WHITEPAPER

Version 0.3

Hugo LALIN

contact@virgocoin.io

VIRGO : a high-performance cryptocurrency for the creation of decentralized applications.

ABSTRACT

Global adoption of cryptocurrencies is difficult to consider in the current state, given the complexity of the market and the blockchain trilemma: scalability, security, and decentralization.
We offer an innovative payment solution based on a distributed register which in essence is a directed acyclic graph (DAG) based on a unique protocol in which the different nodes of the network confirm the transactions independently of each other by following pre-established rules, without consensus and decentralized.
This architecture will simplify the market by allowing large numbers of applications in various sectors to operate under Virgo, thus streamlining the ecosystem and therefore the ease of use of all these solutions: One cryptocurrency for a large number of uses.

TABLE OF CONTENTS

INTRODUCTION

Since its emergence, blockchain has demonstrated that it can allow the creation of a more fluid economy with no trusted third party. This makes it possible to improve many sectors of activity where the need for trust is an issue, such as finance, data management, or still supply chains. However, this technology, which should already be a standard in these sectors hardly, settles because of its complexity. Indeed, there are currently as many different blockchains as there are projects, and we think that this draws a large part of its advantages from technology: While the blockchain must make it possible to streamline exchanges, we find ourselves at constantly having to own and exchange a large number of currencies depending on our use, which is anything but fluid; and at the same time we have to trust a multitude of different blockchain technologies and teams behind which is also contrary to the very purpose of technology, as evidenced by an upsurge in projects being in fact scams.

This increase in the number of solutions is partly due to fundamental problems related to what has been called the "blockchain trilemma" namely : scalability, security, and decentralization. Until now, new technologies have not tried to solve this trilemma but rather preferred to adapt their solutions to the sector they were targeting.

To resolve these persistent problems, VIRGO has chosen to create a new architecture thanks to a Directed Acyclic Graph (DAG), which aims to solve scalability issues with existing public distributed registry technologies. DAG-based distributed registry technologies are showing signs of being particularly able to overcome the scalability limitations inherent in network blockchain-based payment systems. Indeed, while in blockchain-based networks, a larger scale has undesirable effects on network usability, in DAG-based networks, the opposite is generally true: greater use of the network results in improved fluidity of the network.

The fact remains that DAG-based cryptocurrencies such as IOTA do not have success in overcoming this trilemma because they are generally less secure and decentralized than blockchains. In VIRGO, there are deterministic criteria for knowing when a transaction is considered to be final. This is where VIRGO intends to distinguish itself from the traditional infrastructure of distributed registers since this peculiarity confirms transactions without the need for consensus while successfully being decentralized.

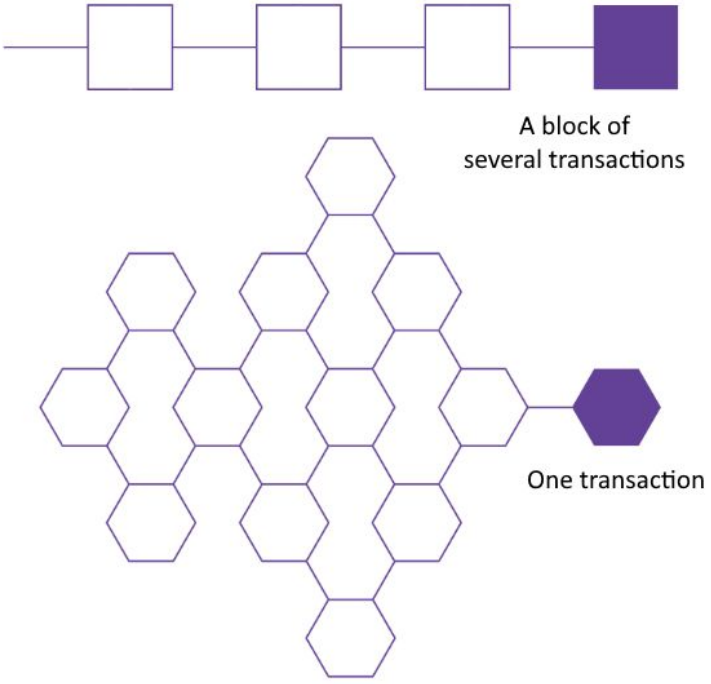
The final objective is to allow decentralized cloud storage applications such as VPN and IOT to benefit from instant transactions at a very low cost so simpler to use, and much more environmentally friendly.

