
FAIR
block

3rd GENERATION BLOCK CHAIN SYSTEM

Foreword

In 2008, Satoshi Nakamoto published a paper Bitcoin: A Peer-to-Peer Electronic Cash System, which created the notion of bitcoin and block chain. This system is “based on cryptology instead of credit, and enables both sides to make direct payment after they reach a consensus without interference of the third party intermediary agent”. The appearance of block chain does not only escalate the generation of “bitcoin”, a brand new currency system, but also brings a revolution of internet decentralization. Till now, block chain technology has undergone 10 years’ continuous iteration, is greatly beyond the preliminary definition of “a peer-to-peer electronic cash system’, and becomes a revolutionary internet underlying architecture. Block chain technology can construct a highly efficient and reliable value delivery system, and drive the internet to be an infrastructure to build social trust.

FairBlock devotes to conversing this vision into reality. Based on the newly constructed 3rd generation block chain system, we are trying to establish a de-centralized, free and stable brand new internet trust system, mainly solve the development pain point in pan-entertainment field in block chain application development which takes strong interactive application such as games as a classical representative, and promote the popularization and application of block chain technology to the general public.

FAIR Block 白皮书

CONTENTS

1, Development and Current Situation of Block Chain Technology

Bitcoin block chain was born in 2008 as the first block chain network in big scale and stable running. It is defined to be “a mechanism to enable the whole world to reach consensus for the same database content shared to the public”. After achieving the function of electronic currency, how to use block chain technology in other fields besides currency becomes the development direction of the 2nd generation block chain technology---- Ethereum initiated by Vitalik Buterin in 2014 is the most successful attempt.

The biggest innovation of Ethereum is it creates a block with perfect function and built-in Turing complete programming language (which can be deeply and free regulated). It allows customer to compile any complicated contract, autonomous agent and relation which are totally existed in block chain and delivered by block chain. The user of Ethereum can achieve any transaction type by the built-in scripting language code in the contract, including customized currency, financial derivatives, identity system or even more complicated decentralized application.

The most central feature of Ethereum is called “smart contract”. The simplest definition of smart contract is “it’s an encrypted box which includes value and can only be opened after meeting some specific conditions”. Since the block is embedded with Turing complete programming language and meanwhile has features like value-awareness, block chain awareness and multi-states etc., it’s hard for Ethereum to make use of smart contract to achieve applications unimaginable in the past.

Currently there’re hundreds of applications derived based on Ethereum, covering finance, IoT, virtual asset transaction, games, gambling and many other industries. Many operating modes were hard to be run or with a too high running cost, but decentralized autonomy makes it possible. Currently relatively well-known applications include:

The DAO. Founded with Ether capital, the DAO targets at constructing a new and decentralized business mode for enterprises and non-profit organizations. But it suffered hacker attack only 2 months after the project started. Above 3.6 million Ether were stolen. And finally Ethereum Foundation decided to handle this attack by Ethereum fork.

The Rudimental. It attracts independent artists to conduct crowdfunding creation.

FreeMyVunk, a virtual goods trading platform.

Ujo Music. It makes use of smart contract for music selling.

2, Issues in Block Chain

2.1, Performance Issue and Resources Waste Caused by Mining

Though Ethereum greatly expands the application of block chain by smart contract, the application scope is still restricted by several issues. The first one is performance issue caused by using Proof of Work (PoW) to establish consensus mechanism. Cryptocurrency takes the block chain as foundation, and the black chain itself needs to use Hash function to validate data to make it correct. To use PoW on cryptocurrency is a rational design. To be in details, the one who is the first to find computers allocated everywhere, and combines with the exhaustive predicted value of the data which should have been packed (nonce) shall gain the packing right (account charging right) of this block chain. After being found out, nonce will be packed to blocks together with data and Hash value to broadcast. After being confirmed and admitted by most of nodes, the packer can get the reward because of packing this block. Currently, cryptocurrency in the market including bitcoin and Ethereum etc. shall use PoW. In view of the performance growth brought by Moore's Law, in actual practice, it’s usually set up to increase the difficulty of nonce with the increase of computing power in competition, to maintain its reasonable operating speed.

As the current most widely used consensus mechanism, PoW architecture is simple and reliable, and is relatively fair to a great extent. More computing power invested to a node, more probability to get the account charging right. And the malicious hacker has to input computing power above half of the total power (51% attack) to guarantee to tamper the result. It makes hacker attack be in terribly high cost and hard to realize.

But the account charging right (probability to get reward) is proportional to computing power. With the prevalence and appreciation of cryptocurrency, all nodes will inevitably keep enhancing their computing power to earn more rewards. Bust most computing power is wasted in meaningless Hash computation, which leads to great computing power and energy waste. There’s statistics showing that currently the electric energy spent in account

charging right (also named mining) has exceeded the total amount of that in a small country. It's pitiful that though large amount of energy sources are consumed, due to the inherent low efficiency, the block chain network performance based on PoW is far to meet the request. Take the most promising Ethereum as an example, current 25 transactions can be handled within a second in the entire Ethereum network. This low performance fails to meet the requirement of modern internet application.

2.2, High Transaction Cost

The process of fighting for accounting charging right is essentially a competition of computing power, which will inevitably cause the surge of hardware cost of nodes. The transaction fee involved in Ethereum block chain will be finally calculated with Ether. There shall be a certain amount of "Gas" for each transaction (i.e. to assign the value of startgas), and fees necessary to pay for each unit Gas (i.e. gas price). Before transaction execution, Ether at the value of startgas*gasprice will be deducted from the account of transaction initiator. All operations during transaction execution, including database reading and writing, message sending, and calculation in each step, will consume a certain amount of Gas. Though in the design of Ethereum, the amount of Gas needed to pay for each transaction is fixed, the fees of Gas is still be designed dynamically by users. Meanwhile, higher transaction fee to be paid, more active for nodes to pack and handle this transaction. Currently, to conduct the simplest Ether transfer transaction, it needs to pay 0.1 Ether as the transaction fee. This makes small volume of Ether transaction impossible. In Ether design, each step of a complex contract transaction needs to consume a certain amount of Gas. And even in executing some complicated smart contract, the transaction fee to be paid has already exceeded the contract itself.

2.3, Contradiction of Side Chain Interaction and Isolation

Another urgent issue needing a prompt solution is side chain. Side chain is a fork generated based on the main block chain. Generally it's to realize some specific purposes or functions. Before the generation of Ethereum, this fork usually conducts by means of hard fork. Take Bitcoin as an example, the familiar "bitcoin" usually refers to Bitcoin Core (BTC), which has been running for about 10 years steadily since the foundation by the legendary person Dorian S. Nakamoto. BTC can be said to be the No.1 in digital currency field. Now its market value exceeds 170 billion USD. But technically speaking, comparing

with other emerging digital currencies, BTC is not competitive. The block size is just 1MB, and the transaction delay and transaction fee will be higher and higher.

In terms of problems of BTC, some teams including BitcoinABC, Bitcoin Unlimited and BitcoinXT etc. work together to develop Bitcoin Cash (also named BCH or BCC). BCH was found in Aug 1, 2017. It forked when bitcoin block chain was 478,559 long, and it now becomes the first hard fork currency of BTC. BCH brings various upgrading features, including big block with 8M capacity, supporting two-way replay protection etc. On Nov 13, 2017, because the emergency difficulty adjustment mechanism (EDA) initially adopted was unstable (sometimes the BCH mining difficulty will be adjusted to above 3 times of BTC), BCH will conduct hard fork again, and increase DAA difficulty adjustment mechanism.

