MAP Protocol Whitepaper 3.0

August 17, 2020

Contents

1 UpdateList

This version is MAP protocol V3.0. It will introduce more details of Ultra-light Local Chain Verification Protocol and MAP-POS consensus. Also, it will describe some application scenarios of MAP protocol.

2 Overall

MAP protocol is an open and completely decentralized chain-to-chain inter-operation protocol that enables the interoperability of multiple independently verifiable consensus blockchains without a relay chain. MAP protocol expects to construct a peer-to-peer future internet with a huge amount of flexible interoperabilities through chain-to-chain TPS, privacy computing, storage, security, and other resources. Resources and capabilities of each chain will be integrated into an inter-chain and form a unified blockchain infrastructure for all blockchain application developers. This underlying infrastructure can be used for finance, AI, IoT, traceability, and governance.

Compare with Polkadot and Cosmos, MAP has essential differences. If we consider each blockchain as a decentralized computer (which can automatically execute programs and store data), Polkadot and Cosmos are committed to building a cloud-like computing system and all inter-chain interoperability can only be achieved through a specified relay chain or a hub chain. MAP protocol is a TCP/IP-like protocol, through the free and open interoperability between blockchains, forming a borderless, open ecosystem composed of heterogeneous interoperable blockchains, providing solid underlying infrastructure for blockchain applications.

At the same time, we will provide a benchmark chain (MAP Standard Chain) that decouples the consensus component and the state transition component. Also, MAP protocol will provide a general framework MATE (MAP Application Tool Environment), which is used for doing MAP blockchain development with in-build consensus module, interoperability module, libp2p module and other modules. Developers can easily start building their own blockchain and easily realize the interoperation between chains. It worth mentioning that MAP Standard Chain is not the only benchmark chain to be used in the MAP system, anyone can build other benchmark chains similar to MAP Standard Chain to server the ecosystem.

Besides MAP Standard Chain, we will also build a peer-to-peer electronic cash payment chain and etc., to provide complete infrastructure services for financial applications such as DEX, Dpayment, and Defi.

3 Background Introduction

Since the birth of the Bitcoin in 2009, thousands of public chains have been produced by using blockchain technology and those are widely used in multiple application fields such as IoT, finance, governance, identity management, and traceability. In the future, more public chains will be made. As an emerging and immature industry, some normative, simple, open-source, safe, and technical standards that do not harm the interests of others will be generally recognized and cited by the public chain to themselves, such as the BIP32 standard of HD wallets and LibP2P Discovery agreement, etc. There is a pain point in the current blockchain industry: Scalability and Decentralization cannot coexist. Most public chain networks are limited to 10-30 transactions per second(E.g. Bitcoin, Ethereum), and blockchain projects that can achieve higher TPS mostly use weakly centralized designs (such as EOS)-the reason of this limitation come from its underlying consensus architecture: state transition machine (state machine), so that all participants agree on all possibilities, validity, and history. If you want to increase scalability, you must reduce the size of the participants, which will inevitably weaken the degree of decentralization. Therefore, in order to solve this dilemma, Polkadot suggests decoupling the consensus component and the state transition component, running the consensus engine on the Relay Chain, and running the state machine on each parallel chain. This approach can indeed solve the dilemma that single-chain scalability and decentralization cannot coexist, but it will bring new problems, that is, with the increase of parallel chains in the ecology, the processing capacity of relay chains also has bottlenecks. At the same time, the design of all parallel chains anchored in the relay chain also caused the safety of the entire ecology to depend heavily on the robustness of the relay chain. Although the blockchain running on it is decentralized, this relay chain is indeed the center of the entire ecology, which is contrary to its original intention at the beginning. The sharding technology of Ethereum 2.0 is essentially the same. Beacon Chain and Relay Chain have similar roles and will become the center and bottleneck of the entire ecology, which determines the scalability and security of the entire ecology. In order to solve these problems and to implement the technical banner of blockchain decentralization, we have proposed the MAP protocol, which is a low-cost and high-security blockchain interoperation protocol designed to serve for the underlying communication of the future blockchain ecosystem as a possible standard paradigm.

4 MAP Protocol

MAP is a universal, secure, and low-cost chain-to-chain interoperation protocol. As more and more blockchains use MAP protocol to interlink with other chains, a multi-chain resource and performance sharing bottom layer internet will be formed. In addition, it will become a robust and complete infrastructure for various DAPP. At the same time, we will develop two blockchains that support MAP to provide some services and demonstrations: one benchmark chain called MAP Standard Chain and one electronic cash payment chain.

4.1 MAP Protocol Design Philosophy

Unlike other cross-chain platforms with many restrictions, MAP protocol is committed to creating a free inter-chain protocol. The goal of MAP protocol is to build a large-scale open collaboration infrastructure with a large number of interoperable blockchains, which can serve various DAPPs such as finance, IoT, and traceability. By building a free interchain protocol that ensures interoperability between chains, each future blockchain can interoperate with other blockchains through this protocol, and each DAPP can be freely selected according to the usage scenario and configure the underlying features required by it, you can choose the one with the higher degree of decentralization, or with higher scalability or privacy computing. Cooperations between blockchains through MAP protocol to provide complete and easy-to-use infrastructures for DAPP. With the development of the blockchain industry, more blockchains that focus on specific technical areas will appear in the future, rather than large and capable for all use cases. This pattern has already appeared in practice, such as smart contracts for Ethereum, high TPS for EOS, privacy computing for Zcash, and data on-chain for Link. The blockchain will continue to improve in the future and form a large-scale Internet infrastructure through MAP protocol.

In the early stage of the infrastructure, we will provide a benchmark chain(MAP Standard Chain) that decouples consensus components and state transition components, as well as a public chain focused on the field of electronic cash payments that can interoperate with MAP Standard Chain. It should be noted here that the biggest difference from other Relay Chain-based solutions is that the benchmark chain is open to competition. Whether it can gain trust from other chains is obtained through free competition. In the future, there will be other public chains that similar to MAP Standard Chain through competition. A large number of verifiable and globally dependent dynamic data structures can exist in these public chains. These public chains will abandon complex functional applications (there will be not many numbers of DAPP), only focusing on the security of its own consensus and the validity of its global data. Any other blockchain can obtain these global data through MAP protocol and perform subsequent operations.

