# Hamster White Paper

Update: 2021-10-19

Version: 0.0.2

Author：XXX

目录

[Hamster White Paper 1](#_Toc97822735)

[1 Background of the project 3](#_Toc97822736)

[1.1 Vision Objectives 3](#_Toc97822737)

[2 System Roles 5](#_Toc97822738)

[2.1 Node Roles 5](#_Toc97822739)

[2.2 User Roles 6](#_Toc97822740)

[3 System use cases 8](#_Toc97822741)

[3.1 Nodal tasks 8](#_Toc97822742)

[3.2 User interaction 9](#_Toc97822743)

[3.3 Smart Contracts 9](#_Toc97822744)

[4 System design 10](#_Toc97822745)

[4.1 Normal processing subsystem 11](#_Toc97822746)

[4.2 Exception handling subsystem 11](#_Toc97822747)

[4.3 Verifiable random numbers 13](#_Toc97822748)

[4.5 DDNS 13](#_Toc97822749)

[4.6 Cross-chain interoperability 13](#_Toc97822750)

[4.7 Computing network nodes 14](#_Toc97822751)

[5 Implementation programme 16](#_Toc97822752)

[5.1 Network Upgrade 16](#_Toc97822753)

[5.2 Governance in the chain 17](#_Toc97822754)

[5.3 Consensus algorithms 17](#_Toc97822755)

[5.4 Supporting facilities 17](#_Toc97822756)

[6 Economic models 18](#_Toc97822757)

[6.1 Token Allocation Ratio 18](#_Toc97822758)

[6.2 Token value basis 19](#_Toc97822759)

[6.3 Anti-predigging mechanisms 19](#_Toc97822760)

[6.3 Staking and revenue rules 19](#_Toc97822761)

[Details to be worked out when the token allocation ratio has been determined (to be added) 20](#_Toc97822762)

[6.3 A developmental approach to incentive design 20](#_Toc97822763)

[7 Product Roadmap 20](#_Toc97822764)

[8 References 21](#_Toc97822765)

## 1 Background of the project

With the rapid development of blockchain technology, well-known public chains such as Bitcoin and Ethereum continue to upgrade and their ecosystems and communities continue to grow. new concepts and new things such as AMM, DEX, DeFi, NFT, and emerging public chain design solutions (Polkadot, Solana...), metaverse, and other new concepts and things are emerging, and the whole crypto world is gradually connecting its major segments together.

From the perspective of the development history of blockchain, when the infrastructure setting is built to a certain stage, it will start to enter the trial stage of application, and when the infrastructure starts to constrain the development of application, it will start the construction of infrastructure again. At present, our blockchain network is deployed and operated among major machine rooms, so that the trend of blockchain services will tend to be more and more centralised. It can be envisaged that when all the machine rooms hold the nodes of the blockchain, it is equivalent to a giant monopoly in the blockchain world. Can we make the arithmetic power of the nodes secure, efficient, decentralised and able to really implement a shared computing service network on the ground.

### 1.1 Vision Objectives

Hamster is a shared services public chain network designed specifically to meet these objectives, and is a blockchain infrastructure capable of accommodating a large number of computing devices. Any computing device can easily access the Hamster network, with minimal learning curve and maintenance costs. The low barrier to entry and the high degree of decentralisation are very much in line with the spirit of the blockchain world.

Hamster aims to build an ecosystem on top of shared services, where service providers no longer need to rely on the support of a single cloud vendor, but only need to use the computing resources in the Hamster network, and the system will automatically schedule the resources needed to run the service and build their own services in the form of edge computing.



Top view of Hamster

## 2 System Roles

In order to provide a clearer discussion of the design and operating principles of the various aspects of the Hamster network, it is possible to look at and define the Hamster network in terms of its physical composition and participants. The entire network is physically composed of a large number of [network nodes] and [external users].



