Fractal Core Technical White Paper

Fractal Team*

v 0.1

Abstract: For the token economy, it is essential for a bottom-level public chain platform with the primary goal of supporting the development of token economy. Fractal is such a project. Fractal Core is the first core product of Fractal. This white paper describes the core composition, technical innovation and implementation of Fractal Core in details.

1 Introduction

Fractal is a public chain project jointly initiated by FCoin digital asset trading platform and token economy supporters, with the aim to not only support FCoin's own practice and exploration, but also the core aims of the development of the token economy in the future. Fractal Core is the first core product of the Fractal project, an application-level blockchain framework with the basic functions required for a high-performance public chain[1].

Fractal Core guarantees a balance between decentralization and efficiency through an improved DPOS consensus protocol protocol[2]. Through the optimization of transaction fee collection and reward mechanism, it promotes the fairness of community interest distribution and the positive cycle of the token economy. FVM, the light-weighted smart contract virtual machine has efficient performance and support powerful development languages. FVM has a lightweight execution environment, low-coupling design, rich tools and libraries, making it easy for blockchain developers around the world to develop, deploy, and even upgrade their smart contracts when allowed. Fractal Core endogenously supports the issuance and circulation of the Tokens and has designed snapshot function for application scenarios such as dividends and voting. In addition, through a flexible Map-Sidechain mechanism, Fractal Core can map any type of real-world assets to Fractal and achieve efficient circulation and diversified governance through sidechain mechanisms. From a business perspective, through the Map-Sidechain mechanism, a large number of rich business models that conform to modern business rules and the spirit of the Internet can be introduced and created to promote the in-depth development of the token economy.

*Email: team@fractalproject.com

2 Technology Implementation

2.1 Consensus Algorithm

2.1.1 BFT-DPOS

Consensus is the basis of blockchain. The concept of consensus varies in different contexts, such as machine consensus, governance consensus, and market consensus. The consensus involved in this paper refers to the machine consensus, that is, in an insecure peer-to-peer network and the absence of central node coordination, through the predefined algorithm rules (consensus algorithm), to ensure that the distributed ledger is consistent on different network nodes, and the compute nodes operate consistently, safely, and stably.

Fractal Core uses the DPOS algorithm to enable all users who hold FToken on the blockchain network to participate in the block generation work. Each FToken holder can get corresponding votes number base on FToken he/she held and vote for the witness node. 9 witness nodes will be generated after the voting is completed, whom will be eligible for the block generating.

Fractal Core adds Byzantine Fault Tolerance[3] improvements to traditional DPOS algorithms to reduce block validation time. The traditional DPOS block generating and confirmation is to gradually increase the effectiveness of the blocks generated by some witness nodes block after the new block is generated. The new block is considered to be irreversible after a certain amount of confirmation. Once the new block is generated, Fractal Core witness nodes immediately spread the block and request other witness nodes to confirm. the block can be considered irreversible after more than two-thirds of the witness nodes confirmation. With this improvement, Fractal Core can confirm that the block is irreversible within milliseconds after the block is generated, as shown in Figure 1.

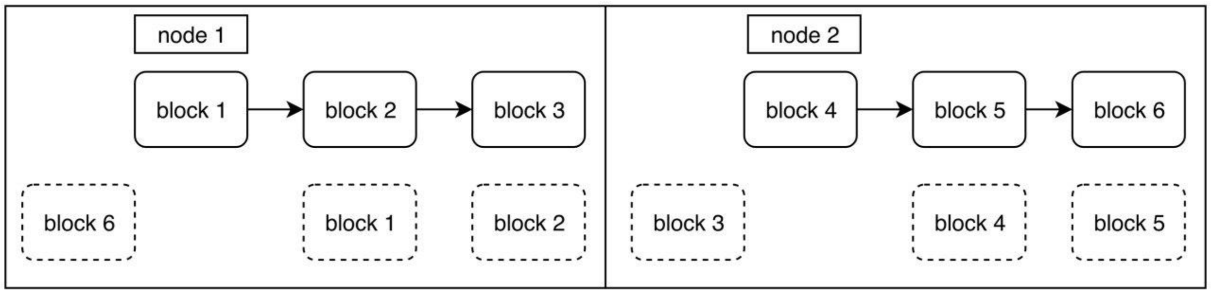

Figure 1. DPOS+BFT Blockchain Confirmation

After the voting is completed, the witness nodes will sort to ensure that the block-generating order is fixed. The blockchain network generates one block every two seconds, and each witness node continuously generates six blocks in turn. It will skip if a witness node does not generate a block within the specified time, and other witness nodes continue to generate the blocks; the witness node will be disqualified and replaced by new one if it does not generate the block within 24 hours.