The most important point is that other blockchains are not only anchored on the benchmark chain but can also directly communicate with other blockchains without global data directly through MAP. If you think of Polkadot's parallel chain as a computer (the state database on it can be considered as a disk, and the virtual machine can be considered as memory), then Polkadot's goal is to build a server-client network based on Relay Chain. The parachains are all clients, and they need to get the data of the shared database from the server (Relay Chain). Our goal is to build a Peer-to-Peer communication protocol so that each blockchain can interoperate with any other blockchain in the entire blockchain world.

4.2 Consensus Algorithm

There will be many blockchains with different characteristics in the blockchain ecology under MAP protocol, and we will support blockchains with various consensus algorithms to coexist in such an ecosystem. However, there will be some differences according to the specifications of our interoperability agreement.

POW: Starting with Bitcoin, the POW consensus algorithm can be regarded as the longest time-tested consensus algorithm in the blockchain field. It provided the strongest degree of decentralization and provide a safe consensus base for the entire ecology. Other chains in the ecosystem can directly use Flyclient's ultra-light verification protocol to read the shared data on POW public chains. However, there is a big problem with the consensus of the POW type. A large amount of computing power is concentrated on the Bitcoin network, and the security of other POW type public chains cannot be guaranteed by sufficient computing power.

POS: The POS consensus algorithm is a supplement to the POW consensus algorithm, which first appeared in the PPCoin project. It was born to solve the waste of POW resources. Ethereum 2.0 used Casper which is also a kind of POS consensus algorithm. In the future, the underlying public chain of ecology may use it and provide shared global data for use by other application public chains in the ecosystem, just like the POW-type public chain. The shared data on it can also be directly read by other chains in the ecosystem through Flyclient's ultra-light verification protocol. The MAP Standard Chain we mentioned earlier will use POS consensus, more details will be mentioned in the next chapter.

Proof of X: This includes Proof of Space, Proof of Elapsed-Time, and various other POX consensuses. These consensuses have a common feature, that is, they can be independently verified by any node without additional information (the same is true for POW and POS, but it was discussed separately above because of its particularity). These types of consensus are suitable for some public chains with special needs because of their characteristics. At the same time, their data can also be read by other chains in the ecosystem through flyclient's ultra-light verification protocol.

Delegated Proof of X: This includes various types of delegated public chains, including Delegated proof of work and Delegated proof of stake. The characteristic of these consensuses is that some delegates need to be elected, and then these delegates perform operations such as consensus generation.

However, these election processes may be off-chain or unpredictable, and therefore cannot be directly verified by other nodes. However, because the size of the cluster participating in the consensus has shrunk, such a public chain has strong scalability and is very suitable for some practical application scenarios. The data of these chains cannot be directly read by other chains in the ecosystem through flyclient's ultra-light verification protocol. At this time, these public chains need to be anchored on some other type of public chain and various core security data (such as representative's vote information, representative's identification, etc.) on the anchor public chain needed to be updated regularly. At this time, the security of interoperation requires additional consideration of the security of the anchor public chain to obtain these core security data.

4.3 MAP Protocol Sub Modules

MAP protocol includes two core sub-module:

(1) Ultra Light Verification Protocol(ULVP): ULVP is a sub-linear verification protocol for POX-based blockchain.

(2) Smart Script System With Combinable Trigger Conditions: Smart script system with combinable trigger conditions is the transaction execution module dealing with mixed trigger conditions.

4.3.1 Ultra-light Local Chain Verification Protocol

For a larger group of users, the need of reading partial data from the blockchain is required. However, the simplified payment verificationSPV[6] technology requires users to download the block header chain and save it locally. The idea of straightly using SPV proofs is unrealistic in many critical use-cases such as mobile phones or IoTs because the header chain grows linearly as the blockchain grows. For example, the headers of Ethereum is growing at a pace of around 1 GB per year.

The effort about finding a sub-linear verification method began with Kiayias et al [1]'s research on NIPOPOW. Their work can be applied to the POW type of public chain in the invariable difficulty setting. Most POW-based blockchain can not use this technology because they are under a variable difficulty setting. Is there a sub-linear verification method to work in a more general environment? The answer is Yes. Bunz et al [4] proposed

Flyclient to extend NIPOPOW to any Proof-of-X public chain (POW, POS, POET, etc.). MAP protocol uses Flyclient technology to construct Ultra Light Verification Protocol.

If a node wants to query a predicate relate to one blockchain using SPV, he needs to download a header chain. Then, he can request some randomly selected full node to relay the corresponding Merkle branch proof. After that, he can verify the Merkle branch proof based on his local stored header chain. For ULVP, the first step is to validate one target block that exists in the blockchain using MMR and random sampling. Once this step is done, the remaining part is to get the Merkle branch proof corresponding one predicate to relate to the target block and verify this proof. The second step is the same as SPV, I will elaborate the first step and show why this is called Ultra Light Verification Protocol.

Firstly, I want to introduce a new data structure named Merkle Mountain Range(MMR). This is a variation of the Merkle tree. The leaf nodes of MMR are block headers. Figure 1 shows the MMR construct by 12 block headers(from B0 to B11). The root of this MMR would be store in B12. Then, it is easy to provide Merkle branch proof corresponding any block from B0 to B11 if B12 was given. For example, if one verifier node needs to verify B5(blue marked) which was contained in the blockchain whose tail block is B12, then the prover can generate the Merkle branch proof(green marked) and relay it to the verifier. The verifier node can recover the root based on the Merkle branch proof and compare the recovering root to the root stored in B12. If the two roots are the same, then it means B5 is indeed in this blockchain, otherwise means B5 is not.

