Technical White Paper of

MOAC – Mother of All Chains

June 8th, 2017

[Abstract]

MOAC is to design a scalable and resilient Blockchain that supports transactions, data access, control flow in a layered structure. It creates the framework to allow users to execute Smart Contract in an efficient way. It also provides the architecture to spawn sub blockchains using underlying infrastructure quickly and easily. It is a Blockchain platform with necessary plumbing parts available to sub blockchains, providing solution for idea test, private chain deployment, complex task processing, and Decentralized applications, etc.

[Background]

Blockchain technology has been evolving very quickly since the introduction of Bitcoin in 2008. Over the past nine years, many Blockchain techniques have been explored to try out various ideas, in the hope to expand the Blockchain usage, performance and applications.

At the same time, the token value of Blockchain plays a vital role in the adoption of the blockchains. It helps to incentive more participants to join the ecosystem, also acts extremely useful to augment the existing payment system in a more efficient way. However, as Blockchain in its still early stage, it suffers from couples of problem. .

1. Difficult to try new idea

   New idea means a new Blockchain. It requires extensive overhead to implement a new Blockchain idea, by setting up servers, develop teams, establishing community, attracting new users, etc.

2. Difficult to upgrade

   Once Blockchain has been deployed and in production, it is very difficult to add/modify/delete features. Any of those is either soft fork or hard fork. Either fork requires tremendous effort and economic consequences.

3. Incompatible among chains

Different chains have different schemes, such as consensus protocol, currency features, and adoption requirements. These schemes prevent the interconnection or exchange among multiple chains.

4. Split participant group

   For each Blockchain, the user base is different. Mining rigs and validators are dedicated for that chain only. No two blockchains can share any of them.

Also, as most cryptocurrencies' value increases dramatically in the last year, applications based on existing blockchain platforms suffer from higher fees and longer latencies.

[MOAC solution]

Overview

Over the past several years, multiple consensus protocols have been adopted and tried out including POW, POS, BFT, and hybrid of those. However, not a single protocol can solve all problems. Typically, POW can be deployed in a large scale network and scales very well. It is the most verified consensus protocol among all. But it suffers from problems like large power consumption, low throughput, high latency, and higher barrier to participate. POS (DPOS) is better on power consumption and could be configured to perform faster. However, the protocol is complex to implement, and economical consequence has not been fully tested in large scale. And it typically deploys in a smaller network scale. BFT family is normally used in a much smaller scenario and can perform much better in term of throughput and latency. So its usage is mostly focused on private chain or enterprise applications.

To be able to deploy in a large network scale and benefit from more participants and decentralization, while keeping the high throughput and low latency, MOAC introduces the layered consensus stacks to make above goals practical. It is the Blockchain for blockchains. MOAC itself will be deployed in public network with large number of validators. It provides following:

1. Layered configuration structure
2. Transaction, Smart Contract and Data Access support
3. Data flow, control flow and processing units, to form a distributed Von Neumann architecture.
4. Validators could be configured to support multiple overlapping sub blockchains.
5. Pluggable validating scheme to support injection of user defined protocols, make it easy to deploy new sub blockchains using existing validators.
6. Encourage user with smaller processing power to participate in the validation process.

Consensus Protocol

We realize simply extend any current consensus protocol won't be able to meet the all requirements. Typical solution of combining two types of different consensus protocols results in multiple chains or side chains. This is the approach we want to avoid, as it introduces more problems than solve them. Our solution to the consensus dilemma is to build a layered consensus stack, but to keep everything synced in a single Blockchain.

For the underlying layer, we utilize POW as the main consensus protocol, because POW is a widely tested protocol and most robust solution to a large scale network setup. Currently MOAC uses POW similar to Ethereum.

The drawback of POW is compensated in the top layer. Only critical transactions and control flow transactions are processed in the POW layer. The top layer adopts POS consensus protocol with sharding technique to provide faster and higher throughput solution.

Each POW node has one Smart Contract Server node. The Smart Contract Server (SCS) identity is fully verifiable by the corresponding POW node. Each SCS node will present holding stake to be able to process the top layer user requests.

Note that SCS processes Smart Contract calls. All transaction in the top layer is in form of Smart Contract calls. Not all SCS will perform the processing single transaction at the same time. Rather, part of selected SCS will process specific transaction.

