DC Technology White Paper

1. Overview

1.1. New Opportunities in Decentralized Systems and Cloud Computing Services

Decentralized ledger technology refers to transaction accounting being completed by multiple nodes distributed in different locations. These nodes participate in supervising the legality of transactions and also jointly attest to them. Blockchain is a form of distributed ledger technology, spread across a peer-to-peer network and managed by it. As a distributed ledger, it can operate without a central server and maintain its data quality through database replication and trust computing. However, the structure of blockchain distinguishes it from other types of distributed ledgers. Data on the blockchain is grouped and organized into blocks, which are connected in chronological order to form a chain, and secured using cryptographic technology. Based on this distributed data structure, blockchain can bring consensus to a zero-trust decentralized network.

Whether it can solve practical social problems and truly enhance the efficiency of social production activities is a key criterion for the sustainability and popularity of the technology. The impact of information technology on social production relations and efficiency has been verified and is in a process of continuous deepening, with storage and computing as the two core foundational elements of the information technology productivity revolution. Therefore, blockchain technology needs not only to provide mechanisms for consensus and de-trust (i.e., "value decentralization") but also to provide decentralized infrastructure for the two core production elements of storage and computing (i.e., "storage decentralization" and "computing decentralization"). Looking at the existing mainstream decentralized consensus mechanisms, blockchain technology is widely used in trust aggregation (production relations) infrastructure, often accompanied by significant consumption of computational power and storage resources, while support for decentralized storage and computing is relatively lacking.

Existing storage and computing scenarios are widely carried on centralized cloud computing platforms, where computing and storage services are the most important components of the cloud service market. In centralized cloud services, computing power and cloud storage are concentrated in centralized servers and disk arrays, allowing users to connect to the cloud at any time, from anywhere, through any network-enabled device to access computing power and easily retrieve data. However, this centralized service model has issues with service stability, high network bandwidth costs, and limited data transmission capacity. More critically, when centralized platforms encounter operational issues, it can affect nearly all service recipients—the larger the scale of centralized cloud computing, the greater the security concerns related to operations. This project aims to start with blockchain's trust aggregation, innovate through technology, and enter from decentralized cloud service scenarios to achieve a trustworthy, reliable, efficient, and ubiquitous decentralized cloud service ecosystem.

1.2. DC Overview

DC(Decentralized Cloud) is essentially DePIN (Decentralized Physical Infrastructure Networks). Unlike existing cloud computing, DC is decentralized, without a central authority, and not controlled by any individual or institution. Anyone can participate and run nodes anywhere, with automatic networking. With the consensus services provided by blockchain, each node can independently offer cloud computing

services and serve as backups for other nodes in the DC network. Compared to traditional cloud computing, DC is not just a cloud or a cluster of clouds, but the cloud itself. DC provides developers with a highly convenient decentralized network infrastructure, enabling them to use their traditional internet development experience to develop decentralized applications. Technically, DC implements a full-stack ecosystem from the incentive layer (consensus) + network layer + persistence layer (storage) + application layer (computing). The user experience of DApps based on DC is like using traditional internet applications.

2. DC's Technical Goals

DC's main goal is to provide users with decentralized cloud services, and to bring revolutionary changes to the Web3.0 industry through a new technological approach.

- Eliminate the barriers for users to use Web3.0 applications
- Eliminate the barriers for enterprises and developers to develop Web3.0 applications
- Address the current performance bottlenecks in the Web3.0 business
- Address the data storage bottlenecks in the current Web3.0 ecosystem
- Break through the current business scenario limitations of Web3.0 applications
- Provide infrastructure for personalized applications in the AI era

3. DC Consensus Mechanism

DC network aims to provide a secure and efficient blockchain solution by integrating TEE technology with NPOS consensus mechanism, overcoming the limitations of traditional technology in processing speed and data storage capacity. This innovative combination of consensus mechanisms provides a solid foundation for the development and application of the DC network.

3.1. Trusted Execution Environment (TEE)

Trusted Execution Environment (TEE) is a security technology based on hardware security modules that enhances the security of computing and communication systems. The development of TEE technology has evolved from traditional hardware chip security models to more advanced implementations, such as Intel's SGX technology and the ARM-based open-source framework TrustZone. These technologies ensure security through the following five core specifications:

a. Endorsement Key
The endorsement key must be randomly generated and unchangeable. The private key must be securely stored and inaccessible except through a designated interface in a black-box manner. The public key is used for authentication and encryption of sensitive data to be sent.

b. Secure Input and Output
Input and output refer to the interaction between the user and the system, including pathways such as keyboards, peripherals, and network interfaces. Secure input and output mean that there is a protected path between the system user and the accessed process.

c. Memory Curtaining
Memory curtaining extends general storage protection technologies to provide completely independent storage areas. Even the operating system itself does not have full access to the curtained area, ensuring that runtime data is secure even if an attacker controls the operating system.

d. Sealed Storage
Sealed storage protects private information by bundling it with the platform environment configuration information used by the user. This means that data stored in sealed storage can only be read in the same secure environment.

e. Remote Attestation
Remote attestation involves generating a software certificate for the current system using the endorsement key. Any changes on the system can be detected and verified remotely by the authorized party through the certificate, ensuring the security and trustworthiness of the system's execution logic.