Illustration: Node + User Schematic (to be designed)

### 2.1 Node Roles

The network nodes can be logically divided into mainly：

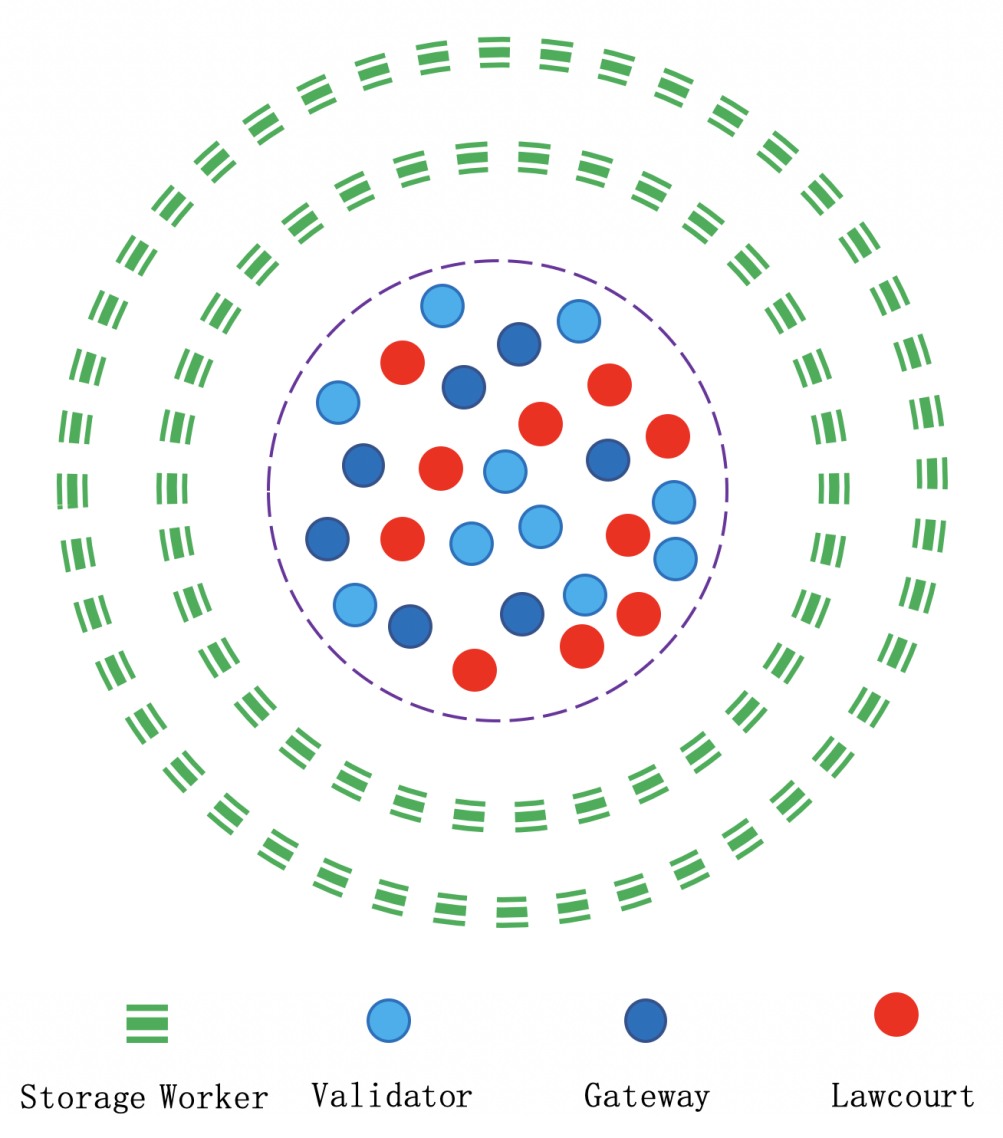
【Validator】,The bookkeeping node, the block producer, is required to generate and maintain the ledger data.

【Candidate】，Candidates for Validator, who may be elected as Validator in each election round.

【Gateway】，Communication gateway, must have a public IP address

【Provider】，Shared service nodes, which need to have ordinary computing and storage capacity and relatively stable access to the network.

【Arbitrator】，Arbitration nodes, which are required to adjudicate on malicious falsification of shared service nodes by Providers, chronic dropouts, etc.



Validator

Gateway

Arbitrator

Provider

Schematic diagram of the various types of nodes (to be redesigned, see Polkadot)

### 2.2 User Roles

External users can be logically divided mainly into.

【Holder】，Hamster-Token（HST）Holders，

【Nominator】，Pledges the Holder of the HST Election Validator and Candidate.【Customer】，Earn revenue by joining different blockchain service networks through the arithmetic power on Hamster.

【Governor】，the on-chain governors elected by all Holders.