The first step of ULVP is to validate one target block that exists in the blockchain. Thanks to MMR introduced above, this problem is equivalent to prove one given tail block of a blockchain is indeed the tail block of the main network. For any POX-based blockchain,it would allow any node to verify the validity of each block individually ensuring that the block creator has spent (or burnt) a certain amount of a resource uniquely for this block. For POW, the resource is hash power, and for POS, the resource is stake. Under the assumption that adversary cannot control more than 50% resource of the whole network. This means that the adversary can not craft more blocks than honest majority. Thus if the adversary want to cheat a verifier, he need to create some malicious blocks. ULVP would design a sampling algorithm to make sure the verifier detects malicious blocks with overwhelming probability. Thus the verifier could use a small sampling set of block headers to check

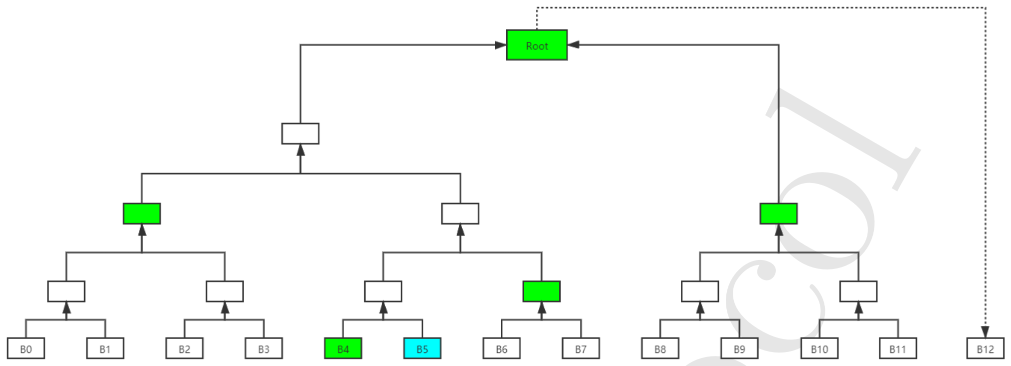

Figure 1: Merkle mountain range construct by block headers

the validity of the tail block of the main network.

For example, for POW-based blockchain the resource used for validity is hash power.Let us denote the advantage of the honest node is α. This means that the adversary can generate α blocks when an honest party generates one block. Let us see Figure 3: we set $\alpha = 0.4$, then, when an honest party generates 5 blocks, the adversary can mine 2 blocks(yellow blocks). If he wants to cheat the verifier, he will need to fill 4 malicious blocks(blue blocks) to exceed the honest chain. If the algorithm can sample enough so that the verifier can detect these malicious blocks, then this algorithm can be used to prove the tail block one prover claim is indeed filled by all valid-mined blocks. Thus, the verifier can compare the length of all the valid candidate tail blocks and simply select the longest one. This would guarantee that the verifier picks the exact same tail block as the honest majority picks.

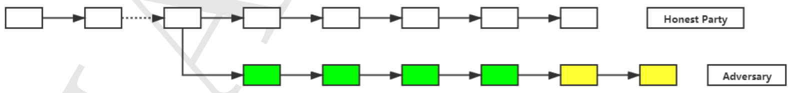

Figure 2: Adversary fill some invalid blocks to exceed honest party

Now the problem becomes how to design a sampling algorithm to make sure the verifier detects malicious blocks with overwhelming probability under the POW set. It can prove that for any sampling distribution over the blocks defined by a non-increasing PDF f, there exists another distribution with an equal or higher probability of catching the adversary. Thus, the optimal distribution over the blocks must be defined by an increasing PDF. Furthermore, Bunz et al [4] prove that the optimal distribution defined by the PDF $g(x) = \frac{1}{(x-1)ln\delta}$ maximizes the probability of catching any adversary that optimizes the placement of invalid blocks. And if we use this optimal distribution, we can sample $\lambda * log_{\frac{1}{\alpha}} n$ blocks to bound the failure probability of catching the malicious blocks below $2^{-\lambda}$, where n is the length of the blockchain.

The Ultra Light Verification Protocol(ULVP) has the following three characteristics:

(1) **Security:** ULVP can ensure that the result of the predicate query is consistent with the honest majority's choice. An adversary can cheat the verifier with a negligible probability(less than $2^{-\lambda}$). It should be noted here that the security we are talking about is the security of the verification protocol itself, not the security of the consensus of the blockchain.

(2) **Succinctness:** ULVP can ensure that the network resources and local storage resources the verifier node need when acquiring the result of a predicate are sub-linear growth, that is, they do not increase significantly with the height of the chain. For example, the length of Ethereum blockchain is approximate $n = 2^{22}$, and suppose one attacker can hold around one third hash power of the total network, the proof size is below 400 KB with a failure probability of 2^{-50}, which is significant small compare to 5GB which SPV requires.

(3) **Non-interactive:** ULVP can ensure that the verification process is non-interactive, that is, the final result would not be obtained through multiple rounds of question and answer communication.

4.3.2 Smart Script with Combinable Trigger Conditions

When the lightweight local chain and cross-chain verification can be achieved, it is necessary to consider the specific interoperability execution steps. Any operation on the chain is triggered by certain pre-events. Once these specified pre-events are triggered, subsequent events can be executed in sequence. For multi-chain systems, the triggering mechanism of the operations on the chain is more complicated. We can summarize them into three categories:

(1) **The Status of the Chain:** for example, the length of the chain, whether there is a transaction in a block, the balance in an account,

the tokens in a smart contract, etc. This information can be obtained directly from all nodes of the chain through the chain verification protocol.

(2) **Other Non-chain Verifiable Information:** such as pre-image mapping of hash locks, transaction signatures, zero-knowledge proof functions, etc. This type of information can be independently verified locally by any node, generally some cryptographic evidence. This information can generally be obtained in webcasts.

(3) **On-chain Status on Another Chain:** for example, the length of another chain, the account status of another chain, etc. We need to obtain this kind of information through the cross-chain verification protocol mentioned above.

In the future, the blockchain smart script needs to provide such a function: it can set one or more trigger conditions. Once the trigger conditions are met (multiple trigger conditions can be combined according to the logic of AND and OR), the smart contract can be used on specific transactions that are published on the blockchain. The trigger condition here can be one of the three categories or a combination of the above three categories, and the subsequent transactions issued by the smart contract can be on local chains or on other designated chains. For example, the two-way pegged sidechain has the third trigger condition. The side chain needs to verify whether the balance of a specified account on the main chain is locked for a sufficient time (the third condition). Once the trigger is successful, the smart contract will allow specific accounts to generate a corresponding number of main chain tokens on the side chain. For the atomic-swap transaction scheme, its trigger condition is a combination of the first and second items. The transaction trigger on the chain requires that the account balance of the chain is sufficient, and it has a specific signature and a preset map or time of a certain hash lock.

