the current state or of the new state, the verification request message containing:

- state identification data representing an identifier of the current state, state machine identification data representing an identification of a selected state machine used by the first node, and verification data calculated with a one-way function of which the input variables comprise the current state and the new state;
- 5 in response to the receiving, the integrated circuit processor running the verification software to verify the intended state transition for compliance with the rules of the selected state machine, the running comprising:
- selecting a verification state machine equal to the selected state machine using the state machine identification data;
- 10 verifying with the verification data whether or not the intended state transition meets the rules and constraints of the equal verification state machine; and
- if the state transition meets the rules of the equal verification state machine, outputting a confirmation message to the first node.
- 15 5. The method of one or more of the preceding claims, comprising determining by the integrated circuit processor in the second node whether the state identification data corresponds to a current state identified by the data in the register as being allowed to be modified, and if the state does not correspond rejecting the intended state transition.
- 20 6. The method of one or more of the preceding claims, wherein the second node outputs a rejection message to the first node if the state transition does not meet the rules, and in response to receiving the rejection message the first node rejects the intended state transition and maintains the data record unchanged.
- 25 7. The method of one or more of the preceding claims, wherein the network comprises at least two first nodes and wherein the set of verification state machines of second node comprises an equal verification state machine for all state machines of all the first nodes.
8. The method of one of more of the preceding claims, wherein each first node only stores the data records of the distributed database for which the first node is pre-set, or requested, to
- 30 perform operations.
9. The method of one or more of the preceding claims, wherein the network comprises at least two first nodes and wherein the first nodes are communicatively connected to exchange

messages about state transition to the data records to synchronize corresponding data records between them.

- 5 10. The method of claims 3 and one or more of the other preceding claims, comprising, the transmitting first node transmitting the intended transition message to the recipient first node when the rules and constraints of the selected state machine impose approval of the recipient first node to the intended state transition, and wherein the verification message includes a signature of the recipient node for the intended transition message.
- 10 11. The method of one or more of the preceding claims, wherein the network is a peer-to-peer network.
12. The method of one or more of the preceding claims, wherein the network is a private network.
- 15 13. The method of one or more of the preceding claims, wherein the confirmation message further includes a time-stamp.
14. The method of one or more of the preceding claims, wherein the database represents an unused state transition output data model.
- 20 15. The method of one or more of the preceding claims, wherein the intended transition message is only transmitted to selected first nodes, selected by the transmitting first node based on the intended state transition.
- 25 16. The method of one or more of the preceding claims, wherein the database is a relational database.
17. The method of one or more of the preceding claims, wherein the verification data comprise a transition identification value unique for the state transition, such as a cryptographic hash value.
- 30 18. The method of claim 17, wherein the transition identification value is unique for the current state, for example a vector commitment, such a Merkle tree root hash value calculated from a Merkle tree of which at least one leaves is the current state.

19. The method of claim 18, wherein the Merkle tree has at least three leaves, the leaves comprising current state, new state, digital signatures of the state transition.
- 5 20. The method of claim 17 or 18, wherein the verification data comprise a zero-knowledge proof determined in accordance with a predetermined zero-knowledge protocol, the protocol having as secret witness parameter the intended state transition.
- 10 21. The method of claim 20, wherein the second node verifies with the proof using the equal verification state machine whether or not the Merkle tree root hash value has been calculated from the intended state transition, in accordance with the verification rules of the zero-knowledge protocol.
- 15 22. The method of claim 20 or 21, wherein the second node verifies with the proof whether or not the state transition satisfies the rules and constraints of the selected state machine, and the second node verifies the proof with the verification state machine in accordance with verification rules of the zero-knowledge protocol.
- 20 23. The method of one or more of claims 20-22, wherein the verification request message comprises the transition identification value as public input parameter to the verifier part of the zero-knowledge protocol, and the second node uses the transition identification value as public input parameter in the verification.
- 25 24. The method of one or more of claims 20-23, wherein the zero-knowledge protocol is a zero-knowledge succinct, non-interactive, argument of knowledge protocol.
- 30 25. The method of one or more of claims 20-24, wherein the selected state machine has an interface which is coupled by the integrated circuit processor of the first node to a proof generating machine to generate the proof and wherein the equal verification state machine has an interface which is coupled by the integrated circuit processor of the second node to a verifier machine to verify the proof.
26. The method of one or more of the preceding claims, wherein:

the verification data comprise one or more digital signatures, each signed with a private key of a public-private key combination, as required by the rules and constraints of the selected state machine for the intended state transition;

the second node is provided with public keys corresponding to the private keys of the public-private key combinations used in the digital signatures;

and when verifying the intended state transition, the second node determines from the verification state machine the digital signatures required for the intended state transition, decrypts the digital signature using the public key and determines whether or not all digital signatures decrypt to the same digital message.