DC's local consensus and data security schemes are primarily server-side, utilizing Intel SGX-based TEE technology as its initial solution. Compared to complex algorithmic solutions, TEE-based solutions are simpler and more effective in implementation logic. With the rapid development of technology, TEE has strong development momentum and a technological ecosystem, supporting trusted execution of complex computational logic and meeting the full capabilities of cloud computing, thus enabling DC to create a decentralized cloud service ecosystem.

3.2. NPOS Consensus

NPoS (Nominated Proof of Stake) is a consensus algorithm designed by Polkadot based on the PoS algorithm. Validators run nodes to produce and confirm blocks, while nominators can stake their tokens to gain nomination rights and nominate trusted validators to earn rewards. DC integrates Polkadot's NPOS algorithm. The validator election algorithm is the core of the NPoS mechanism. The election process must be fair, representative, and secure. The Phragmén algorithm is designed to ensure each election maintains these qualities, ensuring that no validator pool is either over-represented or under-represented but proportional to their stakes.

In NPoS, each validator node receives the same amount of rewards, and nominators distribute the rewards earned by their supported validator nodes according to their stake proportions in those nodes. This mechanism incentivizes nominators to seek validators with fewer staked tokens, thereby increasing their proportional stake and advantage in the distribution. This ensures that validators have roughly equal stake amounts, keeping the network sufficiently decentralized.

3.3. DC's Combined Consensus Mechanism

DC's consensus consists of off-chain and on-chain parts. Off-chain consensus is primarily responsible for the operational logic consensus of DC's cloud service nodes, including mechanisms for nodes joining or leaving the network, staking, reporting, data synchronization between nodes, and external services. On-chain consensus is used to maintain DC's account system, incentive mechanisms, and data indexing mechanisms. Off-chain consensus is mainly implemented based on TEE technology, which ensures the security and trustworthiness of the operational logic of cloud service nodes, preventing unauthorized program-running nodes from joining the network and ensuring that DC's cloud service nodes provide services according to established business logic. On-chain consensus is mainly based on the NPOS consensus mechanism, ensuring the openness, fairness, reliability, and free participation of DC's account system, incentive mechanisms, and data indexing mechanisms.

4. DC Architecture

DC is a P2P peer-to-peer network that can scale horizontally indefinitely, with nodes free to enter and leave. By combining TEE technology with the NPOS consensus mechanism, DC aims to provide a secure and efficient blockchain solution, overcoming traditional blockchain technology's limitations in processing speed and data storage capacity. This innovative combination of consensus mechanisms provides a solid foundation for the development and application of DC.

DC mainly consists of DCChain and DCNode. DCChain is a blockchain network responsible for maintaining DC's account system, incentive mechanisms, and data indexing mechanisms. DCNode is a cloud service node, and multiple DCNodes form DC's cloud service network, providing decentralized cloud services to DC users through the account system, incentive mechanisms, and data indexing mechanisms provided by DCChain.

4.1. **DCChain**

DCChain is a public blockchain developed on Substrate, adopting a unified account system that integrates the accounts and signatures (i.e., public and private keys) of both Ethereum EVM and Substrate systems, facilitating compatibility with Ethereum wallets and DApp integrations. It serves as the brain of DC, creating a highly trustworthy, fair, reliable, and transparent consensus mechanism for all participants in the DC ecosystem, providing the following consensus capabilities for the entire DC network:

4.1.1. Maintaining the account system for DC

- Provides identity consensus for users throughout the network ecosystem, offering user identification and labeling for all business operations within DC
- Supports a unique account and password system in the DC network ecosystem, aligning user experience with existing internet application logins;
- Users own the sovereignty of their accounts in DC, which can be freely transferred. To distinguish from ERC20 format accounts on the blockchain, user accounts used for password logins in the DC network are referred to as NFT accounts;
- When users log into the network ecosystem using their DC NFT account and password, DCChain provides an index from the NFT account to the user's basic information storage and backup nodes, directing login requests to the corresponding DC node to provide login responses. Upon successful login, the private key of the corresponding DCChain application signature account is extracted through the logged-in NFT account, enabling users to conduct subsequent business operations.

