White Paper v1.0, Released 30th July, 2020

# Libonomy: Artificial Intelligence-based Next Generation Blockchain

Fredrik Johansson, Sarmad Khan, Hamza Gul Kakar, Muhammad Omaid
[fredrik, sarmad, hamza, omaid] @libonomy.com

***Preface*- This document is a technical vision to develop a blockchain system that can revolutionize the whole ecosystem of the blockchain industry proposed by the team of Libonomy. The document covers the major technical aspects of the development stage of the Libonomy blockchain, further highlighting the major improvements and technological advancements proposed by the Libonomy blockchain. The paper focuses on the significance of the advancement and the benefits it can offer to the whole industry. The specification covered in the documents does not entirely make this a comprehensive or a final design. The mechanism will be further added, extended or improved in response to the ideas received from the community and critiques. The document helps in giving an insight of the Libonomy blockchain core and its sub-modules. The details provided in the document covers the core of the protocol and its various aspects. This paper will be revised in future when the prototyping will take place and provide information on upgradation. The final version of the document will be based on the upgradation of the protocol, its testing results and with additional plans on extending the protocol to different aspects of the blockchain.**

## I. INTRODUCTION

The current systems have given the world an ability to achieve more than just making a crypto transaction on a blockchain. Although each system provides its unique features, they face issues in many aspects of the real-world scenarios. In this part of the paper, we will highlight the most critical failures of the current systems.

### *A. Scalability:*

Resource is consumed by the system to carry out a certain operation on the blockchain. Under peak requests, the problem that arises is whether the system can behave consistently or will it grow over time depending on the availability of resources on the network. These concerns are merely not hypothetical but very practical if the network that is introduced is to solve these problems.

### *B. Self-governing:*

There is a huge concern regarding the fact that during peak requests is the system autonomous enough to decide to distribute resources. A system like this needs to be capable enough to make autonomous decisions in securing, improving and inter-linking the other systems. It is a statement of great advancement for the system to self-govern.

### *C. Interoperability:*

Interoperability was discovered recently and it solves the problem of blockchains working in silos. The current system's concern is whether it will communicate with other systems or not. The idea of trading transactions from one blockchain to another is a huge milestone to achieve. With the present applications upgrading it is necessary for the system to align itself with the growth in the industry streamlined with interoperability.

### *D. Fairness:*

Blockchain became popular because one of its crucial features is a decentralized system. Users from around the globe are connected to various blockchains, giving them a right to equal representation. It is a matter of great responsibility that the current system proposed is fair in allowing regular computers to participate in the network to reach decentralization. The system is also under analyses to present itself fair in matters of the distribution of rewards.

The present implementations of the blockchains can only illustrate what they highlight when they are running on a system with high hardware specifications, which in return breaks the overall intent of fairness and true decentralization. As highlighted by Ethereum Parity client, it achieved throughput around 3,000 transactions per second which is only possible when it is running on a high-performance system. However, most practical implementations of the blockchains are limited to around 15 ~ 30 transactions per second because in the public network not all the nodes possess the same resources as tested during their initial prototyping.

The other reason included is that the current consensus algorithms are limited to some extent, this slows down the processing time and the systems are currently based on the architecture of state transition mechanism. Hence, to reach consensus the underlying system needs to share the history of its root and reach the agreement based on the validity. This architecture is followed by POW and POS

based systems such as Bitcoin, Ethereum, NXT and Bitshares. To become successful a trade-off has to be made, therefore, the resulting system has proven to be a great success so far and making historical changes in the industry. Although, the resultant protocol is subjected to limitations in terms of scalability, security, interoperability and fairness. A trade-off system is subject to certain risks and failures, failing to accommodate improvements, only to become a performant and upgradeable solution for the future needs.

Therefore, there is a need for a robust, optimal and autonomous solution, which not only satisfies and accommodates the current systems but the future systems as well. Libonomy is the proposed system for all these problems. It is powered by a consensus engine that is not only powered by AI but is also autonomous, interoperable, secure and scalable. As there is a need for a completely new architecture which could serve as a foundation to build future blockchains efficiently, along with Libonomy, we have introduced the five-layered architecture each with its strengths and functionality, revolutionizing the blockchain industry uniquely.

## II. HISTORY

### *A. Consensus algorithms*

In recent times, an immense amount of research has been conducted in distributed data recording, peer-to-peer transmission, consensus mechanism, encryption algorithm and other computer technologies. SHA256 algorithm was proposed by Guilford J.D which is employed in the blockchain. The original exchange of any length recorded is computed twice by SHA256 algorithm so that it can acquire the hash value and the hash value's length is 256. One of the many hashing applications is the Merkle tree and POW. The Merkle tree has a structure of a tree, where every leaf node has a hash value and a non-leaf node carries its child node's hash value. It stores transaction information and generates digital signatures. It increases the scalability and improves efficiency of the blockchain. It can verify data without extracting the complete blockchain network node. Timestamp was introduced to record the time of block data to solve the problem of "double spending", making it possible for data to reconstruct the history. In addition to proof of existence, timestamp ensures that the database is not manipulated and saves from fraudulent activity. In peer-to-peer technology there is no central node or existence of any hierarchy structure, every node on the network has equal status. Each node will undertake the network routing, data validation and data transmission. To secure data transmission and allow ownership verification, blockchain uses the asymmetric encryption algorithm called ECC (Elliptic Curve Cryptography), with each user having a pair of keys, one public and one private. Users sign the transaction information with ECC, meanwhile, other users can verify the signature with the public key of the signed user. Furthermore, the public key is also used to identify different users and construct their Bitcoin addresses.

