Vreath

A New Consensus Algorithm Makes CryptoCurrency Easily Available

Sora Suegami

Japanese High School Student

suegamisora@gmail.com

December 19, 2018

Abstract

Cryptocurrency and blockchain technology makes it possible that unspecified persons consent to any state transition and electronic value in trustless peer-to-peer networks. However, today there are high hurdles to get cryptocurrency at first and use blockchain.

Vreath, our new public blockchain system with virtual machines, solve this problem. While thus far only mining winners could gain mining fees, our original consensus algorithm Proof of Unit share fees among all miners by measuring not only the largest hash power but also small one. Moreover, it achieves both security of proof of work and rapidity of proof of stake. Even smartphone users having no cryptocurrency can get mining fees with Vreath.

We also invented four scalability solutions: Parallel verification, NGV, Sharding and Child chains. Parallel verification reduces costs of state transitions by executing smart contracts parallelly. NGV and sharding grow block capacity of the main chain with high security and decentralization. Child chain realizes any state transitions on trustful chains with main chain security level.

Vreath aims to bring everyone cryptocurrency and become the largest value platform in the world. You can circulate any value on Vreath and anyone can access them without any hurdles. All people bring value breath to society.

Contents

Part I
Introduction

Satoshi Nakamoto invented a way to express values on the internet without any trusted third party.[1] As you know, that is called blockchain and is said to be one of the most innovative technologies. Ethereum exceedingly increased its potential, enabling any state transitions to be executed in Ethereum virtual machine with no trusted third party.[2] Anybody can design flowings of values and run them all over the world by deploying programs called smart contracts. We believe that any value circulates around our society and money is brought to persons who actually give values to others when blockchains become common.

However, there are still high hurdles to overcome to do that. One of them is the great deal of trouble getting cryptocurrency at first. Generally, blockchain users need it to pay their transaction fees and the transaction-fee-system is indispensable to prevent spam attacks. Despite that, users are required to go through many troublesome steps such as: register with a cryptocurrency exchange, KYC, depositing legal currency and so on. They prevent applications based on blockchains from inviting new users.

Vreath is developed to break the hurdle. Vreath enables anybody to get cryptocurrency instantly and automatically. They don't need even bank accounts. All smartphone users directly become users of the cryptocurrency, blockchain and Vreath. Besides, Vreath is scalable enough to accept many transactions from large users. We want to make our blockchain the largest value platform in the world, and on it, any values and money circulate without any limitations, "to bring value breath to society". Vreath exists for that.

We invented new concepts such as states are divided by state owners' addresses and users change them parallely with mining. They realize our new consensus algorithm and scalability solutions: even smartphone users can prove their hash power and gain fees with the consensus algorithm, changing states asynchronously and securely by declaring what to change. All of the technical elements are linked together and indispensable for our future.

Part II
Technology

1 Definition

1. Vreath virtual machine(VVM): virtual machine executing smart contracts in Vreath.
2. token: a module that has its own states, information and smart contract code.
3. address: a user identifier that consists of hash calculated from the public keys and token name.
4. request tx: a transaction that requests state transitions.
5. refresh tx: a transaction that approves a request tx and presents new states calculated by executing smart contracts.
6. requester: a publisher of request tx.
7. refresher: a publisher of refresh tx.
8. unit: a symbol of hash power created from the refresh tx.
9. miner: a publisher of unit with mining.
10. key block: a block that elects the next leader making micro blocks.
11. micro block: a block that includes transactions recognized to be valid.
12. validator: a publisher of blocks, buying units from miners to succeed in making key blocks.
13. VRT: our native currency used for transaction fees, miner's fees, validator's fees and so on.
14. gas: a cost to execute smart contracts. Requester pays it for the refresher.

