HERE
COMES
THE
EVOLUTION

Noosphere

WHITEPAPER

NOOSPHERE FOUNDATION | VERSION 1.1 | SEPTEMBER 2018

Contents:

Summary

Noosphere Foundation presents a new vision of how blockchain technology can be used today by businesses across various industries. Noosphere is a proprietary development featuring a modern blockchain platform and a whole new ecosystem called **Heterogeneous Taraxacum**.
Noosphere is a blockchain-based, service-oriented environment designed to host a plethora of SaaS located in shards. A number of innovative solutions built on top of a blockchain, including service-oriented sharding, a flexible consensus mechanism and an inter-service exchange, have all given **Noosphere** a great many advantages compared to other blockchains.

Noosphere Foundation, with its proprietary blockchain platform, offers a wide-ranging suite of instruments and services available to users of the ecosystem: from decentralized data storage and computing capabilities to bespoke security and Python-written smart contract services. On top of that, third party developers are welcome to build their own services within the Taraxacum Ecosystem, integrate their own encryption protocols and digital signatures. Priority is given to information security: the ecosystem employs special algorithms to protect against most common blockchain threats like Sibylla, Eclipse Attack, 51% Attack, etc. A special focus is placed on the system's ability to integrate third party blockchain solutions into the Heterogeneous Taraxacum Ecosystem: it is done via an open API of Noosphere and a set of connector interfaces enabling users to transact in all types of cryptocurrency. Noosphere Foundation strongly supports the continued development of Heterogeneous Taraxacum Ecosystem by operating an in-house technical center and encouraging third party developers to contribute by creating their own, in-platform software services.

1 Concept

Noosphere is a new step in the evolution of cloud technology, cloud computing as it offers Blockchain-as-a-Service to traditional businesses worldwide. Despite a plethora of large cloud platforms being available in the market and resource and service prices dropping year-on-year, all of those still remain centrally managed. Naturally, centralization raises a number of issues: data privacy, data security and the dependence on service provider, the latter putting at risk the level of freedom of any business.

Noosphere is a blockchain-based, service-oriented sharding platform designed to provide cloud services and broad cloud computing capabilities. The Noosphere platform incorporates a multi-blockchain cloud ecosystem called Heterogeneous Taraxacum in which blockchain technology and cloud solutions can be utilized to the fullest advantage by any traditional business.

Noosphere's innovative decentralized architecture offers users a breakthrough corporate cloud experience in the likes of Azure, AWS and Google Cloud, taking scalability and data security to new heights.

The Noosphere Platform gives direct access to an array of cloud computing capabilities, customized products and services which can handle a range of tasks facing businesses and corporations around the world: analysis and processing of large amounts of data, internal and external corporate communications, creation of business applications, to name but a few.

1.1 A NEW BLOCKCHAIN MODEL: NOOSPHERE BLOCKCHAIN AS A SERVICE

Not every business can afford to hire a dedicated staff of developers to integrate the latest technology solutions into the company's operational workflows. This is why today's businesses need not just a blockchain system, but customized platforms and ready-made services easily embedded in a company's management structure. The technical and ideological development of blockchain is a very fast process with newer versions of blockchain satisfying the current needs of traditional business better. Noosphere Blockchain is the next generation of blockchain that caters to any industry (fintech, health care, logistic etc.) by offering a valuable model of cooperation based on shard-oriented services easily integrated into the current IT-landscape of real business.

The Noosphere Blockchain as a service (NBaaS) model enables end-customers/clients and professional developers to instantly deploy and test their solutions based on blockchain technology, bypassing any capital costs at testing phase. This economical and reliable solution is based on a cloud platform allowing companies to use up-to-date technology for interaction.

At the corporate level, blockchain is used as a common data layer on top of which applications of a new type can be created. Based on the principles of decentralization and data reliability, applications allow for cost reduction and fraud risk minimization.

As Noosphere continues to expand its development, new cloud-based decentralized services and applications will emerge helping to optimize most day-to-day processes run by SMB (small- and medium-sized business). Noosphere aims for a continuous improvement of the decentralized cloud ecosystem that hosts Cross Platform-DApps & Services. By taking advantage of the innovative architecture of blockchain, all implemented solutions will boast an infinite scalability and unprecedented level of security. Said advantages have been technically achieved through improvements of current blockchains and eliminating the legacy problems suffered by previous versions of blockchains. Noosphere advantages are supported by special ecosystem's services ensuring unprecedented levels of security (Loki and DDAP services) and scalability (DAR & DDNS, ABG and PVM services).

1.2 NOOSPHERE IS BASED ON SERVICE-ORIENTED SHARDING FOR BLOCKCHAINS

Noosphere is based on **Service-Oriented Sharding for Blockchains**, a technology that will unleash unlimited capabilities for businesses, including SMB that will obtain an easily scalable, high-throughput blockchain.

Thanks to **Noosphere**, blockchain will be made adaptable to whatever requirements an organization or a firm may have. The key value of the proposed concept lies in solving the cornerstone issues: scalability, compatibility and general security, regardless of blockchain's internal specifications. **Noosphere** offers its users convenient and efficient venues for business application of blockchain technology and smart contracts.

2 Noosphere – High-End Blockchain

2.1 BESPOKE SECURITY

Handling business-critical data places utmost importance on security and confidentiality. A user must be sure that their data will not be transferred to third parties or otherwise disposed of without their knowledge. If the data is stored centrally at a single owner's end, then it becomes a target for industrial spies, hackers and even may capture the attention of pro-government organizations.

