Claims

1. A computer implemented method of providing a data structure for storing and/or managing data associated with one or more transactions pertaining to a distributed ledger, the method implemented by one or more processors associated with a data management system (DBMS) for the distributed ledger, the method comprising the steps of:

creating a database transaction type, said database transaction type comprising:

an identifier for a database D that is to be used to obtain data for one or more transactions pertaining to a distributed ledger; and

one or more table transaction identifiers (TTxID), wherein a given TTxID among the one or more TTxIDs is specific to a given table among one or more tables associated with the identified database D, wherein the given TTxID relates to a transaction pertaining to a distributed ledger for the given table; and

creating a table transaction type, said table transaction type comprising:

an identifier for a table T, the table being among the one of more tables associated with the identified database D; and

one or more entries associated with the identified table T.

2. The method as claimed in claim 1 wherein the table transaction type further comprises:

one or more entry transaction identifiers (ETxID), wherein a given ETxID among the one or more ETxIDs being specific to a given entry among the one or more entries associated with the identified table T, and wherein the given ETxID relates to a transaction pertaining to the distributed ledger for the given entry;

the method further comprising:

creating an entry transaction type, the entry transaction type comprising:

one or more data pairs, wherein each data pair relates to an entry in the identified table T, and wherein each data pair comprises a reference R identifying a position in the identified table, and a value VL for the identified position.

3. The method as claimed in claims 1 or 2 wherein the transaction relating to the one or more TTxIDs or ETxIDs is associated with one or more unspent transaction outputs (UTXOs).

4. The method as claimed in claim 3 wherein the UXTO for the given transaction includes an output script that is spendable by a subsequent transaction associated with the identified database D and/or table T.

5. The method as claimed in any one of claims 3 or 4 wherein the UTXO for the given transaction includes a non-spendable output that is associated with the spendable output script for the given transaction.

6. A data structure for storing and/or managing data associated with one or more transactions pertaining to a distributed ledger, the data structure being managed by one or more processors associated by a database management system (DBMS) for the distributed ledger, the data structure including a data transaction type and a table transaction type created according to the method according to any one of claims 1 to 5.

7. The data structure as claimed in claim 6 wherein the data structure is a nested transaction storage structure , wherein the data structure further comprises an entry data type created according to the method of any one of claims 2 to 5, said data structure configured to store the identified database D based on storing TTxIDs and ETxIDs associated with the identified database D.

8. A computer implemented method for implementing one or more transactions associated with a distributed ledger, the transaction using the data structure provided by one of claims 6 or 7, the method implemented by one or more processors associated with a data management system (DBMS) and comprising the steps of:

creating a database including the steps of :

generating a first transaction to create a first database, the first transaction based on the database transaction type of the data structure; and

providing an output for the first transaction based on the created first database;

responsive to a request or a transaction associated with the first database; updating the first database including the steps of :

generating a second transaction to create a first table, the second transaction based on the table transaction type of the data structure; and

providing an output for the second transaction based on the created first table, and including an identifier of the second transaction (TTxID);

generating a third transaction based on the output of the second transaction including the steps of:

updating the database transaction type of the first database with the identifier associated with the second transaction (TTxID); and

providing an output for the third transaction based on the updated first database.

9. The method as claimed in claim 8 further comprising the steps of :

responsive to a request or a transaction associated with the first table, populating the first table including the steps of:

generating a fourth transaction to create a first entry in the first table, the fourth transaction based on the entry transaction type of the data structure; and

providing an output for the fourth transaction based on the created entry, and including an identifier of the fourth transaction (ETxID);

generating a fifth transaction based on the output of the fourth transaction including the steps of:

updating the table transaction type of the first table with the identifier associated with the fourth transaction (ETxID); and

providing an output for the fifth transaction, the output based on the updated first table, and an identifier of the fifth transaction (TTxID); and

generating a sixth transaction based on the output of the fifth transaction including the steps of:

updating the database transaction type of the first database with the identifier associated with the fifth transaction (TTxID); and

providing an output for the sixth transaction based on the updated first database.