In theory, no fork will appear for Fractal Core. All witness nodes generate blocks in a co-operative way, rather than competing with each other like POW[4]. Even with a temporary fork, the longest chain must grow faster than the short chain (Figure 2). The system will automatically discard the short chain and choose the longest chain through the longest chain mechanism similar to POW. If a malicious node appears, it will be voted out and no longer have the right to generate the blocks.

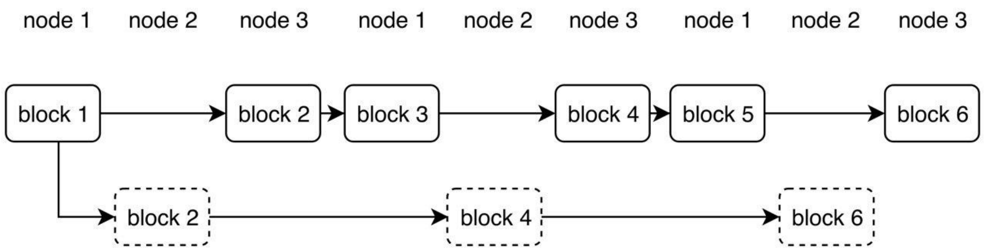

Figure 2. The Longest Chain Mechanism

2.1.2 Voting Mechanism

Every user has the right to vote. Users can get votes to vote based on the amount of FToken they hold. And the user can choose to use part of their FToken instead of all during the process.

The attenuation mechanism is added to the voting process in order to make the voting more active: when the user does not vote for a long time, his voting weight will be attenuated, and votes number will be attenuated weekly. If the user does not vote for one year, his votes number will be reduced to half, but fully restored once he resumes voting. FToken holders can get votes for voting and change their own votes option at any time. If the voting option has been changed, the votes will be removed from the original witness node and added to new witness node during vote transfer. Users who change their voting preferences will not cause double voting.

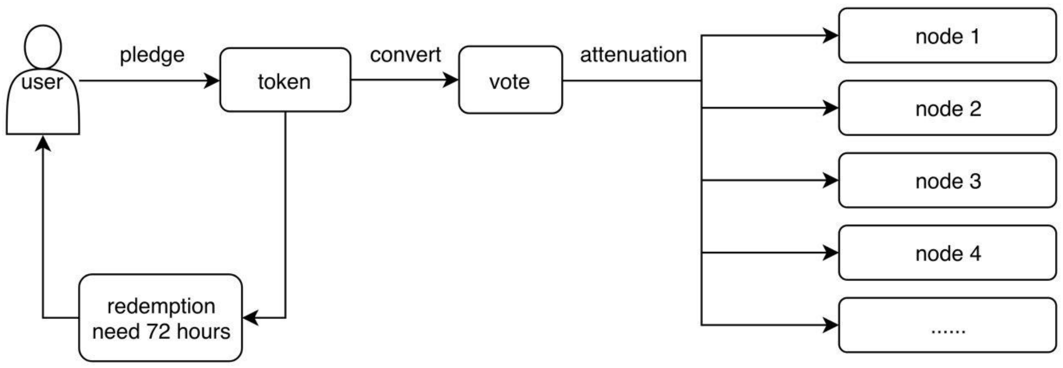

Figure 3. Voting Process

Community can initiate voting requests for functions, dividend mechanisms, or any proposal. The result of the vote is crucial because it determines the evolution of Fractal. Therefore, users should treasure their voting rights, so that the voting results fully reflect the overall will of the community.

2.1.3 Transaction Fee Charging

Since FToken will not be issued in the future, Fractal Core will no longer provide the coinbase reward for the witness node. The block generating reward of witness node is only the transaction fee charging of each transaction. Fractal Core will manage the resources on the chain in a unified manner and charge the users according to the resources using. The user needs to transfer a part of his token as the transaction fee to a Transaction fee account. After sending the transaction, Fractal Core will deduct the fee according to the network resources, computing resources and storage resources consumed in the network, and transfer the reward to the witness nodes handling these transactions. The purpose of the transaction fee is mainly to encourage the witness node to work, make the network more stable, and prevent malicious attacks on the network by setting the attack cost.

The users can adjust the priority of the transaction and set different transaction fees when sending the transactions. The higher handling fee can improve the transaction processing speed and ensure that the transaction is preferentially packaged when the network is congested. The fees are not all paid to the witness nodes but are distributed to the witness nodes and the FToken holders in a ratio of 1:4. The witness node receives 20% fees, while the remaining 80% is transferred to a fixed account and distributed regularly to all FToken holders.