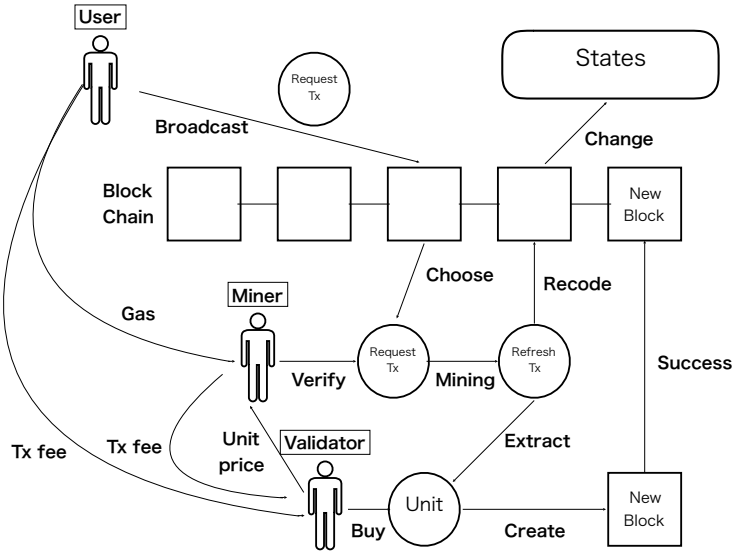


Figure 1: The complete picture

2 New Concepts

2.1 Token Base

Most blockchains adopt an UTXO model or an account base model to manage their states. However, Vreath devises a new model, Token Base. Each token has some token states, one token information and one code executed in VVM.

Token states are distinguished by state owners' addresses. They have the following properties.

- nonce: the number of times the state is changed.
- token: the name of the belonging token.
- owner: the state owner's address.
- amount: the amount of the token that the owner has.
- data: key-value data linked to the state.

Token information is assigned to each token and summarizes its information. It simplifies state processing. There are the following properties in token information.

- nonce: the number of times the information is changed.
- token: the name of the the token.
- issued: the amount of the issued token.
- code: the hash of the its program code.

This model makes state transitions simple and secure. They are functional language style, changing only input data and outputting new states. They don't change data in variables like procedural languages. Developers only have to consider how input states change, so they can easily prevent unexpected state changes by attackers. Besides, declaring which states are changed helps parallelize state transitions. It is indispensable for Vreath to be scalable.

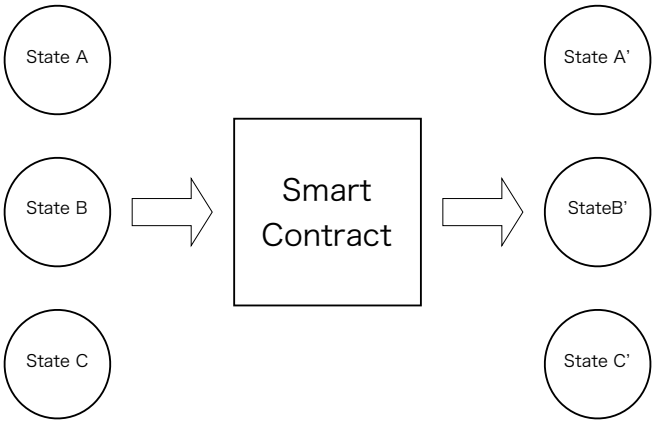


Figure 2: Our smart contracts are pure functions

2.2 Request Tx and Refresh Tx

There are two kinds of transactions in Vreath: request transaction (request tx) and refresh transaction (refresh tx). They make it unnecessary for all nodes to execute the same smart contracts like other blockchain systems.

Request tx requests state transitions such as remittance, storing data and so on, but they are not executed yet. It is broadcasted by users to change their states. The following infomation is included specifically:

- base states: the states to change.
- input data: any data input to the smart contract.

Refresh tx approves a request tx and presents new states. Only refreshers and validators execute smart contracts and the others accept new states in refresh txs. There is a possibility that they are invalid. Parallel verification (we explain it in the section 3.2) prevents that. The following infomation is included specifically.

- request tx hash: the hash of the request tx to approve.
- height: the block height contains the request tx to approve.
- block-hash: the block hash contains the request tx to approve.
- output state: the new states after executing the smart contract.
- nonce: the number used once for mining. (We explain it in the next section.)

While some refreshers try to approve the same request tx, only one refresh tx is committed to the blockchain. This limitation promotes rapid verification of request txs. (We explain the reason in the section 3.2.)

