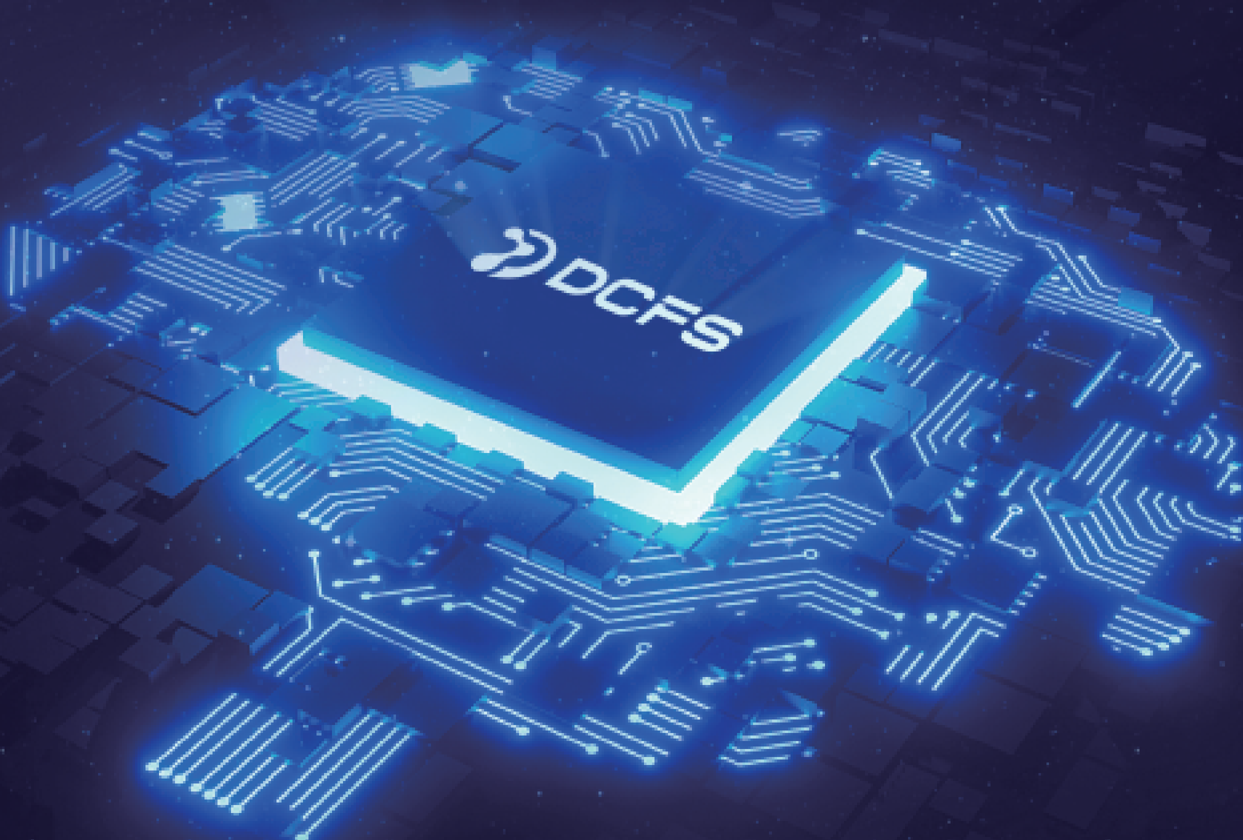
DCFS white paper



PREFACE

Be it distributed network, memory, or storage, the ultimate goal is to achieve distributed computing where data flows across the computer nodes, while each node has access to shared data in some way, and the final output from distributed computing is stored and exported persistently. Distributed computing has become an important engine for driving business growth, meanwhile as an important underlying technology for the new infrastructure, it is also making its debut in emerging fields such as artificial intelligence, big data, edge computing, and 5G.

In recent years, three basic consensus has been formed within the blockchain industry through continuous exploration and trial and error, which are:

- First, the inherent attributes of distribution and alliance of blockchain, relying on which, credible transactions around the common transaction goals of the participants are achieved.

- Secondly, blockchain is changing the current transaction model. The original transaction model carries out transactions with physical collateral or third-party guarantees，which is beneficial to business development in the IT era, however, in the DT era, the development of social economy and new generation of information technology calls for a transaction model with more efficiency and higher agility, and the error of the original model is increasing with constantly presenting drawbacks. The credible transaction model constructed by the data-based blockchain has shortened the distance between demand and information, and supports the new demand of digital economy development.

- Thirdly, blockchain,being the infrastructure of digital economy, focuses on the service industry and empowers thousands of industries. The fundament of industry development is data, and the original "data pool" model is unable to "irrigate" the various branches and sections of the industry. The blockchain is constructing a "data river" model for the safe flow of data elements and is serving the industry through credible flow, to achieve the goal of serving the development of digital economy in the industry. With the advent of DT era, it has become the consensus of all to eliminate information silos. Blockchain, relying on its various features, including immutability, traceability and programmability, forms a multi-mediated network mode by means of equal rights and common governance of node alliance, and takes in other technologies, such as IOT, 5G, data network, cloud computing, big data, AI and future quantum encryption, to form an integrated and open system architecture, revolving the end-to-end trusted flow of data, and tackle the issue of data sharing under the traditional centralized model, and accelerate the process of industrial development of digital economy. The new era asks for the matching and evolving of the blockchain technology and its concept, and we are well aware that there is still huge development potentials for blockchain. The only way for us to lay a solid foundation for the construction and high-speed development of the digital economy is to keep innovating and staying ahead of the curve by building a convergent and open blockchain infrastructure.

CONTENTS

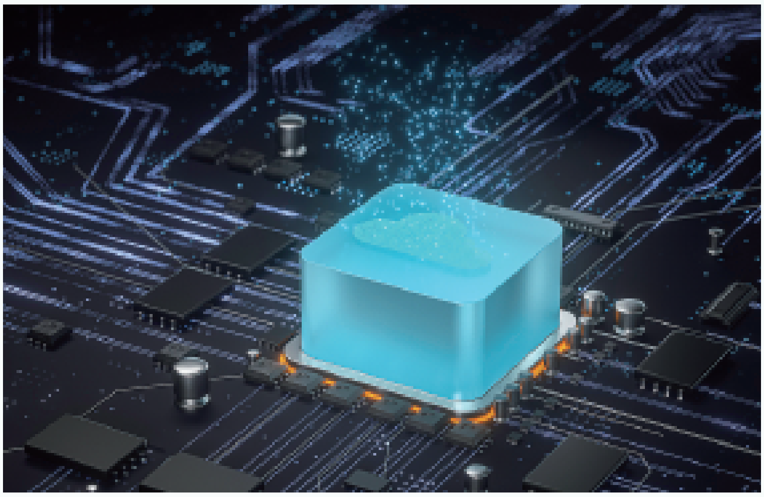
PROJECT BACKGROUND

In the implementation of cloud computing, big data and other new technologies, computing power has become an important supporting force for the in depth development of the digital economy. Whether it is the new infrastructure, industrial Internet, or various intelligent cities underway, servers, high-performance computing clusters, artificial intelligence hardware and other basic and intelligent computing can always be found underneath.

Cloud computing already has a mature business model as well as sound technical support. But the technical threshold for the providers of these technologies is high. And cloud computing technologies are highly centralized so that computing power can be distributed from the top to the bottom. This leads to high costs for some smaller-scale computing resources to operate individually but cannot be provided to large cloud service providers. DCFS classifies computer resources in respect of computing power and loosely accesses the platform, while using blockchain technology to guarantee data reliability, thus solving the problem of scheduling small and medium-sized computing resources across the network.

1.1 Computing power infrastructure

With the acceleration of digitalization, the digital economy will become a major driver of future global economic growth through new business models and formats. The share of digital economy as a component of GDP is increasing yearly. How can the digital economy be used to empower the overall economic development is a key development strategy for most countries, among which, computing power is an important driver for the development of the digital economy and a fundamental element for digital technology to deliver its benefits. The global digital transformation has entered the stage of multiplying innovation, and the proportion of digital economy in each country will continue to increase. Computing power is an important factor for the sustainable development of digital technology, and is the core productivity in the digital economy era. DCFS, with "the computing power as the core, and application as the driver", will develop the distributed computing file system (hereinafter referred to as "DCFS"), a public chain for computing power infrastructure in the digital economy.



1.2 Our vision

With the core idea of "computing power as the core and application as the driver", we focus on promoting the application of emerging technologies and technological innovation, and the synergistic development of its application level and computing power, forming a closed loop in which computing power, application level and economic growth are mutually driven.

We aim to provide a cost-effective, low-latency, high-density, wide-coverage, flexible, secure and infinitely scalable distributed high-performance computing network for all industries.

1.3 The future market for computing power

1.3.1 Computing power is the core productivity in the digital economy era.

The global digital transformation has entered the stage of multiplying innovation, the proportion of digital economy of each country will continue to increase, and computing power is an important factor in the continued development of digital technology and the core productivity of the digital economy era.

1.3.2 The more synergistic the application level of emerging technologies and computing power, the more obvious the boost of computing power on the economy.

The computing capacity and efficiency are the basic guarantee to support the implementation of applications of emerging technologies, while the computing power infrastructure provides sustainable development for the computing capacity, efficiency and application level, the mutual boosting of which makes the greater economic benefits possible.

1.3.3 The Internet industry enables the application of emerging technologies and technological innovation by investing in computing power.

The Internet has the most extensive application of emerging technologies such as AI, IoT and big data, and has the highest share of spending in AI. The Internet achieves large-scale AI technology development with the help of computing power facilities and applies AI to multiple product services and operational intelligence scenarios.

1.3.4 Emerging technology applications are accelerators pushing the development of computing power.

Global enterprise AI spending will reach $49.87 billion in 2020 and is expected to reach $96.28 billion in 2023. Artificial intelligence is driven by computing power, data and algorithms, among which sufficient computing power can better handle the increasingly large data and complex algorithms; the extensive application of IoT drives the rapid development of edge computing, which reshapes IT infrastructure and will change the form of computing, storage and network. Cloud computing is the most important way to improve computing efficiency, by deploying IT infrastructure to support the collaborative development of the cloud and edge computing, it will facilitate the realization of emerging applications, improve the overall computing power index, and gain greater economic returns. Emerging technologies such as artificial intelligence, IoT, and cloud computing will be the core drivers of future IT spending.

1.3.5 Open computing is the development direction of future computing power infrastructure.

Open computing is a complete industry ecology of data center, representing the leading technology innovation in data centers. Internet and IT enterprises are actively participating in open computing, promoting data centers to develop in the direction of openness, integration and intelligence, and driving technology sharing and standard interoperability in the data center field. Meanwhile, open computing is gradually penetrating traditional industries, with the deployment of open standard IT infrastructure in telecom, finance, gaming, and e-commerce.

1.3.6 The new normal has given rise to new business models and computing power has become fundamental to economic and social development.