【Monitor】，Monitor, a random sample of nodes to monitor and check the operational status of nodes

## 3 System use cases

Since the Hamster network consists of nodes and users, it can be viewed from the perspective of the system use cases: what the nodes need to do, what services they provide, what the users can do, what they need to do, and how they interact with the Hamster network.

### 3.1 Nodal tasks

|  |  |  |  |
| --- | --- | --- | --- |
| **Node type** | **Order of magnitude** | **Features** | **Elected route** |
| **Validator** | 100 to 1,000 | Building a blockchain ledger  Monitoring Gateway (long dropouts)  Executing smart contracts | Pledge + Election (NPoS) |
| **Gateway** | 1,000 to 10,000 | Temporary storage of user uploaded files  Generate file correction code slice  Communication between Relay and Storage  Monitoring Storage (long dropouts)  Buffer for fast fetch of file slices | Pledge |
| **Compute** | 10,000 to 1 million | Generate available computing resource services  Maintain heartbeat links with 3 to 5 Gateway | Pledge |
| **Arbitrator** | 100 to 1,000 | Ruling on Fisherman's probe results | Pledge + Election (NPoS) or by Candidate on a part-time basis |
| **Monitor** | 1,000 - Unlimited | Randomly probing the availability of file splits  Running a raffle game to serve raffle users  Mobile Sweepstakes Mining | Pledge |

Matching table: Nodal tasks (to be adjusted)

The Hamster network adopts a hybrid consensus algorithm of NPoS+BABE+GRANDPA. The Validator is elected by the NPoS algorithm and is responsible for receiving and packaging transactions sent by users, generating and broadcasting blocks to the entire network according to the BABE algorithm, and executing the GRANDPA block termination algorithm, which allows all nodes in the entire network to agree on the state of the blockchain ledger The NPoS algorithm is used to generate and broadcast blocks to the entire network. In the NPoS algorithm, a new round of elections is held at regular intervals (e.g. 24 hours), and Candidate nodes that are not currently Validators can wait for the next round of elections. While waiting, the Candidate does not participate in block generation, but if the Candidate is also the Arbitrator, he/she has to adjudicate the random sampling results sent by the Monitor.

