ASKCOIN

The AskCoin White Paper:
The coin to exchange knowledge with value

Guopeng Tan
tan@askcoin.org

Yiding Wu
dindinw@askcoin.org

Hongbo Xia
hongbo@askcoin.org

August 31, 2017
Version 0.1

Abstract

The AskCoin system is a blockchain infrastructure designed specifically for the knowledge sharing platforms. It works with existing Q&A platforms like zhihu.com, stackoverflow.com and quora.com etc. We consider a common decentralized blockchain platform for all these websites (we call them Apps) is necessary. It's a user-motivated way for people to provide valuable answers which would be rewarded by an independent cryptocurrency. The author can now exchange his/her knowledge with value much faster, easier and more transparent by using the platform-independent token which can be used anywhere in the AskCoin ecosystem.

1 Introduction

> “The success of people's efforts will depend not only on the money they themselves use but also on the effects of the money others use.”
>
> *F.A. Hayek*

Since the Bitcoin[1] was first introduced by Satoshi Nakamoto in January of 2009, it has been more than 8 years to prove its success. Bitcoin is the first cryptocurrency which is now widely used and accepted as a decentralized digital asset and value store. Since Vitalik Buterin first published the original Ethereum Whitepaper[2] in November 2013, it's 4 years efforts to build a decentralized platform for creating smart contracts and decentralized applications (Dapps). These successes connect with the blockchain technology.

Along with more than 700 alt-coins and 100+ platform based assets or tokens, the total market capitalization has topped more than $100 billion according to coinmarketcap.com. The blockchain technology, which is the base of all of those platforms, applications and tokens, has already impacted many industries and is trying to change every human being's life all around the world.

In the meantime, the evolution of the underlying blockchain technology keeps going. There are several important improvements on the technologies, including decentralized consensus algorithm, P2P network, distributed data store and most importantly, the data structure first described by Satoshi Nakamoto, the chain of blocks.

The problem of the data structure of linked blocks is that all blocks are linked into a single chain. The larger the chain grows, the larger the requirements for scalability and efficiency become as time goes by. We want to look for a new solution to improve the scalability and efficiency of the decentralized public ledger. Our solution is to use the DAG (Directed Acyclic Graph)[3], a different data structure rather than the chain of blocks. In the DAG-based design, the transactions do not have to connect into a single chain, the whole set of transactions is no longer a chain but a directed acyclic graph.

DAG-based designs are attracting more attention recently. Sergio Demian Lerner's the concept project DagCoin[4] was first introduced in 2015. Currently, there are two cryptocurrency projects which are famous for their DAG technology. One is IOTA's tangle[5] project and the other is the Byteball[6] project. Askcoin is trying to become the third DAG-based project. Additionally, the IPLD[7] project also provides some graph inspired data structures like merkle-graph and merkle-dag etc.

As we have mentioned above, blockchain technology is a great revolution in many aspects. Bitcoin has changed the payment system a lot in the current world. Ethereum's ICOs have brought many benefits to new companies and projects. In other fields, Steem[8] is a blockchain-based social media platform where anyone can earn rewards. Here we are introducing AskCoin, which is the Steem alike. A similar slogan for AskCoin is that AskCoin is a blockchain-based ask-and-answer platform where anyone can earn rewards by his/her knowledge.

Unlike Steem, we use the DAG technique. We believe the DAG-based design has many advantages and most-suitable for the project. And we will not fork from any current existing projects. Instead, we will try to build the system from scratch by using the JAVA programming language.

2 Cryptography and Address

Askcoin, like many other cryptocurrencies, is using the elliptic curve cryptography as the public key encryption to ensure the security requirements of confidentiality, authentication, integrity, and non-repudiation without compromising efficiency. Askcoin uses ASK address to identify Askcoin transactions, the address is derived from the Askcoin public key, from the private and public key pair generated by elliptic curve cryptography.

2.1 Private key and Public key

Askcoin chooses to use the ed25519[9] elliptic curve cryptography instead of the secp256k1[10] which is used by the Bitcoin and the Ethereum and a lot of altcoins. The ed25519 is one of the fastest and safest digital signature algorithms. It can achieve very fast message signing. In a reference implementation, ed25519 takes only 87548 cycles to sign a message, a quad-core 2.4GHz CPU signs 109000 messages per second. Key generation is almost as fast as signing.

