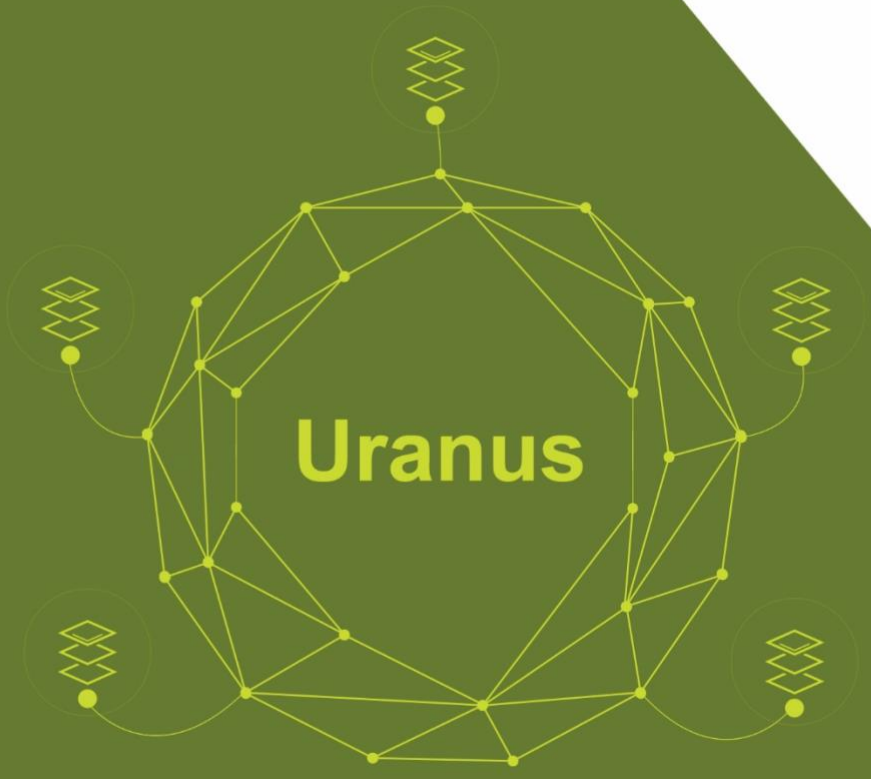
Technical White Paper

Uranus

V2.0



Uranus, the primordial god of sky, was a symbol of hope and future, delivering light and power to every corner in the world.

Table of Contents

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

1 Executive Summary

Uranus like a Uber and Airbnb platform in the computing power resources provides a computing power resource platform of the players, for the players and by the players beyond conventional public clouds.To address the current pain points of public cloud, such as unfit for distributed computing, vulnerability to theft of data assets, high tariff, lack of flexibitity and difficult migration, Uranus seamlessly links and integrates redundant computing power in the world through establishment of an expandable public blockchain and an distributed container technology, and provides efficient, cost-effective and decentralized computing service for users.

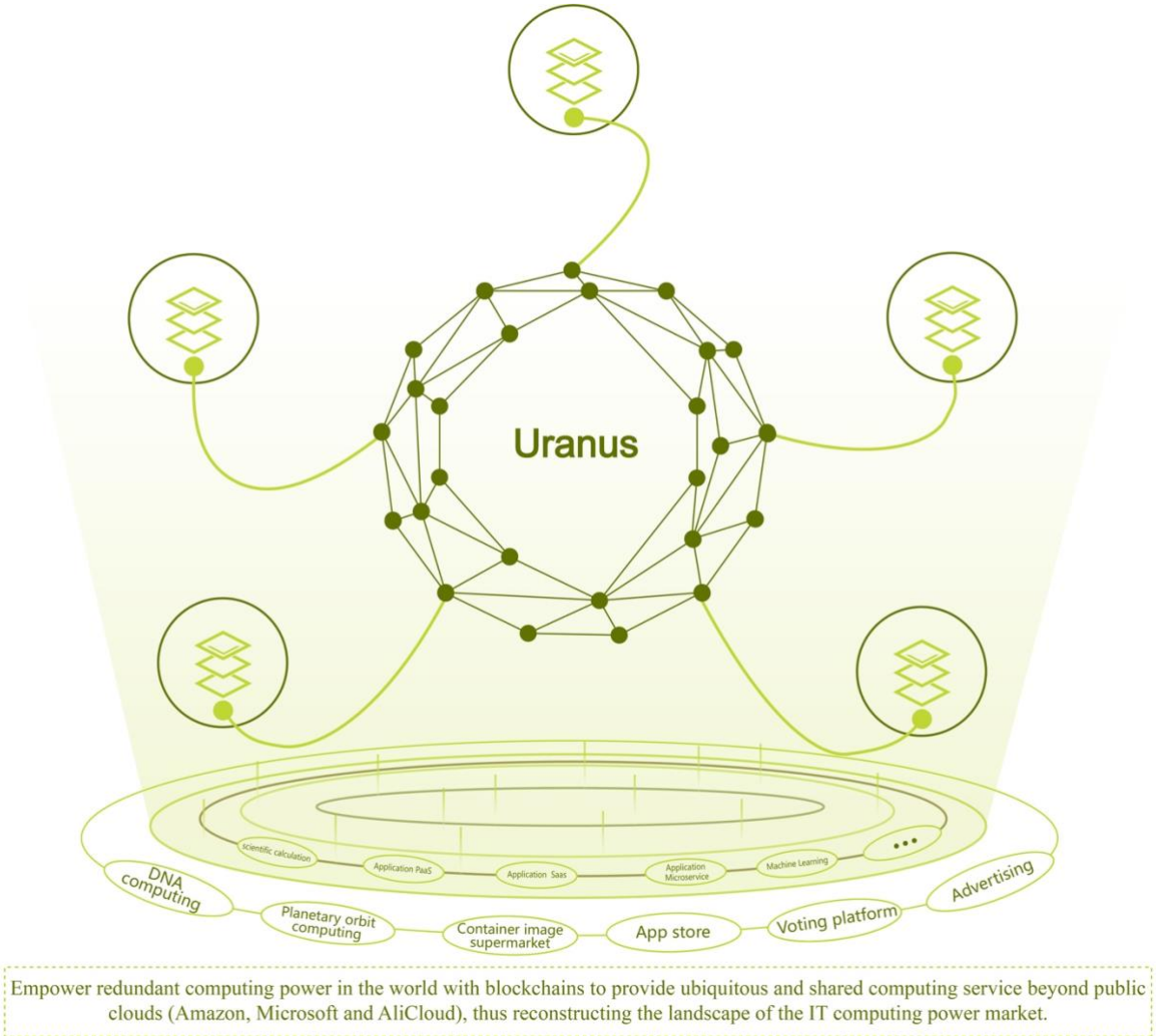


Figure 1 Vision of Uranus

Uranus will be a globalized computing resource platform with hundreds of thousands resource contributors and resources users. It completely changes the conventional centralized cloud computing business model and resource structure thereof. Its computing power resource platform may be used in large computing projects based on

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

distributed computing, such as: protein structure computing in Volunteer Project, computing of new planetary orbits and analysis of weather data. It may also be used in a great many projects based on edge computing, such as: search, voting, games, advertising and IoT node management. This platform may also serve as an underlying service platform for other public blockchains, for example, provide one-click deployment of various chains and applications. Particularly, on this new container architecture for distributed computing, container image supermarkets and application stores may be built, thus constructing an application ecosphere based on distributed computing and edge computing, and creating huge social and economic values.

The members of the executive team of the Uranus project have profound theoretical foundation and average 20 years' practical experience in this field. They have successfully established a few enterprises whose products have entered some of top 20 Chinese corporations. Under the leadership of the executive team, the technology development team of the Uranus project has completed the development and commercial delivery of many major projects over the years. Advisory Board of the Uranus project consists of internationally first-rate technical experts in this field, senior practitioners in crypto economy and leading figures in the media sector, providing further guarantee for the success of the project.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

2 Mission

Empower redundant computing power in the world with blockchains to provide ubiquitous and shared computing service beyond public clouds (Amazon, Microsoft and AliCloud), thus reconstructing the landscape of the IT computing power market.

Use containers to consolidate various applications based on distributed computing to build an ecosphere based on public blockchain.

As Mark Waiser, a proposer of ubiquitous computing, says: Technology should create calm, that which informs but doesn't demand our focus or attention.

At present, on a global scale, resource sharing has become an important aspect of technological progress and social development. Car sharing, housing sharing, and even bicycle sharing have become a reality. Next, people will naturally think of the sharing of computer resources. Although cloud computing with the sharing of computing resources and storage resources as the main purpose has been widely used, large public cloud computing platforms such as Amazon, Microsoft, and Ali, and tens of thousands of private cloud systems emerged in the global IT market, neither the public cloud platforms nor the private cloud systems can perfectly support distributed node-based computing, and the physical resources provided by them do not include a large number of personal PCs, business PCs, data centers of small and medium-sized enterprises, servers, workstations, thin terminals, other embedded terminals, and billions of mobile terminals in the market. The redundancy and waste of computing power are measured in a few hundreds billions USD. Moreover, the monopoly in the public cloud market has become a barrier to technological progress, cost reduction and efficiency improvement. To this end, the use of redundant computing power in the society (as shown in the figure below) will reconstruct the world's computing market and pattern and promote the development of the IT industry. Therefore, Ubiquitous Computers Pool emerged at the right moment.

In the 1980s, American scientist Mark Waiser proposed the concept of Ubiquitous Computing. He envisaged embedding computers in various sizes into everything, and then letting computers serve the people quietly through wireless communications. Ubiquitous computing complies with the following principles:

- The purpose of a computer is to help you do something else;
- The best computer is a quiet, invisible servant;

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

- The more you can do by intuition, the smarter you are; the computer should extend your unconscious;
- Technology should create calm.

From a good idea to implementation, it generally requires technological progress and commercial interests to drive. For example, the Internet of Things (IoT) is the most typical implementation of ubiquitous computing.

The Uranus project has a close relationship with ubiquitous computing. It should be noted that Ubiquitous Computing and Ubiquitous Computers have a slight differences in terms, but their commercial goals to be achieved are not the same. What Ubiquitous Computing wants to achieve is computing everywhere, and what Ubiquitous Computers needs to do is to free up existing redundant computing power. Their common point is what Mark Waiser said when designing Calm Technology: that which informs but doesn't demand our focus or attention.

The commercial interests that drive the sharing of ubiquitously existing computing power are huge redundant computing power in the world. The technological progress that drives the sharing of ubiquitously existing computing power is from blockchain, cloud native and microservice. Thanks to technological progress in blockchain, cloud native and microservice, it makes the sharing of redundant ubiquitously existing computing power possible.

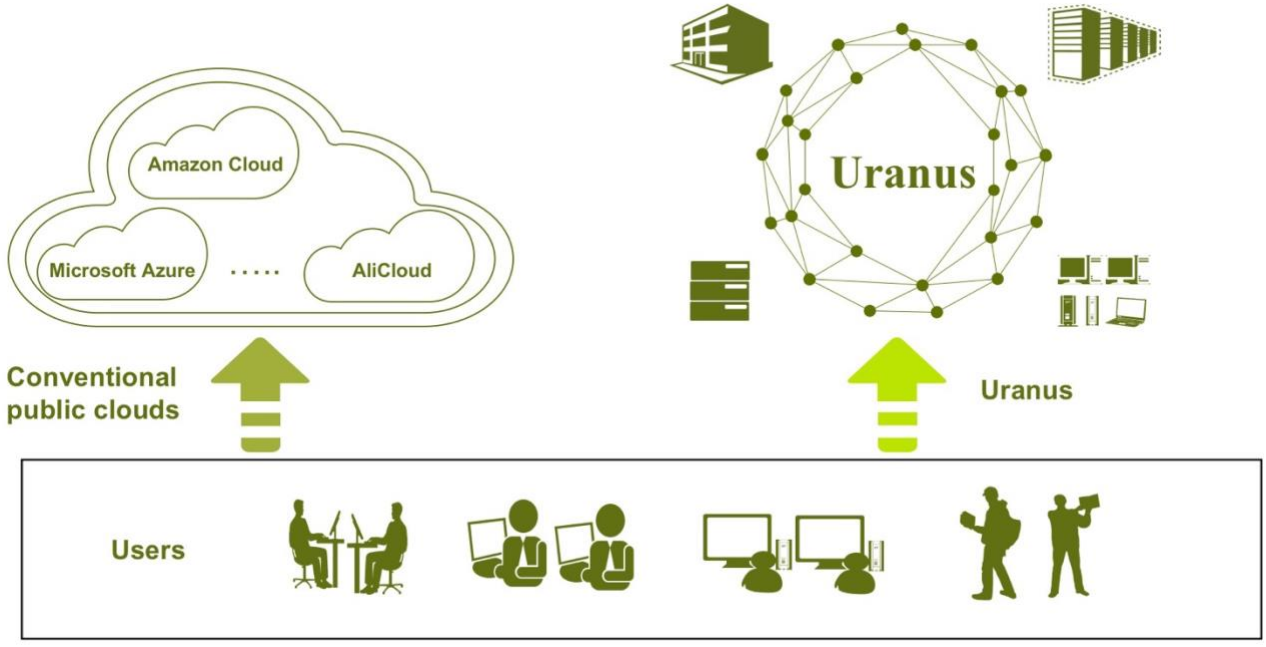


Figure 2 Mission of Uranus

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

In summary, the Uranus system not only builds a public cloud service based on redundant computing power in the society but also is an IaaS established on the basis of blockchain. It can support any future PaaS and SaaS based on blockchain. This platform can also be called as BlockStack, which includes blockchain-based application stores.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

3 Pain Points

3.1 Problems of Conventional Public Clouds

With the rapid development of hotspot technologies such as cloud computing, big data, IoT, smart city, and artificial intelligence, the global public cloud market has also maintained rapid growth. According to the Gartner research report, the global public cloud market had grown from 68.3 billion dollars in 2010 to 307.4 billion dollars in 2017, with the compound annual growth rate maintained above 17%, and the average growth rate is expected to remain above 15% from 2018 to 2020, likely reaching 383.4 billion dollars by 2020. Nevertheless, in 2017, the global public cloud computing market accounted for only 7% of the total global IT expenditures. Even after five years, the share of public clouds in the world's trillion-dollar IT spending market will be still less than 12%. It is because public cloud is facing unprecedented challenges while it is growing at a high speed.

Data Security Risk

For enterprises or users, once their core business data is stored on the public cloud, it is equivalent to handing over the company's "core value" to other people for safekeeping. This is the most worrying security risk for cloud computing users. It is also an important route of leakage of user data. Here is Gartner's statistics on public cloud data security issues for Top500 companies over the past few years:

User name	Public cloud provider	Details of leakage
Arby's	Oracle Cloud	Hackers stole consumer credit card data at Arby'sRestaurant Group (ARG)
The Pentagon	AWS	AWS S3 configuration error, resulting in accidental disclosure of the information of 1.8 billion citizens
Uber	AWS	Data leakage incident of 57 million accounts and the names and driving license numbers of 600,000 American drivers. GitHub obtained the accounts and passwords of Uber engineers in AWS
World Wrestling Federation	AWS	S3 configuration problems
Qudian	Ali Cloud	Data leakage of millions of students
Equifax	AWS	Data leakage of about 143 million users –financial institutions
Accenture	AWS	Enormous password data was stored in four cloud servers without encryption protection, causing leakage of highly sensitive passwords and keys.