To request competing transitions is not allowed by our protocol. For example, when state A is requested and not refreshed yet, it can't be requested again. This request lock system prevents bona fide refreshers from calculating invalid outputs because of competition.

2.3 Unit with Parallel Mining

Making a refresh tx, refreshers need mining. A unit is created from the refresh tx by extracting this information: hash of the specified request tx, block height, block hash, nonce, address of the refresher, hash of output states and price of the unit. It symbolizes the hash power. It has to meet three requirements:

1. The request tx exists in the block specified with the hash and height.
2. The refresh tx which approves the request tx and presents the same output states is already committed into the blockchain.
3. This inequality is satisfied.

   $$hash(request\ tx\ hash + block\ height + block\ hash + nonce + refresher\ address + output\ states\ hash + unit\ price) \leq target$$

Parallel mining is the way to regard hash power as work by units no matter how small hash power is. Even smartphone users can gain mining fees: the difficulty ($= 1/target$) for mining is constant and so low that an average CPU can solve a mining problem within 10 seconds. They don't have to carry out mining first as their mining problems are distinguished by refresher's addresses: it means units are made parallelly. (The inequality to be satisfied requires refresher's address for the hash algorithm.) In other words, Parallel mining parallely measures not how fast difficult problems are solved, but how many times easy ones are solved.

3 Solutions

3.1 Proof of Unit

Proof of Unit (PoU) is our original consensus algorithm. It helps users get cryptocurrency with no hurdle. Moreover, it makes Vreath secure, valuable, scalable and flexible by combining proof of work (PoW) with proof of stake (PoS).

It is the most innovative point that users can readily be miners. PoU encourages validators to pay miners their VRTs as mining fees in the following steps:

1. Miners make units with parallel mining and broadcast them. They need not VRT but hash power.
2. Validators buy units by remitting their prices. They are caused by transactions about units.
3. Validators try to add their blocks into the blockchain: the success probability is directly linked to their balances of bought units. This inequality is satisfied.

$$sha256(previous\ key\ block\ hash + validator's\ address + timestamp) \leq 2^{256} * \frac{unit\ balance}{difficulty}$$

4. The validator fortunate enough to make the valid next block gains the block fee, new issue and transaction fees.

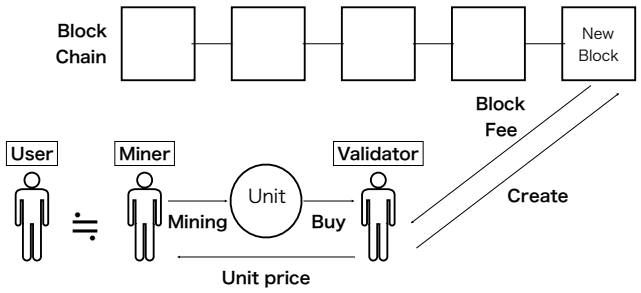


Figure 3: Proof of Unit

In terms of money flowing, miners receive VRTs from validators and validators gain block fees. This is win-win, sustainable and decentralized. In addition, miners themselves have to do nothing as not humans, but their PCs or smartphones calculate mining problems, therefore they can get cryptocurrency with no hurdles. They are used for transaction fees, remittances, Dapps and so on.

Bought units steadily disappear at every block. If validators fail to make blocks within the block time, their units don't pay, so PoU requires cost to make blocks.

PoU is the also hybrid of PoW and PoS. It is regarded as PoW wrapped in PoS, taking both advantages and offsetting their weakpoints as shown in the following table:

item	PoW	PoS
advantages	• Strong store of value with electricity. • High cost to fork the chain. • Incentive for new miners to enter the mining competition.	• Tolerance for 51% attack. • No incentive for holders to attck the chain. • Quick verification speed.
weakpoints	• Slow verification speed. • Large consumption of energy. • Unfair mining competition because of capital investment.	• Nothing at stake. • Long range attack. • Low fluidity.

PoU is as fast as PoS when it comes to the longest and valid chain, but it doesn't have the PoS problems such as nothing at stake and long range attack. First, attackers who fork the chain have to make units by themselves since units use block hashes for the hash algorithm. PoU turns into PoW with forking. Second, the total number of units isn't constant, so even early validators have little power to make blocks.