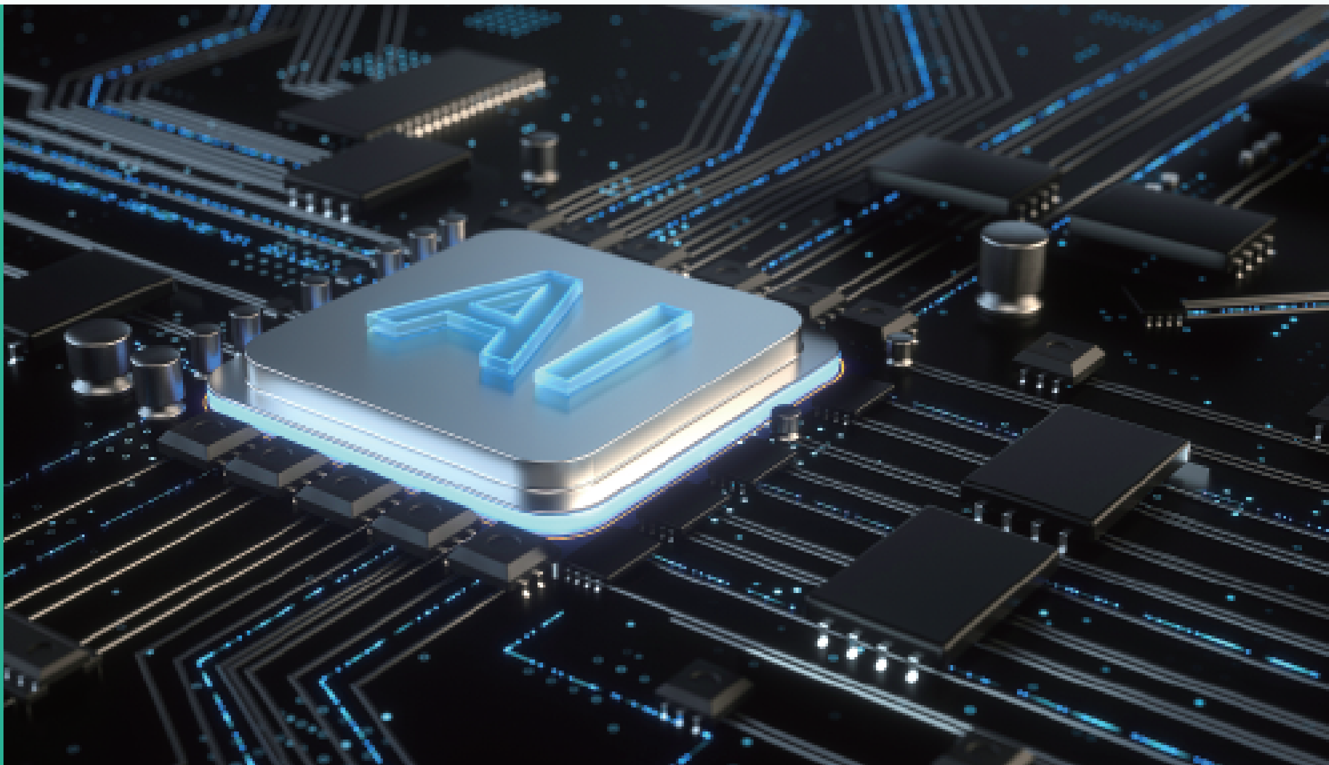
Since 2020, the digital economy has become more and more significant in driving the social economy, giving rise to many new application scenarios such as online education, Internet healthcare, teleconferencing, unmanned retail, etc. The features of online economic life and intelligent social services have become more obvious, and the changes in users' consumption habits and usage patterns have prompted enterprises to accelerate their transformation. Under the new normal, digital economy will become the key to growth and innovation for enterprises, and computing power will provide the key support.

PROJECT OVERVIEW

As new services such as artificial intelligence, 5G, HPC, and edge computing are gradually being implemented and popularized, higher requirements for computing power diversity are being put forward.

Based on blockchain technology, we have developed a highly efficient, secure, low-cost, private, and scalable distributed high-performance computing power system network to meet the specific computing power needs of various application scenarios through the support of multiple computing power, and to provide a perfect peripheral computing power infrastructure for applications based on the soft and hard ware collaboration architecture.

High performance computing nodes in DCFS can provide high performance computing services for multiple industries, including data preparation, data processing, model training and model-based reasoning for artificial intelligence, face recognition, graphics rendering, blockchain zero-knowledge computing, etc. High performance computing nodes in DCFS can be of various forms, including full-function nodes running on large GPU, NPU, and FPGA server clusters (permanent nodes), idle spare GPU/CPU server computing nodes of small and medium-sized enterprises, and idle GPU/CPU computing nodes of individuals.



2.1 Issues to be tackled by DCFS

Low cost	The core issue to solve of the original design of DCFS is the high cost investment for enterprises or individuals with computing power needs. Through the unique model of DCFS ecosystem, computing power resources can be obtained at low cost, and through the underlying computing power infrastructure, enterprises or individuals can obtain computing power resources at a lower price.
Low latency	To meet the demand for low latency, DCFS should be solutions should be built at the "edge" nearest to the business site to reduce the processing latency; it can directly pair the computing power resources through DCFS with the computing power nodes that have already disclosed their own regional locations.
Massive data	According to Gartner, in 2020, there will be up to 25 billion smart devices connected to the Internet and generating 50 trillion GB of data worldwide, which is very difficult to transmit and process directly and the cost is extremely high, however the unique load balancing technology enables each node VM to cooperate with each other to share the concurrency pressure, and distribute analysis and filtering to save the network bandwidth.
Privacy and security	DCFS should be able to protect the privacy of every participant in the ecosystem and let the participants freely decide the level of openness of the information. DCFS guarantees this through encryption algorithms and separation mechanisms.
Synergy optimization	To solve the problems of scattered data distribution and difficult collaboration, and reduce the difficulty of use by enterprises or individuals, we realize cross-regional collaborative computing through DCFS, achieving interconnection and collaborative analysis by avoiding unnecessary ETL processes and reducing data relocation.

2.1 Issues to be tackled by DCFS

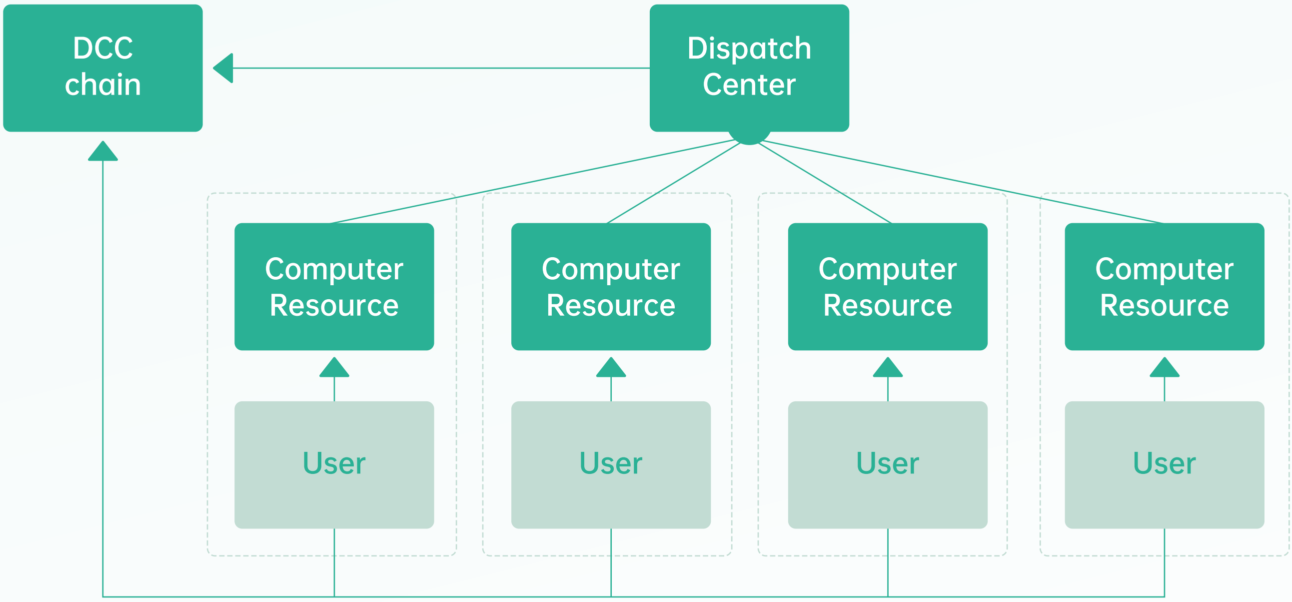
Operation and maintenance autonomy	In the event of network disconnection, service of the edge side should not be affected, which means the edge side should be able to process offline and self-recover. DCFS should provide business self-healing capability, where the node VM failures can be alerted and concurrency at multi-level is possible, meanwhile, a small number of nodes are upgraded or rebooted at a time without affecting the overall cluster business, in which the upgrade batches are automatically scheduled based on the dependencies of components and instances. During the upgrade process, the faulty nodes are isolated and then troubleshoot after the upgrade is completed. The process is carried out repeatedly until all nodes in the cluster are upgraded to the new version to ensure business continuity.
Big data storage and computing coupling optimization	The traditional three-copy storage for big data has low cost performance, with a common 10PB of storage space, the effective capacity is only 3PB; meanwhile, there is a disequilibrium of storage and computing resources with the storage utilization usually over 70% while CPU utilization is less than 50%, and when scaling, computing and storage resources are scaled as well, which is a waste of resources. DCFS adopts a scheme of separation of big data storage and computing, which enables independent on-demand scaling of computing and storage while maintain the performance to easily cope with business surges and improve overall resource utilization.

TECHNICAL INTRODUCTION

3.1 Design concept

DCFS primarily consists of the DCC chain, a computing power scheduling center and decentralized computing power providers. The computing power scheduling center audits and monitors the connected computing power in real time and submits the computing power proof to the blockchain, while DCFS distributes the revenue based on the computing power proof. In addition, DCFS is also compatible with EVM and has the ability to support various popular DAPPs.

3.2 Brief introduction of technical architecture



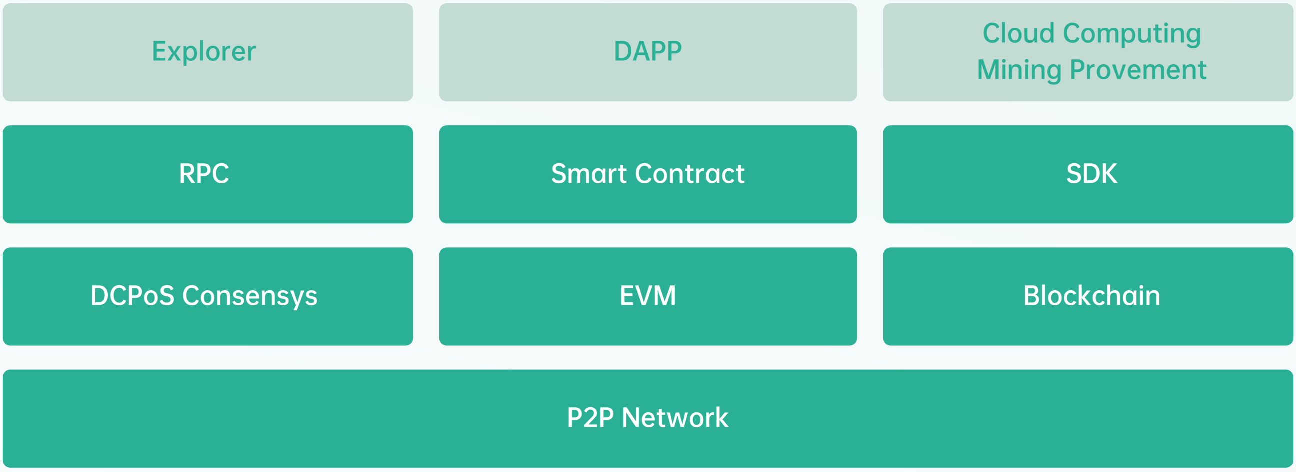
As shown in the figure, DCFS runs independently on the Internet as the core of the economic and business models. The scheduling center classifies and evaluates various computing resources offline, and finally uploads the proofs to DCFS in the form of a standard computing power. The scheduling center will then monitor the on-chain computing power in real time and upload the proof of computing power validity to the chain as well, which is rewarded by DCFS periodically. See below for details of the reward rules.

