King DAG White Paper

https://www.kdag.io

Contents

0 Summary

0.1 KDAG Mission

KDAG is the underlying infrastructure of a new generation of value networks, dedicated to building a new generation of underlying trusted network protocols, and providing efficient, convenient, secure, and stable development and deployment environments to customers worldwide.

Its unique KDAG architecture completely replaces the traditional chain structure. A disruptive breakthrough in the theory of the traditional directed acyclic graph (DAG). The KDAG structure is used to organize the blocks. While achieving complete decentralization and completeness, under its KDAG architecture, the TPS can reach 30,000+ per second. Break the performance bottleneck of the consensus mechanism. The technically pioneered "hug algorithm" and "surf effect". The "hug algorithm" instead of consensus completely solves the data consistency, and the "surf effect" greatly improves the random attribute of the node's legal reference, and realizes the high security of transaction privacy.

0.2. Technical characteristics

0.2.1 Hug Algorithm

Embrace instead of consensus to solve data consistency. Through the original KDAG structure, the nodes that embrace it will get legal transaction references and achieve complete decentralization.

0.2.2 Surf effect

Randomness screening like the waves, the introduction of nodes to obtain the legitimacy of the transaction. Achieve high security of transaction privacy.

0.2.3 KDAG Structure

Break the traditional chain structure and design a new consensus mechanism on KDAG. While achieving complete decentralization and absolute security, it breaks through performance bottlenecks, and TPS can reach 30,000+ per second.

0.3. Ecological construction

KDAG is committed to open ecology, open applications, and combining with other ecological cooperations. As the underlying infrastructure of the new generation value network, it can fully integrate with big data, cloud computing, artificial intelligence, 5G and other technologies, and can seamlessly connect with other blockchain networks to realize the KDAG business community.

1 Background

1.1 Blockchain background

The birth of the blockchain is first of all the result of the evolution of the entire business society.

The first half of the human world was a centralized process. Various powers and the establishment of business institutions, corporate systems, etc., continued to the centralized mechanism in the digital world. In the Internet world, Internet banks, WeChat, Alipay, etc. maintain the trust relationship of the entire digital world.

Developed to the present moment, the centralized organizational structure is already facing certain obstacles that hinder economic development:

The first is privacy protection. At present, user data is concentrated on platforms such as WeChat, and the data belongs to the users themselves.

The second is cost. As the number of nodes increases, the cost of data usage will gradually increase.

The third is the issue of ownership.

These problems have led to the entire digital world calling for a decentralized business architecture.

The second background is that with the evolution of technology, Moore's Law has led to a gradual reduction in the cost of distributed computing and distributed storage. In some areas, the efficiency and cost of distributed architecture have advantages over centralization, and the blockchain was born under this historical background. Even if there is no blockchain, there will be other technologies. This is the result of the evolution of business and technology evolution in space and time.

Blockchain, as an integrated application of distributed data storage, point-to-point transmission, consensus mechanism, and encryption algorithms, is considered to be a disruptive innovation in computing models after mainframes, personal computers, and the Internet. New technological innovations and industrial changes. Blockchain technology originates from the foundational paper "Bitcoin A Peer-to-Peer Electronic Cash System" published in 2008 by a scholar named "Satoshi Nakamoto". In a narrow sense, a blockchain is a type of chain data structure that combines data blocks in a sequential manner in a chronological order, and it is a cryptographically immutable and unforgeable distributed ledger. Broadly speaking, blockchain technology uses blockchain data structures to verify and store data, uses distributed node consensus algorithms to generate and update data, uses cryptography to ensure the security of data transmission and access, and uses automated script code. A new distributed infrastructure and computing paradigm composed of smart contracts to program and manipulate data.

At present, blockchain technology is referred to by many large institutions as a major breakthrough technology that completely changes the way businesses and institutions operate. At the same time, just like new-generation information technologies such as cloud computing, big data, and the Internet of Things, blockchain technology is not a single information technology. Features.

Although the blockchain technology still has problems such as scalability, privacy and security, and the open source projects are not mature enough, the existing applications have fully proved the value of the blockchain. In the future, as the blockchain technology continues to mature, its application will bring the following values:

The first is to promote the development of a new generation of information technology industries. With the continuous deepening of the application of blockchain technology, it will create new opportunities for the development of next-generation information technologies such as cloud computing, big data, the Internet of Things, and artificial intelligence. For example, as Wanxiang, Weizhong and other key companies continue to promote the in-depth application of the Baas platform, it will definitely drive the development of cloud computing and big data. Such opportunities will be conducive to the upgrading of information technology and will also help promote the leap-forward development of the information industry.

The second is to provide technical support for economic and social transformation and upgrading. As blockchain technology is widely used in various economic and social fields such as financial services, supply chain management, cultural entertainment, intelligent manufacturing, social welfare, and education and employment, the industry's business processes will be optimized, operating costs will be reduced, and collaboration efficiency will be improved. Provide systematic support for economic and social transformation and upgrading. For example, with the continuous maturity of the application of blockchain technology in copyright transactions and protection, it will play a positive role in promoting the transformation and development of the cultural and entertainment industry.

The third is to cultivate new entrepreneurial innovation opportunities. Existing applications at home and abroad have proven that blockchain technology, as a tool for large-scale collaboration, can promote the breadth and depth of transactions in different economies to a new level, and can effectively reduce transaction costs. For example, Wanxiang will

combine the construction of "Innovative Energy Conservation City" to build a blockchain entrepreneurial innovation platform, which will provide platform support for individual and SME entrepreneurial innovation, because the foundation for the application of blockchain technology in the future. The foreseeable future is that with the widespread use of blockchain technology, new business models will emerge in large numbers, creating new opportunities for entrepreneurial innovation.

The fourth is to provide technical means for the improvement of social management and governance. As the application of blockchain technology in the fields of public management, social security, intellectual property management and protection, and land ownership management continues to mature and deepen, it will effectively increase public participation, reduce social operating costs, and improve the quality and efficiency of social management The promotion of social management and governance has an important role to play. For example, Ant Financial has applied blockchain to public welfare donations, setting an example for the whole society to improve the transparency and trust of public welfare activities, and also provides a practical reference for blockchain technology to improve the level of social management and governance.

With the advent of a new round of industrial revolution, the role of next-generation information technologies such as cloud computing, big data, and the Internet of Things in smart manufacturing, finance, energy, healthcare, and other industries has become increasingly important. Since the "Twelfth Five-Year Plan" was established as one of the seven strategic emerging industries, China's new generation of information technology has developed rapidly and gradually became the direction of deepening the application of information technology in various industries. From the perspective of development trends at home and abroad and the development and evolution path of blockchain technology, the development of blockchain technology and applications requires new generations of information technology such as cloud computing, big data, and the Internet of Things as basic support. At the

same time, the development of blockchain technology and applications Promoting the development of the new generation of information technology industry has an important role to play.

1.2 Public chain background

Public chain: Any customer can use it, any node can enter it. All nodes participate in consensus and read and write data together. Strong decentralized features. Examples: Bitcoin and Ethereum.

Consensus mechanism refers to a mathematical algorithm that establishes trust between different trust subject nodes and obtains rights and interests. It is provided to distributed network reference nodes for confirming data changes in the ledger caused by transaction actions, and can achieve ultimate consistency.