Blockchain, like any other online information system, is a popular target for hackers. In addition to DDoS being the most widespread type of attack, we know of multiple new methods conceived specifically to influence blockchain platforms: Sibylla, Eclipse Attack, 51% Attack and many others. Having analyzed all those threats and the current deficiencies that expose the existing blockchain architectures to hacks, Noosphere has been able to equip its subsystems with the most effective and proactive remedies.

Given potential threats, Noosphere comes armed with the most advanced and proven cryptographic data protection technologies: secp256k1 и BLAKE 512, however, the ecosystem has a strong potential for further development considering the implementation of ED25519 with X25519 keys, SHA256, Keccak 512. Besides, the algorithms offered by Noosphere ensure broad flexibility allowing users to create encryption protocols, digital signatures and, if needed, run post-quantum cryptography algorithms.

2.2 FAULT TOLERANCE

Thanks to its decentralized architecture, Noosphere provides distributed computing resources with a close-to-zero failure probability at growth.

The application of decentralization principles within the Noosphere platform guarantees a high level of fault tolerance of the entire system: even in cases of a complete or a partial outage of individual nodes, the system itself remains operational.

Experiments have been conducted and the results indicate that even 30% of Noosphere's key nodes failing will not affect the speed of building new blocks, the consensus mechanism and any other key features of blockchain.

2.3 COST EFFICIENCY

The Noosphere platform is created under the umbrella of Noosphere Foundation, a non-profit organization that promotes the development and support of the free blockchain developer community. All services developed by Noosphere are delivered as shareware to end users. As opposed to commercially available centralized cloud services, decentralized solutions require by far less overheads to maintain the hardware infrastructure and hence a flexible and competitive financial policy.

Third-party developers contributing new services to the Noosphere platform may charge a fee for their use. All in-platform services and applications use tokens as a payment means; those are linked to the main Noosphere token at the time of creating a related service. This said, service developers determine financial policies independently: frequency of payments, bonuses and specials. The convenience of such an approach cannot be overestimated by end users as they will be able to plan their expenses more effectively and choose between an array of payment plans.

2.4 GLOBAL PRESENCE

The distributed decentralized architecture enables Noosphere's presence in every part of the world. No other cloud service provider can match Noosphere by the level of resources and capabilities to deploy their data centers around the world.

When creating new services or using existing ones, a developer or an end user can choose the necessary resources for data processing, a certain region of presence and in which geographical area their data should be stored.

2.5 APPLICABILITY

The Noosphere platform imposes no limits on developers with regards to the tools they use and application fields they explore with the in-platform services being created. The fields in which blockchain technology and smart contracts can be applied are endless: finance, logistics, medicine, elections and thousands of other spheres where strict requirements to data security and authenticity are a must. Going beyond the traditional use of blockchain platforms, the Noosphere has developed a smart-contract wizard and a tool kit for conducting external ICOs.

3 Broad functionality

Noosphere creates a pioneering structure - the Heterogeneous Taraxacum Ecosystem. Due to the open design and development principles, as well as the core technologies used, all services in the Noosphere system have unlimited potential for use.

This ecosystem consists of a set of services, already operational or currently in development, and it will also include a huge number of third party developed services with the assistance by Noosphere Foundation. Heterogeneous Taraxacum Ecosystem offers great opportunities for IT and blockchain market players to build their own decentralized solutions using Python smart contracts. The flexible architecture of the ecosystem simplifies integration or building new software services and the services' synergy makes it more attractive to third party developer teams. Those teams and their services may use, by default, all the current features of Taraxacum, such as decentralized data storage or cloud computing.

The system understands various programming languages and algorithms and, therefore, third party applications can be built in every shard of blockchain. Big data handling, distributed neuron networks, data backup and storage systems, high-performance computing are only a small part of what Noosphere can offer a user.

While each service adopted by Noosphere becomes its integral part, it exchanges data with other in-platform services, thus transforming Noosphere into a fully-fledged distributed cloud ecosystem unleashing unlimited possibilities.

3.1 FREEDOM OF DEVELOPMENT

The currently available cloud service platforms are handicapped by a limited set of functional tools which often fall short of meeting consumer requirements. The core of the Noosphere system and all the services developed by the Noosphere Foundation are open and accessible to whoever wishes to study and use them.

Noosphere's further development is driven by the input from the free community to creating new decentralized services and apps and, therefore, the platform sets no restrictions on developer toolkits. Service software can be run on any platform - Linux, Unix, Microsoft Windows, MacOS and can be written in any programming language - C, C++, Java, Python, JavaScript, Go, C# and others.

This level of flexibility has been made possible through creation a fully-fledged API for all core services within Heterogeneous Taraxacum Ecosystem. It means that external teams' services in the future will be available and interactive at all levels: data, transactions, synchronization requests, etc.

3.2 ADOPTION SPEED

The Noosphere test network features an extended toolkit for testing services in the sandbox where it takes almost no time to develop and adopt new functionality. The built-in service monitoring software, highly customizable and easy to change, allows for monitoring system load and fault tolerance values.

One should bear in mind that the end user always chooses user-friendly solutions. To this effect, Noosphere has created NooStore of DApps & Services - a portal to all platform-based solutions and a medium that developers can use to promote and sell their applications and services.