All of these are done at the highest security level. It has a 2^{128} security target, breaking it has similar difficulty to breaking NIST P-256, RSA with 3000-bit keys.

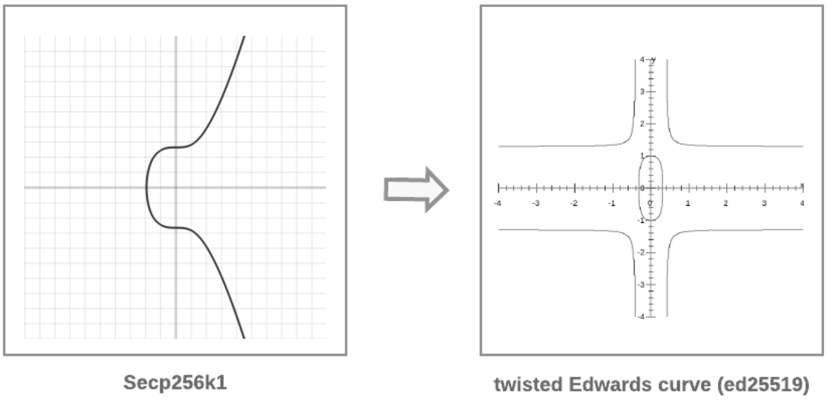


Figure 1: switch secp256k1 to ed25519

Askcoin choses to use the Twisted Edwards Curve[11] together with the ed25519 digital signature algorithms.

The equation of twisted Edwards curve is defined as:

$$ax^2 + y^2 = 1 + dx^2y^2 \tag{1}$$

An Edwards curve is a twisted Edwards curve with $a = 1$, The sum of these points $(x_1, y_1), (x_2, y_2)$ on twisted Edwards curve is:

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2}\right) \tag{2}$$

2.2 ASK Address

Askcoin addresses are the unique identifiers that are used in the Askcoin transaction on the Askcoin DAG to denote senders and recipients. The ASK address is derived from a public key and Bech32/base32[12] encoded. Comparing with the Bitcoin address's Base58 encoding[13], Bech32 address encoding[14] is more efficient and powerful. Askcoin address encoding is a derived version of Bech32 address encoding which is introduced by BIP173[14].

Some highlights of Bech32 address encoding included:

- Case insensitive: easier to read/write
- Simple to convert (no bignum logic)
- Only 17% larger than Base58