### *B. Proof-of-Work (POW)*

PoW is a cryptographic puzzle first presented by C.Dwork and M.Noar. The foundation for it was set to prevent spams and curb the denial of service attacks. Satoshi Nakamoto was amongst the first to adopt this system in the Bitcoin system. Further, a hybrid protocol was presented by Bentov et al, that relied on PoW and Proof of Stake protocols and combined both of their advantages, establishing an element more superior. Ateniese et al proposed an alternative to PoW that is Proof of Space, which specified the amount of memory rather relied on memory access as in PoW. Arthur Gervais et al introduced "a novel quantitative framework to analyze the security and performance implications of various consensus and network parameters of PoW blockchains" (Gervais et al., 2016). They devised optimal adversarial strategies to affect double-spending and selfish mining taking into account real-world constraints and attacks. Alex Biryukov et al introduced Equihash that "an asymmetric proof-of-work with tunable parameters", it is a "PoW based on the generalized birthday problem and enhanced Wagner's algorithm for it" (Biryukov et al., 2017).

### *C. Proof-of-Stake (POS)*

Peercoin first time used Proof of Stake in 2012. PoS generally means proof of ownership of the currency. PoS does not have mining so it does not utilize computing power, like PoW. It solves the energy problem in the current blockchain system such as Bitcoin and Ethereum. The nodes possess a certain amount of stake, that is the currency, in a blockchain. The higher the stake of the party the more likely it is to release a new block and become the leader. A reward is also issued in PoS protocol just like it is issued in PoW. PoS is a more cost-effective method and saves energy. However, there is a problem of monopoly in PoS, which is unfair for many participants. Yuefei Gao et al proposed Proof of Stake sharding protocol to increase scalability. Fahad Saleh introduced the 'first formal economic model' of PoS and explained how the consensus works under it (Saleh, 2018).

### *D. Delegated Proof-of-Stake (DPOS)*

DPOS is a relatively new consensus algorithm that is better than energy inefficient and poorly protected PoW and PoS. It ensures the representation of transactions within a blockchain. DPOS is a fast, outstanding and advantageous consensus algorithm model. To solve the consensus problem, DPOS uses voting and elections, which is fairer and saves computing power. Every holder of the stake can vote, fulfilling a certain number of representatives and all have equal rights. To maintain the 'long-term purity' representatives can be changed by holders at any time. Its main advantage is that it saves computation energy and is more cost-effective than PoW and PoS. DPOS removes the biases caused by PoS with equity and decentralizes the decision making on the network.

### *E. Practical Byzantine Fault Tolerance (PBFT)*

The Practical Byzantine Fault Tolerance (PBFT) is an algorithm that can tolerate Byzantine faults caused by the Byzantine General Problem. Miguel Castro and Barbara Liskov first introduced it in their paper, solving the problem caused by faulty nodes' low efficiency. PBFT is based on message authentication codes that go through three-phase protocols and automatically cast the replicas if failure occurs. It depends on three-phase messages before to execute operations. PBFT consensus is highly efficient and enables high-frequency exchanging. All the nodes in the network are identified and all the faulty nodes are restricted in the network. The requirements set for this consensus algorithm is challenging to apply it to public blockchain Also, the great amount of calculations required for this consensus protocol made it impossible to employ.

### *F. Blockchain Applications*

#### *1) Side Chains*

Side-chains are a new and innovative addition to the Bitcoin protocol which develops a connection between the main Bitcoin chain and an additional side-chain. The interaction will let the side-chains transfer each other's assets with two-way peg. The vision for this framework is to increase the functionality and enhance capabilities through pegging with some other chains for the Bitcoin currency. This allows more extensibility that the Bitcoin system usually allows.

Fundamentally, the validity of side-chains does not depend on provisions, the tokens of one chain are only secured by side-chain when it provides its miners' incentives to convert the data that can be represented by standard approved format. The security of the Bitcoin network cannot be easily changed for other blockchains. Furthermore, it is impossible and unfeasible to merge-mines of Bitcoin miners with side-chain and validate side chain's changes in this proposal.

Cosmos is another innovation that allows trust-free communication between multiple chains to take place. It has deployed the Nakamoto PoW consensus method for Jae Kwon's Tendermint algorithm leading to interchain communication. Essentially, it connects heterogeneous chains called zones with a master chain called Hub. This interchain communication is restricted only to the transfer of digital assets and not random information. Interchain communication allows a return path for data, e.g. to verify and validate the status of transfer from the sender.

One of the significant unsolved problems is defining validator sets for the zoned chains and stimulating them like side-chains. The common assumption is that each zone holds a token of a certain value and pays them with it. The early stages of the design still lack thorough details to achieve scalability over validity. However, the lack of coherence between the zones and the hub can be beneficial as it can lead to additional flexibility over the zoned chains compared to a system with strong connections.

## III. LIBONOMY

In this section, we will cover the components of the Libonomy blockchain, Aphelion Consensus protocol, the architecture of the blockchain and all the protocol specifications of Libonomy.

### *A. Aphelion*

Aphelion consensus protocol is based on artificial intelligence and utilizes multiple machine learning algorithms to fairly distribute the resources among the network, to reach the network and application-level consensus in the protocol. The current blockchains lack a fair distribution of the nodes in the network and there is always a bottleneck in terms of performance. In Aphelion, we have proposed a new mechanism for the blockchain which comprises different pools and accommodates high-performance and low-performance systems. Aphelion consensus mechanism is initially being applied at two major parts in the blockchain i.e. at the time of pool assignment and in Power Pool (validator/Mining Pool). To facilitate fair and optimal assignment of the nodes in

the network, classification agents are programmed to classify the nodes, joining the network, and assign them the respective pool based on their Power Index. The classification agent is primarily based on Artificial Neural Network (ANN) which is trained based on the data extracted from the nodes present in the network, helps to predict and classify the nodes in the network. Artificial intelligence has proved to be a breakthrough in the industry but the blockchain industry still lacks systems which are purely based on artificial intelligence. The major difficulty in utilizing artificial intelligence in decentralized networks is that artificial agents need to be decentralized as well which is why current systems aren't able to achieve that breakthrough. However, Aphelion will fill that gap with the introduction of the consensus engine based on artificial intelligence that is completely decentralized thereby proving to be the first of its kind not only in the blockchain industry but also in the whole software industry as well.