3.3 SDK - SOFTWARE DEVELOPMENT KIT

Standards have been specially designed for Noosphere architecture to simplify the development of services on it. The standards are convenient and easy to access and include API building principles, service-to-service linkage, access to the Noosphere system's core, routing and others. With the wide variety of services at hand, the standards include common mechanisms for using those services, thus ensuring accelerated adoption and operation. In order to give third-party developers some pointers, the Noosphere Open Knowledge Base will be created containing brief guidelines and system interaction examples.

Among Noosphere's extra advantages over all existing cloud solutions is having a multi-function toolkit available for any user to develop and implement new services. Each such service, upon being developed, becomes part of the overall ecosystem expanding Noosphere's capabilities. All created services are not mere user applications; they are implemented on cloud-based decentralized technology and enable developers to monetize their R&D products by providing services to end users [for more details on the Service Ecosystem, see paragraph 3.5].

3.4 BLOCKCHAIN INTEROPERABILITY

Noosphere creates an interoperable system resolving the global issue of blockchains' plurality and isolation. As there is no communication between various blockchains, the technology is severely handicapped where application fields are concerned. No business can operate in isolation from the outside world. Interaction with counterparties, partners, auditors and regulatory authorities calls for better flexibility on the part of a business on the communication side.

Noosphere's services are provided with APIs facilitating third-party blockchains' integration. Also, creating a gateway for a crucial integration is easy and fast.

Noosphere's ecosystem is technically capable to accommodate any third party blockchain. For convenience and transparency of use, each platform is accessed through special service gateways (connector interfaces) providing users with unified data exchange protocols. This way, inert centralized structures can be replaced allowing for organizations from different business environments to quickly and efficiently build optimized blockchain solutions to secure their needs.

3.5 SERVICE ECOSYSTEM

Within the **Heterogeneous Taraxacum Ecosystem**, Noosphere Foundation's developer community creates the basic business-critical services based on the principles of decentralization and security. Their goal is to lay a solid basis on which more complicated services will be able to run once they are developed by interested parties (third party teams of developers, Noosphere community, etc.).

Heterogeneous Taraxacum Ecosystem is an integrated scheme consisting of a set of software services based on decentralized blockchain shards. Noosphere Foundation creates the core shards, supporting the synergy of ecosystem and providing a quick start for third party teams of developers.

Each component of the ecosystem is a single service shard that can function independently of the others or vice versa in close relationship with other service shards. At some point, throughout its life cycle, such a shard can serve as the basis for creating a multitude of other shards.

1 **DAR & DDNS** – Dynamic Application Routing & Dynamic Domain Name System - a service that ensures load balancing, data stream routing and IP-level fault tolerance with integrated dynamic DNS functionality for any applications.

2 **NTM** – Noosphere Transaction Mixer - an additional transaction animation service usable not only for any cryptocurrency, but also in transmission of any confidential data

3 **DDAP** – Decentralized Directory Access Protocol - a service offering LDAP functionality with enhanced safety and fault tolerance requirements.

4 **ACS** – Autonomous Copyright System - a copyright protection service.

5 **Loki** – a service for protection against DDoS attacks that enables timely detection of threats through continuous communication channel monitoring and redirecting traffic to special nodes in order to simulate success for the attacker.

6 **EBS** – Effective Backup Service - a fast and efficient data backup service that makes it possible to skyrocket backup speeds and snapshot generation speeds as compared with centralized services due to automatic data sharing between backup nodes in the course of transmission.

7 **DHPC** – Decentralized High Performance Computing - a service that facilitates transfer of existing computing tasks and software into the Noosphere system and adaptation for the use of distributed computing capacity and is also involved in this type of computing.

8 **PVM** – Python Virtual Machine – a virtual smart contract processing machine based on a Python-like language including Smart Contract Designer per built-in templates. This service features two mechanisms: writing a smart contract based on a template and without any programming or doing so using the programming code for smart contract in the Python language.

9 **ABG** – Any Blockchain Gate – a template service that offers standardized API to integrate operations with any third-party blockchain.

4 Architecture and Technology

Despite all their advantages over traditional centralized systems, decentralized blockchain technologies have a number of weak points. Those primarily relate a comparatively low throughput capacity and limited scalability. Once the said problems have been solved, blockchain systems will successfully compete with centralized platforms.

4.1 SERVICE-ORIENTED SHARDING

Sharding is at the heart of the Noosphere platform's architecture. A standard method of using shards is separate processing of transactions, which are distributed among servicing shards according to a certain criterion. It will make it possible to increase considerably the throughput capacity and the speed of operation of the system as compared against other systems, in which a single copy of the blockchain is stored in nodes.

Each shard of the Noosphere platform is a blockchain capable of operating absolutely independently as it does not require any permanent connection to the other shards. Moreover, a shard is the system's basic blockchain. Thus, Noosphere's elementary composite unit is a shard.

The idea of service-oriented sharding was first mentioned in a study on Service-Oriented Sharding for Blockchains by Adam Efe Gencer. The key distinctive feature of this idea is the use of shards as services.

Each shard represents an isolated service that performs certain regulated functions and is capable of information sharing with other shards in the system. New shard-based services are dynamically added to the system and expand the system-wide functionality. On top of that, service-oriented sharding makes it possible to implement easily any sidechains and to use any third-party blockchain as a shard in the manner explained above.

