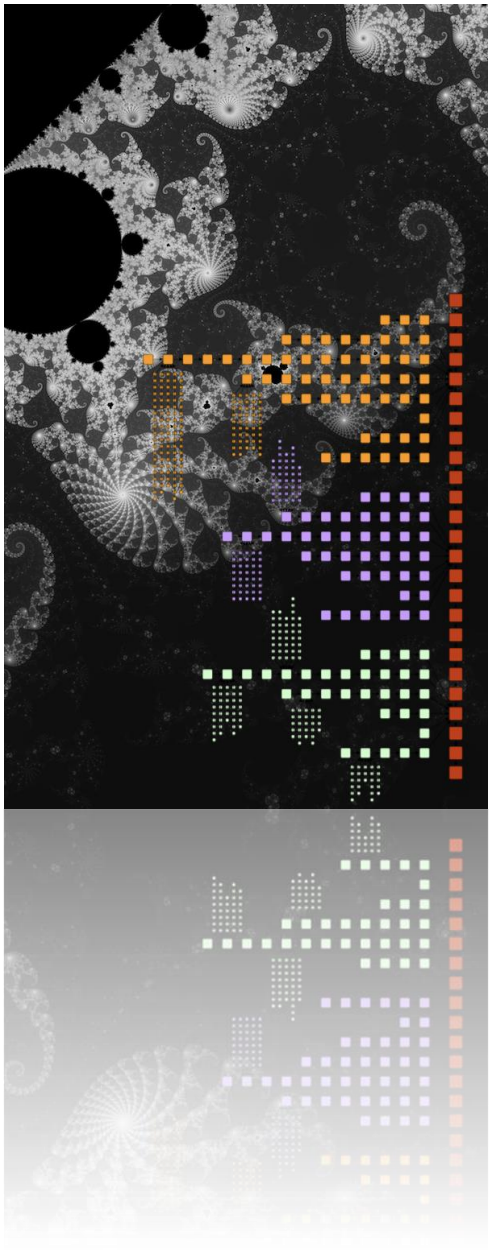


ASSETREE

Distributed Fractal Ledger

ABSTRACT

ASSETREE: Construction of a high-concurrency, extensible and partitioned distributed ledger system that is based on DFL technology, which is used to create a blockchain network that connects any variety of real assets from the real world to the virtual world.

Z. Johnson

Technical impact whitepaper of SharesChain

Contents

I. Introduction

1.1 Single Chain Blockchain

Bitcoin Blockchain

The Bitcoin Blockchain consumes large amounts of energy. The Proof-of-Work Consensus Protocol of the Bitcoin blockchain system needs to consume lots of computing resources and energy, and it is still growing rapidly which is indicative of needing even larger amounts of computing resources and energy as time progresses.

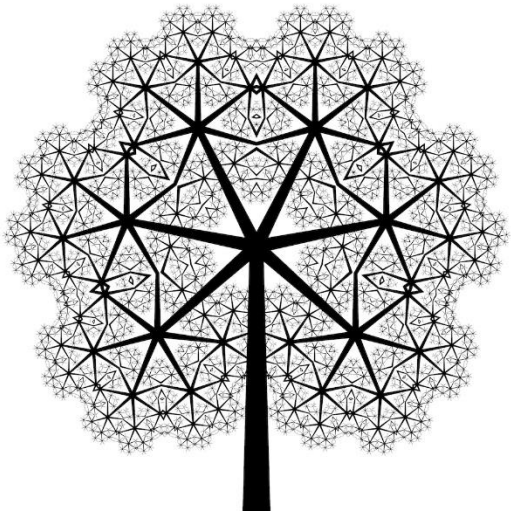
One result of this fact is that democratic characteristics fade. The Bitcoin blockchain requires more than 50% of the computing resources to be held in the hands of honest users to ensure the system runs consistently. However, Satoshi Nakamoto may not anticipate that the computing resources in the Bitcoin blockchain are concentrated in a few large mining pools, which made the currency blockchain deviate from its early-claimed democratic characteristics. Discussion about expansion and multiple bifurcation in Bitcoin's history prove that the Bitcoin blockchain has formed a highly centralized community structure.

The transaction rate is extremely low. The size of a Bitcoin block is 1M, which can accommodate about 2,000 transactions. Because one block is generated every 10 minutes on average, the Bitcoin blockchain can support three to seven transactions per second.