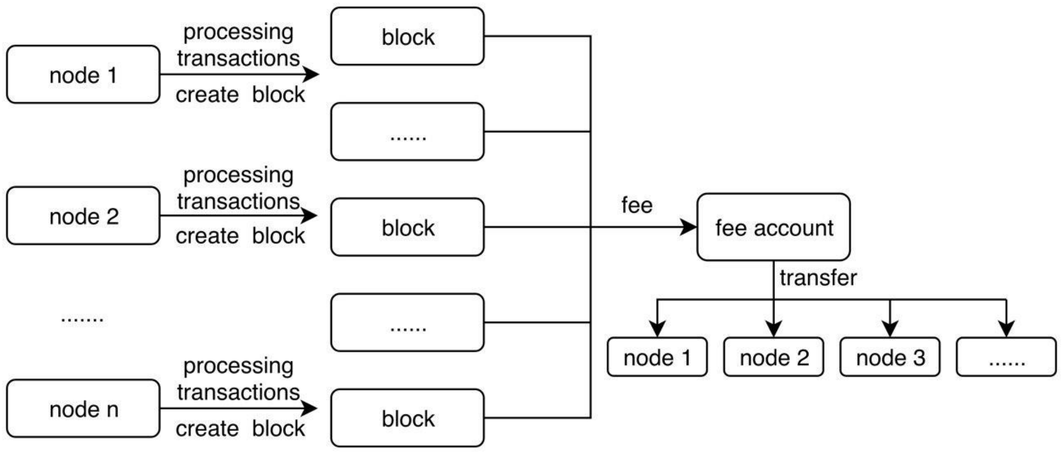

Figure 4. Transaction Fee Process

2.2 Virtual Machine

Fractal Core uses a lightweight, smart contract virtual machine FVM that delivers high performance and supports powerful development languages.

2.2.1 Compatibility

The Fractal Core Smart Contract Virtual Machine (FVM) is based on WebAssembly (WASM for short). WASM supports C/C++, Go, Java, JavaScript and many other programming languages. FVM can run smart contract applications developed in multiple languages. WASM is also the development direction of Ethereum's next-generation virtual contract engine EWASM, so EVM[5] can also easily access Fractal Core.

2.2.2 Lightweight Execution Environment

The WASM-based Fractal Core virtual machine allows precompiled contracts to be close to the execution efficiency of the original program and has the ability of quickly starting.

2.2.3 Low-coupling Design

Fractal Core does not limit the specific contract languages or the virtual machines. Any virtual machine that meets the sandbox mechanism and has sufficient operational efficiency can interface with the program API and run on Fractal Core. This design reduces the coupling between virtual machine, contract language and blockchain, and improves the universality of the contract system.

2.2.4 Smart Contract Deployment

The user can send the smart contract file (wasm format) and the contract external interface file (abi format) as a transaction to the node through the client. The node binds the smart contract code to the account that sent the transaction and persistently stores it on the blockchain. The contract call also can send a request to the node through the client, the node will find the corresponding contract code on the blockchain, call the virtual machine to execute the contract and return the result to the client.

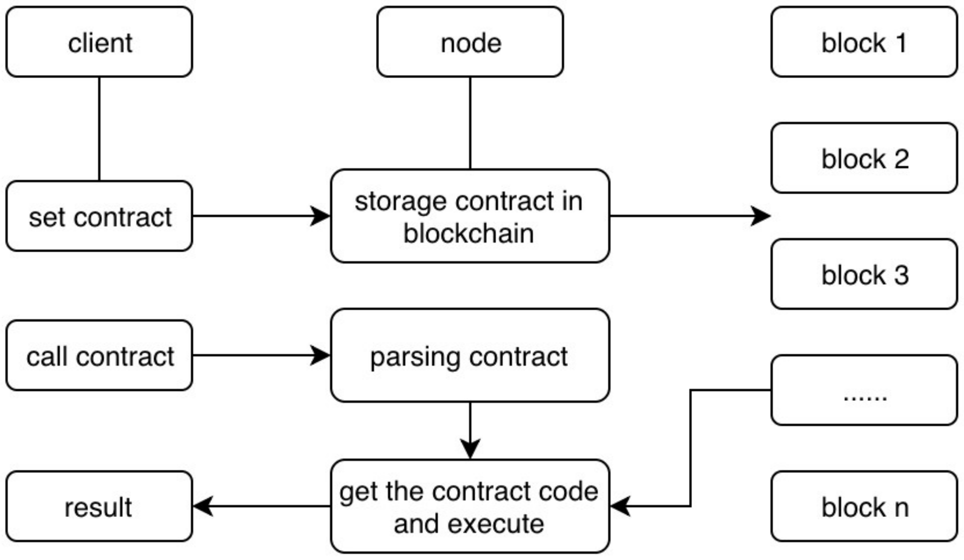

Figure 5. Contract Deployment

2.2.5 Smart Contract Upgrade

Unlike Ethereum, Fractal Core supports smart contract upgrades. Contract developers can collect feedback information during the testing phase and refine the contract in a timely manner through contract upgrades. For the concerns of contract users caused by this function, Fractal Core offers a solution to limit contract upgrades. The contract developer can modify the permission of the contract account, change the contract permission into multi-signature permissions, and the assignment of permissions can be queried by everyone.