4.1.2. Serving as the incentive layer for DC

DC rewards participants based on their contributions with DCT tokens.

a. Contributors to the DC network

- Cloud service space providers (operating DC cloud service nodes)
- DCChain validators and nominators
- Developers of DApps based on DC

b. Sources of rewards for contributors

- DCT tokens issued by DC based on the inflation rate
  Awarded to blockchain validators (including nominators) and cloud service space providers

- DCT tokens spent by DApp users to exchange for cloud service space
  Proportionately awarded to DApp developers and cloud service space providers

c. Ensuring fairness in the distribution of benefits

- DC records data such as network contributors, contribution levels, issuance rules, and user DCT token consumption information on the blockchain, ensuring the openness, fairness, reliability, and free participation of the DC economic ecosystem.

d. Punishment mechanisms for malicious behavior

- DC can track malicious behavior within the network through data recorded by DCChain, imposing penalties on those who engage in such activities, primarily including disruptions to chain consensus and damage to stored data.

4.1.3. Serving as the data indexing layer for the DC network

DCChain records data indexing information in DC on the blockchain, providing data indexing services for the DC data storage layer, ensuring the reliability and availability of the DC data storage layer. Data indexing mainly includes:

- Data storage location indexing
  DApps can quickly locate the list of cloud service nodes that store file or database information based on the CID of the file or database, allowing for rapid data retrieval.
- Cloud service node indexing
  DApps can quickly obtain basic information about cloud service nodes from DCChain, including network status, storage capacity, and accessible internet addresses.
- User information indexing
  DApps can quickly locate the cloud service node containing user information based on the NFT account, providing fast guidance services for user NFT account password login logic. For more details, refer to [DC network's account system](## 4).

4.1.4. Serving as the smart contract layer for DC

DCChain supports the development and deployment of smart contracts, providing support for DC's DApps. DCChain is fully compatible with EVM, allowing any DApp running on Ethereum to be seamlessly migrated to DC.

4.2. DCNode

The cloud service node network of DC is composed of numerous self-organizing DCNodes, responsible for providing cloud services to DC users. Each DCNode operates in a TEE environment, ensuring absolute privacy and security of data. DCNode provides cloud services to DC users through the account system, incentive mechanisms, and data indexing mechanisms provided by DCChain. The main functions of DCNode include:

1. Basic Data Storage
   DCNode utilizes the IPFS network to provide file data storage services for users. Users can store files in the cloud service space provided by DCNode through DC's DApp. DCNode provides the necessary

storage space based on user demands and ensures that there are no fewer than three copies of data backups in DCNode, guaranteeing the reliability and availability of data.

2. Database Storage
   DCNode offers a unique multi-threaded distributed database storage service, allowing DApps that support multi-terminal concurrent access to store databases in the cloud service space provided by DCNode. DCNode provides the necessary storage space based on user demands and ensures that each database has no fewer than three copies of data backups, guaranteeing the reliability and availability of database data.
3. Social Routing
   DCNode provides a unique social routing service, allowing DApps to encrypt and store users' basic information, social relationships, and social process data in the cloud service space provided by DCNode, facilitating developers to quickly develop social DApps similar to Facebook, Twitter, and LinkedIn.
4. Message Caching
   DCNode offers up to 10 days of peer-to-peer message caching services, ensuring that users do not miss messages from other users or platforms when they are offline.

5. Networking Mechanism of DC Network Cloud Service Nodes

DCNode is a self-organizing node that operates in a TEE environment. DCNodes are interconnected through a P2P network and use the consensus mechanism provided by DCChain to synchronize data, ensuring the consistency of the entire DC's data.

5.1. Node Onboarding Mechanism

1. Run the DCNode node program to obtain the node's PeerId and the node's Erc20 account.
   (Note: the private key of this Erc20 account is encrypted and stored in the node's TEE environment, inaccessible to anyone).
2. Through the DC portal or an Ethereum-compatible wallet, transfer a small amount of DCT to the node's Erc20 account for transaction fees when submitting transactions to the DCChain.
3. Bind the node's PeerId to an Erc20 account owned by the user for staking, and transfer the required amount of DCT for staking to the bound Erc20 account.
4. After detecting successful staking, the DCNode node program automatically initiates an onboarding application carrying TEE remote attestation and waits for the confirmation of DCChain's consensus mechanism.
5. A random group of other networked DCNodes, upon detecting a new node's onboarding application, automatically initiates verification of the new node's TEE remote attestation. If this attestation fails, a report request is submitted to the blockchain.
6. The new applicant DCNode re-initiates the onboarding application after waiting for 300 blocks. DCChain will judge whether the new node's application has passed the TEE remote attestation verification by other networked nodes. If passed, the new node successfully joins the network; otherwise, it fails and a corresponding amount of staked DCT tokens are deducted.
7. Once successfully networked, the DCNode begins providing cloud services to DC users and takes on the role of initiating verification tasks carrying TEE remote attestations.