Ethereum Blockchain

For Ethereum, the transaction rate is limited. Ethereum's current performance (7-15TPS) is extremely weak in the face of any scenario with large-scale transactions.

There is a lack of concurrency mechanisms with Ethereum. Ethereum, a distributed computing network, is responsible for the distributed computing of global Ethereum-based projects. Its single-chain design of a single-thread model leads to conclusions that slightly large-scale calculation requests will result in a "queue jam", which will further lead to a single Ether-net application (such as CryptoKitties) affecting the computing power behind the global Ethereum application. As a universal computing network, as Ethereum strives to be, it severely lacks the necessary concurrent mechanisms to function under severe loads.



1.2 DAG Blockchain

There are many challenges for the DAG high-speed asynchronous blockchain technology. For example, there's no strict requirements for the strong consistency of the whole network transaction status; it is necessary to control the convergence degree of the units without subunits to avoid too many unconfirmed transactions while obtaining concurrent validation. It is necessary to support enough SubChains to acquire adequate concurrency capabilities while building the MainChain. Though the concept of the blockchain is eliminated, the efficiency may degenerate to a degree that is similar to a SingleChain when the number of chains is relatively small.

In fact, any distributed ledger system has their own positioning, including the system adopting DAG technology, such as IOTA; however, it is at an expense of the strong consistency of the whole network transaction. The adaptive capacity in the Internet of Things is excellent [1], while Byteball focuses on business in the field of payment.

The positioning of Assetree is to obtain high-speed parallel ability without losing proper consistency, and to support the effective operation of massive real asset application in the real world.

1.3 Assetree as a Fractal Ledger Structure

The world is nonlinear, and fractal patterns and objects are everywhere. Complex structures tend to consist of simple units. **We are building a strong-consistency, application-isolated and high-concurrency distributed ledger system, that will be used for the registration, management, and transference of assets.** This enables us to develop a structure like a tree – spreading out at its roots, developing branches that can, themselves, branch outward. The tree-structure blockchain (which is why we call it Assetree) is the maximally efficient way of inducing this logic. Its logical structure is stable, clear and it is very beneficial to the divisional design that is based on addressing issues of space and capacity. The chain structure has also been proven to be able to have a perfect distributed consistency solution – the tree structure blockchain can support the extension of fractal structures and can support high-dimensional design: The Distributed Fractal Ledger (DFL).

DFL emphasizes the simplicity of the infrastructure, the connectivity of the infrastructure, and the recursive nature of the connection rules. Different infrastructure, under different random distribution, can build a different DFL, and can obtain different access paths and efficiency levels. Assetree meets the fractal structure characteristics. On the other hand, we only need to define the rules of infrastructure, then we can duplicate a tree with recursion to manage all the ledger data.

The Consistency Problem is the most basic problem in distributed computing. It refers to a set of operations for nodes in distributed networks and achieves common identity to the processed results under the guarantee of agreement protocols. The recognition process can be expressed by consensus algorithm, and the structure and state of the data will seriously affect the performance of the consensus algorithm. Due to the existence of CAP principle (one of the important principles of distributed computing that was put forth by Eric Brewer in 2000, at the ACM conference): i.e. a distributed system cannot satisfy the Consistency, Availability and Partition at the same time. This indicates that we need to make our choice when we design the system. We will remove the uncertainty of the system, even at the expense of efficiency and space. At the same time, according to the possible number of nodes and the concurrency value, we can design the hierarchy of the ledger structure to be built with recursion as well as the degree of addressing space partition through partition management of the address space.