This Whitepaper will detail VIRGO's solution to ensure compatibility between all users around the world and create an ecosystem that enables real-time transactions and data sharing for ease of use for both professionals and individuals.

I – PROTOCOL'S OVERVIEW

1 / Operation

Unlike most cryptocurrencies, VIRGO is not based on a blockchain, but on an acyclic oriented graph (Or DAG, for Directed Acyclic Graph), which can be considered as a two-dimensional blockchain.



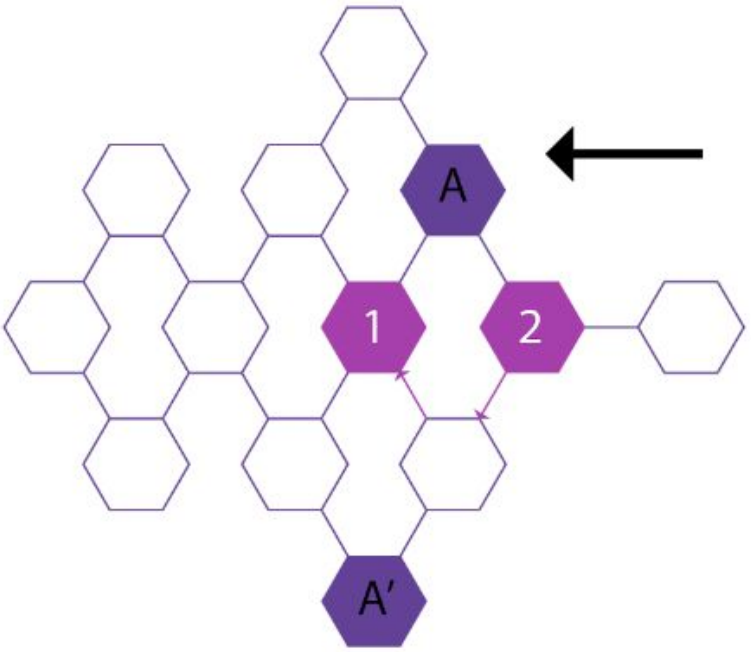
As can be seen in the figure above, the main difference is that each vertex (understand block) can be linked to several parent vertices (appeared before) instead of one on a classic blockchain. In a classic cryptocurrency like bitcoin, group transactions by blocks allow to give an order to them. The blocks being generated at intervals of regular time via mining and each referencing the block before it, all forming a chain (hence the term blockchain) We, therefore, know that a transaction took place before another by looking at the order of their respective blocks, this makes it possible to "do the accounts" in the correct order and avoid having a user who would use the same money multiple times

Blocks should be spaced and small enough for the majority of computers forming the network have time to receive the most recent, otherwise, two parts of the network may receive a new different block, causing two different versions of the blockchain, we then speak of « Fork ». This limits the number of transactions per seconds can be managed by the network and if this limit is reached the transactions with the highest fees are preferred, causing an explosion in the cost of these (Bitcoin has for example already reached costs amounting to more than $ 70). Moreover, transactions are not instantaneous:

we must wait for at least one new block to be generated (often several for security reasons, it all depends on the merchant), which can take a few seconds as 40 minutes for Bitcoin.

With VIRGO, the graph allows us to get rid of the blocks: each vertex is now a single transaction and the order of it is done naturally because each transaction refers to one or more previous transactions (this is shown in the graph, see Figure 2), so they no longer need to arrive simultaneously on all computers on the network.

But using a DAG creates a new problem: transactions do not necessarily have an explicit order.