### *B. **Feature Extraction***

To mitigate the issue of taking control of the network and throughput of the blockchain, a feature extraction agent is used to extract the system information which can be used to predict not only the performance of the system but also its overall effect on the network, in terms of scalability and security. To feed data into the classification agent, a feature extraction engine is used to extract the data of the nodes at the time of their initialization and communicate the extracted data to all the classification agents running on the network. The initial scheme involves extracting the computing power, bandwidth, network contribution and life span of a given node; passing this data via the communication layer to the rest of the swarm.

The overall process of extraction and feeding of the dataset to classification includes the following operations:

#### *1) Features*

The four major features that are required for the classification agent includes the node's computing power, bandwidth, contribution and life of a node. For these features, the extraction algorithm after extracting the data, tags the data with the node's key to remove any kind of duplication and prepares the feature matrix.

X1= {Power Ratio, Life-time, contribution}
X2= {Latency, Connections, Bandwidth}
X3 = {Assignment probability, Removal probability}
Input Matrix of characteristics:
M= {X1, X2, X3}

The node's throughput capability in the network can be calculated by passing it to the THROUGHPUT_FUN(M)

#### *2) Dataset Cleansing*

Once the features are extracted and are received by the classification agent, the dataset for the agent is prepared. The cleansing process includes data detecting and correcting corrupt or inaccurate records, which will be modified according to a specific use case.

#### *3) Normalization*

After cleansing the data, it is being normalized to avoid any inaccurate learning. Normalization is one of the most important steps before the data is given to any Machine Learning or Artificial Intelligence Algorithm. To normalize the dataset linear scaling formula is used.

#### *4) Splitting Dataset*

The dataset is split into three portions. 70% of the data will be used for training, 15% of the data will be used for testing and the remaining 15% of the data will be used for validation of the testing data.

### *C. **Classification Agent***

The classification agent utilizes the artificial neural network (ANN) for the classification and clustering information received. During the node's initialization, its power index is extracted via p2p channel and is communicated to all the agents present in the network, this will lead the way for the classification agent to work. In the testnet simulation the parameters of the ANN are assigned as follows:

learning Rate = 0.001;
neurons = 100;
epochs = 1000;
hidden layer = 1
Note: The parameters are tuned based on tests carried out and the outcome achieved. These parameters are subjected to change over time.

The processing done by the classification agent can be broken down into the following manner:

#### *1) Feed Forward Process*

In this process, the output of the network is calculated. To get output in the feed forward process the input is multiplied with weights.

$$net = \sum_{i=0}^{n} w_i x_i + b$$

$$output = \sigma(\text{net})$$

Where w represents the weights and x represents features of our dataset.

*Note: The weights are dynamically changed by the algorithm as the blockchain state is scaled, the weights are subjected to change as well.*

### *2) Back Propagation Process*

Back propagation is the process in which we calculate errors and gradient descent during our training for classification agents. Then, the weights are updated based on error for the gradient.

σ(x) is sigmoid function

$$= \frac{1}{1 + e^{-x}}$$

Derivative of σ(x) is as

$$= \sigma(x)\,(1 - \sigma(x))$$

## *D. Activation Functions*

We are utilizing the sigmoid function from different activation functions. Purpose of activation functions is to contain output within a specific range. Sigmoid output ranges between 0-1.

$$Sigmoid\ function = \frac{1}{1 + e^{-x}}$$

## *E. Notations*

- $x_{ij}$ = i-th input to unit j
- $w_{ji}$ = weight associated with ith input to unit j

1. $Net_j = {}_{ji} X_{ji}$
2. = output computed by unit j
3. $t_j$ = target output for unit j
4. σ = sigmoid (the activation) function.

### *1) Error Gradient for Sigmoid*

$$\frac{\partial E}{\partial W_i} = \sum_{d \in D} (t_d - o_d) o_d (1 - o_d) x_{i,d}$$

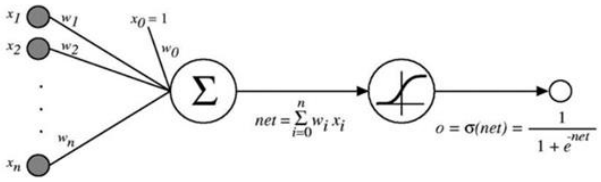
### *2) Error for Hidden Node j*

$$\delta_j = o_j (1 - o_j) \sum \delta_k w_{k,j}$$

## *F. Back Propagation Algorithm*

Weights initialization
Until converges, Do
For each training set Do

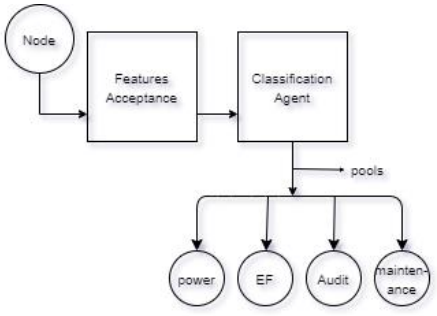
- Input to network and compute outputs.
- For each output unit k
- $\delta_k = o_k (1 - o_k)(t_k - o_k)$
- For each hidden unit h
- $\delta_h = o_h (1 - o_h) \sum w_{h,j}\, \delta_k$
- Update each network weight
- $w_{i,j} = w_{i,j} + \delta w_{i,j}$




## *G. Node Pool*

To ensure system-wide security, fairness as well as to achieve interoperability and scalability, Libonomy system comprises first of its kind pooling architecture. The pooling system of Libonomy comprises of following parts:

- Power pool
- Exploit Finding Pool
- Audit Pool
- Maintenance Pool




Pool Classification

*The classification agent at pool level classifies the nodes in to respective pools based on their features extracted. At the time of joining the network, features are extracted from the node and are transferred to the classification agent which in return assigns the respective pool to the node.*

*Note: It is to be highlighted here that Aphelion protocol also includes the capability of banning the node permanently if any kind of unusual activity is exposed by the algorithm. In that way not only the node's assets will be lost but also it won't be able to even participate in any network which runs on Aphelion protocol.*

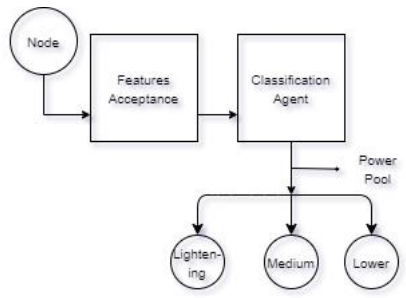
The pools enable clustering mechanisms, introduce a consensus protocol and make it compatible with other blockchain systems.