Figure 3 Leak incidents of existing public cloud providers

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

The current public cloud platforms may lead to security risks in the following typical situations:

- User data is lost due to hacking of security breaches;
- Data is not encrypted or quarantined during transmission, resulting in information disclosure;
- The user data is not systematically backed up for disaster recovery when it is stored on the cloud;
- Whether users can safely and seamlessly migrate out the data from the public cloud.

Expensive and Unfriendly Billing Model

The billing models of the existing public cloud platforms mainly include annual/monthly model and volume-based billing model. The former can realize centralized management of virtual machines, increase virtual machine density and resource utilization, and improve profitability. However, no matter whether users use it or not, suppliers must provide 365*24 services in accordance with the contract, and the data center costs will remain high. For users, the annual/monthly model looks economical, but in fact it is not. The minimum billing units for CPU, memory, disk, and network are all month, quarter, or year. High-quality IDCs often require users to prepay in advance in a lump. This method requires a large amount of initial investment for users, and it is not easy to adjust according to the actual needs.

Flexible volume-based billing shall be one of the core features of public cloud, but the extensive pricing models daunt users and are miserable for service providers.

Unfit for Distributed Applications

The digital waves are sweeping and innovating in almost all traditional industries. The innovations in education, healthcare, transportation and public services are changing with each passing day. Almost all the data is moving to the cloud, and stored and calculated through cloud and connected to each other via the Internet. Cloud computing is a core support technology of digitalization. However, cloud computing may not be the only path to digitization for various industries, and in particular it cannot satisfy the distributed applications with real-time and local computing needs.

In the conventional cloud computing model, people may click a button and wait for completion of centralized background operations before giving a response to results. This model can satisfy many of the application scenarios, but for some applications that need real-time response between milliseconds, if the existing cloud computing model is used to transmit data to remote cloud terminals via a lagging and jittering

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

network with an uncontrollable distance, obviously it cannot meet the application requirements of real-time computing. At this point, if the network, computing, storage, application and data that are close to the devices or data sources are integrated and the nearest edge computing is used, it will be more suitable for these applications that require real-time computing. In addition to the requirement for real-time response, a lot of distributed computing also needs to be done remotely at the edge.

In the long run, on the way to the Internet of Everything, some application scenarios will use centralized cloud computing, and some scenarios will distribute computing to the edge for the nearest computing. The two will play with their respective advantages and collaborate and complement with each other.

3.2 Redundancy of Ubiquitously Existing Computing Power

Data from IDC shows that ubiquitous computing terminals (personal PCs, commercial office terminals, traditional physical servers, and data center resources) are generally underutilized, as shown in the following figure:

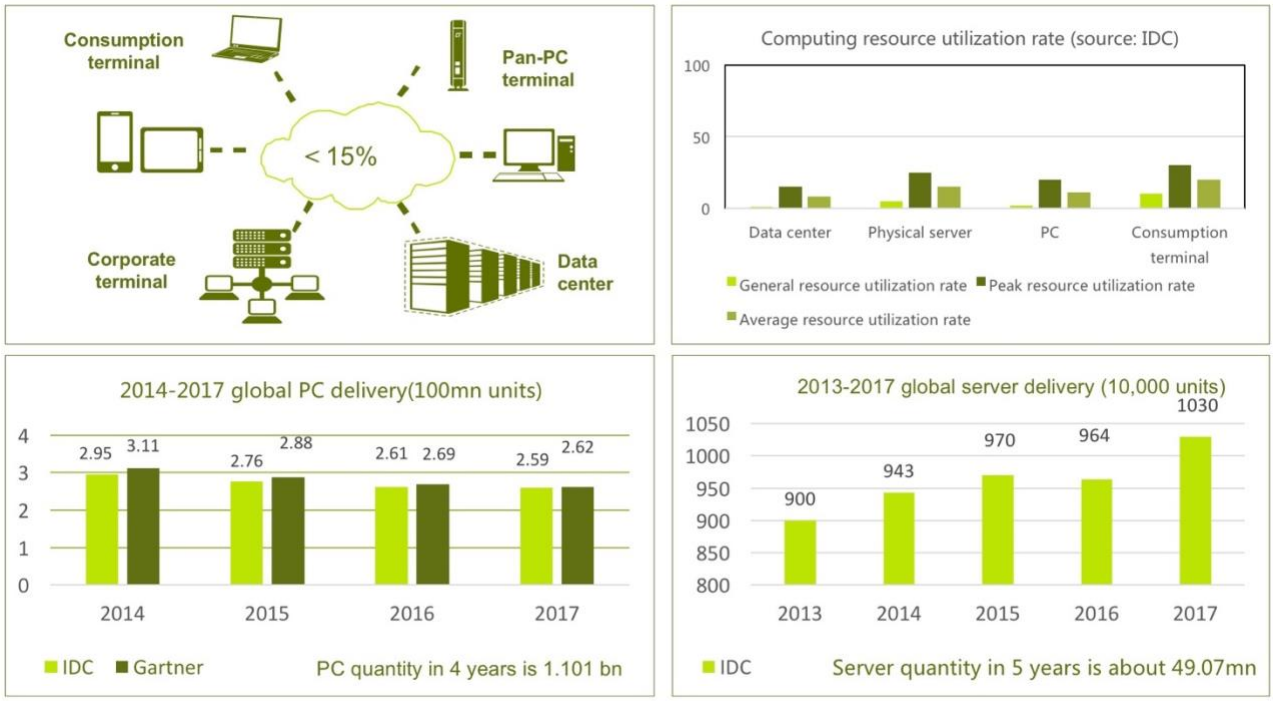


Figure 4 Statistics on sources and volume of global computing power

- The utilization rate of ubiquitous PC computing power resources is less than 15% on average and the redundancy in computing power is severe;
- The current average utilization rate of computing power resources in major public clouds, private clouds, telecommunication operators, and IDC service providers is also lower than 10%-15%, and a large amount of computing resources are idle;

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

- Assuming that the value of a PC is USD 400, then the total computing power of global PCs will have a value of USD 460 billions;
- Assuming the value of a server is USD 8,000, then the total computing power of global servers will have a value of USD 800 billions.

To be specific:

The purchase volume of commercial PCs is about 262 million units a year worldwide. If we suppose the value of a PC is USD 400, the computing resources worth 560 billion* (0.85) = $476 billion are idle. Considering the 5-year lifecycle of PCs, at least USD 2380 billion of computing resources are idle. If servers, workstations and personal PCs are considered, the idle computing resources in the worldwide market will exceed USD 4 trillions.

Therefore, the market for sharing of ubiquitous redundant computing power is huge. Even if part of it is released, the direct and indirect economic value will be on a scale of a few hundred billions. At the same time, it will promote the change of market structure and technological progress in the world.

3.3 The Sharing of Ubiquitously Existing Computing Power Faces Huge Technical and Business Challenges

Past two years in the U.S. market, excellent players ever attempted to use Docker and Kubernets (k8s) to integrate redundant data center resources in the society to form an "alliance" type public cloud platform beyond Amazon Cloud and break the monopoly structure of the American public cloud market. This can be considered as a subset of the sharing of ubiquitously existing computing power in the fields of servers and data centers, but this attempt faces enormous challenge in both technology and business.

First of all, because the contributors of redundant server resources come from different organizations, which are not commercially affiliated with each other, the collaboration and cooperation in the market are difficult. Secondly, such a resource sharing system must have an efficient, trustworthy, secure, real-time transaction mechanism (i.e., Calm Technology) to minimize the cost of sharing and achieve the purpose of effective use of redundant resources. Lastly, because the physical layer and the connection layer of each terminal vary greatly. In order to achieve sharing of computing power, it is necessary to use popular applications on every terminal that participates in the sharing of computing power, and at the same time to effectively maintain and manage the problems encountered during use.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Prior to the application of blockchain technology, the industry could not solve the above first and second problems. Even though the third problem could be solved by “cloud native” and “microservice”, the technical considerations in the aspects of “cloud native” and “microservice” are based on a relatively consistent central computer room. The interworking and interaction between two channels cannot be resolved. How to realize value delivery based on blockchain and interworking and interaction between service delivery channels is also one of the core issues that need to be solved in the future.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

4 Solution

The technical team of Uranus is from two sectors including computing resource pool and blockchain. The experts from the computing resource pool have at least 10 years' experience in the development of open source software, cloud computing, containers, virtualization and terminal systems. Its products have been successfully commercialized and entered the top 20 Chinese corporations, military and government markets. The experts from the blockchain have more than 6 years' practical experience in the development and successful application of the consensus algorithm developed by public blockchain, proof of workload and development of smart contracts. As shown in Figure 5, Uranus is a platform built by the experts from the two teams by integrating the technologies in these two fields to achieve sharing of ubiquitously existing computing power.

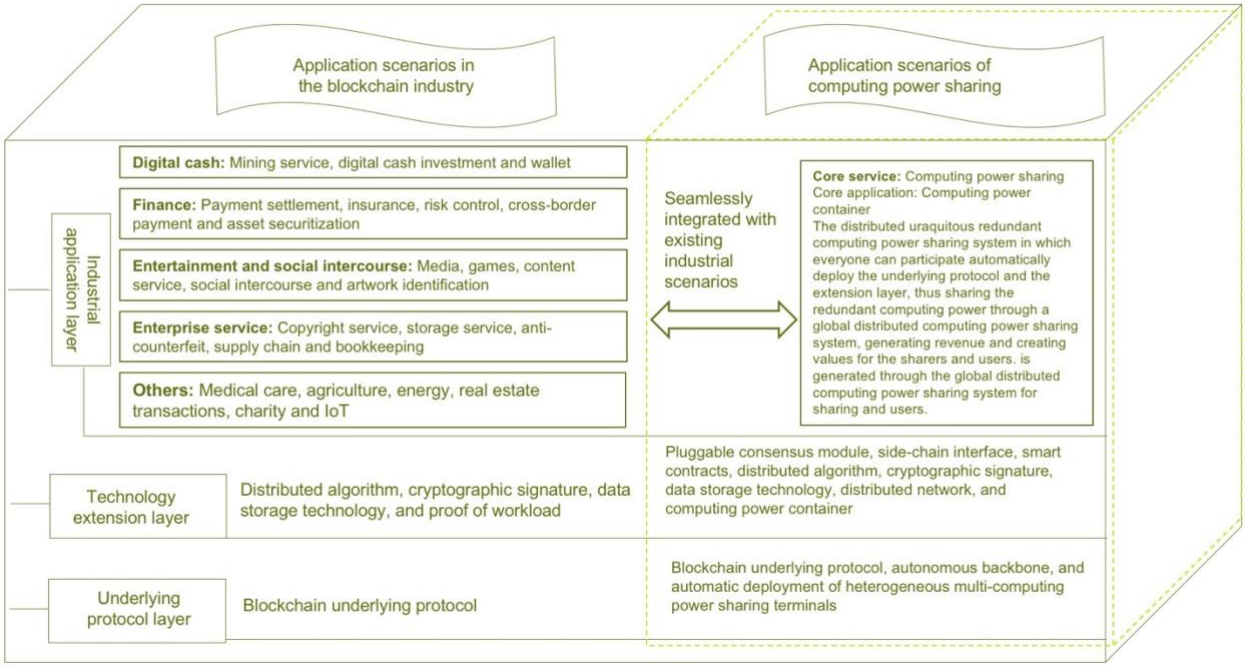


Figure 5 Solution for Uranus platform

Below, the elements of the Uranus platform will be described in brief:

Unit of Measurement of Standardized Computing Power

Computing power is a complex and multidimensional resource, and is related to use time and application. Therefore, standardizing the unit of computing power is a basis for sharing of ubiquitously existing computing power.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Blockchain Connects the Terminals with Redundant Computing Power to Complete Value Delivery

The blockchain can provide efficient, secure, and reliable value delivery between thousands thousands hundreds of thousands terminals that contribute redundant computing power. However, in order to apply the blockchain to the sharing of ubiquitously existing computing power, we need to build a customized public blockchain with corresponding extended functions.

Build a Service Delivery Channel for Terminals with Redundant Computing Power

The blockchain cannot deliver services. For this reason, it is necessary to build an additional channel that can deliver applications and computing results.

Implement the Interworking and Interaction Between Two Channels

Real-time interworking and interaction are needed between two channels. The current public blockchain does not support such interworking and interaction. For this purpose, a new public blockchain needs to be created to add new functions and realize real-time interworking and interaction between the two channels.

Establish a Smart Contract with Rich Content

The extended smart contract not only stipulates the content of the transactions between the computing power contributors and the users of computing power, but also stipulates the handling processes and the handling plans of various possible problems during the sharing of computing power to guarantee each sharing of computing power is completed automatically without manual intervention.

Build Computing Power Containers

The computing power container is a basic configuration that ensures the applications supported by each platform can be implemented successfully after standard computing power units are determined. The core role of this basic configuration is to ensure successful execution and return of results on all the terminals participating in sharing of computing power.

Transform the Existing Distributed Cloud Native Architecture

The cloud native architecture of the existing distributed system is mainly used on centralized and relatively consistent physical terminals. In the ubiquitous terminal environment, this consistency is greatly reduced. To this end, transforming the existing cloud native architecture is an important link to solve the sharing of ubiquitously

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

existing computing power.

Establish Perfect Application Stores

The application stores have all validated APPs, guarantee the feasibility of the applications and can also dynamically balance the load of the terminal system through the load balancing technology.

