

CLAIMS

What is claimed is:

1. A method for updating a smart contract ~~implemented~~executed on a blockchain, comprising:
 - segregating the smart contract into an updateable smart contract and a plurality of non-updateable smart contracts, where ~~each~~the non-updateable smart contract handles persistent data storage and retrieval to the blockchain for data that needs to be permanently accessible and the updateable smart contract does not contain data storage and retrieval to the blockchain;
 - deploying ~~the~~each non-updateable smart contract and a first version of the updateable smart contract;
 - the first version of the updateable smart contract self-registering with the non-updateable smart contracts upon deployment by providing the non-updateable smart contracts with an address of the first version of the updateable smart contract;
 - the non-updateable smart contracts storing the address of the first version of the updateable smart contract to the blockchain;
 - at least one of the non-updateable smart contracts retrieving permanent data from the blockchain and passing the permanent data to the first version of the updateable smart contract;
 - the updateable smart contract performing a computation on the permanent data and returning updated data based on the computation of the permanent data back to the at least one non-updateable smart contract; and
 - the at least one non-updateable smart contract storing the updated data to the blockchain.
2. A method as in claim 1, wherein the updateable smart contract and the non-updateable smart contracts are connected via a fixed interface, and wherein the address of the first version of the updateable smart contract is provided to the non-updateable smart contracts via the fixed interface, the non-updateable smart contract passes the permanent data to the first version of the updateable smart contract via the fixed interface, and the updateable smart contract returns

updated data based on the computation of the permanent data back to the non-updateable smart contracts via the fixed interface.

3. A method as in claim 1, wherein providing the non-updateable smart contracts with an address of the first version of the updateable smart contract comprises calling an interface of the non-updateable smart contracts with the address of the first version of the updateable smart contract.

4. A method as in claim 3, further comprising:

deploying a second version of the updateable smart contract;

calling the interface of the non-updateable smart contracts with an address of the second version of the updateable smart contract;

the non-updateable smart contracts storing the address of the second version of the updateable smart contract to the blockchain; and

the non-updateable smart contracts invoking the second version of the updateable smart contract by retrieving the address of the second version of the updateable smart contract from the blockchain and calling the second version of the updateable smart contract at the retrieved address.

5. A method as claimed in claim 4, wherein the segregating comprises segregating the smart contracts into the non-updateable smart contract and a plurality of the updateable smart contracts, the deploying steps comprise deploying first and second versions of each updateable smart contract, the calling steps comprise calling respective interfaces with addresses of the respective versions of the respective updateable smart contracts, the storing steps comprise storing the addresses of the respective versions of the respective updateable smart contracts to the blockchain, and the invoking step comprises retrieving the addresses of the second versions of the updateable smart contracts from the blockchain and calling the second versions of the updateable smart contracts at the retrieved addresses.

6. A method as in claim 4, wherein the segregating comprises segregating the smart contract into the non-updateable smart contracts and a plurality of the updateable smart contracts, the deploying steps comprise deploying first and

second versions of each updateable smart contract, the calling steps comprise calling the interface with an identifier associated with and the respective addresses of the respective updateable smart contracts, the storing steps comprise storing the identifiers and addresses of the respective versions of the respective updateable smart contracts to the blockchain, and the invoking step comprises retrieving the identifiers and addresses of the second versions of the updateable smart contracts from the blockchain and calling the second versions of the updateable smart contracts identified by the identifier at the retrieved addresses.

7. A method as in claim 4, wherein each version of the updateable smart contract is self-registering with its address and the calling steps comprise respective versions of the self-registering updateable smart contract calling the interface of the non-updateable smart contract with the address of the respective versions of the self-registering updateable smart contract.

8. A method as in claim 4, wherein the segregating comprises segregating the smart contract into the non-updateable smart contracts and a plurality of the self-registering updateable smart contracts, the deploying steps comprise deploying first and second versions of each self-registering updateable smart contract, ~~the updateable smart contract is self-registering~~ and the calling steps comprise the self-registering updateable smart contracts calling respective interfaces of the non-updateable smart contracts with the addresses of the respective versions of the respective self-registering updateable smart contracts, the storing steps comprise storing the addresses of the respective versions of the respective updateable smart contracts to the blockchain, and the invoking step comprises retrieving the addresses of the second versions of the updateable smart contracts from the blockchain and calling the second versions of the updateable smart contracts at the retrieved addresses.

9. A method as in claim 4, wherein the non-updateable smart contracts retrieves permanent data from the blockchain and passes the permanent data to the updateable smart contract, and the updateable smart contract computes and

returns data based on computation of the permanent data back to the non-updateable smart contracts for permanent storage to the blockchain.

~~10. A method as in claim 4, wherein the segregating comprises segregating the smart contract into a plurality of non-updateable smart contracts.~~

~~11~~10. A method as in claim 1, further comprising:

hard-coding an address of the at least one non-updateable smart contract into the first version of the updateable smart contract prior to deploying the first version of the updateable smart contract;

deploying a second version of the updateable smart contract;

calling the second version of the updateable smart contract;

the second version of the updateable smart contract calling the at least one non-updateable smart contract at the hard-coded address; and

the at least one non-updateable smart contract storing any permanent data to the blockchain.