4.4 Blockchain Node General Communication Protocol

The existing blockchain underlying node discovery and data transmission algorithms are not interoperable. Bitcoin nodes cannot directly find Ethereum nodes through the P2P protocol and they have to establish a TCP connection to realize it. However, MAP protocol requires all blockchain P2P networks can communicate with each other. Therefore, we need a unified and standardized P2P communication protocol. The Protocol Lab's LibP2P protocol just meets our needs. LibP2P is a basic module built for P2P networks. It highly abstracts the mainstream transmission protocol, so that the application layer does not need to worry about the specific bottom layer implementation, in which it has achieved the cross-environment and cross-protocol P2P node communication. Currently, Ethereum 2.0, Polkadot, and other projects have announced that they will use LibP2P as their underlying node communication algorithm. MAP protocol also chose the LibP2P algorithm as our node communication algorithm.

4.4.1 LibP2P Protocol

In the past, when developing internet applications, it was only necessary to focus on the upper layer logic of the application without reimplementing the underlying communication protocol (TCP/IP). The original intention of the LibP2P design is to support the future decentralized network protocol. The purpose of it is to allow developers to develop decentralized applications without having to pay attention to the specific implementation of the underlying layer. Finally, cross-environment and cross-protocol node communication are realized.

In a distributed peer-to-peer network, the relationship between nodes is no longer the traditional server-client model, which requires that each node can perform the role of the server to process responses but also act the role of the client to send requests. In this complex situation, we need a common communication protocol that can support multiple communication protocols to support the inter-communication between arbitrary nodes. The communication protocol needs to support traditional unencrypted TCP/IP communication as well as encrypted communication protocols such as TLS. The protocol needs to include both node discovery and the establishment of long and short connections, as well as a series of functions such as encrypted data transmission. LibP2P is a general protocol that meets all the above requirements.

In an ecosystem of multi-chain interoperability based on the MAP protocol, node discovery and communication in different chains will be involved. Therefore, a common communication protocol must be supported between all chain nodes, in which all nodes are in a large P2P network. At the same time, the particular nodes for different chains require to be under different subnet structures. Therefore, the network structure should be a multi-level, structured network topology. And LibP2P supports structured, unstructured, mixed, and centralized network topology, which also happens to meet our needs.

4.4.2 General ChainID Specifications

Under the regulation of MAP protocol, a unified set of ChainID rules must be formulated to identify different blockchains. Each chain will be assigned a unique ChainID. The function of this ChainID is similar to today's IP address and port in order to locate and identify a chain in the MAP ecosystem. When transmitting information that requires signatures, ChainID needs to be included in the signature data to prevent repeated transmission attacks. The distribution information of ChainID also needs to be updated in real-time on an underlying public chain so that other chains can obtain relevant information. The current solution is to deploy a smart script that manages the distribution information of ChainID on the MAP standard chain, and there will be a committee managing this smart script. The committee needs to review the public chain projects that apply to join this ecosystem, and the public chain that passes the review will be assigned a standardized ChainID and updated into the script. The script only has the function of assigning standardized ChainID, and does not have other centralized management functions. At the same time, public chains that are not assigned standardized ChainID can also communicate with other public chains within the protocol, however, their security cannot be guaranteed. In this case, we do not recommend interoperating with public chains that do not have standardized ChainID.

4.5 Block Transfer Protocol Based on IBLT Reversible Bloom Lookup Table

MAP protocol's block transmission protocol based on IBLT and the purpose of it is to solve the secondary transmission of transactions during the blockchain transmissions problem and improve the communication transmission efficiency of the entire system.

In the current blockchain design, transaction collection and block packaging transactions are asynchronous, and all transactions are first collected independently by each node and stored in the memory pool. Later, if a node successfully packages the transaction into the block, the block needs to be broadcast, and the transaction in the block will also be broadcast together. However, most of the transactions in this block already exist in the transaction pool of most nodes. This is the transaction secondary transmission problem in the blockchain transmission. In order to solve this problem, Bitcoin and Ethereum communities have proposed the Compact Block solution, that is, when transmitting blocks, only the block header and the hash summary list of transactions are transmitted. The receiving node compares it's own based on this hash summary list. If any transaction is missing in the transaction pool, it will request the sending node to send the complete content of the missing transaction again. Because the hash digest is much smaller than the original data of the transaction, it can save a lot of bandwidth. However, this summary list grows linearly with block size, so it still consumes network resources when facing large blocks.

The new data structure Invertible Bloom Lookup Tables (IBLT) proposed by Goodrich[3] can further optimize the transmission protocol of the block. This data structure is generally used for set reconciliation. This data structure can ensure that the amount of data required by the two communication parties to obtain each other's set is only related to the difference between the two sets and not to the total amount of the two sets. For example, suppose the block packaged by node A contains 10,000 transactions, and the transaction pool of node B also has 10,000 transactions, and only 100 transactions in the block are not in the transaction pool of node B. Under the transmission protocol or the Compact Block protocol, node A needs to send data with a size that is positively related to 10,000 transactions (whether it is 10,000 transactions or 10,000 transaction summaries). However, if we use MAP's transfer protocol, we can compress the 10,000 transactions of node A into an IBLT equivalent to about 100 transactions of data volume and send it to node B. Node B can use the IBLT and its own transaction pool extract the entire transaction list and obtain the complete information of the entire block. Such a scheme can greatly improve the transmission efficiency of the block and enhance the scalability of the entire system.

5 MAP Blockchains

We will develop a set of public chains that meet the MAP protocol, including MAP Standard Chain and MAP electronic cash payment protocol. The MAP Standard Chain is to provide an anchor point for various types of blockchain and cross-chains in the early stage of the ecosystem, but this anchoring is not mandatory, and any public chain can provide security with the MAP Standard Chain together in MAP's ecosystem. The MAP electronic cash payment protocol is a high-performance blockchain equipped with SMART intelligent scripting system. It uses the DPOS consensus mechanism and can interoperate with the MAP Standard Chain. The goal is to provide electronic cash payment services based on blockchain technology service.

5.1 MAP Blockchains Consensus

MAP Standard Chain consensus will use POS Consensus based on Verifiable Random Function(VRF)[5] design, and MAP electronic cash payment chain will use DPOS consensus for scalability and high throughput purpose.

