QCoin

White paper v. 0.2

July 2024

This game has no name
It will never be the same

For informational purposes only.
Please read the disclaimer.

Contents

1 Introduction

1.1 Preface

To fully comprehend the essence and purpose of the document outlined herein, as well as appreciate the value of the solution proposed in it, it is essential to first revert to the basic questions that underpin the foundation of modern finance and economics, namely:

- What is money?
- What is the difference between good money and bad money?
- How do we measure quality of the money?
- What does it take to make good money?
- What does it all mean for digital money?
- What makes this or that cryptocurrency a good money?

According to the generally accepted definition, the primary functions, which distinguish money, are:

- Acting as a medium of exchange

- Being a unit of account
- Being a store of value
- Acting as a standard of deferred payment

Ergo, the qualities of good money, including digital money, and cryptocurrencies, should consist of:

- Being a medium of exchange
- Being a unit of account
- Being a store of value
- Being used for deferred payments

In this document, we will outline our proposal that foresees the development and release of an instrument that can be construed, and act as, good crypto money. In the post–quantum world, good money, as well as good crypto money, will be endowed with the qualities that allow it to transcend the inherent characteristics of money and beyond.

2 Market Overview

2.1 Blockchain in a Nutshell

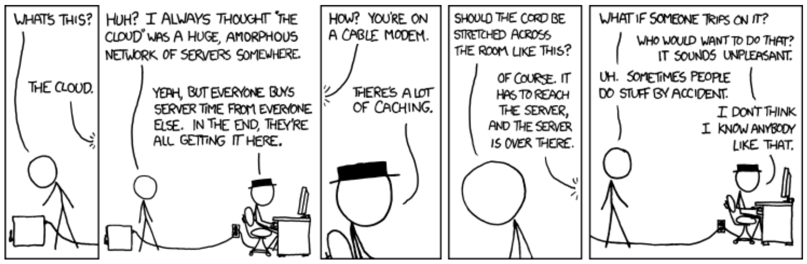
Before going into the technical and marketing details, it is worth defining what is actually a “blockchain–based network”, the way it is used and understood now, which is sometimes different from the historical “Bitcoin–style” definition.

Originally, the idea behind the blockchain was to have a linear list of blocks, with the next block confirming the previous one and, by induction, all blocks before it starting from the first one (*the genesis block*). The blocks contain cryptographically signed, sequential, formally verifiable transactions describing changes in the “state of the world”. Such an architecture allows all the observers to (*) verify that each transaction is following the pre–described set of rules and thus is valid and (**) independently construct the current “state of the world” from the genesis block (“empty world”).

The “state of the world” is sometimes vaguely and intriguingly described in the white papers, giving an impression of something universal, complicated and powerful in itself. Unfortunately, once we get to the bottom of it, the “state of the world” is usually a relatively simple key–value pair. True, in the early implementations (even in the Bitcoin itself) it was possible to attach small bits of data to the transactions, adding history to the “key–value pair” model, but such functionality was pretty much abandoned by the modern blockchains, one of the main reasons being efficiency.

Funny enough, even the idea of the blockchain itself as a “verifiable history from the genesis block that could be used to arrive at the current state of the world” can be abandoned. If there is a mechanism to build trust and guarantee that all participants share the same view of the world and agree on advancing it, such history becomes unnecessary, and is not feasible in some implementations. It is difficult to say it openly, since we believe in marketing being the main reason for that, but if we get to the bottom of the matter, in some modern blockchains, it is effectively impossible. We will discuss this factor in greater detail later.

In summary, *a "modern blockchain network" is essentially a key–value storage with a certain mechanism of trust in a trustless environment.* The rest is either necessitated by the implementation, or added for marketing purposes.


WHAT'S THIS?
THE CLOUD.
HUH? I ALWAYS THOUGHT "THE CLOUD" WAS A HUGE, AMORPHOUS NETWORK OF SERVERS SOMEWHERE.
YEAH, BUT EVERYONE BUYS SERVER TIME FROM EVERYONE ELSE. IN THE END, THEY'RE ALL GETTING IT HERE.
HOW? YOU'RE ON A CABLE MODEM.
THERE'S A LOT OF CACHING.
SHOULD THE CORD BE STRETCHED ACROSS THE ROOM LIKE THIS?
OF COURSE. IT HAS TO REACH THE SERVER, AND THE SERVER IS OVER THERE.
WHAT IF SOMEONE TRIPS ON IT?
WHO WOULD WANT TO DO THAT? IT SOUNDS UNPLEASANT.
UH. SOMETIMES PEOPLE DO STUFF BY ACCIDENT.
I DON'T THINK I KNOW ANYBODY LIKE THAT.

