What is claimed is:

1. An apparatus, comprising:

a memory of a compute device associated with a distributed database implemented by a plurality of compute devices via a network operatively coupled to the plurality of compute devices; and

a processor operatively coupled to the memory, and configured to:

receive, from a compute device from the plurality of compute devices, a state proof associated with a state of the distributed database, the state proof including:

data associated with the state,

a timestamp associated with the state,

a first identifier of the distributed database, and

a set of address books associated with the distributed database, each address book from the set of address books associated with a version of the distributed database during a time period different from a time period associated with the version of the distributed database associated with each remaining address book from the set of address books, the set of address books having a chronological order; and

determine validity of the data at a time associated with the timestamp by:

verifying that the first identifier of the distributed database is correct based on a second identifier of the distributed database stored in the memory,

verifying that the data associated with the state has been digitally signed by a predetermined number of compute devices from the plurality of compute devices, and

other than an initial address book from the set of address books, verifying that each address book from the set of address books is digitally signed by a predetermined number of compute devices from a set of compute devices associated with an immediately preceding address book in the chronological order and from the set of address books.

2. The apparatus of claim 1, wherein the processor is configured to receive the state proof as part of the compute device reconnecting with the distributed database.

3. The apparatus of claim 1, wherein the processor is configured to discard the data in response to determining that the data is invalid.

4. The apparatus of claim 1, wherein the processor is configured to process the data as valid in response to determining that the data is valid.

5. The apparatus of claim 1, wherein the state is associated with a Merkle tree, the data is stored as a leaf record of the Merkle tree, the processor configured to determine the validity of the data at the time by:

verifying that a Merkle path is valid for a sequence from a Merkle tree root to the leaf record.

6. The apparatus of claim 1, wherein the second identifier of the distributed database is a hash value calculated using the initial address book.

7. The apparatus of claim 1, wherein the data associated with the state is a portion of data stored in the distributed database.

8. The apparatus of claim 1, wherein a digital signature associated with the data is an aggregate signature associated with each compute device from the plurality of compute devices that has digitally signed the data.

9. The apparatus of claim 1, wherein the compute device from the plurality of compute devices is a first compute device from the plurality of compute devices,

a second compute device from the plurality of compute devices digitally signs the data using a private key associated with the second compute device, the processor configured to verify the second compute device has digitally signed the data using a public key associated with the second compute device.

10. The apparatus of claim 1, wherein:

each address book from the set of address books is a set of public keys, each key from the set of public keys being associated with an amount of stake,

for each address book from the set of address books, each public key from the set of public keys associated with that address book is associated with a compute device from a

plurality of compute devices that implements the version of the distributed database during the time period associated with that address book.

11. The apparatus of claim 1, wherein the processor is configured to define a new address book in response to receiving an event including at least one of:

a transaction to add a compute device to the plurality of compute devices,

a transaction to remove a compute device from the plurality of compute devices, or

a transaction to update an amount of stake associated with a compute device from the plurality of compute devices.

12. The apparatus of claim 1, wherein the predetermined number of compute devices from the plurality of compute devices is based on a total number of compute devices within the plurality of compute devices.

13. The apparatus of claim 1, wherein the predetermined number of compute devices from the plurality of compute devices is based on a stake associated with each compute device from the plurality of compute devices.

14. The apparatus of claim 1, wherein the verifying that the data associated with the state has been digitally signed includes verifying that a hash value of the data associated with the state has been digitally signed by the predetermined number of compute devices from the plurality of compute devices.

15. The apparatus of claim 1, wherein the compute device from the plurality of compute devices is a first compute device from the plurality of compute devices,

the compute device associated with the distributed database is a second compute device from the plurality of compute devices implementing the distributed database.

16. The apparatus of claim 1, wherein the compute device associated with the distributed database is (1) associated with a user of the distributed database, (2) not from the plurality of compute devices implementing the distributed database, and (3) operatively coupled to the compute device from the plurality of compute devices implementing the distributed database.

17. A non-transitory processor-readable medium storing code representing instructions to be executed by a processor, the code comprising code to cause the processor to:

receive, from a compute device from a plurality of compute devices that implements a distributed database via a network, a state proof associated with a state of the distributed database, the state proof including:

data stored as a set of leaf records of a Merkle tree of the state,

a Merkle path associated with the data, and

a set of address books associated with the distributed database, each address book from the set of address books associated with a version of the distributed database during a time period different from a time period associated with the version of the distributed database associated with each remaining address book from the set of address books, the set of address books having a chronological order; and

determine validity of the data by:

verifying the Merkle path as valid for a sequence from a root of the Merkle tree to the leaf record, and

other than an initial address book from the set of address books, verifying that each address book from the set of address books is digitally signed by a predetermined number of compute devices from a set of compute devices associated with an immediately preceding address book in the chronological order and from the set of address books.