5.1.1 POS Consensus Based on VRF

Blockchain consensus algorithms are generally divided into two categories. The first category can be independently verified, such as POW and POS. The characteristic of this type of algorithm is that the rules for selecting blocks are based on cryptography. Other nodes in the cluster can independently verify blocks' validity without additional information. On the contrary, the other type of consensus cannot be independently verified, such as DPoS, DPoW (Delegated Proof of Work). This type of consensus algorithm needs to select a small set of nodes from a large set of nodes as delegates, and then the small set of nodes will utilize Byzantine Fault-tolerant Algorithms for blocks to reach consensus. All other nodes must additionally obtain the consensus information from the small set of nodes to verify the validity of the block. The advantages of the first type of independently verifiable consensus algorithms are the high degree of decentralization and security, but the performance of the first type of algorithms is relatively poor. The second type of non-independently verifiable consensus algorithm has the advantage of high performance, but the security and decentralization are relatively weak due to the selection process of the small set and the vulnerability of the small set itself.

Our MAP standard chain needs to decouple consensus and state transition, and provide some ecologically shared data for other chains to use. Therefore, the demand for the security and decentralization of the consensus algorithm will be higher than the demand for scalability. Therefore, our consensus algorithm will be a consensus algorithm that can be independently verified. We have two initial choices, PoW and PoS. However, after comparison and consideration, we decided to choose the PoS instead of the PoW as the consensus engine of the MAP protocol. The main reasons are as follows:

(1) PoW requires the support of computing power to maintain its security. However, at present, a large amount of computing power is concentrated on Bitcoin. If the new PoW public chain cannot obtain enough computing power, it cannot resist 51% attacks.

(2) PoW-type consensus will perform a large number of hash calculations to obtain the final consensus, which will waste a massive amount of energy (except for ensuring the safety of the public chain, there are no other benefits of those energy consumption)

(3) The PoW consensus also heavily depends on hardware equipment. The replacement of mining machines will lead to unfair competition among n-odes, which will cause more centralization. This has already been proved by some monopoly of mining pools.

In the early design of the PoS consensus algorithm, it was simply to calculate mining probability based on currency holding time and currency holding volume as the weight. However, the blockchains based on this PoS consensus would be vulnerable under "long-range attack" or "nothing at stake attack". MAP standard chain would use an original POS consensus which is robust with these two attacks. We call it MAP-POS. which is designed based on Verifiable Random Function (VRF)[5].

We first introduce VRF. VRF is a pseudo-random function that provides publicly verifiable proofs of its outputs' correctness. VRF has a triple of algorithms Keygen, Evaluate, and Verify. Keygen would take a random number as input and generate a pair of keys (private key SK and public key PK). Evaluate will take private key and message(M) as input and output a pseudo-random output number(Y) and a zero-knowledge proof ρ. Verify would take PK, M, Y, and ρ as input and return true if all information matches. Thus we can use Verify to check that the pseudo-random number is indeed created by the owner of the public key.

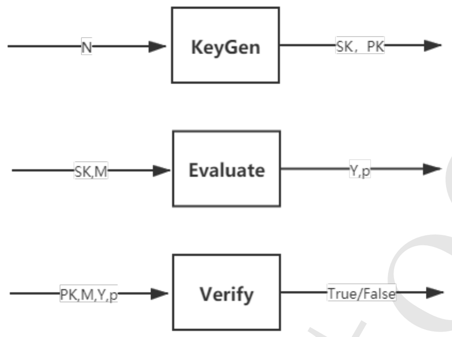

Figure 3: VRF's triple of algorithms

In MAP-POS consensus, the block is generated in epoch-based. Each epoch would be divided into several slots. Each slot would be assigned to some candidate nodes for block generation(the number of candidates is not deterministic).

For each epoch, the first block, called genesis block, was derived by each node locally based on the information from the last epoch. It has two essential parts: the stake distribution and an initial random seed(X). The stake distribution, which records the public keys of all candidates and its stake portion, would be stored in a Merkle tree. The initial random seed(X) is derived from the pseudo-random numbers of blocks from the last epoch. We will elaborate on this process later.

And the rest blocks of the epoch is a normal block. These blocks are generated by some candidates by the following rules: Each candidate uses the initial random seed(X) and slot number generates a message(X(i)). And then he would use this message(X(i)) and his private key(Sk(i)) as an input of the evaluate function, and it would output a pseudo-random number Y(i) and a zero-knowledge proof $\rho(i)$. The candidate thus can check the qualification for the block generation of this slot: the candidate is qualified if the pseudo-random number Y(i) is smaller than the production of his stake portion S(i) and a predefined threshold T. The candidate only broadcast the block in the slot which he is qualified for. The block is constructed by three parts: the block header, transaction list, and witness. The block header would

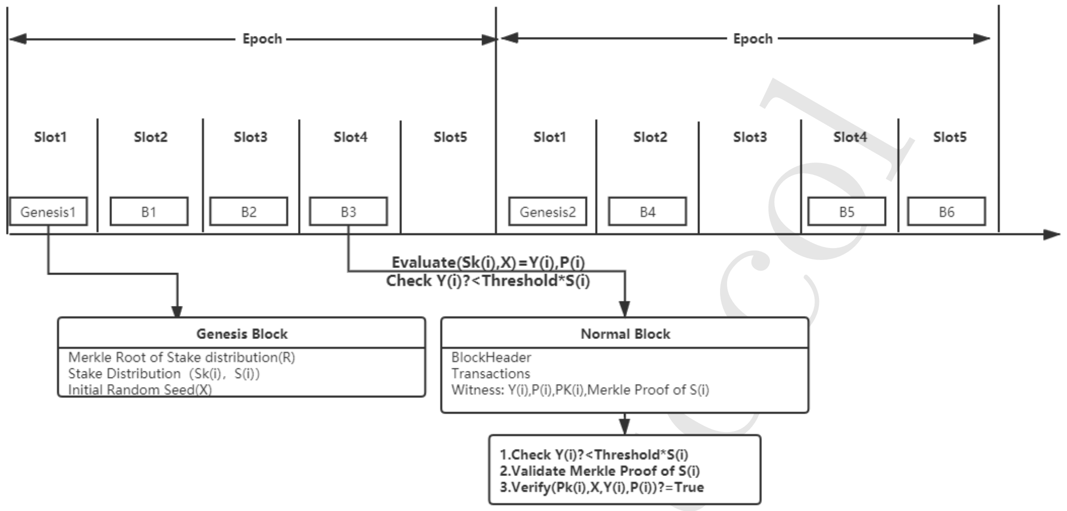
Epoch
Epoch
Slot1
Slot2
Slot3
Slot4
Slot5
Slot1
Slot2
Slot3
Slot4
Slot5
Genesis1
B1
B2
B3
Genesis2
B4
B5
B6
Evaluate(Sk(i),X)=Y(i),P(i)
Check Y(i)?<Threshold*S(i)
Genesis Block
Merkle Root of Stake distribution(R)
Stake Distribution（Sk(i)，S(i)）
Initial Random Seed(X)
Normal Block
BlockHeader
Transactions
Witness: Y(i),P(i),PK(i),Merkle Proof of S(i)
1.Check Y(i)?<Threshold*S(i)
2.Validate Merkle Proof of S(i)
3.Verify(Pk(i),X,Y(i),P(i))?=True

