What is claimed is:

1. A method, comprising:

receiving, at a first compute device from a plurality of compute devices, data associated with a first transaction, each compute device from the plurality of compute devices having a separate instance of a distributed database interconnected via a network operatively coupled to the plurality of compute devices;

receiving, from a second compute device from the plurality of compute devices, data associated with a second transaction;

defining a first database clone object associated with a state associated with the distributed database, the first database clone object including a first identifier and a pointer to a database in a memory storing an initial value of the state;

defining a second database clone object associated with the state, the second database clone object including a second identifier different from the first identifier and a pointer to the database in the memory storing the initial value of the state;

receiving an indication of an initial transaction order based on an event associated with the first transaction and an event associated with the second transaction;

defining, at a first time, a first updated value of the state based on the initial value of the state and the initial transaction order;

storing the first updated value of the state in the database and as associated with the first database clone object such that a request to read the state via the first database clone object and at a second time after the first time returns the first updated value of the state;

receiving, after the first time, an indication of a consensus transaction order based on the event associated with the first transaction and the event associated with the second transaction;

defining, at a third time after the first time, a second updated value of the state based on the initial value of the state and the consensus transaction order; and

storing the second updated value of the state in the database and as associated with the second database clone object such that a request to read the state via the second database clone object and at a fourth time after the third time returns the second updated value of the state.

2. The method of claim 1, further comprising:

retrieving the first updated value of the state in response to a query to the first database clone object at a fifth time after the first time; and

retrieving the second updated value of the state in response to a query to the second database clone object at a sixth time after the third time.

3. The method of claim 1, wherein the first updated value of the state and the second updated value of the state are associated with a state of the first compute device, the state of the first compute device being associated with at least one of an amount of cryptocurrency held by the first compute device, an indication of ownership of an item, or a state of a multi-player game.

4. The method of claim 1, further comprising:

returning the initial value of the state in response to a request to read the state associated with the distributed database via the second database clone object at a fifth time after the first time and before the third time; and

returning the second updated value of the state in response to a request to read the state associated with the distributed database via the second database clone object at a sixth time after the third time.

5. The method of claim 1, further comprising:

returning the initial value of the state in response to a request to read the state associated with the distributed database via the first database clone object at a fifth time before the first time; and

returning the first updated value of the state in response to a request to read the state associated with the distributed database via the first database clone object at a sixth time after the first time.

6. The method of claim 1, wherein the event associated with the first transaction is a first event and the event associated with the second transaction is a second event,

the first event includes an identifier of a third event and an identifier of a fourth event,

the second event includes an identifier of a fifth event and an identifier of a sixth event.

7. An apparatus, comprising:

a memory associated with an instance of a distributed database at a first compute device configured to be included within a plurality of compute devices that implements the distributed database via a network operatively coupled to the plurality of compute devices, the memory storing an initial value of a state associated with the distributed database; and

a processor, operatively coupled to the memory,

the processor configured to define a first database clone object associated with the state and a second database clone object associated with the state, the first database clone object including a first identifier and a pointer to a database in the memory storing the initial value of the state associated with the distributed database, the second database clone object including a second identifier different from the first identifier and a pointer to the database in the memory storing the initial value of the state associated with the distributed database,

the processor configured to update, at a first time, a value of the state and associated with the first database clone object by defining a first copy of the initial value of the state associated with the distributed database as associated with the first identifier and updating a value of the first copy of the initial value of the state associated with the distributed database based on a set of events associated with the distributed database to define a first updated value of the state associated with the distributed database,

the processor configured to update, at a second time after the first time, a value of the state and associated with the second database clone object by defining a second copy of the initial value of the state associated with the distributed database as associated with the second identifier and updating a value of the second copy of the initial value of the state associated with the distributed database based on the set of events to define a second updated value of the state associated with the distributed database.

8. The apparatus of claim 7, wherein:

the processor is configured to read from a first location in the database the initial value of the state in response to receiving, via the first database clone object and at a time prior to the first time, a request to read the state associated with the distributed database, and

the processor is configured to read from a second location in the database the first updated value of the state in response to receiving, via the first database clone object and at a time after the first time, a request to read the state associated with the distributed database.

9. The apparatus of claim 7, wherein:

the processor is configured to read from a location in the database the initial value of the state in response to receiving, via the first database clone object and at a third time prior to the first time, a request to read the state associated with the distributed database, and

the processor is configured to read from the location in the database the initial value of the state in response to receiving, via the second database clone object and at a fourth time prior to the first time, a request to read the state associated with the distributed database.

10. The apparatus of claim 7, wherein:

the processor is configured to read from a first location in the database the first updated value of the state in response to receiving, via the first database clone object and at a third time after the first time and before the second time, a request to read the state associated with the distributed database, and

the processor is configured to read from a second location in the database the initial value of the state in response to receiving, via the second database clone object and at a fourth time after the first time and before the second time, a request to read the state associated with the distributed database.