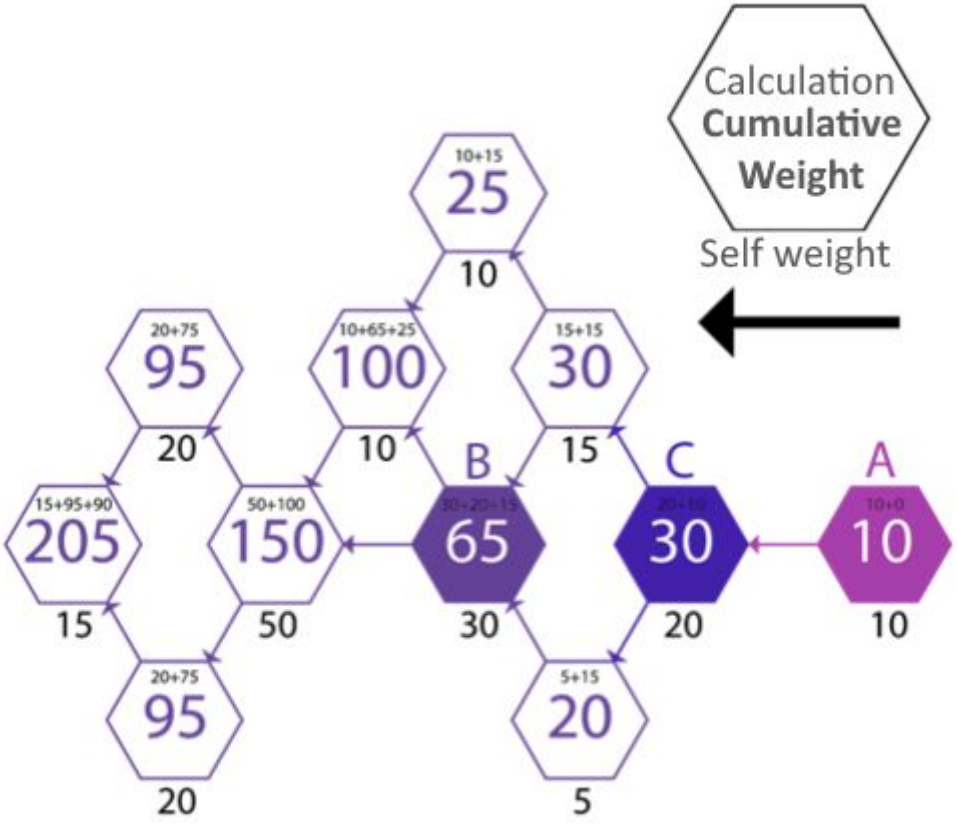
For example in this figure, we can start from transaction 2 to go to the 1 is going up the graph, we therefore directly know that transaction 1 has been issued before. But there is no path between A and AT' going up only, so we can not know which transaction was issued first

If AT and AT' don't try to spend the same funds that is fine but if this is the case then it is necessary to decide and invent an order to it: this decision cannot be taken randomly because we would get a price range and we can't just refuse both transactions otherwise anyone could cancel a payment and get their money back by issuing a new transaction that competes with the old one.

The solutions used by other DAG cryptocurrencies are either to make sure that creating a transaction is taking too long for someone to do this timely handling (This is the case with IOTA), requesting proof of work as an example, or to designate trusted third parties who will decide (case of ByteBall).

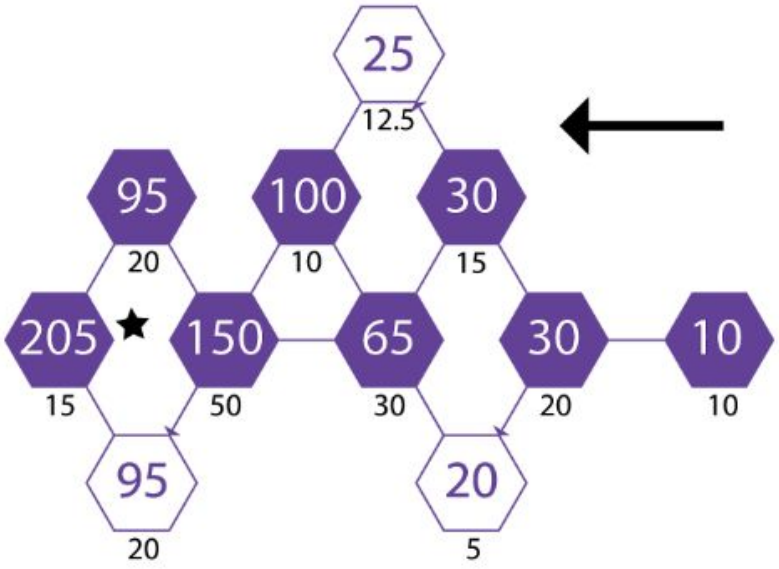
These two solutions do not suit us: the first one is only secure if the network receives several transactions large enough that no one can beat it, which is not necessarily the case in normal times (IOTA currently uses a centralized master system for deciding because it does not receive at all enough transactions).
The second one is based on trust in these third parties, which in both cases goes against the very principle of cryptocurrencies.