Figure 4: MAP-POS consensus design

contain some core information of the block, such as parent hash, slot number, block height, root of the transaction tree, root of the state tree, etc. The transaction list contained all the transaction which be included in this block. The witness part would include the pseudo-random number Y(i), the zero-knowledge proof $\rho(i)$, the public key(Pk(i)) of the block creator and a Merkle proof of the stake portion(S(i)) of the block creator. Any nodes can verify the validity of this block by the following process: first, he need check the pseudo-random number Y(i) in withness is indeed below the production of the stake portion S(i) and the predefined threshold T. Then he checks that if the Merkle proof of the stake portion(S(i)) is valid. Finally, he would use the verify function of VRF to check that public key Pk(i), message X(i), pseudo-random number Y(i), and zero-knowledge proof $\rho(i)$ are consistent. All the information he needed for checking is included in the genesis block and this normal block, thus this algorithm is individually verifiable.

MAP-PoS consensus protocol has the following characteristics:

(1) There is no miner monopoly. To return the basis of the original intention of Satoshi Nakamoto's PoW consensus design, in which to ensure that all nodes participating in the consensus have a fair go.

(2) Regardless of the number of nodes participating in the consensus through-

out the network, the amount of calculation required would not be too high as to prevent waste of resources like PoW.

(3) The protocol is self-verifiable. It means that each participant can check the validity of any block individually to ensure that the block miner has spent a certain amount of stake uniquely for this block. This protocol is compatible with MAP protocol.

5.1.2 Mixed consensus of DPOS and PBFT

The requirement of MAP electronic cash payment is to provide electronic payment services based on blockchain technology. On top of it, there is a high-performance SMART intelligent script system that can develop smart contracts and DApps that meet the electronic cash application requirement. Therefore, in order not to let consensus become the bottleneck of the entire blockchain system, we decided to choose a non-self-verifiable type of consensus algorithm to provide scalability and performance. We chose a mixed consensus algorithm of DPOS and PBFT.

(1) First, we define a certain time interval as an Epoch. For each Epoch, we will select the next Epoch's incumbent committee through proof of stake. All users can obtain voting rights by pledge tokens to choose their trusted nodes.

(2) In the elected Epoch, the committee produces the blocks turn by turn through the PBFT algorithm and publish them to the entire network through broadcasting.

(3) In addition, we will design a punishment mechanism called slashing to deal with the issues of the individual or collective corruption of elected committee members

5.2 SMART SYSTEM

SMART (Sustainable MAP RunTime) is a key component of the MAP protocol. It can build a standard public chain in modularity while providing a smart contract platform for electronic cash payment and other application chains. SMART includes the following components

(1) **MAP-VM:** a trusty-worthy WebAssembly virtual machine

(2) **Delta language:** Smart contract language suitable for SMART development;

(3) **Runtime:** Runtime environment based on MAP VM.

Based on SMART implementation, MAP can provide the scalability of the interactive chain and introduce other assets on the chain. The payment system built on SMART has tens of thousands of TPS throughput in the real network environment, and the confirmation time can be reduced to 2-3 seconds.

5.2.1 MAP-VM

MAP-VM is a virtual machine running contract code in SMART. MAP-VM is a high-performance virtual machine based on WebAssembly. WebAssembly is a new generation code format that can be run directly in the browser. Compared with the EVM used by Ethereum, MAP-VM can run code at a speed close to the native computer hardware speed, while also providing a more flexible contract interface.

Compared with V8, Chakra, Spidermonkey, and other execution engines, MAP-VM is designed for consensus execution and enhances various features to adapt to contract code execution. Since the WebAssembly open standard is still developing, many new features are being added to the standard, and MAP-VM uses a subset of WebAssembly.

In addition to implementing the WebAssembly standard specification, MAP-VM provides the following enhanced features:

For soft floating-point arithmetic, the operation of the contract depends on the execution environment of a deterministic operation. MAP-VM uses soft floating-point arithmetic. Compared with the floating-point arithmetic of the local CPU instruction set, it can provide absolute consistency.

Deterministic execution, the contract code has a definite and consistent running result in different environments, and all abnormal call and operation errors are accurately handled. Compared with v8, MAP-VM, Spidermonkey's highly optimized JIT engine can guarantee the absolute consistency of the calculation process and calculation results.

Efficient measurement. In MAP protocol's consensus, both computing and storage are scarce resources. MAP-VM uses the gas mechanism similar to Ethereum. It can accurately calculate the computing resource consumption of the code while the wasm is executed, without affecting the overall performance.