11. The apparatus of claim 7, wherein:

the processor is configured to read from a first location in the database the initial value of the state in response to receiving, via the first database clone object and at a third time prior to the first time, a request to read the state associated with the distributed database,

the processor is configured to read from the first location in the database the initial value of the state in response to receiving, via the second database clone object and at a fourth time prior to the second time, a request to read the state associated with the distributed database,

the processor is configured to read from a second location in the database the first updated value of the state in response to receiving, via the first database clone object and at a fifth time after the first time, a request to read the state associated with the distributed database, and

the processor is configured to read from a third location in the database the second updated value of the state in response to receiving, via the second database clone object and at a sixth time after the second time, a request to read the state associated with the distributed database.

12. The apparatus of claim 7, wherein the instance of the distributed database is different from the database in the memory storing the state associated with the distributed database.

13. The apparatus of claim 7, wherein the second updated value is a current value of the state associated with the distributed database at a third time after the second time.

14. The apparatus of claim 7, wherein the processor is configured to update the value of the state and associated with the second database clone object by updating the value of the second copy of the initial value of the state associated with the distributed database based on a consensus order of the set of events to define the second updated value of the state associated with the distributed database.

15. The apparatus of claim 7, wherein:

the initial value of the state is stored within the database in the memory as associated with a third identifier different from the first identifier and the second identifier,

the first updated value of the state is stored within the database in the memory as associated with the first identifier, and

the second updated value of the state is stored within the database in the memory as associated with the second identifier.

16. The apparatus of claim 7, wherein:

in response to a query to the first database clone object, the processor is configured to read from a location in the database associated with the first identifier or a location in the database associated with a third identifier associated with the initial value of the state,

the processor configured to ignore a location in the database associated with the second identifier in response to the query to the first database clone object.

17. The apparatus of claim 7, wherein the state associated with the distributed database is associated with a state of the first compute device, the state of the first compute device being associated with at least one of an amount of cryptocurrency held by the first compute device, an indication of ownership of an item, or a state of a multi-player game.

18. The apparatus of claim 7, wherein each event from the set of events includes an identifier of at least two other events.

19. 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:

define a first database clone object associated with a state associated with a distributed database interconnected via a network operatively coupled to a plurality of compute devices that implements the distributed database, the first database clone object including a first identifier and a pointer to a database in a memory storing an initial value of the state associated with the distributed database;

define a second database clone object associated with the state, the second database clone object including a second identifier different from the first identifier and a pointer to the database in the memory storing the initial value of the state associated with the distributed database;

receive, at a first time, an indication of an initial order of a set of events associated with the distributed database;

update, in response to receiving the indication of the initial order of the set of events, a value of the state and associated with the first database clone object by defining a first copy of the initial value of the state associated with the distributed database as associated with the first identifier and updating a value of the first copy of the initial value of the state associated with the distributed database based on the set of events to define a first updated value of the state associated with the distributed database;

receive, at a second time after the first time, an indication of a consensus order of the set of events associated with the distributed database; and

update, in response to receiving the indication of the consensus order of the set of events, a value of the state and associated with the second database clone object by defining a second copy of the initial value of the state associated with the distributed database as associated with the second identifier and updating a value of the second copy of the initial value of the state associated with the distributed database based on the consensus order of the set of events to define a second updated value of the state associated with the distributed database, the second updated value being a value of the state associated with the distributed database and corresponding to the consensus order.

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

read from a first location in the database the initial value of the state in response to receiving, via the first database clone object and at a third time prior to the first time, a request to read the state associated with the distributed database;

read from the first location in the database the initial value of the state in response to receiving, via the second database clone object and at a fourth time prior to the second time, a request to read the state associated with the distributed database;

read from a second location in the database the first updated value of the state in response to receiving, via the first database clone object and at a fifth time after the first time, a request to read the state associated with the distributed database; and

read from a third location in the database the second updated value of the state in response to receiving, via the second database clone object and at a sixth time after the second time, a request to read the state associated with the distributed database.

21. The non-transitory processor-readable medium of claim 19, wherein:

the initial value of the state is stored within the database in the memory as associated with a third identifier different from the first identifier and the second identifier,

the first updated value of the state is stored within the database in the memory as associated with the first identifier, and

the second updated value of the state is stored within the database in the memory as associated with the second identifier.

22. The non-transitory processor-readable medium of claim 19, wherein an instance of the distributed database is different from the database in the memory storing the state associated with the distributed database.

23. The non-transitory processor-readable medium of claim 19, wherein each event from the set of events includes an identifier of at least two other events.