2.2.6 Contract Model and Execution Process

The Fractal Core smart contract consists of two parts: the action set and the data definition.

(1) action refers to the function of a smart contract;

(2) The data definition refers to the data structure of the contract.

The client executes the functions provided by the smart contract by sending a transaction (the transaction contains one or more actions). If a transaction contains multiple actions, the transaction will only succeed if all actions are successful, otherwise, the data will be rolled back.

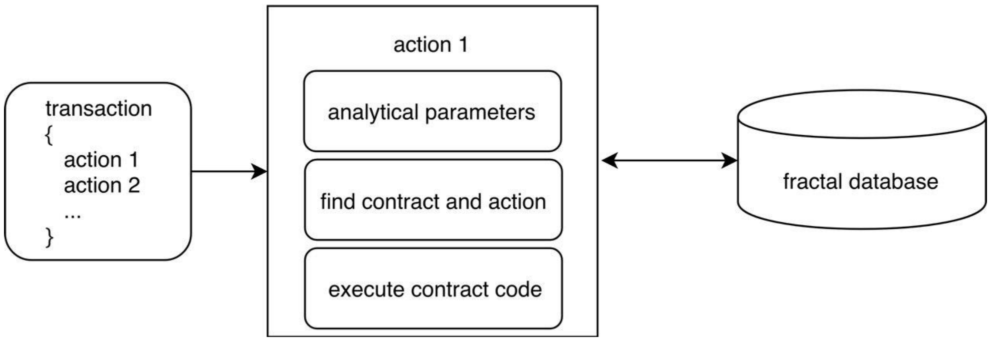

Figure 6. Action Processing

2.2.7 Tools and Libraries

If you want to run a smart contract on FVM, you must convert the program file to wasm format. Fractal Core provides conversion tools for C++ file, which help converted into wasm format and abi format for the convenience of contract developers. Considering the multiple application scenarios of smart contracts, Fractal Core provides a range of C/C++ standard libraries such as databases, inter-contract communication, containers, encryption algorithms, transactions, assets, and etc.

2.3 Account

Fractal Core uses the account model that enables a unique, easy-to-read string on the chain to represent the account. Accounts can be named by the creator based on his/her personal preferences. Fractal Core uses an elliptic curve digital signature algorithm[6], supports multiple signatures[7], and individuals or organizations owned accounts.

The account is bound to the user's public key. The user needs to sign it with the private key corresponding to the account when conducting a transaction or pushing some other transaction to the blockchain.

2.4 Snapshot

In order to meet the actual needs of dividends, voting and other scenarios, Fractal Core designed the asset snapshot function. The user can create a snapshot file and read the snapshot file information by RPC (Remote Procedure Call) request sent to witness node through the client.

2.4.1 Storage of Token Information

The Token information is stored in the memory of the node in the form of a multi-index table. Multi-index tables have more powerful data search capabilities and performance to retrieve data items within O(1). The multi-index table can provide multiple access interfaces for the same data, allowing the data to be created, deleted, updated and retrieved according to any index.