5.2. Node Offboarding Mechanism

1. Node owners can initiate an offboarding application at any time through the DC portal. Upon receiving the DCNode's offboarding application, DC automatically starts the data backup task on that node (the node can go offline as the DC network itself has multiple data backups).
2. If the node owner directly stops the DCNode node program and does not initiate an offboarding application within 7 days, DC will deduct a certain amount of the staked DCT tokens as a penalty.

6. Account System of the DC Network

DC users' NFT accounts and passwords are linked to their application signing private keys, meaning each DC NFT account and password pair corresponds to a specific signing private key or mnemonic phrase. To support cross-terminal login to Web3.0 applications, DC implements the following:

1. Store the private key information corresponding to the NFT account and password in DC.
2. The private key information and basic user information stored in DC are completely private and cannot be decrypted by anyone other than the user.
3. When logging in with an NFT account and password, users can safely and quickly extract the corresponding application private key information from DC.
4. The application private key information stored in DC can only be extracted by users who possess both the NFT account and password, and the entire login and information extraction process is fully encrypted.

The specific implementation process is as follows:
Steps consistent with existing Web3.0:

5. Users locally generate an application account, including a private key or derived mnemonic phrase.
6. Guide users to record the mnemonic phrase.
7. Complete the creation of a decentralized chain-based application account, simultaneously deriving an ERC20 format chain account through the mnemonic phrase.

DC network NFT account-specific steps:

8. Guide users to enter basic information such as NFT account and password.
9. In DC, check whether the NFT account has been registered. If registered, guide users to enter another NFT account.
10. Check if the user's chain account has a balance. If there is no balance, guide users to deposit funds into the chain account (it is recommended that application developers establish a faucet service during the early stages of application promotion to automatically transfer funds needed to bind the account for new registered users).
11. The chain account subscribes to cloud service space for the user's application account on DCChain.
12. Submit the binding relationship between the application private key or mnemonic phrase and the NFT account to DC (if during the submission process, the account is bound by someone else or there is insufficient balance, return to step 4).
13. Users can directly log in on any terminal using the NFT account and password.
14. Users can modify their login password as needed.

6.1. DC Account Security Design

Since the node list that stores user account information can be retrieved through the NFT account, it is essential to secure the user information stored in DCNode. DC ensures the security of user information

through the following measures:

1. Encrypt blockchain account private keys using AES256
   Utilize the NFT account and password combined with a series of encryption processes to generate an AES256 key, which is uniquely bound to the blockchain account private key. This ensures that only those who know both the NFT account and password can decrypt the relevant information of the bound application account private key.

2. The encrypted blockchain account private keys are only stored in networked DCNodes
   The client SDK ensures that the encrypted application account private keys are stored only in networked and online DCNodes. Since each networked DCNode has been certified through TEE-based remote attestation technology, ensuring that the programs running on the nodes are unmodified, legitimate, and operate within a TEE environment, the data stored on networked nodes is secure, private, and reliable.

3. DCNode encapsulates the encrypted blockchain account private keys using TEE
   DCNode uses TEE encapsulation technology to secure the encrypted blockchain private keys stored on the nodes, ensuring that even the physical device owners of DCNode cannot extract the encrypted blockchain private keys.

4. The key used in the login process encryption does not contain complete account password information
   The key parameter extraction algorithm for the NFT account in the request information is securely designed:

   - By hashing the account and part of the password separately, then splitting and recombining them, multiple hashing ensures that it is impossible to reverse-engineer the account password from the key.
   - Part of the key is used in the calculation, and after hashing, part of it is taken for a second hash to ensure it cannot be brute-forced to obtain the exact NFT account and password.

5. The entire login process is encrypted, ensuring communication security

   - When the client sends a login request to DCNode, the communication process is encrypted using the public key of DCNode, only the requested DCNode program can decrypt it (all private keys of DCNode are randomly generated, encapsulated using TEE technology, and stored securely, inaccessible to node owners).
   - Feedback data is also encrypted using the key negotiated during the login request process, ensuring that only the requester can decrypt it.

6. DCNode controls the number of failed login attempts within a window, rejecting login requests that exceed this limit
   DCNode controls the login failure window for user-initiated NFT account and password login requests, ensuring that the same NFT account can only make a limited number of requests within a certain period, preventing hackers from extracting the encrypted blockchain private key through brute force methods.