18. The non-transitory processor-readable medium of claim 17, wherein the code to cause the processor to receive includes code to cause the processor to receive the state proof in response to a request to verify the data sent to the compute device.

19. The non-transitory processor-readable medium of claim 17, further comprising code to cause the processor to:

disregard the data in response to determining that the data is invalid.

20. The non-transitory processor-readable medium of claim 17, wherein the code to cause the processor to determine validity of the data includes code to cause the processor to:

verify that a predetermined number of compute devices from the plurality of compute devices have digitally signed the root of the Merkle tree.

21. The non-transitory processor-readable medium of claim 17, wherein the verifying the Merkle path includes using a set of hash values associated with a set of sibling nodes of each node on a sequence from the leaf record to the root of the Merkle tree.

22. The non-transitory processor-readable medium of claim 17, wherein the code to cause the processor to determine validity of the data includes code to cause the processor to:

verify that a hash value of the data has been digitally signed by a predetermined number of compute devices from the plurality of compute devices.

23. The non-transitory processor-readable medium of claim 17, wherein the predetermined number of compute devices from the set of compute devices is based on a stake associated with each compute device from the set of compute devices.

~~24. A method, comprising:~~

~~calculating, at a first time, an identifier for a distributed database using a first address book of the distributed database, the first address book including a public key associated with each compute device from a first plurality of compute devices implementing the distributed database at the first time;~~

~~receiving a transaction to at least one of (1) add a compute device to the first plurality of compute devices, (2) remove a compute device from the first plurality of compute devices, or (3) modify a compute device from the first plurality of compute devices, to define a second plurality of compute devices;~~

~~defining, at a second time after the first time, a second address book including a public key associated with each compute device from the second plurality of compute devices;~~

~~receiving, from a compute device from the second plurality of compute devices, a state proof associated with data of the distributed database after the second time; and~~

~~verifying the data of the distributed database by confirming that a predetermined number of compute devices from the first plurality of compute devices have digitally signed the second address book.~~

~~25. The method of claim 24, wherein the compute device from the second plurality of compute devices is a first compute device from the second plurality of compute devices,~~

the receiving the state proof includes receiving the state proof as part of a second compute device from the second plurality of compute devices reconnecting with the distributed database.

26. The method of claim 24, wherein the calculating includes calculating the identifier using a hash function with the first address book of the distributed database as an input.

27. The method of claim 24, wherein the verifying the data is further based on verifying that the data has been digitally signed by a predetermined number of compute devices from the second plurality of compute devices.

28. The method of claim 24, wherein:

the identifier for the distributed database is a first instance of the identifier for the distributed database;

the state proof includes a second instance of the identifier for the distributed database; and

the verifying the data includes verifying that the first instance of the identifier for the distributed database matches the second instance of the identifier for the distributed database.

29. The method of claim 24, wherein:

the data is stored as a leaf record of a Merkle tree;

the state proof includes a Merkle path associated with the data; and

the verifying the data includes verifying that the Merkle path is valid for a sequence from a root of the Merkle tree to the leaf record of the Merkle tree.

30. The method of claim 24, wherein the state proof is a first state proof and the data is first data, the method further comprising:

receiving a transaction to at least one of : (1) add a compute device to the second plurality of compute devices, (2) remove a compute device from the second plurality of compute devices, or (3) modify a compute device from the second plurality of compute devices, to define a third plurality of compute devices;

defining, at a third time after the second time, a third address book including a public key associated with each compute device from the third plurality of compute devices;

receiving, from a compute device from the third plurality of compute devices, a second state proof associated with second data of the distributed database after the third time; and

verifying the second data of the distributed database by confirming that a predetermined number of compute devices from the second plurality of compute devices have digitally signed the third address book and that a predetermined number of compute devices from the first plurality of compute devices have digitally signed the second address book.

31. The method of claim 24, wherein the predetermined number of compute devices from the first plurality of compute devices is based on a total number of compute devices within the first plurality of compute devices.

32. The method of claim 24, wherein the predetermined number of compute devices from the first plurality of compute devices is based on a stake associated with each compute device from the first plurality of compute devices.