4.2 SHARDING TECHNICAL

Sharding is a method of horizontal data partitioning. The sharding process as such is divided
into three components:

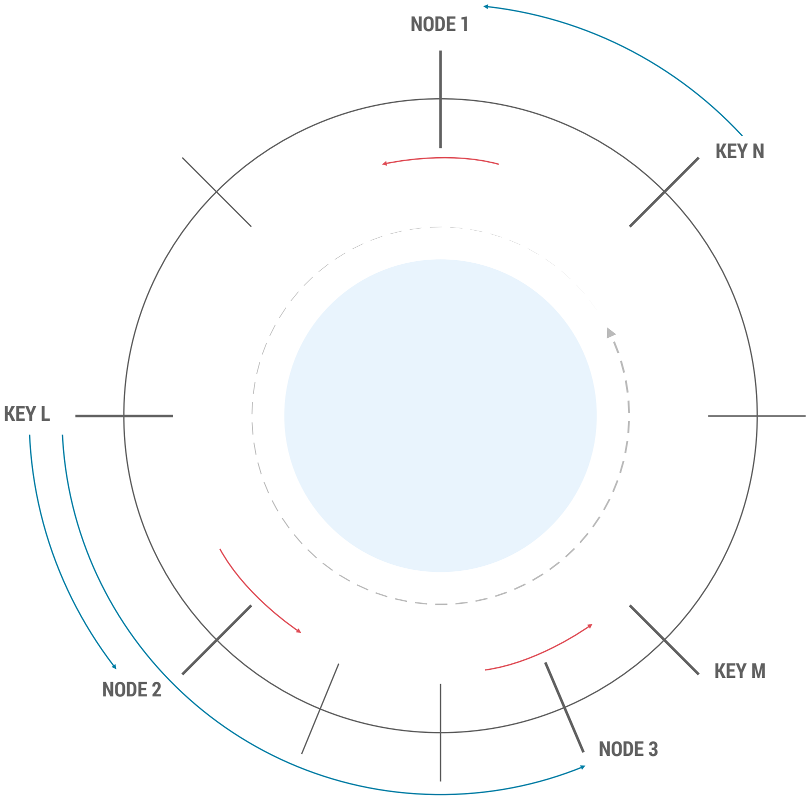
- Choosing a sharding function;
- Re-sharding, or redistribution of data;
- Data routing (determination of their physical location).

In its simplest form the sharding function is f(x_1,x_2,x_3,...,x_n) = y, where x_n is a sharding key and y is the shard. The right choice of sharding keys has a direct bearing on the sharding efficiency. These keys usually consist of a number of available nodes and the ID of the user who owns the data. The sharding function f is a special hashing function. It may be either consistent hashing or HRW hashing.

The efficiency of the selected function is obvious at the time of data re-sharding, when n out of K nodes become disabled and data must be redistributed among the nodes remaining online, or, vice versa, the network load goes up sharply and new shards must be generated for load distribution. The existing approaches to sharding in blockchain platforms neglect the importance of this problem and deal with selection of a hashing function only, in the belief that the network load is constant and the initially selected shard configuration will not change over time.

Consistent hashing is a special type of hashing characterized by the fact that, when the hash table is restructured, only K/n keys on average must be reassigned, where K is the number of keys and n is the number of slots. By contrast, changing the number of slots in the majority of traditional hash tables results in the reassigning of almost all keys. Consistent hashing eliminates the need for rehashing the keys when the selection of active nodes is changed (addition / deletion). Instead, the whole selection of hashed keys of nodes and data is in a closed circle, and with addition of new nodes the load can be distributed dynamically without the need for re-hashing. HRW hashing is a similar function, but its capabilities are much wider, and consistent hashing can be considered a special case in particular situations. HRW-hashing also distributes the key sets in the circle using a unified hash function. Unlike consistent hashing, HRW does not require any precomputation or key storage. Object O_i is placed on one of n nodes N 1 , ..., N n , calculating n hash values h (O i , N j) and selecting node N k, which yields the highest value for the hash function. If a new node N n + 1 is added, the new locations or queries of the objects will calculate n + 1 hash values and select the highest of them. If an object that is already located in the system at N k is compared with this new node N n + 1, it will be loaded anew and cashed at N n + 1. In future, all customers will get it from this node, while the old cashed copy at N k will eventually be substituted with a local cash control algorithm. If N k is disabled, its objects will be evenly reassigned to the remaining n-1 sites. Skeleton type variants of the HRW algorithm can reduce the object placement time O (n) to O (log (n)) due to a lessor global homogeneity of placement. However, if n is not excessively big, the placement cost O (n) of the basic HRW will hardly pose any problems. HRW fully avoids all overhead charges and difficulties related to correct processing of several keys for each node and associated metadata.