7. Introducing the Bcrypt algorithm to slow down the generation speed of the AES256 key, preventing offline brute force attacks on the request key
   During the process of generating the AES256 key with the NFT account and password, Bcrypt is

introduced and modified to ensure that the encryption salt can be derived from the NFT account password using a specific algorithm. Using the slow hashing principle of Bcrypt (cost set to 12, which can be increased as device performance generally improves), hackers cannot perform offline brute force attacks even if they obtain the encrypted blockchain private key.

7. DC Security Mechanisms

The security of DC is divided into two main parts: one is the security mechanism of DCChain, and the other is the security mechanism of the DCNode network.

7.1. DCChain security mechanisms

DCChain security mechanisms mainly include the following aspects:

- Consensus Mechanism
  DCChain uses the NPOS consensus mechanism, which maximally attracts ecosystem users to participate in the network consensus, ensuring the security of DC.
- Governance Mechanism
  DC has a decentralized governance structure, allowing token holders to vote on network upgrades and changes. This mechanism can respond quickly to security threats and vulnerabilities, allowing timely repairs and upgrades to maintain network security.
- Runtime Upgrades
  DCChain allows runtime logic of the blockchain to be upgraded without a hard fork. This means that if security vulnerabilities are discovered or improvements are needed, updates can be deployed quickly and seamlessly, enhancing the overall network security.
- Resistance to Denial of Service Attacks (DDoS)
  DCChain, developed based on Substrate, is designed to resist DDoS attacks through various mechanisms (such as limiting request rates and selectively accepting connections), mitigating potential denial of service attacks.

7.2. DCNode network security mechanisms

DCNode network security mechanisms mainly include the following aspects:

- Node Onboarding Staking Mechanism
  All DCNodes must stake a certain amount of DCT tokens before networking. If a node acts maliciously or damages data, it will be penalized.
- Node Code Logic Control Mechanism
  The code of DCNode runs entirely in a TEE environment, so DCNode can only run code logic certified by DC, ensuring that DCNode's business logic is transparent, controllable, and tamper-proof.
- Node Onboarding Cooling-off Period Mechanism
  DCNode nodes must wait for a cooling-off period of 300 blocks after sending an onboarding application before they can network normally. During the cooling-off period, new DCNode nodes undergo TEE remote attestation verification by other networked nodes, and only after all verifications are passed can they join the network.
- Node Signature Private Key TEE Encapsulation Mechanism
  The signature private key used by nodes for blockchain interactions is sealed using TEE technology,

making it inaccessible to anyone, ensuring that all digital signatures of networked nodes are executed according to predetermined code logic, free from any interference.

- User Transaction Security Mechanism
  All transactions involving user transactions must carry a user's digital signature and be verified before being submitted to the blockchain, ensuring the security of user transactions.
- TEE Remote Attestation Mechanism
  All transactions involving DC security must carry the TEE remote attestation of DCNode nodes and be submitted to the blockchain, subject to remote attestation verification by other DCNode nodes.

8. DC Data Storage and Security Mechanisms

The file storage mechanism of DC mainly includes the following aspects:

- Data Storage and Transmission Mechanism
  DC provides data transfer and storage services to users through the IPFS protocol.
- Data Storage Location Index
  The binding relationship between user data storage information and PeerId is recorded on DCChain, facilitating rapid retrieval.
- Data Storage Backup Mechanism
  DC ensures that user-uploaded data is backed up in at least three copies (user account information is backed up in at least five copies) to ensure data reliability and availability.
- Storage Permission Control Mechanism
  DC manages data based on ownership, ensuring that data can only be manipulated by users who have sovereignty over it.
- Data Security Check Mechanism
  Every DCNode monitors the data stored on the node. If any data is found to be tampered with or damaged, it will automatically notify the network and initiate a task to restore data from backups.
- Data Encryption Mechanism
  All sensitive data in DC is encrypted and invisible to anyone except the user who owns the data sovereignty.
- Controllable Data Storage Location
  Users in DC can specify the location of data storage to ensure that data is stored on trusted nodes.
- Data Deletion Mechanism
  Users in DC can delete their data at any time to ensure data privacy.
- Controllable Number of Backups
  DC allows users to increase the number of backups for important files individually.

9. DC Decentralized Database Mechanism

After modifying ThreadDb, combined with the indexing function of DCChain, DC has created a relational-like decentralized database storage mechanism, which mainly includes the following aspects:

- Decentralization
  Utilizing the decentralized nature of IPFS, data is allowed to be distributed across multiple nodes, enhancing data persistence and resistance to censorship.
- Data Synchronization
  It supports real-time synchronization of data between different devices or users, which is particularly important for building collaborative applications and social networking applications.