- Better checksums for prime power
- More compact in QR

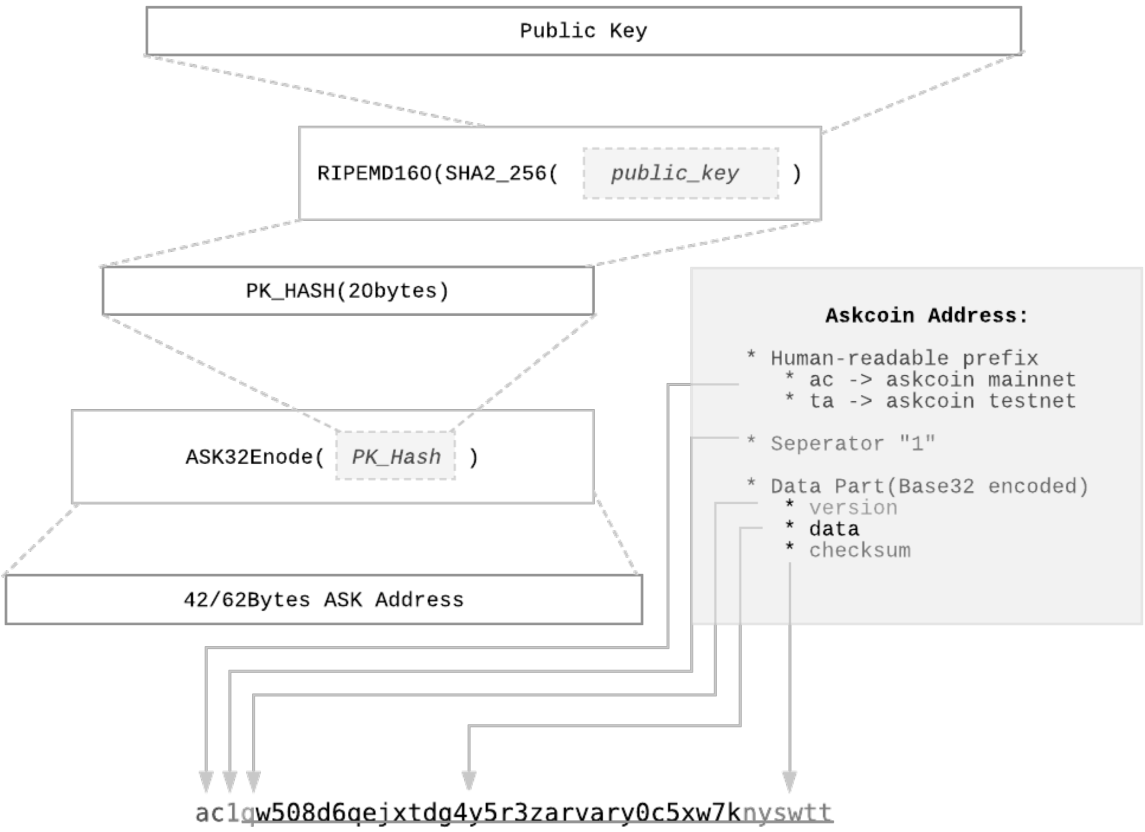


Figure 2: the Askcoin Adddress

3 Architecture

Askcoin uses DAG (Directed Acyclic Graph)[3] based technology, different from traditional blockchain technology, such as Bitcoin and Ethereum. Figure 3 shows the overview of PoW (Proof of Work) blockchain like Bitcoin:

The transactions are packed into blocks, and each block links to its previous block. This way, the block directly or indirectly approves its own and ancestor blocks' transactions. The double spend transaction would be rejected to be packed into block, or compete among candidate blocks. Finally, each UTXO would have only one spending transaction be accepted by whole network through PoW.

We believe DAG offers many advantages in the use cases where AskCoin aims to address. Before AskCoin, there are IOTA[5] and Byteball[6] projects which are famous for their DAG techniques. Considering IOTA is mainly applicable for massive transactions with small work, and its basic

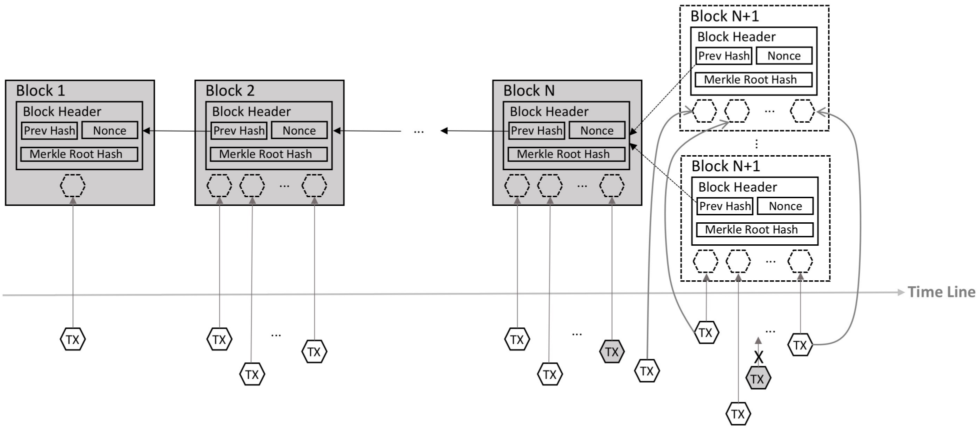


Figure 3: PoW BlockChain (Bitcoin) Overview

idea is still PoW, MainChain based DAG mechanism, which was firstly introduced in the Byteball project, is more suitable for AskCoin. Our team learned a lot from the Byteball project and will implement our own MainChain selection algorithm.

DAG has no concept of block, so it has no block time issue (Bitcoin has a 10 mins average block time, and the Ethereum has a 15 secs average block time). It means that DAG-based blockchain can have much shorter confirm-time. Without fixed blocks, there is also no capacity issue of block (currently the Bitcoin's block capacity is 2k+ transactions, and the Ethereum's block capacity is 200+ transactions). This means DAG-based blockchain is capable to have huge TPS. The transaction throughput may only depend on the network bandwidth, CPU processing time (like the signature verification time) and the storage limit.