1) **Pow** (Proof of Work): PoW is poorly regulated and requires the entire network to participate in consensus calculations, and the performance efficiency is not high. Only 50% of the nodes in the entire network are allowed to fail. The PoW consensus mechanism adopted by Bitcoin;

2) **PoS** (Proof of Equity): It is required that network nodes must provide a certain number of token certificates. Similar to the replenishment of listed companies, node certificates hold more tokens, and the higher the probability of obtaining accounting rights. Improved performance and security, weak supervision, and still only allow 50% of nodes in the entire network to fail;

3) **DPoS** (share authorization certificate): In order to prevent large mining pools from monopolizing the entire network computing power, currency holding nodes elect a number of proxy nodes for verification and bookkeeping, similar to the board voting system. The advantages can effectively reduce the time to participate in the consensus verification, improve the speed of block generation, and perform similarly to the PoS mechanism in terms of supervisability and fault tolerance. EOS uses the DPoS consensus mechanism;

4) **PBFT** (Practical Byzantine Fault Tolerance): The PBFT system needs to

be deployed on at least 3f + 1 nodes. A maximum of f malicious nodes can tolerate Byzantine faults. The overall system status is determined by 2f + 1 nodes.

5) If in a trusted network environment such as a good state and no malicious nodes, the system platform can adopt more mature distributed one-of-a-kind solutions such as Raft, colleges and universities complete transactions and reach consensus, occupy less resources and have higher performance .

1.3 Blockchain 1.0

Blockchain 1.0 is the basic version of blockchain technology, which can realize programmable currency. It is a cryptocurrency application related to transfer, remittance and digital payment. Through this level of application, blockchain technology first plays a role in agitating financial markets. Large financial institutions such as the New York Stock Exchange, Goldman Sachs, Chi Mei, Citi, Nasdaq, etc. have all entered the blockchain field in the past year.

1.4 Blockchain 2.0

Blockchain 2.0 is programmable finance. It is a blockchain application in the economic, market, and financial fields, such as stocks, bonds, futures, loans, mortgages, property rights, smart property, and smart contracts

In addition to building a currency system, blockchain also has many application opportunities in the field of pan-finance. Based on the programmable characteristics of the blockchain, people have tried to add smart contracts to the blockchain system to form programmable finance, of which smart contracts are the representative.

The core of smart contracts is the use of procedural algorithms instead of people to execute contracts. These contracts require a combination and coordination of automated assets, processes, and systems. The contract contains three basic elements: offer, commitment, value exchange, and effectively defines a new application form, which makes

the blockchain expand from the original currency system to other application areas of finance, including in the areas of equity crowdfunding, securities trading, etc Began to gradually land applications. Traditional financial institutions are also vigorously studying blockchain technology with a view to combining it with traditional financial applications.

1.5 Blockchain 3.0

Blockchain 3.0 is the core of the Internet of Value. The blockchain can confirm, measure and store the property rights of each piece of information and bytes representing the Internet, so that assets can be tracked, controlled and traded on the blockchain.

The core of the Value Internet is to build a global distributed accounting system from the blockchain.It can not only record transactions in the financial industry, but can record almost anything that can be expressed in code form: sharing Right to use cars, status of lights, birth and death certificates, marriage certificates, education, financial accounts, medical procedures, insurance claims, voting, energy.

Therefore, with the development of blockchain technology, its application can be extended to any demanding field, including audit notarization, medical treatment, voting, logistics and other fields, and then to the entire society.

2 Design principles

2.1 Overall goal

Different from the well-known bitcoin, which uses a chain structure, the pow consensus mechanism of the blockchain requires a lot of energy to compete for the right to package the blocks, let alone let efficiency be slower.

So how to solve this problem in DAG?

Each newly added unit is not only added to one block in the long chain, but also to all previous blocks. Suppose that when you publish a new transaction, there are two valid blocks in the front, then your block will actively link to the first two at the same time. Each new unit in the DAG verifies and confirms the parent unit, The parent unit slowly reaches the genesis unit and includes the hash of its parent unit into its own unit. Over time, the blockchains of all transactions are interconnected to form a graph structure. If you want to change the data, it is not just a matter of several blocks, but the data of the entire block diagram.

Compared with DAG, this model has higher complexity and is harder to change, so the biggest hidden problem of Bitcoin and Ethereum is solved here, that is, there is no definite unchangeable final state. In theory, if there is enough computing power and sufficient block production speed to generate a longer hidden chain, the previous block can be overturned.

2.2 Scheme

Data structure

Tangle (Tangle) is based on Directed Acyclic Graph (DAG), rather than a continuous chain architecture, adding blocks on a regular basis. With KDAG, higher transaction throughput (through parallel verification) can be achieved, and no transaction fees are charged. With the continuous development of Tangle, more and more participants will initiate transactions, the entire system will become more and more secure and fast, the confirmation time will be shortened, and transactions will be completed faster and faster. Consensus mechanism

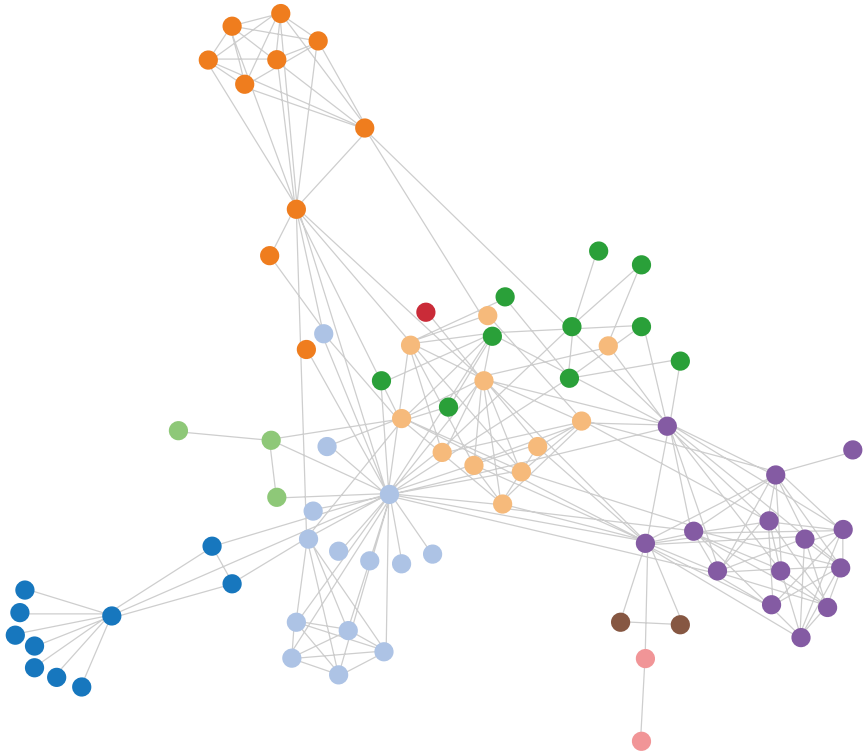
Blockchain consensus is accomplished through a very strict mechanism. Adding the next block to the blockchain requires multiple parties to compete and obtain block rewards or transaction fees. Because of this, consensus and transaction generation are separated and completed by a small group of people on the network, usually with a high threshold (not everyone will use a mining machine, and the increasingly concentrated computing power of the mining pool makes people Decentralization of

mind and heart), which will lead to further centralization.