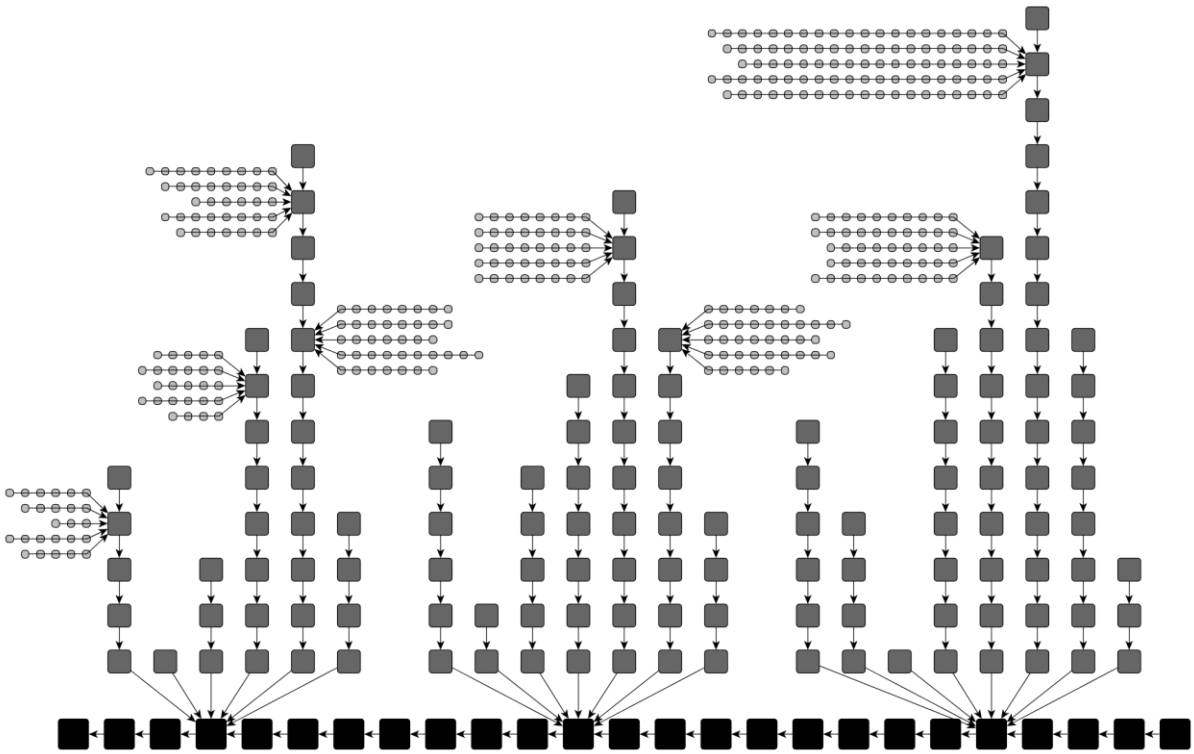


FIG. 1 Schematic Diagram of the Assetree Ledger

1.4 Smart Contract Application

If the bottom tier of the blockchain is compared to the data tier, the smart contract belongs to the business tier. Assetree advocates the decoupling design of the data tier and the business tier; the storage of blockchain data should use professional block database (BlockDB) to read, write and index; the processing of smart contracts should be handled by the Smart Oracle Server, and thus it provides the ability to interact with the Internet outside blockchain system. Assetree will support the assets by using smart contracts when transacting in the network, and this will make both parties involved in the assets transaction able to meet the limitation of factors such as the issuer, or to enhance the issuer's ability to program and control assets.

1.5 Connect Real Assets

It is difficult to guarantee the speed and extensibility of massive real assets relying on a single mechanism.

Assetree supports:

- ✓ The configuration of the SubChain consensus algorithm and parameter definition when assets are issued.
- ✓ A design of any kind of application that is an isolated distributed ledger system to avoid the impact of a single application's effect on other applications, so that multiple applications can run concurrently.
- ✓ A ledger system of address partition that obtains the concurrency capability in applications.

II. ASSETREE Network Structure

- Assetree is composed of the MainChain that is responsible for assets issuance and the SubChain that responsible for asset transactions. Please refer to the following diagram for a pictographic representation.

