Jingtum Technical White Paper

SWTC Foundation

January 2018

Contents

1 Introduction

This section will introduce some important aspects of blockchain technologies and current main solutions.

1.1 Background

Almost all of the technological development of computer systems focus on centralized solutions, including the Internet, big data, cloud computing, mobile communications, and so on. It is very natural to choose the centralized architecture, since it can simulate most of the users' needs very well and the concept of centralization has permeated itself with the development of computer technology.

Decentralisation has always existed, from the very beginning of Napster to the subsequent BitTorrent. This concept was not used in mainstream applications until the birth of Bitcoin in 2008. Decentralization holds disruptive power and attracts public attention.

For the first time, Bitcoin offers a decentralized trusted solution. Based on cryptography and mining (proof of work), it solves the problem of double spending of electronic assets in distrustful networks. There are two reasons for its success: 1. the widespread distrust of central banks caused by the subprime mortgage crisis in the USA, and 2. the usage of cryptography to provide an epoch-making blockchain technology. This offers a practical decentralized solution first for the financial industry. Moreover, the implementation of blockchain technology is reliable and straightforward, which makes it very popular.

Compared to traditional financial technologies, blockchain-based applications developed rapidly. From its birth to the present, a variety of digital cryptocurrencies (such as Ethereum, Ripple) were created. Each new currency had been designed to solve a specific practical problem in a better way. However, a variety of coins, including Bitcoin itself, still has various limitations of its own. On the contrary, the underlying technology, blockchain, which supports Bitcoin, has tremendous power. In the beginning, mainstream society, including governments, was hostile to Bitcoin. However, its immense success has forced mainstream society (in particular banks) to recognize and learn about it, especially in the area of blockchain technology. Many banks have invested a lot in this field, to keep updated with the latest technological developments. At the same time, it has become more popular to use blockchain technology to reform the traditional financial industry. The latest developments include Citibank's CitiCoin, the Bank of England's RSCoin, and the idea of digital money mentioned by Zhou Xiaochuan, Chinese Central Bank governor.

1.2 Block and Data

Blockchain technology is the foundation of each electronic cryptocurrency. A blockchain is a transaction database, which stores the information shared by all nodes in the system, which is called the distributed encrypted master ledger. Through blockchain technology, the system implements features which do not require a central authority or a trusted third party to coordinate interaction, validate transactions and supervise behaviors. A complete copy of a blockchain contains every transaction that has been executed, such that any participating nodes can access any historical information on the network. In simple terms, a blockchain consists of three elements: shared state, updating rule and history-sensitive model. These three elements solve the three main problems of the distributed encrypted master ledger: 1. data storage functionality, 2. the updating rules for all nodes, to solve the problem of data security and, 3. using history to keep data consistent. In this way, blockchain technique enables data to achieve consistency by protocols in a network, which consists of multiple independent computers for the usage of the digital encryption technology. The security of data is thus guaranteed.

1.3 Consensus

The consistency of data is realized through the consensus protocol. A consensus protocol specifies a uniform rule for all computing nodes in a system. The Byzantine behavior of each node, and the ratio of the implementation of this rule, determines whether the whole system can achieve data consistency. By CAP Theorem, in a distributed system, consistency, availability and partition tolerance cannot be obtained at the same time. Therefore, it is essential for the whole system to choose a proper consensus for specific applications. The common consensuses for blockchain-based systems are PoW, PBFT, PoS, etc.

1.3.1 Proof of Work (POW)

Bitcoin and other similar coins use “mining” to ensure that each node selects the same blockchain. Their approach is to make the generation of each block very expensive, meanwhile, the protocol guarantees that all nodes agree to choose the longest chain, so even when the blockchain has forks, the system can still converge to the longest fork and abandon the shorter ones quickly. In the long run, the blockchain is unique.

1.3.2 Proof of Stake (POS)

Consider POW’s high energy consumption and other shortcomings, POS has attracted more and more attention as an alternative solution. Peercoin was the first cryptocurrency to use it, which approach is that each node verifies the transactions in the system by the proportion of shares held by each node. Because everyone is a stakeholder in the system, the normal rational participants should maintain the system operation. The exact details of each implementation of POS are different.

1.3.3 PBFT

Multiple nodes consensus method is adopted to ensure that each block is voted by everyone. The problem of Byzantine Generals is mathematically solved. Theoretically, 1/3 fault-tolerant rate in the system can be guaranteed.