PoU also has weakpoints. The large distribution speed of VRT might reduce its market value and the number of units (hash power). It enables the attacker to launch a 51% attack with ease since it becomes low cost to surpass the total of bona fide miners' hash power. These harsh circumstances would eventually be resolved by ascent of the market value.

3.2 Parallel Verification

Parallel verification reduces the costs to execute smart contracts. Requesters request state transitions by request txs. Some refreshers compute them and present output states. Validators verify them and take valid transactions in their blocks.

Refreshers calculate state transitions to gain gas from requesters as fees whose prices are usually higher than those of the units. However, each requester pays them to only one refresher who committed the refresh tx into then blockchain. (Second and following refresh txs become invalid.) That is to say, they submit refresh txs as fast as possible to gain gas. However, if refreshers fail in being first, they can sell them as units. The requester doesn't have to pay much gas for heavy computing because refreshers have the incentive that there are only a few competitors. Deep Learning, simulation and other heavy processing becomes executable on VVM.

Naturally enough, invalid refresh tx might be broadcasted, but they are not committed into blocks by validators because most validators make no blocks after an invalid one. Invalid refreshers have as much damage as mining costs.

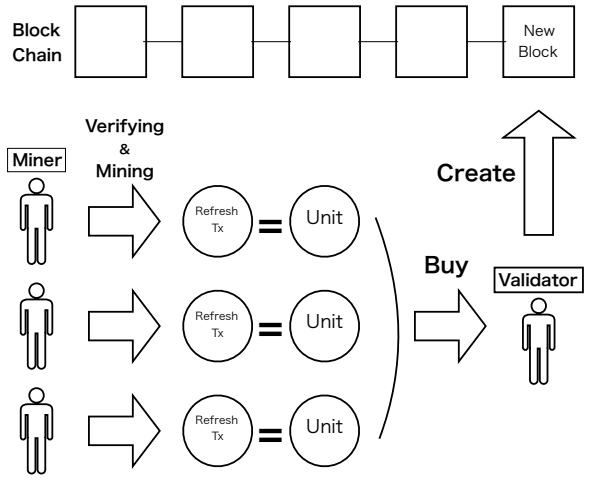


Figure 4: Parallel Verification

3.3 NGV (Next Generation Vreath)

We adopt Bitcoin-NG technology[3] to achieve scalability and decentralization on the first layer. It has the following feature:

> Bitcoin-NG achieves this performance improvement by decoupling Bitcoin's blockchain operation into two planes: leader election and transaction serialization. It divides time into epochs, where each epoch has a single leader. As in Bitcoin, leader election is performed randomly and infrequently. Once a leader is chosen, it is entitled to serialize transactions unilaterally until a new leader is chosen, marking the end of the former's epoch.

Key block is made at one-minute intervals, electing the next leader that makes the next micro blocks. It has validator's public key. A hash of key block has to satisfy the Proof of Unit inequality(presented in section 3.1). It takes one minute on average to get valid timestamp and issue new key blocks.

Micro block is made at one-second intervals until the next key block is broadcasted or the last leader broadcasts 60 micro blocks, including transactions recognized to be valid on the blockchain. Its hash is not required to satisfy any conditional expression; the leader validator only has to compute hash once per micro block.

The block fee is distributed 40% to the previous leader and 60% to the next one like Bitcoin-NG: this mechanism incentivizes to follow the protocol.[3]

When the leader makes the micro block that approves invalid transactions, the next leader forks the chain and takes it away. It encourages validators to make valid blocks.

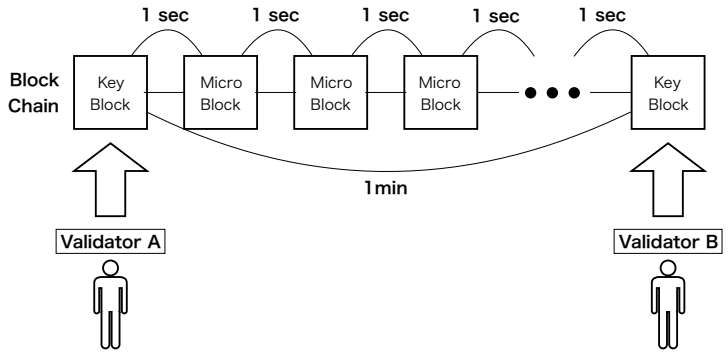


Figure 5: NGV

3.4 Sharding

Sharding makes our blockchain scalable on the first layer, committing lots of trsansactions and reducing validators' storage and network costs. Transactions data, states and validators are assigned to shard chains. Raw data of states about VRT and unit and address-hash map data about other tokens are stored in the main chain and all nodes.