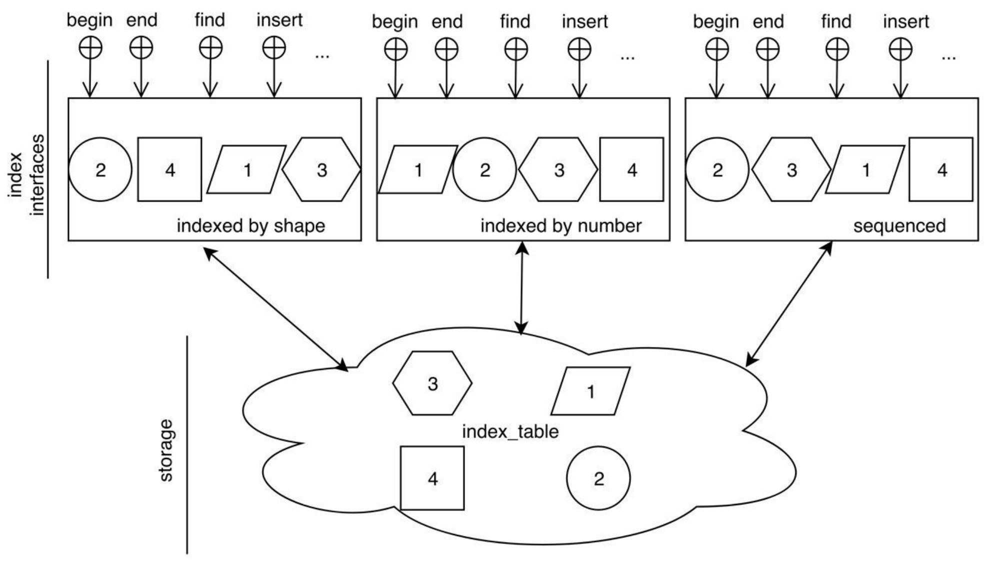

Figure 7. Token Information Storage

2.4.2 Creating a Snapshot File

When the user sends an RPC request to create a snapshot file to the node through the client, the request needs to include a Token contract (such as a system Token contract provided by Fractal Core) and a specific Token symbol parameter. The node will find the Token-holding accounts and balance in the multi-index container through the Token contract and symbol and store the information in the binary file (Token contract, Token symbol and current block height are reflected in the file name for easy query).

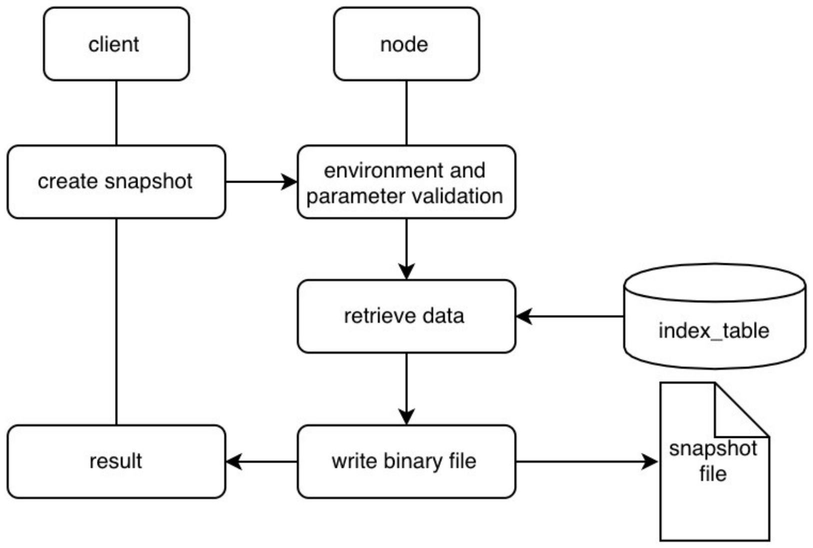

Figure 8. Snapshot Creation

2.4.3 Reading Snapshot File Information

When the user sends an RPC request to read a snapshot file to the node through the client, the request needs to include a Token contract, a specific Token symbol and block height parameter. The node can search through the parameter to find whether there is a snapshot file that meets the requirements in the specified path.

If it exists, the node will obtain the information such as the Token account and balance from the file and return it to the client, and the client will display the data in JSON format.