1.4 Smart Contract

Blockchain technology itself is a transaction-oriented distributed storage solution. In real scenarios, various applications can be created on this transaction-oriented solution. All of them can be implemented by a snippet of programs that are called Smart Contract. Usually, Smart Contracts are considered as part of the Blockchain. It is usually presented together with cryptocurrencies, but it is more appropriate to regard smart contract as the application of blockchain technology.

A “Smart” Contract is actually dumb, because it is defined by fixed code and executed line by line. The code is prewritten, and once it is executed the process cannot be interfered with by third parties.

In order to implement Smart Contracts, usually, cryptocurrencies need to provide support within the consensus mechanism by scripting languages or Turing complete programming languages. The latter usually requires a virtual machine to isolate them from other modules.

1.5 Bitcoin

Bitcoin is the first widely used cryptocurrency based on Blockchain, and known as the “digital gold” of cryptocurrencies. It serves as a decentralized trusted electronic currency. The Bitcoin-based applications are limited by its functionality. Some shortcomings of bitcoin are:

1. Slow execution. The generation of each block takes 10 minutes, and the

confirmation time is much longer.

2. Limited trade capacity. The number of transactions contained by each block is limited for its size. At the same time, due to the decision mechanism of Bitcoin, it is hard to effectively increase this number.

3. Huge energy consumption for POW consensus. The road ahead for bitcoin is limited since the value of Bitcoin must be maintained by the continuous growth of computing power.

4. Implementation of scripting languages. Bitcoin uses the scripting languages to implement some simple contract functions. The contract itself does not support Turing complete programming languages.

5. Crowded network. The Bitcoin network itself has been very crowded with transactions, that limits Bitcoin-based applications.

1.6 Sidechain

Since the Bitcoin blockchain cannot fully provide the function of Turing complete contract, one possible solution is to use a sidechain. A sidechain is a blockchain system that runs individually outside the bitcoin blockchain. This system can be implemented in either a similar or a different way compared to Bitcoin. A user may switch flexibly between the bitcoin system and the sidechain system. The main advantage of using a sidechain would be the flexible implementation of the possible choice of Turing complete implementation, thus compensating for the weaknesses of the bitcoin systems.

Generally speaking, when entering the sidechain from the bitcoin, the user sends bitcoins to a system address to lock. Then, in the corresponding sidechain system, the relative amount of the sidechain currency will be sent to the user's wallet. This process is relatively simple and easy to implement. When a user wants to convert the sidechain currency, he can obtain the corresponding bitcoins in the bitcoin system by some verification (e.g., SPV verification).

Because of the characteristics of Bitcoin, any changes to its protocol will require a lengthy discussion. Thus, any extended functionality can only be integrated by soft forking. There is no possibility for hard forking. Therefore, this significantly limits the implementation of sidechain, and the problem remains as to how to go back to the bitcoin system from the sidechain system.

It is very difficult to integrate the smart contract (Turing complete) by reforming the bitcoin system, and the sidechain system is disproportionately matched for bitcoin systems. As long as the sidechain develops to some degree of profitability, hackers will attack the sidechain to gain additional benefits. Moreover, the flaws in the complex protocols are obvious, thus it is very difficult for this approach to work.

1.7 Ethereum

Ethereum is a significant update to Bitcoin. Its main feature is to support the Turing complete smart contract. It uses the POW consensus to speed up the block generation and execute contracts through a virtual machine, so that contract execution can also modify consensus output. Its main shortcomings include:

1. POW wastes resources, and POS implementation is uncertain in the future.
2. Each node needs to verify all the contracts. Hence the execution efficiency of the whole system cannot outperform the efficiency of a single computer.
3. Communication among smart contracts is very difficult.
4. The bug of contracts directly affects the stability of the system because of the consensus between contracts and transactions is bound.
5. Possible hard forking will continue to cause damage to the entire system.

1.8 Centralization and decentralization

Decentralization is the most essential characteristics of blockchain technology. During the implementation of a blockchain system, its actual deployment and some other factors may decrease the element of decentralization, e.g., partial centralization, a partial concentration of computing power, and a partially structured network.

Bitcoin is the most open, public and decentralized system. However, due to its design principles of POW consensus protocol, and the development of the computing power. Mining pools gradually gather the computing power in the network. A single mining pool has the enormous computing power and becomes a virtual center of Bitcoin, whose importance expands with the growth of computing power. Besides some blockchain systems contain centers in the beginning. For example, Ripple, and hyperledger initially contain validating nodes in the system design. A node in the network can be approved to be a validating node – this is partial decentralization.