In the KDAG system, every participant in the network can trade and actively participate in consensus. To be more specific, you directly locate two transactions (main transaction and branch transaction), and indirectly locate other transactions in the child tangle. In this way, verification can be performed synchronously, and the network can remain completely decentralized, without the need for miners to pass on trust and to pay transaction fees.

2.3 Deployment architecture

Any intermediate node can only access a small part of the information for each value transfer request. Therefore, the algorithm naturally provides good privacy protection in terms of transmission volume. However, each node needs to know the destination path request for each transmission in order to place it in the appropriate queue. If we also need to hide the transmission destination, we need to add a routing layer between the channel and the second layer network. At the routing layer, messages are encapsulated in the encryption layer. Encrypted data is transmitted through a series of network nodes called routing nodes. Each node is a routing node. Every time a node passes, it obtains the outermost routing path to reveal the next destination of the data. In this way, when the root path is read, the message arrives at the destination and thus forms a transmission route. The transmission situation of the entire network is shown in the following figure:



2.4 Technical standards

2.4.1 Availability

Through zero-knowledge proof, the impossible triangle can be solved to the maximum.

2.4.2 Throughput

Combining layering and sharding technology, the on-chain transaction TPS can reach 20,000.

2.4.3 Fault tolerance

Under the premise of not exceeding 51% attacks, the fault tolerance of the entire system can reach 99.99%.

2.4.4 Scalability

Adjust the scalability of the system through a scalable second-tier network.

2.4.5 Security

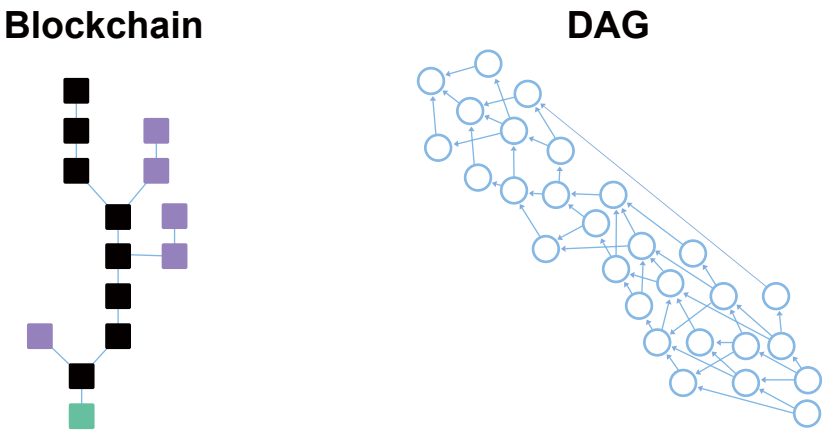
Through the exploration of unknown and deep learning, massive data analysis, benefit from uncertainty, and resist quantum attacks.

3 Technical implementation

3.1 Introduction to DAG

DAG: Directed Acyclic Graph, Chinese means "directed acyclic graph"

DAG was originally a commonly used data structure in the computer field. Due to the excellent characteristics brought about by the unique topology, it is often used to deal with a variety of algorithm scenarios such as dynamic planning, navigation to find the shortest path, data compression ...



The difference between traditional blockchain and DAG, simply put:

1. **Unit**: The block unit is Block (block), and the DAG unit is TX (transaction);

2. **Topology**: The blockchain is a single chain composed of Block blocks, which can only be written sequentially in accordance with the block time, like a single-core single-threaded CPU; DAG is a network of transaction units that can write transactions asynchronously and concurrently , As if multi-core multi-threaded CPU;

3. **Granularity**: Each block unit of the blockchain records multiple transactions of multiple users, and each unit of the DAG records a single user transaction.

Several issues of traditional blockchain technology

1) **Efficiency**: Traditional blockchain technology is based on Block. Bitcoin's efficiency has been relatively low. Due to the BlockChain chain storage structure, the entire network can only have a single chain at the same time. Blocks cannot be executed concurrently based on the POW consensus mechanism For example, Bitcoin generates one block every ten minutes, and six blocks can be confirmed, which takes about one hour. Ethereum has greatly improved, and the block production speed also takes more than ten seconds.

2) **Deterministic problem**: Bitcoin and Ethereum have a 51% hash power attack problem. The biggest hidden danger of the POW consensus is that there is no certain final state that cannot be changed. If a group controls 51% hash power and launches an attack The Bitcoin system is bound to collapse; considering the miners' group in the real world and the fast computing power of quantum computers, this danger is real.

3) **Centralization problem**: In the block-based POW consensus, miners can form a centralized mine group on the one hand, and miners who have obtained package trading rights have huge powers. They can choose which transactions enter the block and which transactions Without being processed, it is even possible to package only transactions that are in their own interest. Such a risk is now a fact.

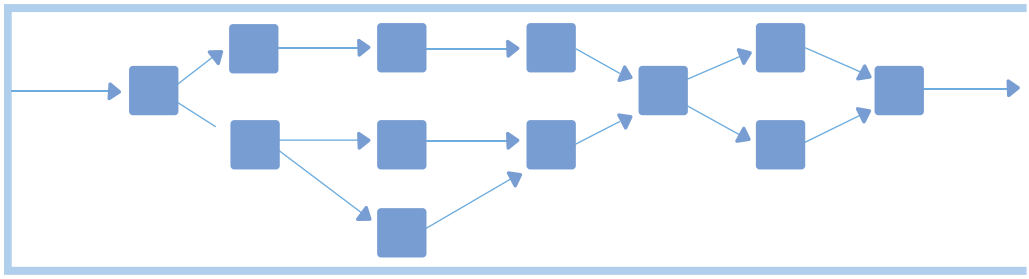
4) **Energy Consumption**: As the traditional blockchain is based on the POW computing power proof of work and reached a consensus mechanism, the energy consumption of bitcoin mining has been equal to the electricity consumption of a country in Argentina, and the IMF and multi-national governments are mining energy for virtual currencies. Consumption is critical.

3.2 Origin of DAG

The earliest introduction of the DAG concept as a consensus algorithm in the blockchain was in 2013. Bitcointalik.org was proposed by an Israeli Hebrew University scholar with the ID avivz78, which is the GHOST protocol, as a solution for expanding the transaction processing capacity of Bitcoin; The POS consensus protocol Casper described by Vitalik in the Ethereum Purple Book is also a POS variant based on the GHOST POW protocol.

Later, some people in the NXT community proposed using the DAG topology to store blocks to solve the efficiency problem of the blockchain. There is only one single chain in the blockchain, and the blocks cannot be executed concurrently. If the chain storage structure of the block is changed, the mesh topology that becomes a DAG can be written concurrently. With the same block packing time, N blocks can be packed in parallel in the network, and transactions in the network can accommodate N times.

At this time, the combination of DAG and blockchain is still based on the solution of similar side chains. Transaction packaging can be performed in different branch chains in parallel to achieve the purpose of improving performance. At this time, DAG still has the concept of blocks.



In September 2015, Sergio Demian Lerner published the article "DagCoin: a cryptocurrency without blocks" and proposed the concept of DAG-Chain. For the first time, the DAG network was upgraded from the coarse-grained block packaging to the transaction-based level. Papers, no code implementation.

The idea of DagCoin allows each transaction to directly participate in maintaining the transaction order of the entire network. After the transaction is initiated, it broadcasts the entire network directly, skipping the stage of packaging blocks, and achieves the so-called Blockless. This eliminates the time required to package the transaction. As mentioned earlier, the original combination of DAG and blockchain was to solve the problem of efficiency. Now, no package confirmation is required. The network confirmation is broadcast directly after the transaction is initiated. Theoretically, the efficiency has achieved a qualitative leap. DAG further evolved into a solution that completely abandoned the blockchain.