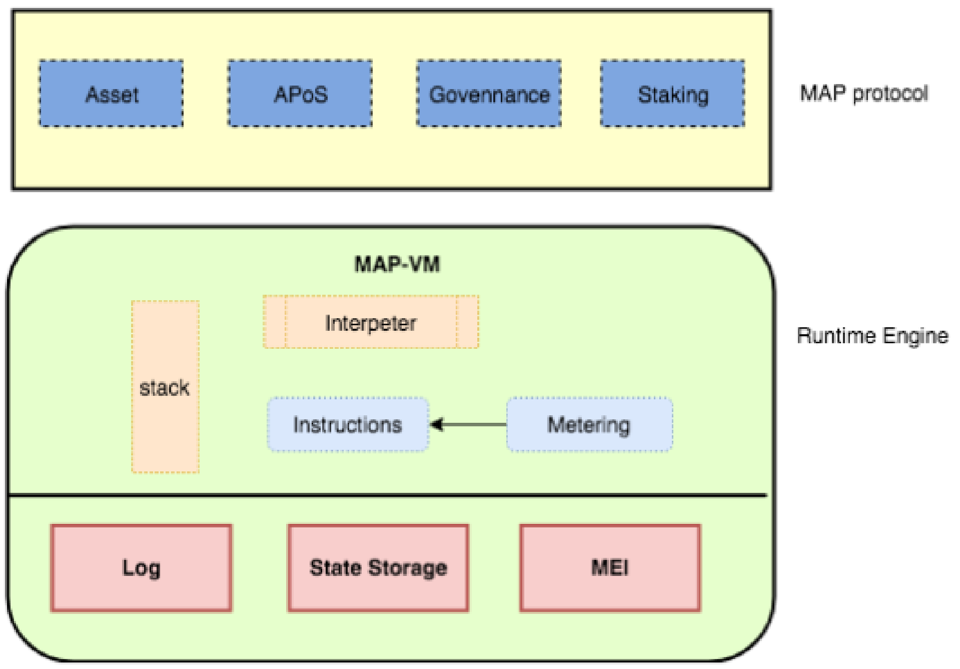

Figure 5: SMART SYSTEM

MAP-VM is a stack-based virtual machine that can provide function library export and import custom function interfaces like JavaScript. MAP-VM will import a standard set of host functions, defined as MEI (MAP Environment Interface). MEI provides basic APIs to support the operation of wasm, including algorithm signature, digest operation, block data, state storage, and other operations.

5.2.2 Delta Language

In SMART, Delta is used to implement MAP's runtime code. Delta contract language is a JavaScript-like programming language. Delta's language features are specifically designed for contract platforms, and the ABI interface is closer to WebAssembly. Unlike Solidity on Ethereum, the Delta language supports more native base types, which can provide higher performance. The Delta contract code is directly compiled into the wasm format. Compared with the EVM contract compiled by Solidity, the Delta code has a ten-fold improvement in terms of calculation performance, providing greater throughput for transaction processing.

The Delta language is based on static typing and is syntactically similar to JavaScript. In order to adapt to the contract operating environment, Delta has a built-in State statement and Event function, which can directly link the MAP ABI defined by SMART.

The most important improvement in the Delta language is support for the global state. In Solidity, you need to define variables and mapping and other modifiers in the contract to declare the storage structure in the global state. Delta will use native types to support storage type declarations providing safer type operations for states. Contract developers can more intuitively define operations on global resources and design contract security codes.

5.2.3 Runtime

In the SMART architecture, the runtime protocol can be written and implemented in Rust or Delta language and can be compiled into the underlying wasm blob by the compiler. MAP-VM loads the compiled wasm code and waits for transaction execution after initializing the global environment. The runtime code running on MAP-VM can export a series of application interfaces to providing functions such as consensus protocol, multi-currency payment, and community voting. MAP protocol defines the following basic agreements:

(1) staking

(2) asset

(3) government

In SMART, a series of MABI calls are defined as messages. The messages in MAP-VM are different from the transactions in Ethereum. In Ethereum, account addresses are explicitly divided into external accounts and contract accounts. When an ordinary transfer is initiated, the sender must specify the Value field of the amount, and if it is a contract call, the Input field of the contract must be specified. In MABI calls, all transactions use a unified message format, which is eventually decoded into WebAssembly code for execution, no distinction between transfer calls and contract calls.

In addition to the current state of the account, MAP protocol will separately store the execution process of the transaction and save the data in the Log. Log supports quick query for data indexing. When the user wants to determine that a transaction is executed correctly and obtains the expected state, he can query the transaction log according to the index through the transaction hash to extract the execution data from the stored data.

The transaction in MAP protocol is a general signed message data. Unlike Ethereum, the transaction structure in MAP protocol does not have a separate value field for the transfer amount but instead uses a similar system contract to provide the transfer function. When the user initiates a transfer, the sender needs to use a signature package contract to call the data, and then send it to the p2p network. The sender will not get any return value, but a unique transaction hash. After the transaction is packaged by consensus and executed by SMART, users can query the account status after the transaction.

A transaction in MAP protocol can be represented by a state transfer function:

$$S' = SMART - Exec(account, func, input, S). \tag{1}$$

S is the state of the account before the transaction, the input is encoded contract call data, func is the signature of the function called by the contract, and account is the sender can support multiple signature formats.

MAP prtocol can get the sender information from the signing data, and after decoding the transaction data to get the function signature and input data parameter. After the transaction data is verified and sorted, it is executed in MAP-VM to change the user's status. MAP's account system is different from Bitcoin and Ethereum. MAP's account is not the public key hash of a certain signature algorithm, but an account ID that supports multiple signature algorithm formats. Thanks to the account format of the multi-signature algorithm, MAP can natively support a variety of crypto assets with different signatures to realize multi-currency payment scenarios.

MAP-VM will accumulate gas consumption during the execution of transactions. VM has built-in WebAssembly measurement, which can accumulate all instruction calculations, stack occupation, and resource consumption of state storage. When insufficient gas and execution errors occur, the VM will deduct the gas consumption and return the state to the account state before the transaction.

In Bitcoin and Ethereum, when the community wants to modify the consensus protocol, it needs to provide an incompatible version. After the miner agrees and adopts it, a hard fork is initiated at a specific height. If the community disagrees with the hard fork, users will have to face governance issues such as community splits and replay attacks. Benefiting from SMART architecture design and on-chain governance, each running node can initiate a new runtime proposal without a hard fork, and after the vote is passed, the protocol consensus can be upgraded on the designated height chain.