DAG(Directed Acyclic Graph) is a finite directed graph with no directed cycles. A triditional DAG looks like this:

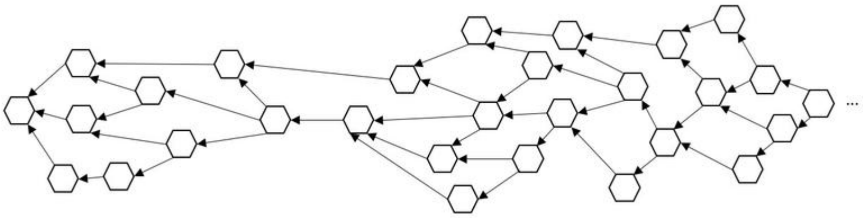


Figure 4: A topological ordering of a directed acyclic graph

In Askcoin DAG, the transactions are represented as vertices. The directed link edge represents confirming relationship between transactions, so called child-parent link. Child transaction must happen after its parents. During creating a transaction, it will choose certain present transactions as its parents. The child transaction directly confirms its parent transactions, indirectly confirms parents' parent transactions, etc. For a new transaction, it's preferred to make it confirm as many ancestors as possible by a limited number of parents. So, childless transactions are more likely to be chosen as parents, and DAG would grow in one direction. In an ideal network, and all the participants behave honestly, we can simplify the relationship, the DAG would be drawn like:

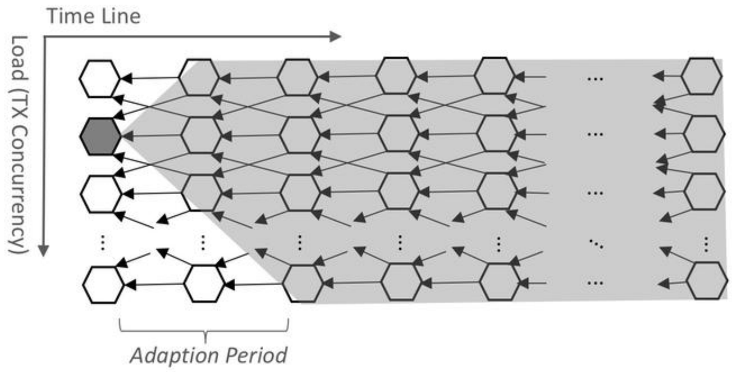


Figure 5: DAG Ideal Child-Parent Confirming

The load (concurrent transactions) is in proportion to TPS (Transactions Per Second). When the load is low enough, the DAG would behave as a linked list. Probabilistically, an old transaction would be confirmed by almost all the new transactions after an adaption period. On the other hand, a new transaction confirms almost all the old transactions before an adaption period. The adaption period is related to the load.

3.1 MainChain

By specific algorithm, we can choose certain trusted transactions, which linked together as a chain. We call it main chain. For each transaction on main chain, we allocate main chain index, which is increasing one by one away from Genesis transaction.

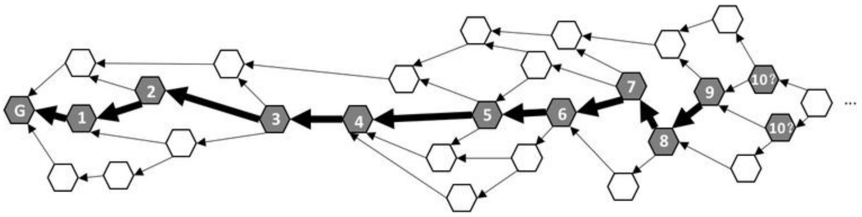


Figure 6: DAG Main Chain and Index

Along with the main chain (away from Genesis), to decide which one to be next main chain transaction, we need enough information of posterior transactions, which confirm (support) the candidate. The most important thing is to avoid the main chain sneaking off to attacker's private graph. To achieve this, special user, witness is introduced to maintain the reality of public DAG. Witnesses are reputable users, they (at least majority of them) would behave honestly.

Once we constructed the main chain, the transactions directly or indirectly confirmed by the main chain transaction are also get finalized.