Various computing power resources of different sizes can access the scheduling center in a distributed manner and register on DCFS. In order to guarantee the quality of the computing power resources connected to the platform, the platform has certain reward and punishment mechanisms to ensure that users can use these computing power resources stably.

3.2.1 Computing power proof module

DCFS

DCFS is a high-performance blockchain with supporting EVMs. DPoS ensures that a block can be generated consistently every 6 seconds. Through smart contracts, DCFS is able to accept proofs of computing power register and proofs of computing power monitoring submitted by the scheduling center. Through these proofs, the contract automatically assigns computing power block rewards to registered users. In practice, users and the scheduling center interact with DCFS through JSONRPC. Meanwhile, the DCC chain supports EVM and ETH interfaces, and exchanges and third-party applications can easily adapt and access the chain.

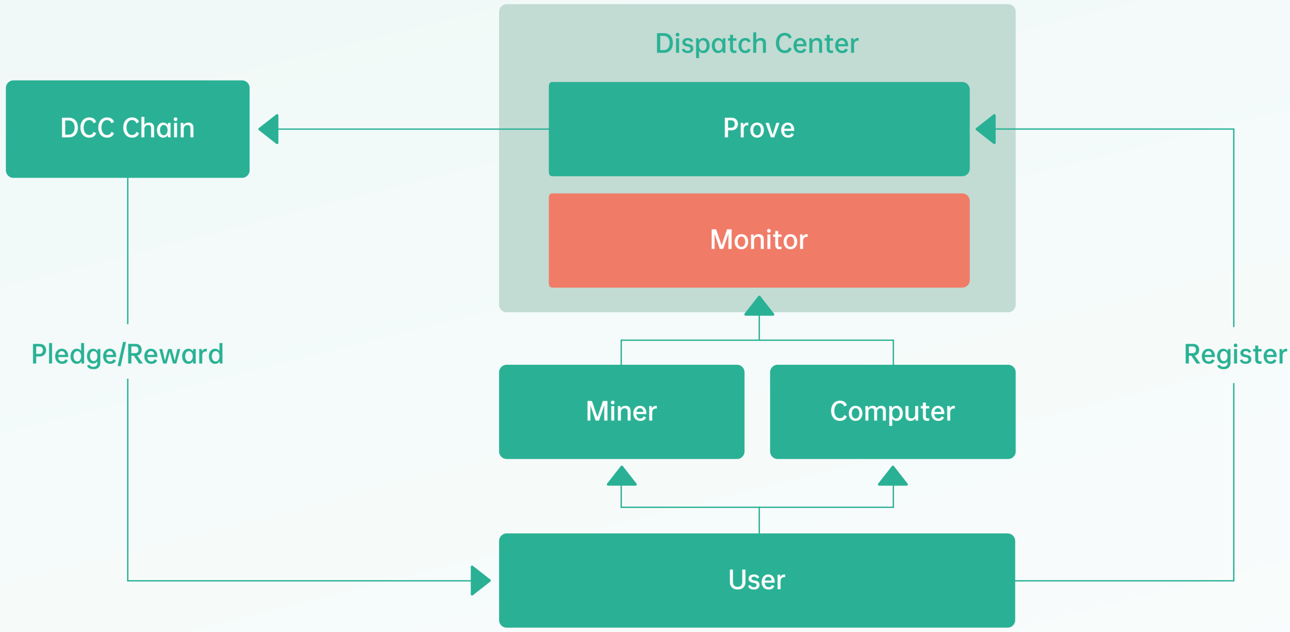


DCFS is designed to be a multi-layer, multi-module architecture.

- P2P network - Primarily for blockchain data transfer and synchronization.
- DPoS - Guarantees stable block generation in a defined time frame.
- EVM - Ethereum smart contracts compatible.
- Blockchain - Stores all block data.
- SDK - Provides development packages in different languages.
- Smart Contract - On-chain smart contracts written in Solidity.
- RPC - JSON RPC interface. Various blockchain applications can be developed based on DCC chain.
- Explorer - Block Explorer.
- DAPP - Decentralized applications supporting DeFi, GameFi and others.
- Cloud Computing Mining Proof - Smart contracts that support on-chain proof of computing power and execute computing power block reward logic.

Scheduling center

There are many types of computing devices, and the evaluation of computing power requires professional technical means. The scheduling center, being an important role in the chain, takes the responsibility of standardizing computing power. In the initial stage, the scheduling center will use technical means to evaluate and access some typical devices and monitor the computing power in real time while recording the proof of the computing power work to DCFS in a timely manner.



3.3 Consensus

DCC uses DPoS consensus with low transaction cost, low transaction latency, high transaction concurrency, and supports up to 21 validators. DPoS is a combination of PoA and Pos. The administrator submits a proposal for a new validator and waits for the other active validators to pass, with a majority passing to qualify. Any address can be pledged to an address that qualifies as a validator, and when that validator ranks in the top 21 in pledges, it becomes an active validator for the next epoch. All active validators are sorted according to predefined rules and take turns to pack blocks. If a validator fails to pack a block in time during its turn, packing is randomly assigned among active validators who have not participated in the past n/2 (n is the number of active validators) blocks. There are at least n/2+1 active validators working properly to ensure the functioning of the blockchain. The difficulty to generate a block is 2 when it is generated normally and 1 when it is not generated in the predefined order. When there is forking, the blockchain selects the corresponding fork based on the cumulative maximum difficulty.

- **Validator**

  Responsible for packing on-chain transactions into blocks.

- **Block validator**

  The set of validators currently responsible for packing, a max. of 21.

- **epoch**

  The time interval between block generation, currently on DCC 1 epoch = 200 blocks.
  At the end of each epoch, the blockchain interacts with the system contract to update the block validator.

3.4 System contract

The management of current validators is done by the system contract.

Proposal	Proposal is responsible for managing access to the validator and managing the validator's proposals and votes.
Validators	Validators are responsible for the ranking management of validators, staking and unstaking operations, and block reward distribution, etc.
Punish	Punish is responsible for penalizing the operation of active validators that are not functioning properly.

Blockchain invoking the system contract: At the end of each block, the Validators is invoked and the fees for all transactions in the block are assigned to the active validators. When a validator is not working properly, the Punish is invoked to punish the validator. At the end of each epoch, the Validators is invoked to update the active validators based on their ranking.

3.5 Staking

For any account, any number of tokens can be pledged to the Validators, with a minimum of 16 DCCs. If you want to cancel the pledge, carry out following operation : Send the Validators the non-pledged transaction of the validator; wait for 86,400 blocks, then send the transaction to the Validators to withdraw all the pledged coins on that validator.

3.6 Penalty

Whenever a validator is found to have failed to pack a block as predefined, the Punish contract is automatically invoked at the end of that block and the number of validators is counted, which, if reaches 24, all of the validator's earnings are penalized; if reaches 48, the validator is removed from the list of active validators and disqualified.

3.7 Technical implementation

3.7.1 Hybrid and multi-cloud solutions

The DCFS cloud platform will rely on an effective hybrid and multi-cloud approach to seamlessly manage, govern and help secure servers, Kubernetes clusters and applications in local, multi-cloud and edge environments from any one location. Always bringing the latest services to any infrastructure and benefit from elastic scaling, automation, unified management and industry-leading security. Users can build and deploy cloud-native applications in their own data centers, consolidating local virtualized applications onto a hyper-converged infrastructure with the best price/performance and already connected to the cloud. Bring compute, storage and artificial intelligence to Internet of Things (IoT) devices. Run machine learning and advanced analytics at the edge to get real-time results.

3.7.2 High performance computing

High performance computing (HPC) is a complete set of computing, network and storage resources integrated with workload business process services for HPC applications. The DCFS cloud computing platform will have a purpose-built HPC infrastructure, solutions and optimized application services, realizing in a more competitive price/performance compared to local options and additional HPC benefits.

The DCFS cloud computing platform will maximize the use of all CPUs, GPUs, FPGAs and fast interconnect capabilities with cost-controlled optimized performance, reducing the assignment completion time from days to minutes. Users can build and manage their own dedicated HPC clusters, enable end-to-end application lifecycle management in the cloud, and build and train new AI models more quickly with automated machine learning, auto-scaling cloud computing and built-in DevOps.

3.7.3 Confidential computing

Move existing workloads to the DCFS cloud computing platform and keep them private without changing any code. With third-generation AMD EYPC technology, the entire contents of the VM are opaque to cloud administrators, enabling secure, independent computing. The trusted boot feature measures the integrity of confidential VMs. The runtime state of these VMs is fully encrypted, protecting the data even when it is in use. The keys used for this RAM encryption are generated within the CPU and never leave the CPU.

Optimize application-level confidentiality with Intel SGX. Migrate existing applications directly to secure Enclave. Use confidential nodes in containers supported by the DCFS cloud computing platform to manage keys confidentially. Use open neural network exchange (ONNX) for runtime support to achieve confidentiality.

Provides protection for sensitive and prescriptive data processed in the cloud by isolating computing in a hardware-based trusted execution environment (TEE). Protects data in use from being used by cloud providers, administrators, or users. Generate solutions based on secure hardware using familiar tools, software, and cloud infrastructure.

3.7.4 Backup and disaster recovery

The DCFS cloud platform will provide a simple, secure, scalable and cost effective end-to-end backup and disaster recovery solution that can be integrated with native data protection solutions. The DCFS cloud platform backup and disaster recovery solution is easy to build, cloud-native, highly available, and recoverable.

Help protect backup environments with built-in hybrid and cloud security and compliance with a wide range of security and privacy regulations. Use platform backup to protect data from deletion and ransomware threats by isolating original data from backup data as well as accidental deletion protection and multiple authentication.

Achieve low recovery point objectives (RPOs) and recovery time objectives (RTOs) for any mission-critical workload in the organization. Reduce the cost of deploying, monitoring, patching and scaling a local disaster recovery infrastructure without having to manage backup resources or build a secondary data center. The DCFS cloud platform provides a zero-infrastructure solution with flexible strategies for optimizing backup storage.