### *1) Power Pool*

In the power pool, nodes are differentiated into three different classes namely Lightening Nodes, Medium Nodes and Lower Nodes. With the inclusion of pools and their subclasses Aphelion protocol gives the blockchain an ability to accommodate any kind of node to join the network and participate in validating the transactions. The system can accommodate the commercial systems and gives them the ability to participate in mining/verification. To ensure the maximum throughput and to save the protocol from negative effects of low performant systems- the sub-classes are introduced. To participate as a lightning node, the node must be high performant and must meet the minimum requirements.
*Note: The requirements for such systems will be updated on GitHub WIKI of Libonomy.*

So, based on the threshold or the specifications of the nodes the node will serve as either the lightning, medium or lower node. The hierarchy of the nodes can be represented as,

*Lightening>Medium>Lower*




Power pool Classification

*The classification agent at power pool level classifies the nodes in to further three sub-classes of nodes i.e lightening,medium and lower nodes. The nodes are classified on the basis of their performance. Which helps the consensus protocol to distribute the requests to the nodes based on their performance*

To ensure that each node participates in transaction verification/mining, the protocol includes the process of randomization with time slots. Each node is assigned a respective time slot in which it will participate invalidating the transaction received. Later, it will be moved to the maintenance pool which serves as the backup or reserved place, where nodes are placed in an idle mode for a limited time slot.

### *2) Exploit Finding Pool*

The exploit finding pool serves as the security layer for the blockchain and to save the blockchain from any kind of exploit that could disrupt the whole system. Many blockchains have faced the issue of security vulnerabilities and as the whole blockchain is designed to serve as an autonomous system so Aphelion security layers mitigate all these issues by the inclusion of the EF pool.
*Note: The requirements for such system's will be updated on GitHub WIKI of Libonomy.*

In the initial phase of the security algorithm, it is programmed to gather information and data of the nodes. The algorithm ascertains which of the network services are available on nodes such as both the UDP and TCP ports are being scanned.
The algorithm also scans to find any known vulnerabilities on the targeted system regarding the services it is currently running. This enables in keeping track of the number of services available on each port. The algorithm will also scan an entire operating system for any known vulnerabilities and weaknesses, for any software configuration problems and will try to block the services it has been blacklisted with. The algorithm will also try to discover weaknesses and audit the network activity of the system and determine hosts, services, fingerprinting and the firewalls of the system.
The nodes in the EF pool work collectively to verify and index the blocks included by the power pool. The security algorithm constantly runs and overlooks the nodes and their operations in the EF pool. When any vulnerability is exposed by the system the decision agent is reported and the necessary steps are taken by the system to secure the system beforehand. The dataset is constantly being provided to the decision agent which predicts the behavior of the nodes in the complex scenarios. Exploiting attempts are carried by the algorithm on the isolated copy of the current blockchain, to ensure if there is any kind of vulnerability present and calculate the steps that can be taken by the system for it to behave as it is intended to.
*Note: It is to be understood that in the stable release of Libonomy 2.0 the current pen-test algorithm is to be upgraded with the inclusion AI agent.*

### *3) Audit Pool*

Due to inclusion of multi-pool architecture and the decision agent, which serves as the key part in managing the overall operations of the system, there is a possibility that invalid data can be provided to the agent which in general can even break down the whole blockchain. So, to overcome this issue, an audit pool is included in the system which serves as the audit layer for the blockchain. The audit pool overlooks the overall working of the pools in the network. To join the audit pool, the node must satisfy the threshold to be regarded as a trustful node i.e. contribution, verifications, rating and pool votes. In order to join the network node will also be subject to stake in the protocol so that no malicious activity can be carried out by the nodes. Only then the node will be able to join the pool. There is also a possibility of another assumption that a node secretly behaves as a trustful node but after joining the pool it can feed invalid data. To mitigate this issue, the Aphelion protocol includes randomized round trip and multi-layered gossip mechanism. Through this process, nodes are randomly appointed to participate and no node, at any instant, gets to know whether or not it's the one being appointed. The algorithm is programmed to feed the decision agent with auditing data without the node's knowledge, in this manner the information from the node is extracted anonymously. After the round has been completed, the node's logs are updated with the metadata and the rest of the report exists only with the decision agent.

To make a decision, votes are collected by the nodes, but to reduce cost and to be fair, the votes are also cast without the knowledge of the nodes. The multi-layered gossip includes the nested data approach, where the node's data communicated is indexed over the swarm. By extracting the history of the nodes' index and unmarshalling the data, the protocol calculates the node's acceptance on the data and feeds the calculated value to the decision agent, where the rest of the steps are taken by the decision agent. This ensures privacy, fairness, truthfulness and security; after the round, the extracted data is useless because the algorithm is also programmed with the redundancy removal.

After the decision agent takes the necessary steps, audit nodes are programmed to broadcast the rules over the whole network, the firewalls and state-machine of each pool is updated with the new rules. This may include the increase in threshold, the removal of the node or the banning of the node from the whole network.