Own a Mining Pool to Carry Values

The tokens of this system are generated by the volume of shared computing power and the digital function. It stores some of the tokens in its own mining pool that are available to buyers. The tokens obtained by buyers can be used to exchange equivalent computing power.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

5 Technology

The Uranus system aims to provide a decentralized blockchain-based platform for the sharing of ubiquitously existing computing power that is flexible and pay-as-you-go and support user self-service. The design goal pursued by the Uranus platform is:

- System autonomy, no vendor lockout
- User experience is supreme
- Flexible and pay-as-you-go
- High throughput and scalability
- Full support for cloud native

5.1 Architecture of the Uranus System

The Uranus system is a decentralized platform for the sharing of ubiquitously existing computing power based on blockchain technology and cloud native/microservice. It provides a fairer and more cost-effective resource trading market for all resource contributors, resource users, and industry ecology participants. It will gradually reconstruct the existing global landscape of cloud computing currently built by Amazon, Microsoft, Ali and other companies, and promote the rapid development of the next-generation decentralized edge computing and fog computing. Based on the concept of integrating blockchain and computing power sharing, it has innovatively established a new intelligent resource delivery system and value delivery system, as well as Intercommunication and collaboration between the two systems.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

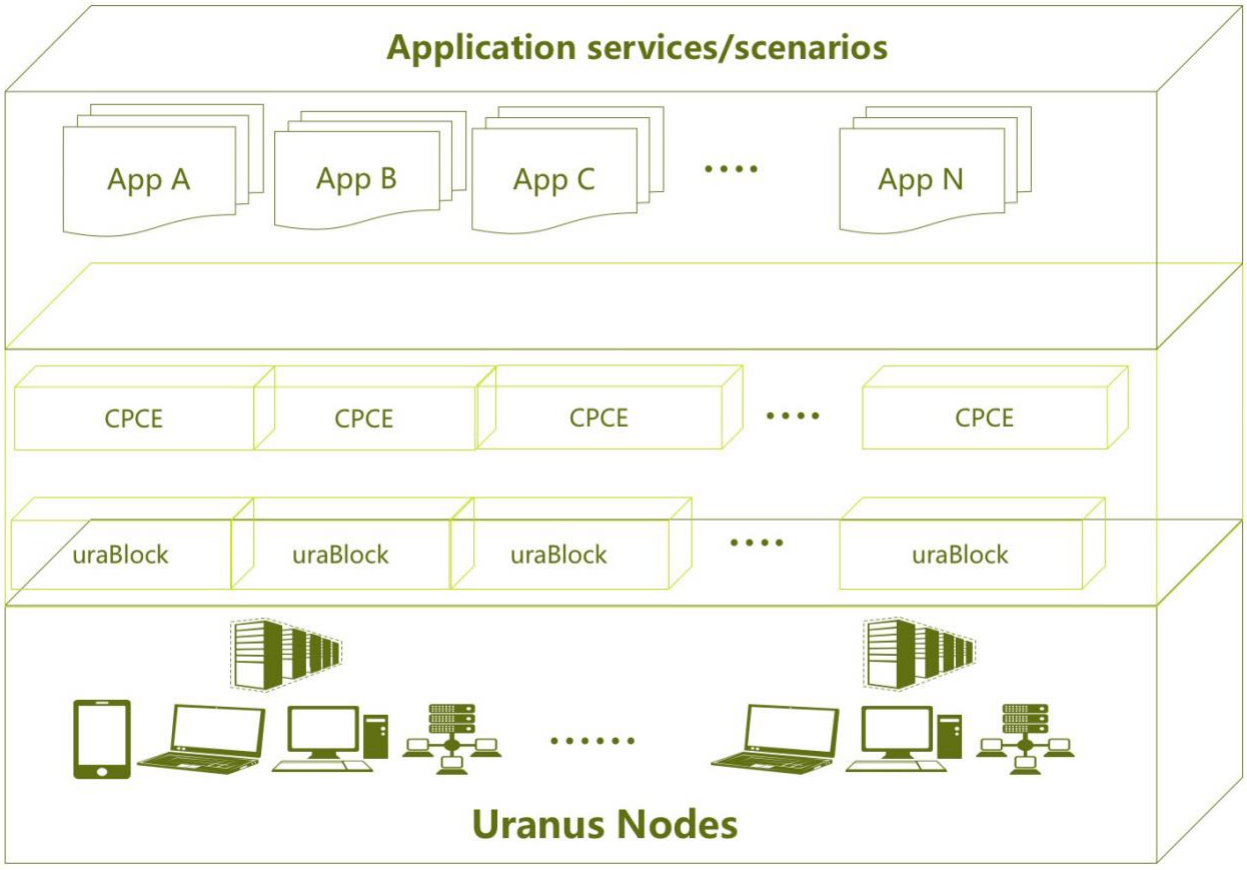


Figure 6 Schematic diagram of the architecture of the Uranus

Uranus mainly comprises a service delivery system and a value delivery system in terms of architecture:

Service delivery layer

- Ubiquitous cloud infrastructure, and standardized container metering
- Use on demand, automatic settlement
- Blockchain-based, decentralized computing resource pool
- High throughput, high availability, and low cost to meet the needs of modern business
- Container-packaged, dynamically managed cloud native applications

Value delivery layer (Uranus chain):

- Basic chain
- uraTiller module
- Enhanced consensus module

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

The core technologies of Uranus center on innovative chain technology for computing power sharing, smart contract technology for computing power sharing, contribution validation mechanism for computing power sharing, computing power engine management scheduling, computing power container Worker, Agent and image, and high-speed P2P network distribution technology.

5.2 Uranus Chain Technology

Executive Summary

Blockchain is a technical system that stores data in a block structure, is maintained by multiple parties, uses cryptographic technology to ensure data transmission and access and realizes data storage. Its decentralizing, trust-free and tamper-proof features make it a first choice for digital asset Ledger in the early days. Today, the basic chain has evolved to meet the high-throughput, high-performance demands at a commercial level while guaranteeing decentralization and trust intermediary. At the same time, vertical industry innovations based on the basic chain have mushroomed and the industry has entered an unprecedented period of rapid development.

System Architecture

The Uranus system is a technical platform for the sharing of ubiquitously existing computing power based on blockchain technology and the increasingly popular cloud native and microservice. It builds a decentralized IaaS&PaaS lightweight operating platform, and provides a broader and more cost-effective resource trading market for all resource demanders and sharers. This will gradually reshape the current entire cloud computing world, and promote the rapid development of the next-generation distributed edge computing and fog computing, where the Blockchain technology is a cornerstone for the construction of value mutual trust and value delivery networks.

The Uranus system is mainly composed of core components such as uraBlock, Uranus management engine, uraTiler, and access gateway. The figure below describes the logical architecture of a complete node in the Uranus blockchain network:

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

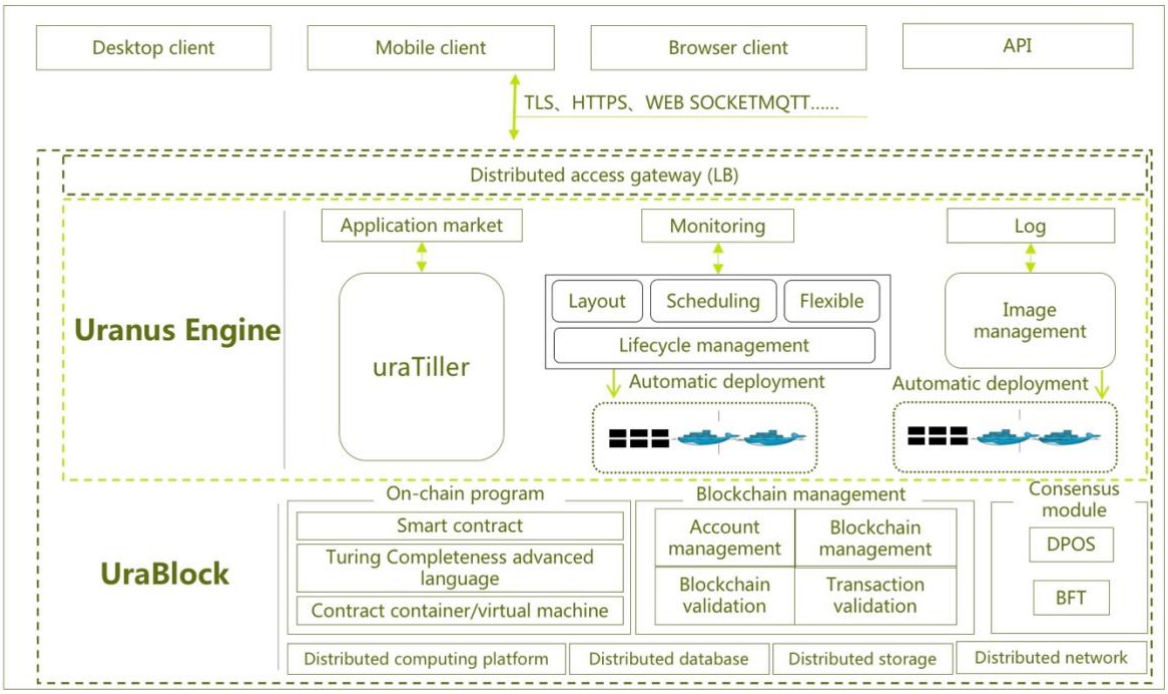


Figure 7 Architecture of the Uranus system

Uranus Chain

The storage layer is responsible for storing all the data on the chain. The data on the chain is stored in the block in a unit of piece and in a form of the Merkel-tree structure. In addition to the traditional transaction data, it also stores the data relevant with computing power containers. The node will write the computing power container information added at each block time into the block, mainly including the following information: container address, the address of the node to which the container is bound, and the resource weight values of the container (CPU, memory, disk resources, etc. are obtained through calculation of optimized algorithm). If the container is already serving externally, the corresponding contract ID will be marked, too.

The network layer is responsible for the interconnection and interworking of blockchain data. Because the statistics and management of the computing power nodes are needed, the data channel and management channel for Computing Power Container Engine (CPCE) are added.

The protocol layer contains a blockchain consensus algorithm. The Uranus chain will use DPOS+BFT technology to build a new incentive mechanism. At the same time, in order to motivate service providers to come up with more and better computing power containers to join the network, we have also added an Proof-of-onlinetime and Proof-of Container mechanism to provide a better incentive mechanism.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

The extension layer is the implementation layer of the main business logics. The operation of smart contract, the support of DApp applications, the management and scheduling of containers, etc., are all implemented here.

Uranus Chain Consensus Algorithm

The Uranus chain consensus module innovatively proposed a DPOS+BFT hybrid consensus algorithm based on DPOS (Delegated Proof of Stake) and in reference of Tendermint algorithm with Byzantine fault tolerance. Under the DPOS mechanism, the influence of each stakeholder is directly proportional to its interests, and no stakeholders are excluded from this influence. Other consensus systems in the market exclude the participation of the vast majority of stakeholders because there are many different ways to replace stakeholders. The popular ones are invitation-only mechanism and bid-ranking mechanism. Some systems technically allow the participation of all stakeholders, but they will be ignored directly by Big players who produce a large number of blocks. Only the DPOS mechanism can ensure that the blocks are evenly distributed to most people, and everyone has an economically viable way to influence these people.

Tendermint is a high-performance consensus algorithm designed exclusively for blockchain. Based on the assumption of a semi-synchronous network, Tendermint can tolerate 1/3 Byzantine nodes. The consensus nodes exchange voting information for each block through the peer-to-peer consensus message exchange protocol, rapidly forming a majority consensus. The voting result eventually will be recorded in the blocks.

The method we use is to firstly randomly sequence the delegate lists on the current round (assure the sequence of the delegates on each round is different and the sequence of the delegates on the next round cannot be predicted), and then let every delegate create a block by round-robin. DPOS currently is a recognized highly efficient consensus mechanism on the blockchain. The main drawback of this algorithm is that if a delegate node is betrayed, it may broadcast multiple inconsistent blocks. These blocks may include Double Speending, resulting in the entire network being forked. In the voting mechanism of DPOS, most users do not vote actively, and it is difficult to find out evil doing in time. In addition, because the DPOS mechanism lacks an economic punishment mechanism against evil doing, it also results in a lower cost in evil doing.

To solve this problem, we took reference to Tendermint's algorithm. Tendermint requires the validator to mortgage certain tokens. If evil doing is confirmed, the deposit will be forfeited. Before Tendermint conducts the next round of voting, the validator

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

will wait for a short time to receive a complete proposal block from the proposer. This dependence on timeout makes Tendermint a weak synchronization protocol rather than an asynchronous protocol. However, the rest of the protocol is asynchronous, and only when more than 2/3 of the validator set is received, the validator will take the next step.

Based on the above analysis, Uranus system DPOS+BFT consensus protocol workflow:

1) Validator registration

During registration, the validator needs to provide a deposit Vb, which must be greater than a minimum deposit parameter B_Min set by the system. At any time, the registrant can withdraw from the registration, but the deposit will have a defrosting period, which can generally be set to be one month.

2) A stakeholder lends the deposit to the validator

Any stakeholder may lend his own tokens to a validator as a deposit. Assuming the deposit that the stakeholder lends the validator is Lb, then the total deposit held by the validator Vt = Vb + Lb

3) Stakeholders elect validators by voting

The stakeholders may vote on the validators. Suppose that the total number of tokens of voters is St and the validator's ranking weight is Wt= St*Vt. If the validation does not receive any votes, its weight will be Wt=Vt.

4) Election for qualified validators

Registered verification nodes maintain a table. The top 48 validators ranked by weight (dynamic validation) may participate in the competitive block generation. The validators in the table are called qualified validators.

5) Package Transaction of qualified validators

The candidate block header hash value Hn is generated using the parent block header hash value Hp, the current transaction Merkle root Hm, and the timestamp T, a random number nonce, and then a Propose transaction is broadcast. Each qualified validator can only issue one Propose transaction per round. The one who repeatedly sends a Propose transaction on the same round will be considered as an evil node. If it is reported, the deposit will be forfeited. Unless the system does not reach a consensus within a specific period of time (consensus timeout parameter of the system), the qualified validator may not change the nonce to regenerate a new block hash value and

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

initiate a Propose transaction.

6) Confirmation of transaction information

After receiving the Propose transaction, each qualified validation node first checks whether the node that initiates the Propose transaction is in the list of qualified validators. If not, it will ignore this proposal. If it is a proposal initiated by a qualified validator, other validators will validate the transactions in this block. If there is any illegal or double-spend transaction in this block, the proposal will be ignored. Finally, the validator will use H20, the last 20 digits of the hash value of the candidate block in the proposal to calculate a voting weight value Vw = Wt/H20, vote on the proposed transaction corresponding to the current maximum Vw, and concurrently send the VoteBlock message. After the original proposer receives VoteBlock messages from more than 2/3 validators, it will initiate a CommitRequest message. Each validation node will validate whether the VoteBlock messages corresponding to the CommitRequest message exceed 2/3. If it exceeds 2/3, Commit of the CommitRequest message will be performed and the Commit message will be broadcast. Each qualified validation node broadcasts NewHeight transactions after receiving more than 2/3 Commit transactions. After receiving a NewHeight transaction, each node will confirm the transaction and update their respective local blockchains. The NewHeight transaction also allows a new block to be finally confirmed and cannot be forked before the block.

The following three attributes are needed to measure the blockchain consensus algorithm:

1) Agreement: Agreement means that all the honest nodes form a consistent decision
2) Validity: Validity means that transactions in the blockchain all come from honest nodes.
3) Liveness: Means that the consensus algorithm of the blockchain can always form a consistent decision without deadlock.
4)

The Fish-Lynch-Patterson (FLP) Law proves that in an asynchronous communication network environment, since it is impossible to judge whether a process is crash or message delay, as long as there is an unreliable process, no consensus can be reached on all non-failure processes.

In the actual situation, it is generally necessary to make some assumptions in order to avoid the non-consensus limitation of FLP.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

The Uranus consensus algorithm is assumed in a weakly synchronized network environment. Among any selected 48 qualified validators, the validators less than 1/3 are Byzantine nodes. Below we show the proof for the agreement, validity and liveness of the Uranus consensus algorithm under this assumption.

1) Proof for agreement

In any round of consensus process, the blocks of Uranus block are generated by 48 qualified validators. The final confirmation of a block requires the votes of more than 2/3 of the 48 validators, including reception of more than 2/3 of VoteBlock and Commit messages. If there is a fork, that means at least more than 1/3 of honest validators have voted for two or more blocks on the same round (because in the most extreme case, it is assumed that 1/3 of the Byzantine nodes have voted for two nodes). It runs counter to the assumption that the honest nodes must execute the voting strictly according to the agreement, so forks won't appear.

2) Proof for validity

Assuming that there is a block generated by Byzantine validator in the Uranus blockchain, and the block contains one or more illegal transaction, or one or more double-spend transaction, then at least 2/3 of the validators have voted on this block in the VoteBlock and Commit stage. Assuming that 1/3 of the Byzantine validators have voted on this block, another 1/3 of the votes must come from non-Byzantine validators. Since we assume that non-Byzantine validators must vote according to the protocol rules, there is a contradiction, thereby proving that Uranus's blockchain does not include illegal transactions.

3) Proof for liveness

Since each new block requires 2/3 of the votes for confirmation, in some extreme cases, for example, the network is delayed, or less than 2/3 of honest validators are online, a consensus may not appear on some consensus rounds. The system will set a time limit for each round of consensus. After timeout, each validator will re-initiate a new block Proposal. If there are repeated situations in which the system cannot reach a consensus, the community needs to initiate new voting to remove the offline validators and reselect validators.