The selection of SCS is through initialization of Smart Contract call or Flush call. The init/flush call is actually passed to the POW and achieves consensus in underlying layer. The init/flush call will include the selection criteria including percentage of processing nodes. Then each POW will invoke that call on its SCS with an EHDRand algorithm. SCS will determine if itself is selected to process this Smart Contract. Note this is a deterministic process and SCS participation can be verified by anyone.

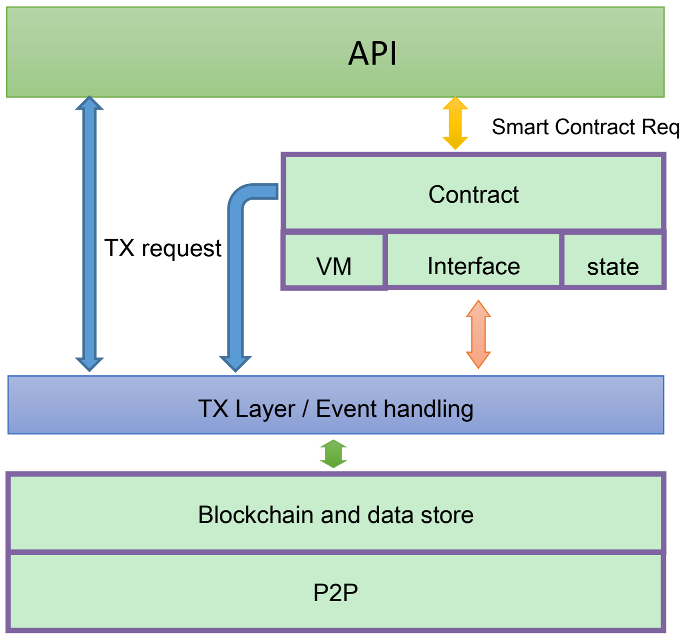
Once group of SCS is selected for certain Smart Contract, they will communicate with each other for form a small consensus group. This group will process the following Smart Contract calls on that Smart Contract. Also, the behavior how they agree each other could be specified by the init/flush call. Effectively these SCS nodes form a sub Blockchain and perform the consensus based on the predefined protocol or user defined protocol. Please note the consensus protocol is different from the actual Smart Contract code.

Smart Contract state is saved in each SCS. However, that is not on the actual Blockchain. To achieve the benefit of Blockchain, the state needs to flush into the underlying Blockchain periodically or on demand.

When flushing, in the consensus mode, SCS node will initiate a data store request on the underlying POW nodes. The current state will be written into the Blockchain and referenceable by a HASH. Note all POW node will do the same operation. For those nodes who are not part of the consensus group, they will not do anything. The participating SCS will get the proposed state and verify it with its own state. If it can prove that the proposed state is incorrect, it will initiate another data store request with corrected state and referencing the incorrect state hash. If no rebuttal data store request is initiated for predefined rounds, SCS node will initiate the final flush Smart Contract call with the correct state hash. The involved transactions in the correct state will be processed at every POW node. The SCS node who sends out the incorrect state will forfeit its stake.

In MOAC, most transactions will be processed in the top layer, while only small portion of control flow transactions are processed in the POW layer. This is feasible because top layer provides fast, flexible and low cost service, while POW layer provides slow, reliant and expensive service.