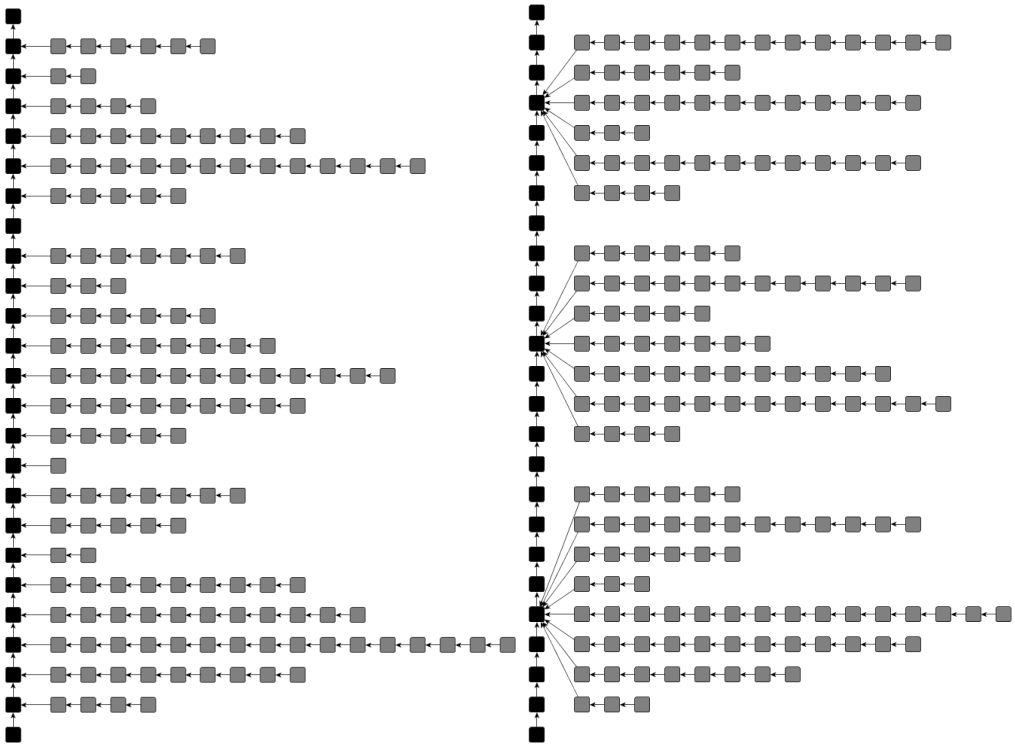


FIG. 2 Assetree Subchain Diagram

- MainChain (MainChain, MC), as shown in FIG. 1, is the dark black chain in the direction of the Y-axis, and there's only one MainChain; SubChain (SubChain, SC), other chains beyond the MainChain are SubChains, the first block of each chain uniquely point to one block on the MainChain (GenesisBlock, GB), as shown in FIG. 2.
- Each application can have one or more SubChains, which can be distinguished by a modulus according to the account address. For example, to divide into 16 SubChains, then all the transactions that the result of address 16 Modulo Operation of a payer user is equal to zero form blockchain to connect to the #0 SubChain; the maximum value of SubChains is 1 million.
- Assetree utilizes the Merkle tree to do quick verification about whether the transaction data is modified.
- Assetree adopts the UTXO transaction record model, and it imposes a strict constraint in terms that the sources of funds should belong to same payer, and thus limits the scope of exporting funds to serval receiver addresses of payers, and one certain SubChain space. In the case of multi-party payment, multi-party collection and cross >=3 chains, the upper application should be responsible for avoiding or splitting into multiple 2-chains transactions.
- Assetree supports a new transaction model: split UTXO into ITXO that only keeps the information of assets received by a particular recipient, and OTXO only keeps information of an assets payer (sender), and so it

will separately package ITXO and OTXO into corresponding SubChain, which allows the system to disconnect the relevance of both trading parties, which makes their transaction history untraceable.

- Assetree was designed to support three types of nodes:
    - Super node, which synchronizes all the ledger data, and supports the transaction confirmation and consensus of all types of assets, and the service of Smart Oracle.
    - Whole node, which synchronizes all the ledger data of a certain kind of asset which has a low load and supports the confirmation and consensus of single type of asset, and the service of Smart Oracle.
    - Light node, which synchronizes part of the ledger data of a certain type of asset and its own address related to the SubChain, which is suitable for assets management and transaction.

III. Assetree Assets Issuance

- Assetree issues its native Gas Token for use in stimulating the construction of its own blockchain network: SCTK; **SCTK defines the number of SubChains as one million, which can support a concurrent value of one billion in the future if calculations accord themselves to 1000TPS per single chain on average.**
- The "assets issuance" information of application forms blocks (GB) and records this information on the MainChain. A series of configuration information including: name, ID, total circulation, the minimum unit, whether issuance supports additional factors or not, whether the issuance supports the mining tokens or not, whether it supports burning or not, the number of SubChains, a consensus algorithm, random selection rate R of consensus nodes, distribution agreement, and assets all come with smart contract assets. The Tokens issued by different applications are different.
- **A certain amount of SCTK is required to be paid by the application party if they want to issue assets, and this value is proportional to the number of SubChains. If the application Token is issued successfully, these SCTK will be burned by the network.**

IV. ASSETREE Assets Management

Assetree is committed to providing rich operational interfaces and services for upper assets registration, management and flow logic of the business process. Some examples include: Multi-account management, permission configuration, information publishing, point-to-point communication, and the data encryption and storage of assets, etc.