Although Uranus cannot fully guarantee Liveness at any time, it accords with FLP's non-consensus law, and at the same time, Uranus's governance mechanism can be used to avoid the failure to reach a consensus.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

The DPOS+BFT consensus algorithm of Uranus is an innovation based on the DPOS and Tendermint consensus algorithms of BitShares/EOS. It not only avoids the weak points of excessive concentration of validators in BitShares and EOS (101 validators in BitSares and 21 validators in EOS generate blocks in turn) and lack of measures for economic sanctions against evil validators. Meanwhile it also avoids the unfairness in the Tendermint consensus algorithm. That is, the more the interests there are, the more likely the blocks are generated. The consensus algorithm of Uranus increases the randomness to random generation of hash values similar to Bitcoin, and reduces the probability of evildoing of validators.

uraTiller module

The uraTiller module contains Tiller virtual machine and Tiller native smart contracts and interfaces of external data, and logically links on-chain data and off-chain data. uraTiller has innovatively designed a nested virtual machine model that realizes seamless integration between the Tiller virtual machine and the underlying virtual machine to ensure data security and efficient execution of complex business logic.

Because most of the component and modules of the Uranus system have realized the microservice and containerization, and the communication among all microservices use a message bus, the interaction between off-chain data and other modules may realize functional expansion by subscribing the messages on the bus.

The uraTiller microservice module communicates with the Uranus engine and delivers the confirmed business-related information to the Uranus engine. If the user's business information meets the default scheduling and allocation of the system, the computing power container will execute it automatically. Otherwise, the system will guide the user to Complete a transaction with a third-party supplier, establish a smart contract, and execute the user's special requirements and deployments.

Uranus Computing Power Engine

The Uranus computing power engine module is a infrastructure module used for large-scale scheduling, and automatic deployment, extension and management of containers. The Uranus computing power engine module draws on the design concept of Borg and Kubernetes. It can:

- Automatically deploy and reproduce containers
- Flexibly Scale-out
- Manage containers across machines and organize better load balancing of containers

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

- Dynamically upgrade application containers
- Self-repairing capability

Uranus Smart Contract

Smart contract is a set of promises defined in a digital form, including agreements in which the contract participants can execute these promises. The Smart contracts in the Uranus project are mainly used to achieve conditional limitation and matching of users and computing resources, definition of fault tolerance conditions, and event triggering.

In the Uranus system, two types of Smart contracts are operated. One type is explanatory Smart contracts similar to Ethereum, and the other type is native Smart contracts in the uraTiller module. The native Smart contracts of uraTiller can access a variety of data sources beyond the blockchain, such as various APIs, file systems, URLs, databases, and other data sources, as well as the business logics completely irrelevant with chain. Explanatory Smart contracts are only used to process user transaction information.

In the Uranus system, Oracle runs as a native Smart contract. In this way, the contract can be directly converted into a machine code and executed, greatly improving the execution efficiency of the contract and satisfying the complex business logic scenarios in the cloud environment.

5.2.1 PoC (Proof of Contribution) Mechanism

Uranus uses a PoC algorithm to schedule the computing power containers entering the chain. Machines with a higher contribution value are more likely to be scheduled and allocated to users in priority, thus earning more tokens.

Contribution Value Calculation Method:

Total contribution value of containers = basic contribution value + service contribution value + chain contribution value

Basic Contribution Value:

The basic contribution value is obtained through multi-dimensional scoring and statistics of the container's CPU computing power, uplink bandwidth, sharable storage space, actual available memory, effective online duration and other information.

CPU computing power is scored based on CPU logic frequency by gradient. The

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

memory is scored based on the current effective remaining memory by gradient. The uplink bandwidth is scored based on the measured value by gradient. The available shared memory space is scored based on measured value by gradient.

Basic contribution value =

CPU gradient value * CPU coefficient + memory gradient * memory coefficient +bandwidth gradient*bandwidth coefficient +storage gradient*storage coefficient

The value of each coefficient (CPU coefficient, memory coefficient, etc.) is comprehensively considered by analyzing the current total volume of this resource in the network and the current usage amount of this resource in the network, and analyzed and obtained by using Poisson distribution queuing algorithm. For the introduction to the algorithm section, please refer to: 4.2.2 - Mechanism for automatic adjustment of system resource allocation.

Service Contribution Value:

Service contribution value obtains a comprehensive score through statistical analysis of the container's past transaction records and user's use feedback.

Chain Contribution Value:

Chain contribution value is obtained when a container account as a member of the chain makes contribution to the chain. For example, the account actively participates in the voting of the entire network and maintains the stability of the chain network.

The total contribution value of a single container is accumulated with the increase of online duration. The absolute value is halved after passing through T blocks. The scheduling is realized through partition scheduling round robin algorithm. The secondary algorithm is similar to the system thread queue scheduling of the modern operating systems (Windows and Linux both use this algorithm and its modifications). The advantage of this algorithm is that it allows better container resources to provide more execution opportunities, but it also provides certain implementation guarantee for containers with relatively low performance, so long as the time is long enough, it can still have a chance to be scheduled.

Due to the limited space, this section does not discuss the partition scheduling round robin algorithm. Interested readers may refer to related books.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

5.2.2 Mechanism for Automatic Adjustment of System Resource Allocation

As user's demand is random and the resources in the system are relatively fixed for a specific period of time, in order to ensure the stable operation of the system, a mechanism for automatic adjustment of system resource allocation based on the principle of negative feedback is needed. A resource allocation weight factor σ (Sigma) is used.

Container score=cpu metering$^{\times\sigma_c}$ + disk metering $^{\times\sigma_s}$ +bandwidth metering$^{\times\sigma_b}$

Where: σ_c, σ_s and σ_b are the weight factors of cpu, disk (storage) and bandwidth. These factors are a real-time reflection to the extent of use of system resources. When the system is small, these factors are easily calculated. However, when the system exceeds a certain scale, the calculation of these factors not only consumes a large amount of resources, and cannot be real time, and even the actual values cannot be calculated. For this reason, the maximum likelihood parameter estimation method in probability and mathematical statistics is needed. These factors can be relatively accurately calculated in a small sample size.

Assuming that the unit measurement demand of CPU, the unit measurement demand of disks and the unit measurement demand of bandwidth that appear in unit time are X_c, X_s and X_b respectively, then when system resources are relatively stable, these three variables are all random variables that comply with Poisson Distribution. It is a very classic mathematical model of the queuing theory. Of course, after the system is established, when the sample size is large enough, we need to do a statistical test on this assumption.

Below we will briefly introduce Poisson Distribution to educe the estimation methods of these three factors σ_c, σ_s and σ_b.

It is assumed that x complies with Poisson Distribution. As the value of X here is an integer, X complies with discrete Poisson Distribution, and the probability of x=k (k=0, 1, 2, 3, …, ∞) is:

$$P(x=k) = \frac{\lambda^k}{k!} \cdot e^{-\lambda}, k=0, 1, 2, 3, \ldots \infty$$

Here, λ is an interesting parameter. It is both the expected value of the Poisson distribution function and the variance of the Poisson distribution function. The

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

expected value of a random variable is the "maximum possible value" of a random variable, and the variance is the size of the spread range. The expected value and variance of the Poisson distribution function can be calculated by the following formula (omitting the derivation process)

Expected value $$E(X)=\sum_{k=0}^{\infty}\left(\frac{\lambda^k}{k!}.e^{-\lambda}\right)\times K=\lambda$$

Variance $$D(x)=E(x-\lambda)^2$$

$$=\lambda$$

In the above calculation process, it is assumed that λ is a known constant, but in actual applications, it is an unknown parameter. Its value is estimated by using the maximum likelihood parameter estimation method in mathematical statistics and some samples. Below we will introduce the maximum likelihood estimation (omitting the derivation process)

Assuming that there is a set of sample observations from the population, the maximum likelihood function is

$$L(x_1,x_2,...,x_n\lambda)=\sum_{i=1}^{n}\frac{\lambda^{x1}}{x_i!}e^{-\lambda}=e^{-n\lambda}\sum_{i=1}^{n}\frac{\lambda^{xi}}{x_i!}$$

get $$\hat{\lambda}=\bar{\dot{x}}$$

i.e., maximum likelihood estimator of λ

The above is just a simple theoretical explanation. In practical applications, whether the three factors mentioned above accord with the Poisson distribution needs to be tested. The sample size and time period of the test shall be calculated accurately. The maximum likelihood estimator is based on the sample size. In this sample size, the error of the maximum likelihood estimator needs to be further calculated. It may be necessary to have an empirical correction factor in addition to the theoretical calculation. Whether this correction factor converges or not and whether the negative feedback adjustment mechanism is satisfied or not need more detailed mathematical derivation. They are not described in this document.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

5.3 Computing Power Container Engine (CPCE)

The distributed nodes of the Uranus system are an integral part of the global Uranus resource pool. They constitute the cornerstones of a distributed computing power resource pool that is able to match giant vendors for centralized cloud service (Amazon, Microsoft, Ali, etc.) and are also carriers of ubiquitous computing service supply. The core components of Uranus include computing power nodes, computing power containers, container Worker and Agent, and peripheral distributed auxiliary systems.

Uranus CPCE (Computing Power Container Engine) also runs on the nodes that make up the Uranus blockchain. The Uranus CPCE provides an application-oriented container cluster deployment and management system. The goal of Uranus CPCE is to eliminate the burden of scheduling of physical/virtual computing, network and storage infrastructure and allow the application operators and developers to fully focus on container-centered primitives for self-service operation. Uranus CPCE also provides a stable, compatible foundation for building customized workflows and more advanced automation tasks. Uranus CPCE has perfect cluster management capabilities, including a multi-level security protection and access mechanism, a multi-tenant application support capability, a transparent service registration and service discovery mechanism, a built-in load balancer, a failure discovery and self-repair capability, rolling upgrading of service and online capacity expansion, a scalable resource auto-scheduling mechanism, and a multi-granular resource quota management capability. Uranus CPCE also provides perfect management tools covering development, deployment testing, and operation and maintenance monitoring.

5.3.1 Computing Power Container Engine Architecture

The Uranus CPCE management engine draws on Borg's design concepts, such as Pod, Service, Labels, and single-Pod single IP. The overall architecture of the Uranus engine is very similar to that of Google Borg and Kubernetes, as shown in the following figure:

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

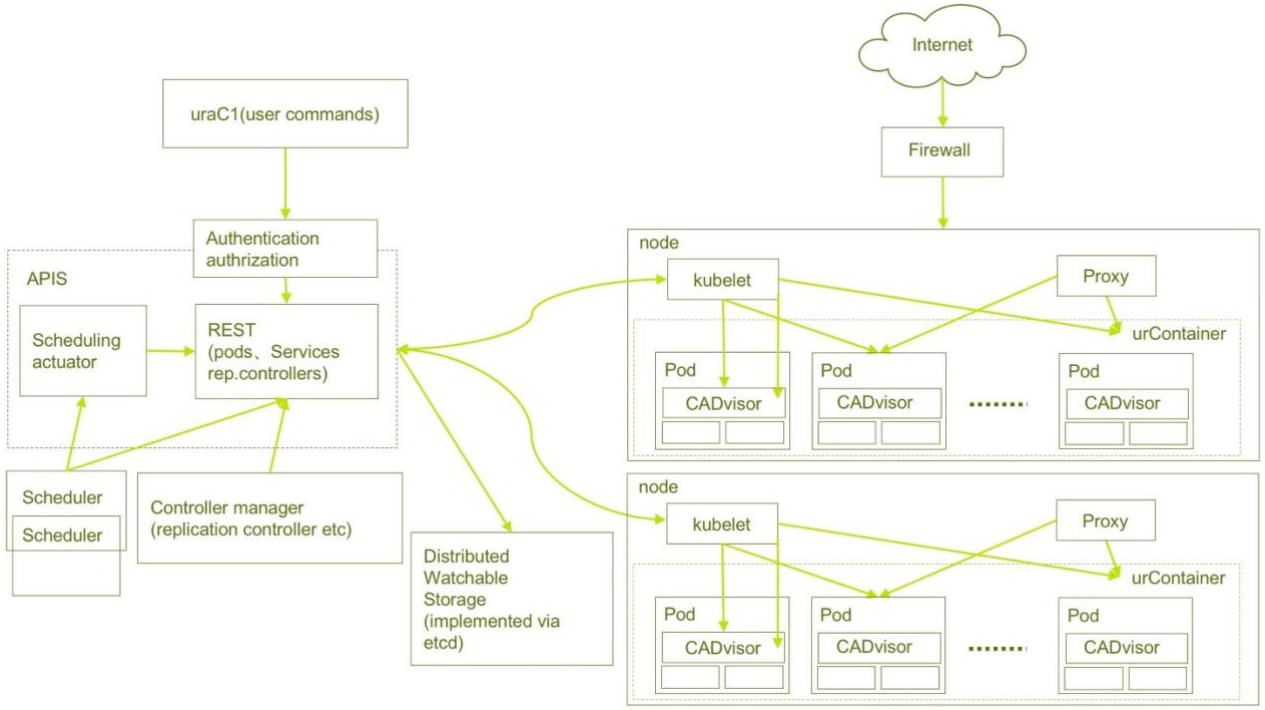


Figure 8 Uranus CPCE architecture

The Uranus CPCE engine comprises the following core components:

- etcd saves the state of the entire cluster;
- Apiserver provides a unique entry for resource operation and the mechanisms for authentication, authorization, access control, API registration, and discovery, etc.;
- The controller manager is responsible to maintain the status of the cluster, such as fault detection, automatic expansion, and rolling updates;
- The scheduler is responsible for resource scheduling, and schedules PoD to the corresponding machines according to the predetermined scheduling strategy;
- kubelet is responsible for maintaining container's life cycle, and is also responsible for the management of Volume (CVI) and network (CNI);
- Container runtime is responsible for image management and the real operation of the Pod and containers (CRI);
- kube-proxy is responsible for providing service discovery and load balancing within the cluster.

In addition to the core components, there are some recommended Add-ons:

- kube-dns is responsible for providing DNS services for the entire cluster;
- Ingress Controller provides external network access for services;
- Heapster provides resource monitoring;

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

- Dashboard provides GUI;
- Federation provides clusters across the availability zones.