For the transactions not located on the main chain, we also define a main chain index, it's same as main chain index of main chain transaction directly or indirectly confirming it for the first

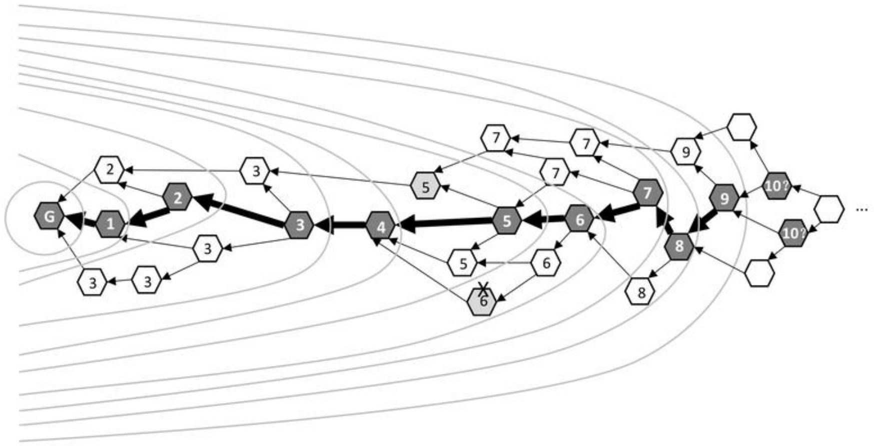


Figure 7: Transaction Finalization by Main Chain

time. This way, we can groups transactions by main chain index. Like the annual ring, main chain grows smoothly.

The Askcoin DAG main chain mechanism is similar to traditional blockchain. Each transaction on the main chain behaves as block header. One of its parent (previous main chain transaction) is same as “Previous Hash” in block header, it links the main chain indexes as a chain. The other parents are corresponding to “Merkle Root Hash” in block header, it used to verify the other transactions with the main chain index.

Unlike traditional blockchain, Askcoin DAG node would not directly reject the double spending transaction received from other nodes. To make sure the graph consistency among all the network nodes, this kind of double spending transactions are accepted first. After the corresponding main chain transaction get finalized, all the nodes would follow the same rules to mark only one of these double spending transactions as valid. Smaller main chain index transaction wins the larger one. For the same main chain index, smaller transaction hash wins. For the example in Figure 7, the double spending transaction with main chain index 5 wins its competitor with main chain index 6. If a client connecting only one node to publish its new transaction, node can safely reject it if it’s double spending transaction (just like blockchain reject double spending transaction getting into memory pool). This can minimize the double spending transactions flooding. To make double spending being detected and winner choosing as quickly as possible, Askcoin uses pre-final mechanism. Once a witness confirms one of double spending transaction, we can predict the possible main chain index to be assigned to this transaction, and pre-finalize the double spending transaction’s validity according to be rule defined above.

3.2 Hubs

Hubs are the key components of Askcoin network, just like miners in Bitcoin network. Hubs constitute the consensus network, which is responsible for reaching consensus towards the state of the main chain. In the other word, each hub would adopt the same algorithm to independently decide the main chain based on its own Askcoin DAG. In the certain time, hubs may see different

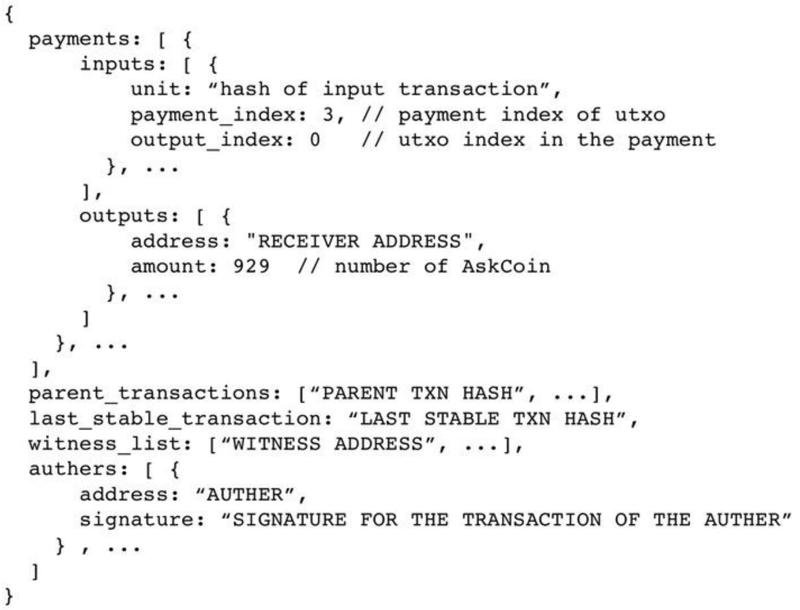
Askcoin DAG due to network latency, and the main chain decision may be slightly different, one may have more finalized main chain transactions than others. But after the graph gets synced, the main chain must be finally the same.