Assetree, then, also supports:

- Inter-account asset operation, including freezing, staging, pledging, burning, etc.
- Multi-role account addresses and allowing for transaction control of a multi-account signature on assets along with control of multi-account information modification.
- Inter-account communication channels, and the realization of inter-account encrypted and unencrypted real-time reliable communication.
- Account data management, and the implementation of encrypted and unencrypted storage and modification of account data.

V. ASSETREE Assets Transaction

- The transaction message of SCTK forms blockchain and record in SCTK's own SubChains; SCTK transactions need to pay a certain amount of SCTK, and upon completion of the transaction, the SCTK will be burned, and the miners will receive a certain number of SCTK as awards.
- Each type of asset has its own SubChain to carry all distributed computing after issurance; The asset transaction message of the application forms blocks and record in the SubChain;
- When a user initiates an asset transaction, a certain amount of application Token need to be paid. The more they pay, the more likely they are to be packed by the miners;

VI. Assetree Consensus Protocol

The consensus mechanism in the public chain is generally a POW or a POS mechanism, and the influence of nodes on consensus directly depends on their resources in the network. **By using a random algorithm, the circumstance that consensus is confined to a few nodes or mining pools can be avoided.** This will also avoid white list attacks, avoid the need to trust strange subjects, and it allows for votes regarding any problems of human intervention, etc. It greatly strengthens the democratization of whole system. At the same time, Assetree by default will adopt a similar POS Consensus Protocol to avoid the issue where mining consumes vast amounts of energy.

Assetree network maintains several requests in a Message Queue according to the type of Message (trade, communications, storage, etc.) Super nodes and whole nodes of the consensus nodes support multithreading listening and message processing. Each thread must be configured with a SCTK account address (ADR) that corresponds to the "listened" SubChain address space, and this is to be used for the charges calculation when participate in consensus.

There are two kinds of Listen Thread for transaction message.

One kind is to listen and process Cross-SubChain transactions (Double-Thread). For example, thread DT1 listens to the transactions on SubChain #X and #Y, i.e. all the transactions that occurred between #X and #Y will be listened.

The other is to listen and process the transactions in *singular* SubChains (Single-Thread). All such transactions include light nodes that can participate in consensus computing. In addition, by configuring DT and ST with a multiple-chain internal address, the thread can gain the ability to listen to multi-chain internal transactions; however, it must be noted that this thread cannot concurrently-process multiple chain pairs. Of course, as a response, one can start multiple threads to get the concurrent-processing ability of chain pairs, which inherently avoids that problem altogether.

For a cross-SubChain transaction message, if it involves >=3-chain transactions, it will be split into multiple two-chain transaction messages by the business tier. When the transaction message comes, there is an option to perform consensus processing: participate in a DT thread in the consensus node and this is responsible for the consensus processing of two related SubChains, and thus they are bundled together. If one of the SubChains fail, the other will fail too. This avoids an inconsistent transaction message on two SubChains. Of course, the transaction message will not be calculated repeatedly. This is because the entirety of the transaction message of a certain user only needs a statistical analysis of a single SubChain ledger.

Assetree ensure randomness and democracy by the selection of random seeds (such as hash value of the latest block in the SubChain); this is unlike DPOS, which is not an equity agency, but randomly selected consensus nodes. Then the system will determine appropriate accounting qualifications according to equity. Let's call it RPOS (Random POS). For example, let us assume that the hash of the latest block in the SubChain is Hash(HDR), and the nodes that want to participate in consensus computing need to configure its listened SubChain pair (#X#Y or #X#X) message; then, every candidate will pledge a certain number of SCTK as a qualification certificate of accounting, thus becoming Hash(HDR) + Hash(ADR) = (INT)H. If H mod 1/R=0, this user will be elected as a witness (otherwise, they will no longer participate in the calculation). Then, assume that IDX is defined as the number of

rounds of election, H=(INT)Hash("H+IDX"). This will elect again among the witness, until the number of elected witnesses is less than C (configurable, such as 17). In the case which the witness who pledges the most SCTK assembly blocks will then perform a "fast signature" and consensus through the PBFT algorithm with other witnesses. The selection process of RPOS nodes do not need candidate nodes to perform complicated computing or mass communication. Every node can compute that if they are selected with fixed computing methods (the selection of random factors will be further optimized to avoid repeated selection). Other nodes can also be verified. Then the consensus is successful, and the witness will share the transaction fee and network reward according to the pledged SCTK ratio. Otherwise, the pledged SCTK in terms of the accounting process will be burned.