5.3.1.1. Uranus Resource Control Management

Through the addition of a separate Resource Controller, the Watch mechanism adopts a full asynchronous non-blocking full event-driven model. In case of insufficient resources, an asynchronous resource request will be initiated to automatically handle the subsequent processes. Once the resources are ready, rescheduling will be triggered at once. During the request, the intermediate layer also prepares the resource pool in advance, initializes Worker and prepares the installation steps in the container image in advance. The creation process is as follows:

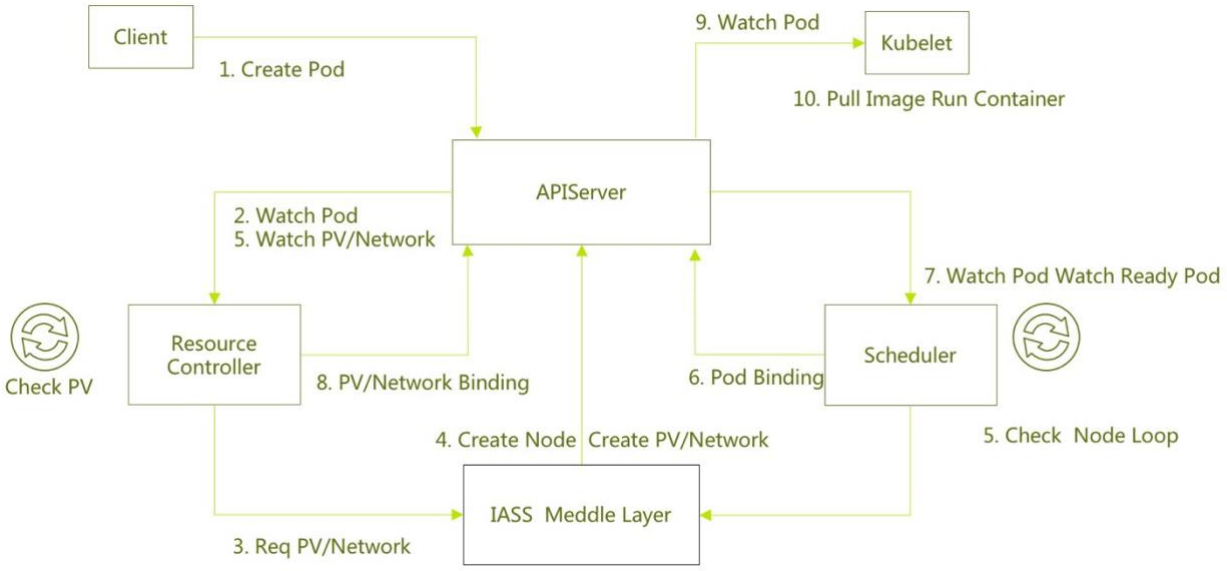


Figure 9 Uranus resource control management

Resource Controller manages all the resource-related underlying working. The specific Pod creation steps are as follows:

- The upper client requests apiserver to create a Pod;
- The Resource Controller watches a new Pod and checks whether PV and Network have been created. At the same time, scheduler also finds that the new Pod is not scheduled and tries to schedule it;
- Because the resources were not prepared in advance, initially the Resource Controller didn't find PV and Network that matched the Pod. It requested the IAAS middle layer to create storage and network resources.

  Scheduler also requests the IAAS middle layer to create a container resource of the corresponding specifications because it cannot find a schedulable node. In this case, the Pod does not reenter the scheduling queue any more. The rescheduling

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

will be conducted only after everything is ready.

- Generally, the IAAS meddie layer creates resources slowly, and only provides asynchronous interfaces. PV, Network, and Node resources will be registered through apiserver immediately after the underlying resources are ready.
- When the ResourceController finds that both PV and Network are ready, it will bind them to Pod. After the Node applied for by Pod is registered, and PV and Network are bound, Pod will be set in a ResourceReady state.
- When Scheduler once again watches Pod in a ResourceReady state, it will retrigger the scheduling process;
- Successful Pod scheduling is bound with new dynamic creation of Node;
- When the kubelet corresponding to Node watches that the newly scheduled Pod has not started yet, it will pull the image at first and then start the container.

5.3.1.2. Multi-Region Scale Computing Power Container Management

When the scale of computing power containers is large, applications and services will be very numerous and the deployment will be extensive. Basically, it will cover all the regions in the world, and the network connectivity in different regions is very different. Moreover, the release is frequent and the iterative cycle is short. In this case, it is considered to deploy computing power container management across multiple regions to reduce the management, operation and maintenance costs of each computing power pool.

In the management of computing power containers, cluster scheduling-related API is added to the Uranus engine to optimize the status management of sub-clusters, and better perform distributed federal coordination processing tasks, as shown in followingfigure:

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

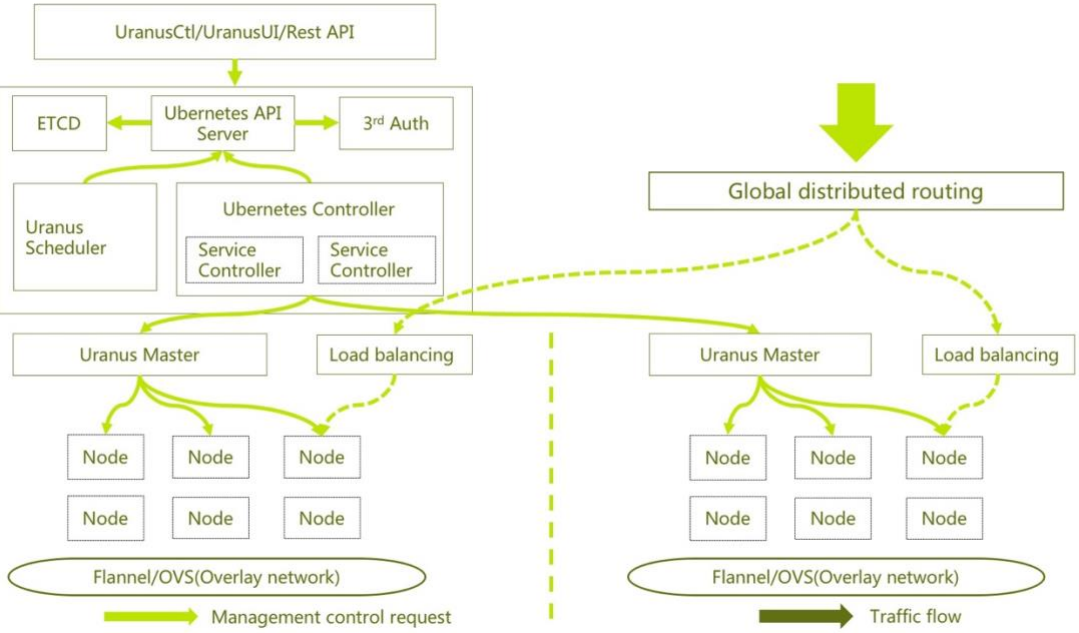


Figure 10 Uranus multi-region scale computing power management

In the original architecture, a Node manages a Worker node. Here, a Node manages multiple clusters. The Cluster Controller will maintain the health status of every cluster and monitor load status, while the Service Controller will find cross-cluster services.

We further promoted list-Watch, improved asynchronous message delivery in distributed systems and greatly enhanced key performance indicators such as system performance and data agreement.

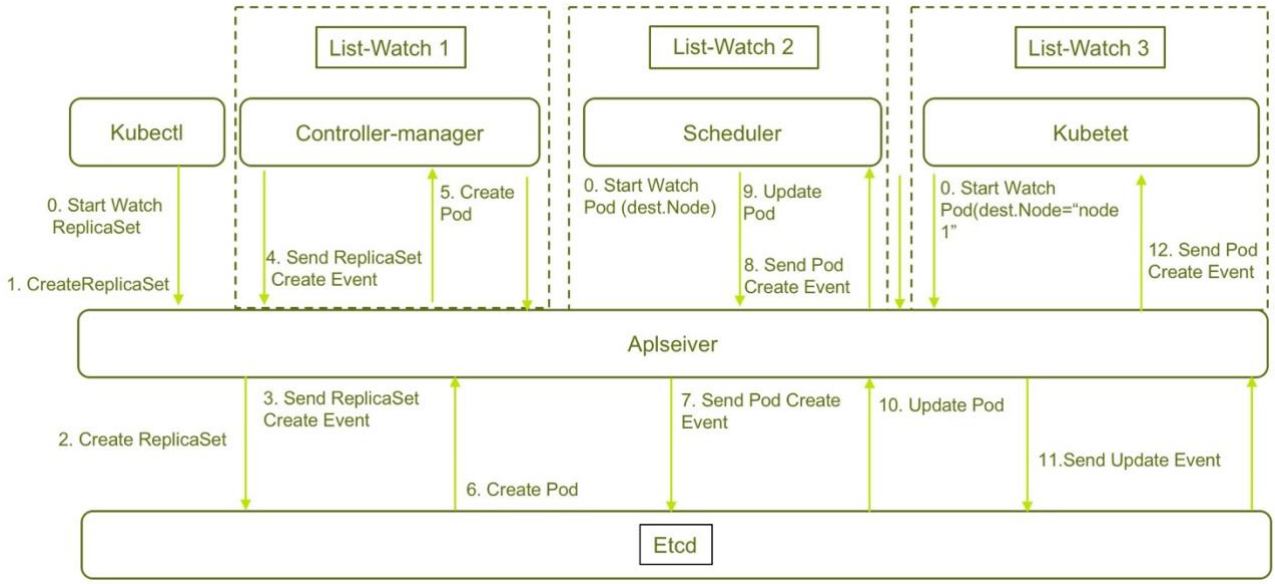


Figure 11 Uranus Controller cross-node scheduling management

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

In general, the cluster management architectures are similar. They all have Controllers, Schedulers, and Agents. Here, List-Watch mechanism is used to realize decoupling of the interactions among components and embody the design concept of Everything talk to API. It is similar to the message bus in SOA, but its difference from the message bus is unified processing of data storage and event notification. While receiving a new event, the cluster may get this new data, including whether the data change is addition or update of an object.

From the above process, it can be seen that each component achieves natural decoupling through List-Watch of different objects (even if it is a same object, the state is not the same). Each component handles different phases of an application life cycle, and ensures the sequence of processing flow among multiple components in an asynchronous distributed system. At the same time, List-Watch only monitors events, so every time when Watch creates a pod for example, it acquires incremental information. The number of interactions acquired is very small. There is a possibility of loss when a message/event is delivered via the message bus. This requires a fault-tolerant mechanism to guarantee the eventual agreement. The engine solves this issue by a List. For example, the ReplicaSet Controller will periodically provide a full-amount List for ReplicaSet. The acquired data is the application data that the current system needs. It will check the current instance running status of each application, do the corresponding processing, delete the redundant and make up the insufficient.

5.3.1.3. Computing Power Container Service Discovery and Deployment

The standard plug-in in the Uranus engine: kube-dns may provide DNS service inside the cluster, and access Uranus engine service through resolution of service names by DNS. The Uranus engine service is made up of a set of PODs, which are containerized applications. These PODs previously used load balancers. If there is a Worker cluster, which has an internal service called Mysql composed of a group of Mysql PODs, other applications in this cluster can access this Mysql service via DNS, and PODs can resolve services across clusters transparently.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

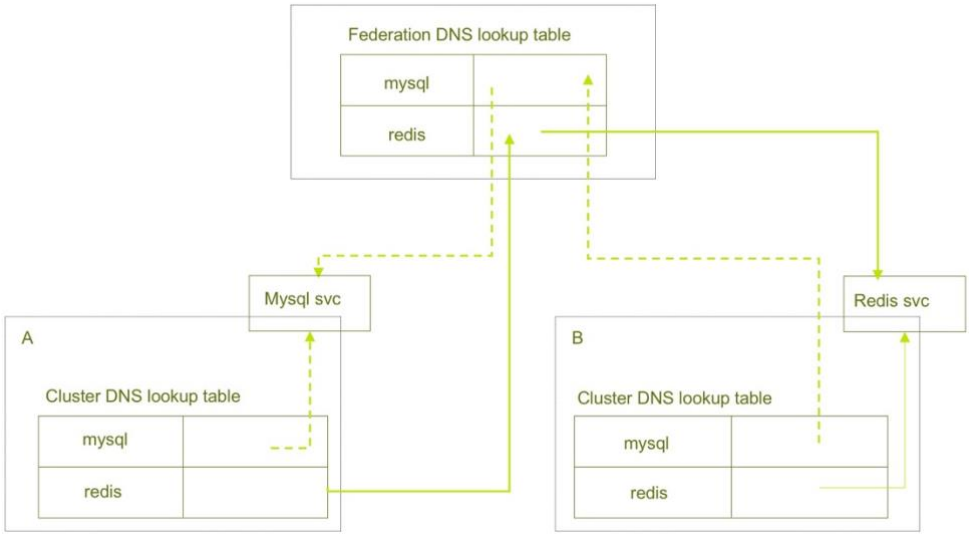


Figure 12 Uranus computing power container discovery and deployment

Deployment example:

Through the automatic scheduling of a Node, the computing power container is allocated to four different Worker machines through HAProxy.

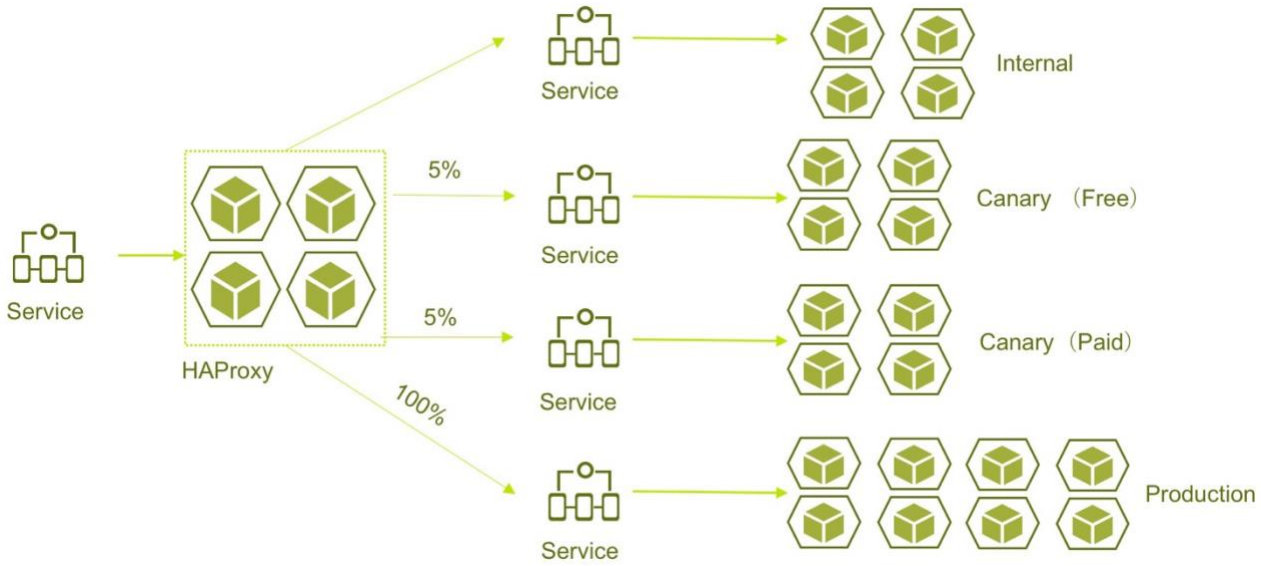


Figure 13 Single-node computing power container scheduling management

Two internal Pods, four free Pods, four paid Pods, and eight production Pods are allocated to each microservice, plus four HAProxy Pods.

A typical service yaml file is as follows:

apiVersion: v1

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

```
kind: Service
metadata:
   name: account-service-production
   labels:
app: account-service- production
       tier: service
       lb: private
spec:
    ports:
    - port: 8080
        name: http
        targetPort: 8080
        protocol: TCP
   seleUranusor:
        app: account-service
        tier: service
        track: production
```

The API gateway in the system will allocate an X-Traffic-Group identifier to each user request header. According to this identifier, HAProxy will route user requests to different deployment environments. When the scale of production is expanded, monitoring sometimes cannot effectively track the status of each deployment, and the excess or inadequacy of the number of the copies of each pod. The system will automatically monitor and manage various deployment status of the operation.

In addition, the automatic health check tool will automatically check whether the HAProxy related to this service is healthy or not and check how many copies each Pod contains. It will also check the four deployments behind the HAProxy (internal, free, paid, production) have at least two valid endpoints. If any error occurs, the system will automatically send an alarm notice via email.

5.3.1.4. Affinity and AntiAffinity in Computing Power Container Scheduling

Due to the characteristics of the distributed peer-to-peer system, the Uranus engine enhances Pod scheduling, involving multiple-scheduler configuration changes, node

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Affinity/AntiAffinity characteristics, PodAffinity/AntiAffinity characteristics, stain and tolerance characteristics, and node problem reporting characteristics.

