

CLAIMS

What is claimed is:

1. A tangible, non-transitory, machine-readable medium storing instructions that when executed by one or more processors effectuate operations comprising:

obtaining, with a computer system, a first request by a first program associated with a first sub-partition of a first partition of a tamper-evident data store of a decentralized computing platform to read data stored in either (i) a second sub-partition of the first partition of the tamper-evident data store of the decentralized computing platform, or (ii) a second partition of the tamper-evident data store of the decentralized computing platform, wherein:

the tamper-evident data store comprises a plurality of partitions, including the first partition and the second partition, each of the plurality of partitions corresponding to a different directed acyclic graph of cryptographic hash pointers,

the decentralized computing platform is configured to associate programs executed on the decentralized computing platform with partitions and sub-partitions of partitions,

the decentralized computing platform is configured to prevent programs from reading data in sub-partitions or partitions with which the programs are not associated, and

programs executable on the decentralized computing platform are authorized to write program state to the sub-partitions and partitions with which respective programs are associated;

determining, with the computer system, with a subset of peer computing nodes of a set of peer computing nodes upon which the decentralized computing platform executes, that the first program is authorized to read from the second sub-partition or the second partition; and

in response to the determination, causing, with the computer system, the requested data to be read from the second sub-partition or the second partition and provided to the first program.

2. The medium of claim 1, wherein:

the first request is a request to read data stored in the second sub-partition of the first partition of the tamper-evident data store of the decentralized computing platform; and

determining that the first program is authorized to read from the second sub-partition or the second partition comprises determining that the first program is authorized to read from the second sub-partition.

3. The medium of claim 1, wherein:

the first request is a request to read data stored in the second partition of the tamper-evident data store of the decentralized computing platform; and

determining that the first program is authorized to read from the second partition or the second partition comprises determining that the first program is authorized to read from the second partition.

4. The medium of claim 3, wherein:

the first request is a request to read data stored in a third sub-partition of the second partition; and

determining that the first program is authorized to read from the second partition comprises determining that the first program is authorized to read from the third sub-partition.

5. The medium of claim 3, wherein:

causing the requested data to be read comprises obtaining an identifier of a most recent data node determined to have been added to the directed acyclic graph of cryptographic hash pointers corresponding to the second partition, the identifier distinguishing the most recent data node from other data nodes of the directed acyclic graph of cryptographic hash pointers corresponding to the second partition; and

storing the identifier in association with the requested data in the first partition.

6. The medium of claim 5, wherein obtaining the identifier of the most recent data node determined to have been added to the added to the directed acyclic graph of cryptographic hash pointers corresponding to the second partition comprises:

querying a subset of peer computing nodes of the decentralized computing platform that determine consensus regarding state of the second partition with a query requesting a respective most recent data node determined by the respective peer computing nodes to have been added to the determined to have been added to the directed acyclic graph of cryptographic hash pointers corresponding to the second partition; and

determining the most recent data node based on a plurality of query responses of the subset of peer computing nodes of the decentralized computing platform that determine consensus regarding state of the second partition.

7. The medium of claim 6, wherein determining the most recent data node based on the plurality of query responses comprises:

determining that a first subset of peer computing nodes have determined that a first data node is the most recent data node;

determining that a second subset of peer computing nodes have determined that a second data node is the most recent data node; and

obtaining a first value associated with the first data node indicating a sequence with which data nodes are added to the directed acyclic graph of cryptographic hash pointers corresponding to the second partition;

obtaining a second value associated with the second data node indicating the sequence with which data nodes are added to the directed acyclic graph of cryptographic hash pointers corresponding to the second partition; and

determining, based on a comparison of the first value and the second value, that the second data node was added to the directed acyclic graph of cryptographic hash pointers corresponding to the second partition before the first data node and, in response, selecting the second data node as the most recent data node and causing the requested data to be read from the second data node.

8. The medium of claim 5, wherein obtaining the identifier of the most recent data node determined to have been added to the directed acyclic graph of cryptographic hash pointers corresponding to the second partition comprises:

obtaining an identifier of a data node that all peer computing nodes of the decentralized computing platform that determine consensus regarding state of the second partition agree has been added to the directed acyclic graph of cryptographic hash pointers corresponding to the second partition and, in response, causing the requested data to be read from the identified data node that all peer computing nodes of the decentralized computing platform that determine consensus regarding state of the second partition agree has been added to the directed acyclic graph of cryptographic hash pointers corresponding to the second partition.