The other consensus among hubs is on how to handle double spending transactions. As depicted in previous section, the rule is smaller main chain index or smaller transaction hash winning. Once, the graph is synced, the final decision as well as pre-final decision would be consistent for all honest hubs. Even for the hubs with different snapshot of graph in a certain time, the pre-final and final decision must not be opposite. It's because the decisions are all based on witness transactions, when a witness transaction comes, its ancestors must be the same. So, the decision on its ancestors must be the same for all honest hubs.

3.3 Light Clients

Light clients mainly interact with hubs on new transaction posting and transaction finality SPV (Simplified Payment Verification). Light clients do not store all the transactions information, instead, they download transactions on demand.

To post new transaction, light client need to request corresponding hub for candidate parents (childless transaction on this hub) and the latest stable transaction on main chain. After organise and sign the message, the transaction format sent from client to hub:



```
{
  payments: [ {
      inputs: [ {
          unit: “hash of input transaction”,
          payment_index: 3, // payment index of utxo
          output_index: 0   // utxo index in the payment
        }, ...
      ],
      outputs: [ {
          address: "RECEIVER ADDRESS",
          amount: 929  // number of AskCoin
        }, ...
      ]
    }, ...
  ],
  parent_transactions: [“PARENT TXN HASH”, ...],
  last_stable_transaction: “LAST STABLE TXN HASH”,
  witness_list: [“WITNESS ADDRESS”, ...],
  authers: [ {
      address: “AUTHER”,
      signature: “SIGNATURE FOR THE TRANSACTION OF THE AUTHER”
    } , ...
  ]
}
```

Figure 8: Transaction Message Format

To do the transaction finality SPV, hub would send limited information to client. Because client only trusts its own witness (actually, witness's signature). Hub will try to convince client the transaction payment is directly or indirectly confirmed by the witness.

As shown in Figure 9, firstly, walk from the last main chain transaction back until meet majority

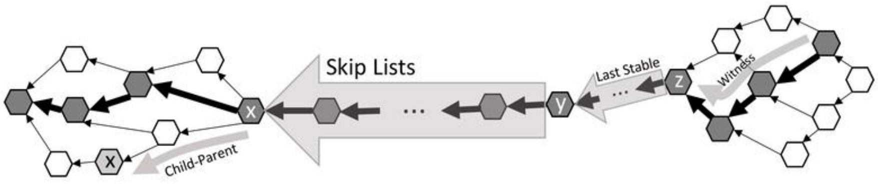


Figure 9: Light Client SPV

of witnesses signed transactions (Z), witness signature protects last stable transaction. Then, jump to the witness trusted last stable transaction (Y). And then, through the parent and skip lists of the stable transaction, we can reach the transaction (X) on the main chain with same main chain index as the transaction to be verify. Finally, we can reach and verify the transaction (x) through child-parent link.

To minimize the information sent from hub to client for SPV, skip lists is used for final (stable) transactions on the main chain. In Byteball, only transactions, whose main chain index can be divisible by 10 have a skip list, which lists the nearest previous main chain transaction whose index has the same or smaller number of zero at the end. For example, the transaction at main chain index 190 has a skip list that references the transaction at main chain index 180. The transaction at main chain index 3000 has a skip list that references the transactions at main chain indexes 2990, 2900, and 2000. In addition to the above skip list mechanism, Askcoin plans to introduce skip lists for main chain index cannot be divided by 10, simply skip to its previous main chain transactions with bigger number of zero at the end. For example, main chain index 1929 has a skip list references main chain indexes 1920, 1900, 1000. This way to ease the burden of light client on SPV further. But more work for hubs to build the skip list, we will balance them during Askcoin implementation.

3.4 Witnesses

Witnesses themselves behave as full nodes, they are non-anonymous reputable people or companies. It's expected that they will actively and honestly post their own transactions. Witness transactions are the key foundation for all hubs to get consensus on the state of main chain.