In July 2016, based on the creation post posted by the Bitcointalk Forum, IOTA was born, and ByteBall also debuted. IOTA and Byteball were the first real technical implementation of the DAG network and the most dazzling leaders in this field. At this time, The prototype of the DAG chain family, known as Block Less, is basically formed.

In a sentence: DAG is a new generation of blockchain facing the future. From a macro perspective of the graph theory topology model, it evolves from single chain to tree and mesh, from block granularity to transaction granularity, and from a single point transition. To concurrent writing; it is an innovation of the blockchain from capacity to speed.

3.3 HashGraph

Hashgraph is a Gossip gossip protocol consensus algorithm developed by Leemon Baird. All nodes randomly share their known transactions

with other nodes, so eventually all transactions can be passed to each node. Hashgraph is very fast (more than 250,000 transactions per second). Due to closed source and patents, HG is suitable for private chains or alliance chains. It will not be applied to public chains and get scale verification in the short term.

Hashgraph pioneered asynchronous BFT consensus in the public chain environment. A major problem with traditional BFT is that the message complexity is too high, it consumes the network bandwidth of the system, and it cannot deal with dynamic networks well. Here Hashgraph introduces the traditional Gossip Protocol, and adds unique innovations, plus a virtual voting mechanism, so that when consensus is needed, it will not cause a sudden large-scale messaging storm.

Hashgraph and Algorand improved the BFT application scenarios and conditions from different angles to enable BFT consensus to be applied to the public chain system.HG spread the hash map through gossip and virtual voting based on the hash map. Communication requirements are minimized, and local calculations ensure consensus efficiency.

The latest Hashgraph "public chain" business introduction book states that it is planned to switch to POS, and supports DOPS, and allows holders who do not run full nodes to choose agents and share revenue.

Hashgraph gathers the strengths of various companies, and has made great breakthroughs in scalability, security, and consensus-building costs, but the technology is difficult and has not been run in a large-scale public chain environment. As described in the white paper, then Hashgraph is enough to become an important milestone for exploration on the trusted Internet. It may break through the limitations of the blockchain and achieve a strong attempt to achieve the ultimate ideal of the blockchain from an innovative path.

3.4 DAG algorithm logic

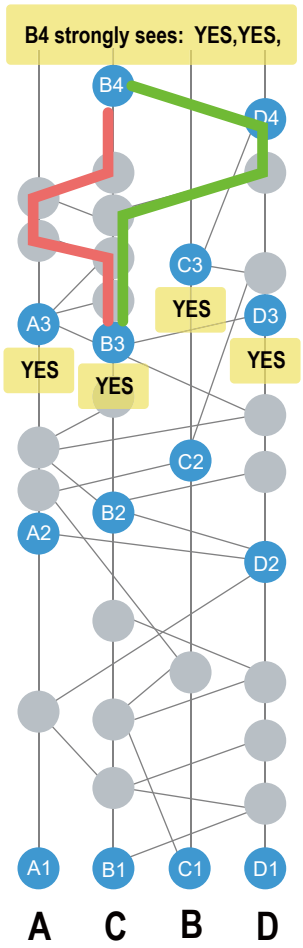
Suppose there are 4 nodes in the network (A, B, C, D), each node sends a transaction, the transaction is included in an event gossip to other nodes,

and a gossip will know the peer of this node Unknown transactions are randomly sent to other nodes, and each node maintains a complete graph. Through a voting algorithm, each event is time stamped. Before explaining the specific logic, let's take a look at the event data structure:

```
type Event struct
{
Transactions [][]byte //the payload
selfParent string
otherParent string
Creator []byte //creator's public key
Timestamp time.Time //creator's claimed timestamp of the event's creation
roundReceived *int
consensusTimestamp time.Time
}
```

The Transactions field is all the transactions contained in the Event, selfParent and otherParent are the hash of the Event parent Event, including the parent Event created by itself and the parent Event created by other nodes, Creator is the creator's public key, and Timestamp is the time when the Event was created Timestamp. RoundRecevied is the consensus of the event by the famous witnesses in several layers of round.

Let's take a look at the specific consensus process:



1) A, B, C, and D each create a rootEvent during initialization, and then B randomly selects a node (assuming D is selected), and then B sends to D all the events that it knows that D does not know (here Only the rootEvent created by B at the beginning), D creates a new Event (the SelfParent of the Event is the rootEvent of D, the otherParent is the rootEvent of B), and then D sends all the events (including the newly created) that it knows randomly. Create a new event for B, B, so B knows 4 events (two created by himself and two created by D), and D knows 3 events (not including B's last creation).

2) B then selects A randomly, and then sends 4 events that he knows to A, and A creates a new Event, which keeps growing and forms a graph structure.

3) Famous witnesses are determined. First of all, let's take a look at several concepts, see and strongly sees.

See means that there is an ancestral relationship between Events. Suppose there are Events (B2) and Event (A3). Assuming that B2 is the ancestor of A3 and A3 is the descendant of B2, then A3 can see B2.

Strongly sees is an ancestor-grandchild relationship between two events, and all the nodes in all paths connected by these two events and more than 2/3 of the nodes are regarded as one event strongly seeing the other event.

Witness is the first Event in a round (the rootEvents are all witnesses). The method for determining a round is that an Event can strongly see more than two-thirds of the witness in the current round, then the round of the event is increased by one.

The determining mechanism of famous witness is that the witness in the next round is voted on the visible witness in the previous layer, and the witness in the next round is used to count votes. The specific rule is that if a witness (A3) Visibility (B2) is visible, then vote YES. When all voted performance (A3, B3, C3, D3) voted YES for voted performance (B2), then voted performance (B2) is declared as famous, and then Next level

