SmartX Technology Whitepaper

Content Page

1 Project Introduction

1.1 About SmartX

SAT is the main chain project developed by SmartX's community-centred team. It is a decentralized, minable BlockDAG project with a "transaction-based DAG" structure that can achieve limitless performance expansion, which is essentially equivalent to a traditional blockchain system that can achieve limitless horizontal expansion.

SmartX's platform will allow blockchain technology to be seamlessly adopted in use cases such as private communication, in-game payment as well as in products for example, coupons and points given by shops and merchants. It links service providers and users around the world and uses social entertainment as a gateway to building a trustworthy and secure social entertainment ecosystem.

SmartX will be an integrated and multi-functional platform for cashflow, dataflow as well as "value-flow". In SmartX's trust-based structure, intrinsic values of individuals and entities will be delivered through the network to form an ecosystem where both tangible and intangible values can be properly recognized. Such an ecosystem will greatly improve the efficiency of society as a whole.

1.2 Overview

With the advancement of blockchain systems, transaction speed has become a developmental bottleneck for the industrialization of blockchain technology in applications such as DAPP, Internet of Things and trusted data exchange. BTC/BCH and others achieve their expansion goals purely by increasing the size of the blocks, which is not a desirable method. In addition, BTC/ETH sub-networks, such as the Lightning Network and cross-chain technology, actually improve TPS at the expense of security. In addition, they also require more conditions (such as both parties of a transaction must be online concurrently, need for a pledge, etc.) for a transaction to go through. Facing these non-ideal solutions, an alternative solution appeared and that is BlockDAG which we are presenting in this paper. Early project such as IOTA and the most recent Conflux, fall under this category.

SmartX has its own proprietary, innovative and all-new DAG structural algorithm - B-DAG algorithm, which can integrate the transaction partitions created by any nodes around the world. This is vastly different from BTC/BCH, Ethereum and other current mainstream POW-based systems, which must halt all transactions (into "pending" status) before packaging them at the same time and yet the transaction blocks packaged at different nodes are still mutually exclusive (although Ethereum has GHOST and Uncle Block mechanisms, these are not enough).

It is the same as a highway that has only one exit. No matter how many lanes or how many cars there are, there is only one exit in the end. The exit speed determines the traffic and those at the back have to queue up. The SmartX project is equivalent to opening up an unlimited number of highways to make limitless traffic expansion possible.

SmartX generates a stable master component (MC) through Epoch and links it linearly according to the B-DAG algorithm which, if viewed from the perspective of the master component, resembles a bitcoin-like blockchain structure. However, the entire system is a in fact a transaction-based DAG structure that can be divided into independent partitions. These partitions do not interfere with each other and can be integrated after generation, hence ensuring limitless performance extension.

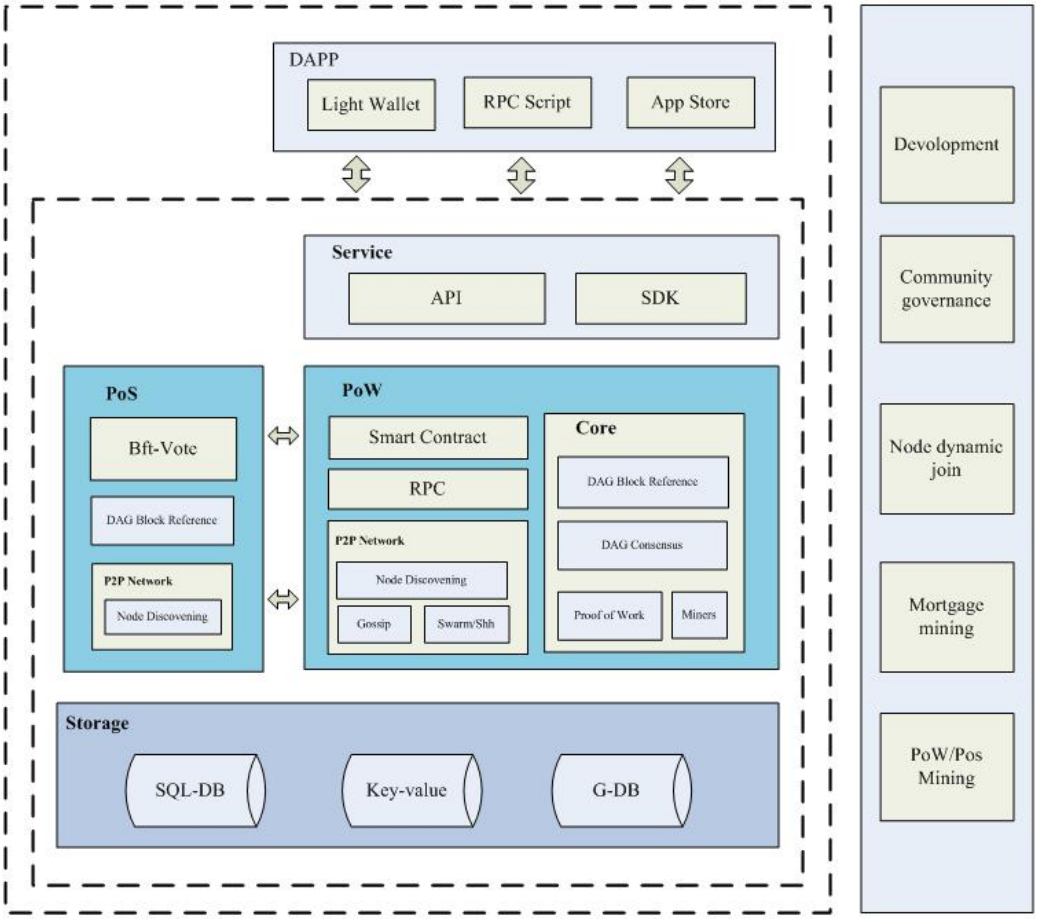
With the SmartX structure, double-spending detection will be possible sooner or later. The algorithm is stable and consistent globally. Once a node sees the global data of an Epoch, double-spending detection can be achieved with a stable sorting algorithm (B-DAG algorithm).