5.3 The Economic Model of the Token of MAP protocol

5.3.1 Incentive Distribution Plan for The Initial Circulation

The initial issuance of MAP protocol is 10 billion tokens, which would be allocated according to the following distribution ratio:

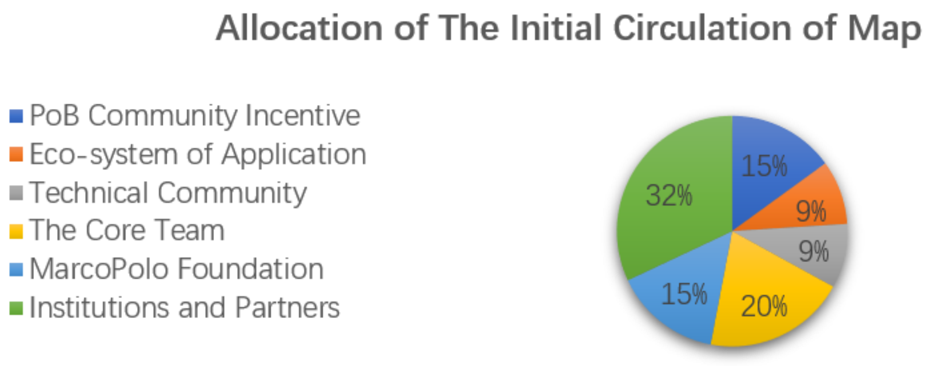

Figure 6: Allocation of The Initial Circulation of MAP protocol

5.3.2 MAP Standard Chain Consensus Incentive

After the launch of MAP Standard Chain, all kinds of network nodes will be awarded with the block generation reward according to the consensus mechanism. The amount of his awards is correlated with the amount of staking in the whole network.

5.3.3 MAP protocol Economic Model

MAP is the native token issued by the Map protocol. The initial total amount is 10 billion. With the following application scenarios to help to improve the intrinsic value of MAP protocol as well as promote the development of the MAP protocol.

(1) The public chain or consortium chain using the MAP protocol requires the MAP protocol to allocate the inter-chain network ID. Each inter-chain network ID is unique under the inter-chain network system. ID adopts the form of pledged MAP Token to obtain an ID. As more and more public chains use the MAP protocol for inter-chain, the market will have more and more demands for ID and more and more demand for MAP protocol. Pledge MAP Token to obtain ID on the MAP standard chain;

(2) The governance of the MAP protocol is carried out on the MAP standard chain;

Table 1: Table of Initial Circulation of MAP

Ratio	Holders	Purpose and Governance
9%	Technical Community	To motivate developers who contribute to the underlying technical development of MAP protocol.
9%	Eco-system of Application	Incentive for public chains and DAPPs to use MAP protocol.
15%	PoB Community Incentive	To motivate members in the community to contribute to the expansion and consolidation of the Consensus of MAP protocol.
15%	MAP Foundation	For project operation and business expansion, managed by the MAP Foundation.
20%	The Core Team	Incentive for the core team, this part won't be unlocked for the first two years.
32%	Institutions and Partners	Investment institutions and partners who made contributions to MAP protocol.

(3) Reward MAP tokens as incentives for using MAP public chains and DAPPs. Also for development incentives for technical contributions to the MAP protocol including improve MAP protocol, build Dapp and expanding technical communities, etc.

(4) The Map standard chain uses a POS mechanism, and nodes need to pledge MAP Token to obtain mining rights.

(5) MAP Token is used to motivate the user community and expand the user base.

(6) MAP token as the GAS of the MAP. The POS miners will obtain gas in MAP for every transaction.

Under the premise of developing and expanding the MAP protocol, as the project develops and evolves, the above scenarios will be increased or reduced.

6 Application Scenario

6.1 Defi and Dex

Defi, especially Dex, is one important application scenario of MAP protocol.Defi project focuses on making full use of the low trust cost of blockchain peer-to-peer transactions, removing third-party guarantee agencies, and directly providing cryptocurrency lending/transaction services for both parties. It is currently one of the most valuable areas of blockchain technology. Although there are more and more DeFi projects, people are still not sure what DeFi is. On the one hand, Defi needs further publicity and promotion among users. On the other hand, Defi is not safe and efficient enough compared to Cefi. MAP solves these two problems to a certain extent. First of all, MAP will be connected with many blockchain projects, which will have a certain promotional effect on Defi. In addition, the essence of Defi can be seen as inter-chain calculation, and MAP provides a powerful chain interoperation function. Through the chain interoperation, different digit assets could be exchange directly. It can make the decentralized exchange safer and more efficient.

6.2 IPFS

MAP protocol also can provide support for IPFS system.IPFS is a hypermedia transmission protocol based on content addressing, versioning, and peer-to-peer. It allows participants in the network to store, request, and transmit verifiable data to each other IPFS can create a more open, fast and secure Internet. The application of IPFS can be realized on the MAP protocol. IPFS is built on the LIBP2P protocol, and so is our MAP protocol. This means MAP protocol is compatible with IPFS system. IPFS system could be used as the layer 2 of MAP protocol. In this layer, Dapp could be deployed and data could be stored. Based on the support of the MAP protocol to the IPFS system, they can obtain more open storage space, lower storage costs, faster file search and more efficient file transfer. And, based on the characteristics of the MAP protocol itself, this layer could interact with MAP protocol to achieve interoperability with other blockchain system.

6.3 Privacy computing

MAP protocol can provide users with private computing capabilities. Symmetrical encryption techniques are used to realize distributed encrypted storage of user data. Through the blockchain interoperability and smart script of MAP protocol, we can realize authorized access to encrypted data, and the privacy transmission function can be realized directly. Combined with the hardware support of TEE, we can realize the function of private computing. Privacy computing on MAP protocol has a wide range of applications. With the development of blockchain technology, more and more data are online. The privacy computing function of MAP protocol provides a privacy guarantee for data interaction on the chain. In the medical scene, the privacy computing function can deal with the hospital encrypted data, which is one of the important technologies to realize intelligent medical treatment. In the financial scene, the privacy computing function can collect and process the encrypted data authorized by the user.

References

[1] Aggelos Kiayias, Nikolaos Lamprou, and Aikaterini-Panagiota Stouka. Proofs of proofs-of-work with sublinear complexity. International Con-

ference on Financial Cryptography and Data Security, pages 61-78. Springer, 2016.

[2] Aggelos Kiayias, Andrew Miller, and Dionysis Zindros. Non-interactive proofs of proof-of-work, 2017.

[3] Goodrich, M., Mitzenmacher, M.: Invertible bloom lookup tables. In: Conf. on Comm., Control, and Computing. pp. 792-799 (Sept 2011)

[4] Benedikt Bünz, Lucianna Kiffer, Loi Luu, and Mahdi Zamani. FlyClient: Super-Light Clients for Cryptocurrencies,2019

[5] Micali Silvio, Rabin Michael O, Vadhan Salil P. Verifiable random functions. Proceedings of the 40th IEEE Symposium on Foundations of Computer Science, 120-130, 1999

[6] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.