Besides, various hard fork versions of BTC such as Bitcoin Gold（BTG）and Bitcoin Diamond（BCD）etc. have been generated. The hard fork of BTC is essentially a set of new block chain (side chain) derived by upgrading or adjusting codes, based on a time node of bitcoin original block chain network (main chain). Though the new block chain is homologous as the original block chain, actually it's totally independent from main chain. And the side chain cannot communicate with the main chain. Meanwhile, the side chain needs to establish its independent nodes (miner) network. As to nodes, though multiple nodes programs of different block chains can be run in a server equipment, a node program can only serve for one block chain.

Ethereum makes use of the feature of smart contract to design ERC 20 token specification. Different from BTC fork, tokens issued based on Ether are essentially a part of smart contract running in main block chain. "Token side chain" runs in smart contract in a virtual state. It's different from the fork of BTC, which generally means a new "coin" is born with independent block chain and node network. The derived "token side chain" of Ethereum in the form of smart contract is more used to issue tokens for some specific functions or service. But tokens issued in ERC20 standard can be compatible immediately for Ethereum wallet and exchange. Some famous tokens including ETH, EOS, XUC, OMG and ITC etc. are issued based on Ethereum.

As to simple token issuing, to use smart contract on Ethereum is a simple and highly efficient good idea. But for some more complicated applications, to run in the form of smart

contract in main block chain will cause a serious safety loophole. Due to the “side chain” in the form of smart contract and main block chain are not isolated compulsorily as BTC side chain, the defect in side chain design may directly affect the main chain. The Dao incident which brought a heavy loss to Ethereum is a typical example.

DAO refers to the decentralized autonomous organization. It’s to compile codes for organization rule and decision-making organizations, to remove the needs of written documents and reduce management staff, thus to create a decentralized management architecture. DAO usually needs a group of people to compile running specification (smart contract). It starts the financing after establishing the contract. In this phase people start to invest more capital to purchase tokens, to represent their ownership (this process is called crowd selling or initial coin offering (ICO)). After finishing financing, DAO will start to run. And people will vote for the development decision of DAO according to their quantity of ownership.

However, as one of the most successful crowdfunding projects in the history, The DAP Smart Contract has a mortal loophole, which caused the biggest robbery in the history. The hacker made use of the loophole to steal 30% Ether of the DAO project, worth 55 million USD in market value at that time. The DAO incident also revealed the big problem in Ethereum smart contract system. Codes which run in Ethereum main block chain in the form of smart contract cannot be separated physically from the main chain. Once there’s a serious BUG, it may affect the network safety of the whole block chain.

Besides safety factor, the derived side chain with smart contract is essentially a program running in Ethereum, which will inevitably further increase the volume and complexity of block chain. It makes the situation worse for the whole Ethereum nodes network.

2.4, Difficulty to Interact with Real World

In Ethereum, there’re two entities to initiate and receive transactions: users and smart contract. Smart contract can be regarded as an automatic agent in Ethereum network. It has Ethereum address and account sum, and can send and accept transactions. When there’s someone sending transactions to the contract, it will be activated and starts to run its own programs, such as to change its internal status or send some transactions.

The code of Ethereum smart contract, also called “Ethereum virtual machine code (EVM code), is compiled by low-level bytecode language based on stack. The code constitutes a series of bytes. Each byte represents a kind of operation. Generally codes will be in infinite loop. Once the program counter increases 1, there’ll execute 1 operation, unless all codes have been executed or there’s error, STOP or Return command. The operation can only visit 3 kinds of spaces for data storage: stack, internal storage or long term storage of contracts.

The biggest problem of users in smart contract design is that codes running in EVM cannot visit and call data outside of the block chain network. For example, financial derivatives is the most common application in smart contract, and it’s an application most easily to be realized with codes. In actual process, the main challenge is that most derivatives contracts need to be combined with a contract specially used for data release. But this needs to rely on some special agencies to regularly maintain and update data, and provide an interface to allow other contracts to send query message to gain key financial data.

3, Core Technology of FairBlock

3.1, Consensus Mechanism

Proof of work and Proof of stake are two most important concepts in block chain technology, and the core of block chain. Block chain in itself is a distributed ledger, so it inevitably has two issues below:

A) How to build the concept of time sequence in decentralized network?

B) When there are multiple nodes to finish record transaction, whose record shall be adopted?

The first and second generations of block chain usually use POW to solve the consensus problem in distributed system. PoW (Proof of Work) is a consensus mechanism based on computing power pricing. Miners create a block by solving a complex but meaningless mathematical problem, and gain a certain amount of coins as rewards. Each miner’s ability to solve problem totally depends on their own computing power. To earn rewards, miners will compete with each other, keep upgrading their computing power, and

waste numerous resources and energies. It will lead to higher and higher transaction fee, but it's not beneficial to improve transaction speed. Besides, the coin-holder cannot participate in any decision making. The decision making power is centralized in several mining pools, which goes against the decentralization philosophy.

As a symbol for the 3rd generation block chain, PoS (Proof of Stake) is a consensus mechanism based on on-chain currency pricing. PoS uses cash holding to replace computing power. It enables cash holder to more engage in mining process, and don't need to calculate complicated math problem, thus to avoid resource and energy waste. There are mainly 4 PoS solutions: PoS based on Byzantine fault tolerance, PoS based on chain, PoW/PoS mixture and PoS (DPoS) based on authorization. PoS based on Byzantine fault tolerance is with a relatively low fault tolerance, and fault nodes and malignity nodes shall not exceed 1/3 of the total number of miners. And validators' quantity is restricted to achieve a relatively short confirmation time. PoS based on chain is essentially a currency pricing adaptation of PoW. PoW/PoS mixture is just a transitional scheme, and it will finally be replaced by a pure PoS mechanism. PoS based on authorization reaches consensus by selecting agent, and sacrifices the concept of decentralization, which is not suitable for public chain. After studying existing PoW and PoS mechanism, FairBlock puts forward a new PoS scheme: PoS (cPoS) based on competition.

3.1.1, cPOS

FairBlock generates and allocates 2.1 billion FBC in Genesis Block. Blocks creation after it is to be finished by Forging Committee. To solve the common problem in PoS mechanism that the rich becomes richer, except for genesis block, creation process in other blocks will not generate new coins. All revenue come from transaction fees.

3.1.2, Forging Committee

Forging Committee is a smart contract, including several committee member nodes which have forging right. Each node has the chance to create blocks. To stimulate forging, all transaction fee will be gained after a block is successfully forged.

Every node can apply to join the forging committee, but at least needs to pay 1 FBC as cash deposit. If the forging person does evil on purpose, the cash deposit will be confiscated. Meanwhile, we have special mechanism to prevent nodes from trying to apply

to join the committee at a sum of cash deposit lower than 1 FBC. The responsibility of the committee is to create new blocks. If a forging committee member fails to perform forging obligations for continuously 3 times and is expelled from the committee, the cash deposit will be withheld for a certain time. To withhold the cash deposit is a kind of punishment mechanism, to punish forging committee members who fail to fulfill responsibilities. It's more harmful and with harsher punishment not to create blocks than dropping out from the committee.

3.1.3, Forging Group and Main Forging Committee Member

The voting right of forging committee member is related to the sum of cash deposit. A newly joined committee member will not get the voting right immediately, and it needs to wait till there're 100,000 blocks. With the increase of blocks' height, the voting right will be continuously accumulated. If a forging committee member successfully adds the block to block chain, the voting right will be reset to be 0. Since everyone can check the current voting right for each forging committee member, the member will be divided into different groups according to the last two digits of the address. The member who gets the highest voting right will be elected to be the main forging committee member. The successive forging group tends to validate and recognize the block forged by main forging committee.