Affinity and AntiAffinity are runtime scheduling policies, mainly including nodeAffinity (HostAffinity), PodAffinity and PodAntiAffinity.

- NodeAffinity: It is used to specify which node a Pod can be deployed on or which nodes it cannot be deployed on, and solve the problems of Pod and host.
- PodAffinity: It is used to specify with which pods the Pod can be deployed together under a same topological structure.
- PodAntiAffinity: PodAntiAffinity: It is used to specify with which pods the Pod cannot be deployed together under a same topological structure and to solve the relationship between Pod and Pod together with PodAffinity.

There are three rules that can be set when these three types of Affinity are used:

- RequiredSchedulingRequiredExecution: This rule indicates that the corresponding Affinity rule must be satisfied during the first scheduling. If there isn't any node that meets the requirements, no scheduling will be conducted. If the corresponding Affinity rule is no longer satisfied during operation of the Pod, rescheduling will be conducted.
- RequiredSchedulingIgnoredExecution: This rule indicates that the corresponding Affinity rule must be satisfied during the first scheduling. If there isn't any node that meets the requirements, no scheduling will be conducted. In the subsequent operation of the Pod, whether these rules are satisfied or not will not be checked.
- PreferedSchedulingIgnoredExecution: This rule indicates that the corresponding Affinity rule should be satisfied as much as possible during the first scheduling. If there isn't any node that meets the requirements, scheduling will be conducted, too. In the subsequent operation of the Pod, such check will not be conducted.

During operation of computing power container across multiple physical domains, containerization and microservices may encounter the problems of Affinity and AntiAffinity. Originally, multiple components may be installed on one virtual machine and there will be communication between processes. However, during containerized split, the containers are often split directly based on processes. For example, business processes are put in a container, monitoring log processing or local data are put in another container, and they have independent life cycles. In this case, if they are distributed to two farther points in the network, the request will be forwarded for many times and the performance will be poor. Therefore, we hope to realize the nearest deployment through Affinity, and then enhance the network capacity to realize the

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

nearest routing for communication and reduce the loss of the network. AntiAffinity is mainly out of the consideration of high reliability to maximally disperse instances so that when a node is failed, only one instance or one tenth of the instances are affected.

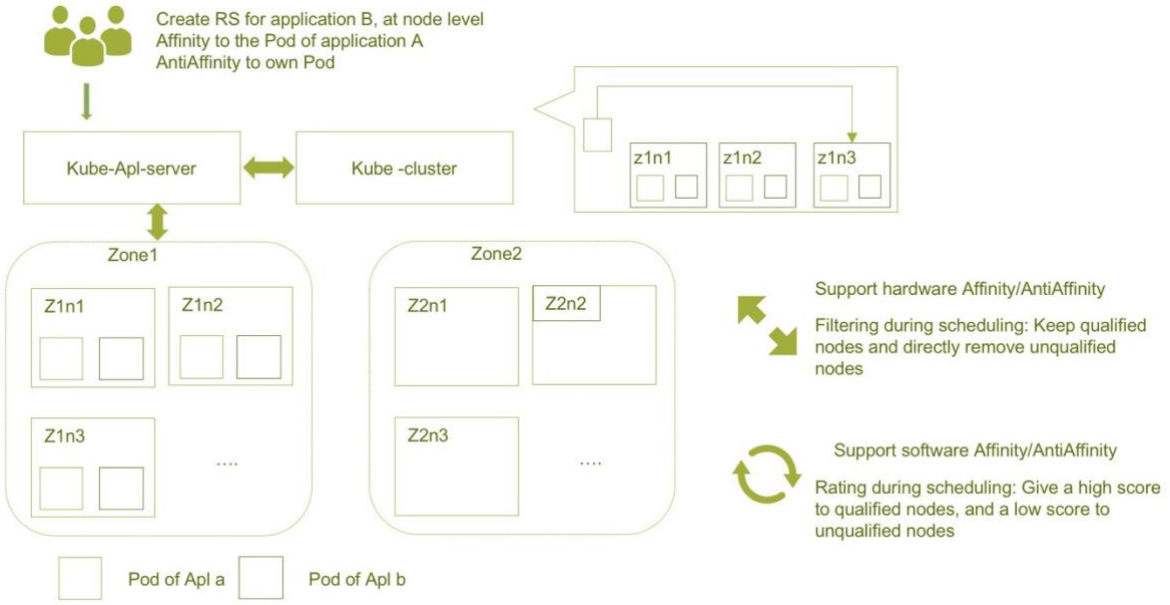


Figure 14 Affinity strategy of uraContainer

In the support of Affinity and AntiAffinity, both rigid and flexible supports (as shown in the figure) will be realized because if they are all rigid conditions, improper configuration may easily lead to failure of application deployment. Here, in essence there are two algorithms, but their realization logics are close. The hard algorithm filters out all the unacceptable nodes to ensure the nodes left at last must meet the conditions. In a flexible case, these harsh requirements are not needed, and only the best effort is required. If the conditions are not met, we still hope the application is successfully scheduled and deployed. Here, we adopt sequencing by score. The ones with a high degree of compliance with Affinity and AntiAffinity will be given a high score, vice versa. Workers are selected in a downward order of scores.

5.3.1.5. Workload Migration and Fault Recovery of Computing Power Containers

We use uraContainer to realize the pooling of distributed computing resources and the standardized measurement of virtual computing resources. In the space of a computing power container, various enterprise-level mainstream applications can be widely supported. When a single computing power container on a single pod triggers the standard health threshold of computing power, Uranus will automatically help it to migrate from the current Pod or the optimal Pod that complies with the Affinity algorithm to the target healthy Pod, ensuring that business applications always can provide efficient and stable service capabilities. At the same time, in a complex distributed network, the failure of physical hardware and interruption of the network

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

make the interruption of any Uranus node unavoidable. However, the built-in self-healing mechanism for workload failures in the Uranus system will automatically help the user to select a new target computing power container from the optimum Uranus computing power pool in the shortest possible time to restore the business and ensure execution of the task. The following figure shows an example for workload migration and fault recovery of the uraContainer:

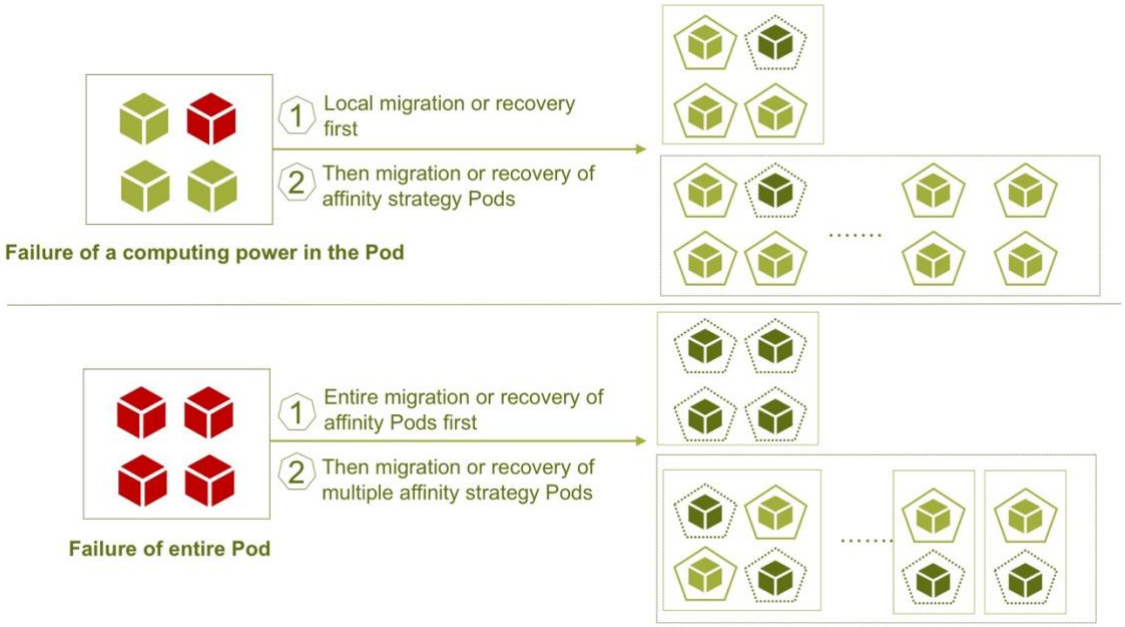


Figure 15 Workload migration and fault recovery of uraContainer

5.3.2 Computing Power Container uraContainer

Although container technology has many advantages, some outstanding basic security problems need to be solved. For example, during operation, the containers can hardly guarantee complete isolation between them. Based on Pouch technology and Kata Containers, we have created a more powerful uraContainer that is suitable for a ubiquitous computing environment.

We know that Pouch is a lightweight container technology. It has features such as high speed, high efficiency, high portability and low resource consumption. It ever helped Ali achieve faster internal business delivery, and at the same time, also raised the utilization rate of physical resources of the data center on a super large scale. In addition, Hyper and Intel Clear Linux are also exploring the combination of VM and Container technologies to enhance isolation between containers while minimizing performance loss. They have jointly launched Kata Containers.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Our enhanced uraContainer is as shown below:

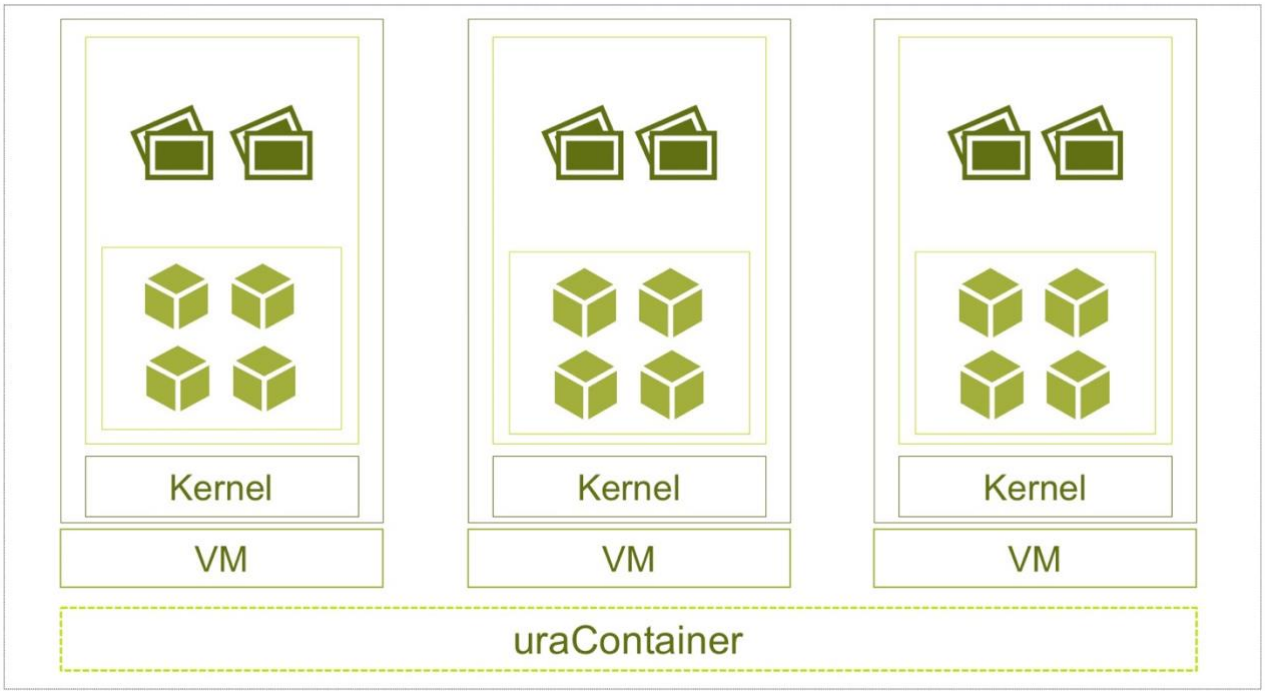


Figure 16 Architecture of the uraContainer system

Technically, the essence of uraContainer is a container running on a hypervisor. Based on the customized kernel, it adds SR-IOV, multi-queue management and other features. uraContainer allows users to achieve both the security of a virtual machine and the high efficiency and manageability of container technology. uraContainer can mask hardware differences, is compatible with the OCI specification and Kubernetes container operation standards and shows strong isolation, lightweight, and fast startup capability while supporting standard image formats.

5.3.3 Worker and Agent Design

In the Uranus architecture, the Uranus engine will contact the Worker host, deploy containers, manage images and schedule tasks. Workers are the hosts really run by Pods, either physical or virtual. In order to manage Pods, each Worker node at least should run uraContainer, agent and proxy services.

In the Uranus engine architecture, Pods must be used to manage containers. Each Pod may contain one or more container. Pod is a set of closely related containers. They share PID, IPC, Network and UTS namespace. They are the basic units for the scheduling of the Uranus engine. Pod is designed to support multiple containers to share network and file systems in a Pod. Services can be accomplished through a simple and efficient way of inter-process communication and file sharing.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

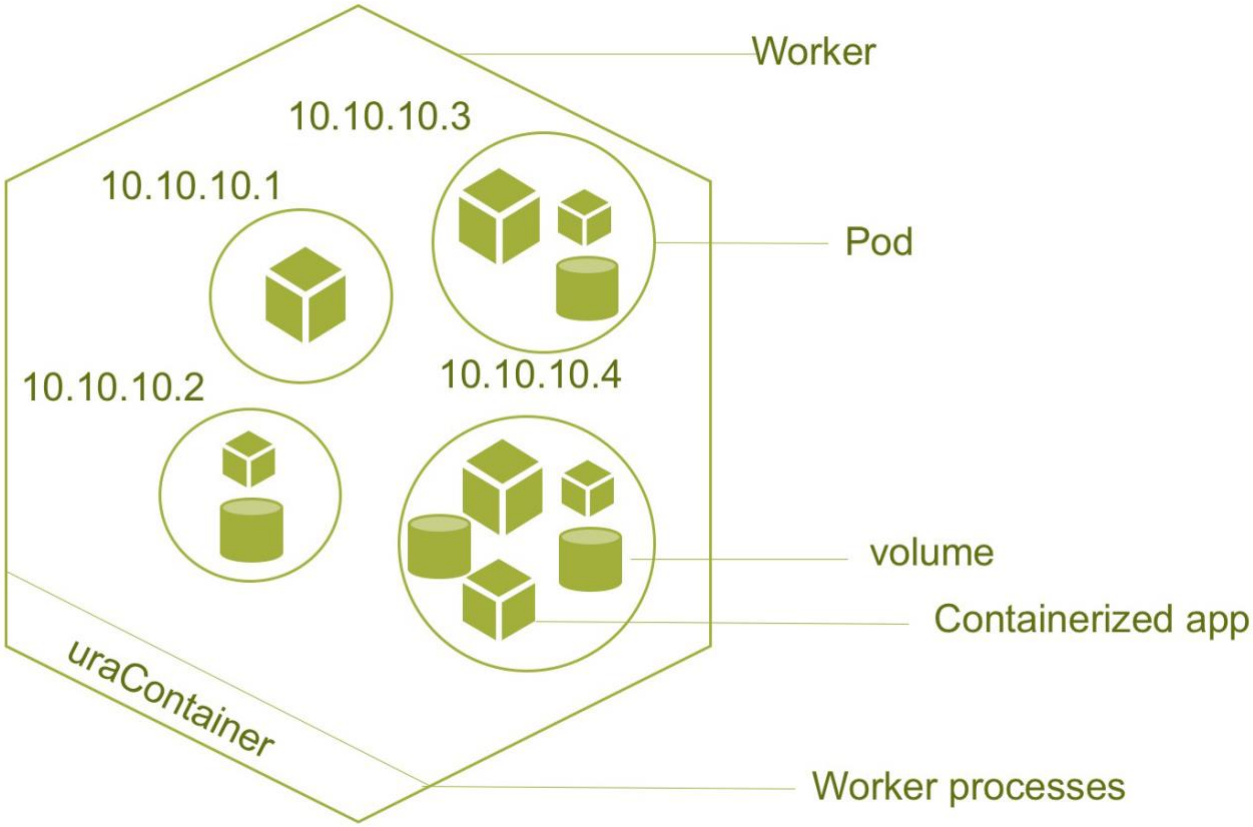


Figure 17 uraContainer Work and Agent architecture

5.3.4 Image and P2P Network Distribution

Although container technology has brought many benefits, the inefficient distribution of container images has posed a great challenge to us. We know that in a distributed computing environment, the cloud native applications of users will grow explosively. Under tens of hundreds of thousands deployment scenarios, the sharing of files and container images will still be unable to meet actual applications. Fortunately, based on the above scenarios, we have developed Beefly, a large-scale distribution tool. Beefly is a universal file distribution system based on CDN and intelligent P2SP technology. At the same time, in combination with IPFS technology, it solves the problems of time-consuming distribution, low success rate and bandwidth waste under large-scale file distribution scenarios, greatly improving the ability to publish and deploy container images.