10. The method as claimed in claim 9 comprising the steps of :

responsive to a request or a transaction associated with the first table, modifying the first table including the steps of:

generating a seventh transaction to modify the reference and/or the value of the data pair associated with the first entry in the first table;

updating the entry transaction type of the first entry with a modified entry based on the seventh transaction; and

providing an output for the seventh transaction based on the modified entry, and including an identifier of the seventh transaction (ETxID);

generating an eighth transaction based on the output of the seventh transaction including the steps of:

updating the table transaction type of the first table with the identifier associated with the seventh transaction (ETxID); and

providing an output for the eighth transaction based on the updated first table, and including an identifier of the eighth transaction(TTxID); and

generating a ninth transaction based on the output of the eighth transaction including the steps of:

updating the database transaction type of the first database with the identifier associated with the eighth transaction (TTxID); and

providing an output for the sixth transaction based on the updated first database.

11. The method as claimed in any one of claims 9 or 10 comprising the steps of:

responsive to a request or a transaction associated with the first table, modifying the first table including the steps of:

generating a tenth transaction to delete the first entry in the first table, including the steps of:

updating the table transaction type of the first table to remove the identifier associated with the fourth transaction (ETxID); and

providing an output for the tenth transaction based on the updated first table, and including an identifier of the tenth transaction (TTxID); and

generating an eleventh transaction based on the output of the tenth transaction including the steps of:

updating the database transaction type of the first database with the identifier associated with the tenth transaction (TTxID); and

providing an output for the eleventh transaction, the output including an identifier of the updated first database.

12. A computer implemented method for implementing one or more transactions associated with a distributed ledger, the transaction using the data structure provided by any one of clams 6 or 7, the method implemented by one or more processors associated with a client entity and comprising the steps of:

sending a request to one or more processors associated with a DBMS for creating a first database;

monitoring a distributed ledger for at least one transaction output associated with the created first database D;

sending a request to the one or more processors associated with the DBMS for creating a first table for the first database;

monitoring the distributed ledger for at least one transaction output associated with the first table (TTxID); and

sending a request to the one or more processors associated with the DBMS for updating the first table with a first entry, wherein the first entry includes a data pair comprising a reference identifying a position in the first table, and a value for the identified position.

13. A computer implemented method for implementing one or more transactions associated with a distributed ledger, the transaction using the data structure provided by any one of clams 6 or 7, the method implemented by one or more processors associated with a client entity and comprising the steps of:

monitoring the distributed ledger for at least one transaction output associated with a first entry (ETxID) in a first table of a first database;

sending a request to one or more processors associated with a DBMS for modifying the first entry, the request including a request to modify a reference and/or a value of the data pair associated with the first entry;

monitoring a distributed ledger for at least one transaction output associated with an identifier for an updated first table (TTxID) and/or an updated first database D incorporating the modified first entry.

14. A computer implemented method for implementing one or more transactions associated with a distributed ledger, the transaction using the data structure provided by any one of clams 6 or 7, the method implemented by one or more processors associated with a client entity and comprising the steps of:

monitoring the distributed ledger for at least one transaction output associated with an identifier for a first table (TTxID) of a first database;

sending a request to one or more processors associated with a DBMS for deleting a first entry of the first table;

monitoring a distributed ledger for at least one transaction output associated with an identifier for an updated first table (TTxID) and/or an updated first database.