3.1.4, Reward and Punishment

The reward of forging committee member comprises of two parts: 1) Forging committee member will get all transaction fee in the block he newly creates; 2) All cash deposit of the evil address can be gained after the member reports the forging member who does evil. Due to little resources are consumed in the process of creating blocks in cPOS system, the member can get considerable profit even though there's just transaction fee as reward. In such a situation, the rich will not be richer because of the extra rewards caused by bitcoin or Ether.

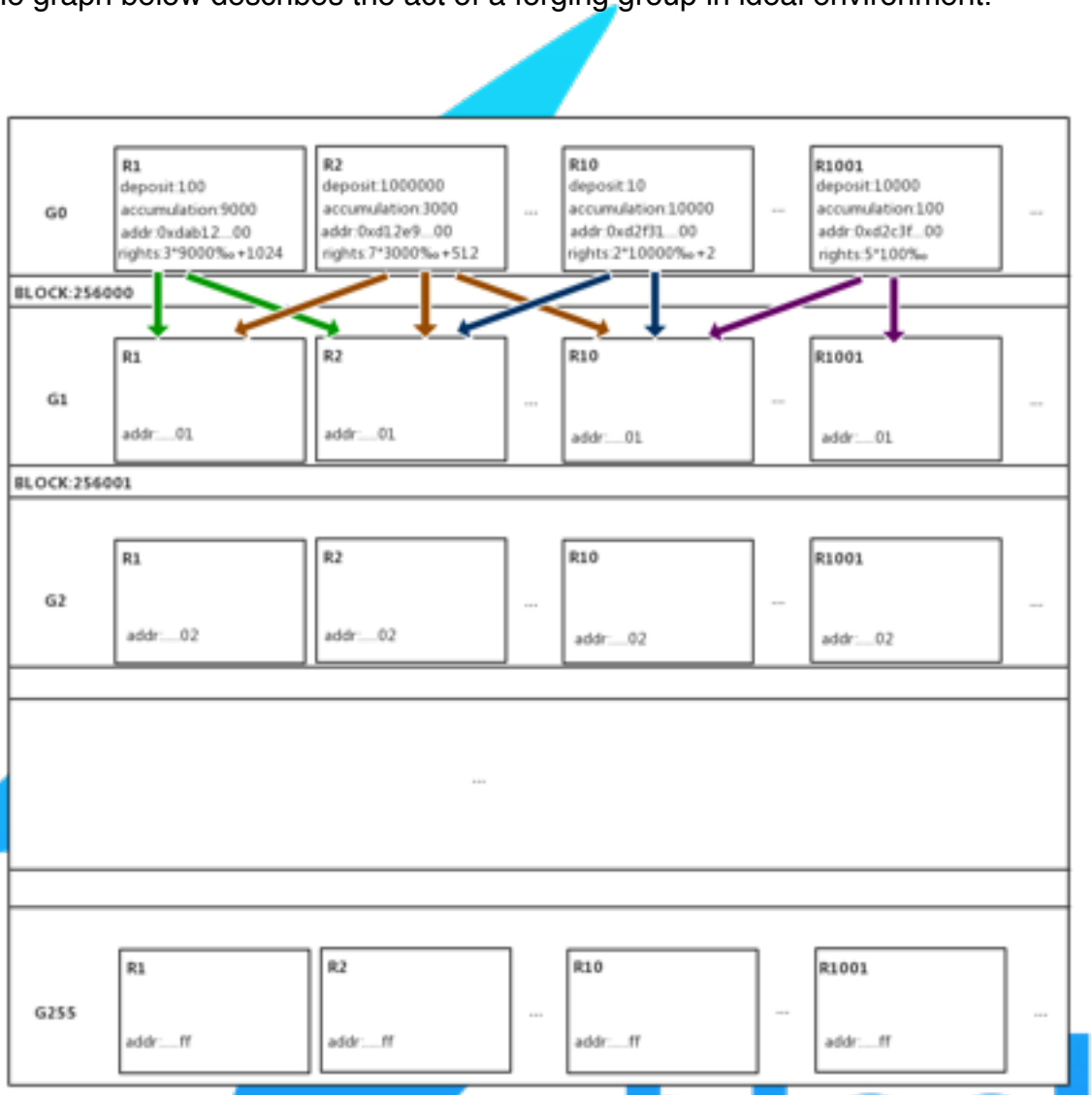
3.1.5, Consensus Building and Forging Process

No matter which kind of forks appearing in the chain, it will be the correct main chain if it's with the highest voting right. Because the main forger has extremely high voting right, it will reach a consensus within an extremely short block length to remove the fork rapidly.

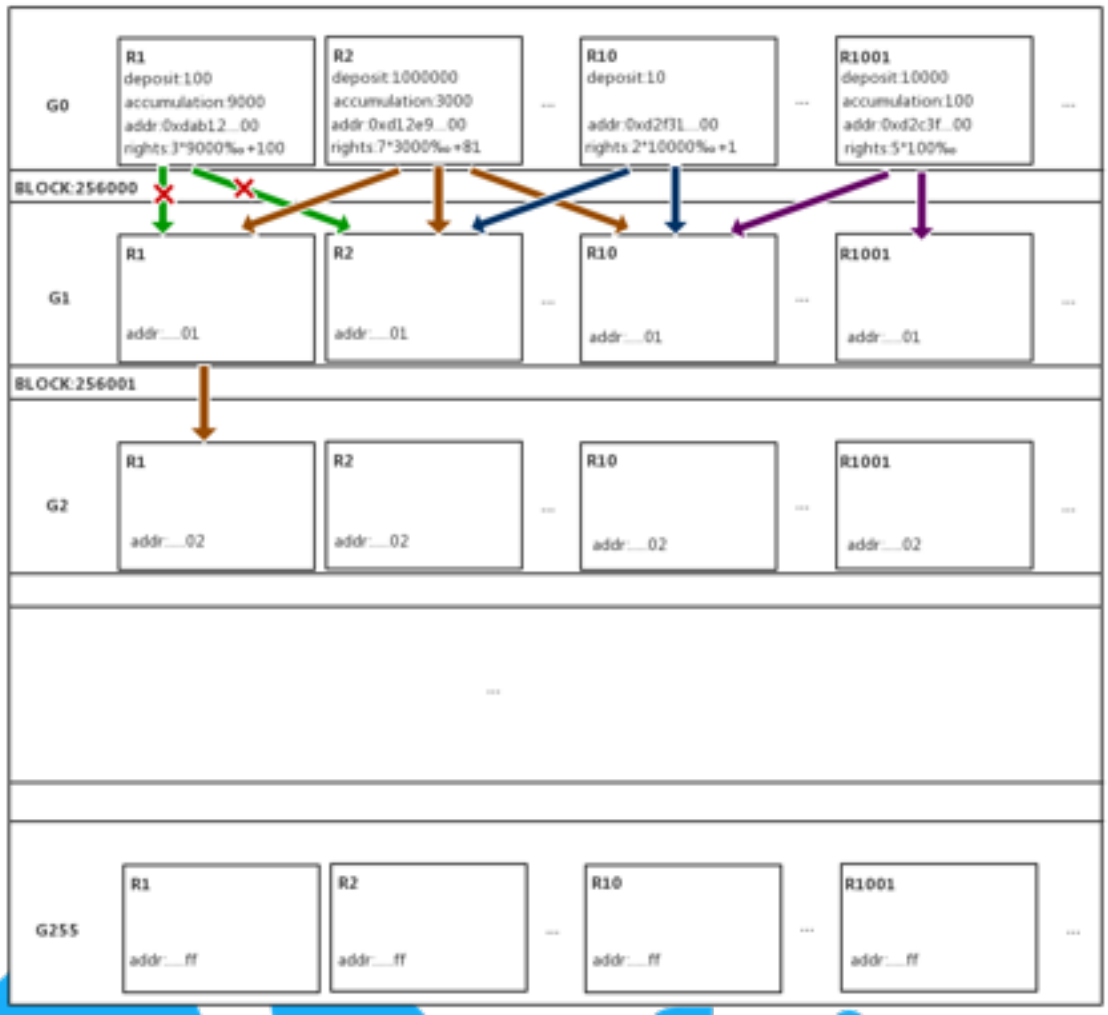
To facilitate understanding, here we just describe the forging process and selection

strategy of the top 2 highest voting right forgers. The actual situation may be more complicated, but it's with the same principle. Rn refers to the forging committee member whose voting right ranks No.1 in the group.