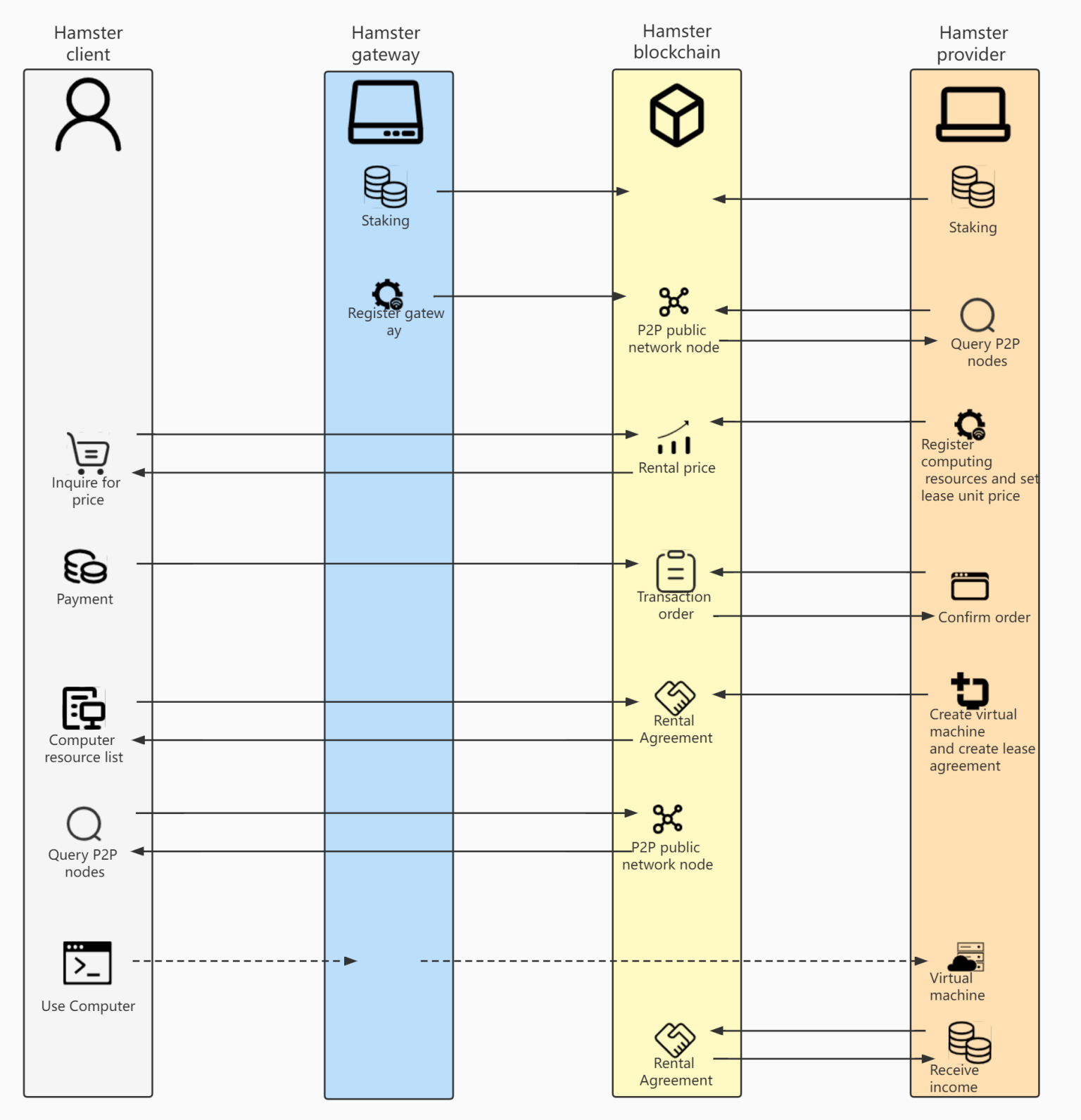
### 3.2 User interaction

The Hamster network uses an NPoS node election algorithm, where Holders can vote for Candidate nodes to help them become Validators, thus becoming Nominators and sharing in the block production rewards that Validator nodes receive.

### 3.3 Smart Contracts

Compatible with EVM virtual machines, providing excellent support for seamless migration of Ethereum's existing DAPP projects.

## 4 System design

The Hamster Network is a public chain system whose core function is to provide a distributed shared computing service to the outside world under an economic incentive model. The overall design is divided into a normal processing flow and an exception processing flow.

Top-level interaction timing diagram (to be optimised)

### 4.1 Normal processing subsystem

Most blockchain network service providers have the following costs: physical machine resources, pledge costs for service node applications, O&M deployment costs for blockchain services, and operating costs for blockchain services. Hamster is helping users with physical machine resources and O&M deployment costs for blockchain services. It is a new option for users to join other blockchain web services.

The user downloads the client and joins the Compute Services network, which is a p2p protocol layer network layer where all compute resources shared in Hamster are shared and visible. All shared computing resources in Hamster are visible on the client. The user can also choose to deploy the service on the client, after selecting the service and selecting the shared computing resource node, click on deploy, then the computing resource party will download the service plugin and deploy the node service with one click, after deploying the node service the user will need to configure the application on the client, a successful configuration will successfully deploy the blockchain service. We can see the running status of the service through the client.

### 4.2 Exception handling subsystem

As with any man-made system, there may be anomalous events that occur in the Hamster network that need to be dealt with efficiently and in a timely manner. Therefore the anomaly handling function includes (but is not limited to) subsystems such as node drop monitoring, network link detection, arbitration voting bodies, sampling detection, etc. to complete the security of the system.

There are several anomalies in the computing network platform.：

1. resource provider network fluctuations
2. gateway provider network fluctuations
3. resource provider node drops
4. Gateway provider node drops
5. Resource provider node forgery

Conducting program monitoring nodes:

1. online proof of resource nodes
2. online proof of gateway nodes
3. gateway node verification resource node online proof
4. random query node specification proof
5. arbitration group design
6. malicious penalty design

With regard to network fluctuations, we associate multiple gateway providers with one resource node and multiple resource nodes with gateway providers, enabling multiple lines for linking and network communication to minimise the impact of network fluctuations.

With regard to abnormal dropouts, we consider a dropout from a machine that has joined the resource node to be a malicious means. The system incorporates a heartbeat monitoring system to detect abnormal dropouts. The resource provider is required to submit proof that the node is online at certain intervals, including information about the resource and the linked gateway node. The gateway node provider is also required to submit its own node online certificate at certain intervals. It also checks the linked resource nodes and reports the proof of the resource node's network link information so that the system will calculate the health of the node based on the current reported proof.

