What is claimed is:

1. An apparatus, comprising:

a memory storing 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; and

a processor operatively coupled to the memory, the processor configured to select a second compute device from the plurality of compute devices with which to sync,

the processor configured to identify, based on a directed acyclic graph (DAG) associated with the instance of the distributed database and encoding a partial order of a plurality of events, a set of events from the plurality of events and to send to the second compute device as part of the sync, the DAG associated with the instance of the distributed database not indicating that the second compute device has received the set of events,

the processor configured to receive, as part of the sync, a first event from the second compute device from the plurality of compute devices, the first event including (1) an identifier of a first ancestor event and (2) an identifier of a second ancestor event,

the processor configured to receive a second event from a third compute device from the plurality of compute devices, the second event including (1) an identifier of a third ancestor event and (2) an identifier of a fourth ancestor event,

the processor configured to update the DAG based on the first event and the second event,

the processor configured to calculate a consensus order associated with the distributed database based on the DAG,

the processor configured to calculate a state value associated with events in the consensus order,

the processor configured to delete, from the instance of the distributed database, events used to calculate the state value.

2. The apparatus of claim 1, wherein the processor is configured to calculate the consensus order using the partial order identified in the DAG.

3. The apparatus of claim 1, wherein the processor is configured to receive the first event at a first time, the processor is configured to receive the second event at a second time after the first time,

the processor configured to define a third event including (1) an identifier of the first event and (2) an identifier of a fifth ancestor event, the fifth ancestor event being defined by the processor,

the processor configured to define a fourth event including (1) an identifier of the third event and (2) an identifier of the second event.

4. The apparatus of claim 1, wherein the identifier of the first ancestor event is a cryptographic hash of the first ancestor event.

5. The apparatus of claim 1, wherein the first event includes a first set of transactions and the second event includes a second set of transactions, the consensus order determining an order of the first set of transactions with respect to the second set of transactions.

6. The apparatus of claim 1, wherein:

the identifier of the first ancestor event indicates that the first ancestor event is ordered prior to the first event and the identifier of the second ancestor event indicates that the second ancestor event is ordered prior to the first event, and

the identifier of the third ancestor event indicates that the third ancestor event is ordered prior to the second event and the identifier of the fourth ancestor event indicates that the fourth ancestor event is ordered prior to the second event.

7. The apparatus of claim 1, wherein the processor is configured to update the DAG based on the first event, the second event, the identifier of the first ancestor event such that the first ancestor event is ordered prior to the first event in the DAG, the identifier of the second ancestor event such that the second ancestor event is ordered prior to the first event in the DAG, the identifier of the third ancestor event such that the third ancestor event is ordered prior to the second event in the DAG, and the identifier of the fourth ancestor event such that the fourth ancestor event is ordered prior to the second event in the DAG.

8. The apparatus of claim 1, wherein the processor is configured to send a representation of the consensus order to a fourth compute device from the plurality of compute devices.

9. The apparatus of claim 1, wherein the processor is configured to:

define a third event including the state value; and

send the third event to a fourth compute device from the plurality of compute devices.

10. The apparatus of claim 1, wherein the processor is configured to:

define a third event including the state value to notify other compute devices from the plurality of compute devices of the consensus order,

digitally sign the state value, and

confirm the state value based on a threshold number of compute devices from the plurality of compute devices digitally signing the state value, the processor configured to delete the events used to calculate the state value in response to confirming the state value.

11. The apparatus of claim 1, wherein the first event is digitally signed by the second compute device from the plurality of compute devices.

12. 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:

select, at a first compute device configured to be included within a plurality of compute devices that implements a distributed database via a network, a second compute device from the plurality of compute devices with which to sync;

identify, based on a directed acyclic graph (DAG) associated with an instance of the distributed database at the first compute device and encoding a partial order of a plurality of events, a set of events from the plurality of events and to send to the second compute device as part of the sync, the DAG associated with the instance of the distributed database not indicating that the second compute device has received the set of events;

receive, at a first compute device and as part of the sync, an event from the second compute device from the plurality of compute devices, the event including (1) an identifier of a first ancestor event and (2) an identifier of a second ancestor event;

update the DAG based on the event, the first ancestor event, and the second ancestor event;

calculate, based on the DAG, a consensus order associated with the distributed database;

calculate a state value associated with events in the consensus order; and

delete, from the instance of the distributed database, events used to calculate the state value.