### *4) Maintenance Pool*

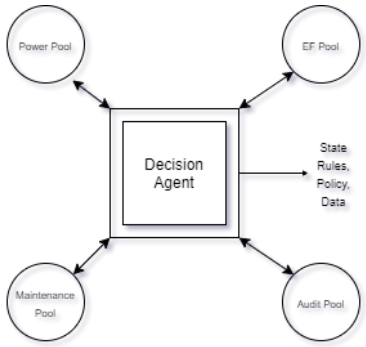
The Aphelion protocol includes the AI agents that collaborate in maintaining the network. The subclass approach makes it easy for many nodes to leave and join the network periodically in a public blockchain system. Along with the decision agent, EF pool and Audit pool there is an assumption that the algorithm will be able to expose the malicious nodes in the network beforehand and they can either be suspended or permanently banned from the network. If the node leaves the network and some important operation is being carried out by the node also than the node from the maintenance pool can be used as a replacement in order to keep the resources intact. For this purpose, Aphelion protocol also includes the maintenance which could also be termed as the reserved pool for the nodes.

Generally, it could be termed as the node's resting place. In the maintenance pool, nodes are kept to serve as a replacement for any other node in any other pool, based on the whole size of the network. Being a part of the maintenance pool doesn't entirely refer to it becoming the permanent pool for the node. As the Aphelion protocol resolves the issues of fairness, trustfulness and decentralization so each node in the network is given a chance to contribute in the network, for this purpose the protocol includes the mechanism of randomized round trip and time slot. Whenever a node has processed some amount of transaction and the time slot assigned has been finished then the node is replaced with another node in the maintenance pool. Now this replacement and movement doesn't make the node to be kept waiting for multiple hours based on the network size but instead, the node is replaced with other nodes in the threshold of 2 ~ 3 seconds or more, based on the threshold in the network. According to the analysis carried out on computer performance, it has been concluded that making the system to conduct complex operations consistently can reduce its efficiency and life span. The mechanism ensures steady system performance, to save the nodes from long and high computation, to reduce node cost, to ensure system reliability, increase the life-span of Aphelion protocol with round trip and time slots and ensure high throughput.

*Note: It is to be highlighted here that the number of nodes to be kept in the maintenance pool depends upon the network size and threshold rule applied by the decision agent.*

## *H. Decision Agents*

.




**Decision Agent And Pools**

Decision agent is bidirectionally connected to the pools of the network where they all the rules, policies are communicated by decision agent, then its the responsibility of respective pools to implement the policy and follow the rules or apply the rules. The data communicated by the pools include the audit report and the data set creation for the retraining in the future.

## *I. Consensus*

Aphelion uses AI agents, to utilize the benefits of other consensus engines such as the approach of voting and validation. The protocol is empowered with the approach of digital voting mechanism i.e. virtual voting. To facilitate robust communication among the nodes, Aphelion utilizes the mechanism of multi-layered gossiping with FULL-SYNC and META-SYNC approach. In Aphelion, the nodes follow the mechanism of carrying out the validation collectively in the power pool so that it can verify the blocks, i.e. list of transactions. Using the multi-layered gossiping, the nodes hold the capability to instantly transmit any information received from its peers to the whole swarm. The nodes are identified by their public key in the network and this key pair also serves as the purpose of reward. The validation process in the pool is carried out in multiple stages and the higher pools validate the proposed blocks. This process is followed by the mechanism of virtual voting.

The casting of votes by the nodes can in general increase the cost and can affect the performance of the blockchain in peak conditions. For this reason, virtual voting mechanism is being utilized in the protocol partially. To achieve optimal performance of the voting mechanism, to reduce the network latency and to achieve the highest throughput in complex conditions over the network, the protocol make's use of multi-layered gossip. In multi-layered gossiping, the nodes are subjected to include the indexing of their swarm history, i.e. the data index carried out for validation purpose, peer/gossip tree etc. Using this information, the voting algorithm accounts the votes of the nodes without their knowledge. The voting mechanism follows the approach of anonymity, and removes any chance of a malicious attack in the voting round. During this period, the nodes aren't required to use their computing power to cast a vote which greatly reduces the cost and improves the performance in the network.

It is to be noted that during this whole process, when the nodes validate the blocks, a voting tree is prepared, the voting result is achieved and the addition of blocks is completed. At the end of their cycle, the dataset containing the path of each step followed by the algorithm and results of the whole process is prepared which will be utilized by the protocol's automation agent i.e. Consensus Agent.

*Note: In the early stages of the research and development consensus agent wasn't included. The consensus agent is still being looked upon at the research and development level.*

## *J. The Consensus Agent*

The purpose of the consensus agent is to serve as the autonomous decision-maker for the consensus in regards to transaction verification and block addition in the blockchain. It serves as an autonomous agent; the basic requirement of the algorithm is that it should be trained on a large amount of dataset to make a valid decision. Due to lack of work conducted in the blockchain community, another layer in the autonomous agent is also included which could save the protocol from taking any wrong decision, with the lack of dataset. The purpose of this layer is to ensure that whenever any decision is taken by the algorithm it doesn't match the pre-conditions of the path highlighted. In this way, the nodes at the higher pool, instantly start to verify the proposed blocks so that no invalid block is added in the blockchain.

The autonomous agent utilizes the dataset that the algorithm is trained to analyze which is the process of nodes validating transactions and adding blocks in the blockchain. For this purpose, it is trained with the knowledge of similar paths and pre-conditions to verify and validate. When a transaction with a known path or similar path is received by the agent, the autonomous agent decides to either add the block or pass it to the power pool if no similarity index is present. This is based on the highest similarity ratio without wasting resources in the validation and verification process. In a block, there are thousands of transactions carried out which, in most of the cases, are similar to one another based on their data and size. However, they do differentiate with one another based on hash, nonce and other information. So, to help the protocol to accommodate the throughput of thousands of transactions in a second, the autonomous agent plays a great role in this regard with the help of a high level.