Complete decentralization means absolute freedom and absolute privacy. However, from a practical point of view, this is only an ideal situation and a goal. Pure freedom is not always good, and the choice of the whole network is not always optimal. Instead, it is more important for a better application to be developed and applied at the right time through the proper methods.

2 Jingtum blockchain

This section will present the details of the Jingtum blockchain, and its improvements and optimization to the existing systems.

Design goals

The Foundation is developing the Jingtum blockchain (the Jingtum Chain) which is designed to be a stable and user-friendly platform for Enterprises and users. The Jingtum Chain is decentralized with open-source software based on a cryptographic protocol. It exists on a peer-to-peer network hosting the public transaction ledger. Thus, enterprises may easily access Jingtum Chain and enjoy the benefits of blockchain technology, without fully delving into the details of the underlying technology. Further, on the Jingtum Chain, companies have a flexible choice whether or not to share their customers. Each new application built on the Jingtum Chain will attract new customers to the ecosystem, and at the same time, applications are exposed to a larger pool of existing customers. Such an ecosystem will lead to a virtuous cycle realizing “all for one and one for all”.

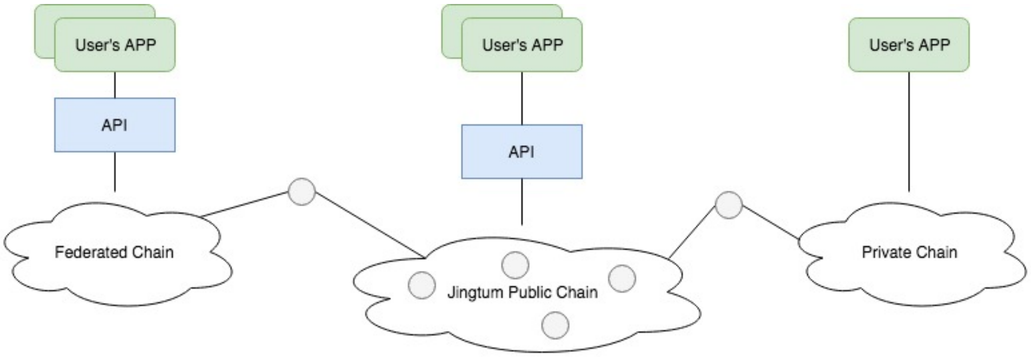
2.1 Effective decentralization

Due to the shortcomings of existing blockchains, the Jingtum Chain will adopt an effective solution of decentralization. That is to avoid the lack of efficiency and yet maintain the reliability of decentralization at the same time. The solution is to choose an efficient optimization interval, and not an extreme point. In the spectrum from absolute centralization to complete decentralization, there is a broad range of possibilities – there is no reason only to choose one of these two extremes. According to different application scenarios in various industries, the most economical and convenient balance points could be found, which solution would meet users' requirements, i.e., an effective decentralization point may be selected where users enjoy the security and cost advantages of decentralization, yet at the same time avoiding the efficiency reduction through excessive decentralization.

The Jingtum Chain is designed to select the most effective balance point, and optimizes it automatically to the best point according to the user's specific scenarios and requirements. According to users' needs, they can access the public chain, or build private blockchains or even union blockchains through the tools available on the Jingtum Chain. These private and federated blockchains may choose whether to connect to the public Jingtum Chain.

The consensus protocol of Jingtum Chain adopts randomized BFT. However, Jingtum Chain selects the validating nodes by POA (proof of Application). The core of Jingtum Chain contains several validating nodes that maintain the underlying validating network for the system. This network opens for public application access. DAPPs on Jingtum Chain refer to the applications based on Jingtum Chain for specific users. These applications can directly access to the public Jingtum Chain through the API provided by the Jingtum Chain, or deploy its private blockchain with the technology of Jingtum Chain. These applications can maintain a validating node. Such a node may implement two functions:

1. It is involved in the consensus of public nodes in the network on the Jingtum Chain;
2. It allows applications to connect to the network of the Jingtum Chain. If the application itself deploys a private chain, this node can convert the user's private token to Jingtum tokens.



Certainly, if an application only uses API to access the blockchain, there is no need to deploy a single authentication node.

When a user's private chain is connected to Jingtum Chain, usually, a gateway is required to implement the issuance and conversion of the user's token.