NOOSPHERE HRW HASHING

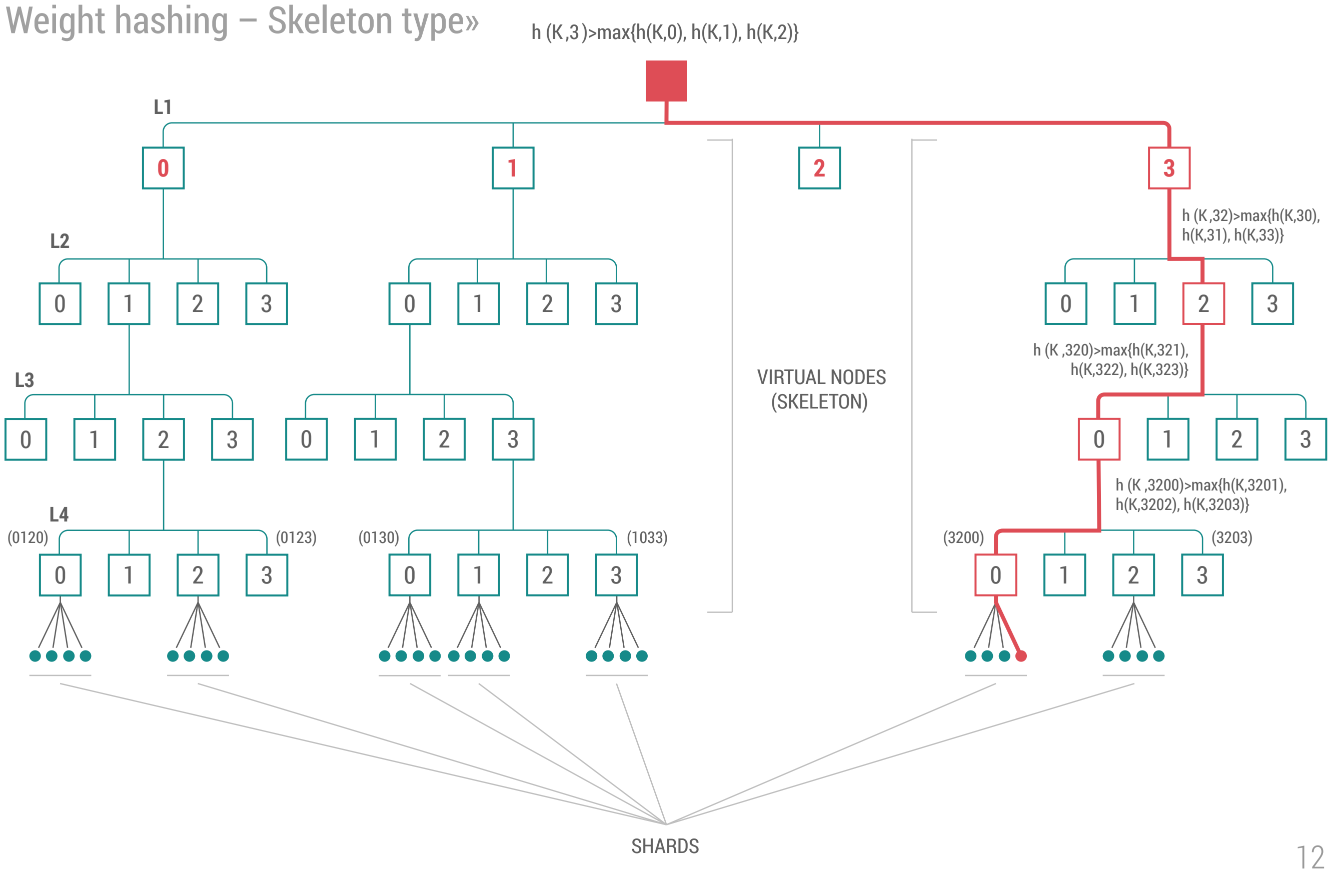


When n is extremely high, the skeleton type variant of the HRW algorithm makes it possible to reduce considerably the object placement time. This approach creates a virtual hierarchy structure and reaches the runtime of O (log n) by applying HRW at each level of hierarchy downward. At the first step, constant m is selected and n node is formed in c = n / m shards S 1, = { N 1 , N 2 , ..., N m }, S 2 = {N m + 1 , N m + 2 , ..., N2m }, ... At the second step, a virtual hierarchy is created using these shards placed on the leaves of T tree of virtual nodes, of which each one has branching f .
The enclosed diagram shows a shard sized m = 4, with skeleton branching f = 4 and the total number of real nodes equal to 128.
Instead of applying HRW to all of the 128 real nodes, we can first apply HRW to the 32 lowest virtual nodes by selecting one of them. Then we apply HRW to four real nodes in own shard and select the winning node. We need only 32 + 4 = 36 hashes, not 128.
We can start with any level of the virtual hierarchy, not only from the root. A low-level launch requires a great number of hashes, but it can improve load distribution in case of failures. Besides, the virtual hierarchy needs no saving. It can be created as required, because the names of virtual nodes are just base-f representation prefixes. It is clear that T has height h = O (log c) = O (log n), as m and f are both constant. The job performed at each level, O (1), as f is constant.

For each given object O it is obvious that this method selects each shard and, consequently, each of n nodes with equal probability. If the selected node is unavailable, we can choose another one inside the same shard in a conventional manner. Or we could go up one or more tiers in the skeleton and choose an alternative among virtual sister nodes at the same level and then go down in the hierarchy to real nodes as described above.

The m value can be selected on the basis of such factors as the expected failure frequency and the desired load balance ratio. A higher m value leads to a lower load in the event of a failure due to higher search costs.

