

Claims

1. A computer implemented method for enabling the execution of one or more smart contracts associated with a blockchain, the method implemented by a platform processor associated with an application programming interface (API), the method comprising the steps of:
 - receiving a request from a client, the request relating to a smart contract (SC), the request from the client being based on a Hypertext Transfer Protocol (HTTP) transmission protocol format;
 - accessing the smart contract SC pertaining to the request;
 - identifying an event stream (ES) associated with the blockchain, the event stream ES being specific to the smart contract SC, the event stream representing the states of the smart contract SC;
 - determining a current state ES_n of the smart contract SC based on the identified event stream ES, wherein n is an integer from 0 to N , each integer n representing a current state or current number of events associated with the smart contract SC, and N represents a maximum or final value or state of n ;
 - invoking execution of the smart contract SC based on the received request;
 - invoking updating the event stream ES pertaining to the smart contract SC by the steps including :
 - processing a new event E_n based on the received request; and
 - obtaining an updated current state of the smart contract ES_{n+1} based on the new event E_n ; and
 - storing or providing a result based on the updated current state ES_{n+1} for the smart contract.
2. The method as claimed in claim 1, wherein the step of invoking execution of the smart contract SC further comprises:
 - obtaining an identifier specific to the smart contract SC from the received request;
 - based on the identifier, accessing the given smart contract SC from a contract store;
 - providing the smart contract SC to one or more processors for executing one or more software routines or programming instructions associated with the smart contract.

3. The method as claimed in claims 1 or 2 wherein based on a determination that $n = 0$, the step of processing a new event E_n includes the steps of:
 - identifying the new event E_n for the smart contract SC in the received request as a first event to create the respective event stream ES pertaining to the smart contract SC;
 - processing the received event E_n by creating a blockchain transaction TX_0 including a first unspent output $UTXO_{0_dust}$ that is a dust output.
 - submitting the transaction TX_0 to the blockchain; and
 - providing a result associated with the updated event stream ES.
4. The method as claimed in claim 3 wherein the blockchain transaction TX_0 includes an identifier or a hash of the smart contract SC.
5. The method as claimed in claims 1 or 2 wherein based on a determination that $0 < n < N$, the step of processing a new event E_n includes the steps of:
 - identifying the new event E_n for the smart contract SC in the received request as a current event to amend the respective event stream ES pertaining to the smart contract SC;
 - processing the received event E_n by creating a blockchain transaction TX_n including:
 - a first input spending a dust output associated with a previous transaction for the event stream ES;
 - a first unspent transaction output $UTXO_{n_dust}$ being a dust output; and
 - a final unspent transaction output $UTXO_{n_data}$ associated with event data representing the current event E_n ;
 - submitting the transaction TX_n to the blockchain; and
 - providing a result associated with the updated event stream ES.
6. The method as claimed in claim 5 where the event data for the event E_n in the final unspent transaction output ($UTXO_{n_data}$) for the current blockchain transaction TX_n includes a hash of said event data.
7. The method as claimed in claim 6 where the hash of said event data is applied by the platform processor.
8. The method as claimed in claim 6 where the hash of said event data is applied by the client prior to being included in the request that is received by the platform processor.

9. The method as claimed in claim 6 where the event data for the event E_n in the final unspent transaction output ($UTXO_{n_data}$) for the current blockchain transaction TX_n includes raw event data.

10. The method as claimed in claims 1 or 2 wherein based on a determination that that $n = N$, the step of processing a new event E_n includes the steps of:

- identifying the new event E_n for the smart contract SC in the received request as a final event to terminate the event stream ES;
- processing the received event E_n by creating a blockchain transaction TX_N including:
 - a first input spending a dust output associated with a previous transaction for the event stream ES; and
 - a final unspent output $UTXO_N$ that is associated with a digital asset that is more than a defined dust output value;
- submitting the created transaction TX_N to the blockchain; and
- providing a result associated with the event stream ES in TX_N .

11. The method as claimed in any one of claims 3 to 10 wherein the step of submitting comprises including the created transaction in a subsequent block associated with the blockchain to be mined.

12. The method as claimed in any one of claims 3 to 11 comprising the step of identifying the event stream ES based on a transaction identifier associated with the submitted blockchain transaction.

13. The method as claimed in any one of claims 3 to 12 comprising the step of identifying a state of the smart contract SC based on a transaction identifier associated with the submitted blockchain transaction.

14. The method as claimed in any one of claims 3 to 13 wherein the created blockchain transaction further comprises:

- an input associated with a digital asset; and
- one or more change outputs associated with the digital asset.

15. The method as claimed in claim 14 wherein the digital asset is associated with an operational float.

16. The method as claimed in any one of claim 3 to 15 including the step of determining a hierarchical deterministic key chain K to be used with the event stream ES pertaining to the smart contract SC , wherein the key chain K includes a series of private/public key pairs derived from a selected parent key pair, such that $K = K_{n=0 \text{ to } N}$.