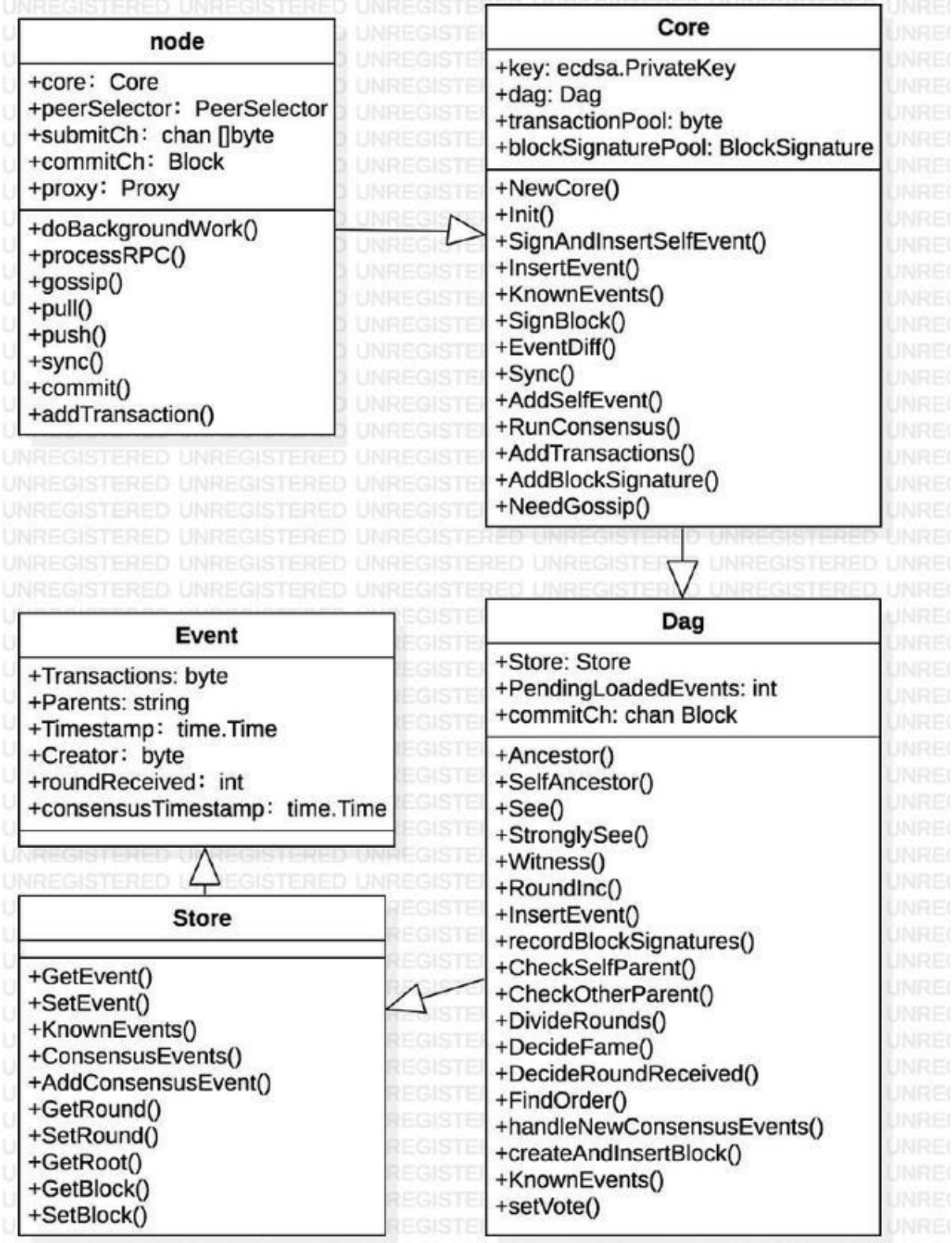
If witness (B4) is strongly visible to participating witness (A3, B3, C3, D3), then the vote is YES, then the vote is valid. When the number of valid votes exceeds 2/3, then the voted witness (B2) is selected as famous witness.

4) When the famous witnesses are determined, we can find a consensusTimestamp and a roundRecevied for events. The rule is that when an event (X) can be seen by all the famous witnesses in the next round, then the roundRecevied of the event (X) Is to see its round value of all famous witnesses, the event (X) consensusTimestamp determination rule is to find all the ancestor events of the famous witness and the events of the descendants of the event (X), and timestamp these events Sort, then find the middle timestamp as the consensusTimestamp of the event (X).

The events that are marked with consensusTimestamp are the consensus

events, and then the events are sorted according to roundRecevied.

3.5 DAG implementation



The basic theoretical part is as described above. Let ’s take a look at how the project uses the DAG algorithm as a consensus project. This project mainly implements the mapping of the same RoundReceived transaction to a block after forming a consensus in dag to form a linear area. Blockchain data structure, using ethereum's ledger structure.

The above class diagram mainly reflects the core class modules of DAG, modules such as crypto and network communication are not included.

Event is the basic data structure of DAG. As mentioned earlier, Store is responsible for the storage and management of data. The DAG class is the core of the DAG consensus algorithm. The main method is to implement the visible and strongly visible judgments. To get the known Event and vote.

The core logic of Core's entire node includes inserting events, merging events, and running consensus.

The node module is responsible for the gossip between nodes, processes the received gossip requests and returns, and is responsible for the ethereum's proxy communication with the nodes.

Combining the above, we look at the specific business process, from the initiation of a transaction to the entire logic of the transaction into a block.

1) We start an ethereum node and a DAG node, and they are connected through proxy.

2) Ethereum calls proxy to pass the transaction to the DAG node. The DAG node calls Core's AddTransctions to add the received transaction to the transactionpool.

3) The gossip heartbeat detects that a gossip needs to be initiated, it will create a new event, package all the transactions in the transactionpool into the event, and then insert it into the store.

4) Randomly select a peer, initiate gossip, pull events from the target peer that they do not know about the other party, and pass the events

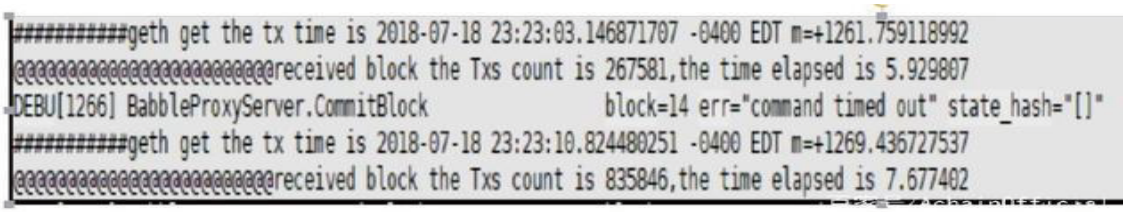
that they know to the other party.

5) Get the event returned by the other party, call Core's EventDiff to merge the events, and add events in the pedding area, then run RunConsensus to DivideRounds, call DecideFame to determine famous witnesses.

6) Through multiple gossip, famous witnesses are determined, and then FindOrder is called to decide roundReceived and consensusTimestamp, then sort.

7) Call handleNewConsensusEvents to take out the new consensus event, and package the transactions into a block, and empty the pedding area.

8) Return the block to ethereum, ethereum executes the transactions in the block, and returns the ledger status to the DAG. The DAG puts the ledger status into a block, signs it, and broadcasts the signature to other DAG nodes. Other DAG nodes receive the signature , Verify that it is correct, merge with your signature into the block, and broadcast your signature.



The picture above is our success after passing the 4-node test. We can see that one block contains 267581 transactions, which takes about 6 seconds, and the other block contains 835,846 transactions. It can be seen that in a limited In the node test environment, the number of gossips required to reach a consensus is limited. Therefore, by increasing the data load (number of transactions) of a single network IO (gossip), the block production interval is not seriously affected. The average block production time is 3- 5 seconds. But the number of transactions in the block has increased qualitatively.

4 Important conclusions

4.1 Consistent evolution of the ledger:

The core problem of the blockchain is to solve the problem of consistent evolution of distributed ledgers in a public network environment. We analyze from two issues:

1) Who will keep the books.

2) How non-bookkeepers verify that the bookkeeper is lying.

Take mature Bitcoin and Ethereum as examples, that is, the pow consensus algorithm, which uses the hashing power to compete for bookkeeping rights. The Merkle Tree or Merkle Patricia tree is used as the basic data structure of the ledger to make "non-bookkeepers" fast Verifying that the bookkeepers have lied, and the introduction of a punishment mechanism (payment of computing power), so that the bookkeepers have no incentive to account for bad debts from an economic perspective, and eventually make Bitcoin and Ethereum become the digital currency field. Successful practitioner of blockchain technology.