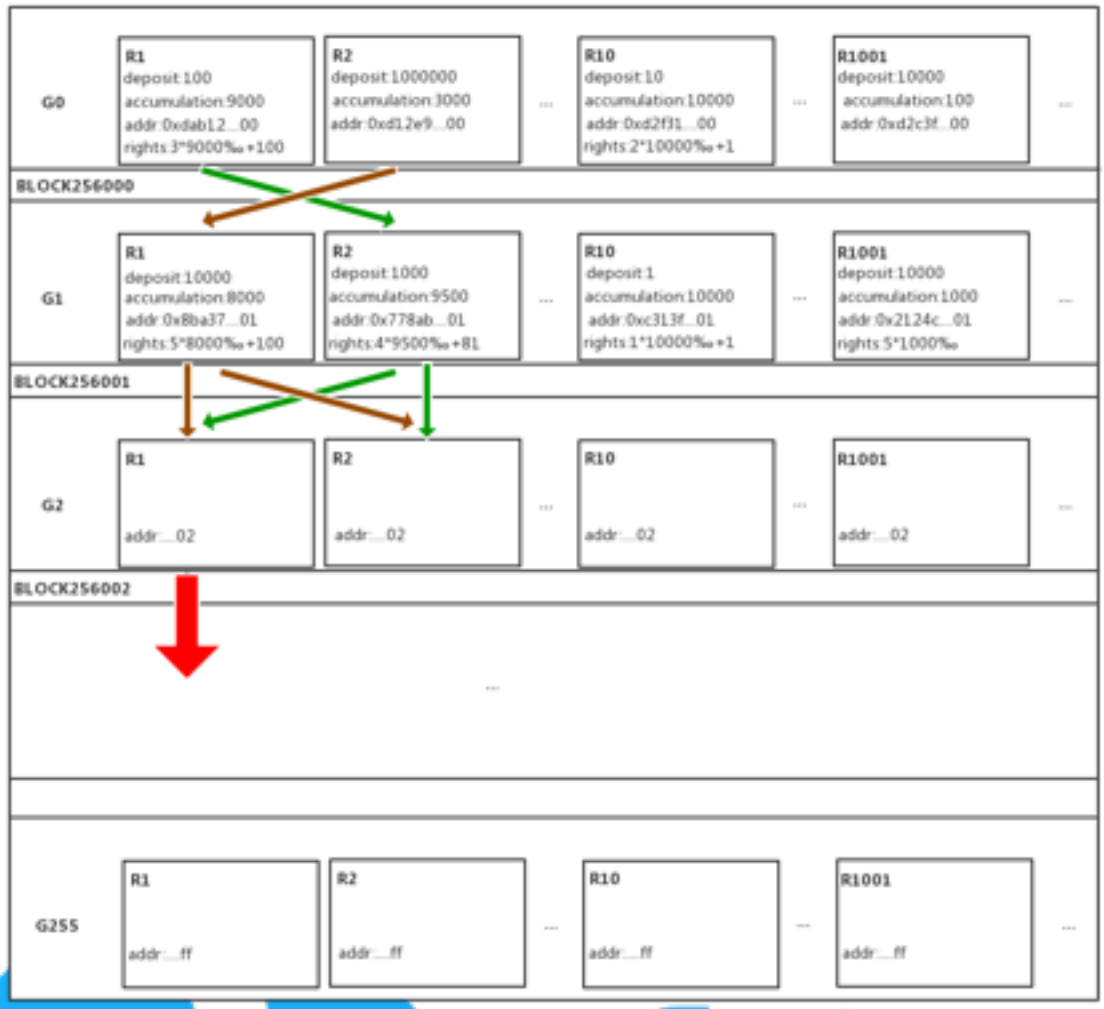
The graph below describes the act of a forging group in ideal environment.



Network environment in actual world is extremely complicated. Broadcast sent by the voting committee member with the highest voting right may not be accepted by the next forging group. This graph shows the alternative scheme in such a case.



The total voting right relates to grouping, cash deposit, accumulated block height, ranked voting right and address, so it's hard for forging committee members to collude to cheat. But network issue or other unknown reasons cannot be excluded. The forger with the highest voting right accepts blocks created by the forger of the second highest voting right in the last block, as shown in the graph.



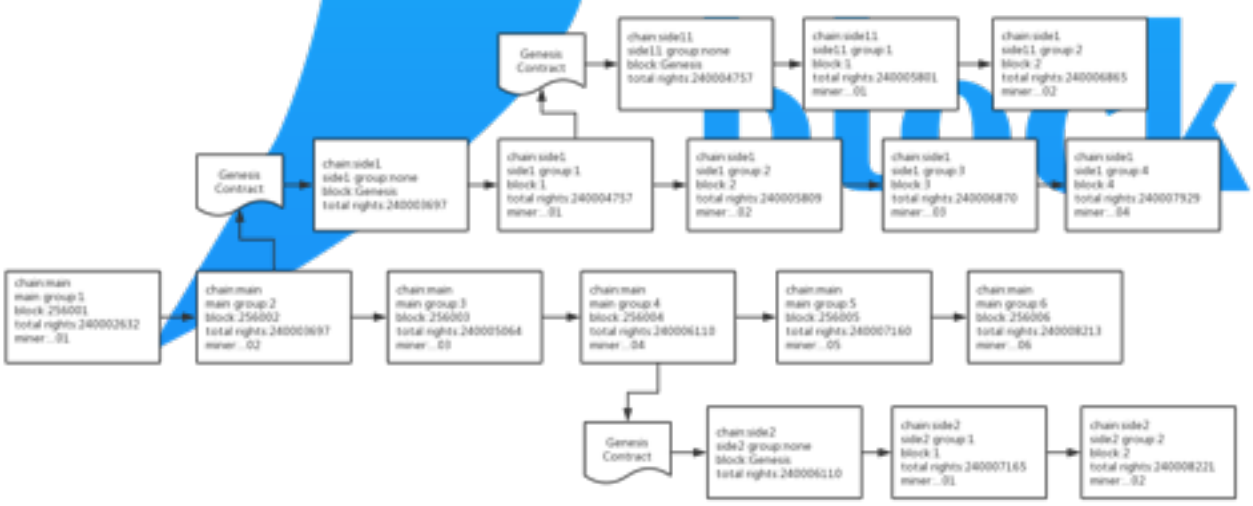
Based on cPoS mechanism, the fork can always be removed within a relatively short block height, as shown in the graph. There're forks at block height 256,000 and 256,001. And the forger with the highest voting right in G2 group chooses one chain in it. The total voting right of the chain becomes remarkably greater and has a relatively high probability to win. Forger in G3 group will continue to create blocks based on this chain.