15. A computer implemented method for managing access to a data management system (DBMS), the DBMS managing data pertaining to transactions associated with a distributed ledger, the data being provided in a database, the database using the data structure of claims 6 or 7, the method comprising the step of :

responsive to a request for registration with the DBMS, creating a record associated with a given user, the record including an identifier for the given user among a plurality of users;

obtaining a public key P associated with the given user; the public key being part of a cryptographic key pair associated with the given user, the cryptographic key pair including a private key V for the given user;

obtaining or assigning one or more attributes associated with the given user, wherein each attribute is associated with a setting relating to access to a database managed by the DBMS;

updating the record based on the obtained public key P and the one or more attributes associated with the given user; and

storing or providing the updated record in association with the DBMS.

16. The method as claimed in claim 15 wherein the setting for a given attribute confirms if the given attribute is associated with permission to perform one or more of a read, write, modify or delete operation associated with the database.

17. The method as claimed in any one of claims 15 or 16 wherein the setting for each attribute is provided in the request, or the setting is obtained from a computing resource associated with the DBMS, or the setting is predetermined for each attribute for the database managed by the DBMS.

18. A computer implemented method for implementing a data management system (DBMS) for managing a database pertaining to transactions associated with a distributed ledger, the database using the data structure of claims 6 or 7, the method comprising the steps of :

obtaining a request for an operation associated with the database from a client entity, the client entity associated with a given user among a plurality of users, wherein the request includes a digital signature associated with the given user;

determining that the given user is authorised to access the database;

responsive to successful determination, verifying that the digital signature is associated with the given user based on a user record associated with the DBMS; and

responsive to successful verification, generating a message based on a current instance of the user record and the request.

19. The method as claimed in claim 18 wherein the request is a request to read, write, modify or delete data in the database.

20. The method as claimed in any one of claims 15 to 19 wherein the database is implemented using the data structure as provided by any one of claims 6 or 7.

21. The method as claimed in any one of claims 18 to 20 wherein the user record is created upon registration of the given user with the DBMS according to the method of any one of claims 15 to 17.

22. The method as claimed in claim 21 wherein the step of determining that the user is authorised includes determining that a user record associated with the DBMS is present for the given user upon registration of the given user with the DBMS.

23. The method as claimed in any one of claims 21 or 22 wherein the step of verifying the digital signature is based on determining that the private key V or a signing key K used to sign the request is associated with the public key P associated with the given user in the user record.

24. The method as claimed in any one of claims 18 to 23 comprising, responsive to the determination being unsuccessful, generating an error notification to specify that the given user is not authorised or is not registered to access the database.

25. The method as claimed in any one of claims 18 to 23 comprising, responsive to the verification of the digital signature being unsuccessful, generating an error notification to specify that the signature applied to the request is not valid for the given user.

26. A computer implemented method for registering with a data management system (DBMS), the DBMS associated with data pertaining to transactions associated with a distributed ledger, the method being implemented by one or more processors

associated with a client entity, the client entity associated with a given user among a plurality of users, the method comprising the steps of :

generating a request for registering with the DBMS, the request associated with the given user;

providing or accessing one or more attributes associated with the given user, each attribute associated with a setting relating to the database managed by the DBMS, the database using the data structure of claims 6 or 7;

generating a private key V associated with the given user, the private key being part of a cryptographic key pair that includes a public key P for the given user;

calculating the public key P based on the private key V; and

sending the public key P and the one or more attributes to the DBMS.

27. A computer implemented method for accessing data associated with a data management system (DBMS), the DBMS associated with data pertaining to transactions associated with a distributed ledger, the method being implemented by one or more processors associated with a client entity, the client entity associated with a given user among a plurality of users, the method comprising the steps of :

generating a request associated with a database, the database using the data structure of claims 6 or 7, the request associated with the given user, and the database managed by the DBMS;

generating a digital signature based on a private key V for the given user, the private key being part of a cryptographic key pair that includes a public key P for the given user; the public key P being provided to the DBMS;

applying the digital signature to the request; and

sending the signed request to the DBMS.

28. The method as claimed in claim 27 comprising, generating a signing key K based on the private key V, wherein the signing key is an ephemeral key.