1.3 System Structure

Centralized system has a single point of failure which could jeopardize the whole network. Blockchain, on the other hand, is a rule-based decentralized system. Perfect decentralization requires that there is no single owner nor centralized stakeholders controlling the network, and thus eliminating the threats brought forth by a malicious single node. This is not achievable without leveraging on an immense amount of energy and resources.

For example, a single transaction executed on a centralized cloud server such as WeChat Pay will be completed within a split of a second, using less than 1k bytes of disk capacity and bandwidth, with negligible energy consumption. However, a similar transaction might take up to 100 thousands times more of the hardware resources and million times more of energy to complete in an ETH system (assuming there are 10,000 nodes in the ETH system).

In contrast, SmartX manages to combine some of the advantages of both centralized and decentralized system. The system structure will operate in a federal model. The main net will be composed of multiple PoW master nodes and PoS voting nodes and both types of nodes will need to pledge collateral prior to joining the network. The PoS voting nodes will be elected from the community by voting. The PoW master nodes, which are allowed to join and exit freely, will serve as the mining pool nodes that accept requests for transaction and ledger downloading, as well as submission of computing power from various mining machines. The common nodes will form the exterior of the network. By simply connecting to the common nodes, products and services such as light wallet, PRC script, SDK API, or APPs that connect to SmartX, will be made available for common users.



1.4 Transaction Verification and Confirmation

A transaction (TX) will first be sent to the storage pool of the nearest master node (NodeA). In a certain period T, NodeA will produce the main unit M to proactively make reference to this TX, and at the same time, combing TX to form a larger block - M-muster, which will then be broadcasted to all the master nodes in the network. Every master node will operate in the same way simultaneously and the M with the highest PoW will become the MC and emerges as the winning block. Different from BTC/ETH, the rest of the main units with lower PoW will not become invalid. Instead, they will be directly or indirectly referred to by the MC to form the B-DAG tree structure.

When MC of period T is confirmed, all the M and TX directly or indirectly referred by this MC will also be confirmed. This process is a classic PoW process. Because of this, the B-DAG tree structure is actually capable of operating independently. If SmartX stops here, it will be largely similar to other DAG projects such as IOTA.

However, SmartX introduces PoS consensus, a 3F+1 voters' network, into the network to accelerate the convergence of this B-DAG structure. When M with the highest PoW becomes the MC, it must be confirmed by the PoS voting nodes. In the SmartX mixed consensus mechanism combining both PoW and PoS, PoS serves as the network accelerator

for PoW consensus.

The introduction of PoS voting mechanism is not only helpful in the convergence of the dynamic structure of B-DAG, it can also be leveraged to vote for other on-chain consensus activities such as balance snapshot, data trimming, irreversible nodes, time consensus, node QoS services etc.. PoS voting nodes will be compensated with a fixed reward whenever a rewarding block is produced.

The advantage of using a PoW/PoS mixed consensus is that, by doing so, the DAG network will be able to satisfy real-life needs without sacrificing decentralization, security and TPS.