The amount of SCTK rewarded in the network is inherently designed to decrease with the passage of time. The future revenue of witnesses will mainly come from application Tokens that corresponds to the listened SubChain (the transaction fees). This purpose is to increase the intensity of competition and allow the best applications to get highest recognition of the network. The auto-burn and native reward process of witness-based SCTK will be broadcast in the form of special transactions to super nodes, and the super nodes will batch confirm, reach consensus, account to the SubChain corresponding to SCTK, and will achieve the consistency of the information.

In earlier stages, Assetree plans to use RPOS + PBFT as its consensus algorithm and we assume a generic processing capacity of 1000 TPS. In the meantime, Assetree provide an expansion interface of the consensus algorithm, which allow the block assembler to choose a more efficient algorithm. Refer to endnote [2].

Consensus nodes can set several listened concurrence treads (ST, DT) according to its type (super node, whole node and light node) to participate in whole network consensus computing. Consensus computing on chain pairs (#X, #Y) is linear. In theory, on the premise that the transaction is uniformly distributed, the more consensus nodes, the greater the concurrent volume supported by the whole network remains true.

VII. Smart Contract

Assetree supports the Turing complete Smart Contract system, which is designed to support Smart Oracle. Smart contracts are the protocol of business tier, and the bottom tier blockchain is the protocol of data tier. The decoupling design of these two protocols is the direction that Assetree pursues, to support the complexity of the upper tier business.

VIII. Extraneous Description and Conclusion

The parameters and values mentioned in this paper will be adjusted scientifically in the process of subsequent mathematical simulation verification.

If it is needed to support the consensus of light nodes on a transaction application, we can use the account that belongs to a certain SubChain address space, to perform related assets operation (i.e. this will make sure there is no cross-chain transaction that occurred). This is an ideal choice for small applications with an intelligent terminal. In fact, there are no strict boundaries between the three types of nodes, which are restricted by computer performance, network rate and human configuration.

DFL is a design pattern, which has its adaptability. Different DFL systems can be designed according to specific business scenarios, which will also result in different computational complexities, such as a consensus algorithm, assets statistics of accounts and address management, etc. Combination of address space can be considered for a higher level of fractal extensions to this framework. One is to select a global, larger address space, and another is to composite small space. Generally, we don't need to worry about whether the address space is enough to use. The number of atoms in the universe is on the order of magnitude of 10^{80}, and the number of addresses of a 40-bit address is on the order of magnitude of 10^{48}, while the number is 10^{77} for a 64-bit address. This indicates the amount of time and growth this system can handle is quite sufficient for a substantive period of time. The DFL is very suitable for space division and has the greatest advantage of space dividing as well as its ability to manage transactions by address. Thus, the probability of uncertainty in similar DAG technology is greatly reduced. This is precisely why the power behind Assetree ought not go unrecognized; it has expansive capabilities that will result in stability and in the world of asset trading and asset management, this is not a luxury but a necessity.

IX. References

[1] https://iota.org/IOTA_Whitepaper.pdf
[2] https://arxiv.org/pdf/1607.01341.pdf
[3] https://github.com/ethereum/EIPs/issues/650
[4] http://pmg.csail.mit.edu/papers/osdi99.pdf
[5] http://ethfans.org/posts/Sharding-FAQ
[6] http://ethfans.org/posts/ethereum-sharding-and-finality
[7] https://byteball.org/Byteball.pdf
[8] https://Bitcoin.org/Bitcoin.pdf
[9] https://en.Bitcoin.it/wiki/Atomic_crosschain_transaction
[10] https://github.com/Bitcoin/Bitcoin
[11] https://trustnote.org/TrustNote-WhitePaper-en.pdf
[12] https://github.com/iotaledger
[13] https://github.com/EOSIO/Documentation