Beefly combines various innovative technologies such as smart compression and intelligent flow control to solve various document distribution problems under the scenarios of large-scale file downloading and cross-network isolation, and improves data preheating and other service capabilities, as shown in the figure below:

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

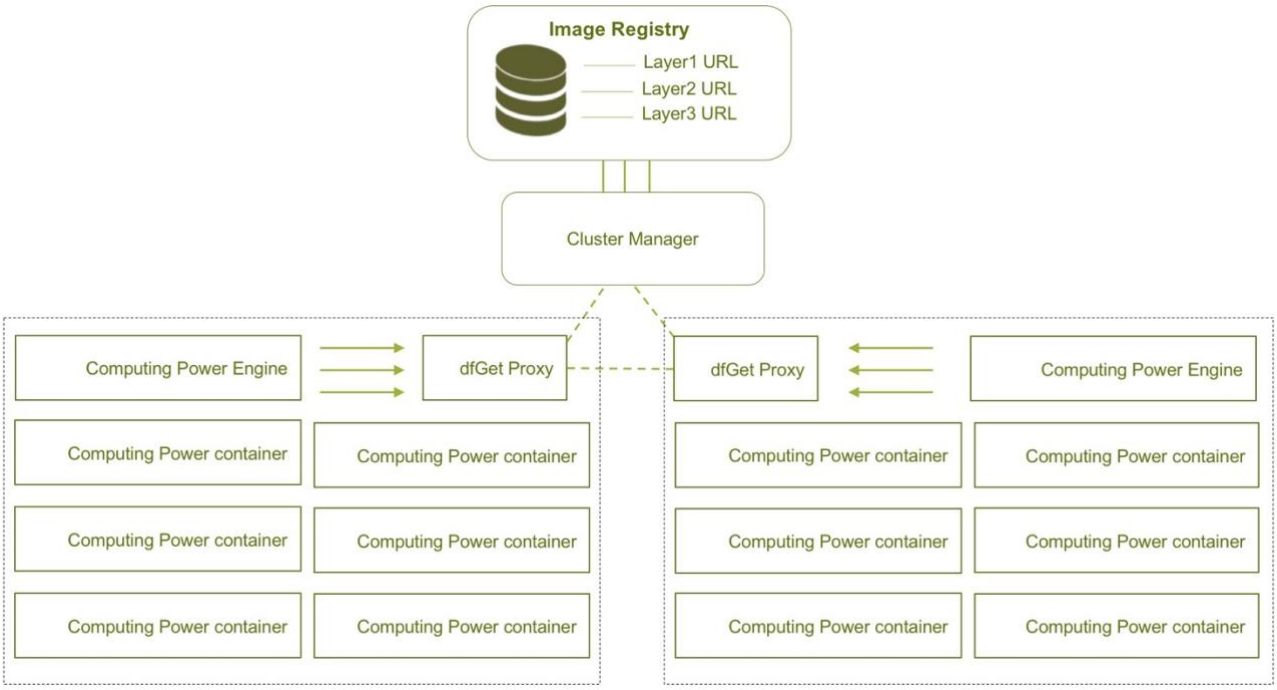


Figure 18 uraContainer image and P2P distribution

At the same time, Beefly has done some work in safe transmission. When sensitive files (such as key files or account data files) are being downloaded, the security of the transmission must be effectively guaranteed. In this regard, Beefly mainly does two jobs: Support the header data carrying HTTP to satisfy the file sources that need to be authenticated through the header; use a symmetric encryption algorithm to encrypt the file content during transmission.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

6 Application Scenarios and Ecology System

6.1 Application Scenarios

The Uranus platform may perform some large projects based on distributed computing, such as the famous Volunteer Project. In addition, because it can automatically complete computing tasks in real time, it is more suitable for small programs, and distributed users with ad-hoc demand at multiple nodes. Once a user accesses the Uranus system, no further commercial contract is required. The user only needs to transact directly with the computing power contributors when a demand is generated. The transaction is completed automatically. The demander only needs to download the application to be run (if it has been downloaded, no need to download it again) and issue a running command. The system will automatically search the nodes that can contribute the corresponding computing power at this moment and run the application after the initial transaction. After the operation is completed, it will return the result and complete settlement automatically. That is in the light of Mark Waiser's spirit of Calm Technology.

The Uranus project empowers the computing services through blockchain so that the resources in it can be used for specific purposes, and executes related tasks. At least the following market scenarios are included:

Distributed implementation of general-purpose computing tasks

- DNA computing
- Computing of planetary orbits
- Analysis of weather data

Projects based on edge computing

- IoT node management and deployment
- Voting platform
- Advertising
- Games
- Data acquisition

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Application Ecology (Block as a Service)

- One-click deployment
- Container image supermarket and App store
-

Distributed implementation of general-purpose computing tasks

American scientists ever organized a project called “Volunteer Computing,” trying to break down large-scale general-purpose computing tasks such as protein structure analysis, planetary orbit computing, and large-scale weather data analysis into a large number of tasks on idle private computing terminals through the distributed computing structure. However, due to the lack of an effective value delivery mechanism, these projects were piled up. The Uranus project based on the public blockchain is a good solution to the value delivery mechanism and truly achieves the distributed implementation of general-purpose computing tasks.

Projects Based on Edge Computing

The conventional cloud computing model can issue computing instructions and give response to results based on centralize operations. This model can satisfy many of the application scenarios, but for other applications that need real-time response between milliseconds, if the existing cloud computing model is used to transmit data to cloud terminals thousands thousands miles away via a lagging and jittering network with an uncontrollable distance and send back the results after operation, obviously it cannot meet the application requirements of real-time computing. If the network, computing, storage, application and data that are close to the devices or data sources are integrated, the nearest edge computing is used and the response result is returned, it will be more suitable for these applications that require real-time computing, as shown in the figure below:

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

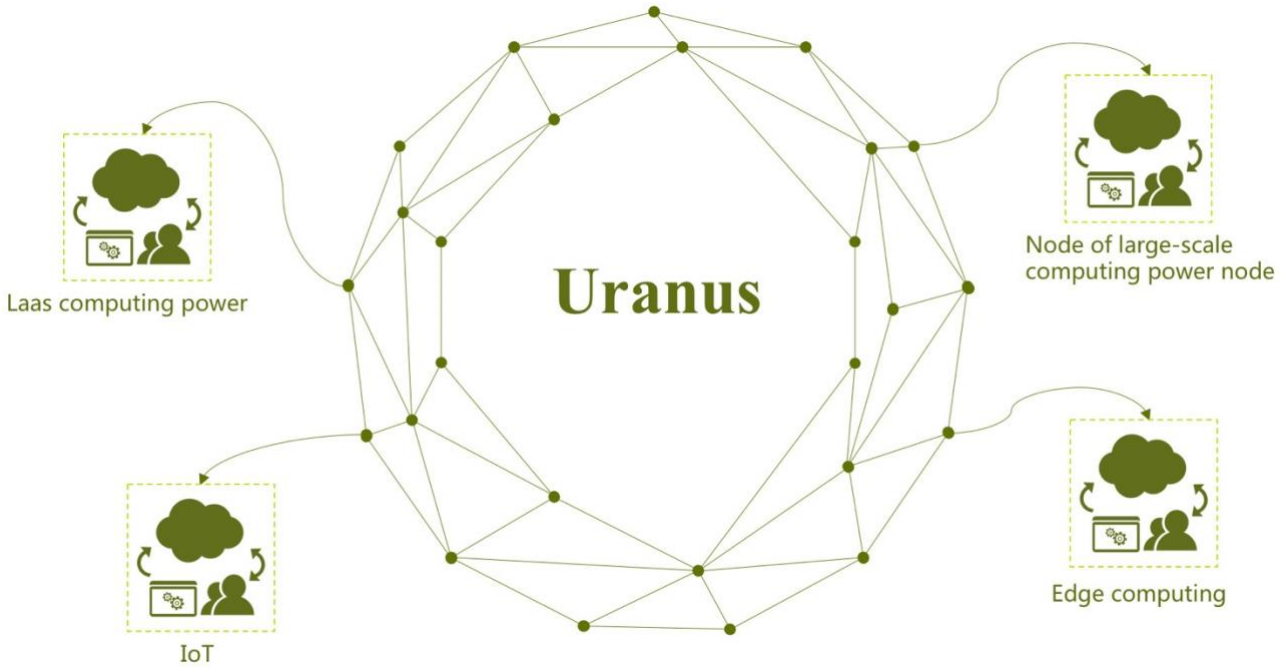


Figure 19 Application scenarios in the category of edge computing

In addition, some computing based on distributed nodes, such as IoT node management, voting platform, games, and advertising, needs to be completed through the architecture of distributed computing and edge computing, nor conventional public cloud based on a centralized strcture.

Application Ecology

The Uranus system actually provides a Block as a Service (Baas). It may provide one-click deployment for various chains and applications. On this distributed container architecture based on pubic blockchain, container image supermarkets and application stores may be provided, thus building an ecosystem that has thousands thousands hundreds of thousands computing contributors and users and is based on public blockchain and distributed computing and in which the application developers and application users could cooperate with each other.

6.2 Business Application Examples

In order to explain how Uranus platform meet the needs of specific commercial applications, here we give a few examples for network data acquisition, voting, games, and IoT terminals.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Network Data Acquisition

The enormous data continuously emerging on the Internet is an important source of business, technologies and technology innovation. On the Internet, data is automatically acquired (grabbed) and processed and widely used to prevent potential security risks, improve user experience and tap business value. The most popular method is to develop an automated program that requests data from web servers (usually HTML forms or other web files are used), then resolves the data and extracts the needed information. The most common problems encountered during network data acquisition are: forms are rejected by the server after submission, and the IP address is blocked by the website and inaccessible. Although we can temporarily solve the problem of network data acquisition through proxy servers or variable IP technologies, how to achieve mass commercialized and sustainable data acquisition from multiple automated network nodes is still a question to be answered. Uranus network links enormous devices from all over the world. Owing to its breadth and comprehensiveness of resources, it will be the best carrier for global ubiquitous network data acquisition.

Voting Platform

With the rapid development of Internet+, online voting has been widely used in various business. In order to ensure the fairness of voting and eliminate paid posters, paid votes and other illegal activities, the voting systems usually adopt: random ID, multiple verification codes, limited IP address segment, limited voting start time and other means to eliminate paid posters and paid votes.

Some teams with strong overall strength, for example with more than 10,000 voting members and own software development technicians will often develop their own voting platforms so that the operators can publish tasks on the platforms and the voters will receive tasks on the platforms. This mode saves a lot of manpower and material resources, but the independent IP and voting terminal of each user required by the voting platforms has become a threshold that cannot be stepped over in the industry.

The Uranus network links a great many computing devices around the world. Every computing node terminal has an independent IP and can be used as a qualified computing terminal node. The Uranus computing node will be an ideal carrier for this application scenario. The platform resources are flexible, economical, safe and pay-as-you-go.

Games

Data shows that the revenue of the global game market reached USD 108.9 billion in 2017, including USD 27.5 billion from China region, and the number of digital game

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

users exceeded 600 million, representing an increase of 17.7% year-on-year.

Behind the rapid growth of the game industry and hundreds of millions of users enjoying the fun of games are hundreds of thousands game server nodes, which have achieved this miracle. The ability to support the operation of computing nodes for large-scale games is the key of the soft strength. In addition, how to rapidly expand, meet the demand for computing power and rationally utilize the computing nodes at off-peak hours will become the key to the profitability of game operators.

IoT Terminals

Market research firm IDC predicts that by 2020, there will be more than 50 billion terminals and devices hooked to the Internet. In the future, more than 50% of the data needs to be analyzed, processed, and stored on the edge of the network.

For example, if you buy a new smart door lock, you need to rely on a home gateway for biometric processing, and a harmful gas detector does not need to send the collected data back to the center, but processes it locally.

The core capability of the public cloud is the computing power based on data centers, but in the era of the Internet of Everything, this capability needs to be seamlessly extended to the edge computing that is closer to terminals, forming center + edge synergistic computing.

6.3 Container Image and Application Ecology System

The Uranus platform aims to establish an expanded public blockchain and optimized container scheduling technology to seamlessly link and integrate redundant computing power in the society and provide efficient, cost-effective and decentralized computing services for numerous computing contributors and users, and build a computing power sharing ecosystem that is owned, managed and shared by the above players, thus reshaping the pattern of the IT computing power market and the production relationship between resource providers and users. Now we take the container image market ecology and the application market ecology as examples.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Figure 20 Composition of the computing power sharing ecosystem

Container Image Ecology

There are numerous resource contributors, resource users and platform application developers on the Uranus platform. With the deployment of numerous computing nodes, the joining of platform contributors and users, and the continuous innovation of application developers, the world's largest container image market with the most comprehensive coverage of industries and technologies will be automatically formed. Based on this ecology, the platform resource contributors may not only contribute their redundant computing power but also upload their users' images to the platform by single click to harvest gains. The users will be freed from the traditional complex and inefficient application environment deployment and configuration and focus on business innovation. The platform developers will also benefit from multiple parties. A benign, sustainable, and automatically evolving container image market will continue to provide efficient support for social operations and reduce overall costs, and create values for the contributors, users, developers, and other parties participating in the platform ecosystem.

Ecology of the Application Market

With the rapid development of blockchain technology and the deepening of the applications of the computing power sharing platform, thousands thousands hundreds of thousands resource contributors and computing users will continue to access and more application developers will be attracted by the market size. More and better platform applications will be uploaded to the platform. The intensive effects of platform ecosystem participants will greatly increase the application value of the platform and the benign development of platform ecosystem, and result in geometric

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

growth of the applications, users and transactions. The platform contributors and users will benefit from an efficient, cost-effective and decentralized ecosystem. The open protocols, clear benefit distribution mechanism, enormous threshold-free accessible users, and extremely low cost of computing and bandwidth resources will attract numerous developers to migrate to this platform.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

7 Development Roadmap

According to the market demand, we have formulated the following implementation roadmap in five phases.