1.5 B-DAG Technology and Dynamic Capacity-expansion of Blocks

Theoretically, every master node in SmartX is capable of independently producing blocks, which are MC-Musters containing all the MCs and TXs referred by a single MC. These MC-Musters are highly scalable in theory and millions of transactions could be packaged into one MC-muster. Hence, B-DAG technology can be one of the dynamic capacity-expansion solutions even to traditional blockchain such as BTC.

It is clear that the demand for high TPS has bottlenecked the development of blockchain as it is much required in cases such as Fomo3D games, high-frequency on-chain games etc.. SmartX will try to solve this problem by leveraging on B-DAG technology.

2 The DAG Structure of SmartX

2.1 DAG Structure

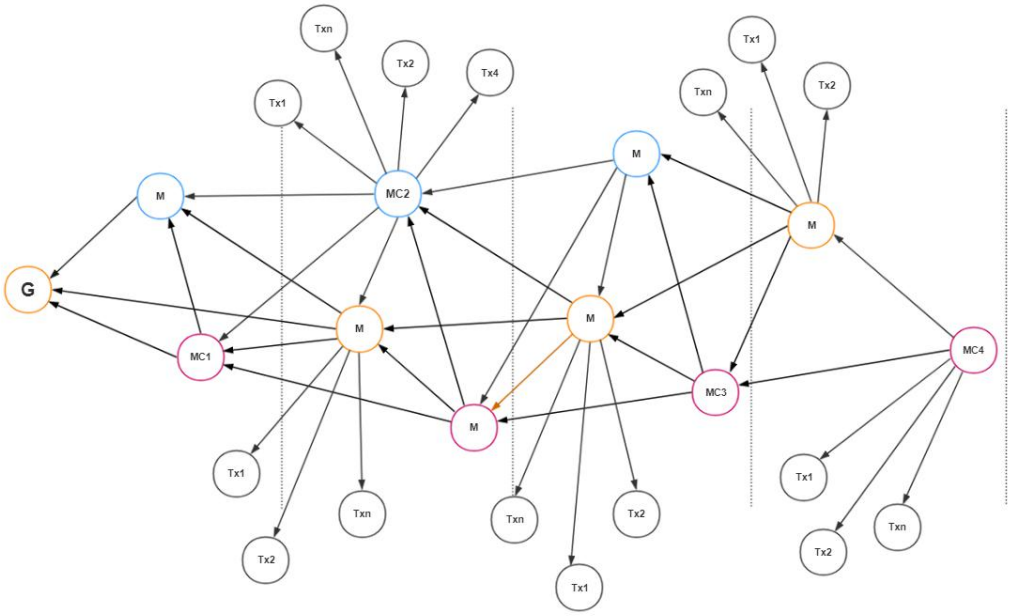
Unlike the traditional BTC blockchain, a SmartX block is a transaction (TX) and a TX is a block. In addition to the TX component, the SmartX system also has an MC that generates an Epoch at a certain interval to make reference to previously un-referenced transactions. If we look at every single MC and TX, intuitively, it resembles the common DAG structure; but if we look at every single Epoch instead, as time passes, it will resemble a BTC blockchain structure whose transactions are not in order if only the SAT main chain references (will be elaborated on later) are retained and the rest of the MC reference lines are removed. This structure combines some of the advantages of Blockchain and DAG technology. In this paper, we call it B-DAG technology.

Similar to most blockchain systems, the risk of DDOS attacks is eliminated through an administrative fee. However, as the B-DAG model does not require the entire network to halt before packaging transactions into one large block, it is unable to control the

transaction volume. When the transaction volume reaches a very high level, it will create an avalanche effect. The method to resolve this issue is to hand over the valve that controls transaction volume to the master nodes. The master nodes dynamically configure the valve settings based on their respective node performance metrics.

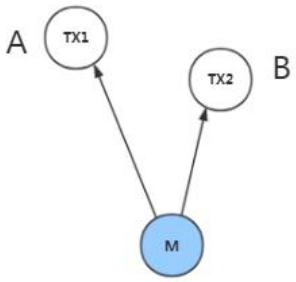
Master nodes: Mining nodes that create new blocks. These nodes are similar to the BTC/ETH mining pool nodes, except for that they will need to pledge a certain amount of SAT to join the network.

A typical B-DAG structure can be found below:



2.2 Simple Block Structure

Main block M makes reference to 2 transactions - TX1 and TX2. This was signed and confirmed by PoS nodes A, B, C and D.