2.2 The architecture of the Jingtum Chain

A design goal of the Jingtum Chain is to avoid the shortcomings in current blockchain technology. Unlike Bitcoin, the Jingtum Chain integrates smart contract originally. Either unlike Ethereum, the underlying blockchain adopts a more reasonable multi-layer method, which separates the execution of the smart contract from the transactions; this would avoid the problem of contracts affecting the whole system, and allowing more flexible implementation of smart contracts.

The architecture of Jingtum Chain is as follows:

1. Instead of using POW, which results in wastage of resources, Jingtum Chain reaches consensus by RBFT; besides, it has high-speed parallel processing capability and supports mass users.

2. Jingtum Chain is multi-layered, the bottom layer is called TX Layer, which is responsible for handling the most basic transactions. The layer above, called the Contract Layer, deals with contracts. The elements of the contract (code, state, storage, transaction) are separated: the transaction part is transferred to the TX Layer and executed; the other parts are executed in the Contract Layer. This architecture separates the execution of the contract from the resulting transactions, hence allowing the contract and transaction to match the corresponding protocols by their respective characteristics, achieving maximum efficiency and security.

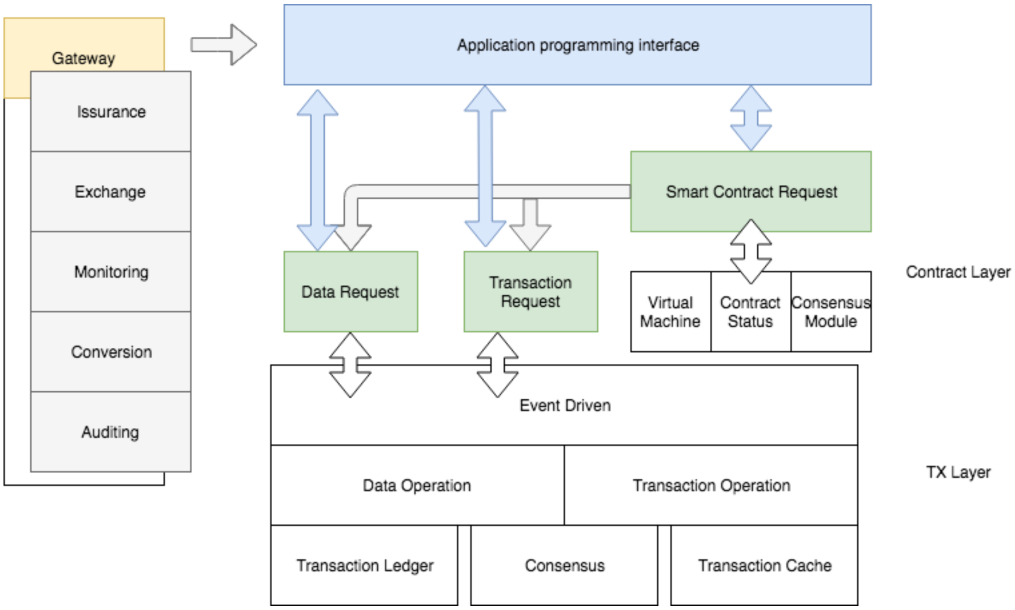
3. To address the increasing needs for supporting data of blockchain applications, the Jingtum Chain provides BLHR (block level hash record) data support, which enables users to save data signatures to the blockchain easily.

4. In order to improve the processing power of the whole system, sharding is introduced to the consensus node, so that not all nodes are required to do the same thing. Instead, for each transaction, a node is selected automatically and randomly to process the transaction. On the one hand,

this method effectively takes advantage of the processing power of many nodes, thus maintaining sufficient fault tolerance. On the other hand, this significantly reduces the information flow between networks and improves the overall efficiency of the network.

5. When a contract is created, the user can identify the number of consensus nodes and the conditions for it. A user can flexibly keep the balance between the cost and reliability, on the other hand, the Contract Layer could be more efficient with more contracts be handled. By this abstraction, the security of the contract system will not be affected.

6. The execution speed of the smart contract is decoupled from the ledger close speed of the TX Layer. The change in contract status depends on the consensus rate of the contract nodes.



2.3 Blockchain data

Blockchain is tamper-proof. All blocks are linked together to become a single chain by historical correlation - once a data is recorded, it cannot be tampered with. Direct modification of the data results in invalidation of subsequent blocks. Thus, this feature is widely used in areas of data security, identification and so on.