NOOSPHERE «Highest Random Weight hashing – Skeleton type»


h (K,3)>max{h(K,0), h(K,1), h(K,2)}
L1
0
1
2
3
h (K,32)>max{h(K,30), h(K,31), h(K,33)}
L2
0
1
2
3
h (K ,320)>max{h(K,321), h(K,322), h(K,323)}
L3
VIRTUAL NODES (SKELETON)
h (K ,3200)>max{h(K,3201), h(K,3202), h(K,3203)}
L4
(0120)
(0123)
(0130)
(1033)
(3200)
(3203)
SHARDS

Of no less importance is the issue of data routing. It acts as a determinant factor of the speed of data presentation to the end user and data exchange between shards. The simplest and most inefficient approach is to store routing tables with customers. But this method is deprived of scalability. The variants of using individually separated coordinators and proxies are out of the question due to decentralization of the blockchain system architecture.

Therefore, data routing is for node systems themselves to take care of. The simplest solution that makes it possible to find, within logarithmic time $\boldsymbol{log_{n}}$, a source of required data is for nodes to store information about their neighbors and a limited set of nonadjacent nodes. Thus, at any request, if the requested data are not stored on the current node, it routes the request to its neighbors (if they hold the key) or further on to a random nonadjacent node.

4.3 SCALABILITY AND CONSENSUS

Sharding is the basis of the Noosphere architecture. Each created service is a separate shard with its own set of computing resources. Service shards are both independent being able to exist on their own and capable of exchanging data by dividing input data stream processing among themselves.

In order to accelerate data processing within each shard, Noosphere Foundation has developed a special consensus algorithm - Convolutional BFT. Its core distinctive feature is the capability of efficient data convolution when an information block is distributed among participants for consensus purposes. Each information block is not transmitted in its original form, but in a convoluted form that occupies a relatively small volume and allows reducing data transmission costs as compared with traditionally implemented blockchain consensuses.

4.4 CBFT TECHNICAL

The network architecture of consensus systems of the existing blockchain platforms is based on the principles of stationarity of the system’s operating period and is built exclusively with the use of the Poisson flow of events. The need to analyze the type of distribution law for the input data stream and the length of transition periods is determined by the fact that the latter may constitute a considerable portion of the system’s operating period, while the law of input stream distribution may have a substantial impact on statistical indicators of the system’s output parameters. This is why optimization of the system’s operating characteristics in general is impossible without due consideration for the nonstationarity period and the impact of the type of the law of distribution of the input data stream.

The CBFT algorithm is based on 23 nodes consensus, where each node (except for the main one) is an entry point into the system for external users. Each node is sending data to all other nodes. 23 nodes, by a vote, elect the main node (a stem-node) which, by chance, elects a node-validator of the next block in a chain.
The validator node forms the next block from incoming transactions and sends it to the other participants of CBFT consensus. Each node verifies the block and votes with its own signature (EDS). A time meter is used by the validator node to summarize the voting results and, in case a consensus is reached, it sends this new block to the main node (stem-node). The cycle repeats until the stem-nodes and all active nodes (participants of consensus) are changed as a result of a new election in the blockchain platform.
While the CBFT algorithm uses the PoS principle, the proof of nodes’ reliability is their possession of Noosphere tokens (NZT). The first election of trusted nodes by token holders with a sufficient share of tokens is a point at which the blockchain platform starts to function and validator nodes are defined within the system.

Let’s consider that we have N node-candidates in our voting process and the algorithm is focused on choosing 23 winners – M_1, M_2…M_{23} from N-pull candidates. Also we have W voters of tokenholders, who might vote for any M_n candidate with their X tokens of whole Y amount of all tokens in Noosphere. Then, results R (in time t) of voting assignment as function may be defined as:

$R (t) = f (X, M_n, W)$;

At that X > L, L – a minimum number of shares for voting.

Election M_1…M_{23} of nodes in consensus is a result of maximization of R (t) function when X tokens given by voters (W) and received by candidates (M_n).

At that X => 0.15*Y;

Current CBFT algorithm starts only with the main node (stem-node) and that’s why the next round is the election of the stem-node. The same principle of maximum of votes is used in election of the main-node (stem-node) M_0 by 23 nodes in consensus; each of 23 validator nodes has only one vote.
Then, results Rsn (in time t1) of voting assignment as function may be defined as:

$R_{sn} (t_1) = f (M_0, M_1…M_{23})$;

Voting for yourself is prohibited and each node has only one vote to cast in each round of an election. Also, a tie-breaker algorithm is applied to avoid a tie.

In practice, the system input stream intensity is not a constant value, as it changes over time. These changes are related to the period of mass mailing of the candidate block in the process of reaching consensus among the system’s active participants and the duration of request processing in the system, namely, analysis of the candidate block, which consists of a set of transactions of the current validation period.

The development of the CBFT consensus algorithm has generated some statistical information that reflects changes in the intensity of the input data stream in the distribution system depending on the time of distribution of the candidate block, its size and duration of processing. The collected statistical data are approximated using interpolated cubic spline $S_g(t)$ of a piecewise-polynomial form with prescribed boundary conditions, i.e. at each sector $[t_j, t_{j+1}]$ numbered j approximating function$S_g(t)$ takes a polynomial form

$$P_j(t) = \sum_{i=0}^{k-1} a_i^{(j)} (t - t_j)^i, k - 1 = 3$$

The boundary conditions lie in the periodicity condition, i.e. coincidence of the values of the first and second derivatives on the boundaries of interval $[t_1, t_N]$. Spline construction amounts to determination of a set of coefficients $a_i^{(j)}$ by linear system solving. Interpolating spline $S_g(t)$ is built in such a way as to meet interpolation condition $S_g(t_i) = y_g(t_i)$,i = 1,…,N. for tabulated function y .
In order to determine the specific dependence of the input stream data intensity on $\lambda_g(t)$ at certain average intensity λ the obtained spline $S_g(t)$ should be multiplied by the average intensity and divided by the average value of the spline itself $\bar{}(S_g(t))$:

$$\lambda_g(t) = \frac{\lambda}{\overline{S_g}(t)} S_g(t)$$

$$\overline{S_g}(t) = \frac{\sum_{i=1}^{N} S_g(t_i)}{N}$$

where λ is the actual average intensity of the input stream.