- User Control
  Users have full control over their data and can choose the scope of data to be shared and synchronized. This differs from traditional centralized services where users typically have limited control over their data.
- Privacy and Security
  Data can be encrypted for storage, ensuring user privacy even in public or semi-public settings.
- Multi-node Backup
  DC ensures that database data is backed up in at least three copies across the network, ensuring data reliability and availability.

10. DC Decentralized Social Capabilities

The purpose of creating decentralized social capabilities in DC is to allow users to freely establish social relationships within DC and share and transmit data within these relationships, free from the control of centralized social platforms. The decentralized social capabilities of DC mainly include the following aspects:

- User Information Storage
  DC encrypts and stores users' basic information, social relationships, and social interaction data in DCNode nodes, ensuring user privacy and security.
- User Routing
  DC provides fast user routing services through the data indexing mechanism provided by DCChain, ensuring users can quickly find their friends.
- Topic Routing
  Users can quickly route to the list of DCNode nodes associated with a social topic through DC and quickly find social topics of interest.
- User Social Interaction Data Storage
  Whether based on social topics or interactions with friends, all social interaction data is encrypted and stored in DC, inaccessible to unauthorized personnel.
- Message Caching
  DC provides a point-to-point message caching service for up to 10 days, ensuring users do not miss messages from other users or the platform while offline.

11. Service Provision Mechanisms of DC

DC supports DApps based on both smart contract and non-smart contract models, providing a comprehensive set of service provision mechanisms, which mainly include the following aspects:

- Support for DApps Based on Smart Contract Model
  DC is compatible with Ethereum smart contracts, allowing any Ethereum DApp to be seamlessly migrated to DC.
- Support for DApps Based on Non-Smart Contract Model
  DC allows anyone to develop decentralized server-less DApps, which are serviced through DC's cloud service nodes.
- Cloud Service Node Capabilities
  To enable ordinary developers to quickly develop DApps, DC provides developers with RPC interfaces similar to centralized cloud services and highly integrated SDKs. These cover functionalities such as

user information management, file storage, database synchronization, message communication, and social services.

12.DC Network Cloud Service Consumption Mechanism

DC operates on a user-pay model, where users can pay for services using DCT tokens. Depending on how users use DC, there are different payment methods, mainly including the following two:

- Direct Blockchain Interaction for Business Operations
  In this mode, users can pay for their business operations using DCT tokens, including direct calls to interfaces on DCChain for DCT token transfers or utilizing on-chain smart contract functions.
- Interaction with Cloud Service Nodes via DApps
  In this model, users do not need to interact directly with DCChain. All business operations interact with cloud service nodes. For business requests that need to be recorded on the blockchain, after generating an Ed25519 signature, the cloud service nodes submit the blockchain request on behalf of the user. In this case, users do not have to pay any blockchain interaction fees. However, before interacting with DCNode, users need to subscribe to DC cloud service space using DCT tokens for a specified period, similar to a monthly or annual subscription model. In this way, interactions with DCNode are real-time and deterministic, without waiting for blockchain confirmation times or paying blockchain interaction fees.

(Note: DC is designed so that online DCNodes interacting with DCChain also do not need to pay transaction fees in most cases.)

The first method is a common blockchain operation mode. The introduction of the second method is due to the limitation of current blockchain networks where all operations must be performed on-chain, which is not feasible for developing super applications because even the lowest gas fees are still costs. Users will only pay for operations that generate final value, inevitably leading to many business logics not being realized. DC provides support for non-deterministic and non-value-generating business logic through services similar to centralized services provided by cloud service nodes. Additionally, blockchain's transaction ordering and block confirmation logic do not change with any business needs, leading to uncertainties in how long it will take for a DApp to respond to a business request. Therefore, if a DApp's business logic interacts too much directly with the blockchain, it can severely affect the user experience. DC, apart from necessary on-chain operations like DCT token transfers and cloud service space subscriptions, delegates other blockchain interactions to DCNode to ensure continuity in the user experience and correctness of data needing to be recorded on the blockchain.

13. Application Development Practices Based on DC Cloud Services

Developers can develop DApps based on on-chain smart contracts with DC, similar to traditional blockchain projects. The real innovation of DC lies in allowing developers to develop DApps based on DCNode that cover performance, user experience, and business scenarios beyond traditional internet applications. Developers can choose from the following methods based on their business needs:

- Development Based on DC Client SDK
  To reduce the difficulty for developers, DC provides a complete toolchain including SDKs supporting multiple languages for mobile and PC platforms, and a Docker-based development environment. In this way, developers can quickly develop DApps using their existing development experience and technology stack, without needing any blockchain-specific knowledge.