Layered Architecture

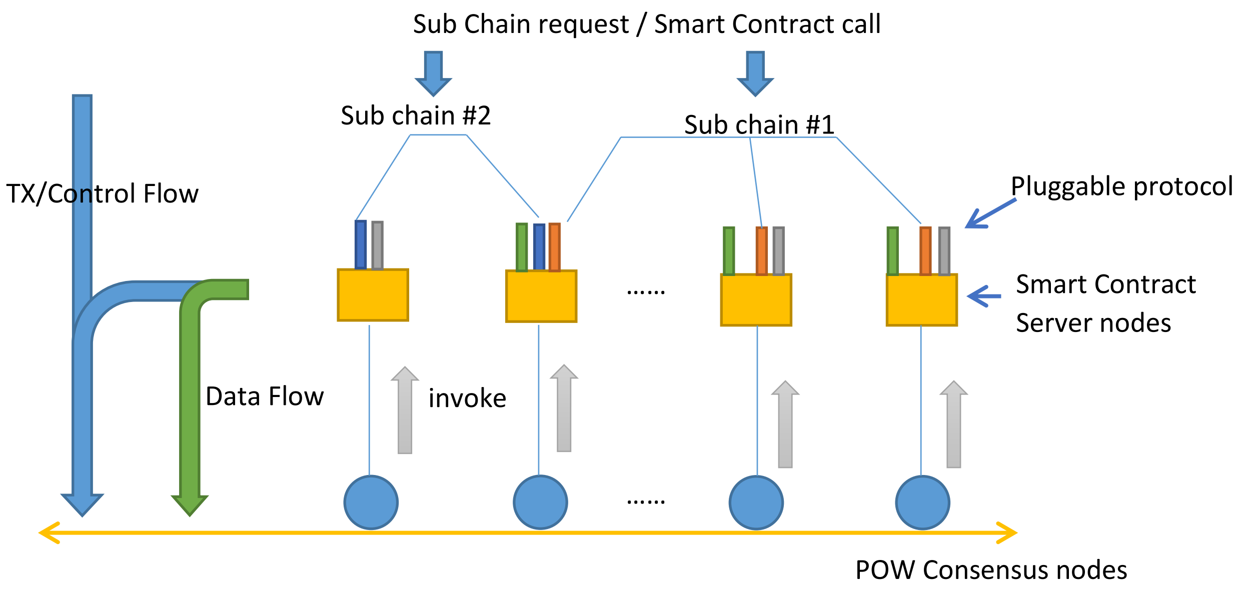


1. P2P network layer. This layer defines p2p protocol.
2. Blockchain layer. This layer handle all operation related to Blockchain operation, like consensus, data access, etc.
3. TX layer. This layer handles TX request and reply. It also processes of the control TX request and if necessary, invokes Smart Contract related operations.
4. Smart Contract layer. This layer performs smart contract execution inside virtual machine and also keeps a temporary contract state.
5. API handles end-user input and gets the output from lower layers.

MOAC topology

The POW consensus node participates in a volunteer way. Each node contributes its computing power to solve the computation-intensive problem and verifies the validity of transactions in the agreed transaction set.

Besides the POW consensus on transaction and data store set, each POW node is associated with one Smart Contract Server. SCS node could be local to the POW node, or it could be the remote node. The Smart Contract Server (SCS) identity is fully verifiable by the corresponding POW node.



Smart contract request (create/invoke/flush) is enclosed in the Control flow TX_c and is first processed in underlying layer. Then each POW node sends the contract request to its SCS via an asynchronous call. SCS will send additional Control flow TX_c and Data Store TX_s to underlying layer if needed.

Execution of smart contract is in an efficient sharded way. All the SCS can be configured at run time to process different sections of smart contracts. The whole system throughput could be 10x-100x faster than that of traditional way. The sharded execution group marshals the sharding state into underlying Blockchain through Control flow TX_c and Data Store TX_s.

Wallet/Address

Wallet and address are interchangeable in this document. Address holds the balance of the digital currency. Each address has its corresponding secret key. Secret key is used to sign the TX originated by this address.

Smart Contract

Smart contract is same as normal address. It has a unique public address. One difference is that contract secret is discarded at creation. So no one can distribute the balance of a contract to anyone else, other than the consensus protocol.

Contract holds four basic elements: {code, state, [call], balance}. Code is generated by user. State holds current internal information. Balance is the digital currency the contract has. It also stores history of function calls on that contract.

Transaction

Transaction is the basic operation inside the system. Each address can send/receive balance to other address. And there are also Smart Contract based TXs. These TXs are used to trigger the work flow of Smart Contract.

Three basic transaction types in the system: payment transaction TX_p, Data Store TX_s, Control flow TX_c. They are processed in underlying POW consensus nodes. All nodes agree on the same world state.

a) Payment TX: {sender->receiver:amount}

   Basic transaction to move fund from one to another. Sender will need to sign the transaction using secret key. The signature is verifiable by anyone.

b) Contract control TX

   1) Contract init TX {code, sender, init_amount, execution type, sharding config}

      User sends the init TX to start a new contract. In the contract, user will need to specify the contract code, init fund, execution type: fast or normal, sharding config.

   2) Contract Flush TX {contract_address, flush_target_state, flush_steps}

      Flush TX is to allow POW node agree on the already executed bulk transactions and flush them into the Blockchain.