*Note: Consensus agent initially will run on the un-supervised machine algorithm. The specifications of the algorithm will be released in the upcoming version of the*

*paper. The system will be subjected to include swarm intelligence (SI) based on the prototyping of the module.*

### *K. Clusters Chain*

To give the protocol the ability of interoperability and interact with other blockchain systems, the protocol suggests that other blockchain systems, built on top of Aphelion, are clustered and can interact with one another. Libonomy serves as the initial core and on top of it, at the blockchain layer, the systems utilizing the Aphelion are clustered into different regions. These clusters of blockchains run parallel with Libonomy and the Interaction Channel will transmit the information among different clusters and cores. At the time of cluster creation, each cluster can use its dedicated pool or it can utilize the other pools open to the community. It can use its nodes in these pools to overall increase the performance of all the clusters present. For blockchains to interact with one another it is necessary for every system that all communication flow must be carried through the Interaction Channel. These interaction channels route the traffic between clusters and the core. Whenever Aphelion is being implemented in another blockchain, the protocol manages the configurations for the interaction channel, structuring of the pools and the consensus.

Interaction channel serves to not only enable the future blockchain implementations to be used as an interoperable solution but can also serve the current systems to inject the interaction channel into their systems via our application layer and carry out the communication to or forth Libonomy.

The final version of the interaction channel of Aphelion is planned to include the routing agent as well. The purpose of the routing agent is to serve to make autonomous decisions on the routing of the traffic among different clusters. It is assumed that if the Aphelion is currently running 10 different clusters of the blockchains and on each cluster, thousands of transactions are being carried out, which includes the cluster-specific and cross-cluster transactions as well. In this scenario, the router will be overwhelmed with the amount of traffic and can reduce the overall performance of the blockchains, in terms of interoperability. To improve the situation, the router should be smart enough to make autonomous decisions instead of consuming the resources of the systems in making complex decisions manually. Therefore, routing agents help in overcoming this issue. The routing agents and all other agents, present in the protocol, are meant to be decentralized and these agents are programmed to behave in a decentralized manner. The routing agents will be trained/programmed with the networking models and will be fed with the datasets prepared by the nodes in the network. This will increase the accuracy of the algorithm in decision making. The routing agents and protocol core routers are programmed to work collectively in traffic routing in the blockchain.

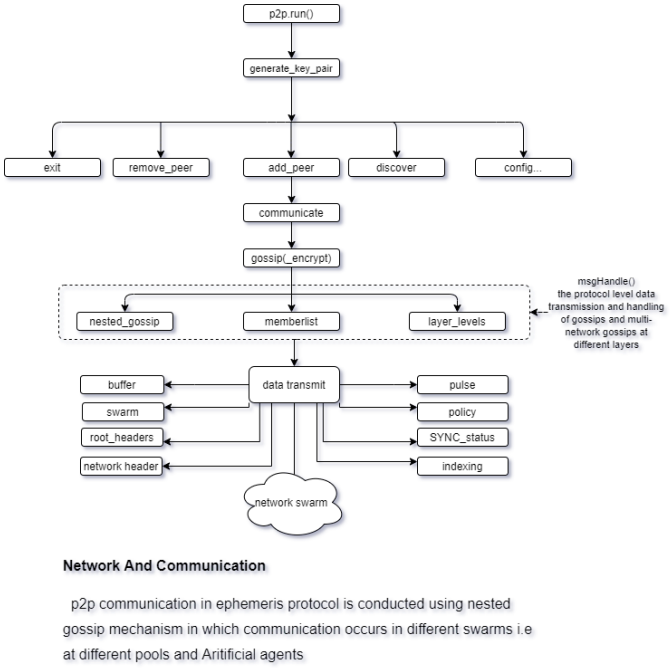
*Note: It should be noted that the need for agents arises due to the support of clusters in the blockchain. There are complex decisions involved, for which artificial intelligence is regarded as one of the most optimal solutions. Prototyping of the model is to be carried out for the unsupervised learning, based on the results received, the protocol will be tuned in that manner. It should also be noted that the algorithm can be retrained based on the state of the blockchain because when the protocol expands there is a need that each algorithm should be retrained with the new dataset.*

### *L. Network and Communication*

For the Aphelion protocol to run, as it is intended or programmed to be, it is necessary that the network layer of the protocol also supports fast communication and can scale over time.

There is a high communication among the nodes for the machine learning models and nodes can communicate raw data among the network. The whole protocol can be badly affected, if the uncompressed data is communicated over the network and the same channel is utilized by the nodes. It puts stress on the network and this could be regarded as the bottleneck for the whole operating system. To solve this problem, at the network layer, p2p utilizes the multi-layered gossip swarming with compression mechanisms, in this part nodes interact with one another through the swarm and share the compressed data in the streams of packets. The algorithm is programmed to distribute the messages in two different mechanisms to ensure optimal distribution of resources takes place and to reduce the network latency. The node can either be **FULL-SYNC** and **META-SYNC** node in it. Whenever the nodes participate in communicating the data over the swarm instead of relaying the complete data, nodes hold the index of the data. The resultant data is then compressed and shared in the swarms. This ensures that the data size is kept at the minimum possible level. The node's status is checked in the swarms, using the heartbeat mechanism and pulse-based communication at each interval. This status can determine the node's synchronization state. Instead of distributing the complete root history, nodes are communicated with the latest state, only if it lies in the **META-SYNC** state. The node, instead of relying on its network resource, can utilize the system resource to re-index or re-construct the data, if need be. Pulse based communication gives the nodes the ability to carry out the communication in byte streams manner. When the node communicates over the network, a digital signal is broadcasted by the node, containing the

streams of metadata. This metadata helps to re-index and reconstruct the required data if it is lost. By utilizing the multi-layering mechanism, it can pave the way for maintaining integrity and achieving fast communication on the network.