- Development Based on DC Entry Application
  DC provides a standalone entry application, similar to a Web3.0 wallet application, but besides basic wallet functionalities, its primary capability is to provide a deployment and runtime environment for lightweight applications. Any lightweight application based on H5 or Python can have decentralized data storage and light user interaction capabilities through the entry application. Anyone can publish applications with one click through this entry application, providing more innovative space for users and enabling everyone to participate in application development and customization in the upcoming AI era, better meeting personalized user needs.

- Development Based on DC's RPC Services
  DC provides RPC interfaces similar to centralized cloud services. Developers can access DCNode through these RPC interfaces to provide services. This method is more flexible but requires developers to have a thorough understanding of decentralized technologies and the entire DC operational mechanism.

14. Technical Implementation

DC's technical implementation mainly includes the following aspects:

14.1. DC's TEE Remote Attestation

DC's Remote Attestation is based on SGX DCAP (Data Center Attestation Primitive), which mainly addresses the reliability issues of software execution and is a crucial feature of TEE to counteract malicious actions. In DC, remote attestation is also central to the decentralized network formation. During the remote attestation process, nodes embed the public key of the currently running TEE, linking the node's identity, execution logic, and platform parameters with the TEE public key on the blockchain. Remote attestation is included in transactions related to network security submitted to DCChain by nodes, and it is verified by a randomly selected group of nodes already on the network, requiring the verifier to prove:

- The node's identity
- The integrity of the node's running logic
- The node is running on a genuine platform with Intel SGX enabled

The basic process of remote attestation in DC is as follows:

1. Each node configures and runs the PCCS (Provisioning Certificate Caching Service), which synchronizes hardware certificates from Intel PCS (Provisioning Certification Service) as needed to validate other nodes' remote attestations;
2. Nodes submit transactions to DCChain that carry remote attestations (such as node onboarding applications, reporting abnormal nodes);
3. DC randomly selects a group of networked nodes to verify the remote attestations;
4. If a node fails the remote attestation verification, it enters the abnormal node reporting process (the reporting transaction itself carries remote attestation for DC to verify);
5. If the number of reported node anomalies exceeds a certain ratio, penalties are imposed on the reported node, including deduction of staked DCT tokens, temporary network bans, etc.

The remote attestation mechanism ensures that the code logic cannot be altered, solving several key issues for DC:

- Ensures the identity of nodes, guaranteeing their trustworthiness;
- Fundamentally prevents witch attacks, generation attacks, resource attacks, and data tampering.

14.2. Proof of Empty Disk

To measure the storage supply of nodes, we have defined an empty disk encapsulation mechanism that allows nodes to effectively track the storage space declared by nodes within the TEE. The encapsulation file of a node consists of many 1GB-sized encapsulation files, each generated as follows: the node encrypts the file name with an encryption key in the TEE environment, undergoes a series of HASH processes to generate a seed of length 256 for the corresponding file name of the encapsulation file, and then generates the complete 1GB encapsulation file in a chained manner using this seed. This mode ensures that the node's storage space is genuine rather than fake, as the integrity of the file can be verified at any position by random sampling.

14.3. Upgrade of DCNode Based on TEE

The Substrate framework provides a powerful non-forking upgrade mechanism, but DC's DCNode mainly operates in a TEE environment, making this mechanism not directly applicable to this part. The DC team, based on the characteristics of Substrate chains and TEE, uses Substrate's non-forking upgrade mechanism for the upgrade of DCChain, while the upgrade of DCNode uses the TEE upgrade mechanism. By introducing the DCUpgrade application running in the TEE to assist in the upgrade, the main purpose of DCUpgrade is to transfer the keys from the old version of DCNode in the TEE environment to the new version of DCNode, ensuring that the new version of DCNode can decrypt the data from the old version. The upgrade process is as follows:

1. The DCUpgrade node program running in the TEE environment will regularly monitor the version number of the latest DCNode program on DCChain. If a new version confirmed by the DC network signature is found, it will automatically download the new version of the DCNode program.
2. DCUpgrade requests application keys from the running DCNode program.
3. The running old version of DCNode program initiates a TEE local proof challenge request to DCUpgrade with a random number.
4. The DCUpgrade node generates a TEE local proof based on the random number in the challenge request and the DC signed EnclaveID, then sends the proof to DCNode.
5. DCNode verifies the proof of DCUpgrade node and, after confirming the signature of DC network on the EnclaveID of DCUpgrade, sends the application key to DCUpgrade.
6. DCUpgrade encapsulates the application key in the TEE environment and shuts down the running DCNode program.
7. DCUpgrade starts the new version of the DCNode program.
8. The new version of DCNode program requests application keys from DCUpgrade.
9. DCUpgrade initiates a TEE local proof challenge request to the new version of DCNode program with a random number.
10. DCNode verifies the proof of DCUpgrade and sends the application key to DCUpgrade.
11. DCUpgrade verifies the local proof of DCNode and, after confirming that DCNode is signed by DC, sends the application key to the running new version of DCNode.
12. The new version of DCNode takes over the work of the old version of DCNode and completes the upgrade.