If JSON is used to describe the structure of a signed simple main block reference, it will be:

{

"hash": "THIS HASH",

```
"time": "2019-08-06 19:10:18",
      "type": "1",
"diff": "THIS diff",
"owner": "OWNER ADR OR PUBKEY",
"nonce": "5CC911F3B8434911B691B6CF7A361333",
            "flds": [{
            "type": "SAT_FIELD_OUT",
            "hash": "A",
            "time": "2019-08-06 21:21:19"
      }, {
            "type": "SAT_FIELD_OUT",
            "hash": "B",
            "time": "2019-08-06 21:21:19"
      }],
      "Signers": ["A", "B", "C", "D"],
      "Signinfo": "SIGN CONTENT",
      "sign":"THIS ECDSA SIGN"
}
```


2.3 Account Model

Like traditional trading systems, SmartX uses a balance account model. During every Epoch consensus processing cycle, if it is found that an OUTPUT account for a certain TXBlock does not exist, the account will be created across the network. The input amount will be transferred to this account and the input must have sufficient balance.

Each account balance is determined by the difference between the input and output transaction components. In a certain Epoch, from the genesis block to the current moment, account Balance = ALL (INPUT) - ALL (OUTPUT). Every transaction component is signed by the INPUT party with an ECDSA private key and the validity of the block is verified by the INPUT public key.

The system ensures that every transaction is treated as idempotent by the state machine, i.e., the end result is the same regardless whether a TX is executed once or multiple times. When the balance is modified, it has to be locked at the same time, and every processes and function of the system must be re-entrant-able. As a result, the account balance is modified in the same way as the state machine: no matter how many transactions are inputted or how many times the transactions are repeated, the end result is the same.

Duplicated transactions are recognized by the random Nonce value of the transaction. If the Nonce value is the same, and if the hash value of the entire package is the same, it is called a re-entrant success; otherwise, it will be called a re-entrant failure. The system processes these two cases separately.

At the same time, every account in the account list will have a Nonce field to examine the existence of repetition of every outgoing transaction. However, if repetition exists in different pools, after adequate ranking, blocks with higher priority will be executed first while blocks with lower priority will only store hash.

In the later block updates, blocks with higher priority will be updated first. Blocks that only store hash, will be carefully examined.

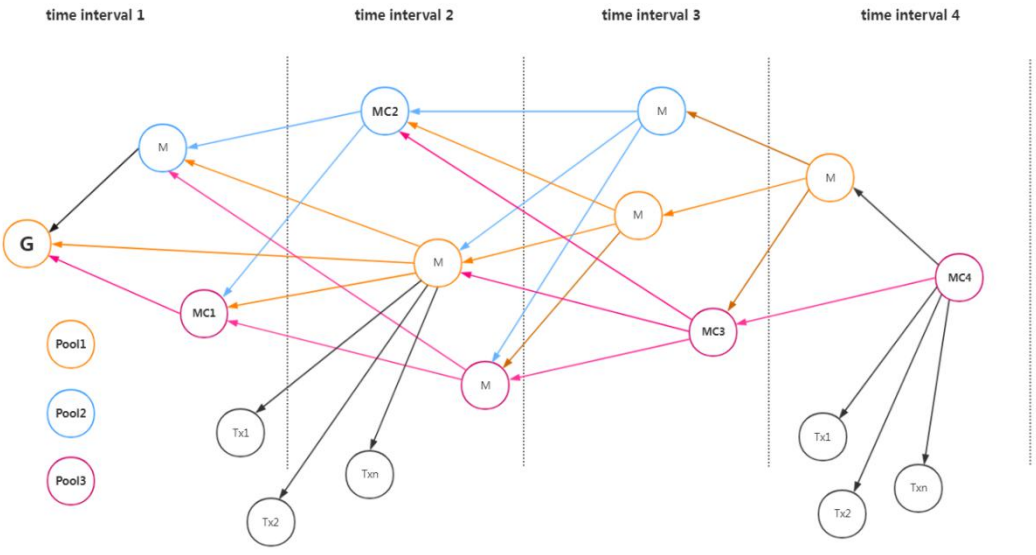
2.4 Reference Relationship

The witness nodes across the entire network generate MC and all the TX under the corresponding Epoch at the same time. All MCs and TXs will refer to each other according to the following rules:

- Convention Rule 1: A witness node can generate an MC that can refer to other MCs that are being broadcasted, but the broadcasted MCs cannot refer to the target MCs.
- Convention Rule 2: A MC can refer to a TX, after which, this TX cannot be referred to by any other MCs.
- Convention Rule 3: Loop references cannot be made between MCs.

Main block: A main block generated by a certain witness node for a certain Epoch. The main block actively searches for all un-referenced broadcasted blocks and TXs in the memory pool.

Broadcasted main block: After a target main block generates and completes the referencing operation, the main block becomes a broadcasted main block after being broadcasted to the connected nodes.



