

CLAIMS

1. A method comprising:

loading journal manager engine instructions for a journal manager engine into journal manager engine memory, the journal manager engine instructions being stored by a journal that stores instructions and a series of transactions including a time of each transaction in the series of transactions, wherein the journal manager engine reads from, and writes to, the journal; and wherein the journal manager engine and the journal are configured as part of a ledger;

receiving, from a first client, a register transaction type request for a first transaction type at the journal manager engine, wherein the register transaction type request is a request to create the first transaction type that the ledger can perform, and wherein the register transaction type request includes a first transaction type operation definition that specifies at least one type of action that can be performed as part of the first transaction type, and wherein the register transaction type request is based at least in part on a template stored in the journal, and wherein the at least one type of action is based on further instructions provided by the first client, and said further instructions are arranged to be stored on said template, and are additional to code used for initial implementation of the ledger;

loading, into the journal manager engine, a register transaction type operation definition for a register transaction type operation in response to the register transaction type request, wherein the register transaction type operation definition is stored in the journal prior to the journal manager engine receiving the register transaction type request;

writing a register transaction type transaction to the journal;

executing the register transaction type operation, in response to the register transaction type request and based at least in part on the register transaction type operation definition, to produce a first transaction type operation definition;

recording the first transaction type operation definition in the journal;

receiving, from a second client, a transaction request for a transaction of the first transaction type; and

performing, based at least in part on executing the further instructions stored on said template, the transaction of the first transaction type.

2. The method of claim 1, wherein the method further comprises loading a definition of hooks into the journal manager based at least in part on the register transaction type request.
3. The method of claim 2, wherein the method further comprises executing callbacks based at least in part on the definition of hooks.
4. The method of claim 1, wherein the transaction request is to recalculate a state of a parameter at a first time, and wherein the journal stores a first state for the parameter at the first time based on a request made at a second time wherein the first state is different than a second state for the parameter at the first time based on a request made at a third time.
5. The method of claim 1, wherein the journal comprises a series of state snapshots, each state snapshot having a time-stamp.
6. The method of claim 5, wherein the method further comprises creating a fingerprint of at least one state snapshot and forwarding the fingerprint to a trusted third party.
7. The method of claim 5, wherein state snapshots are incorporated into one or more state timelines.
8. The method of claim 1, wherein the ledger is extensible at runtime allowing a client to update dynamically which instructions the ledger executes.
9. The method of claim 1, wherein the journal manager engine instructions are based on a template.
10. The method of claim 1, wherein receiving the register transaction type request comprises receiving the register transaction type request via an application programming interface.
11. A system comprising:
 - one or more computers and one or more storage devices on which are stored instructions that are operable, when executed by the one or more computers, to cause the one or more computers to perform operations comprising:
 - loading journal manager engine instructions for a journal manager engine

into journal manager engine memory, the journal manager engine instructions being stored by a journal that stores instructions and a series of transactions including the time of each transaction in the series of transactions, wherein the journal manager engine reads from, and writes to, the journal; and wherein the journal manager engine and the journal are configured as part of a ledger;

receiving, from a first client, a register transaction type request for a first transaction type at the journal manager engine, wherein the register transaction type request is a request to create the first transaction type that the ledger can perform, and wherein the register transaction type request specifies at least one type of action that can be performed as part of the first transaction type, and wherein the register transaction type request is based at least in part on a template stored in the journal, and wherein the at least one type of action is based on further instructions provided by the first client, and said further instructions are arranged to be stored on said template, and are additional to code used for initial implementation of the ledger;

loading, into the journal manager engine, a register transaction type operation definition for a register transaction type operation in response to the register transaction type request, wherein the register transaction type operation definition is stored in the journal prior to the journal manager engine receiving the register transaction type request;

writing a register transaction type request to the journal;

executing the register transaction type operation, in response to the register transaction type request and based at least in part on the register transaction type operation definition, to produce a first transaction type operation definition;

recording the first transaction type operation definition in the journal;

receiving, from a second client, a transaction request for a transaction of the first transaction type; and

performing, based at least in part on executing the further instructions stored on said template, the transaction of the first transaction type.

12. The system of claim 11, wherein the system further comprises allowing a first transaction to be executed as if implemented on a specific past date and to replay subsequent transactions based on executing the first transaction on a specific past date.

13. The system of claim 11, wherein the journal manager engine is extensible at runtime.
14. The system of claim 11, wherein the operations further comprise loading a definition of a hook into the journal manager based at least in part on the register transaction type request.
15. The system of claim 14, wherein the operations further comprise executing a callback based in part on the definition of the hook.
16. The system of claim 11, wherein the transaction request is to recalculate a state of a parameter at a first time, wherein the journal stores a first state for the parameter at the first time based on a request made at a second time, and wherein the first state is different than a second state for the parameter at the first time based on a request made at a third time.
17. The system of claim 11 wherein the journal comprises a series of state snapshots, each state snapshot having a time-stamp, and wherein the operations further comprise creating a fingerprint of a state snapshot and forwarding the fingerprint to a blockchain.
18. The system of claim 11, wherein the journal manager engine instructions are based on a template.
19. A method comprising:
receiving, from a first client, at a journal manager engine a register transaction type request for a first transaction type, the journal manager engine configured to read from and write to a journal that is configured to store a history of object state and a history of code by which object state is changed, wherein the journal manager engine and the journal are configured as part of a ledger; and wherein the register transaction type request is based at least in part on a template and specifies at least one action that can be performed as part of the first transaction type; and wherein the register transaction type request is a request to create the first transaction type that the ledger can perform; and
wherein the at least one type of action is based on further instructions provided by the first client, and said further instructions are arranged to be stored on said template, and are additional to code used for initial implementation of the ledger;
writing the register transaction type request to the journal including an operation

definition for the first transaction type;

receiving, from a second client, a transaction request of the first transaction type;

obtaining the operation definition from the journal, in response to receiving the transaction request;

executing, an operation in response to receiving the transaction request and based at least in part on the operation definition and further based at least in part on executing the further instructions stored on said template, to produce a transaction result; and

recording the transaction result in the journal.