10

27. The method of claim 18 and 26, wherein the digital message is derived from the root hash of the Merkle tree, and for example a hash calculated from the root hash.

28. The method according to any one of the preceding claims, wherein the verification data comprises a zero-knowledge proof determined in accordance with a predetermined zero-knowledge protocol, wherein preferably the one way function comprises a zero knowledge verification function.

29. The method according to any one of the preceding claims, wherein the verification data does not disclose the input variables while the verification data enabling to verify that the current state and the new state satisfy the selected state machine.

30. A first node for a network in which a distributed database is maintained by comprising at least one first node, and at least one second node verifies intended state changes, the first node comprising:

a memory in which at least some data records of the distributed database is stored;
a set of at least one, predetermined and pre-set, state machine;
a filter connected to the memory, to filter predetermined content data out of a data record of the database, and to provide filtered data identifying a current state of the filtered data record, and
an integrated circuit processor arranged to run state transition software to manipulate data records of the distributed database, which integrated circuit processor is connected to the memory and all the state machines of the set;

30

the state transition software comprising instructions executable by the integrated circuit processor to:

5 input a selected data record into a selected state machine and thereby generate an intended state transition of the selected data record from a current state of the selected data record to a new state of the selected data record in accordance with rules and constraints of the selected state machine;

10 pass the data record in the current state through the filter to obtain state identification data representing an identifier of the current state of the selected data record with a content of the selected data record filtered out of the identifier by the filter;

15 calculate verification data with a one-way function of which the input variables comprise the current state and the new state,

 transmit to the second node a verification request message, without the input variables, the verification request message containing: the state identification data, state machine identification data represent an identification of the selected state machine, and the verification data;

the state transition software comprising instructions executable by the integrated circuit processor to:

20 in response to receiving a confirmation message from the second node, accept the intended state transition and adjust in the memory the data record according to the accepted state transition, or to else reject the intended state transition and maintain the data record unchanged.

31. The first node according to claim 30, wherein the verification data comprises a zero-
25 knowledge proof determined in accordance with a predetermined zero-knowledge protocol, wherein preferably the one way function comprises a zero knowledge verification function.

32. The first node according to claim 30 or 31, wherein the verification data does not disclose the input variables while the verification data enabling to verify that the current state and the new
30 state satisfy the selected state machine.

33. A second node for a network comprising a first node and the second node, the second node comprising:

- a register memory in which a register is stored with data which identifies whether or not a current state of a record is allowed to be modified;
- a set of verification state machines, each state machine of the first node being equal to a verification state machines, such that the set of verification state machines comprises, or
- 5 consists of, equals for all the state machines of the set of the first node, and
- an integrated circuit processor arranged to run verification software to verify the intended state transition against a predetermined set of at least one verification rule defined by a verification state machine;
- the verification software comprising instructions executable by the integrated circuit processor to:
- 10 run the verification software in response to receiving at the second node a verification request message without the input variables, the verification request message at least containing: a state identification data, state machine identification data representing an identification of a selected state machine used by the first node, and verification data, the running comprising:
- 15 selecting a verification state machine equal to the selected state machine using the state machine identification data;
- verifying with the verification data whether or not the state transition meets rules and constraints of the verification state machine; and
- if the state transition meet the rules of the equal verification state machine,
- 20 outputting a confirmation message to the first node.

34. The second node according to claim 33, wherein the verification data comprises a zero-knowledge proof determined in accordance with a predetermined zero-knowledge protocol.

- 25 35. The second node according to claim 33 or 34, wherein the verification data does not disclose the input variables while the verification data enabling to verify that the current state and the new state satisfy the selected state machine.

36. A network comprising at least one first node as claimed in any one of claims 30 – 32 and at
- 30 least one second node as claimed in any one of claims 33 - 35, the first and second node being communicatively connected to each other.

37. The network of claim 36, wherein each node has a separate network address, such as a separate IP address.

38. A first node as claimed in any one of claims 30 – 32, a second node as claimed in any one of claims 33 - 35 or the network of one or more of claims 36 - 37, for use, or being used, in the method of one or more of claims 1 - 29.

5

39. A computer program product comprising code portions for performing some or all steps of the method of one or more of claims 1 - 29, when run on a programmable apparatus or representing transition software or verification software for the method of one or more of claims 1 - 29.

10

40. A non-transitory, tangible data carrier readable by a programmable apparatus, with data stored thereon representing code portions for performing some or all steps of the method of one or more of claims 1 - 29 when executed by a programmable apparatus, or representing transition software or verification software for the method of one or more of claims 1 - 29.

15

41. A non-transitory, tangible data carrier readable by a programmable apparatus, with data stored thereon representing records of a distributed database obtained with, or obtainable by the method of one or more of claims 1 - 29.