Now, we can think about blockchain in the database terms. Suddenly, all the characteristics start to relate to something we have been familiar with for the last fifty or so years:

- Consistency
- Availability
- Partition tolerance
- Performance
- Capacity
- Request language

An additional twist here being the fact that we can not fully trust all the database servers (but we trust a majority of them) and we need a mechanism to address that issue.

It is fair to assume that from a user perspective, a good blockchain is differentiated from the bad one by those six parameters (provided that the trust mechanism is working), same as the database. We cannot maximise all of them (CAP theorem[1], to mention at least) at the same time, but now we can better navigate in the blockchain world and make a rational choice.

2.2 How To Build A Blockchain That Doesn't Fall Apart Two Days Later

Once we filter away all the fuss and fury surrounding blockchains, we can go one level deeper and ruminate about the specifics of the key–value database and trust mechanism. Specifically, in our brief overview we cover the following topics:

- **Cryptographic protocols**
- **Consensus mechanism**
- **Consensus protocol**

- **Blockchain network**
- **Smart contracts**
- **Interoperability**
- **Tokenomics**

Below is a brief intro on each subject.

2.3 Cryptographic Protocols

While in many cases, choice of a specific set of cryptographic routines seems to be a technical decision rather than an architectural one (and all the major blockchains on the market made it a technical decision with low priority), there is one very special case when it might be a crucial differentiator: post–quantum cryptography.

Perhaps the first attempt to implement a quantum–resistant blockchain was Quantum–Resistant Ledger (QRL)[2]. However, except for the use of post–quantum cryptography, as of now, it is the first generation PoW blockchain with limited capabilities. There are plans to upgrade it, but as of May 2024, the Proof–of–Stake is still in a testnet phase, and smart contracts are planned for the future. As a result, QRL token price lost 90% of it's value since launch going down from from 3.8 USD/QRL in January'2018 to 0.32 USD/QRL in May'2024, ending up with a total valuation around 20mm USD.

Still, the problem is there and some researchers suggest it might be a time bomb under Bitcoin and other cryptocurrencies[3]. Once someone creates a wide enough quantum computer, it could be used to undermine Proof–of–Work in Bitcoin case, forge the signatures in Proof–of–Stake blockchains, and advance time faster that the traditional CPU in Proof–of–History case.

Remark: SHA256 algorithm used in many networks for hashing (and basically integrating the transactions into the ledger) is considered to be quantum–resistant (it's complexity being reduced from $2^2$56 to $2^1$28 operations), so, forging the ledger itself might be difficult. However, an attacker with a quantum computer might be able to pick the private key based on the exposed public key to generate new transactions mimicking a different user. Such scenario may occur if the attacker establishes a malicious validator node that records the raw (unhashed) public keys before sending the transactions to the designated leader. It is worth noting that such a node can just record keys for a while without using them for the attack, so by the time of the attack, it may have a large number of public keys (and, using the quantum computer, corresponding private keys) and can conduct a large number of fraudulent transactions in a very short period of time. Or, alternatively, conduct them from time to time using the largely unused accounts to keep it under the radar.

Summary: the main fork here is traditional cryptography vs post–quantum cryptography. Considering the advances in quantum computing, it seems too risky to ignore post–quantum cryptography completely.

2.4 A Step Back: Byzantine Generals Problem and Sybil Attack

A consensus mechanism for a distributed key–value store is basically a way to agree on changes in the state of the world. If we have a current state of the world, and a blockchain user wants to change it, we need to make sure that the change happens only if all participants agree with it. Moreover, we need to make sure that if there are two users who want to change that state of the world, their changes are either independent, or the order in which they happen is fixed and agreed upon by all network participants.

Going back to our database analogy, and using the database terms, we have to guarantee eventual consistency, and we have to guarantee that this eventual consistency happens. The problem is exacerbated by the fact that we do not inherently trust the servers. We should assume that if the servers have an incentive to manipulate the database in their interests (in the interests of their holders) they might do so. So, on top of the database–style requirements, we should have a trust mechanism in a trustless environment. This is very close to the well–known Byzantine generals problem[4]. There is a large and expanding set of algorithms that solve this problem under various conditions (BFT algorithms).

The usual defence against malicious actors in BFT algorithms would include duplication, reservation and voting. It usually works when the set of nodes is fixed, however, if we have an open network, the nodes could join freely, and the economic incentives are large enough, we might face the situation when it is profitable to get more malicious servers than honest ones and subvert the consensus. This is known as a *Sybil attack*[5].