3.2, Super Side Chain

The code of super side chain is same as that of main chain. They use the same consensus algorithm (cPOS). It has its own independent block chain. Thanks to PoS consensus algorithm, the super side chain can directly use part of nodes network in the main chain, and can also have its own deployed node network. Traditional PoW algorithm terribly consumes performance, and miner node will not dig two different block chains in the same node.

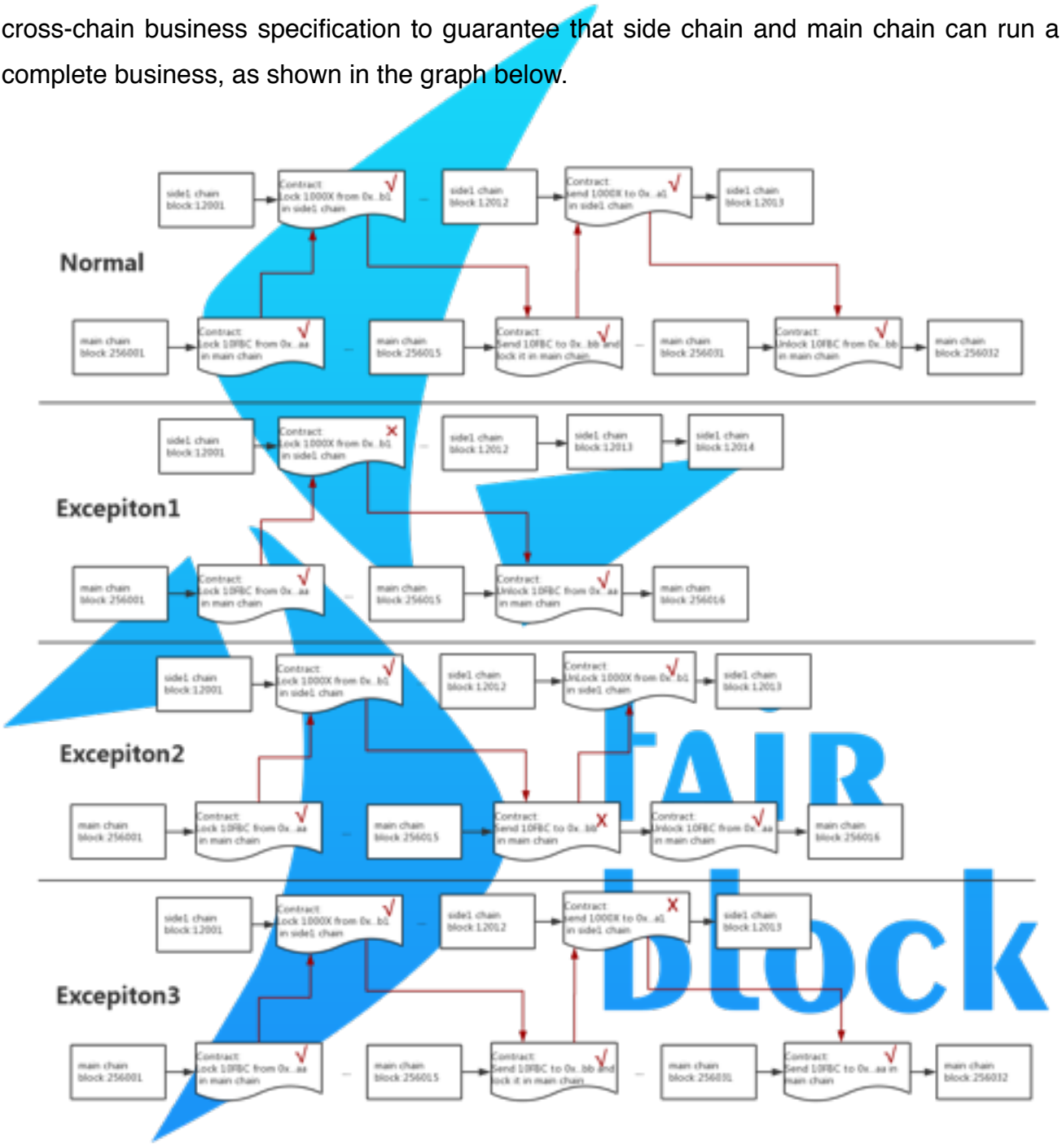
3.2.1, Side Chain Tree

Super side chain supports to take designated side chain as the main chain, and continue to fork to new side chains. And the like. Then it forms a side chain tree, as shown in the graph below.



3.2.2, Cross-Chain Business

In each chain, either smart contract or transaction can be a complete business. Due to side chain and main chain are in different node network, each block chain has different generator, speed and rule, and cannot finish a complete business. Thus we formulate a cross-chain business specification to guarantee that side chain and main chain can run a complete business, as shown in the graph below.



3.2.3, Independence of Side Chains

The advantage of side chain architecture is data independence, no extra communication and storage burden to main chain, no excessive data expansion or

broadcasting, and no code bug spreading to other chains.

Independence is both an advantage and a shortcoming. After the developer completes the development of a new digital currency, he shall also consider operation, i.e. to find enough nodes to run it, or no enough safety can be maintained. But comparing with Ethereum, it's a shortcoming. Once there's a token released in Ethereum, all nodes shall work for it, and provide safety guarantee for it.

3.2.4, Flexibility of Side Chain

But side chain architecture still has the necessity to exist, because not all applications need that high security. Side chains provide users with an optional flexibility. For example, suppose there are 1,000 nodes in main chain, some side chains are important which need 500 nodes to run, while some side chains are not that important, which may just need 100 nodes. This is totally decided by users and developers. But in Ethereum, we can just choose one, i.e. all nodes will run it. It's said that Ethereum plans to achieve a piece-to-piece mechanism. Actually this is already solved in side chain system.

The flexibility of side chain also represents in that parameter of all block chains can be customized, such as some simple parameters like block interval, block rewards, the use of transaction fee etc. But the most important is business logic. Side chain can easily develop transaction type related to your business or smart contract, or even can release its own side chains.

3.2.5, Mutually Beneficial Relation between Main Chain and Side Chain

Main chain and side chain can be beneficial to each other. Main chain provides infrastructure to side chain, while side chain can supplement more nodes and developers to main chain to expand the whole system. The developer of side chain does not need to provide assets. The existing main chain nodes can be used. And only the nodes owner shall have this currency. Besides, main chain and side chain can make use of the cross-chain business specification provided by system to conduct value exchange. It equals to provide a medium of value to assets in side chain. The developer does not need to consider transaction platform.

3.2.6, Productivity Problem Solving

Only after a smart contract has been compiled per side chain publication standard can side chain get automatic release. The developer only needs to care about detailed business logic. This is equivalent to the token development difficulty of Ethereum. And smart contract inside the side chain also uses programming language same to main chain. And it can call the code library of smart contract in main chain, and share the same developer community.

3.2.7, Node Security Problem

Security problem here is different from the problem caused by lack of nodes or code bug aforesaid. Instead, it refers to damage to nodes caused by side chain codes. We hope each miner node can trust and accept any side chain, and the node owner does not need to trust the developer of side chain. This needs to provide a measure of security precaution, such as to prevent side chain code from reading file system, and excessive consuming network, memory or CPU operation. In our system, side chain code will use java script virtual machine which is isolated by sandbox mechanism, then we are to implant a customized require and some regular and safe modules for this virtual machine. And finally to load the code of side chain. We also make use of container technology to manage network, memory or CPU consumption of each side chain, then the side chain installer will have no risk.