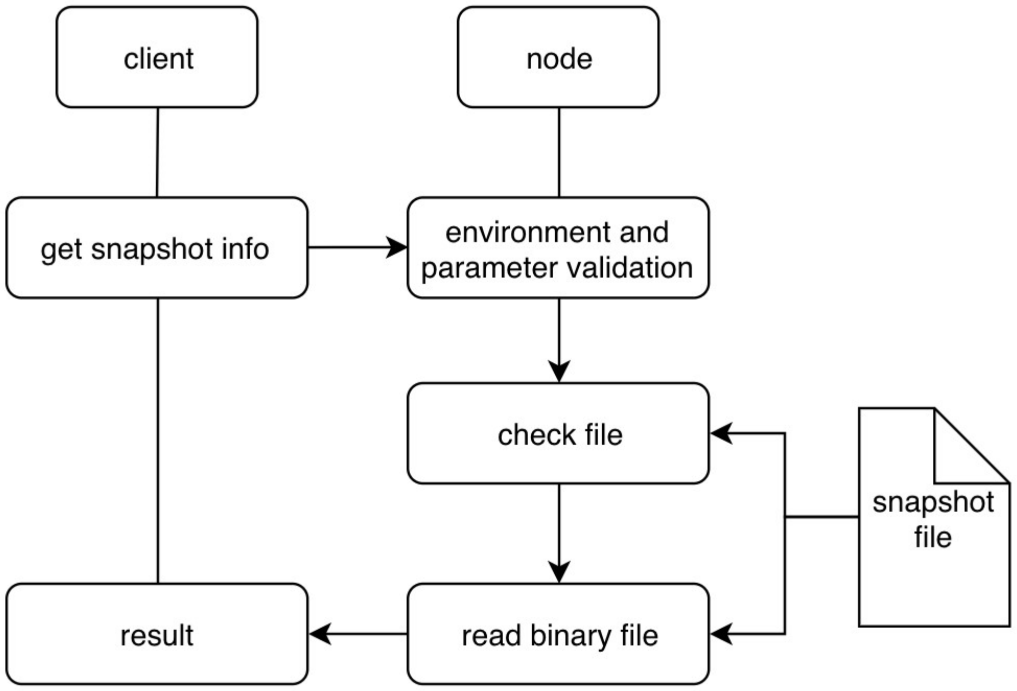

Figure 9. Snapshot Reading

2.5 Sidechain

2.5.1 Sidechain Mode

The sidechain supports both custodian and alliance modes.

(1) Custodian mode refers to sending the asset to the mainchain custodian party (trusted institution or individual). The custodian party will activate the corresponding digital asset on the sidechain after receiving the relevant asset information. Because of the centralized trend of the custodian model, a “declaration” will be required by the custodian party so that users can trust it.

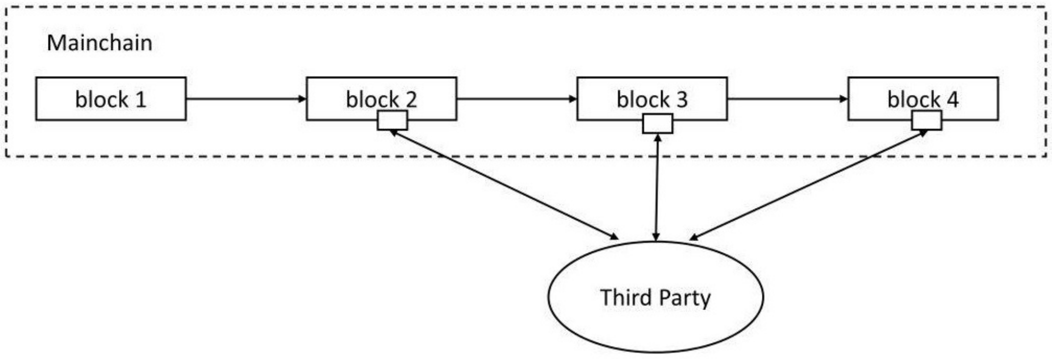

Figure 10. The Custodian Mode

(2) The alliance model uses a custodian alliance to replace a single custodian, using a two-way peg technique to transfer the mainchain assets to the sidechain. The sidechain uses the consensus algorithm of the custodian alliance to confirm the circulation of digital assets. In this model, you need to break through the alliance's own consensus algorithm if you want to steal the frozen digital assets in the mainchain. The sidechain security still depends on the trust of the alliance. The custodian can use consensus algorithms such as POW, POS[8, 9], DPOS, BFT[10], and POA[11]to obtain credit.