The typical usage of this feature would be to keep some information in the metadata of the transaction. Once a transaction is executed and stored in the blockchain, this included metadata is also permanently recorded in it. However, there are several shortcomings with this method:

1. Execution needs transactions. On one hand, some amount of transaction data must be sent, on the other hand, the transaction requires a digital signature. Therefore, the data record must correspond to a user account or wallet, and the corresponding private key information needs to be accessed.
2. The stored macro information is dispersed in every transaction, and every transaction must be traversed to search for it.
3. The process of data storage must be performed correctly by transaction confirmation.

According to this, Jingtum Chain supports BLHR (block, level, hash, record). Users may submit information that needs to be saved directly to the block. Each block has a single location to hold that information. If a user's information is historically correlated, he needs to provide a description of this correlation by himself - the block does not need to understand his application logic and merely needs to record the storage request. When each block is closed, the system automatically records all the BLHR information into the block.

2.4 Jingtum token

The Jingtum token, System Working Token China (SWTC), is the native cryptographic token of the Jingtum Chain. SWTC is designed to be used solely on the Jingtum Chain, which is required as virtual crypto “fuel” for using certain designed functions on Jingtum Chain (such as executing transactions and running the distributed applications on the Jingtum Chain), providing the economic incentives which will encourage participants to contribute and maintain the ecosystem on the Jingtum Chain. Computational resources are required for running various applications and executing transactions on Jingtum Chain. Developers will be able to build decentralized applications on Jingtum Chain, and users of these applications will be required to pay for the consumption of these resources (i.e. ”mining” on Jingtum Chain), and SWTC will be used as the unit of exchange to quantify and pay the costs of the consumed computational resources. SWTC is an integral and indispensable part of the Jingtum Chain, because, in the absence of SWTC, there would be no common unit of exchange to pay for these costs, thus rendering the ecosystem on the Jingtum Chain unsustainable. The commission fee exists to incentivize validators (which maintain the integrity of the Jingtum Chain) and prevent malicious users from an excessive deployment of smart contracts.

SWTC is a non-refundable functional utility token which will be used as the unit of exchange between participants on the Jingtum Chain. SWTC do not in any way represent any shareholding, participation, right, title, or interest of the Foundation, its affiliates, or any other company, enterprise or undertaking, nor will SWTC entitles token holders to any promise of fees, revenue, profits or investment returns, and are not intended to constitute securities in Singapore or any relevant jurisdiction. SWTC may only be utilised on the Jingtum Chain, and ownership of SWTC carries no rights, express or implied, other than the right to use SWTC as a means to enable usage of and interaction with the Jingtum Chain.

In particular, you understand and accept that SWTC:

1. is non-refundable and cannot be exchanged for cash (or its equivalent value in any other virtual currency) or any payment obligation by the Foundation or any affiliate;
2. does not represent or confer on you any right of any form with respect to the Foundation (or any of its affiliates) or its revenues or assets, including without limitation any right to receive future revenue, shares, ownership right or stake, share or security, any voting, distribution, redemption, liquidation, proprietary (including all forms of intellectual property), or other financial or legal rights or equivalent rights, or intellectual property rights or any other form of participation in or relating to the Jingtum Chain, the Foundation and/or its service providers;
3. is not intended to be a representation of money (including electronic money), security, commodity, bond, debt instrument or any other kind of financial instrument or investment;
4. is not a loan to the Foundation or any of its affiliates, is not intended to represent a debt owed by the Foundation or any of its affiliates, and there is no expectation of profit; and
5. does not provide you with any ownership or other interest in the Foundation or any of its affiliates.

The contributions in the token sale will be held by the Foundation (or its affiliate) after the token sale, and contributors will have no economic or legal right over or beneficial interest in these contributions or the assets of that entity after the token sale.

To the extent a secondary market or exchange for trading SWTC does develop, it would be run and operated wholly independently of the Foundation, the sale of SWTC and the Jingtum Chain. The Foundation will not create such secondary markets nor will it act as an exchange for SWTC.