The proposed input stream model implies that the time between arrivals of requests ξ is a continuous random value that can be distributed according to various distribution laws, such as the exponential law, the Poisson law, the normal law or the uniform law.

The Poisson law governs the main consensus mechanism to input stream data into the system and analyze the candidate block. Its functionality may be varied depending on the main factors: net activities, net load and speed of inter-service interaction.

4.5 INTEL SGX

In dealing with data that demand higher safety standards, including passwords, closed keys, personal information, etc., Noosphere uses the Software Guard Extension technology by Intel. This is the only technology currently available in the market that makes it possible to protect user data at the level of a computer's central processing unit.

As a matter of comparison, any other technologies offer protection at the operational system level only and are just the first line of protection against malicious acts. In addition, Intel SGX technology is used in the implementation of particular algorithm PoET and cloud computing when Convolutional BFT cannot be used or when speed requirements are not so high. Consensus algorithm PoET has proved to be efficient in the HyperLedger project.

4.6 SGX - CBFT

For critical services with an extremely high risk of attacks, the hybrid consensus protocol SGX-CBFT can be used. Its operation is similar to Convolutional BFT described earlier, but market participants enjoy an additional capability to validate all adjacent nodes remotely, thanks to the Intel SGX technology.

4.7 INTERSERVICE EXCHANGE

The wider the network, the stronger is the impact of the need of prompt data routing on data transmission. The existing protocols of data routing via the Internet, such as OSPF, BGP, IGRP and others, are not suitable for decentralized systems, as they are based on the existing data processing centers that are unacceptable for a decentralized system by default.

For the purposes of the Noosphere architecture, an ecosystem of service shards, the Swift Torus Routing protocol has been developed. It is based on other principles as distinct from the existing protocols of data routing via the Internet, because it operates with entirely different entities, and the need for prompt data routing among dynamic nodes increases multifold. The search algorithm for an efficient route among services is based on research in the areas of topology, the graph theory and the queuing theory, as well as on the best practices of contemporary graph processing engines like Mosaic, Chaos, Giraph and others. As a matter of comparison, route calculation on the basis of a graph with about 700,000,000 points using the Hilbert-ordered tiling scheme takes one second on four Xeon Phi processors.

As concerns network graph design, discrete-time models are used. In these types of models, network topology changes are presented as a periodically recurring sequence of topology snapshots S, divided by an interval sized $\Delta t=T/S$ *, where T is the period of recurrence of topological states of the set.*

Each snapshot is compared with graph G = (V, E), where V is a set of nodes and E is the number of communication channels. Routing tables are preliminarily calculated for each finite set of graphs {G}, recurring during period T. These tables are distributed among nodes and used at the required point of time Δt.

The current algorithms employed by inter-service exchange (the Swift Torus Routing Protocol) are a simple and smart way to connect different shards - at the transport layer and at the network layer, using the OSI model. Swift Torus Routing is one of core systems in Heterogeneous Taraxacum as it speeds up inter-service exchange using the above-mentioned approaches: a quick data snapshot and the routing table.

4.8 SPEED

Standard and generally accepted data transmission protocols are unable to cater to the needs of present-day platforms. This is true not only for decentralized systems. Centralized platforms rely on wide-band data transmission channels, while decentralized platforms are focused on new consensus algorithms. In both cases, development is restricted, because these solutions can’t ensure long-time sustainability or meet increasing throughput requirements.

Noosphere Foundation has developed a conceptually new data transmission protocol combining efficiently TCP and UDP protocols. This protocol is called UFT - UDP Flow Transmission. An elementary data transmission unit in a blockchain is a transaction. UFT has been designed for the purposes of the technology of transmission of transactions, or short information messages with a predetermined structure. In this way, the available bandwidth can be used 100% with a guaranteed transmission.

5 Noosphere Foundation

Our mission is to develop and promotenew Blockchain technologies by marrying scientific progress and sustainable consumption of energy, technologicaland human resources.

The main objectives of Noosphere Foundation include:

- Promotion and improvement of the Blockchain technology;
- Development of and support to a free community of blockchain developers;
- Development and maintenance of the Noosphere platform;
- Applied research and development in the IT industry;
- Support to and development of new technologies focused on generating renewable sources of energy.
- Development and popularization of the blockchain by humanizing the technology and increasing the public awareness of its benefits.

Noosphere Foundation is in charge of the development and strategic positioning of the Noosphere platform. The Foundation has a stable administrative structure that ensures prompt communications among all participants in the blockchain community. It comprises an administrative center, a technological development center, and a strategic center.

The Administrative center is responsible for general matters of the organization's activities, including financial, legal, and HR issues. It plans the budget, prepares financial statements, performs due diligence of the project, monitors compliance with laws of the countries of operation, and generally administers the activities of Noosphere Foundation.