Of course malicious forgery attacks are still possible for node proofs, and we design random node specification proofs and arbitration groups to avoid forgery attacks. When a random node extracts some nodes of the computational resource network by verifiable random numbers, anyone can check the resource specification through the network link. If the view does not match the correct information, it is considered a malicious forgery attack, and the request is escalated to the arbitration group for arbitration. The arbitration team is elected by Candidate to determine whether the node is a malicious node by way of arbitration. The arbitration team is elected by Candidate to determine whether the node is a malicious node and to enforce the appropriate penalties.

### 4.3 Verifiable random numbers

Take the on-chain verifiable random numbers provided by the BABE module that comes with the Substrate framework and update them once per Epoch. The random number can be proofed by randomly claiming the specification of the query node.

### 4.5 DDNS

In a Hamster network, a user or system selecting a node for deployment needs to link this node, and a public IP is required in the Internet to directly find and associate nodes. But here we have a network between the node and the user's client system through the direct connection feature of p2p and some gateways with public IPs. It is equivalent to building another layer of network on top of the current Internet era, which is essentially a decentralized DNS system (DDNS). Building DDNS on the blockchain is a future trend. When a user visits a website or a machine again, he can install a plug-in in his browser and use it to look up the IP address corresponding to xxx.com on the chain, so as to find the corresponding website or host. A well-designed DDNS will not only serve Hamster itself, but also provide a uniform, common naming service for other applications in the outside world.

### 4.6 Cross-chain interoperability

Embedded cross-chain infrastructure with native support for interoperability with the Ethereum and Polkadot ecosystems; changes to the Validator-Set of the Hamster network are reflected in the block header, making it very useful for implementing block header and chain state finality validation and determination on other public chains (Ethereum, Polkadot). It provides a good underlying language for cross-chain interoperability.

### 4.7 Computing network nodes

In order to participate in the Hamster Network, a Compute Node needs to be running the Hamster Node Software. In the Hamster Network, a Compute Node is not just a normal node, but a Compute Node, and the Hamster Node software listens to everything that happens on the network and can respond to all types of events that occur on the network, including pledges, orders and other events.

The Compute Node automatically executes the Hamster smart contract intermittently to.

* report on the health status of the server
* Report on the fulfilment of contract orders
* automatically collect order proceeds：

In addition, the compute node has a full CLI and UI page for operations, adjusting parent gateways, lease rates, etc.

The Hamster network is a wireless, horizontally scalable, P2P peer-to-peer network with freely accessible nodes. It has the following characteristics:

* Distributed Private Networks

The Hamster network is a P2P distributed private network where all computing nodes can establish direct connections through an upper layer gateway with no third party involvement. This effectively avoids surveillance and establishes a fast, secure and private network link.

* Decentralised gateway

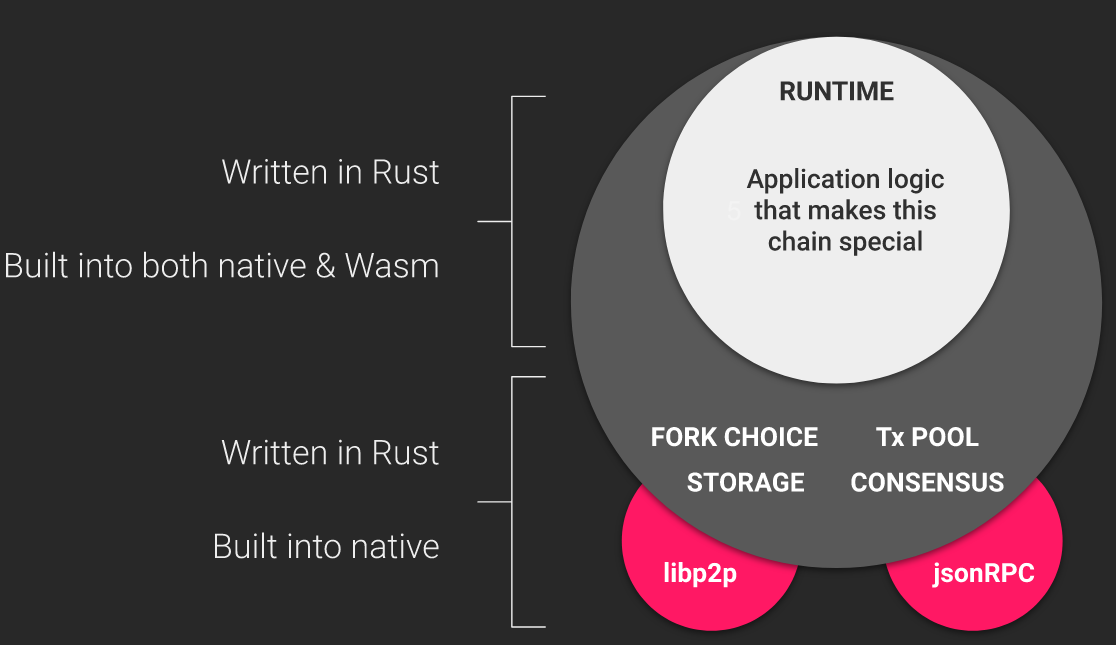
Hamster does not force computing nodes to use a fixed gateway, network nodes are free to choose or configure trusted gateways. Individuals or organisations are free to build Hamster gateways and choose gateways that they trust to act as 'middlemen' for their links.