2.5 Gateway and usercoin

The Jingtum Chain supports not only the SWTC, but also unique ”usercoins” which may be generated by a user. A usercoin can be regarded as a user-defined token of digital asset, and its issuance is initiated by a qualified third party, which is required to pass a compliance and risk assessment of the Jingtum Chain. After that, the third party is eligible to issue its usercoin on the Jingtum Chain. The usercoin is issued through the gateway. The issuer is responsible for its conversion. Once the usercoin has issued - it can be paid, circulated and transferred freely within the ecosystem on the Jingtum Chain, similar to Jingtum tokens, without further intervention by the issuer. However, the conversion of usercoin must be done by the gateway. The Jingtum gateway is the interface for a third party’s assets on the Jingtum Chain. The assets of that third party may enter the network on the Jingtum Chain through the gateway, and the corresponding usercoins are issued. If users wish to convert their assets, this is also done through the gateway. In Jingtum network, a usercoin marked by USDT is a good example. The implementation of the gateway can be either through a single server or a group of server clusters with failure tolerance. If Byzantine fault tolerance is needed, a consensus network is needed.

2.6 Hierarchy and smart contract

The implementation of the smart contract system on the Jingtum Chain is as follows:

1. The TX-driven approach is adopted. Transactions initiate the contract deployment and contract function call. If a user’s balance needs to be modified during the execution, a transaction will be initiated and sent to the TX Layer. All these transactions will be executed and validated in TX Layer, and recorded in the underlying blockchain.

2. Transactions in the TX layer are not affected by contracts.

3. Transactions between TX layer and Contract Layer store the code and

status of contracts. The status of contract refers to the call and parameters of the corresponding contract function. The TX Layer status hashing ensures the consistency of information.

4. The execution of the Contract Layer is performed by multiple contract nodes, which carries out consensus in a deterministic way.
5. Each contract node uses VM to execute codes.
6. The contract node preserves the storage of the contract execution.

Based on this hierarchical design, the Jingtum Chain is further optimized by using asynchronous contract calls. Based on this concept, the Jingtum Chain implements fast call and return of contracts, and allow users to perform smart contracts with sharding. All nodes are not required to do the same thing, which improves the processing capability of the whole system.

2.7 Asynchronous call of smart contract

The current execution of Smart Contract is synchronous – the contract call is triggered by transaction or automatically. During the execution of the contract, the consensus mechanism of the blockchain must wait for its end, plus only proceed to complete the current block consensus after the callbacks are returned. This method of execution for smart contracts has the following shortcomings:

1. The speed of contract execution seriously affects the generation time of the block.

   Since the block consensus depends on the contract execution results, each node must reach a consensus about the consistency of the contract results. Therefore, the speed of the contract execution directly affects the subsequent operation, and any delay in contract execution would also delay the block generation time.

2. The speed of contract execution severely affects contract concurrency supported by the blockchain.

If the frequency of the generation of the blockchain is fixed, in the same period, the execution speed of a contract will directly affect the execution of the other contracts in the same block. In extreme cases, a malicious contract may cause the system unavailable to other contracts, and the number of concurrent processing contracts could be greatly reduced.

3. The ability of fault tolerance is limited during the execution of contracts.

   Because of the synchronous execution mode, during the execution of contracts, a variety of error situations should be taken into account, and the fast processing of all kinds of time-sensitive operation should be realised. For example, the timeout situation of a variety of operations should be processed accordingly.

Some of the existing solutions, such as Ethereum, adopts a "gas" method, estimating computational demands of every contract, thus controlling the total amount of computation supported by the current block according to the gas number of a system, thus ensuring finishing of consensus punctually. The total number of excution supported by the system is limited by the gas number. As the contract gets more and more complex, the number of contracts supported by the whole system is decreasing - in addition, the Ethereum consensus time is limited, and the highest value of gas does not significantly increase. According to the problems in the existing Smart Contracts executions, Jingtum provides a block-crossing asynchronous call contract system, whose block consensus does not depend on the contract execution results. It can improve the executions concurrency of contracts and the number of contracts supported by blocks, i.e. improving the capability of fault tolerance.

The Jingtum Chain's asynchronous call contract system includes the following units:

1. The distributed system validating unit

   It consists of one or more service nodes and several validating nodes that receives the transaction request set TX submitted by users, including the contract call request TX and the payment request TX.

2. The distributed contract execution unit

   The local or remote distributed system execution unit communicates with the distributed system validating unit through a predefined protocol, to obtain the information required by the contract execution, and return the results to validating node when the execution of the contract is done.

3. The contract execution cache unit

   It receives the contract calls from the validating node, sends them to the distributed contract execution unit, receives the contract execution results, and returns the execution state of the current contract to the validating node, in order to realize the asynchronous contract call.