3.3, Security Function

Security function is a specific smart contract according to security function standard in the system. This smart contract can be executed in smart contract virtual machine, then to communicate with external server, including capturing data or finishing a complete business. The sample code is shown as below:

```
1.  //a committee for manage security function
2.  contract SecurityCommittee {
3.    //the struct for voter
4.    struct Voter {
5.     bool agreement;
6.     address voter;
7.    }
8.    //the struct for security function
9.    struct SecurityFunc {
10.    address applicant;
11.    address securityFunc;
12.    uint32 publickey;
13.    mapping(address => Voter) voters;
14.   }
15.   //the return value of call security function
16.   struct SecurityResult{
```

```
    uint32 sign;
    address securityFunc;
    any param;
    any result;
  }
  //global store for all security function
  SecurityFunc[] public securityFuncs;
  // submit a security function, only can be used after the application is pass
  function submit(SecurityFunc func){
      securityFuncs.push(func)
  }
  // vote for a security function, need 2/3 voter's agreement
  function vote(address securityFunc, bool agreement) {

  }
  // check if a security function can be called
  function isPass(address securityFunc) constant returns (bool){

  }
  // calculate the digest by sign and public key from security function
  function calDigest(address securityFunc , uint32 sign) returns(uint32){

  }
  //calculate the Digest of result with param
  function calHash(SecurityResult result) returns(uint32){

  }
  //check if the hash is just the same
  function isSameHash(uint32 src, uint32 dst) returns(bool){

  }
  //call the security function by address with params
  function callSecurity(address securityFunc, any params) returns (SecurityResult){
    if(!this.isPass(securityFunc))
      throw;
    return securityFunc.call(params)
  }
  //check if the result value is generated by the
  function checkSecurityResult(SecurityResult result) returns (bool){
    uint32 digest = this.calDigest(result.securityFunc, result.publickey)
    uint32 hash = this.calHash(result.result)
    return this.isSameHash(digest, hash)
  }
}
```

3.3.1, Closure or Expansion

Standard smart contract is essentially a part of codes running in virtual machine. All data operated by codes are stored in the chain. Under such an architecture, smart contract code cannot call and operate data outside the block chain. This results in that traditional smart contract cannot interact with data outside of block chain, such as failing to conduct DNS query or get security nonce.

In FairBlock block chain architecture, we provide an external function call customized by the user. Codes running in FairBlock chain can make use of security function to call external data, which will greatly expand application scope of smart contract.

3.3.2, How to Define Security

Smart contract is executed by block chain nodes in forging area, which is also called miner. If a security function is not regarded to be secure, the miner will reject to execute the smart contract including this security function. Therefore, we need to introduce security consensus.

Firstly, to provide smart contract of security function needs to pay cash deposit.

Secondly, forging committee will vote for this smart contract. If the majority of miners cast vote of trust, then the security function will be trusted and be called.

Lastly, vote of no trust can be done at any time. If it exceeds half, cash deposit will be confiscated.

Security function shall try best to persuade miners to believe their security, such as providing source code address and authoritative evidentiary materials.

3.3.3, Execution and Validation

The execution and validation transaction of traditional smart contract virtual machine use the same logic code. Security function introduces external data, which may lead to inconsistent call result in validation. (For example, the security function is a generated nonce, but it gets different call results each time).

Therefore, FairBloc virtual machine designs the special block transaction standard. We separate execution and validation to make them be handled by virtual machine separately, to guarantee that all nodes can validate the smart contract including security function.

3.3.4, Distributed Transaction

In a complicated and strictly consistent application scenario, such as multi-party transactions, a simple security function call cannot meet such kind of application. Therefore, we have defined security function standard of distributed transaction, to guarantee that the block chain can conduct secure and consistent data exchange with external data.

3.3.5, Equity

It produces cost to provide an external server of security function, so the smart contract

to provide security function can regulate transaction fee of each call. After getting the revenue, the service provider of security function can be more motivated to provide greater bandwidth, more computing power, and can attract more people to provide security function, thus to bring in competition and reduce cost.

4, On-Chain Application Solution

As a revolutionary internet underlying architecture, functions to be realized by FairBlock are far more than digital currency or electronic contracts. Many daily work, life or even entertainment or game applications can present in block chain in a new form. It's a set of decentralized cooperative system. Thanks to revolutionary technology brought by technical team, we are capable to reconstruct most of internet application in decentralized manner, to make internet safer, more reliable and more imaginable.

4.1, Applications with High Performance Requirement

Modern internet application has an increasingly higher requirement for response speed and server processing capability. Take Facebook, the well-known and biggest social network in the world as an example. It is the originator of social network. Till July 2017, its monthly active user has broken through 2 billion. In every 5 webpage visits, there's 1 for Facebook. Users post above 5 billion messages in Facebook every day, and click "like" for more than 4.5 billion times. Currently there're above 300 billion pictures saved on Facebook, and monthly pic storage capacity increases about 10PB (note, unit conversion: 1PB=1024TB). Another application which has an extremely high requirement is online game. Since most behavior of players in the game needs interaction and information confirmation with the server, in the process of traditional game operation, there will be divided to many independent servers to reduce the pressure of server. If game application is totally running in block chain network, it will be a great challenge to data handling capability in the whole network.

In view of such a huge amount of data, even to integrate all existing block chain network cannot meet the demand. Even the number of block chain users increases by several times, but due to the limit of low efficiency in PoW design, the expansion of block chain network cannot make processing efficiency remarkably increase. Instead, the new increased computing resources are mostly wasted in validation for computing power

competition and to avoid unexpected fork.

On the contrary, FairBlock totally based on cPoS technology can greatly improve the processing efficiency of block chain. In small scale test network, we have already realized 2,000 times/second processing efficiency. Comparing with the 25 times/second low efficiency of Ethereum network, this is a giant leap. And since nodes in cPoS block chain do not need computing power to compete for account charging right, performance of the whole FairBlock network can be further improved with the expansion of network scope, and the new increased computing performance doesn't need to be wasted in meaningless Hash test. In the high performance block chain network of FairBlock, the developer can realize application modes which are hard to be supported by many traditional block chains, such as game, lottery, social contact and sharing, and light blog etc.

The other important application of block chain is IM （instant messaging）. In Ethereum block chain network, even performance cannot cause bottleneck, high transaction fee also hinders the actual practice of such kind of application. You can imagine that every information sent to friends needs to be charged with a high cost. Let alone limited by PoW design, the main income of each node comes from rewards after successfully fighting for account charging right (mining). And the transaction fee paid for each transaction is just a drop in the bucket comparing with mining reward. In this mode, huge cost needs to be spent on each node to improve computing performance to improve the success rate of mining. Transaction (account charging) itself becomes a thing that nodes are unwilling to do. It can be forecasted that the transaction fee can be further improved in the future.

In Fairblock network, thanks to optimized PoS mechanism, mining will be a history. Since not to compete for computing performance, any individual computer can become a node to support Fairblock block chain network, and revenue of nodes will all come from transaction fee. Then the node decentralization problem existed in PoW network is readily solved, and node full competition will bring lower transaction fee. This enables developer to construct totally decentralized piece-to-piece or group instant messaging application in FairBlock. Besides, FairBlock can also pay for small amount of tokens. Applications in small amount but high transaction frequency such as online game provide low cost and decentralized block chain solution.

4.2, Applications to Interact with the World Outside of Chain

Take Ethereum for example, block chain “application”, in the traditional sense, generally refers to a certain kind of smart contract procedure to conduct simple conditional judgement and then automatically handle transactions. The problem lies in that to use Ethereum smart contract to use Turing complete EVM language to compile can theoretically realize advanced applications far more complicated. But actually, this kind of application mostly stays in theory but has not been actually applied. This is surely affected by the low performance and high cost in Ethereum itself, but the most important reason is that Ethereum smart contract cannot visit data outside of block chain.