The priority tasks of **The Strategic planning center** are management and decision-making with regard to the prospects of development of the Noosphere platform. Its subdivisions are in charge of organization of marketing support for projects, ensuring communications among participants in the community, and holding themed conferences, seminars and training courses.

TThe Technological development center is responsible for development, testing and promotion of the Noosphere platform. It provides advisory and engineering support to third party service developers on the platform.

6 RoadMap

This section describes various stages of development of the Noosphere platform, as well as associated technologies and the main service shards that will become the basis for generating new user shards and will expand their functionality.

Each development stage will result in a public presentation of the achieved progress using the developer's testnet open for everyone.

Due to involvement of the free developer community in the development, this roadmap is subject to adjustment if any functions assume a priority importance at some point. All stages of the Noosphere platform's development will be publicly available at github.com.

I ROOT SHARD AND SHARD DESIGNER – Q3 2018

A. Stem node & Twig node - stem node & twig node base functionality allowing for launching in the standalone mode.

B. CBFT – development of the main consensus algorithm of the Noosphere platform.

C. Nodes API Documentation – node interaction interface documents.

D. Rest API for Blockchain explorer.

E. Blockchain explorer – implementation of the platform's main monitoring instrument.

F. Launch of the Noosphere platform's Developer's testnet.

II NZT SHARD AND CORE FUNCTIONS - Q4 2018

A. Parallel transaction execution – development of the node functionality for parallel processing of transactions from the overall input stream.

B. UFT – development of a network protocol focused on transaction processing in a distributed decentralized network for a higher speed of the platform operations.

C. Swift Torus Routing – development of a swift torus routing protocol for acceleration of data synchronization between shards.

D. SPI - shard programming interface – development of Rest API for unitizing the protocol of data exchange between shards.

E. NZT shard – development of the Noosphere platform's shard, which implements the main support function for NZT coin.

F. Rest API for NZT shard – development of an interface for access to NZT shard.

G. Implementation of Stem node & Twig node full functionality.

III NOOSPHERE TESTNET - Q1 2019

A. Launch of the Noosphere platform's testnet in public access.

B. PoET – implementation of an additional consensus algorithm based on the Intel SGX technology.

C. SGX-CBFT - implementation of a consensus algorithm based on Intel SGX and CBFT.

D. PoD – implementation of the type of transaction - "payment on demand".

E. Public demonstration of the speed of operation of the Noosphere platform on PoET & SGX-BFT consensus algorithms.

F. Shard designer – an instrument of designing and creating new shards on the Noosphere platform.

IV NOOSPHERE SERVICES - Q2 2019

A. Launch of Mainnet of the Noosphere platform.

B. NooStore – store implementation for access to the services created on the Noosphere platform.

C. NTM - Noosphere Transaction Mixer.

D. PVM - Python Virtual Machine and Smart

E. Documentation for shard designer – documents for Shard designer.

V NOOSPHERE SERVICES - Q4 2019

A. DAR & DDNS – Dynamic Application Routing & Dynamic Domain Name System.

B. DDAP - Decentralized Directory Access Protocol.

C. ACS - Autonomous Copyright System.

D. Loki - DDoS Protection Service.

VI NOOSPHERE SERVICES - Q1 2020

A. EBS - Effective Backup Service

B. DHPC - Decentralized High Performance Computing

C. ABG - Any Blockchain Gate

If you have any questions, please refer to Technical WP

REFERENCES

Bitcoin-NG: A Scalable Blockchain Protocol, Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse, In Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation (NSDI 2016).

M. Castro and B. Liskov, “Practical Byzantine Fault Tolerance,” in Proceedings of the Third Symposium on Operating Systems Design and Implementation, ser. OSDI ’99. Berkeley, CA, USA: USENIX Association, 1999, pp. 173–186.

Chain Inc. Chain open standard: A secure blockchain protocol for high-scale financial networks. http://chain.com/os/, retrieved Sep. 2016.

CoSi: Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning, Ewa Syta, Iulia Tamas, Dylan Visher, and David Isaac Wolinsky and Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, and Bryan Ford, 37th IEEE Symposium on Security and Privacy (SP 2016).

Elastico: A Secure Sharding Protocol For Open Blockchains, Loi Luu, Viswesh Narayanan, Kunal Baweja, Chaodong Zheng, Seth Gilbert, Prateek Saxena, ACM Conference on Computer and Communications Security (CCS 2016)

A.Efe Gencer, R.van Renesse, E. G¨un Sirer, “Service-Oriented Sharding for Blockchains” Initiative for CryptoCurrencies and Contracts (IC3) Computer Science Department, Cornell University, 2016.

L. Luu, V. Narayanan, K. Baweja, C. Zheng, S. Gilbert, and P. Saxena. A secure sharding protocol for open blockchains. In Conference on Computer and Communications Security, Vienna, Austria, 2016. ACM.

PBFT: Practical Byzantine Fault Tolerance and Proactive Recovery, by Miguel Castro and Barbara Liskov, ACM Transactions on Computer Systems (TOCS), vol. 20, no. 4, Nov. 2002, pp. 398-461.

F. Amirjavid and H. Mcheick. Service Oriented Distributed Computing, Proceedings of 2nd International Conference on Software Technology and Engineering (ICSTE 2010) IEE, San Juan, Puerto Rico, USA, October 3-5, 2010.