The blockchain system of Jingtum Chain uses asynchronous calls as its background core technology. Compared with the existing Smart Contract execution technologies, it has the following advantages:

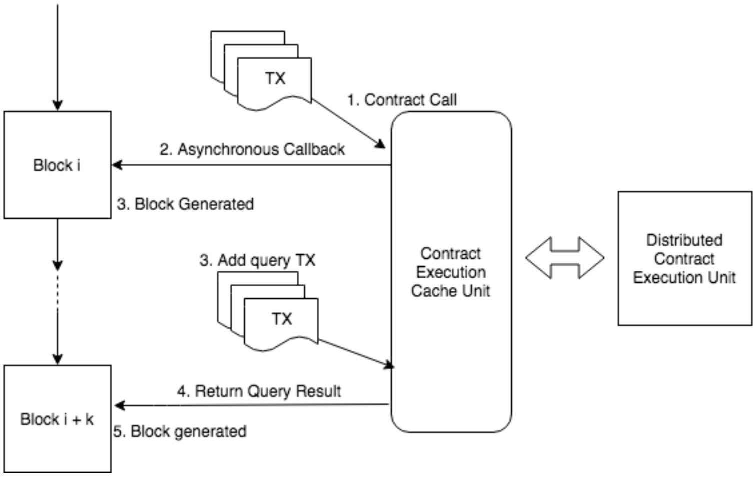
1. It isolates the contract execution from the system consensus unit, the execution of contracts can be remote, so that the execution of contracts no longer holds the resources of the system consensus;

2. It decouples the contract execution unit and the system consensus unit, to makes the contract execution module and consensus module relatively independent, and is pluggable.

3. It sets up the contract execution cache unit between the consensus validating unit and contract execution unit, adopts the asynchronous execution mode in the whole execution of the contract creatively, implements the contract call and cross-blocks execution (between $\mathrm{block}_{\mathrm{i}}$ and $\mathrm{block}_{\mathrm{i+k}}$), with the consensus among validating nodes. This asynchronous contract execution mode improves the executions concurrency of contracts, avoids waiting for the execution results of the contract during the consensus process, greatly increases the number of contracts supported by the block.

4. It improves the fault tolerance of the whole system. On the one hand, the system can set up an appropriate timeout handling mechanism to deal

with the delay of the contract; on the other hand, the user could configure appropriate K value in the contract call to make the long execution contract to be handled appropriately.



2.8 Fast smart contract transactions

The existing Blockchain-based distributed transactions are restricted by the consensus protocol, the generation time and the size of blocks. Blockchain-based transaction speeds are usually measured in seconds or even minutes. Here are some other shortcomings:

1. The transaction requests could be delayed in the distributed system, i.e., the transmission delay between the single node initiation and propagation of the whole network.
2. The consensus process could be delayed. The update of data must be written to the ledger after the completion of consensus, this writing is intermittent, the update is done in each validating cycle, the update of data requested by users must be responded and returned after the update cycle.

3. The existing smart contracts are not only affected by the two points above, but also by the delay in contract execution.

Some of the existing solutions, such as lightning network and tunnels on Bitcoin to speed up the processing of trade request, either their protocols are complicated, or the non-Byzantine fault tolerance is adopted. Hence, the extensive application of these solutions is limited.

Jingtum Chain implements a fast smart contract transaction system. Based on the asynchronous calls of contract, it divides the contract nodes into two kinds: normal transaction contract node and fast transaction contract node. The former one communicates with the validating nodes by a predefined protocol, then accesses required information in contract execution, and returns the results to the validating node after the execution of the contract is completed; the latter performs fast transaction request and returns results to the smart contract access server.

The implementation of fast smart contract call is as followings:

1. Fast transaction initialization

   two or more user's who want to use smart contract make the agreement and create a smart contract, init a fast smart contract initialization request transaction, smart contract access server propagate the transaction to validating nodes, and create a transactions group. Then validating nodes start to make the consensus, and send this transactions group to all the contract nodes after the consensus process finished. According to the predefined PBFT protocol, a Fast Smart Contract Transaction Node (FSCTN) will be selected by the distributed randomized algorithm.