15. Attacks and Threats

15.1. SGX Side-Channel Attacks

Intel SGX technology is based on hardware-based trusted execution environments. Even if attackers gain permissions such as OS, hypervisor, BIOS, and SMM, they cannot directly attack the Enclave. Therefore, attackers often resort to side-channel attacks, such as attacks on page tables, cache, DRAM, etc. Side-channel attacks primarily aim to obtain data from attack surfaces, deduce control flow and data flow information, and ultimately extract code and data information from the Enclave, such as encryption keys, private data, and so on.

In the DC network framework, the core sensitive data in the node's TEE, namely the TEE's private key, is threatened by side-channel attacks. One feasible method to defend against side-channel attacks is to introduce enhanced cryptographic algorithms at the source code level, such as using enhanced elliptic curve and AES algorithms. Enhancements at the source code level help hide data and control flows effectively, thus safeguarding sensitive data within the node's TEE.

15.2. SGX-ROP Attack

Return Oriented Programming (ROP) is a novel attack based on code reuse techniques, where attackers extract instruction fragments from existing libraries or executable files to construct malicious code. By scanning existing dynamic link libraries and executable files, attackers extract usable instruction fragments (gadgets) that end with a ret instruction, connecting these instruction fragments through ret instructions to execute the code flow. To conduct SGX-ROP attacks, malicious programs need to be loaded into a Trusted Execution Environment (TEE) for execution, causing damage to the host, with protective software unable to scan useful information from the SGX Enclave.

DC is an open-source framework where any program code and sources released within the community can be reviewed, fundamentally eliminating the possibility of malicious code damaging the host node. Meanwhile, ROP attacks mainly target local systems, and if malicious code is embedded within the TEE node, it cannot generate remote proofs recognized by the DC network, failing network verification and ensuring no impact on the entire network.

15.3. PlunderVolt and VoltPillager attacks

PlunderVolt and VoltPillager attacks target SGX's encryption keys from both software and hardware perspectives, manipulating processor voltage and frequency to inject controllable hardware faults into Intel's Advanced Encryption Standard instructions, resulting in erroneous outputs that allow attackers to recover encryption keys outside the enclave. DC's mechanisms for workload reporting, file packaging, and Metadata storage rely on encryption algorithms like ECC and AES. To prevent nodes from being attacked by PlunderVolt or VoltPillager to obtain private keys and falsify workload reports, DC runs the entire DCNode as a separate Enclave, preventing attackers from controlling input or obtaining any output, fundamentally avoiding PlunderVolt and VoltPillager attacks.

15.4. In the worst-case scenario

DC can withstand known SGX security vulnerabilities but also has strategies to mitigate potential future threats. Assuming the worst-case scenario occurs (although there are currently no signs of it happening), where a malicious node breaches SGX and obtains the node's private key, it means the node can freely forge TEE software and hardware attestations.

- Workload falsification: Falsifying file integrity verification information could result in DC data not being stored truthfully.
- False node entry: False verification could lead to nodes containing malicious code joining the network.

Response Plan

1. Establish a reasonable single-point storage limit
   By limiting the storage capacity of a single node, false computational power can be restricted, controlling the impact of effective malicious attacks on the network.
2. RICS-V based TEE solution
   Current mainstream TEE solutions face challenges due to their closed-source nature. Being closed-source implies potential vulnerabilities and backdoors, while the RICS-V open instruction set architecture can address this issue at its core. As RICS-V based TEE solutions mature, DC will also support RICS-V based TEE solutions in the future.

16. Technological Evolution

DC focuses on formulating and continuously improving protocols, maintaining an open attitude towards new technologies and new participants. In addition to the technical implementations mentioned earlier, there are several initiatives that can contribute to DC's growth, including but not limited to:

- Supporting various TEE solutions: Initially based on Intel SGX technology, DC will integrate various solutions through a TEE abstraction layer in the future, such as ARM chip's rustZone, AMD's SEV, Software TEE based on TPM module, and future RICS-V based TEE solutions.
- Supporting quantification of computing: With TEE-based support, tasks like decentralized execution similar to FaaS can be quantified, for example, through TEE-based code obfuscation algorithms.
- Supporting robust on-chain governance: To better cater to technological advancements and ecosystem progression, DC will introduce fair and efficient decentralized on-chain governance.
- Integrating into the Web3 ecosystem: DC can address all decentralized cloud service scenarios in the Web3 ecosystem, while also benefiting from the ecosystem acceleration brought by Web3.