Since the problem is to place transactions in time, it seemed to our logic of working from a changing factor over time: the amount of money transferred over the network. Each transaction on the network represent costs, for example, 0.5% of the total. Thanks to these fees we will be able to determine the weight of a transaction, which corresponds to the costs of the latter but also the costs of the transactions above it :



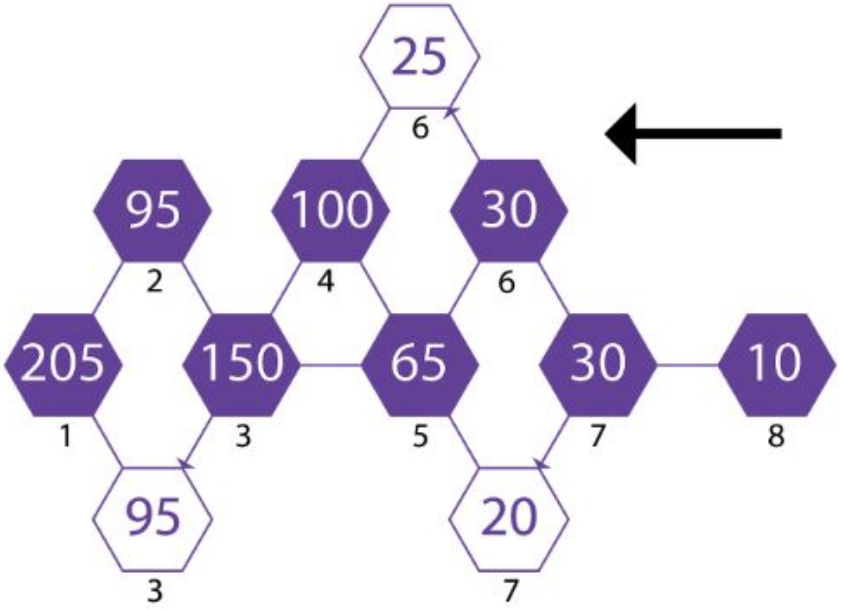
***A**: The transaction transmits its weight to its parent*
***B**: The transaction transmits its weight to its most recent parent and therefore indirectly to its other parent*
***C**: The transaction transfers half of its weight to each of its parents*

The weight of a transaction therefore increases more or less over time fast depending on the network load rate.
Once the weight of the transactions is defined, we will start from the genesis transaction and choose their most weighty child, then their most weighty child, and so on. This will give us the main transaction chain that is the heaviest, which will, therefore, be called the trunk.



**** If two children have the same weight, we choose the one with the smallest hash (the number identifying the transaction) while waiting for one of the two to stand out***

From this trunk, we will be able to give a branch number to all transactions, which simply corresponds to the number of the closest child trunk transaction:

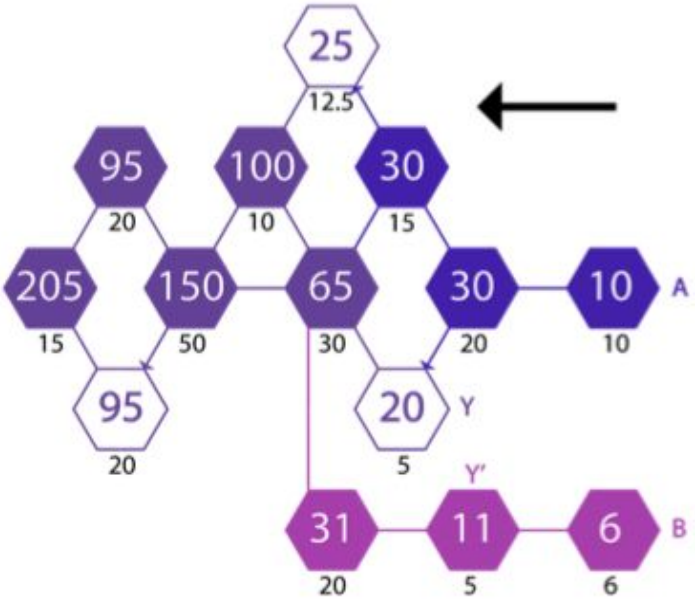


This branch number will allow us to decide on transaction competitors: we will choose the one with the lowest number because its trunk transaction happens before that of the other, and if the transactions have the same branch number then they will all be rejected.

2 / Security and attack possibilities.

Malicious actors seeking financial gain or simply wishing the death of the system could attack VIRGO, like all decentralized cryptocurrencies. In this part, the possible attack scenarios are discussed and the preventive measures are taken by the VIRGO protocol.