Transactions are committed into the main chain by validators in shard chains and the main chain using the following steps:

1. Shard chain validators make blocks in their chains.
2. They broadcast sharding transactions. (We explain it in the next paragraph.)
3. Main chain validators put them into their blocks and commit the results of the state transitions into the main chain.

A sharding transaction presents states in shard chains. Its structure is almost the same as that of block header: hash, height, validator's public key, merkle root of states, merkle root of transactions, difficulty for PoU and so on. However, it specially has an address-hash map of changed states in shard chains. The information makes state sharding possible because it prevents split states from competing, not changing states of duplicate addresses at the same time. As shown in Figure 6, when some shard chains change Alice's state at the same height, only one of them is chosen by the main chain validators. The shard chain not chosen has to throw the block away and use another one. Competing state transitions are never committed into the main chain.

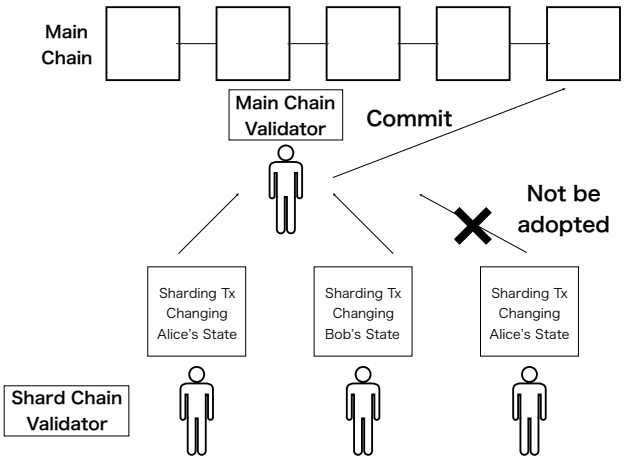


Figure 6: Sharding

The number of shard chains increases with units increasing, which means the growth of miners' hash power and attack costs. We postulate that the minimum amount of units to keep security is a units. When the total of units is na, n shard chains are made. As users grow, the scalability grows. This is an essential principle of our sharding.

3.5 Child Chains

Child chain is a scalability solution on the second layer. Any state transitions are executed on trustful chains with main chain security level. We use Plasma design as a reference. [4]

> We compose blockchains into a tree hierarchy, and treat each as an individual branch blockchain with enforced blockchain history and MapReducible computation committed into merkle proofs. By framing one's ledger entry into a child blockchain which is enforced by the parent chain, one can enable incredible scale with minimized trust (presuming root blockchain availability and correctness).

Administrators of child chains change their states and regularly submit information about state transitions, like the following, into the main chain:

- Changed states address.
- Address-hash map of new states.
- Merkle root of data input to smart contracts.
- Source codes of new smart contracts deployed to the child chain.
- Public keys of new administrators.
- Signature of administrators.

There is a possibility that the submission is invalid. Users can prove frauds to the main chain and punish administrators by challenge games. They require the following information about one-state transitions:

- Raw data of base states (changed states)
- Raw data input to the smart contract.
- (If they exist) Another merkle root of input data.
- (If they exist) Signature of administrators for the above.
- Proof that the input data is included in the merkle tree whose root is submitted by administrators or the challenger.
- Signature of users who caused the state transitions.

In challenge games, validators in the main chain confirm three points: whether hashes of base states are already committed to the main chain in previous administrators' submittals, whether the merkle proof of input data is valid and whether hashes of valid output states calculated with execution of the smart contract are different from those submitted by administrators. Knowing base states, input data and the source code as pure functions, they can calculate valid output states. In addition, when the merkle root which administrators submitted is invalid, the challenger can present a valid one, for valid ones should be broadcasted to convince users in the child chain. When these challenges succeed, the submittals from child chains are regarded as invalid and quashed.