29. The method as clamed in claim 27 or 28 including the method of registration with the DBMS as claimed in claim 26.

30. A computer implemented method for implementing a data management system (DBMS) for managing a database pertaining to transactions associated with a distributed ledger, the method comprising the steps of :

receiving a message associated with a current instance of a user record for a given user among one or more users, and a request associated with the database, the request being provided by the given user;

parsing the request associated with a database, including:

retrieving one or more attributes associated with the given user from the current instance;

verifying that the setting for each of the retrieved attributes is valid for the request made by the given user;

responsive to successful verification, performing one or more operations associated with the database based on the request, wherein the database is based on a data structure provided according to any one of claims 6 or 7, the step of performing further including:

if the request is associated with a read operation, obtaining one or more transaction identifiers (TTxID, ETxID) related to the request , each obtained transaction identifier pertaining to a transaction, whereby the data for said transaction is associated with the data structure; and

accessing the data relating to the obtained transaction identifier (TTxID, ETxID) from the data structure;

if the request is associated with a write or modify or delete operation, then in addition to the step of obtaining, the method further includes generating one or more transactions for spending or processing the transaction(s) associated with the obtained one or more transaction identifiers (TTxID, ETxID) based on the operation; and

updating the data structure with identifiers relating to the generated transactions.

31. The method as claimed in claim 30 wherein the step of verifying includes checking that the setting for a given attribute among the one or more attributes indicates that the given user has permission to access to the database for the request made.

32. The method as claimed in any one of claims 30 or 31 wherein, if the request is associated with a write, modify or delete operation, then the step of performing further includes the method as claimed in any one of claims 8 to 11.

33. The method as claimed in any one of claims 30 to 32, wherein if the request is associated with a write, modify or delete operation, the step of generating further includes:

obtaining an output script associated with each obtained transaction identifier (TTxID, ETxID);

constructing a further transaction based on the output script, the further transaction based a private key V associated with the given user, the private key being part of a cryptographic key pair that includes a public key P for the given user; and

providing the further transaction to the distributed ledger, the further transaction associated with a further transaction identifier (TTxID, ETxID).

34. The method as claimed in claim 33, wherein the step of constructing the further transaction is performed by a Simplified Payment Verification (SPV) client associated with the DBMS, wherein the SPV client communicatively couples the DBMS to the distributed ledger via a communication network.

35. The method as claimed in any one of claims 30 to 34, wherein each obtained transaction identifier (TTxID, ETxID) is associated with a transaction that includes a non-spendable UTXO, said non-spendable UTXO associated with or including a spendable payload, the spendable payload being a spendable UTXO.

36. The method as claimed in claim 35 wherein each spendable and non-spendable UTXO is stored in a UTXO set or memory associated with the DBMS, wherein the non-spendable UTXO is an OP_RETURN output script.

37. The method as claimed in claim 36 wherein one or more spendable UTXOs in the UTXO set or memory are used for constructing the further transaction.

38. A computing device comprising:

a processor; and

memory including executable instructions that, as a result of execution by the processor, causes the device to perform the computer-implemented method of one of claims 12 to 14 or 26 to 29;

wherein the computing device is a client entity.

39. A DBMS system for managing access to data associated with one or more transactions pertaining to a distributed ledger, wherein the data managed by the DBMS is stored in a data structure as provided by any one of claims 6 or 7, the DBMS comprising:

a registration and access management system having one or more processors configured to execute instructions that to perform the computer-implemented method of one of claims 15 to 25;

a transaction execution system having one or more processors configured to execute instructions to perform the computer-implemented method of one of claims 30 to 37;

one or more registers for storing a plurality of transaction identifiers associated with the database; and

a UTXO set or a memory for storing a copy of a plurality of UTXOs associated with the database.

40. A system comprising:

one or more computing devices as claimed in claim 38, each being a client entity;

a DBMS as claimed in claim 39; and

a communication network for communicatively coupling each client entity with the DBMS.