~~12~~11. A system for ~~implementing~~ executing a smart contract on a blockchain, comprising:

a memory to store instructions; and

one or more processors using the instructions stored in the memory to perform operations including:

segregating the smart contract into an updateable smart contract and a plurality of non-updateable smart contracts, where each ~~the~~ non-updateable smart contract handles persistent data storage and retrieval to the blockchain for data that needs to be permanently accessible and the updateable smart contract does not contain data storage and retrieval to the blockchain;

deploying each ~~the~~ non-updateable smart contract and a first version of the updateable smart contract;

the first version of the updateable smart contract self-registering with the non-updateable smart contracts upon deployment by providing the non-updateable smart contract with an address of the first version of the updateable smart contract;

the non-updateable smart contract_s storing the address of the first version of the updateable smart contract to the blockchain;

at least one of the non-updateable smart contract_s retrieving permanent data from the blockchain and passing the permanent data to the first version of the updateable smart contract;

the updateable smart contract performing a computation on the permanent data and returning updated data based on the computation of the permanent data back to the at least one non-updateable smart contract; and

the at least one non-updateable smart contract storing the updated data to the blockchain.

123. A system as in claim 112, further comprising a fixed interface that connects the updateable smart contract and the non-updateable smart contract_s, wherein the address of the first version of the updateable smart contract is provided to the non-updateable smart contract_s via the fixed interface, the non-updateable smart contract pass_{es} the permanent data to the first version of the updateable smart contract via the fixed interface, and the updateable smart contract returns updated data based on the computation of the permanent data back to the non-updateable smart contract_s via the fixed interface.

143. A system as in claim 112, wherein the memory stores further instructions to perform operations including providing the non-updateable smart contract_s with an address of the first version of the updateable smart contract by calling an interface of the non-updateable smart contract_s with the address of the first version of the updateable smart contract.

~~15~~14. A system as in claim ~~14~~13, wherein the memory stores further instructions to perform further operations including:

deploying a second version of the updateable smart contract;

calling the interface of the non-updateable smart contract_s with an address of the second version of the updateable smart contract;

the non-updateable smart contract_s storing the address of the second version of the updateable smart contract to the blockchain; and

the non-updateable smart contracts invoking the second version of the updateable smart contract by retrieving the address of the second version of the updateable smart contract from the blockchain and calling the second version of the updateable smart contract at the retrieved address.

~~16~~15. A system as in claim ~~15~~14, wherein the segregating comprises segregating the smart contract into the non-updateable smart contracts and a plurality of the updateable smart contracts, the deploying comprises deploying first and second versions of each updateable smart contract, the calling comprises calling respective interfaces with addresses of the respective versions of the respective updateable smart contracts, the storing comprises storing the addresses of the respective versions of the respective updateable smart contracts to the blockchain, and the invoking comprises retrieving the addresses of the second versions of the updateable smart contracts from the blockchain and calling the second versions of the updateable smart contracts at the retrieved addresses.

~~17~~16. A system as in claim ~~15~~14, wherein the segregating comprises segregating the smart contract into the non-updateable smart contracts and a plurality of the updateable smart contracts, the deploying comprises deploying first and second versions of each updateable smart contract, the calling comprises calling the interface with an identifier associated with and the respective addresses of the respective updateable smart contracts, the storing comprises storing the identifiers and addresses of the respective versions of the respective updateable smart contracts to the blockchain, and the invoking comprises retrieving the identifiers and addresses of the second versions of the updateable smart contracts from the blockchain and calling the second versions of the updateable smart contracts identified by the identifier at the retrieved addresses.

~~18~~17. A system as in claim ~~15~~14, wherein each version of the updateable smart contract is self-registering with its address and the calling comprises respective versions of the self-registering updateable smart contract calling the interface of the non-updateable smart contracts with the address of the respective versions of the self-registering updateable smart contract.

~~1918.~~ A system as in claim ~~1514~~, wherein the segregating comprises segregating the smart contract into the non-updateable smart contract and a plurality of ~~the self-registering~~ updateable smart contracts, the deploying comprises deploying first and second versions of each ~~self-registering~~ updateable smart contract, ~~the updateable smart contract is self-registering~~ and the calling comprises the self-registering updateable smart contracts calling respective interfaces of the non-updateable smart contracts with the addresses of the respective versions of the respective self-registering updateable smart contracts, the storing comprises storing the addresses of the respective versions of the respective updateable smart contracts to the blockchain, and the invoking comprises retrieving the addresses of the second versions of the updateable smart contracts from the blockchain and calling the second versions of the updateable smart contracts at the retrieved addresses.

~~2019.~~ A system as in claim ~~1514~~, wherein the memory stores further instructions to perform operations including the non-updateable smart contract retrieving permanent data from the blockchain and passing the permanent data to the updateable smart contract, and the updateable smart contract computing and returning data based on computation of the permanent data back to the non-updateable smart contract for permanent storage to the blockchain.

~~21. A system as in claim 15, wherein the memory stores further instructions to perform operations including segregating the smart contract into a plurality of non-updateable smart contracts.~~

~~2220.~~ A system as in claim ~~112~~, wherein the memory stores further instructions to perform operations including:

- hard-coding an address of the ~~at least one~~ non-updateable smart contract into the first version of the updateable smart contract prior to deploying the first version of the updateable smart contract;

- deploying a second version of the updateable smart contract;

- calling the second version of the updateable smart contract;

- the second version of the updateable smart contract calling the ~~at least one~~ non-updateable smart contract at the hard-coded address; and

the [at least one](#) non-updateable smart contract storing any permanent data to the blockchain.