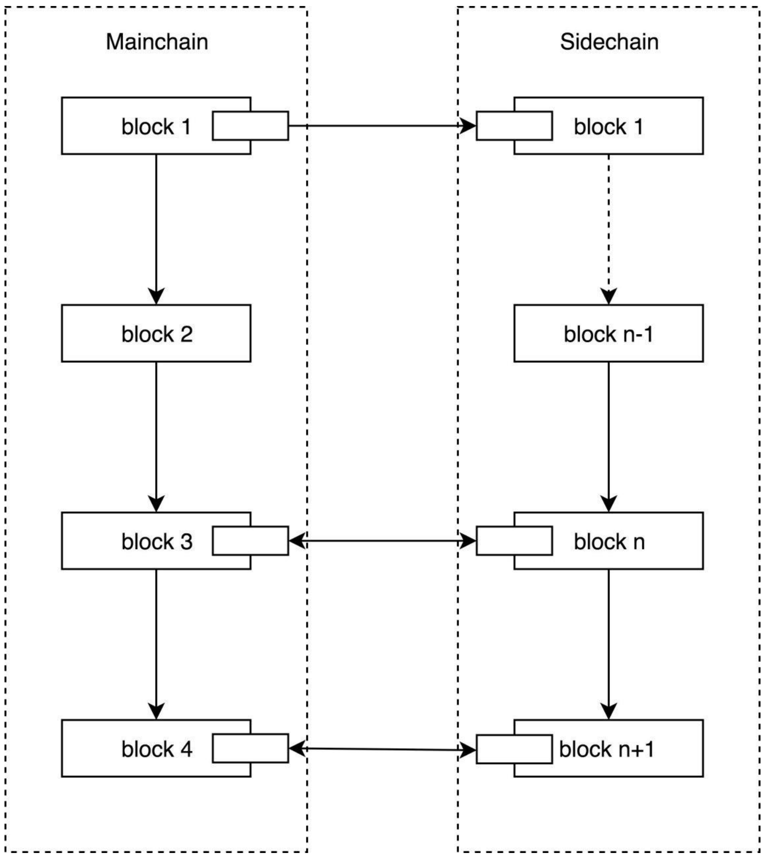

Figure 11. Alliance Mode

The sidechain is a separate blockchain. Under the premise of meeting the Fractal Core technical specifications, the ledger, consensus mechanism, transaction type and contract support can be customized according to their own needs. The sidechain can issue its own digital assets, but it needs to be bound to the mainchain through the Map-Sidechain mechanism. When a digital asset circulates in a sidechain, the corresponding "declaration" on the mainchain will be locked until the digital asset returns from the sidechain back to the mainchain. The Map-Sidechain mechanism can put some customized or high-frequency transactions on the sidechain to extend the mainchain.

2.5.2 Map-Sidechain Process and Role

Map-Sidechain main process:

(1) The initiator deploys a contract of "declaration" on the mainchain.

(2) The creator deploys the sidechain according to the relevant instructions or proof in the contract declaration.

(3) The creator finishes the binding by locking the asset or submitting other certificates.

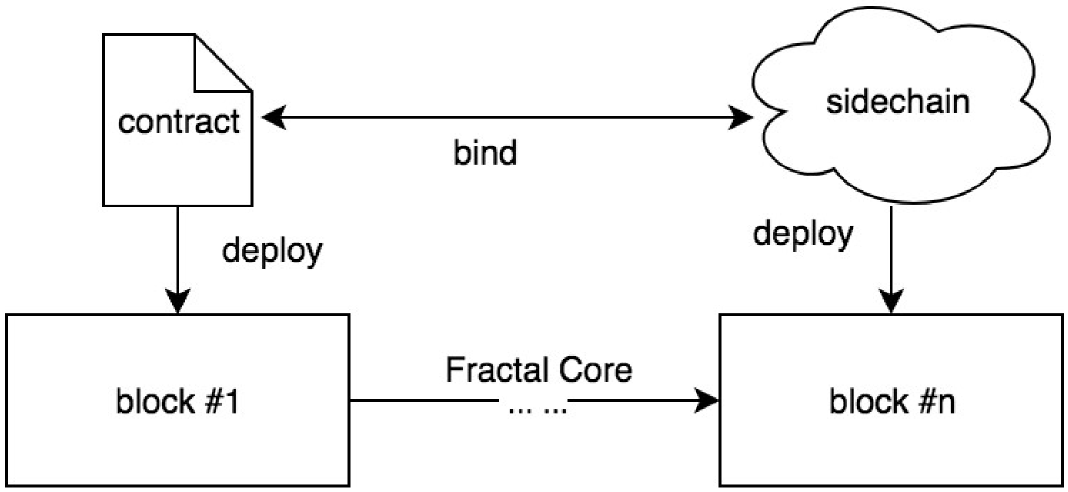

Figure 12. Map-Sidechain Process

There are three types of roles in Map-Sidechain which are ordinary users, sidechain consensus participants, and Fractal Core smart contracts.

Ordinary users are mainly the holders of various assets and Tokens, who can perform inexpensive and frequent interactions on the sidechain.

Sidechain consensus participants are who generate blocks and package the block information into Fractal Core contracts.

Fractal Core smart contract is used to lock assets, store a small amount of sidechain data, verify the rationality of sidechains, and handle anomalies in the sidechains.