We know that there are three important data in a block header of Ethereum. They are the hash of the Merkle tree root node, including: state tree, transaction tree, and receipt tree. In the Ethereum network, the historical data of the world ledger on a node is proved by hashing power, so it is credible. When a miner successfully records a batch of accounts (so-called mining), it pays Computing power. After broadcasting, other nodes will verify whether the account is correct by the miners based on the local trusted ledger data. The core process is:

A. Based on your own local ledger (of course, you must strive to keep the local ledger to be the current longest chain, which is not described in detail here), the transactions in the block are executed in order. Then change the state of the local world ledger, as well as the receipt tree, etc., to form a new world state root hash and a receipt tree root hash.

B. Compare the local new world state with the result given by the bookkeeper. That is, the comparison of the three merkle root hashes. Consistent, then it is recognized that the bookkeeper has not lied, and the local ledger and the bookkeeper's ledger will continue to evolve in unison.

In this way, as the network has to be generated one by one, every ledger in the world has maintained a consistent evolution.

4.2 DAG main chain (hash map) logic:

The biggest difference between the DAG consensus and the pow consensus is that the order of transactions is determined. Ethereum is a single miner. The miners can decide which transactions are included in a block according to their own mining strategy (the level of transaction fees). Fight for bookkeeping rights through hashing power. In the hash graph, as in the analysis of the hash graph algorithm above, the hash graph uses the gossip protocol to agree on the order of transactions in a round and form a block. In short, these transaction orders are determined through negotiation. Once It cannot be changed. In this way, each ledger in the distributed system evolves down the world state according to the same transaction order, and the ledger consistency can be maintained in the end. Our specific process:

1) Hash graph consensus module, through gossip protocol to reach consensus on a round transaction ordering problem in the whole network, and form an intermediate state block, and then hand it to the ledger module to verify and execute the transaction. The Merkel root hash of the transaction tree is used to ensure that the agreed transaction order is not modified.

2) Ledger module verification. After executing the transaction sequence, a new Merkel root (stateRoot) of the world state tree will be generated, and then returned to the hash map consensus layer.

3) The hash graph consensus layer broadcasts the block's transaction

tree root, stateRoot, and other nodes perform the same verification and execution of the transaction sequence consensus just reached. By comparing stateRoot, you can know whether the evolution results are consistent, so that you can collect signature confirmations from each other. After a block is confirmed by the signatures of most nodes in the network, the transactions in the block can be considered to reach the final consensus.

5 Off-chain expansion

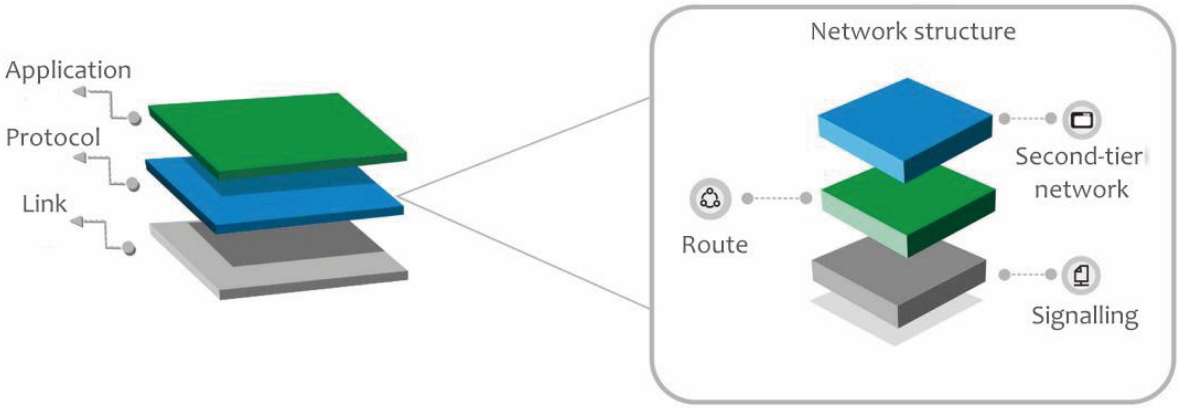
5.1 Technology Stack

As a comprehensive full-stack platform that can be built on existing or future blockchains, DAG includes a separable clean layered architecture and complex off-chain layered modules. This architecture simplifies the design, development, and maintenance of the system so that each node can easily evolve and adapt to change. A well-designed layered architecture with open interfaces that enables

Implement different functions on each layer as long as they support the same cross-layer interface. Each layer only needs to focus on implementing its own functions. Inspired by the successful layered design of the Internet, DAG uses the second layer of off-chain network technology stack, which can be built on different blockchains, called STACK,

It consists of the following layers in a bottom-up order:

- **Channel layer**: Universal status channel and side chain kit.
- **Routing layer**: Value-transfer routing with the best routing strategy.
- **Second-tier network**: development framework and runtime that support cross-applications.

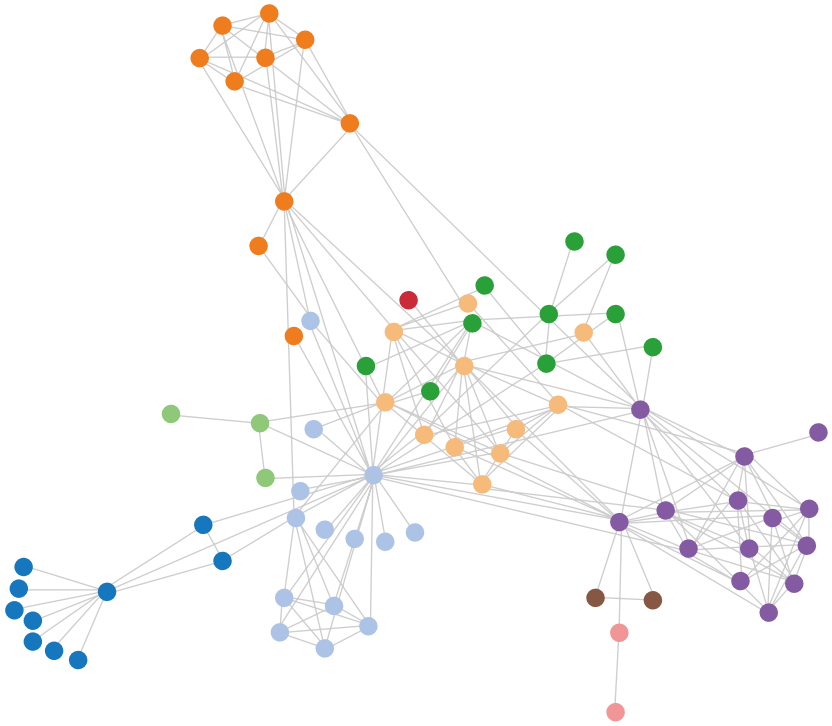


5.2 Failure recovery

Due to the adaptive and multipath nature of DAG, the algorithm is inherently robust and can prevent network failures. For example, when there is a non-responsive node, the DAG can quickly adapt and quickly remove the node to support the remaining available maximum throughput node.