2.1 / The stability coefficient



One of the main point of Virgo's is its ability to manage double-spending because they are a consequence of the network's ability to rearrange itself in the event of temporary separation. Indeed, to succeed a double expense, an attacker must be able to change the trunk after receiving his service. To do this, he must create a series of transactions B who is going to come and attach himself to a point of trunk older than the target transaction, which is here Y. B must contain a concurrent transaction to Y (here Y ') and weigh heavier than the piece of the trunk it seeks to replace (here A), then Y ' will become a member of the trunk and be preferred to Y. This operation, therefore, has a cost amounting to the weight of the piece of the trunk to be modified and becomes useless when the weight of the trunk exceeds the amount of the transaction because the attacker would lose money there.

From this observation, we can determine a stability coefficient for each transaction.
This coefficient ranges from 0 to 255 and results in the following equation:

Coefficient (2) = min (X / (Y * m) * Z, 255)

X = weight of the transaction
Y = transaction amount

Z = smaller coefficient of stability among the inputs.
m = safety multiplier

Therefore, a merchant only has to wait for this coefficient to reach a value that looks good before taking payment into account.You will notice the addition of a "safety multiplier": this allows you to adjust the rate of increase of the stability coefficient according to the user's needs. The higher this coefficient, the longer it takes for the coefficient to reach its maximum value. This of course possibly adds a delay to transactions but it remains perfectly acceptable, it can be calculated as follows:

Deadlines = (A * m) / R * M * F

A = transaction amount
m = safety multiplier
R = number of transactions per second on the network
M = Average amount of a transaction
F = Fee rate

So for an average transaction and fees of 0.5%, it takes 200 transactions per second on the network so that the delay is one second, which is little, considering future decentralized applications that we intend to allow.

(2) The coefficient is between 0 and 255 because it is a range of numbers that take up little space to store (1 byte)

2.2 / The risk of transaction overload and the Penny-Spend attack

For any cryptocurrency, an attacker can send a large number of valid transactions of a small amount to overload the network. But in the case of Virgo, the large capacity of the network makes the attack complicated, and it would even help the network on some scale rather than detract from it by making confirmation of healthy transactions faster.
Also, our percentage transaction fee system quickly makes the transaction very expensive, but such an attack is only useful if it is perpetrated for long enough to dissuade anyone from using the network, so it can be dismissed.

3 / Performances

VIRGO, by his design, has no specific limits in terms of transactions by seconds. This means that its performance only depends on the optimization of its code, the computing power of the machine, which executes it, and the fast Internet network, like blockchain-based cryptocurrencies, which, for the most are limited by the size and speed of transmission of the blocks.

The current experimental implementation gives the opportunity to manage up to 10,000 transactions per second on an Intel I5-4570 processor using 70% of its capacities (The remaining 30% is used mainly by the transaction generator and the operating system, which was Windows 10 in this case).

When adding a transaction, the majority of the computing power is used to apply modifications on the graph (generally adding the weight of the transaction and setting up to date of confidence index), it is a task that boils down to simple operations and repetitive can be done in parallel. However, a "classic" processor is designed to be versatile and perform complex tasks, it is therefore not very efficient in this case. But a graphic processor is precisely designed for simple and repetitive calculations, which can be run in parallel: This is why graphics cards are preferred in cryptocurrency mining or deep learning where they are up to 1000x faster than their counterparts. We can therefore expect a huge gain in performance by favouring graphics cards.

The second point of performance improvement is at the level of verification of signing a transaction which is a mathematically heavy process and which must, therefore, be as optimized as possible, but also flawless because it is on this one that all cryptocurrency is based on. We are currently using an open-source library (Bouncy-Castle Java) which is recognized as being reliable but which is unfortunately not very efficient.

The use of a more efficient implementation of cryptographic tools, as well as the use of a system of mass verification (batch verification) could also greatly increase the performance of our solution.

3.1/ Speed of confirmation

Following the processing performance achieved on the VISA network, the average amount of a transaction is $ 80 and the average number of transactions per second is from 1700. Now imagine that you are having lunch in a fast food restaurant, you have 25 $ and want to pay using Virgo:

Transaction times = 25/80 * 1700 * 0.001 ≈ 0.2s, which is very fast.

Of course, the payment will take a little longer because your device must create and distribute the transaction, but once the transaction is effective, it will reach its maximum stability coefficient almost instantaneously.

Now you want to buy a new car with Virgo, it costs $ 40,000:

Delays = 40,000 / 80 * 1,700 * 0.001 ≈ 300s or 5 minutes, acceptable level for this type of purchase.