17. The method as claimed in any preceding claim wherein the step of invoking updating the event stream ES pertaining to the smart contract further includes the step of accessing a data writer associated with the platform processor for processing the new event E_n .

18. The method as claimed in any one of the preceding claims wherein the result is a snapshot state of the smart contract SC , said result being stored in and/or accessible from an instance state database associated with the platform processor.

19. The method as claimed in any one of the preceding claims wherein the result is provided to the client based on the HTTP transmission protocol format.

20. The method as claimed in any preceding claim wherein the result associated with the current state ES_n of the smart contract SC includes a certificate confirming at least one of the following :

- a transaction identifier within which the event E_n was submitted to the blockchain;
- a Merkle inclusion proof of the transaction to a header in the blockchain
- a copy of the block header in which said transaction was included

21. The method as claimed in any preceding claim wherein the smart contract SC is implemented as a Finite State Machine(FSM).

22. The method as claimed in any preceding claim comprising storing a copy or a record or a log that is based on the result(s) for each event of the event stream ES for the smart contract SC in an off-chain storage resource.

23. The method as claimed in any preceding claim wherein the API is a Hyper Text Transfer Protocol (HTTP) Application programming Interface (API) endpoint, and wherein said endpoint is provided to the client using HTTP Secure (HTTPS).

24. The method as claimed in any one of claims 1 to 23 wherein the API is an alias associated with the platform processor, the alias being specific to the platform processor and provided by an alias-based addressing service, the addressing service having a machine readable resource that is accessible from a defined or well-known location, the machine readable resource including one or more capabilities pertaining to the platform processor, and wherein the alias is associated with an asymmetric cryptographic key pair for authentication.

25. The method as claimed in any preceding claim wherein responsive to receiving the request from the client, the method comprises the step of providing one or more software libraries and/or verification tools for processing data associated with the smart contract SC associated with the request.

26. A computer implemented method for accessing a smart contract associated with a blockchain, the method implemented by one or more processors of a given client among a plurality of clients, the method comprising the steps of:

obtaining or identifying an application programming interface (API) endpoint associated with one or more processors associated with a platform for executing a smart contract SC;

sending a request for one or more events E_n pertaining to the smart contract SC to the platform, wherein the request is sent based on a Hypertext Transfer Protocol (HTTP) transmission protocol format;

obtaining one or more software libraries and verification tools for processing data associated with the smart contract SC; and

receiving a result based on an output script of a blockchain transaction pertaining to the requested event E_n , said result representing an updated state of the smart contract SC, the result being received using a HTTP transmission protocol format.

27. The method as claimed in claim 26 comprising applying a hash function to event data for the event E_n , such that the request includes the hashed event data for the event E_n .

28. The method as claimed in claim 26 or 27 further comprising the steps of:

based on the result, recovering the event stream ES associated with the smart contract SC using the one or more software libraries;

using the one or more verification tools, comparing the recovered event stream ES to the result to verify the state of the smart contract.

29. The method as claimed in any one of claims 26 to 28 wherein the one or more software libraries are associated with a software development kit (SDK) received from the platform.

30. The method as claimed in any one of claims 26 to 27 wherein the client is associated with an alias associated with the platform processor, the alias being specific to the platform processor and provided by an alias-based addressing service, the addressing service having a machine readable resource that is accessible from a defined or well-known location, the machine readable resource including one or more capabilities pertaining to the client, and wherein the alias is associated with an asymmetric cryptographic key pair for authentication.

31. A computing device comprising a processor and memory, the memory including executable instructions that, as a result of execution by the processor, causes the device to perform the computer-implemented method as claimed in any one of claims 1 to 25, the computing device pertaining to a platform processor.

32. A computing device comprising a processor and memory, the memory including executable instructions that, as a result of execution by the processor, causes the device to perform the computer-implemented method as claimed in any one of claims 26 to 3-, wherein the computing device is pertaining to a client.

33. A computer system comprising:
at least one platform processor communicatively coupled to at least one client via a wireless communication network, the at least one platform processor associated with an application programming interface(API) endpoint for processing HTTP requests from the at least one client, the at least one platform processor implemented in accordance with the computing device as claimed in claim 31 ; and the at least one client being implemented in accordance with the computing device as claimed in claim 32, wherein the at least one platform processor is communicatively coupled via the wireless communication network to one of more of the following components of a platform service, the platform service providing a plurality of services associated with a blockchain for a plurality of clients:

- a smart contract store;
- an event stream log;
- an instance state database; and

a data writer;

34. A computer-readable storage medium having stored thereon executable instructions that, as a result of being executed by a processor of a computer, cause the computer to perform the method of any one of claims 1 to 30.