A consensus algorithm of child chains is PBFT: Practical Byzantine Fault Tolerance.[5]

> The algorithm offers both liveness and safety provided at most $[\frac{n-1}{3}]$ out of a total of replicas are simultaneously faulty.

If more than $\frac{2}{3}n$ administrators sige submittals, the main chain accepts them and doesn't force ways to elect validating nodes and leader nodes, allowing only one node to manage its child chain. Such centralization realizes high throughput and Vreath becomes scalable.

The child chain mechanism is capable of changing any state as secure as public blockchains and as fast as conventional centralized systems.

Part III
Roadmap

- Now

  We already developed a program to demonstrate that users can get VRTs with no special operation and remit them to anyone by only pushing a few buttons. However, we have to rewrite core parts because after developing them, we slightly changed some specifications.

Figure 7: First user's balance is zero.

Figure 8: By just waiting, user's balance increases.

- Q1 2019 (January - March)

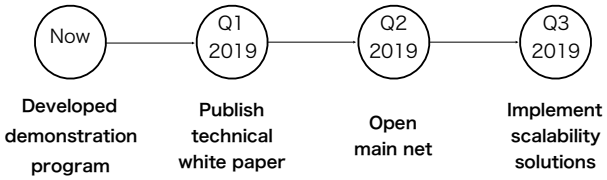
  We plan to decide a detailed specification and publish it in a technical white paper. For example, we have to choose a kind of hash algorithm for parallel mining, mathematically prove the safety of our consensus algorithm and improve models of sharding and child chains.

- Q2 2019 (April - June)

  Our main net is to be opened with Proof of Unit, Parallel Verification and NGV: remittance of VRT and smart contracts on VVM are achieved, but we don't solve the scalability problem at this time.

- Q3 2019 (July - September)

  Our scalability solutions, Sharding and Chaild Chain, is to be implemented to the main net. Vreath will realize huge scalability and become the largest value platform in the world.



To realize this grandiose plan, we want your help. If you are developer, I'd like you to participate in our open source project. The dream of Vreath won't come true without you.

Part IV
Conclusion

Today, cryptocurreny and blockchain technology are still uncommon. We thought that the big trouble of getting cryptocurrency for the first time blocks the spread of it. Vreath solves this. We first proposed three new concepts: Token Base, Request Tx and Refresh Tx and Unit with Parallel Mining. They construct a new consensus algorithm Proof of Unit that shares mining fees not with a few winners, but with everyone that provides even just a *little* hash power. Even smartphone users can gain cryptocurreny as miners without any hurdles - they only wait for their smartphones to finish mining.

Vreath improves not only usability but also scalability. Parallel Verification reduces the costs of state transitions by executing smart contracts parallelly: all nodes don't compute for all state transitions. NGV makes the capacity of blocks big while it keeps decentralization. Sharding is a solution to process a lot of transactions on the first layer. Child Chain is a second layer scalability solution that realizes any state transitions on trustful chains with main chain security level.

Vreath brings visible cryptocurrency and comfortable blockchain technology to people all over the world, and evolves to a worldwide value platform. People can exchange their values with anyone without any hurdles. Our blockchain supports such value breaths as an unsung hero.

References

[1] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", https://bitcoin.org/bitcoin.pdf (November 18, 2018)

[2] Ethereum Developer Team, "A Next-Generation Smart Contract and Decentralized Application Platform", https://github.com/ethereum/wiki/wiki/White-Paper (November 18, 2018)

[3] Ittay Eyal, et al, "Bitcoin-NG: A Scalable Blockchain Protocol", https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-eyal.pdf (November 23, 2018)

[4] Joseph Poon and Vitalik Buterin, "Plasma: Scalable Autonomous Smart Contracts", https://plasma.io/plasma.pdf (November 24, 2018)

[5] Miguel Castro and Barbara Liskov, "Practical Byzantine Fault Tolerance", http://pmg.csail.mit.edu/papers/osdi99.pdf (November 25, 2018)