* Encrypted transmission

The Hamster network uses symmetric encryption between the members of the computing nodes, and only the communicating nodes have their decryption keys between them, making communication between nodes truly secure.

## 5 Implementation programme

The implementation and programming of the Hamster Network is based on Substrate, a highly visible and widely used general blockchain development framework. The Substrate framework consists of four core modules - Consensus, Node, Runtime, and Pallets - that enable the rapid construction of a fully functional and secure public chain. There are a number of advantages and benefits to adopting the Substrate framework, which are discussed below.



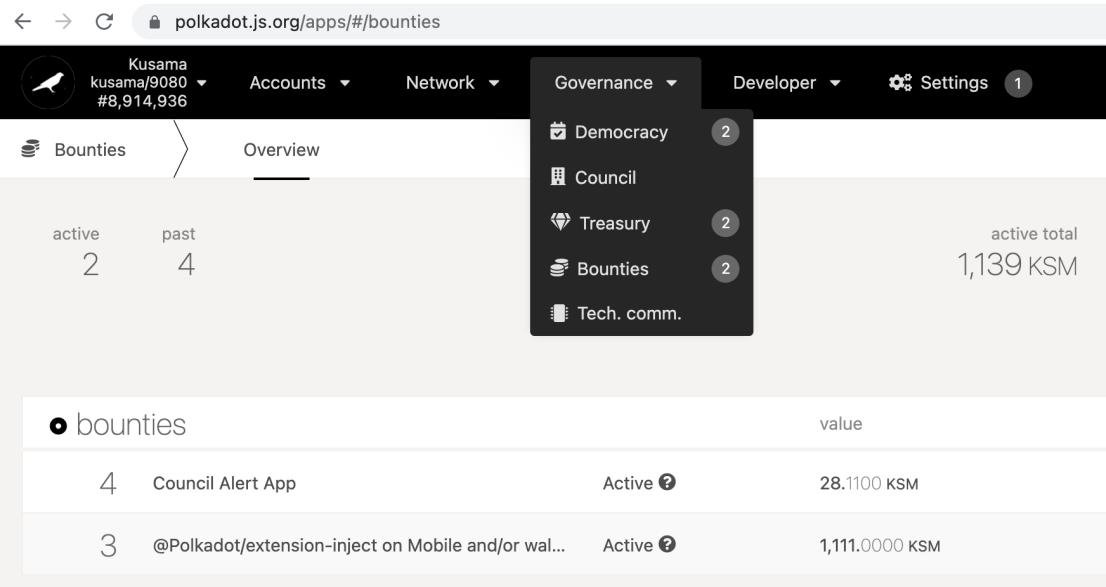
Substrate Schematic diagram of the framework architecture（Needs redrawing）

### 5.1 Network Upgrade

On-chain upgrade, network-wide forkless upgrade via WASM bytecode

### 5.2 Governance in the chain

Flexible governance mechanisms: Democracy, Referendum, Council, Treasury, Technical Committee and other comprehensive decentralised democratic governance mechanisms.



(need to redraw, modify URL, WSURL)

### 5.3 Consensus algorithms

Substrate integrates NPoS+Grandpa+BABE hybrid consensus for security and efficiency.