3.7.5 Computing power resource node access

The DCFS cloud computing platform will provide access interface to computing power resources, users can deploy DCFS resource nodes on local devices and access the DCFS computing power resource network, pledge a certain amount of DCFS tokens in advance, and get access to the platform while maintaining stable device and network conditions, the platform will automatically dispatch all successfully accessed computing power resource nodes and automatically assign relevant computing power tasks. The DCFS cloud computing platform will continuously monitor the nodes that have successfully accessed the computing resources, and will penalize the nodes when they are not accessible for a long time by deducting a certain amount of DCFS pledge tokens according to the time they are offline, and will cancel the nodes' access to the platform if they cannot reach the required pledge amount after they are online.

3.8 Performance and deployment requirements

3.8.1 Cloud computing platform solutions

Introduction to OpenStack

OpenStack is a free software and open source project developed and sponsored by NASA (National Aeronautics and Space Administration) in cooperation with Rackspace and licensed under the Apache license. As an open source cloud computing management platform, it aims to provide software for the construction and management of public and private clouds to help service providers and enterprises to implement cloud infrastructure services (Infrastructure as a Service, IaaS) similar to Amazon EC2 and S3 internally.

OpenStack projects and components

1.Dashboard
Service Name: Dashboard
Project Name: Horzon
Function: Provide a Web front-end console to showcase OpenStack features.

2.Compute
Service Name: Compute
Project Name: Nova
Function: A set of controllers for managing the entire lifecycle of virtual machine instances for single or group users, providing virtual services according to user requirements. Responsible for virtual machine creation, power on, power off, hang, suspend, adjust, migrate, restart, destroy and other operations, configure CPU, memory and other information specifications.

3.Network
Service Name: Network & Address Management (Network)
Project Name: Neutron
Function: Provides network virtualization technology for cloud computing, and provides network connectivity services for other OpenStack services. Provides interfaces for users to define Network, Subnet, Router, configure DHCP, DNS, load balancing, L3 services, network support for GRE, VLAN. The plug-in architecture supports many mainstream network manufacturers and technologies, such as OpenvSwitch.

4.Object Storage
Service Name: Object Storage
Project Name: Swift
Function: A system for object storage in a large-scale scalable system with built-in redundancy and high fault tolerance mechanism, allowing for the storage or retrieval of files. Provides image storage for Glance and volume backup service for Cinder.

5.Block Storage
Service Name: Block Storage
Project Name: Cinder
Function: Provides stable data block storage service for running instances whose plug-in driven architecture facilitates the creation and management of block devices, such as creating and deleting volumes, mounting and unmounting volumes on instances.

6.Identity Service
Service Name: Identity Service
Project Name: Keystone
Function: Provides authentication, service rules and service token functions for other OpenStack services, manages Domains, Projects, Users, Groups, and Roles.

7.Image Service
Service Name: Image Service
Project Name: Glance
Function: A set of virtual machine image search and retrieval system, supporting a variety of virtual machine image formats (AKI, AMI, ARI, ISO, QCOW2, Raw, VDI, VHD, and VMDK), with the ability to create and upload images, delete images, edit the basic information of the image.

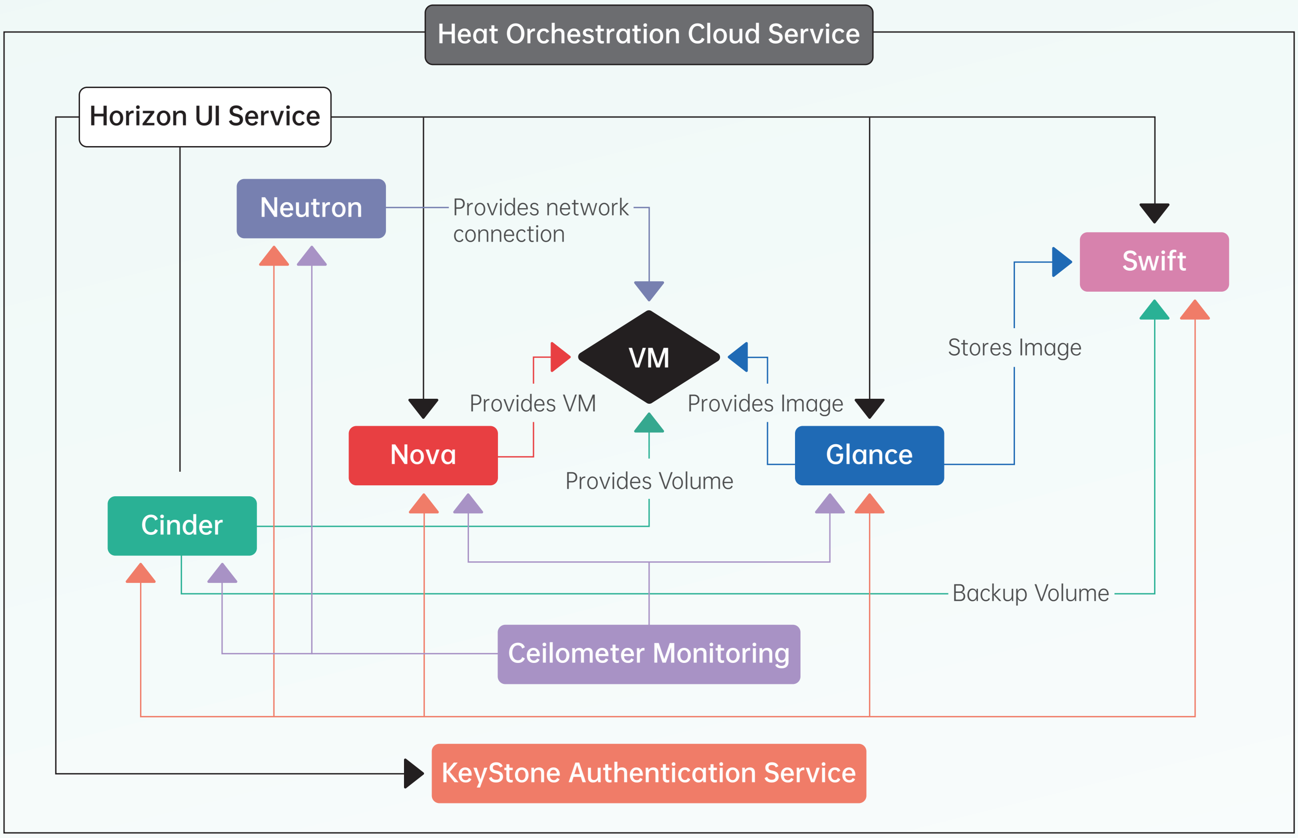
8.Metering Service
Service Name: Metering
Project Name: Ceilometer
Function: It can collect almost all the events that happen inside OpenStack, and then provide data support for metering and monitoring and other services.

9.Orchestration Service
Service name: Orchestration
Project Name: Heat
Function: Provides a collaborative deployment approach defined by templates to automate the deployment of cloud infrastructure software runtime environments (compute, storage, and network resources).

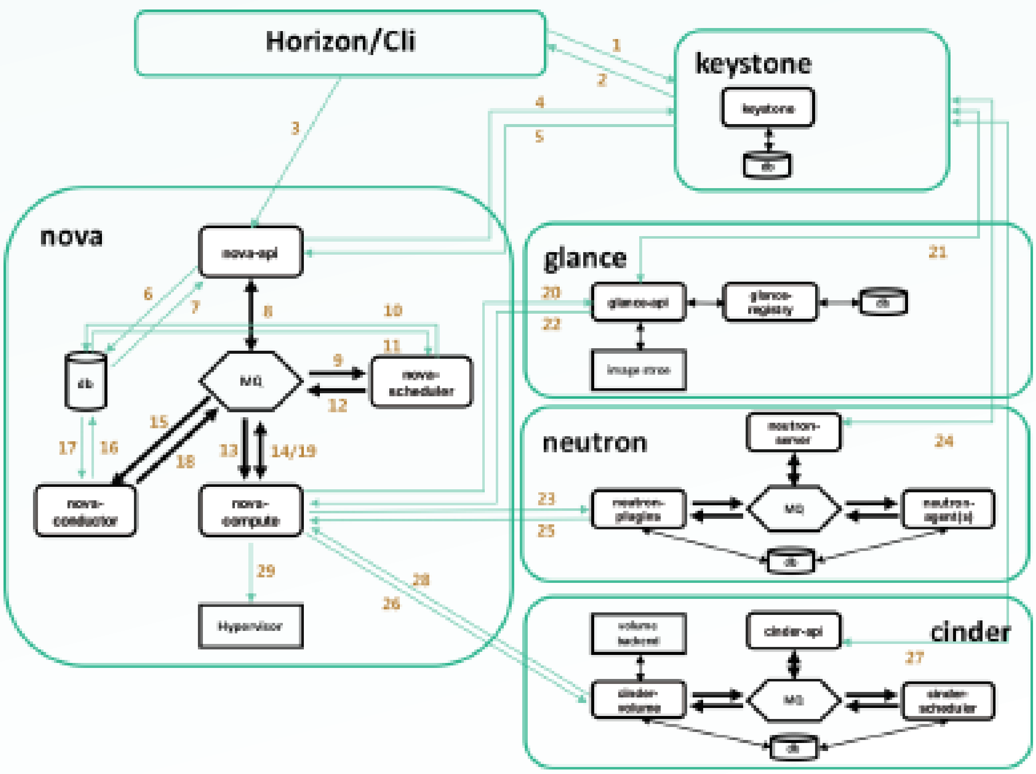
10.Database Service
Service Name: Database Service
Project Name： Trove
Function: Provides scalable and reliable relational and non-relational database engine services for users in OpenStack environments.

3.8.2 OpenStack components in detail

• Components logical relation diagram



• Openstack new cloud hosting flowchart



Virtual machine creation process

1.Interface or command line gets authentication information from keystone via RESTful API.

2.Keystone authenticates information through the user request and generates auth-token to return to the corresponding authentication request.

3.The interface or command line sends a boot instance request (carrying auth-token) to nova-api via RESTful API.

4.The nova-api accepts the request and sends an authentication request to keystone to see if the token is a valid user and token.

5.Keystone verifies the validity of the token, and returns a valid authentication and the corresponding role if it is valid (Note: Certain operations require role privileges to operate).

6.Communication between nova-api and database after authentication.

7.Initialization of the database records for the newly created virtual machine.

8.The nova-api requests from nova-scheduler via rpc.call for resources (Host ID) to create the virtual machine.

9.The nova-scheduler process listens to the message queue for requests from nova-api.

10.The nova-scheduler queries the nova database for computing resources and uses a scheduling algorithm to calculate the hosts that meet the needs for virtual machine creation.

11.For hosts meeting the requirements for VM creation, the nova-scheduler updates the physical host information corresponding to the VM in the database.

12.The nova-scheduler sends a request message corresponding to VM creation to nova-compute via rpc.cast.

13.The nova-compute will fetch the message for VM creation request from the corresponding message queue.

14.The nova-compute requests VM messages from nova-conductor via rpc.call. (Flavor)

15.The nova-conductor gets the nova-compute request message from the message queue.

16.The nova-conductor queries the corresponding information of the VM based on the message.

17.The nova-conductor gets the corresponding information of the VM from the database.

18.The nova-conductor sends the VM information by message to the message queue.