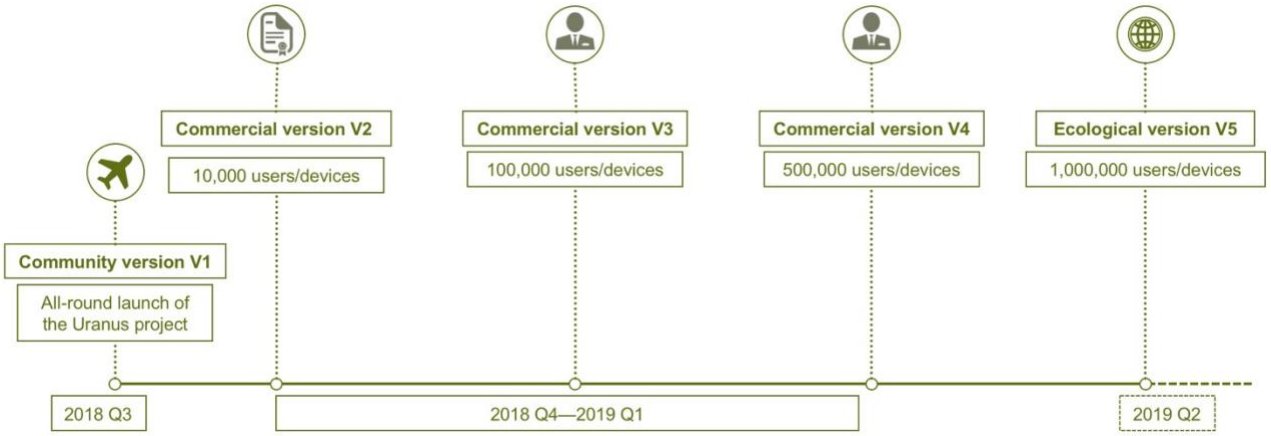


Figure 21 Uranus development roadmap

Phase 1: Community Version (V1)

2018/Q3: Complete closed beta test and release a community version; users may register their own computing devices and use a small number of verified applications.

Phase 2-Phase 4: Commercial Version (V2, V3, V4)

2018/Q4-2019/Q1: Release three commercial versions V2, V3 and V4. In the corresponding development phase, the number of platform users will rapidly increase to the orders of 10000, 100,000 and 500,000. Following the continuous increase of the commercial level of the computing resource pool, the covered industrial and application scenarios will be further enriched and the resource contributors will obtain more revenue. At the same time, with the increase of trading volume, some trading income will be generated.

Phase 5: Ecological Version (V5)

2019/Q2: Based on the scale of the platform contributors and users, the value of the platform is becoming increasingly prominent. In this phase, an ecological strategy will be fully launched to allow more industries and more collaborators to participate in the application creation and contribution, gain benefits, promote the automatic operation and sustainable development of the platform and form a win-win ecosystem.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

8 Team Members

8.1 Executive Team

Chief Executive James Jiang

- Master of Mathematics, Texas State University / Master of Operational Research, Nanjing University of Science and Technology;
- Senior Researcher at Bell Northern Research, USA, owning six US invention patents
- Vice chair of the U.S. National Communications Standards Organization TR45.5.2;
- Established ZTE's subsidiary Shenzhen ZTE Integrated Telecom Ltd./ Shenzhen ZTE Mobile Telecom Co. Ltd and serving as General Manager and Board Chairman;
- Vice President of a subsidiary of VIA Group, and Assistant to the President of the Group;
- Established an enterprise-level desktop cloud company, leading the Chinese market.

Chief Architect Halley Han

- Master of Automation, China Agricultural University;
- Expert at virtualization and cloud computing, Linux kernel engineer and expert at open source technology;
- Expert at distributed virtual computing security delivery and audio and video algorithms, serving clients including the United States Department of Defense, Federal Ministry of Defence (Germany), the Turkish Navy, Citibank, etc.
- Co-founder & CTO of Beijing Aner Kechuang Information Technology Co., Ltd.;
- LINUX LEADER&ARCHITECT of Wyse Technology LLC;
- Chief System Architect of ISoft Infrastructure Software Co., Ltd.;
- Established an enterprise-level desktop cloud company and leading the technical architecture and product development.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

8.2 Technical Advisers

Chief Container Expert, Dr. Liang Sheng

- Master of Computer Science, the University of Science and Technology of China;
- Ph.D. in Computer Science, Yale University;
- CTO of Citrix CloudStack in 2011-2014;
- Founder & CEO of CloudStack, acquired by Citrix in July 2011;
- SEVEN Networks VPE;
- Software Engineering Director of OPENWARE;
- Founder and CTO of Teros Networks;
- Staff Engineer of Sun Microsystems Inc and a main contributor to java virtual machines.

Chief Blockchain Expert Dr. Zou Jun

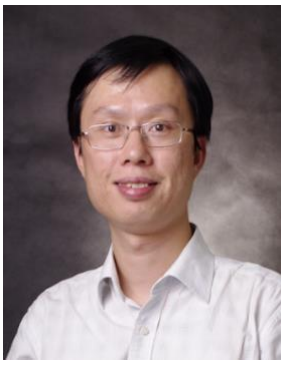
- Doctor of Computer Sciences, Macquarie University, Australia / MBA, Macquarie Business School;
- Chief Architect for Financial Industry, IBM Australia Software Division;
- Published more than 20 papers at leading international conferences and journals, involving cloud computing service contract, blockchain finance and regulatory technology;
- Chief Editor of “Blockchain Guide” and a member of the Editorial Board of “Software Definition Storage”; won an award for the best blockchain paper from ICWS; the consensus algorithm paper written by him as the first author was accepted by “Transaction on Service Computing”, - a top international service computing journal.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Chief Scientist Liren Chen

- Master of Computer Science, Carnegie Mellon University / Master of Computer Science, Tsinghua University;
- Co-founder and CTO of Hubat, researched and developed a search engine for the entire network;
- Worked for Google for seven years, mainly engaged in the R&D of search in Chinese, Japanese and Korean languages, big data and large-scale systems, academic and legal search (Google Analytics, Google+, SRE, etc.);
- The startup company vivisimo was acquired by IBM in 2011;
- The Principal of 360 Mobile Search, and CTO of Pangu Search;
- The startup company yunyun.com was acquired by Sina in 2013;
- The startup company Passion Technology was acquired by Meituan in 2016;
- Co-founder & CTO of an AI and blockchain company.

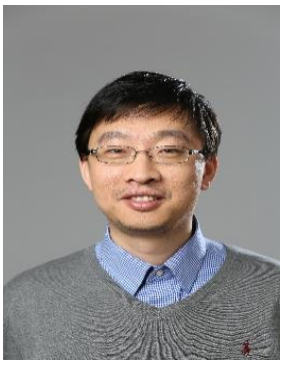
Chief Cloud Computing Advisor Zhang Haining



- Master of Computer Science, Simon Fraser University, Canada;
- Expert at cloud computing and blockchain technology;
- Founder of Harbor enterprise-level container image warehouse;
- Technical expert at open source PaaS cloud platform Cloud Foundry;
- A contributor of super account book projects;
- An author of “Blockchain Guide”;
- Ever worked as a senior architect of IBM Smart Cloud and a senior solution architect of Sun Microsystems.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Chief Large Scale Resource Platform Management Advisor Dr. Yang Haiming



- Master of Electronic Engineering, Tsinghua University;
- Ph.D. in Electrical Engineering, Rensselaer Polytechnic Institute (RPI), United States;
- Worked for Cisco for 5 years, responsible for the R&D and O&M of the unified resource management platform for Cisco Global Data Center;
- Chief Architect in Cloud Computing and OpenStack Technology, IBM Greater China, responsible for the R&D of IBM's OpenStack technology in China, IBM cloud platform IaaS, and the implementation of end-to-end technology integrated with business;
- Chief Architect of CTO System of Jingdong Group, responsible for the technical strategies of Jingdong Group, R&D of blockchain underlying technology, and technology integration.

Chief Financial Technology Advisor Shao Zhou

- Certified software architect and information security and risk management expert of TOGAF;
- A pioneer of IoT technology;
- An explorer of blockchain technology and its commercial application;
- An active organizer and contributor of open source technology communities;
- Chief Financial Technology Architect of AIIB Headquarters;
- Ever served many of Fortune 500 companies, have more than 10 years' experience in the international financial technology industry and rich experience in building and managing efficient development O&M teams, involving investment, asset management and insurance sectors, and ever led and developed several large trans-regional cross-business line complex projects.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

8.3 Product Development

CPO Li Fei

- Founder of China Virtualization Technology Forum;
- Chief Editor of "China Desktop Cloud Standardization White Paper";
- Technical Manager of Realor Information & Technology Co., Ltd. /Beijing Gintel Technology Co., Ltd. in North China;
- Virtual Product Line Product Manager of ISoft Infrastructure Software Co., Ltd.;
- CPO of an enterprise-level cloud computing/desktop cloud company.

Software Architecture and Algorithm Expert Li Mingbo

- Master of Electronic Information Engineering, Communication University of China;
- Manager for Application Virtualization Software Architecture, ISoft Infrastructure Software Co., Ltd.;
- Manager for development of remote display protocol algorithms for 10 years;
- Expert at remote protocol peripheral redirectioning.

P2P Transmission Technology Expert Xing Zhen

- Bachelor of Computer Science, Jilin University;
- P2P transmission technology product manager for 8 years;
- Product manager of enterprise-level cloud computing/distributed computing desktop cloud platform;
- Product manager of enterprise-level cloud computing/file P2P sharing and backup

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Software Architecture and System Architecture Expert Wang Shishuang

- Bachelor of Computer Science, Beijing Jiaotong University;
- Dell-wyse engineer, responsible for linux OS architecture and device driver;
- Senior development manager of linux system architecture and virtualization algorithms for 15 years.

Blockchain Expert Chen Yaowen

- Expert at machine learning;
- Focus on industrial scenarios for deep learning of NLP;
- Focus on blockchain+AI;
- Deep learning of Chinese participle project kcws (github 1700 star).

Senior Expert at Blockchain Zhang Cheng

- Master of the Institute of Computing Technology, Chinese Academy of Sciences, 10 years' experience in engineering architecture;
- Expert at large-scale search engine technology;
- Expert at big data;
- Senior expert at blockchain, intensively studying mainstream public blockchains and the economic and ecological planning of blockchains.

Cryptologist Dr. Tang Yi

- Dr. & Professor of Sun Yat-Sen University, Guangzhou;
- Presided over/participated in the completion of a number of national, provincial and ministerial level scientific research projects and won the second prize for progress in cryptologic technology;
- Published multiple cryptology-related papers;
- Visited Hong Kong Baptist University & Department of Computer Science, University of North Carolina, USA on behalf of Sun Yat-sen University.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Distributed System/Grid Computing Expert Chen Chao

- Master of Computer Science, Jiangnan University;
- 15 years' experience in enterprise-level system architecture consulting, design and development, an Augmentum system architect, leading the design and deployment of Motorola's American 911 system, and the integration of gas dispatching SOA system of China Resources;
- Senior R&D Manager of Cisco China, completing the development and definition of Edge300 series, Fanuc ZeroDownTime project, and the development and definition of Lora IoT gateway;
- Co-founder of Yilujia Network Technology Co., Ltd;
- Youaqu, E+ and Jinwangyigou system architecture and team management;
- Founder of iweihu smart platform.

Distributed System/Grid Computing Expert Yue Yongqiang

- Master of Mathematics, Dalian University of Technology;
- Manager for the R&D of Windows system architecture and applications for 10 years;
- Six years' experience in the architecture and development of cloud computing application platform;
- Eight years' experience in the management and development of ubiquitous terminal devices, and the management of more than 10,000 concurrent terminals;
- App R&D Manager of Beijing Founder Electronics Co., Ltd.

Distributed System/Grid Computing Expert Ma Xiliang

- Bachelor of Computer Science, Hebei University of Engineering;
- Manager for the R&D of Linux cloud computing platform system architecture for eight years;
- Six year's experience in front-end development of cloud computing platforms;
- The developed products have been successfully applied in more than 1000 enterprise-level users.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Expert at Distributed System/ Blockchain Expert Andrew Dong

- A development engineer of Amazon, responsible for rear-end authentication and authorization service of Amazon.cn.
- A core development staff of IBM Labs WebSphere Commerce, which was released in WebSphere Commerce 7.0;
- A researcher of distributed consensus system, blockchain and distributed system of the University of Sydney, and a member of a distributed system research team ranking top in Australia; published patent applications relating to blockchain consensus protocol based on GPU common resources on IEEE Cloudcom;
- Chief Blockchain Architect of Fujitus Australia, successfully developed and delivered multiple banking and financial projects.

8.4 Advisory Board

Advisor Michael Meng

- Senior Vice President of Xinyuan Technology Group and Partner & CMO of RChaintech;
- A senior in the chain and currency circles and an early investor in the currency circles;
- Former Founder & CEO of Turingcat;
- 15 years' work experience in IBM, as the general manager of a branch company and a general manager for smart city.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Advisor Wang Zishang

- Author of "Cloud Management 2.0";
- Council member of China Artificial Intelligence Industry Alliance (AIIA);
- Founder & CEO of Topfun Media (835872), which has hosted TOP50 Golden Apple Award and TFC Global Mobile Games & Intelligent Entertainment Exhibition for 13 years in a row. Participated in multiple famous blockchain projects, such as TOPC, CSDN and POT;
- "Top 10 Women Entrepreneurs of Science and Technology" of China Women's Development Foundation's and "Her Entrepreneurship Mentor", 2017 China Excellence Award for Scientific Management.

Advisor Song Xiaonan

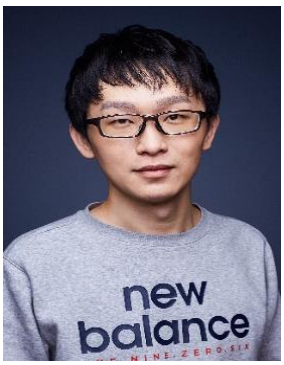
- Expert at Unix and distributed system and grid computing;
- 10 years' experience in the R&D and marketing of cloud computing, having in-depth study on operating systems, middlewares and cloud platform and practice in large projects;
- Worked in the cloud computing department of Huawei, responsible for product planning, business insight, and market promotion and technical cooperation in EMEA region;
- Chief cloud computing architect of Primeton.

Advisor Zhao Peng

- Hyper CEO;
- A renowned expert in the cloud computing open source field;
- Hyperledger angel investor;
- Ever served JP Morgan Chase, Quantum Fund, HSBC, Lehman Brothers, MUFJ and many other international financial institutions;
- Have rich background and experience in the combination of blockchain and financial technology.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Advisor Lv Xinhao



- Co-founder of iotchain;
- Initiated a blockchain team in Xiaomi at the end of 2016 to solve the problem of data island;
- Participated in the translation of mastering bitcoin in 2014, which is a very good technical document about blockchain in Chinese.

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Copyright Notice

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved. Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Disclaimer

Uranus Foundation reserves the right to interpret and modify this document. Any amendments, updates or interpretations of this document will be announced on the website of Uranus Foundation (http://www.uranus.io).

Contact Way

Uranus Foundation LTD.

Website: www.uranus.io

Email: contact@uranus.io

LEGAL DISCLAMIMER: The information and any attachments contained in this document is all copyright and intellectual property of Uranus Foundation reserved.
Without prior written permission of any copy, reproduce, propagation or storage will be prohibited.

Uranus