### 5.4 Supporting facilities

The ecology is vast and full of supporting facilities, such as wallets, DAPP-UI, blockchain browser, chain interaction services, etc.

## 6 Economic models

To better maintain the ongoing security of the Hamster Network itself, the Hamster Network has discovered a base token (Hamster -Token) and has crafted an accompanying economic model.

### 6.1 Token Allocation Ratio



Map: Hamster distribution pie or table Schematic

Validator Node Block Production (Mining) Bonus xx%

(Computer, Monitor, Gateway, Arbitrator pledges can double as Nominator and share the mining rewards at a discount)

Gateway Node Channel Maintenance Incentive xx%

Monitor Random Sampling Awards xx%

Arbitrator Arbitration Awards xx%

Eco-building (e.g. user cross-chain transfers, DApp development) rewards xx%

Foundation set-aside xx%

Market operations set-aside xx%

Development Team Awards xx%

Investment Institutions xx%

### 6.2 Token value basis

The value base of the Hamster Network, the basic services that the Hamster Network can provide to the outside world, including (but not limited to) data storage services, asset transfers, smart contract deployments and operations. Users can hold and spend HamsterTokens to access the various services offered by the Hamster Network, and users pledge HamsterTokens to participate in the maintenance of the network and receive corresponding revenue.

### 6.3 Anti-predigging mechanisms

As the number of nodes may be low in the early stages of the Hamster network launch, HamsterToken pledges are not sufficient and an anti-pre-mining mechanism has been designed to prevent a small number of nodes from receiving too many HamsterToken incentives. The full HamsterToken incentive and release rule will only be activated when the pledged volume reaches xx (TBD).

It is important to note that the anti-pre-mining mechanism is designed to prevent a small number of nodes from gaining too much HamsterToken incentive, but it does not inhibit the normal revenue of individual nodes. The earlier a node joins, the more HamsterToken incentives it will receive, and this incentive rule remains unchanged.

### 6.3 Staking and revenue rules

### Details to be worked out when the token allocation ratio has been determined (to be added)

### 6.3 A developmental approach to incentive design

## 7 Product Roadmap

* Q3 2021
  + Team formation
  + Project Formation
  + Concept development
  + Market Research
* Q4 2021
  + White Paper Development
  + Shared Arithmetic Model
  + Hamster Mainchain Module Providers
  + Hamster Masterchain Module Resource
* Q1 2022
  + pitch deck
  + Gateway module joins shared network integration
  + Hamster Calculus Provider Client
  + Hamster Arithmetic User Client
  + Hamster Master Chain Module Gateway
  + Founding Team
* Q2 2022
  + Awarded web3 project funding
  + Official website online
  + Penalty mechanism
  + pokt project toolkit goes live
* Q3 2022
  + Private Sale
  + DAO
  + Cross-chain protocols
  + Communitoy Build
* Q4 2022
  + Test Network Online
  + Algorithm Search Enhancements
  + IDO & Listing
* Q1~Q2 2023
  + Main Online
  + Follow-up ecological construction
  + Integrating more outstanding projects

## 8 References

[1] Gavin Wood, Polkadot, Substrate and Ethereum

https://medium.com/polkadot-network/polkadot-substrate-and-ethereum-f0bf1ccbfd13

[2] Vitalik Buterin, Ethereum: A next-generation smart contract and decentralized application platform

http://people.cs.georgetown.edu/~clay/classes/fall2017/835/papers/Etherium.pdf

[3] Juan Benet, IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3)

https://github.com/ipfs/papers/raw/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf

[4] Filecoin Spec

<https://spec.filecoin.io>

[5] James S. Plank, Erasure Codes for Storage Systems

<https://www.usenix.org/system/files/login/articles/10_plank-online.pdf>

[6] Substrate Docs

https://substrate.dev

[7] Jeff Burdges, Alfonso Cevallos, Peter Czaban, Rob Habermeier, Syed Hosseini, Fabio Lama, Handan Kilinc Alper, Ximin Luo, Fatemeh Shirazi, Alistair Stewart, Gavin Wood, Overview of Polkadot and its Design Considerations

<https://arxiv.org/abs/2005.13456>

[8] Jaime Galán-Jiménez and Alfonso Gazo-Cervero, Overview and Challenges of Overlay Networks: A Survey

https://www.researchgate.net/publication/50199321\_Overview\_and\_Challenges\_of\_Overlay\_Networks\_A\_Survey