Block chain application in Ethereum usually can just depend on manually maintained data to release contract to achieve and gain external data. But manual intervention goes against the original intent of “de-artificializing”, and may lead to risk of error. But the use of security function can easily implement this kind of application.

In FairBlock block chain, developers can use original security function to solve the problem. A set of safe and anti-tampering security function system can be deployed in block chain. Developers can enable smart contract application to gain data outside the block chain by gateway automation. This process is bi-directional. Smart contract can also send data to addresses outside the block chain by security functions.

Based on this feature, developers can run more advanced smart contract application in FairBlock block chain. Take social application aforesaid as an example, developers will realize the interface of content sharing, to enable users to more conveniently and directly post pictures and articles etc. in other platforms to social network in block chain. And applications on the chain of Fairblock support connecting to 3rd party payment, e.g. various entertainment, game and lottery applications etc. can easily be commercialized. But as to application interface with highly sensitive security such as 3rd party payment, developers can regulate the details and restriction of payment function in smart contract, such as payment frequency, whether 2nd time validation or payment limit set up is necessary or not.

4.3, Development and Application in Side Chain

The complexity of a block chain network depends on the quantity of asset and application running in the chain. As to some block chain applications which are complicated

and not frequently interacting with block chain resources in core area (core function such as currency and authentication), to run directly in main chain in the form of smart contract is not a good choice. Therefore, FairBlock provides the super side chain to realize this function. In fact, we encourage developers to create their own applications in the chain in the form of super side chain, which is independent from main chain in basic level, but provides general interface for side chain and main chain to communicate. The side chain can directly call functions and data in main chain, and can also operate mutually with main chain. Different from hard fork side chain of bitcoin, after the developer establish super side chain, he does not need to establish new nodes network. The original FairBlock nodes will provide service automatically to the super side chain derived from FairBlock.

Since super side chain supports tree-like and multi-layer side chain technology, developers can derive side chain again. Take social network as an example, developers can run the core architecture of social network in the first layer of side chain, such as account and user information etc. But other functions such as chatting and light block etc. or other social network application are running in the 2nd or 3rd layer of side chain. This architecture can enable developers to conveniently build and manage complicated block chain application.

Besides, super side chain can provide developers with safe and isolable application development and test environment. Problems and BUG in side chain will not affect main chain. Catastrophic accident like Ethereum “The Dao” will be strictly isolated in side chain in FairBlock. And if complicated application runs in independent super side chain, it can not only improve the execution efficiency of improving application itself, but also greatly reduce the bloated degree of main chain. Another application of independent side chain is that developers can issue their own digital currency based on FairBlock main chain. And By interaction of super side chain, smart contract running in main chain can make the exchange totally based on applications in the chain.

4.4, Time Block Chain Application

FairBlock can fork to super side chain which just keeps a certain length. Traditional block chain keeps all blocks since the beginning, but FairBlock side chain can support blocks which only keep a certain time length. This can effectively reduce the length of block chain, and equipment’s operation and storage pressure. This enables FairBlock to be

deployed to most equipment with relatively low performance.

Besides, take traditional game application as an example, to normal operation of most users, server generally just needs to keep several hours to several days for backtracking. Except for some core contents, most data don't need to keep too long a time. But if developer hopes to develop games based on block chain, since traditional block chain must save all data, it will make block chain be occupied by numerous useless data. But by use of time block chain technology, developers can choose data which just keep a certain length of block chain. Performance and communication speed of applications like game application or instant messaging can be obviously improved.

4.5, Applications Needing Security Nonce

It's well known that the modern computer cannot automatically generate real "nonce". The common alternative solution is to gain nonce by special "nonce supplier", but this kind of service provider generally uses some natural but random things in the nature to help generating nonce. But limited by the closure of Ethereum virtual machine, smart contract can only generate pseudo nonce. This will bring great potential safety hazard in some applications which have extremely high demand on true nonce.

Take online game application as an example, game is built on nonce. In a game, circumstances like equipment dropping and treasure box opening etc. can all use nonce to control the probability of players to get goods. If to use pseudo nonce, it will make probabilistic events in the game become predictable. In another application urgently needing security nonce such as online gaming based on block chain network, all gambling games are based on probability in essence. Comparing with other complicated games, gambling game is simple, which is realized by codes. Meanwhile, gambling industry has an extremely high demand on anonymity, security and anti-cheating ability. Block chain technology can be regarded as one of the most promising development directions for gambling industry. Though currently lots of on-chain gambling applications appear in Ethereum, limited by the defect that Ethereum itself cannot use security nonce, currently existing gambling applications can just finish digital currency exchange by block chain. But the core probability operation of gambling is still completed in server outside of block chain. This mode is non-transparent and cannot be controlled by the banker in the gambling game. But once the security nonce provided externally is introduced to the application,

gambling application can fully run in block chain, which represents the transparent and fair principle of block chain.

4.6, On-Chain Entertainment Era

Block chain application is generally regarded to be suitable for some serious fields, such as financial industry, business and internet basic service. But with the progress of block chain technology and enhancement of computer performance, more and more developers try to design various entertainment applications in block chain. Take the current popular Crypto Kitties, a pet collection and education simulation game based on Ethereum for example, users can spend Ether to get a “cat” randomly, and meanwhile consumer resources to raise the cat. Users can freely trade their cats with each other, and cats’ value relates to its rarity. In this game, “cat” in essence is a kind of digital asset in Ether pricing in Ethereum. To get and exchange “cat” is conducted in Ethereum network in the form of smart contract. In a certain sense, this is a kind of cat-coin derived from Ether. But this game nearly represents the extremity to develop games in traditional block chain. Some “block chain games” in the market look more complicated, but in essence they are just built in a certain transaction function of digital currency. Its core logic of game code is still running in central server in the traditional mode, but it cannot really protect the interest of players and completely get rid of the control of game makers.

But in FairBlock block chain system, thanks to the unique super side chain and security function etc., we can provide a full set of API with perfect functions to enable various entertainment applications to completely connect to FairBlock block chain, and enjoy safe and decentralized new entertainment experience. Players who value game fairness will welcome and even just support block chain games.

4.7, Brand New Block Chain Ecology

In the past, in a stably running block chain project, each participant only had two ways to gain revenue: “mining” and charging transaction fee. “Mining” is based on inefficient PoW algorithm. It not only restricts the performance of whole block chain, but also wastes large amount of computing power and electric power resources. And taking “mining” revenue as priority and transaction fee as supplement leads to high transaction fee. To regulate the ecosystem, extremely high transaction fee like bitcoin or Ether is not desirable.

FairBlock totally gives up the “mining” mode like PoW algorithm in design. And meanwhile, cPoS brings extremely low performance requirement and brand new competition mode for nodes. With the full competition and further expansion of nodes network, FairBlock users can enjoy quite a low transaction fee.

At the same time, we also bring brand new business ecosystem to block chain network. First of all it’s to provide security function. As one of the most important features in FairBlock block chain system, it can provide a bigger imaginary space for block chain application. And the design of security function or provider can gain continuous income by providing service to other developers. To compete by providing complicated functions and service quality based on security functions will become one of the important business ecologies in block chain.

Another important mode benefits from the appearance of super side chain technology. One of its important features is that it supports tree-like and multi-layer side chain structure. Developers can develop new side chains, which lays the foundation for a brand new side chain application development market. Developers can develop independent block chain application platforms based on FairBlock super side chains, and establish their own unique ecosystem in niche market. This will become the most imaginative business mode in FairBlock.

Considering the possibility of the above ecological mode, FairBlock is truly possible to become a distributed operation system on internet. In the business ecology blueprint of 3rd generation block chain technology in FairBlock, cancer-like “mining” profitable ecology will give way to the ecology that gains profit from programmers’ mutual development and construction. It gains profit because it produces actual value to others, and can more effectively make use of social resources. For example, the sharing economy brought by Uber brings benefit to the public, but not just an armament race with useless hardware played by the minority.