19.The nova-compute fetches the VM information message from the corresponding message queue.

20.The nova-compute gets the authenticated token via keystone's RESTfull API and gets the image needed to create the VM via HTTP request glance-api.

21.The glance-api authenticates to keystone the validity of the token and returns the authentication result.

22.The token passes authentication and nova-compute gets the VM image information (URL).

23.The nova-compute gets the token for authentication k through keystone's RESTfull API and requests neutron-server via HTTP to get the network information needed to create the VM.

24.The neutron-server authenticates to keystone the validity of the token and returns the authentication result.

25.The token passes authentication and the nova-compute gets the VM network information.

26.The nova-compute gets the authenticated token through keystone's RESTfull API and requests the cinder-api via HTTP to get the persistent storage information needed to create the VM.

27.The cinder-api authenticates to keystone the validity of the token, and returns the authentication result.

28.The token passes authentication and the nova-compute gets the VM persistent storage information.

29.The nova-compute invokes the configured virtualization driver to create a VM based on the information of the instance.

Keystone components

● Concepts within keystone

User:oThe object that uses the OpenStack Service is called a user, which can be a person, service, or system.

Role: The process of assigning privileges, in which by assigning a Role to a User, privilege is assigned to a User. It can be simply understood as the relation between positions and employees within a company, where different positions have different privileges. In OpenStack, the token returned to the User by keystone contains a list of Roles, and the service being accessed determines the Role of the User accessing it and the token provided by the User. Role by default is admin and _member_.

Project (Tenant) : It can be understood as a collection of resources owned by a person or service. A Project can contain multiple Users, each of whom can use the resources within the Project according to their assigned privileges. For example, a virtual machine created using Nova should be assigned to a certain Project, and a volume created by Cinder should be assigned to a certain Project. Before a User can access the resources in a Project, it needs to be associated with the Project and specify the Role of the User in the Project.

Policy: In addition to the identity authentication of a User, Openstack also needs to identify whether a User has access to a Service. For keystone, Policy is actually a JSON file, which is /etc/keystone/policy.json by default. With Policy, keystone implements privilege management of Users.

Token: A token is represented as a string that contains resources accessible within a specified scope and for a specified period of time. Tokens are generally held by the User.

Credentials: Credentials are used to confirm the User's identity.

Authentication: The process of determining the identity of a User.

Service: It refers to the component services running in Openstack.

EndPoint: It is an address, usually in the form of a URL that can be used to access and locate an OpenStack service over the network.

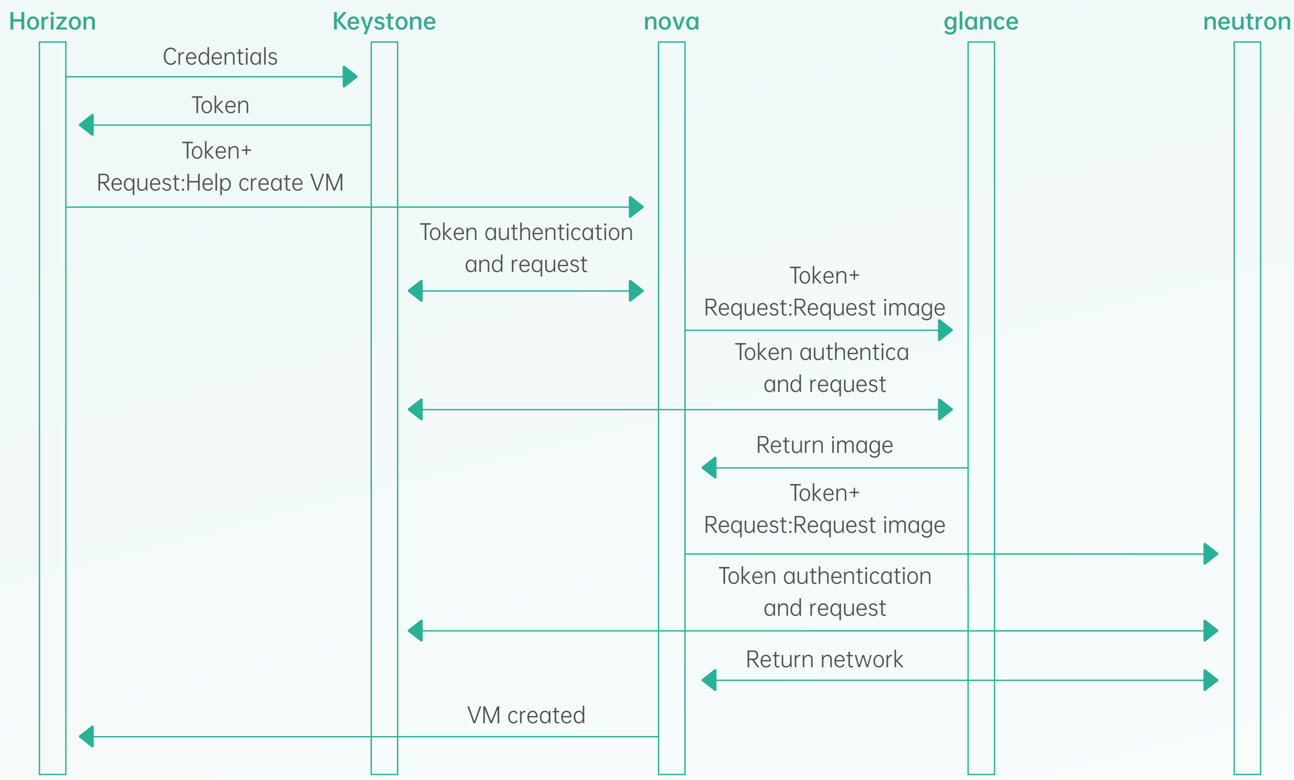
Admin Url: For admin users, Port: 35357

Internal Url: For OpenStack internal service use, Port: 5000

Public Url: For other users, Port: 5000

It should be noted that although there are three URLs, when using EndPoint access, the User's privileges are not related to the URL, other than the Role.

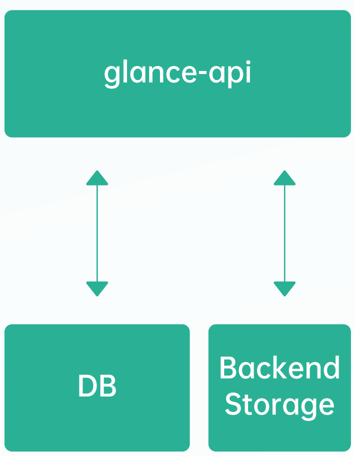
- **Interaction of keystone with various components**



rest api based communication between components
keystone Features:1.Authentication 2.rest api distribution

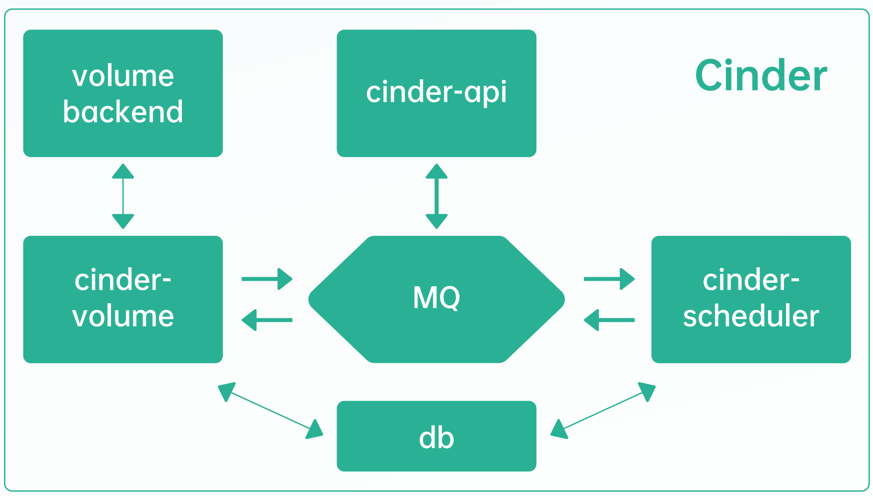
Glance components

Glance is responsible for providing image services.



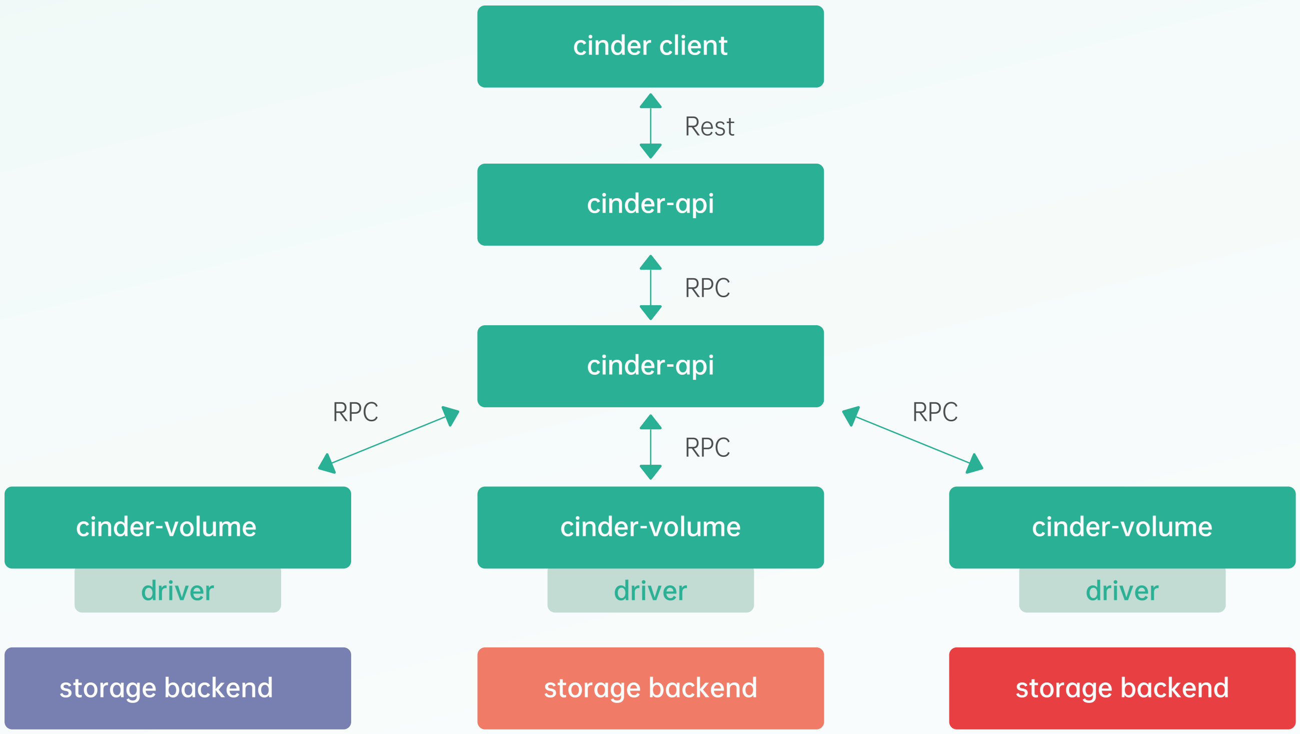
Cinder

Flowchart:

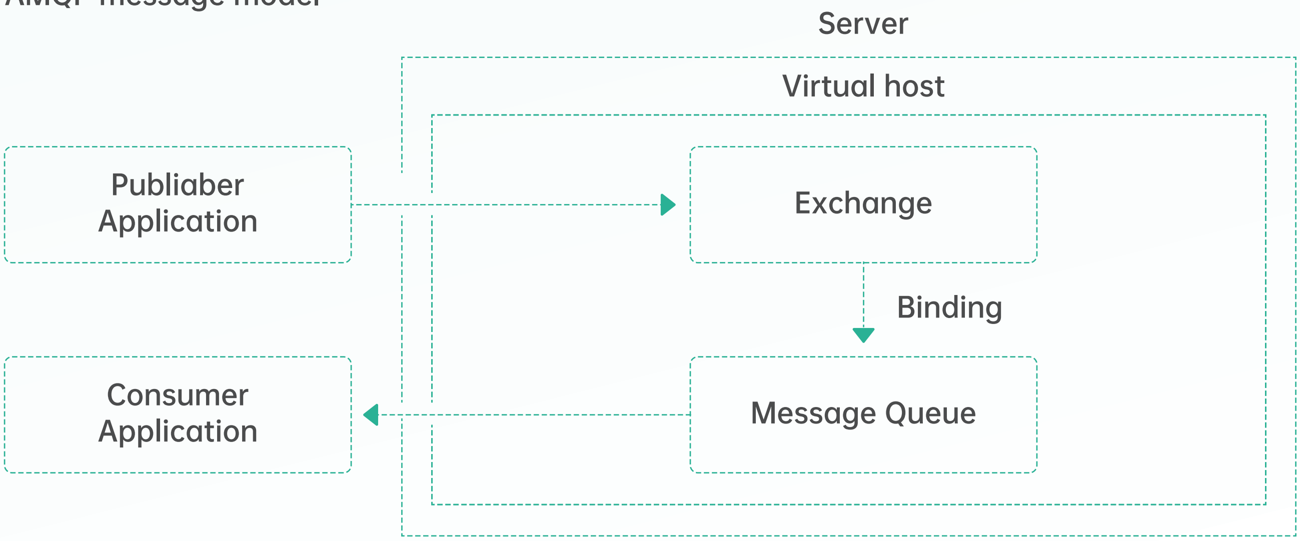


Main components of Cinder and features

The cinder-api is the endpoint of the Cinder service, providing the rest interface that handles client requests and sends RPC requests to the cinder-scheduler component.
The cinder-scheduler is responsible for scheduling Cinder requests, the core part of which is the scheduler_driver, which is the driver of the scheduler manager and is responsible for the specific scheduling of cinder-volume, sending Cinder RPC requests to the selected cinder-volume.
The cinder-volume is responsible for specific volume request processing, with volume storage space provided by various back-end storage. The major storage manufactures have been actively contributing drivers for their storage products to the Cinder community.



AMQP message model



RPC send request

The publisher broadcasts and declares the message address of the RPC request by the client, and the consumer receives and processes the message, and returns the result message of the request if a message answer is required.

The OpenStack RPC module provides three RPC call methods, rpc.call, rpc.cast, and rpc.fanout_cast, to send and receive RPC requests.

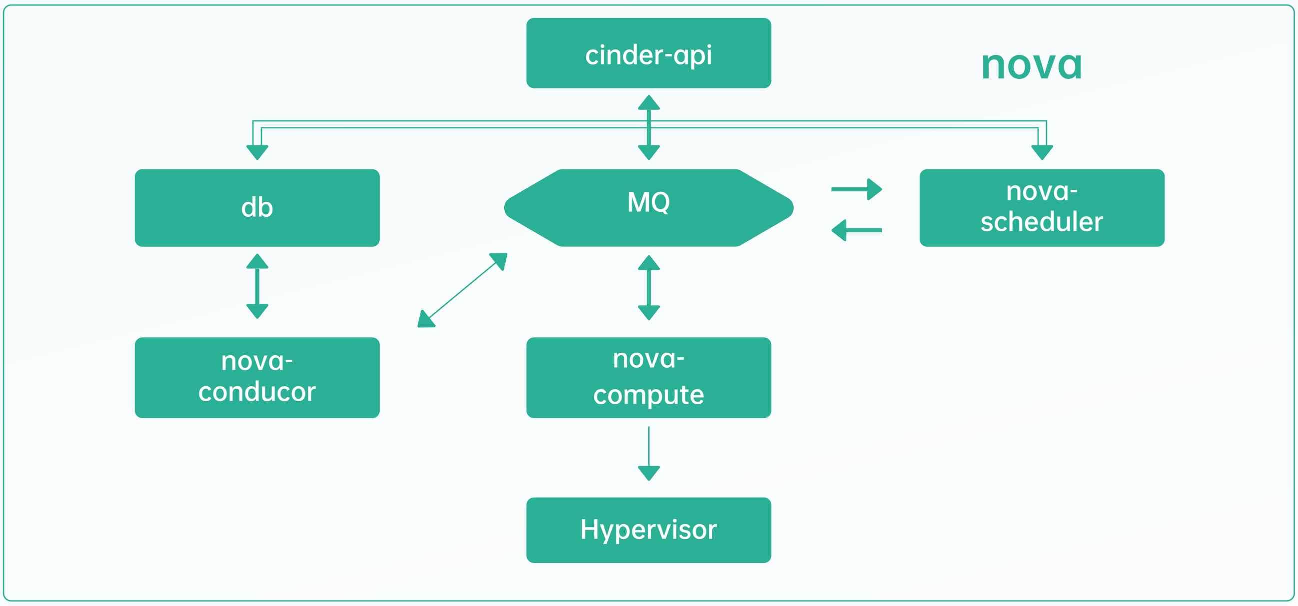
1.The rpc.call sends RPC requests and return the request processing results. As shown in Figure 5, the Topic Publisher sends a message, the Topic Exchange forwards the message to the corresponding Message Queue according to the message address, and the Topic Consumer listens to the Message Queue for messages to be processed and the Direct Publisher will process the request result message and the request sender creates a Direct Consumer to listen to the return result of the message.

2. The rpc.cast sends RPC requests without return, and the request processing flow is shown in Figure 6. The difference with rpc.call is that it does not require the return of request processing results, so there is no Direct Publisher and Direct Consumer processing.

3. The rpc.fanout_cast is used to send RPC broadcast messages without return results.

Nova (similar to that of Cinder)

Flowchart:



Main components of Nova and features

The nova-api receive rest requests.

The nova-scheduler is responsible for scheduling.

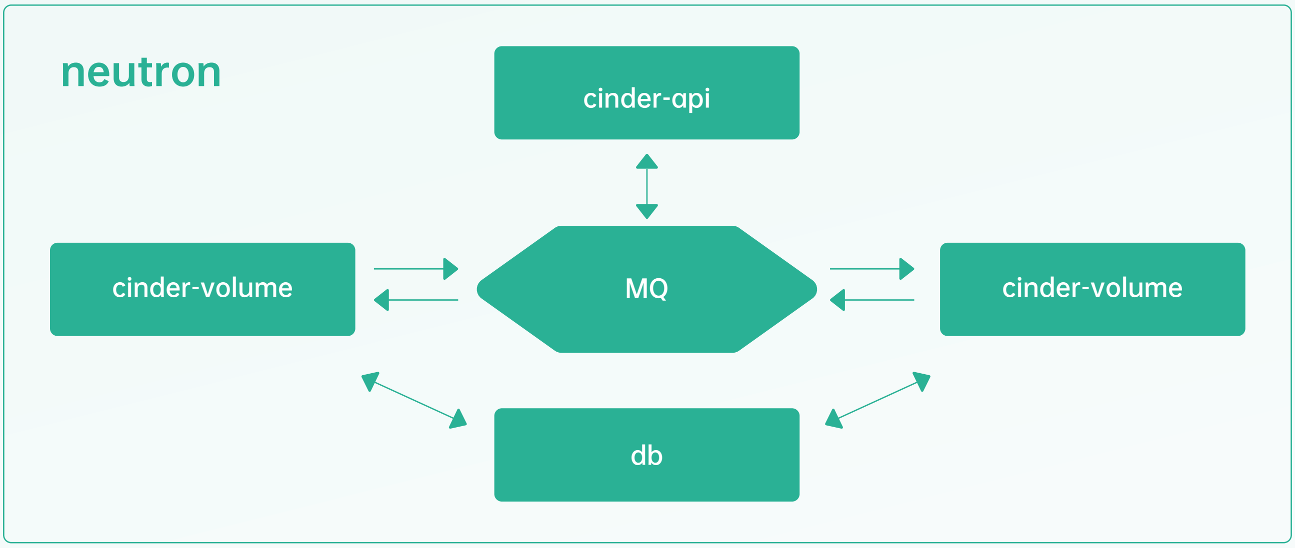
The nova-compute is responsible for calling virtualization drivers, creating virtual machines, etc.

The nova-conductor helps nova-compute access the database and return the query results to nova-compute.

Neutron

On the networking side, OpenStack has gone through an evolution from nova-network (early stage) to Quantum (version F) to Neutron (version H).

Flowchart:



Neutron components and features:

Neutron-server: It provides rest api to the public, receives requests and distributes them to different neutron-plugins.

Neutron-plugin: It processes requests from neutron server, maintains the state of the OpenStack logical network, and invokes the Agent to process the requests. Each vendor developed its own software based on Openstack to emulate its own hardware, and this software is the plugin. In the early days, each vendor developed its own plugin, and the functions were implemented individually, so a lot of code was duplicated; in addition, different vendors had different development standards, resulting in poor compatibility. For this situation neutron-plugin is divided into two parts: Core-plugin and Service-plugin.

Core-plugin: Neutron is ML2 (Modular Layer 2), which is responsible for managing L2 network connections. ML2 primarily includes three types of core resources: network, subnet, and port, and the REST APIs for operating on the three types of resources are regarded as Core APIs by the neutron-server, which is natively supported by Neutron. Among them:

Network: It represents an isolated layer 2 segment, which is a broadcast domain reserved for the creation of its tenants. Subnet and Port are always assigned to a specific network. Types of network include Flat, Vlan, VxLan, Gre, etc.

Subnet: It represents a pool of IPv4/v6 CIDR addresses and their associated configuration, such as gateways, DNS, etc. VM instances in this Subnet will then automatically inherit this configuration. Subnet must be associated with a Network.

Port: It represents a virtual switch port on the virtual switch. The VM's NIC will have a MAC address and IP address once the VIF is connected to the Port. The Port's IP address is assigned from the Subnet address pool.