9. The medium of any one of claims 1-8, wherein the operations further comprise:

executing, with the decentralized computing platform, the first program associated with the first sub-partition; and

executing, with the decentralized computing platform, concurrently with executing the first program, a second program associated with the second sub-partition.

10. The medium of any one of claims 1-8, wherein:

peer computing nodes of the decentralized computing platform have one or more roles in a set of roles;

roles in the set of roles are in relation to partitions or sub-partitions;

the set of roles include a validator role and an observer role;

peer computing nodes holding the validator role are permitted, by the decentralized computing platform, to determine consensus regarding state of partitions or sub-partitions in relation to which the validator role is held; and

peer computing nodes holding the observer role are permitted, by the decentralized computing platform, to read state of partitions or sub-partitions in relation to which the observer role is held.

11. The medium of claim 10, wherein:

a first peer computing node of the decentralized computing platform holds the validator

role in relation to the first sub-partition;

the first peer computing node executes the first program;

the first peer computing node of the decentralized computing platform holds the observer role in relation to the second sub-partition or the second partition;

the first peer computing node reads the requested data;

a second peer computing node of the decentralized computing platform holds the validator role in relation to the first sub-partition;

the second peer computing node executes the first program; and

executing the first program comprises determining a consensus output of the first program based on both output of executing the first program by the first peer computing node and output of executing the first program by the second peer computing node.

12. The medium of any one of claims 1-8, wherein the first program obtains the requested data by calling a second program associated with a shared sub-partition, the second program and data stored in the shared sub-partition being accessible to programs associated with a plurality of different sub-partitions of the first partition.

13. The medium of any one of claims 1-8, wherein the operations further comprise:

detecting, with a virtual machine executed by a peer computing node of the decentralized computing platform, an opcode in a bytecode representation of the first program, wherein the virtual machine has an identifier in memory identifying a state trie from which program state is read, the state trie being part of the directed acyclic graph of cryptographic hash pointers corresponding to the first partition;

in response to detecting the opcode, determining, with the virtual machine, that the opcode or an argument of the opcode indicates the request is to read data from the second partition; and

in response to determining that the opcode or argument of the opcode indicates the request is to read data from the second partition, changing the identifier to identify another state trie that is part of the directed acyclic graph of cryptographic hash pointers corresponding to the second partition.

14. The medium of any one of claims 1-8, wherein the operations further comprise:

obtaining a second request by a second program associated with the first sub-partition of the first partition of the tamper-evident data store of the decentralized computing platform to read data stored in either (i) the second sub-partition of the first partition of the tamper-evident data store of the decentralized computing platform, or (ii) the second partition of the tamper-evident data store of the decentralized computing platform;

determining, with the subset of peer computing nodes of the set of peer computing nodes upon which the decentralized computing platform executes, that the second program is not authorized to read from the second sub-partition or the second partition; and

in response to the determination, preventing the requested data from being provided to the second program.

15. A method, comprising:

obtaining, with a computer system, a first request by a first program associated with a first sub-partition of a first partition of a tamper-evident data store of a decentralized computing platform to read data stored in either (i) a second sub-partition of the first partition of the tamper-evident data store of the decentralized computing platform, or (ii) a second partition of the tamper-evident data store of the decentralized computing platform, wherein:

the tamper-evident data store comprises a plurality of partitions, including the first partition and the second partition, each of the plurality of partitions corresponding to a different directed acyclic graph of cryptographic hash pointers,

the decentralized computing platform is configured to associate programs executed on the decentralized computing platform with partitions and sub-partitions of partitions,

the decentralized computing platform is configured to prevent programs from reading data in sub-partitions or partitions with which the programs are not associated, and

programs executable on the decentralized computing platform are authorized to write program state to the sub-partitions and partitions with which respective programs are associated;

determining, with the computer system, with a subset of peer computing nodes of a set of peer computing nodes upon which the decentralized computing platform executes, that the first

program is authorized to read from the second sub-partition or the second partition; and
in response to the determination, causing, with the computer system, the requested data to
be read from the second sub-partition or the second partition and provided to the first program.