As shown in the figure above, SmartX uses forward-reference rules, which distinguish whether to refer to the main blocks broadcasted by other mining pools according to cycles:

- If the broadcasted main blocks and the current main blocks are in the same cycle or after the current main blocks' cycle, they are not referenced for the time being, but will have to wait until the start of the next cycle to make reference to the broadcasted main blocks.
- If the broadcasted main blocks are in a cycle that is before the current mining pool cycle, these main blocks will be referred to if they comply to the reference rules.
- All the units, M (including the MC), must make reference to the MC that is verified in the last Epoch.
- Al the main blocks that are not directly or indirectly referred to by MC in the last Epoch will be discarded by the main net.

3 SmartX Consensus Mechanism

3.1 PoW-PoS Mixed Consensus Mechanism

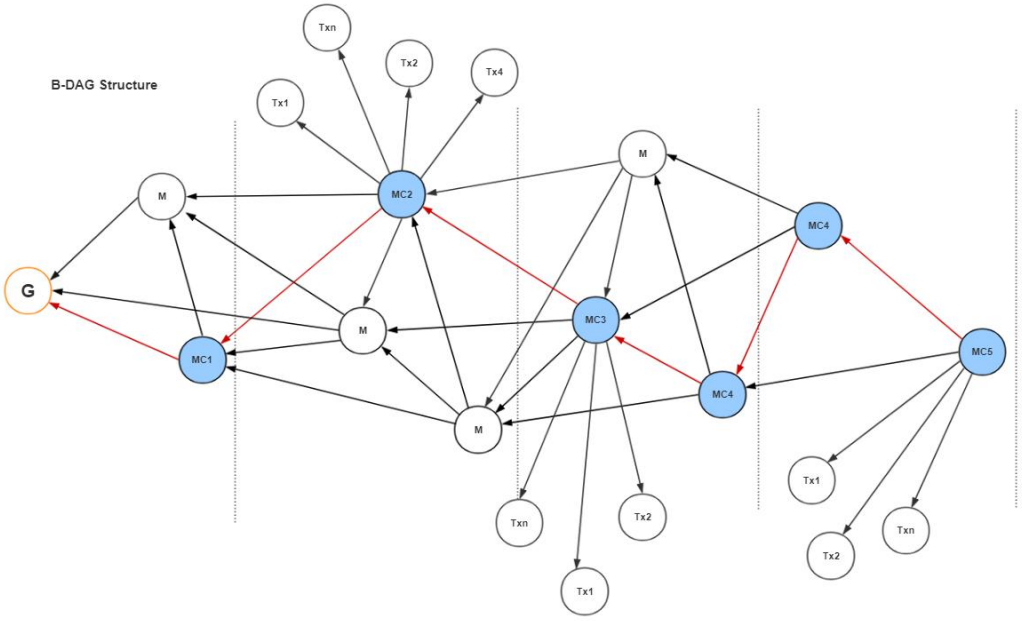
SmartX will adopt a mixed consensus mechanism combining PoW and PoS. Among them, PoW consensus mechanism will choose the block with the highest amount of PoW to be the winning block. The PoS tier will be a 3F+1 BFT consensus network. The MC with the highest amount PoW ought to be the one that is voted by the PoS voters. PoS tier works as an accelerator to fasten the transaction confirmation process.

PoS tier will also be able to accomplish other types of consensus voting for irreversible nodes, balance snapshot, PoS services, penalizing abnormal transactions, time calibration etc.. When an isolated web is segregated, the main net will be protected by both PoW and PoS.

3.2 Most Heavily Weighted Chain

In SmartX, MCs generated for a specific Epoch will make reference to each other. A most heavily weighted MC (MC with the highest amount of PoW) can be selected in a stack of MCs by POS/POW mechanism, which is known as the winning MC (main MC). Main MCs forms a special line in the entire DAG structure throughout the Epoch cycle and this line is called the SAT main chain.