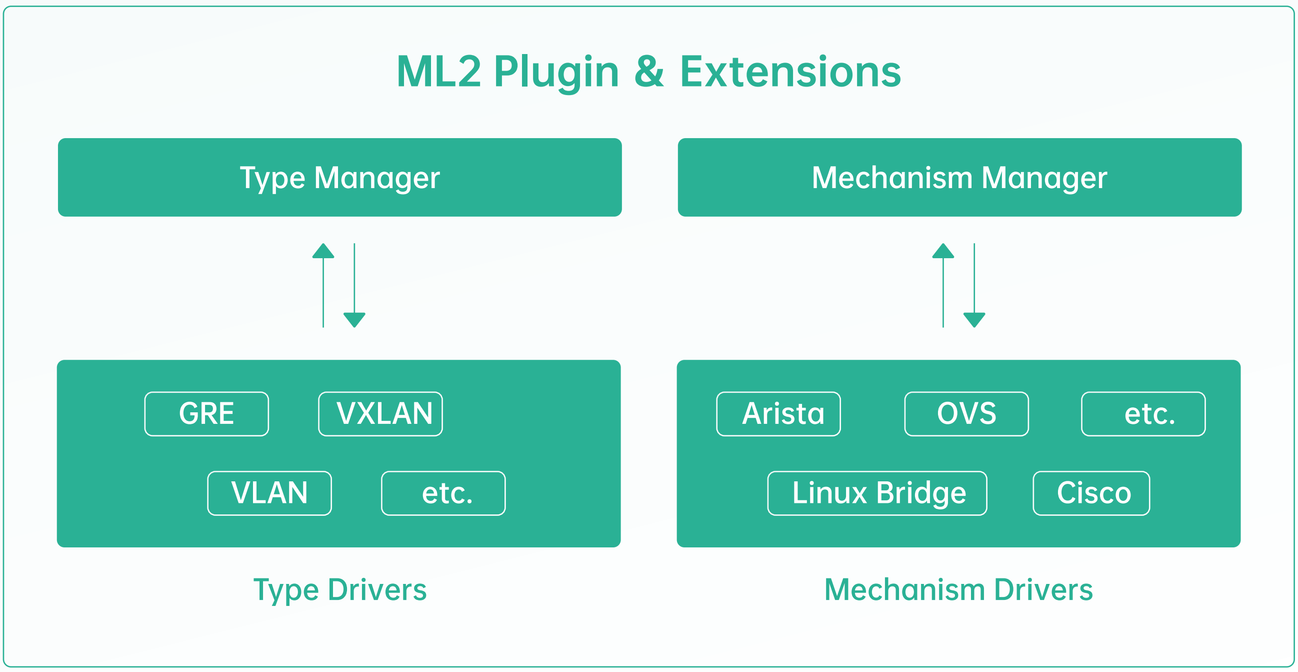
oService-plugin: Plugins other than the core-plugin, including l3 router, firewall, loadbalancer, VPN, metering and so on, mainly for achieving the L3-L7 network services. These plugins are rich in resources to be operated, and the REST API for operating these resources is regarded as Extension API by the neutron-server, which needs to be extended by the manufacturers themselves.

oNeutron-agent: It handles plugin requests and is responsible for actually implementing various network functions on the network provider. There is a one-to-one correspondence with the plugin.

In terms of architecture design, Neutron follows the fully distributed idea of OpenStack, and components communicate with each other through the message mechanism, so that each component and even each process in Neutron can run on any node, as shown in the flowchart above. This microkernel architecture allows developers to focus on the implementation of network services. Currently Neutron provides numerous plug-ins and drivers that can basically meet the needs of various deployments. If these are not enough to support the actual required environment, it is easy to extend the plug-ins or drivers under the Neutron framework.

Neutron optimized Quantum's plug-in mechanism by extracting the independent database implementation from each vendor's L2 plug-in and storing the tenant's business requirements as a public ML2 plug-in, allowing vendors to focus on the implementation of L2 device drivers, while ML2 as the master control can coordinate multi-vendor L2 devices to run together. In Quantum, manufacturers are developing their own service-plugin, which is not compatible and has a high degree of development duplication, so a ML2 mechanism is designed for Neutron, which makes the L2 plug-ins of each manufacturer completely pluggable and facilitates the expansion and use of network resources in L2.

ML2 is the master control of L2, and its implementation includes Type and Mechanism, and each is divided into Manager and Driver; Type refers to the type of L2 network (such as Flat, VLAN, VxLAN, etc.) and is independent of the manufacturer's implementation; Mechanism is the implementation of each manufacturer's own device mechanism, as shown below. If there is a ML2, correspondingly there can be a ML3, but the implementation of L3 in Neutron is only responsible for the function of routing, other functions in traditional routers (such as Firewalls, LB, VPN) are implemented independently, so for the time being, we don't see the actual demand for ML3.



In general, the neutron-server and each neutron-plugin are deployed on the control node or network node, while the neutron agent is deployed on the network node and on the computing node. We will briefly analyze the work of the control-side neutron-server and neutron-plugin first, and then analyze the work of the device-side neutron-agent.

(Note that the L2 plugin and ML2 developed by previous vendors exist in the neutron/plugins directory, while the pluggable ML2 device driver exists in the neutron/plugins/ml2/drivers directory)

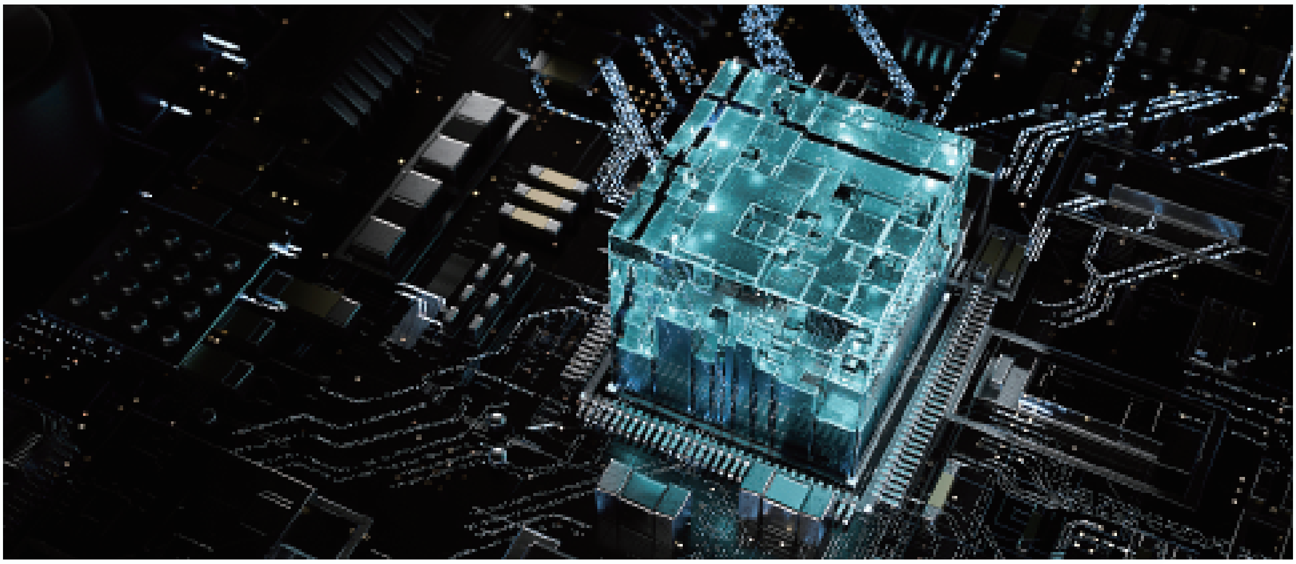
- **Control side implementation**——Starting from the activation of neutron-server, which primarily does two things, the first is to start the wsgi server listening to the Neutron REST API, and the second is to start the rpc service for communication between the core-plugin and agent, which run concurrently as green threads. From the SDN point of view, wsgi is responsible for the northbound interface of Neutron, while the southbound communication mechanism of Neutron mainly relies on rpc to implement (southbound communication mechanisms of manufacturers' plugins may, of course, vary).

In terms of northbound, Neutron's wsgi is templated for deployment via the Paste tool, which receives business requests from the Neutron REST API and then distributes them to the corresponding plugins via APIRouter.

Inside Neutron, the plugin interacts with the database to obtain the global parameters of the business, and then passes the operations and parameters to the Agent on the device through the rpc mechanism (some plugins and ML2 mechanism drivers communicate with the Agent through other means, such as REST API, NETCONF, etc.).

The RPC mechanism can be understood as Neutron's southbound communication mechanism. Neutron's RPC implementation is based on the AMPQ model, and the messages between plugins and agents are usually delivered in the "publish-subscribe" mode, and after agents receive the NotifyApi from the corresponding plugins, they will call back the local CallBack of the device to operate the device and complete the underlying deployment of the business.

- **Device side implementation**——The control side neutron-server receives the northbound REST API requests via wsgi, and the neutron-plugin communicates southbound with the device side via rpc. The device-side agent then communicates upward with the control side via rpc, and downward configures the network device locally.



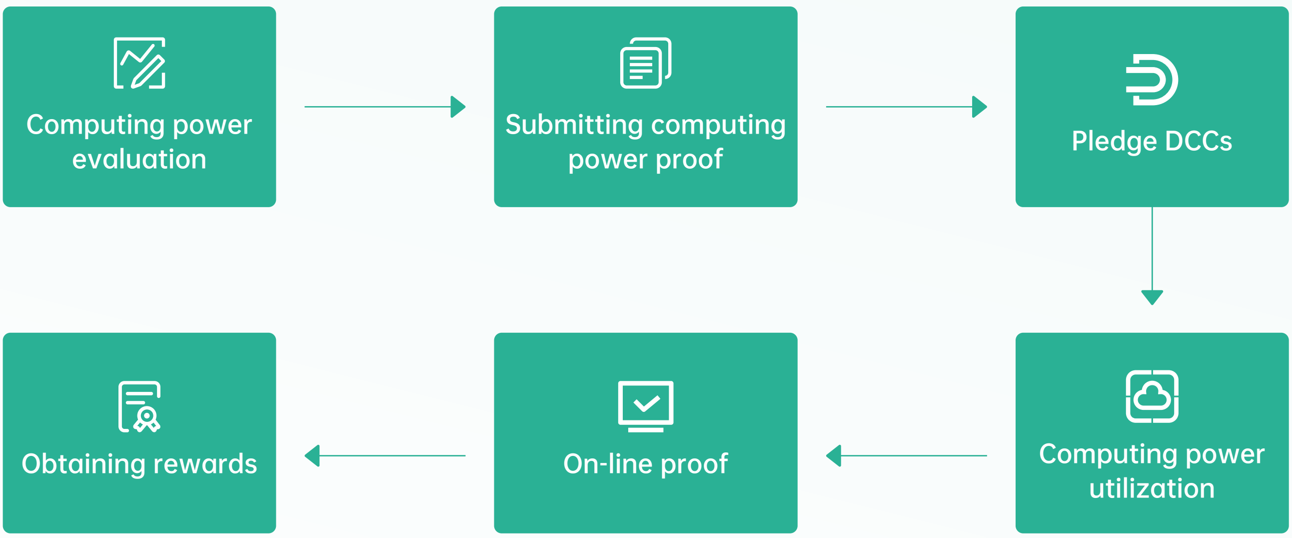
COMPUTING POWER STANDARDS

4.1 Computing power standardization mechanism

·The concept of computing power standardization was mentioned earlier, and because of the diversity of computing power, it is important to standardize the management of which. Initially, we will support two main types of computing power:

1、1.ETHASH computing power   2、Configuration of computing power of fixed computing resources

·For the two types of computing power, we give different weights, both with 1M as the basic unit.