*Note: The gossip communication involves the multi-layered gossip or the nested gossip mechanism, which means that the gossip is layered not only at the swarm level but also at the node level. The protocol is being implemented in Libonomy and even can allow any blockchain solution to utilize our network channel in their applications.*

## *M. **Use Cases***

### *1) Decentralized Exchange*

Decentralized Exchange or distributed exchange runs directly on the blockchain instead of any centralized server to increase the security of the assets and to save the exchange from any hacking attempts. Previous centralized exchanges have faced the issues of security vulnerability, either external or internal, losing millions worth of assets. Vast majority of decentralized exchanges are based on atomic cross-chain transactions. These exchanges allow the users to directly make a transaction by writing on both different blockchains that they are utilizing. Users can directly trade Bitcoins or ether by writing transactions on both the blockchain, without the need of interacting with any matching service. This service doesn't require the blockchains to be interlinked with each other but users need to be active at the time of the trade. Similarly, other exchanges aren't based on atomic cross-chain but run directly on their dedicated blockchain. The main problem is interoperability, which doesn't allow the trade to occur across different blockchains. The other issue with atomic cross-chain, centralized exchange and blockchain specific exchange is scalability, interoperability and compatibility.

To facilitate the communities and blockchain systems, Libonomy allows the exchanges and blockchains to overcome these hurdles. Libonomy includes the application layer at the top of its core which can help in not only creating decentralized exchanges but also in building their cryptocurrency and run their dedicated cluster for that purpose. To interact with the existing blockchains, the application layer is configured to carry out the transaction among different ledgers. Along with Aphelion, the exchange written on top of the application layer is scalable as well. The application layer is configured with a blockchain interaction layer which helps to carry out the transaction among different clusters running on Aphelion. To enable the community to take advantage of Libonomy, the first implementation of its application layer will include Libonomy DEX. Libonomy DEX is built on top of the Aphelion application layer, carrying out transactions with Aphelion based blockchain and non-Aphelion based blockchains, such as Bitcoin, Ethereum, EOS etc.

### *2) Crypto-currency and tokens compatibility*

Aphelion protocol can utilize the ability to chain multiple blockchain clusters around itself and support other blockchains. Aphelion protocol can replicate the non-Aphelion blockchains in separate isolated clusters based on their consensus protocol such as PoW based Ethereum, Bitcoin and others.

This enables compatibility for crypto-currency along with Libonomy and its descendants, to write interoperable smart contracts. These tokens can be used directly on the Libonomy system, leading to smart contracts that can be written in a more secure and interoperable fashion. The smart contracts can be written directly on the Libonomy and users can either use them directly on Ethereum or with any Aphelion powered blockchain. Users use their Ethereum smart contracts in Libonomy after the smart contract is submitted to the protocol through its channeling mechanism which will translate and compile the smart contract for Aphelion. After the transaction is added to the block, it can be used directly on the descendants of the Aphelion-based system or on the system it is originally available in. The transactions can be carried out among the blockchains directly through the application layer. It can write the transaction to the specific blockchain network and its clusters through the routing and interaction channel.

### *3) Hybrid Blockchain*

Libonomy blockchain is programmed not only to serve as a public blockchain but can be utilized as a private blockchain. Organizations require a private blockchain which can serve them to interact with other public or private blockchains, write smart contracts and increase its scalability with the growth of the organization.

To facilitate this, the Libonomy allows its blockchain to be configured for the private sector. The blockchain is programmed with the ability to allow developers to utilize the application layer to configure the nodes and write business logic on top of it. The application layer for the private blockchains is programmed with the capability to write a smart contract. It is also programmed to carry out peer and application configuration. Developers do not have to rely on the core of the system but they can utilize pre-programmed configurations that are designed. They can serve complex environments and accommodate organizations that require complex structure for their product, for e.g. supply chain, ERP systems etc. It is completely the developer's decision whether they want to write their core configurations to satisfy their use case.

The application layer gives the freedom to write business logic to private blockchain interacting with public or private blockchain that are based on Aphelion protocol, with certain policies for communication. They can utilize the blockchain Interaction channel and cluster routing.

### *4) Smart Contract*

Libonomy facilitates the developers to build smart contracts not only on public blockchain but also on private blockchain. There are times when organizations need to run their own permissioned and isolated blockchain to remove any kind of fraudulent activity, maintain system integrity and safe-keep the privacy of all records. The difference between the normal smart contract and Libonomy's (X1-Contracts) is in the performance, transparency, decentralization and security it offers. Public blockchain contracts are decentralized but they are still weak due to faulty written code, poorly conducted audit of the code and defective virtual machine compiling, thereby exposing them to hacking attacks. For the past years, many smart contracts were exposed and millions of dollars were lost. This gave rise to improved standards to build smart contracts even then new problems arose. Aphelion Machine has removed this by empowering its compiler with the ability to conduct an audit of the code before its release to save the developer from any kind of loss. The auditor and compiler work collectively, whenever the code is compiled on the virtual engine, this triggers the auditor to run its pen-test agent to crawl the code and look for the vulnerability by running multiple pen-test techniques.

## *N.* ***Libo Coin (LBY)***

Libonomy's primary asset is Libo Coin (LBY), each verifier and validator can utilize the Libo Coin. Similar to Ethereum's Ether, Libo Coin can be used to pay the transaction fee. Additional rewards are provided to the nodes by the protocol based on the contribution made in the network for distributing the resources on behalf of the network, facilitating the pools and Aphelion agents.

### *1) Reward mechanism and Penalty*

To ensure system-wide security, Aphelion is based on the architecture of the pooling system, where each pool is serving for the block verification. The pool also ensures the security of the whole protocol, in terms of any security breach, loopholes, malicious blocks, invalid blocks and for audit purposes. To encourage the nodes in all the pools to act positively and get rewarded with extra incentives, based on their contribution in the network i.e. their commitment, contribution of resources, reporting and audit, the nodes are awarded with Libo Coin from the reserved supply. To prevent the feature from being abused, the rewarding agent is being fed with the report of the node's history verified by the higher pools of the network to know whether the node is eligible for the reward or not. EF pool and Audit pool are awarded based on operations carried out, the number of exploits reported, based on the quality of audit results obtained and prepared, the incentives increase.