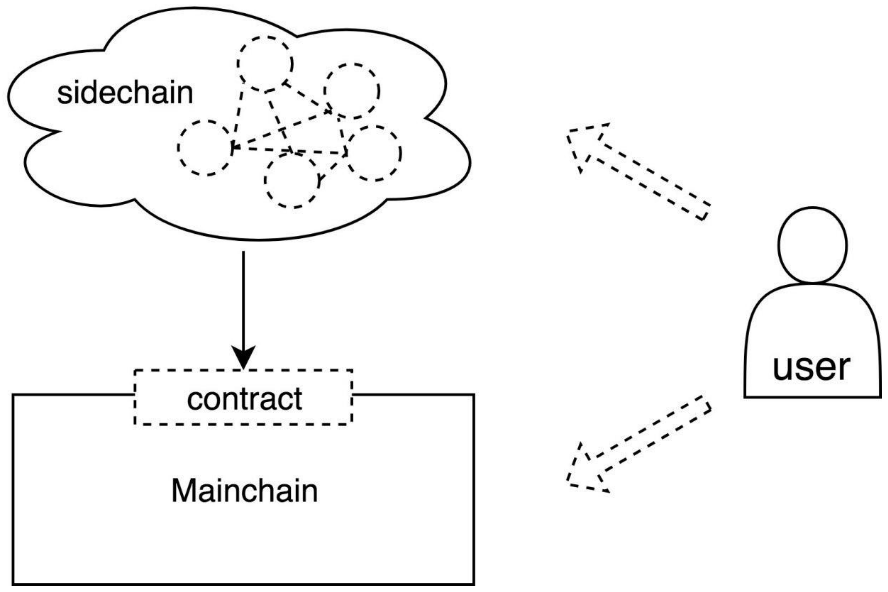

Figure 13. Map-Sidechain Model

2.5.3 Side chain block-generating mode

There are two ways to generate a block from sidechain:

(1) The creator deposits the bound sidechain through the Fractal Core contract, and the mainchain smart contract constructs a deposit transaction and locks, while generating a block on the sidechain;

(2) The sidechain can use a variety of consensus algorithms, such as POW, POS, DPOS, BFT, POA, and so on, to generate the block by consensus nodes.

2.5.4 Sidechain Exit Mode

There are two exit modes which are:

(1) Partial exit: the user transfers the assets declared in the Map to the contract through the side chain, and the corresponding assets can be withdrawn after confirmation of the sidechain consensus participants.

(2) All exit: initiated by the sidechain, and all the issued assets are transferred to the contract. The sidechain user verifies, unbinds the “declaration” or unlocks the deposit after some waiting time. If a user was found to submit a forgery certificate, a certain percentage of the deposit will be deducted and awarded to the evidence submitter, the exit operation will fail.

3 Conclusion

Based on the above technical implementation, Fractal Core can achieve the following characteristics:

High Performance: Fractal Core uses improved efficient consensus algorithm DPOS to achieve processing performance of thousands of TPS. Shorter block time intervals allow for very low transaction delays. Fractal Core enables an efficient payment network and smart contract platform to lay a solid foundation for supporting a wider range of application scenarios.

High Scalability: Through the Map-Sidechain mechanism, Fractal Core allows different applications and assets to run on different sidechains, while allowing them to migrate and convert between the main chain, and also allowing Fractal Core related applications to be easily and efficiently extended. Each side chain enriches the entire Fractal ecosystem through a flexible, customized approach to develop economic models and related applications.

Usability: Through the account model, efficient virtual machines, powerful smart contract language and related development tools, and efficient snapshot mechanism, Fractal Core makes it easy for to use and develop for users and developers.

Value Sharing: Fractal Core witness node distribute 80% of the transaction fee to FT holders, which can motivate FT users to participate in the development and operation of the community and better promote the development of Fractal ecology.

References

[1] Fractal team. Fractal Core µ A Decentralized System For Token Economy[EB/OL]. https://www.fractalproject.com/fractal-whitepaper-en.pdf, October, 2018.

[2] Larimer D. Delegated proof-of-stake (dpos)[J]. Bitshare whitepaper, 2014.

[3] EOS.IO Technical White Paper[EB/OL]. https://github.com/EOSIO/Documentation/blob/master/zh-CN/TechnicalWhit ePaper.md, July, 3, 2017.

[4] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. 2008.

[5] Bhargavan K, Delignat-Lavaud A, Fournet C, et al. Formal verification of smart contracts: Short paper[C]//Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security. ACM, 2016: 91-96.

[6] Johnson D, Menezes A, Vanstone S. The elliptic curve digital signature algorithm (ECDSA)[J]. International journal of information security, 2001, 1(1): 36-63.

[7] Yi L, Bai G, Xiao G. Proxy multi-signature scheme: a new type of proxy signature scheme[J]. Electronics Letters, 2000, 36(6): 527-528.

[8] E. Foundation, "Ethereum’ s white paper," https://github.com/ethereum/wiki/wiki/White-Paper, 2014.

[9] Vitalik Buterin. Ethereum Sharding FAQ. https://github.com/ethereum/wiki/wiki/Sharding-FAQs, August, 2018.

[10] Castro M, Liskov B. Byzantine fault tolerance: U.S. Patent 6,671,821[P]. 2003-12-30.

[11] Bentov I, Lee C, Mizrahi A, et al. proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y[J]. ACM SIGMETRICS Performance Evaluation Review, 2014, 42(3): 34-37.