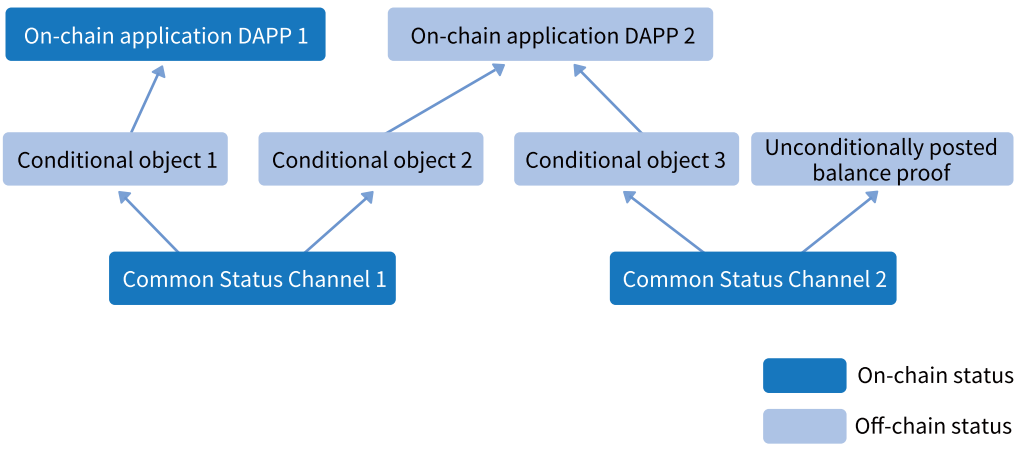
5.3 Privacy

Due to the multipath nature of DAG, any intermediate node can only access a small part of the information for each value transfer request. Therefore, the algorithm naturally provides good privacy protection in terms of transmission volume. However, each node needs to know the destination path request for each transmission in order to place it in the appropriate queue. If we also need to hide the transmission destination, we need to add a routing layer between the channel and the second layer network. At the routing layer, messages are encapsulated in the encryption layer. Encrypted data is transmitted through a series of network nodes called routing nodes. Each node is a routing node. Every time a node passes, it obtains the outermost routing path to reveal the next destination of the data. In this way, when the root path is read, the message arrives at the destination and thus forms a transmission route. The transmission situation of the entire network is shown in the following figure:



5.4 Second-tier network

To help consumers quickly build, operate, and use scalable off-chain decentralized applications, DAG innovates on higher levels of abstraction: application development framework (SDK) and runtime system. And a set of directed acyclic graphs (DAGs) that conditionally depend on the state, and describe how the graph integrates with the state channel network. In order to support scenarios other than simple P2P communication, we establish a conditionally-dependent directed acyclic graph (DAG) state, where edges represent the dependency relationship between them.



The figure illustrates the system model, which is generalized data exchange. The channel in this network is the only protocol contract consistent with the state on the chain. The results of these on-chain state calculations depend on one or more conditional DAG node objects (for example, conditional object 2), which operate completely continuously and are enforced on-chain. What we want to emphasize is that these conditional data objects are not just simple time-hash lock transactions, but can be applied as conditions off-chain.

The contract generated by the contract can be relayed through multiple hops to achieve the final transmission result. In summary, the first layer is a simple time hash lock to ensure that the relay chain resolves the transmission bottleneck in a reasonable time. The second layer locks down the conditions that affect the results. Through these two layers of relay, the value transfer of information between applications 1 and 2 is realized.

5.5 Off-chain Application Development Framework

Because the state dependency graph requires a dedicated development framework. According to the usability principle, DAG provides a complete SDK for creating, tracking, and providing a complete solution for off-chain status. Through the SDK, the speed of popularizing DAG's off-chain solutions has been improved, and it has a good reflection on the applicability of popularization.

Generally, we divide decentralized applications into two categories: simple applications and more complex multiparty applications. Microservices from real-world entities (such as data relay) and streaming media networks that pass through do not need to conditionally rely on other off-chain states, nor do they need a streamlined transport layer API on top of the routing layer. This situation can be met.

The multi-party application scenario has a relatively complex general structure. The SDK defines a set of design patterns and a common framework for developers to express conditional dependencies. We plan

to extend existing smart contracts, through annotation processing and dependency injection, so that dependency information can be explicitly expressed without interference. The compiler then processes the application code, extracts the declared off-chain objects, and generates a conditional dependency graph. The compiler detects invalid or unfillable dependencies and generates human-readable errors to help developers debug.

To help developers further reason about dependencies, the SDK will be able to serialize charts into common formats and make them easy to imagine and present. The SDK also provides a code generator that generates a set of "bridge methods" for interacting with smart contracts that have code available at compile time. The code generator parses the application binary interface (ABI), which specifies all callable functions in a signed smart contract, and generates bridge methods in the corresponding platform-specific language (such as Java). The main advantage is that this method is type-safe and can run faithfully in smart contracts, providing static and powerful compile time, and when the method is dispatched to a second-tier network runtime, it is checked before execution. On the network side, the lifecycle of a multi-party communication DApp is handled during runtime. It also provides a set of metadata for secure multiparty computing, capable of supporting complex user scenarios such as games, communications.

If the receiver fails, whether it is an abnormal stop or a Byzantine conflict, the runtime will forward the dispute to the on-chain state. When the client enters a hang, the runtime handles the exception and performs a corresponding rollback. When the client comes back online, the runtime synchronizes the local state with the on-chain state. We name the decentralized applications running on the DAG DApps.

For local off-chain state management, the conditional state diagram is bundled in the DApp by the SDK of the second layer network and passed to the off-chain runtime for execution. The runtime acts as the basic framework for creation, update, storage, and monitoring. The off-chain state is local to the KDAG network client. It can track internal logic and

cause applications running on it to perform DAG traversal of state updates. This method can also make up for the lack of relay routing capabilities.

The core of the second-tier network runtime is to bundle native virtual machines (VMs) to run smart contracts. Although we intend to deploy the second-tier network to many platforms, networks, mobile devices, and IoT devices, we enable developers to write common business logic only once and run the exact same on-chain smart contract code in each environment Instead of having to implement multiple variants of the same logic. By adopting this principle, our goal is to eliminate code duplication and ensure a high degree of consistency across platforms.

In terms of scalability, the second layer of the network can also be built with the language of the platform-specific part of the DApp, such as the user interface (UI) language that is most suitable for each platform (for example, Kotlin for Android and Swift for iOS) . Through these advanced technologies, the development efficiency of developers has been significantly improved.

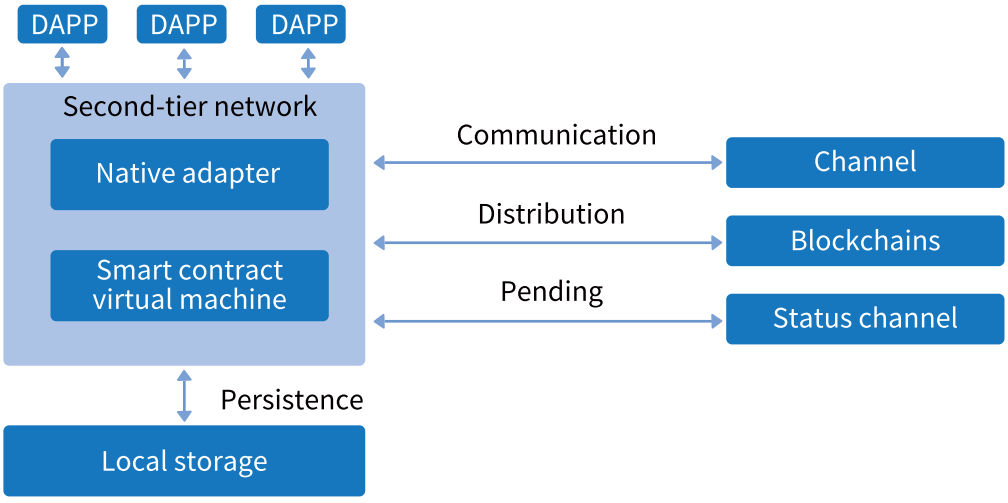
The runtime provides VM native bridging in different languages to enable platform-specific code to interact with the underlying business logic. For example, consider a DApp that eats chicken games running on iOS, and write a user interface in Swift and business logic written in Solidity. Of course, the UI layer needs to query the game state of the VM, and it will be able to bridge through Solidity-Swift. Because the contract's code is available at compile time, at that time, the code generator's SDK generates a bridge method called chicken.getstate, which is dispatched to the VM for the actual query. We take advantage of language's external function interfaces (such as JNI) to reduce the overhead of calling back and forth between smart contracts and native code. Developers can also use the same debugging and analysis tools to debug on-chain smart contracts in related scenarios. In order to truly replicate the state changes that may occur on the chain, in the second layer network runtime environment, the VM executes with the same bytecode as it does. If they are executed on-chain, then there are some

differences to be aware of.

