## CLAIMS

1. A computer-implemented method (400) comprising:

generating, by a client (302, 420) associated with a target blockchain network (306, 410), a request for operating cross-chain data of one or more blockchain networks (316, 318, 320) different from the target blockchain network, wherein the request for operating cross-chain data comprises one or more smart contract computational logics for operating the cross-chain data and wherein the client cannot read data from, and exchange data with, the one or more blockchain networks different from the target blockchain network;

sending, from the client, the request for operating cross-chain data to a smart contract service provider (308, 430) comprising a trusted computation execution environment TEE (310), wherein the smart contract service provider is off the target blockchain network;

receiving (414), by the smart contract service provider (308, 430), from the client (302, 420) associated with the target blockchain network (306, 410), the request for operating cross-chain data of one or more blockchain networks (316, 318, 320) different from the target blockchain network;

fetching by the smart contract service provider (308, 430), client data from the target blockchain network;

sending (435), by the smart contract service provider to a data visiting service provider (312, 440), a request for the cross-chain data;

receiving (439), by the smart contract service provider, the cross-chain data from the data visiting service provider;

generating (441), by the TEE, a result using the client data and the cross-chain data, wherein the result is generated by the TEE executing the smart contract computational logics using the client data and the cross-chain data, independently of the target blockchain network; and

returning (443), by the smart contract service provider, the result to the client.

2. The computer-implemented method of claim 1, wherein the cross-chain data of the one or more blockchain networks (316, 318, 320) further comprise private cross-chain data obtained from a private blockchain network (316).

3. The computer-implemented method of any one of claims 1 or 2, wherein the smart contract computational logics for operating the cross-chain data are self-designed by the client (302, 420).

4. The computer-implemented method of any one of claims 1 or 2, further comprising proving, by the smart contract service provider (308, 430) to the client (302, 420), that the TEE (310) comprises a virtual machine operable to execute the smart contract computational logics in the request for operating cross-chain data.

5. The computer-implemented method of claim 1, further comprising one or more of:

prior to receiving (414) the request for operating cross-chain data from the client (302, 420), proving, by the smart contract service provider (308, 430) to the client, that the smart contract service provider includes the TEE (310);

proving, by the smart contract service provider to the data visiting service provider, that the smart contract service provider includes the TEE, and verifying, by the smart contract service provider, that the data visiting service provider (312, 440) includes a TEE; and

uploading, by the smart contract service provider, the result to the target blockchain network (306, 410), and, prior to uploading the result to the target blockchain network, proving, by the smart contract service provider to the target blockchain network, that the smart contract service provider includes the TEE.

6. The computer-implemented method of claim 1, wherein

the result is signed by the TEE (310) using a private key; and/or

the cross-chain data are obtained from two or more blockchain networks (316, 318, 320).

7. The computer-implemented method of claim 6, wherein

the result is signed by the TEE (310) using a private key, the method further comprising decrypting, by the client, the received result using a public key corresponding to the private key.

8. The computer-implemented method of claim 1, further comprising:

prior to sending the request for operating cross-chain data to the smart contract service provider (308, 430), encrypting, by the client (302, 420), the request for operating cross-chain data.

9. A system comprising:

a client (302, 420) associated with a target blockchain network, comprising:

one or more computers; and

one or more computer-readable memories coupled to the one or more computers and configured with instructions executable by the one or more computers to:

generate, a request for operating cross-chain data of one or more blockchain networks (316, 318, 320) different from the target blockchain network, wherein the request for operating cross-chain data comprises one or more smart-contract computational logics for operating the cross-chain data, and wherein the client cannot read data from, and exchange data with, the one or more blockchain networks different from the target blockchain network;

fetch by a smart contract service provider (308, 430), client data from the target blockchain network;

send the request for operating cross-chain data to a smart contract service provider (308, 430) comprising a trusted computation execution environment (TEE, 310), wherein the smart contract service provider is off the target blockchain network; and

a trusted computation execution environment (TEE, 310), wherein the TEE comprises:

one or more computers; and

one or more computer-readable memories coupled to the one or more computers and configured with instructions executable by the one or more computers to:

receive (414), from the client associated with the target blockchain network, the request for operating cross-chain data of one or more blockchain networks (316, 318, 320) different from the target blockchain network;

send (435) to a data visiting service provider (312, 440), a request for the cross-chain data;

receive (439), the cross-chain data from the data visiting service provider;

generate (441), a result using the client data and the cross-chain data, wherein the result is generated by the TEE executing the smart contract computational logics using the client data and the cross-chain data, independently of the target blockchain network; and

return (443), the result to the client.