Let’s create a fairer world with block chain!

7, Team

7.1, Core

Calvin Ng

Zmax leo

Fenix

7.2, Advisor

Adam Stradling

Founder of Bitcoin.com

Tiago

Founder of Aptoide. Publisher of AppCoins.

Ryan Terribilini

Operating strategist of Google Play, PR Director of Ripple.

Gaurang Torvekar

Co-Founder of Indorse， Co-Founder of Ethereum Singapore Meetups

Andras Kristof

First block architect of FRD. Co-Founder of Bitcoin, Ethereum and Ripple，co-author of Handbook of Digital Currency.

7.3, Partner

Aptoide

Gumi

Gobi Partners

Google Play

Bitcoin.com

Kyber

8, Legal Disclaimer

This is a conceptual document (“White Paper”) describing our proposed FairBlock platform and FBC tokens. It may be amended or replaced at any time. However, we are under no obligation to update this White Paper or to provide the recipient with access to any additional information. This White Paper is for discussion purposes only.

Not available to all persons: the FairBlock platform and FBC tokens are not available to all persons. Participation may be subject to a range of steps, including the need to provide certain information and documents.

No offer of regulated products in any jurisdiction: FBC tokens (as described in this White Paper) are not intended to constitute securities or any other regulated product in any jurisdiction. This White Paper does not constitute a prospectus nor offer document of any sort and is not intended to constitute an offer or solicitation
of securities or any regulated product in any jurisdiction. This White Paper has not been reviewed by any regulatory authority in any jurisdiction.

No advice: this White Paper does not constitute advice in relation to whether you should participate in the FairBlock platform and FBC tokens, nor should it be relied upon in connection with, any contract or purchasing decision.

No representations or warranties: No representations or warranties are made as to the accuracy or completeness of the information, statements, opinions or other matters described in this document or otherwise communicated in connection with the project. Without limitation, no representation or warranty is given as to the achievement or reasonableness of any forward-looking or conceptual statements. Nothing in this document is or should be relied upon as a promise or representation as to the future. To the fullest extent permitted under applicable law, all liability for any loss or damage whatsoever (whether foreseeable or not) arising from or in connection with any person acting on this White Paper, or any aspect of it, notwithstanding any negligence, default or lack of care, is disclaimed. To the extent liability may be restricted but not fully disclaimed, it is restricted to the maximum extent permitted by applicable law.

English version prevails: this White Paper is provided in an official English version only. Any translation is for reference purposes only and is not certified by any person. If there is any inconsistency between a translation and the English version of this White Paper, the English version prevails.

Other companies: The use of any company and/or platform names and trademarks does not imply any affiliation with, or endorsement by, any of those parties. References in this White Paper to specific companies and platforms are for illustrative purposes only.

You must take all necessary professional advice, including in relation to tax and accounting treatment. You must assess the risks and your ability to bear them.

Risk Disclosures

Sophistication. Tokens are often described in exceedingly technical language; a comprehensive understanding of applied cryptography and computer science is required in order to appreciate inherent risks. By using the Services, you represent and warrant that you have sufficient knowledge, market sophistication, experience, and/or professional advice sufficient to undertake a prudent evaluation of the merits and risks of all transactions conducted by you pursuant to the Services. You agree to bear sole responsibility for the aforementioned evaluation.

Malicious Nodes. Some nodes in the FairBlock platform may be malicious and attempt to get rewarded without corresponding contribution; also, attackers may try to ruin the FairBlock ecosystem if they only suffer from minimal penalties. We need strong guarantees to protect the network from malicious attacks to ensure that the transactions are secured and the ecosystem is sustainable. Some attacks that could threaten a blockchain network are listed and discussed as follows.

Sybil Attack. Malicious nodes could create multiple Sybil identities to strive for more rewards or cheat the network. In general, the proof mechanism should have established barriers to prevent Sybil attacks; however, there is no guarantee such barriers will always be
successful.

Disclaimer. FairBlock hereby disclaims all responsibility for any loss or damage arising from or relating to your use of any Services (including, but not limited to, risk of losses due to trading or due to factors beyond its control regarding the viability of any specific blockchain network). FairBlock further disclaims all responsibility for any loss or damages arising from or relating to any cyber- attacks (including without limitation the theft of your personal information), unprecedented surges in trading volume, any disruption or shut down of the Services, or other technical
difficulties with respect to the Services.

Security of the Platform

You acknowledge that information you store or transfer through FairBlock services may become irretrievably lost or corrupted or temporarily unavailable due to a variety of causes, including software failures, protocol changes by third party providers, internet outages, force majeure event or other disasters including third party DDOS attacks, scheduled or unscheduled maintenance, or other causes either within or outside FairBlock's control. You are solely responsible for backing up and maintaining duplicate copies of any information you store or transfer through FairBlock's services.

Utility Purpose Only

Use and purchase of the tokens generated by FairBlock carries significant financial risk. FairBlock hereby expressly disclaims that the transactions taking place on its platform pertain in any way to an offering of securities in any jurisdiction or that any documents published on its platform are solicitations for investment.

Regulatory

Crypto-tokens are being, or may be overseen by the regulatory authorities of various jurisdictions. FairBlock may receive queries, notices, warnings, requests, or rulings from one or more regulatory authorities from time to time, or may even be ordered to suspend or discontinue any action in connection with the Website or Services. The development of the Website may be seriously affected, hindered, or terminated as a result.

Illiquidity and Price Volatility

FairBlock is not responsible for the circulation and trading of FBC tokens on the market. Tokens such as FBC tokens, if traded on markets, usually have extremely volatile prices. Fluctuations in price over short periods of time frequently occur, which price may be denominated in Bitcoin, Ether, US Dollars or any other fiat currency. Such fluctuations could result from market forces (including speculations), regulatory changes, technical innovations, availability of exchanges, and other objective factors and represent changes in the balance of supply and demand. The Seller is not responsible for any secondary market trading of FBC tokens, nor is FairBlock obliged to tame any price volatility of FBC tokens. Careful due diligence should be undertaken by you, with the full understanding that your contributions may not ultimately result in a useable or valuable token and the value of your

contributions may therefore be subject to total loss.
FairBlock does not make any representation or warranty, explicit or implicit, as to the usability or the value of any tokens. You understand and accept that there is no warranty or assurance that you will receive any benefits through any FBC tokens that you hold.

Compliance by Users

You acknowledge and agree that FairBlock is not responsible for determining whether or which laws, rules, or regulations apply or may apply to your transactions (including, without limitation, any anti-money laundering laws, securities laws and tax laws). You acknowledge and agree that you are solely responsible for compliance with all such laws rules, or regulations as may be applicable to your transactions. Without limiting the foregoing, you acknowledge and agree that you are solely responsible for all tax obligations arising from your use of the Services. You further acknowledge and agree that FairBlock shall not be liable, whetherdirectly or indirectly, for any of your tax obligations.

Foundation Compliance.

You acknowledge and agree that FairBlock's recordkeeping and customer verification procedures may be, without prior notice, subject to change at any time as required by applicable regulations or state of the art practices.
Applicable law, regulation, and executive orders may require FairBlock to, upon request by government agencies, freeze or suspend withdrawals or trading (or both), or disclose information regarding your Account(s). In the event such disclosure is compelled, you agree that FairBlock may disclose information regarding your Accounts. While FairBlock will endeavor to, where commercially reasonable, give you prior notice of such disclosure, FairBlock makes no guarantees that such prior notice will be made.