The first difference is that the VM needs to update the storage state locally rather than on the blockchain. To achieve seamless and transparent interoperation between VMs and the rest of the second layer network, we will implement a set of storage backends that bridge the platform-specific API VMs.

The second major difference is that due to the long connection, the local virtual machine can be shut down unexpectedly at any time, such as a software error, a hardware failure, or just a power outage. Therefore, we need to implement robust logging, inspection, and submission protocols. The third small difference is that the logic of the calculation can be omitted because the execution is performed locally, and charging gas is meaningless. The bundled VM needs to be lightweight and high-performance so that it can run well on mobile and IoT devices. They often run on high-performance processors and large memory due to the limitations of mobile device capacity and battery life. The reason is that although we are currently embedding a lightweight Ethereum VM, we are working on a more common virtual machine implementation mechanism (for example, WebAssembly), with the goal of supporting more contract languages and other blockchains.

The overall implementation logic is shown in the following figure:



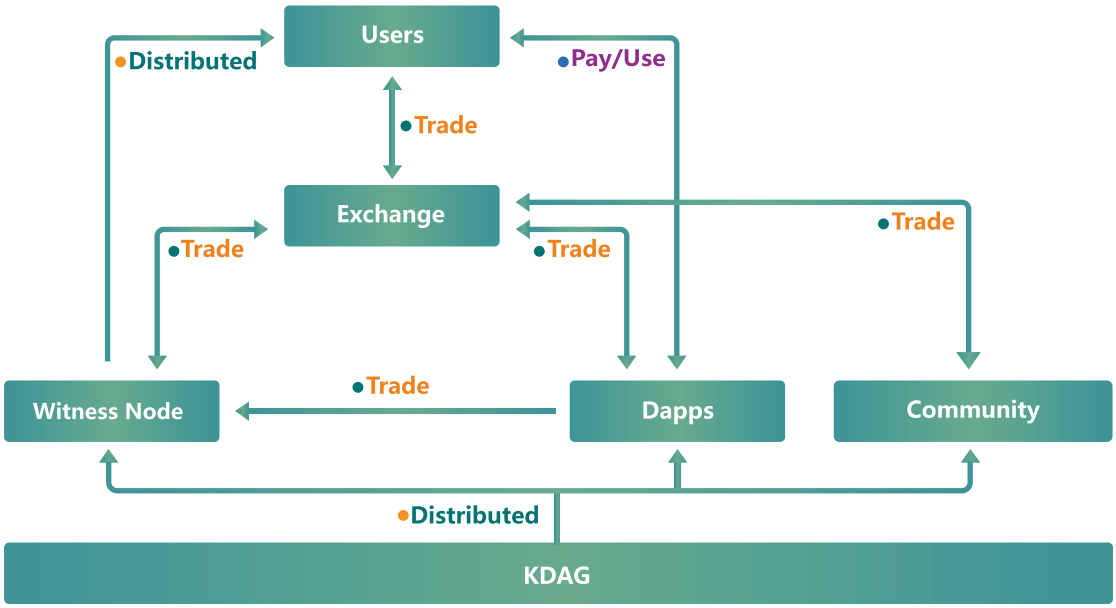
In our approach to VM implementation, we will apply the latest VM technology, including advance compilation (AOT) and just-in-time

compilation (JIT) to achieve near-preliminary compilation and performance through smart contract execution. Rather than interpreting its binary code, as most Ethereum virtual machines currently do, we talk about compiling the code into a lower-level binary machine language to make it closer to the native code. If the code for a contract is available at compile time (for example, the contract is already deployed on the chain), we statically compile ahead of time and link the binary with the rest of the application. For dynamic contracts, they are loaded at runtime, and we analyze them for frequently called functions (ie, "hot" code) and perform on-the-fly compilation. We believe that the combination of these two technologies will achieve a great balance between performance and energy consumption, which is critical for mobile and IoT devices.

5.6 Economic Model of Off-chain Scenarios

TOKEN is the token of the ecosystem on the DAG network and the main component of the system. This token will be used as the platform currency in the KDAG network ecosystem. It does not represent any equity, participation, rights, ownership in any way, nor does it allow token holders to bear any expense commitments, income, profits or investment returns, and therefore does not constitute a security element. TOKEN can only be used on KDAG in Singapore or any relevant jurisdiction.

The following is TOKEN's cryptoeconomic mechanism. The design is based on the principle of a good cryptoeconomic model (token model) should provide additional value and realize a new dynamic circulation mechanism.



5.7 Transactions in the off-chain ecosystem

Any off-chain solution is also trading while gaining scalability. In the following scenario, we describe two basic attributes in an off-chain ecosystem: scalability and liquidity.

5.8 Scalability and Liquidity

The off-chain system first obtains scalability through network liquidity transactions. For example, in the two-party status channel, two related parties can safely send high-speed payments to each other without touching the underlying blockchain, because they deposit tokens into on-chain contracts at the beginning. With a simple hedging mechanism, this work is fine for the end user, because the end user can simply deposit money. Have liquidity to open state channels and enjoy scalable dApps. However, this is a major challenge for off-chain data operators.

Because the operator must lock different tokens through each state channel to improve the overall throughput of the system. Therefore, by operating reliable and scalable chain services and providing corresponding technical capabilities, DAG enables operators to rely on the DAG network and enjoy the convenience while not having to sacrifice

assets. At the same time, in order to ensure the reliability of operators, the off-chain system has also established the necessary review mechanisms to prevent centralized damage to the network and privacy leakage.

5.9 Scalability and Availability

The off-chain second-tier network improves scalability by introducing application state, which imposes an unrealistic "always-on" responsibility on users because the second-tier network should always be available for on-chain disputes. For example, in order to prevent double spending, through a state channel, if a party enters the chain, the counterparty may be hacked or act maliciously and try to solve an old but more favorable state for itself. This data availability issue is even more critical in the sidechain channel of the proposer and requires independent monitoring and verification. Such security issues should be carefully reviewed and guarded accordingly.

It is even more critical in the machine-to-machine communication scenario where the IoT device is located, as the device is unlikely to be always online. Therefore, it is important to design a proper mechanism,

which guarantees the data availability of the off-chain platform. Solving this challenge requires systematic thinking and copper considerations of the entire off-chain ecosystem and existing solutions. By providing decentralized, efficient, simple and flexible important features and security, the availability of data can be solved.