To keep the safety of the whole network, all hubs will choose main chain according to doctrine of democracy. It is to say, only majority of witnesses can impact the main chain selection. Single or minority witness become evil or get hacked would not impact the healthy of whole network. What's more, Askcoin plans to introduce recommendation election mechanism to keep witnesses list healthy.

4 Token

The system will have an internal token called ASKCOIN (ASK for short). The token will be used to pay the transaction fee and make the payment to a person who answered the question. The token can be used with the same value in different APPs that integrates the Askcoin blockchain. That is to say, the coin earned in app1 can also be used in app2. They are the same coins.

The initial supply for the coin is 1,000,000,000 ASK. The entire token supply will never change and no more token will be generated. The coins will be created in the genesis block(transaction) and will be distributed to ICO participants accordingly.

5 Internal Exchange Market

In later phases (details please see the Askcoin roadmap), the system will implement an internal exchange market. The user can exchange ASK with BTC and ETH directly from the mobile wallets. By doing so, the system will become a sidechain of both the Bitcoin and the Ethereum.

This is an innovation from Askcoin project. To make it tradeable with BTC or ETH we can put ASK into the cryptocurrency exchange markets, like poloniex.com or bittrex.com. But this is not a decentralized solution which has platform dependent risks. Some new projects, like cosmos[15] and polkadot[16], is trying to solve the problem in a better and decentralized way. But we believe that making ASK token exchangeable with BTC and ETH internally is also a good idea.

Askcoin network will act as a sidechain of both Bitcoin and Ethereum. Askcoin hubs will become Bitcoin and Ethereum light clients. And we will implement a PBFT similar protocol to submit Bitcoin and Ethereum blockchain headers.

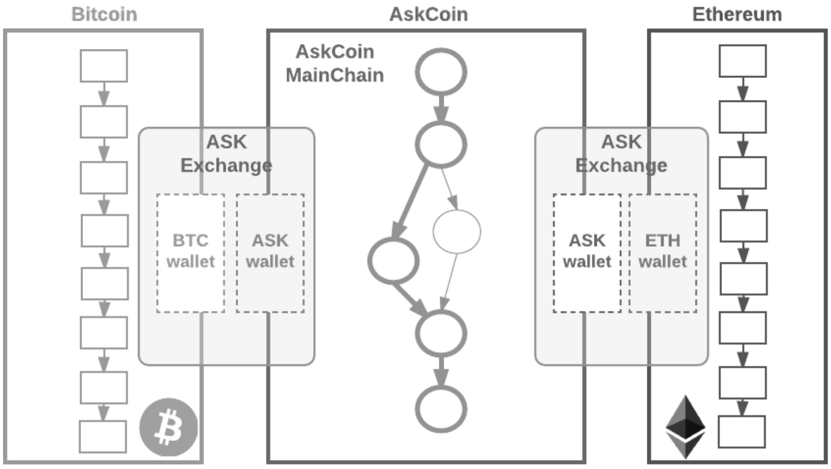


Figure 10: the sidechain of bicoin and ethereum

6 Transactions

Askcoin is designed for ask and answer APPs specifically. The system will have three basic transaction types which we call ASK, ANSWER and PAY.

6.1 The ASK transaction

The ASK transaction asks a question, provides a coin value which will be paid to the answerers later. The payment to the answerers should be issued by the questioner manually. To make sure the questioner will issue the pay transaction, a collateral is also provided. If the questioner failed to do the payment, the coin value for the pay and the collateral will be claimed by the hubs. If no one answers the question in the period of time, the coin value will be returned to the questioner.

Details for the information provided by ASK transactions are:

1. coin value to be paid
2. transaction fees
3. collateral coin value
4. exceed time (too short/long exceed time is not allow)
5. question content hash
6. app information
7. questioner address and signature

6.2 The ANSWER transaction

The ANSWER transaction is straightforward. It answers one ASK transaction.

Details for the information provided by ANSWER transactions are:

1. the ASK transaction hash
2. transaction fees
3. answer content hash
4. app information
5. answerer address and signature

6.3 The PAY transaction

When someone answered the question, and the questioner thinks it is the time to pay, then he/she would issues a PAY transaction to the answerers. The payment receiver can be multiple or single. The questioner can also divide the payment to each person. If he/she thinks no one should be paid or they should not be paid fully, he/she could also pay some to the hubs. After the payment, he/she will get the collateral back.