Last example, micro transactions. You use for a decentralized application including a file-sharing service.

The pair that provides you with the file you want to ask for a total of 0.1 cents, divided in half or 0.05 penny per half of the file:

Delays = 0.005 / 80 * 1700 * 0.001 ≈ 0.03ms, which is ridiculously little.

Even with vastly fewer transactions per second or even smaller ones, VIRGO will always be perfectly suited to micro-transactions and therefore decentralized applications. VISA network statistics are unlikely to be matched quickly, to reduce delays at the start of the network, it would be easy to adapt transaction costs by passing them for example to 0.5%, or to generate transactions in permanence with development funds, funds which will be largely recovered by mining the milestones.

4 / Recommendations

It should be noted that complementary technologies are designed to make effective the VIRGO protocol in all the services offered. Furthermore, "at-will" peer-to-peer subnets. It is clear that if we want to bear a certain number of decentralized applications as large as possible, it is unthinkable to route all messages through the same network. This one would of course be quickly overloaded and would collapse under the weight of very heavy traffic to manage. A more reasonable approach would be to split the whole into networks and this, with the greatest accuracy. The main network, that of VIRGO, would manage transactions financial, and each application would have its network or even several in which pass the data relating to it, all deployable instantly. In this way, they would only pass through the various networks useful information and relating to their networks, distributing traffic, and making the whole more resilient.

To speed up the establishment and strengthen new networks, it will be possible for peers to send messages looking for new connections via other networks they are part of, provided their peers agree to relay these advertising messages (simple measure against abuse). In that sense, if an accident happened to cut the connection of a peer to a small network, it will certainly be able to reconnect by finding new peers through larger networks and not impacted.

Another problem with P2P networks is the latency. Indeed, if the members of the network are chaotically interconnected, so according to the configurations the messages can go back and forth around the globe, greatly increasing the propagation times of these same networks. If we arrange for everyone interconnects with the closest members geographically (and therefore with which latency will be the lowest) then the message propagation time will be longer, uniform, and generally shorter. Moreover, in the case of geographic targeting, the latency will mainly depend on the physical distance between the start and the target, rather than the pseudo-chance that chaos creates.

II - AREAS OF APPLICATION OF VIRGO.

1 / As a currency

Virgo is in essence a general digital asset. It is associated with an identity, an account, and the person who holds this account can transfer it, destroy it, lend it, use it as a guarantee ... as he pleases.
Making money that can be exchanged with other currencies will make it easier to use daily, and its large processing capacity will make trade easier

2 / As decentralized applications

With a decentralized economic system supporting a large scale, it becomes possible to create different places of exchange of resources between people. For example, everyone has a certain amount of unused storage space on their devices; With Virgo, it would be possible for anyone to rent this storage space to those who need it and be paid in real-time for it.
Operating without a data centre, the solution would be more ecological, secure, and would bring more confidentiality than conventional cloud solutions, while being much cheaper.
This is also the first decentralized application that we want to set up. (a document detailing its operation will be available soon.)

But the concept does not stop there and can be extended: to bandwidth by creating, for example, a decentralized VPN market, or even to computing power with decentralized virtual private servers.

3 / In the Internet of Things:

The key point in the overall implementation of the IoT is the transparency of the data collected: To be usable, these must be identifiable and immutable to avoid any fraudulent manipulation of an autonomous decision-making system. Distributed registers like Virgo are the solution to this problem: Indeed, any transaction recorded on the register is by nature immutable and linked to a private key.
Any device, for example, a sensor in a supply chain, would have its private key and communicate its records directly through a transaction, thus making sure that the information comes from the right sensor and allowing other devices to make a decision-based.
Virgo is more suited to this task than other distributed registers because
it is capable of large-scale scaling, which is essential if you want to manage millions of devices simultaneously.
Also, our percentage transaction fee system makes it possible to send micro-transactions at a lower cost, making the solution economical.

Finally, our system is designed to run on any platform: Using no proof of work, creating a transaction on Virgo requires only a small computing power, which is essential to embed the solution on devices sensor type.

CONCLUSION

As can be seen by comparing VIRGO to related work, our protocol is an elegant and effective solution to solve the problem of the "trilemma of blockchain "mentioned above. By solving this problem, we allow an ecosystem of decentralized applications easy to use to emerge, and this same simplicity of use will facilitate the adoption of these services.