The nodes in the audit pool are bound to submit the evidence for the node's malicious activity to look for any vulnerability scan operated by the EF pool and the pool can report it with complete evidence. This leads to the rewarding agent being programmed to ban mechanisms to make sure that the system rewarding mechanism is not abused in any way or any node behaves maliciously in the whole system. It is important to make sure that the audit pool is not reporting invalid information to gain reward from the protocol, the protocol already ensures this by **ROOT VALIDITY** check. In the **ROOT VALIDITY** check, the report submitted by the pool is verified from the reporting pool. This operation is carried out anonymously, none of the nodes in the pool know this audit and proof window. Based on proof check and number of validators, banning is taken place by the audit pool after they are generated by the decision agent in its **NODE PENALTY RULE** policy. The rule is applied among all the pools and the node is removed from the respective pool. The node is marked and removed from the protocol permanently, all the assets of the node are taken by the protocol. This is possible only by using the node's digital signature and the **BAN MARK SIGN.** The **BAN MARK SIGN** will make it impossible for the node

to be unable to connect to the network in any way in future.

## IV. CONCLUSION

We have defined a blockchain that directs the blockchain industry into a world of advancements and improved innovation. It moves the blockchain into a direction of scalability, interoperability, decentralized and a fair distribution of resources so that it can be compatible with the existing blockchain technology. The protocol designed and explained in this paper creates an overall system which is highly rewarding and cost-cutting for its users. The use-cases of our blockchain will expand depending on the industry that utilizes it. The most significant innovation provided is the Artificial Intelligence that is based on the core of the protocol and the unique pooling system. Our blockchain and its cryptocurrency will empower billions of people and ensure an easy global payment for everyone. We have given a document keeping in mind the nature of users, economic incentives and the process of interaction with the system. Our paper discusses the strength and limitation of our blockchain which will yield to further development in future.

## V. REFERENCES

[1] Back et al., "Enabling blockchain innovations with pegged sidechains", Oct. 2014. URL: https://blockstream.com/sidechains.pdf

[2] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. International Conference on Machine Learning (ICML), pages 3478–3487, 2019. URL: https://arxiv.org/pdf/1902.00340.pdf

[3] Copeland and H. Zhong, Tangaroa: A Byzantine Fault Tolerant Raft, Apr. 2018, [online] Available: http://www.scs.stanford.edu/14au-cs244b/labs/projects/copeland_zhong.pdf

[4] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al., Hyperledger fabric: A distributed operating system for permissioned blockchains, 2018, arXiv preprint arXiv:1801.10228.

[5] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, Madars Virza, Zerocash: Decentralized Anonymous Payments from Bitcoin, proceedings of the IEEE Symposium on Security & Privacy (Oakland) 2014, 459-474, IEEE, 2014. URL: http://zerocash-project.org/paper

[6] Gavin Wood. Ethereum: a secure decentralized generalized transaction ledger. 2014. URL: http://gavwood.com/paper.pdf

[7] Herlihy, Maurice. (2018). Atomic Cross-Chain Swaps. 245-254. 10.1145/3212734.3212736.

[8] Hoskinson, Charles. (2017). Why are we building Cardano? A Subjective Approach. URL: https://cardano.org/why/assets/WhyCardanoEN.pdf

[9] Huang D, Ma X, Zhang S. Performance analysis of the raft consensus algorithm for private blockchains. IEEE Trans Syst Man Cybern Syst. 2019;. https://doi.org/10.1109/TSMC.2019.2895471

[10] Eyal, A. E. Gencer, E. G. Sirer and R. van Renesse, "Bitcoin-NG: A scalable blockchain protocol" in , Oct. 2015, [online] Available: https://arxiv.org/pdf/1510.02037v2.pdf

[11] J. Kwon, and E. Buchman, A Network of Distributed Ledgers, Cosmos, 2018. URL: https://cosmos.network/cosmos-whitepaper.pdf

[12] Jentzsch, C. (2016). Decentralized autonomous organization to automate governance. Retrieved August 25, 2017, from https://github.com/slockit/DAO/tree/develop/paper

[13] Li, En & Zeng, Liekang & Zhou, Zhi & Chen, Xu. (2019). Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing. URL: https://arxiv.org/pdf/1910.05316.pdf

[14] Liang J, Li L, Zeng D (2018) Evolutionary dynamics of cryptocurrency transaction networks: An empirical study. PLoS ONE 13(8): e0202202. https://doi.org/10.1371/journal.pone.0202202

[15] Loizou, N., & Richtarik, P. (2016). A new perspective on randomized gossip algorithms. 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP). URL: https://arxiv.org/pdf/1610.04714.pdf

[16] Lombrozo, E., Lau, J., & Wuille, P. (2015, December 21). BIP 141: Segregated Witness (Consensus layer) by CodeShark · Pull Request #265 · bitcoin/bips. Retrieved from https://github.com/bitcoin/bips/pull/265

[17] Lukas Gelbmann. BLS cosigning via a gossip protocol. https://github.com/dedis/student_19_gossip_bls , 2019. Accessed: December 27, 2019.

[18] Nakamoto, Satoshi. “Bitcoin: A Peer-to-Peer Electronic Cash System.” Bitcoin, 2008. URL: https://bitcoin.org/bitcoin.pdf

[19] (n.d.). Retrieved July 22, 2020, from https://www.namecoin.org/

[20] Nxt community. Whitepaper: Nxt. http://wiki.nxtcrypto.org/wiki/Whitepaper:Nxt , 2013.

[21] OmniLayer. (n.d.). OmniLayer/spec. Retrieved July 22, 2020, from https://github.com/mastercoin-MSC/spec

[22] Thaddeus Dryja Joseph Poon. The Bitcoin lightning network: Scalable off-chain instant payments. 2015. URL: http://lightning.network/lightning-network-paper.pdf

[23] Buterin, Ethereum white paper, 2013. Available online: https://github.com/ethereum/wiki/wiki/White-Paper (Accessed 2 April 2018)

[24] Watanabe, Hiroki & Fujimura, Shigeru & Nakadaira, Atsushi & Miyazaki, Yasuhiko & Akutsu, Akihito & Kishigami, Jay. (2016). Blockchain contract: Securing a blockchain applied to smart contracts. 467-468. 10.1109/ICCE.2016.7430693.