Details for the information provided by PAY transactions are:

1. the ASK transaction hash
2. transaction fees
3. detailed payment for each address(the delta, if any, will be claimed by hubs)
4. app information
5. questioner address and signature

7 Ecosystem

The Askcoin infrastructure is designed for ask and answer mode platforms, like zhihu.com, stackoverflow.com. We wish someday they would integrate with our system. We will also try to have more Apps integrated with us. Besides, we will develop a sample App by our own which is aimed for setting up a knowledge sharing community for blockchain technology

Essentially, every App in the ecosystem is an Askcoin wallet. You can ask and answer questions, make the payment, and transfer tokens. We will implement a default mobile wallet (for both Android and IOS) and with an internal exchange market in it.

The whole ecosystem looks like this:

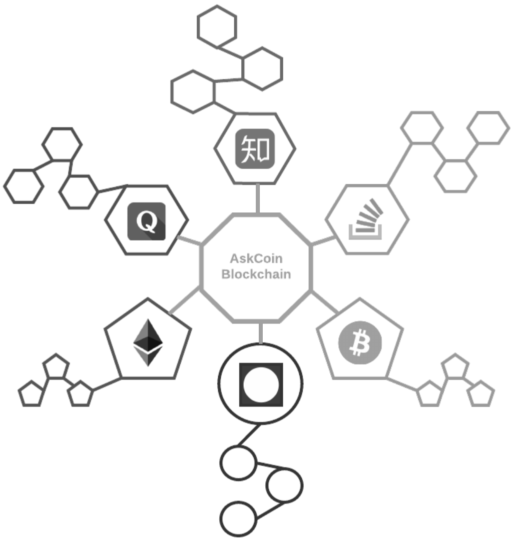


Figure 11: Askcoin blockchain ecosystem

References

[1] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf, Oct 2008.

[2] Vitalik Buterin and Ethereum Wiki. Ethereum White Paper : A Next-Generation Smart Contract and Decentralized Application Platform. https://github.com/ethereum/wiki/wiki/White-Paper.

[3] Wikipedia. Directed acyclic graph. https://en.wikipedia.org/wiki/Directed_acyclic_graph.

[4] Sergio Demian Lerner. DagCoin: a cryptocurrency without blocksDirected acyclic graph. https://bitslog.files.wordpress.com/2015/09/dagcoin-v41.pdf, September 2015.

[5] Serguei Popov for Jinn Labs. The tangle. https://iota.org/IOTA_Whitepaper.pdf, April 2016.

[6] Anton Churyumov. Byteball: A Decentralized System for Storage and Transfer of Value. https://byteball.org/Byteball.pdf, September 2016.

[7] IPLD. The data model of the content-addressable web. https://ipld.io/.

[8] Daniel Larimer, Ned Scott, Valentine Zavgorodnev, Benjamin Johnson, James Calfee, and Michael Vandeberg. Steem: An incentivized, blockchain-based social media platform. https://steem.io/SteemWhitePaper.pdf.

[9] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. https://ed25519.cr.yp.to/ed25519-20110926.pdf, September 2011.

[10] Bitcoin Wiki. Secp256k1. https://en.bitcoin.it/wiki/Secp256k1.

[11] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted Edwards curves. http://eprint.iacr.org/2008/013.pdf, March 2008.

[12] Wikipedia. Base32. https://en.wikipedia.org/wiki/Base32.

[13] Bitcoin Wiki. Base58Check encoding. https://en.bitcoin.it/wiki/Base58Check_encoding.

[14] Pieter Wuille and Greg Maxwell. BIP-173: Base32 address format for native v0-16 witness outputs. https://github.com/bitcoin/bips/blob/master/bip-0173.mediawiki.

[15] Jae Kwon and Ethan Buchman. Cosmos : A Network of Distributed Ledgers. https://cosmos.network/whitepaper.

[16] Gavin Wood. Polkadot : Vision For A Heterogeneous Multi-chain Framework. https://github.com/polkadot-io/polkadotpaper/raw/master/PolkaDotPaper.pdf, Oct 2016.