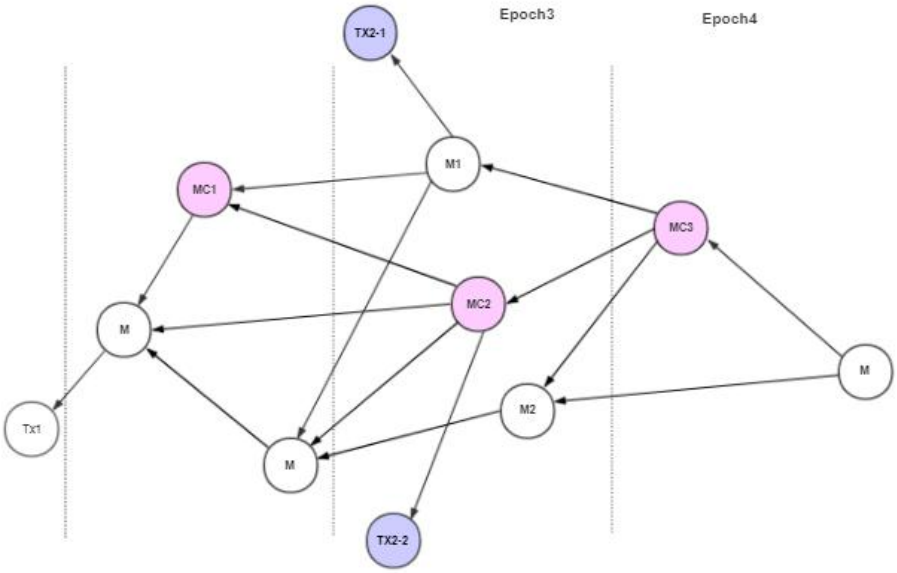
The SAT main chain can sequence all TX serial numbers (according to the topological relationship, time and hash value relationship) and the double spending is eliminated using these serial numbers.



3.3 Double Spending Elimination

In the figure below, there are two double-spending transactions, TX2-1 and TX2-2 in Epoch 3. Since the SAT main chain has confirmed MC1<-MC2<-MC3, MC3 is able to tell based on topological sorting that the serial number of MC1 has priority over MC2. As a result, the serial number of TX2-1 has priority over TX2-2, so TX2 -2 is eliminated. This ensures the successful detection of double spending. The double spending detection sequencing rule is as follows:

- Blocks that are directly reachable by the MC - can be sorted according to the topological sequence between blocks and main MC.
- If the MC of this Epoch is not reachable, then it is sorted based on the shortest path, i.e., the shortest path to get to the genesis block.
- If the sequence cannot be determined using the above rules, then sorting will be done by block hash and time.



3.4 Merkle Hash

Every MCs of the SmartX has a current merkle_hash = hash (MC_hash + Pre-merkle_hash), i.e., its own hash with the hash of all blocks referenced by the MC and the resulted hash. Unlike blockchains such as BTC, SmartX does not have a single-direction traceable merkle hash. But when each Epoch cycle looks back at the Epoch-N cycle, the merkle hash of each MC must be consistent and SAT uses this hash to ensure the final consistency of the data at all nodes.

3.5 Checkpoint and Data Cropping

As time goes by, the blockchain system data will swell so much that ordinary nodes cannot bear the load. In view of this, in theory, as long as the data is taken as a snapshot before a certain checkpoint, then only the account balance sheet and part of the Merkle hash will be all it needed for verification. The system can still participate in mining and trading, which will greatly reduce storage pressure for common nodes.

4 Ecosystem and Mining

4.1 Pledge Mining

SmartX Tokenomics and incentive scheme are carefully designed with the purpose to achieve a balance between different market forces.

SmartX will have several PoW genesis nodes by default and a default amount of 1

million pledged tokens (an even larger amount for PoS), in order to allow participants who are more willing to participate in node operation to participate. SmartX allows for the mining nodes to join and leave at any time. The purpose of this is to select nodes that have a stronger willingness to participate.

SmartX's pledge mining will go through the following stages:

- **Initialization**
  Initialization involves several genesis nodes and a default amount of 10 million pledged tokens. Check if there is a pledge mark in the pledge form. If not, the main block will be rejected.

- **Joining of a new node**
  Initiate a pledge transaction, content: number of pledged tokens, public key, private key signature and time. After most of the nodes have received the block, the node enters the information into the pledge form. After 2 minutes (experience value), the ordinary node can participate in mining. If the pledge transaction is not received by a certain master node, the main blocks created by this node will be rejected.

- **Exiting of a node**
  Initiate a reverse pledge transaction. After most of the master nodes have received it, the pledge mark in the pledge form will be emptied, indicating that the pledged tokens have been withdrawn. The node can transfer away the pledged tokens as scheduled. If another master node finds that the pledge mark is emptied or does not exist, the balance of this node can be transferred away.

4.2 Smart Contract

The DApps developed by smart contracts will enrich the ecosystem. SmartX will support smart contract and use EVM virtual machine temporarily. SmartX is planning to provide support to the further development of smart contracts after the test-net is launched.