·Regardless of the computing power, we need to go through the above steps to get the rewards.

DCFS ECOSYSTEM

5.1 DCFS ecosystem

5.1.1 Token owners

By using DCCoin as a value storage and exchange medium, making it a viable currency, which provides liquidity for nodes and clients.

5.1.2 Clients

DCCs are required to pay for the computation and transmission of file data. The cryptoeconomics of DCC is ultimately designed to meet our clients' file data computation and transmission needs.

5.1.3 Nodes

Nodes that contribute computing power/communication resources are rewarded with public chain tokens based on their contribution of corresponding computing power resources.

1.Computing power nodes: You can become a computing power node by providing the required computing power resources.

2.Super nodes: The top 100 computing power nodes (ranked according to computing power volume) will run for the super node by voting, with 33 places for each period.

5.1.4 Developers

Developers are the bridge between the clients and the DCFS network, who build applications to improve client access to DCFS.

5.1.5 Ecosystem partners

Applications, development tools and infrastructure can be built on a DCFS network.

5.2 Super node campaign

5.2.1 Campaign rules

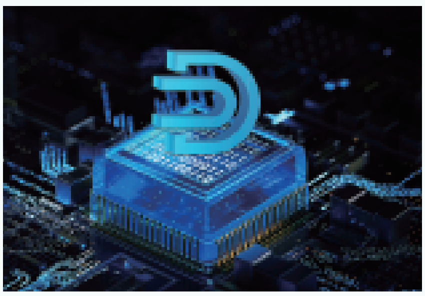
1.Super node campaign can be launched when the number of computing power nodes reaches 500.

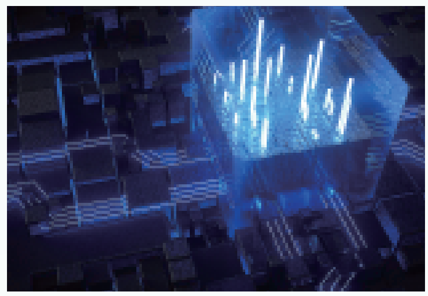
2.Computing power nodes are ranked according to the online effective computing power volume, and the top 100 computing power nodes are candidate nodes and are eligible to run for super node.

3.The top 100 candidate nodes will be locked at the beginning of the voting period, during which super nodes are voted. The computing power nodes with the top 33 votes during the voting period will become super nodes according to the accumulated votes obtained.

4.Term for super nodes voted is 60 days, at the end of which, the number of votes are cleared.

5. The voting period is 7 days during which voting is conducted within the candidate nodes .





5.2.2 Super node campaign matters

1. All DCC owners can vote for the candidate team, each vote requires a certain handling fee. A minimum 1 DCC can be voted, and for every 1 DCC cast, the candidate team will receive 1 vote.

2. Voting will not consume DCC, which will be locked instead.

3. DCCs voted for losing super node candidates will be returned to the users within 7 work days after the end of voting.

4. DCCs voted for the successful super node candidates will be released in two phases within two months after the end of the voting. For instance, by voting 600 DCCs, 300 DCCs will be released each month.

5.2.3 Super node reward benefits

1. Additional block bonus for super nodes during the term for elected super nodes.

2. An election reward during the term.

5.3 Computing power mining reward mechanism

5.3.1 Pledges and computing power mining rewards

This project provides an computing power market platform for the computing power providers to gain revenue. For the users of the computing power, who pay the economic cost to use the computing resource, to protect the rights of the customers, the computing power providers are asked to pledge some DCCs on chain in advance, and when their computing power is successfully used by the customers, the pledged DCCs will be returned to their account after the expiration of the specified pledge period. And when the customers adopt the computing power resources which cannot be used properly during the process or when there is an error, DCFS will automatically deduct part of the pledged DCCs as a penalty. Meanwhile, for providers of stable and high-quality computing power, there will be additional rewards on chain to encourage the provision of such resources.

Pledge

The provider of the computing power is asked to purchase DCCs for pledge before mining. Unpledged users do not receive a revenue, even if they provide the computing power. The pledge ratio is tentatively set at 16 DCC per 1M of computing power. The pledged DCC is returned to the pledge account in a lump sum after 180 days.

Computing power mining rewards

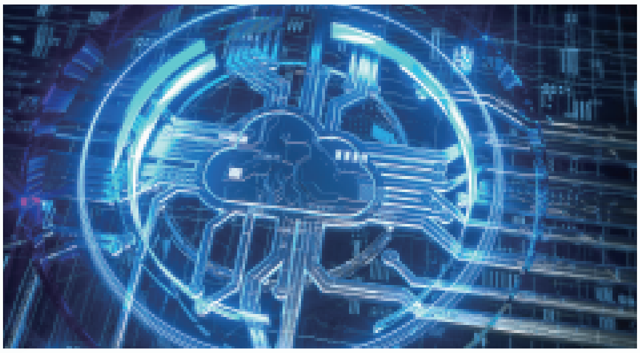
DCFS computing power rewards are divided into four parts, including super node block rewards, computing power node block rewards, computing power node stability rewards, and election rewards.

1.Super node extra block rewards:Receive a block bonus through bookkeeping. The amount is 10% of the basic block rewards.

2.Computing power node basic block rewards:Computing power providers receive a node reward by contributing computing power as well as computing power occupancy, which are allocated based on the network-wide computing power proportion, with the machine computing power expressed as computing power value.

3.Computing power node stability rewards:The stability rewards are issued periodically in a lump sum to the pledged account address after the provision of stable computing power by the provider, the amount of which is 31.25% of the basic block rewards .

4.Election rewards:DCC holders can gain a bonus by being elected as super nodes or voting for super nodes, the amount of which is 15% of the basic block rewards.



5.3.2 Computing power mining output rules

1.64% of which are rewards for contributing computing power resources, i.e. the basic block rewards. The output revenue is released in full on the same day.

2.20% of which is the adopted computing power resources rewards, i.e. the stability block reward, the amount of which is the output revenue released in full on the same day, which is distributed according to the usage rate of the node's computing power adopted; for the contributed computing power not adopted, the reward cannot be obtained and will be automatically destroyed on the day.

3.6.4% of which is the extra rewards for the super node, which is averaged to the daily dividend with each node gains equally, and is released on the same day.

4.9.6% of which is the super node voting contribution rewards, of which 20% of the rewards is distributed proportionately to the 33 super nodes according to the number of votes; the remaining 80% is distributed proportionately to the voters who voted for the elected super nodes according to the number of votes, which is released on the same day.

5.4 Output decrement rule

Total production is halved every 6 years.

5.5 Destruction and penalty mechanism

1.If the contributed computing power is not adopted, stability rewards will also not be available, which will be automatically destroyed on the day.

2. DCCs are required for providing or uploading computing power resources by the computing power nodes.

3. After the launch of mainnet and before the election of super nodes, the rewards (excluding super node extra block rewards and election rewards) will all be automatically destroyed.

4.The computing power resources are not allowed to be withdrawn, disconnected, lost or under attack during the specified pledge period, in the event of which, block rewards cannot be obtained and all pledges will be forfeited and the computing power cleared.

5.For malicious nodes initiating attacks, the mainnet will clear the node's computing power and forfeit all pledged coins.

6.In the event of computing power disconnection, loss or error of the computing power node, if the provider restores the computing power within one day and submits the error report to the mainnet voluntarily, penalty can be waived. For errors remain unfixed for over 24 hours, a penalty of 3-day worth of the online block rewards of the computing power node will be forfeited daily afterwards.

DCCs

DCFS Foundation is formed by experts in the field of distributed computing, cryptography, blockchain, and finance, who aims to serve DCFS public chain and build a global distributed computing network ecosystem. To continuously improve the public chain ecosystem, the Foundation has incubated various applications such as DEX, NFT, DAO and Metaverse to serve the future digital economy.

The token of DCFS public chain is Distributed Computing Coin, or "DCC", with a fixed total number of 500 million tokens that guarantees no additional. **DCC is a decentralized blockchain digital asset issued based on the DCFS, which is the DEP 20 standard token based on the DCFS block-chain.**

6.1 Token allocation scheme

Roles	Proportion	Quantity
DCFS Management Operation and Maintenance Team	1%	5 million
Airdrop (Ecosystem Partners)	1%	5 million
Community Incentives	2%	10 million
DEFI Pool	12%	60 million
Ecosystem Building	20%	100 million
DCFS Foundation	5%	25 million
DCFS Joint Labs	5%	25 million
Computing Power Mining Rewards	54%	270 million

The details are as follows:

1.Total allocable amount for DCFS management and operation team: 1% (5 million DCCs).

a) 0.4% (2 million DCCs) to the world team.

◆ Distribution Rules: Locked for 360 days, and then released in an equal amount daily over 360 days.

b) 0.3% (1.5 million DCCs) to DCFS public chain operation and maintenance staff.

◆ Distribution Rules: Locked for 180 days, and then released in an equal amount daily over 180 days.

c) 0.3% (1.5 million DCCs) to DCFS public chain operators.

◆ Distribution Rules: Locked for 180 days, and then released in an equal amount daily over 180 days.

2. Total allocable amount for airdrop: 1% (5 million DCCs).

3. Total allocable amount for community incentives: 2% (10 million DCCs).

4. Total allocable amount for DEFI pool (pledge or paired pledge output incentive): 12% (60 million DCCs).

5. Total allocable amount for ecosystem building: 20% (100 million DCCs).

6. Total amount allocable for DCFS joint laboratories: 5% (25 million DCCs).

◆ Distribution rules: 15% of the total allocable amount is released annually (until 100%), and the annually amount is released equally each day.

7. Total allocable amount for DCFS Foundation: 5% (25 million DCCs).

◆ Distribution rules: 15% of the total allocable amount is released annually (until 100%), and the annually amount is released equally each day.

8.Total allocable amount for computing power mining rewards (providing computing power resource rewards): 54% (270 million DCCs).

DCFS FOUNDATION

DCFS Foundation is formed by experts in the field of distributed computing, cryptography, blockchain, and finance, who aims to serve DCFS public chain and build a global distributed computing network ecosystem. To continuously improve the public chain ecosystem, the Foundation has incubated various applications such as DEX, NFT, DAO and Metaverse to serve the future digital economy.

COURSE AND ROADMAP

2016-2019
Public Chain Preparation

2019-2020
Economics Discussion

October 2020
Establishment of DCFS Foundation

December 2021
White Paper Release

January 2022
Testnet Launch

January 2022
DEFI Mining/Decentralized Exchange Launch

February 2022
Public Chain Launch

June 2022
NFT Dapps Launch

December 2022
Metaverse Dapps Launch

June 2023
Computing Power Scheduling System Launch

July 2023
Computing Power Mining Available for Testnet

October 2023
Computing Power Mining Available for Mainnet/Super Node Campaign

December 2023
Application of Public Chain Computing Power System in Traditional Industries

January 2024
Development of Various Public Chain Dapps

October 2024
Over 20 Public Chain Dapps

June 2025
Development of Various Public Chain Dapps to Complete the Ecosystem