13. The non-transitory processor-readable medium of claim 12, wherein the identifier of the first ancestor event is a cryptographic hash of the first ancestor event.

14. The non-transitory processor-readable medium of claim 12, wherein the code to cause the processor to calculate the consensus order includes code to cause the processor to calculate the consensus order using the partial order identified in the updated DAG.

15. The non-transitory processor-readable medium of claim 12, wherein the first ancestor event includes a first set of transactions and the second ancestor event includes a second set of transactions, the code further comprising code to cause the processor to:

identify an order of the first set of transactions with respect to the second set of transactions based on the consensus order.

16. The non-transitory processor-readable medium of claim 12, wherein the event is a first event, the code further comprising code to cause the processor to:

receive a second event from a third compute device from the plurality of compute devices, the second event including (1) an identifier of a third ancestor event and (2) an identifier of a fourth ancestor event,

the code to cause the processor to update including code to cause the processor to update the DAG based on the first event, the second event, the first ancestor event, the second ancestor event, the third ancestor event, and the fourth ancestor event.

17. The non-transitory processor-readable medium of claim 12, further comprising code to cause the processor to:

send a representation of the consensus order to a third compute device from the plurality of compute devices.

18. The non-transitory processor-readable medium of claim 12, wherein the event is a first event, the code further comprising code to cause the processor to:

define a second event including the state value; and

send the second event to a third compute device from the plurality of compute devices.

19. The non-transitory processor-readable medium of claim 12, wherein the event is a first event, the code further comprising code to cause the processor to:

define a second event including the state value to notify other compute devices from the plurality of compute devices of the consensus order;

digitally sign the state value; and

confirm the state value based on a threshold number of compute devices from the plurality of compute devices digitally signing the state value, the code to cause the processor to delete includes code to cause the processor to delete the events used to calculate the state value in response to confirming the state value.

20. A method, comprising:

selecting, at a first compute device configured to be included within a plurality of compute devices that implements a distributed database via a network, a second compute device from the plurality of compute devices with which to sync;

identifying, based on a directed acyclic graph (DAG) associated with an instance of the distributed database at the first compute device and encoding a partial order of a plurality of events, a set of events from the plurality of events and to send to the second compute device as part of the sync, the DAG associated with the instance of the distributed database not indicating that the second compute device has received the set of events;

receiving, at the first compute device and from the second compute device as part of the sync, a first event that includes (1) an identifier of a second event and (2) an identifier of a third event, the third event being defined by a third compute device from the plurality of compute devices;

defining, at the first compute device, a fourth event, the fourth event including (1) an identifier of the first event and (2) an identifier of a fifth event; and

updating the DAG based on the first event, the second event, the third event, the fourth event and the fifth event;

calculating a consensus order associated with the distributed database based on the partial order identified in the updated DAG;

calculating a state value associated with events in the consensus order; and

deleting, from the instance of the distributed database, events used to calculate the state value.

21. The method of claim 20, further comprising: receiving the fifth event from a fourth compute device from the plurality of compute devices.

22. The method of claim 20, wherein the identifier of the first event is a cryptographic hash of the first event.

23. The method of claim 20, wherein the second event includes a first set of transactions and the third event includes a second set of transactions, the method further comprising:

identifying an order of the first set of transactions with respect to the second set of transactions based on the consensus order.

24. The method of claim 20, further comprising:

sending a representation of the consensus order to a third compute device from the plurality of compute devices.

25. The method of claim 20, further comprising:

defining a sixth event including the state value; and sending the sixth event to a third compute device from the plurality of compute devices.

26. The method of claim 20, further comprising:

defining a sixth event including the state value to notify other compute devices from the plurality of compute devices of the consensus order;

digitally signing the state value; and

confirming the state value based on a threshold number of compute devices from the plurality of compute devices digitally signing the state value, the deleting being in response to confirming the state value.