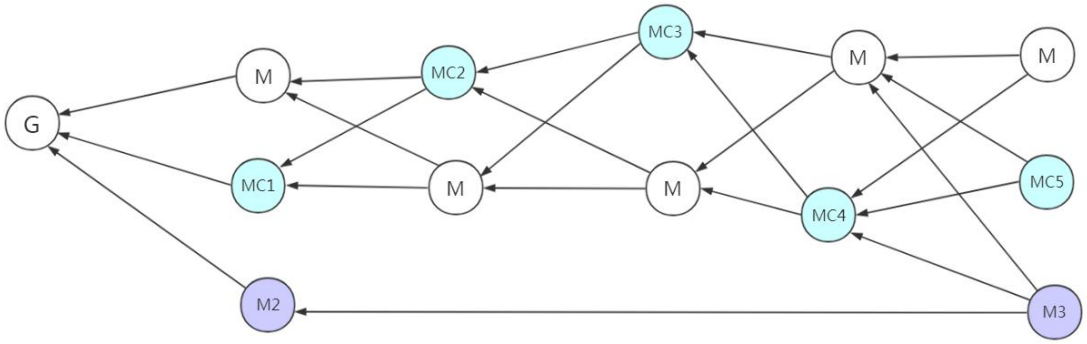
5 Security

5.1 Forking Attack and Solution

Suppose there are 2 forks: the main chain G <- MC1 <- MC2 <- MC3 <- MC4 <- MC5; and an illegally mined chain G <- M2 < M3. If M3 represents a higher difficulty level compared to MC5, then in a normal DAG network, M3 will be the winning block and this will cause the illegally mined chain to replace the main chain while the original main chain will be forced to

roll back. This is known as 51% Attack in the context of BTC. If a single node is able to wield a tremendous amount of computing power, it will be capable of launching such an attack.

In SmartX, however, the 3F+1 nodes will interfere and make a judgement call to prevent M3 from winning in such a situation.



5.2 Timestamp

Like most blockchain systems, the SmartX system has a by-default premise, which is that most of the witness nodes have the same timestamp. If the witness nodes' time deviates too far from the NTP time, the mining thread will be stopped. A block will be rejected if the time of a block received by a certain witness node from a neighbouring node exceeds the NTP time by a certain period. If the node time deviation is within the tolerable range, the delay will be considered to be caused by network malfunctioning.

Time slices will be produced per every fixed period T according to Greenwich timestamp. All the main blocks that are generated based on the current time will be mapped to the Greenwich time slices. If they can agree, then it is confirmed that the timestamps used are the same.

Most BTC-like blockchains are susceptible to timestamp attacks, which is, the next difficulty adjustment value is the total difficulty of several previous blocks divided by the difference between the starting and ending time. Since the timestamp cannot be controlled, an attacker can force the timestamp to fluctuate violently by forging fake timestamps, resulting in a large fluctuation in terms of difficulty adjustment. The direct consequence is that the network cannot generate new blocks or the block are generated too fast which leads to over-mining in effect.

SmartX will avoid such a situation, i.e., blocks are generated periodically and the difficulty is not adjusted. As time passes, the latest MC difficulty is equal to the sum of the difficulty since of the genesis block. The main MC is determined by the sum of all difficulties since the genesis block. Main MC is determined by comparing the total difficulty (under the premise of using POW).

5.3 Sybil Attack

Sybil attack is proposed in 2002 by John R. Douceur. It is a form of attack in a Peer-to-Peer (P2P) network: an attacker uses a single node to forge multiple identities in a P2P network, thereby weakening the network's efficiency and reducing its robustness. The forged identities can be used to monitor or disrupt the normal activities of the network.

In the SmartX network, in order to address the security threats from malicious nodes or node failures, senders of the message will be required to sign every message with his/her private key. The parameters for verifying the signature include the message content, the block height, and the Epoch at which the current block is located. Each public key can only submit one block in a block-output cycle. If there is repeated submission, the block that is received the first will be recognized, and the rest of the blocks will be ignored. If verification fails, the block issued by this public key will be discarded.

6 Experimental Functions for SmartX

** Note: launch dates of the experimental functions described here are not confirmed.*

6.1 Use Penalty Block to Achieve Blacklist

This experimental function is meant to add a new transaction type named Penalty Transaction: once the consensus hit a certain threshold, Penalty Transaction will be executed upon the consent obtained from the voting done by PoS voting nodes and validated after N blocks. Penalty Transactions are meant to defend the network from malicious attacks and hackers and it will impose penalties such as suspension from making transactions for N periods of time or deduction/confiscation of mining rewards.

6.2 SmartX App Store

Apart from EVM virtual machine and smart contract, SmartX plans to introduce a PRC script to facilitate program development based on SmartX. A marketplace for the scripts created - the SmartX store, will also be introduced.