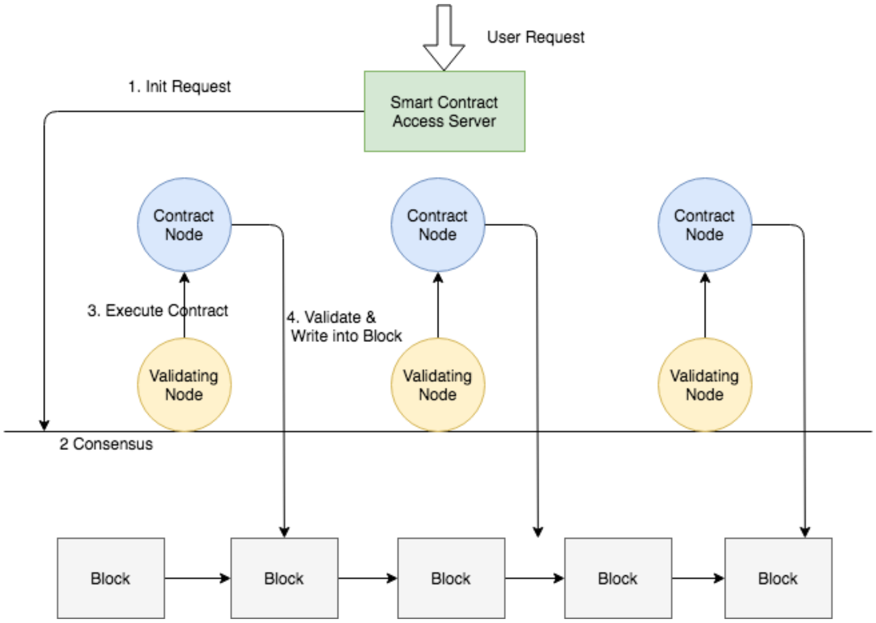
2. Fast transaction implementation

   A user init a fast transaction execution request, smart contract access server forwards the request directly to FSCTN, the node will execute the request, and return the transaction result to smart contract access server. At the same time, the server will record the transaction status, and store the transaction history since last confirmation. The transaction message with FSCNT could be asymmetrically encrypted or symmetrically encrypted in order to hide the transaction information except for the user and the current FSCNT. A user can query the smart contract access server for the transaction status and history.

3. Decentralized Confirmation of fast transactions

   User could query the unconfirmed transactions with a multi-signature confirm transaction, periodically (e.g., 10 min, 1 day, or 1 week), non-periodically (at a specific point fo time). The confirmation transaction is combined with unconfirmed smart contract transactions history, which changes the state since last confirmation and generates a state-exchange transaction. Smart contract access server sends the state-exchange transaction to validating node, then validating node will do the consensus with all the transactions, and then send smart contract transactions to correspond contract node. The finished execution result will be confirmed by the validating nodes, and written into the blocks with other transactions. Then the execution result will return to the user, realize Byzantine Fault Tolerance since the validating nodes achieve the consensus based on transaction history.

So, Jingtum Chain is divided into independent Contract layer and the underlying TX layer. Fast transactions are initiated in the Contract layer and are executed in it, their fast execution results are returned to the underlying layer, TX Layer, and then be validated and written to the blockchain periodically or non-periodically. Clearly, the transaction execution is not affected by the closing time of the block, the block size, and the distributed network communication. Jingtum Chain has the inherent advantages of distributed blockchain, overcomes the delay phenomenon existing in the transaction process and communication, consensus and contract execution process of the current blockchain transactions. As a result, it realizes rapid support for trading in near real time, maintains the Byzantine fault tolerance in the trading system, as well as hide and encrypt the transaction details, and maintain the consistency and integrity



of data of the distributed system.

2.9 Sharding

The fast smart contract transaction can be regarded as a particular case of sharding. From a more general concept, the selection of the execution nodes of smart contract is an implementation of sharding technique.

In addition to the fast transactions described above, if the information synchronization between multiple smart contract nodes is implemented by predefined protocols, i.e., BFT, then a BFT consensus can be reached among them. Of course, if this consensus is adopted, the processing efficiency of the smart contract will be reduced, but it is still a significant improvement compared with the usual situation that all nodes deal with a contract at the same.

3 Expectation

Jingtum Chain uses multi-token support and optimized smart contracts, it is a hierarchical, logically separated fast system, which provides a solid foundation for a variety of applications. The blockchain technology is still a young one, in a relatively early stage of technological development, but is expected to develop much more rapidly in the next few years, and various decentralized applications will be built on the Jingtum Chain. The Foundation will continue to work hard and maintain innovation for blockchain technology and contribute to the development of the whole blockchain community.