   3) Contract Payment TX {sender->contract_address: amount}

      This is similar to normal payment TX, except that POW node will notify contract server about the balance update and who made the contribution.

c) Data store TX {sender->contract_address: data to store}

   This transaction type processed in POW node will not validate any balance related operation.

Sub Blockchain

MOAC system can perform regular payment transactions, data store transactions and Smart Contract transactions. Moreover, it is very convenient to utilize the provided architecture to spawn sub blockchains.

User can configure sub chain using Smart Contract to define sub chain properties (% of participant nodes, consensus protocol, policy, state storage, etc). The creation of sub chain is done through Control flow TX_c. Once sub chain is established, each participant SCS will adopt the pluggable protocol in its execution. Any following requests on the sub chain will be validated by the selected % of SCS.

The block generation of the sub chain is configured to either on-demand or on set time schedule. The on-demand feature is preferred, as it only generate blocks when needed, thus saving valuable resources.

The sub chain deployment can be as easy as sending couples of Smart Contract calls. However, it inherits the secure and robust underlying Blockchain properties. And it can reuse the large pool of existing validators and benefit from the decentralized setup.

The sub chain could utilize Flush contract call to randomly reselect SCS node, to achieve better decentralization and security.

Upgrade sub chain is also easy by just redeploying to a new set of SCS with updated chain property.

Transaction Fee

There are two types of payments that nodes can get from contributing their computational power. Firstly, the POW nodes will get rewarded for each block they mine. This is similar to what currently BITCOIN does. Secondly, the SCS server can be rewarded for their participation of sub chains and their processing work of Smart contracts. Note that this kind of service may not be power-intensive. For example, if a sub chain is based on POS, the SCS can just spend very limited resource for the validation.

This is a big incentive to regular PC users or even mobile users. For the pure POW network, there is almost no chance for regular user to benefit from mining. However, in MOAC setup, user can setup a light POW node with almost no chance to win in mining contest, but, he can setup an SCS associated with that POW node, and gets rewarded for the SCS works it provides. This will encourage more users to join the consensus system and provide more SCS processing power. On the other hand, the Smart Cotnract owner or sub chain creator will need to pay the fee for all SCS working, but is very cost-effective considering the benefit and low startup costs. The whole process will promote a more distributed ecosystem and benefit all parties.

Payment schedule

Block is mined every 10s, with reward of 2 MOAC coin per block. The reward schedule halves every 3,000,000 blocks, equivalent to approx. one year. After block 18,000,000, the reward will be constant of 0.04 MOAC per block. See below. We define 1 MOAC = 1,000,000 Sand. 1 Sand = 1,000 Xiao.

Block#	Reward (1 MOAC = 1,000,000 Sand)
1-3,000,000	2 MOAC
3,000,001-6,000,000	1 MOAC
6,000,001-12,000,000	0.5 MOAC

12,000,001-15,000,000	0.25 MOAC
15,000,001-18,000,000	0.125 MOAC
18,000,001-	0.1 MOAC

Transaction fee is paid in two ways. One is through Transaction. The other is for Smart Contract or sub chain.

Transaction Type	Fee	Pay to
Payment TX_p	20 Sand	POW miner
Data Store TX_s	20 Sand	POW miner
Control flow TX_c	50 Sand	POW miner
Smart Contract Call	1 Xiao	To each SCS

Smart Contract Call cost is set lower than underlying transaction in purpose, thus encouraging the usage of SCS. This can alleviate the pressure on the underlying layer, and also benefit the SCS providers.

[Summary]

To summarize, MOAC utilizes layered architecture to combine the POW feature of robust and scalability and the POS feature of fast performance, quick response, without their shortcomings. The Smart Contract layer forms a platform for complex task and various sub blockchains. The POW nodes together with SCS nodes, construct the flexible and scalable framework that can be reused across many applications. MOAC Blockchain could be valuable to both light customers and compute intensive customers.

[Reference]

MOAC coin total amount increases over years:

ICO	250,000,000
1st year	256,000,000 (approximate)
2nd year	259,000,000 (approximate)
3rd year	260,500,000 (approximate)
4th year	261,250,000 (approximate)
5th year	261,625,000 (approximate)
6th year and beyond	261,625,000 + 300,000 * n

[DISCLAIMER]