6.3 Dynamic PoW Hash Algorithm

Hash functions that uses script languages such as SmartJS to achieve PoW. PoS nodes will initiate an algorithm-updating transaction to place the script in the transacting blocks. This process is similar to joining the main node by making a pledge. After a certain period, the new algorithm will start to operate.

6.4 Distribution of Encrypted Information

The P2P network of SmartX will allow nodes to transmit encrypted information and data. On top of this, the realization of decentralized and encrypted social media App is also achievable by utilizing decentralized storage systems, SWARM and SHH. SmartX, with its excellent system performance, will be able to ensure the quality of user experiences of such Apps.

7 Project Roadmap

- 2019/07 - Complete SmartX infrastructure and basic trading infrastructure
- 2019/09 - Improve SmartX's voting-POW mixed consensus mechanism
- 2020/02 - Bring SmartX test-net online

8 Team

Frank Su

Frank worked at WeChat Pay as a senior engineer and contributed to the development of payment system of various WeChat Pay wealth management products and Hongbao. During his time with WeChat Pay, he applied for 2 patents for his creation. He restructured the code underlining XDAG and was also the ex-chief engineer for EtherZero(ETZ) DAG project.

Bill Ng

Bill was previously a senior engineer at Ping An Insurance, and a developer of XDAG wallet. He is highly experienced with Monero code and an expert in anonymous communication.

Li Zhao (Technology Advisor)

He used to work at WeChat Pay as a senior engineer and contributed to the development of a WeChat Pay wealth management product as well as mobile payment system. He has 12 years of experience in payment gateway and is highly knowledgeable in BlockDAG algorithm. He also restructured XDAG code.

Jessy

Jessy graduated from Stanford University with a Master degree in Marketing. Jessy also obtained a Master degree in Web and New Media Design from Academy of Art University (San Francisco, USA). Jessy is highly experienced in Visual & UI design, as well as product marketing.

JunQ

Senior operation manager with extensive team management experience. Previously worked for Tencent and iDreamsky Games.

9 Token Distribution

- Total of 10 billion
- 43% for community airdrops
- 30% for node mining
- 17% for foundation (locked)
- 5% as reserve (locked)
- 5% for early-stage ecosystem and legal expense (locked)

10 Reference

[1] LWMA Difficulty Algorithm, https://github.com/zawy12/difficulty-algorithms/issues/3.

[2] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters, Twisted Edwards Curves (2008), https://eprint.iacr.org/2008/013.pdf.

[3] Ethash ·ethereum/wiki Wiki - GitHub, https://github.com/ethereum/wiki/wiki/Ethash

[4] Ethereum. Ethereum. https://github.com/ethereum/wiki/wiki/White-Paper

[5] IOTA. IOTA. http://iotatoken.com/IOTA_Whitepaper.pdf

[6] Byteball. Byteball. https://obyte.org/Byteball.pdf

[7] Cardano.Ouroboros A Provably Secure Proof-of-Stake Blockchain Protocol. https://iohk.io/research/papers/#ouroboros-a-provably-secure-proof-of-stake-blockchain-protocol

[8] Cardano.Ouroboros Praos – An adaptively-secure, semi-synchronous proof-of-stake protocol. https://iohk.io/research/papers/#ouroboros-praos-an-adaptively-secure-semi-synchronous-proof-of-stake-protocol

[9] J. Chen and S. Micali. Algorand. Technical report, 2017.URL http://arxiv.org/abs/1607.01341

[10] Peercointalk. Peercoin invalid checkpoint.https://www.peercointalk.org/t/invalidcheckpoint/3691, 2015

[11] D. Dolev. The Byzantine Generals Strike Again. J. Algorithms, 3, (1982), pp. 14-30.

[12] D. Dolev and H.R. Strong. Authenticated algorithms for Byzantine agreement. SIAM Journal on Computing 12 (4), 656-666.

[13] P. Feldman and S. Micali. An Optimal Probabilistic Algorithm for Synchronous Byzantine Agreement. (Preliminary version in STOC 88.) SIAM J. on Computing, 1997

[14] Philipp Winter, Roya Ensafi, Karsten Loesing, and Nick Feamster, Identifying and characterizing Sybils in the Tor network (February 25, 2016), https://arxiv.org/abs/1602.07787.

[15] The Sybil Attack.JR Douceur， https://www.freehaven.net/anonbib/cache/sybil.pdf

[16] Go Ethereum - Postal Services over Swarm, https://github.com/ethersphere/go-ethereum/blob/ddfc0a2a02ce574f4c252068ce81f0f5ada1c1ff/swarm/pss/README.md