In summary, there are two problems that should be solved:

- We want to make sure we can agree on consensus and proceed transactions in a consistent manner, even if some of the nodes act maliciously or are unavailable.
- We want to make sure it would be extremely difficult for an attacker to amass a number of fake (in a certain sense) votes to override an honest majority.

It would be easier to start from the second one.

2.5 Consensus Mechanism: Defending Against Sybil attack

There are several well–known mechanisms that prevent Sybil attack, the most relevant in our context being:

- *Identity or personhood validation.* If each participant is easily identified (or at least known to be a unique human being), it could be hard to forge their votes. This mechanism is used in the Idena blockchain[6], however, the current approaches to it either require strong external parties to conduc5 KYC or could be easily circumvented either by technology or by social engineering (for instance via the Turker–style mechanisms).

  Surprisingly, in some of the smaller blockchains, delegation of the initial coin block to large validators is used pretty much in the identity–of–personhood manner: a foundation controlling the emission of the coins signs a contract with a physical world entity representing the validator.

- *Proof–of–Work.* Perhaps best known for its implementation in the Bitcoin network (and other first and second generation blockchains including Litecoin, Ethereum 1.0, Dogecoin etc.), PoW uses a computationally intensive procedure to make sure that it is economically expensive to spawn enough fake nodes to get control over the network. Basically, one has to control more than 50% of the total node processing power essentially meaning that the cheap nodes won't work. The common reasoning for that is the fact that it would be too expensive to get that much processing power for the attack to be profitable or even feasible.

- *Proof–of–Stake.* Another economic mechanism under which participants vote proportionally to their wealth expressed in the amount of native network tokens they hold.

  It was first introduced by Peercoin[7] in 2012, where it was used in addition to PoW, gradually replacing it. Now, perhaps, the only top cryptocurrency sticking to PoW is Bitcoin.

  Under this mechanismm one has to accumulate more than 50% of the native network wealth to manipulate the database. Usual reasoning is that it is too expensive to get that much wealth and, once someone accumulates that, it will not be economically rational to manipulate the consensus, as it is going to undermine the network's reputation. That might lead to abrupt token price crash, which is going to decimate the malicious actor's wealth, leading to losses higher than any potential gain.

In the blockchain, in its original sense, transactions are packed into batches ("blocks") and then participants agree on which blocks are valid and, as a result, on the changes to the database. After a certain period of time the new vision of the world stipulated by the new block is going to be propagated and universally accepted.

Note: it is worth mentioning that while there is a common belief that PoW blockchains are inherently more resistant to "51% attack", this is not necessarily true. While it is fair to assume that it would be hard for a single individual, or even a group, to own more than 51% of mining capacity of a reasonably large PoW blockchain, with the introduction of mining pools, it seems more feasible to control that capacity indirectly. For instance, as of now (July 24, 2024) the three largest Bitcoin mining pools control 68.9% of Bitcoin mining capacity.

Note: while it is reasonable to assume that 51% attack won't generate enough reward within the target blockchain, it was suggested[8] that 51% attack might generate enough reward to economically justify it outside the target blockchain. While this might be difficult for the Bitcoin with it's market cap around USD 1.2 trillion, it is still very valid for any new blockchain. Unfortunately, this attack could be carried out both for PoW and PoS based blockchains. It is an open question if the system could be organised in a way that this attack will not be possible.

A common extension of PoS is Delegated Proof–of–Stake, under which stakeholders can delegate their stakes to validators that act as their proxies. While it gives smaller stakeholders ability to participate in staking and earn return on their capital, it easily leads to network centralisation. Sharding usually decreases centralisation of dPoS network and tokenomics could either increase or decrease it.

Summary: Proof–of–Work is a proven mechanism, however, it might be difficult to market it as it is getting out of favour among the general public, it might be difficult to support it going forward because of political reasons, and it does not necessarily provide better security and trust than Proof–of–Stake. Proof–of–Stake seems to be a better choice, however, it is important to come up with the mechanism that reduces its centralisation.

2.6 Consensus Protocol: Solving the BFT–style Problem

This section focuses mostly on Proof–of–Stake, however, there is at least one interesting idea coming from the PoW space that is worth mentioning.

A PoW blockchain protocol remains secure as long as the block creation time is much higher than the block propagation time. Initially researchers were more focused on fixing the security issues that arise when we try to increase throughput by varying the block creation rate.

The GHOST (Greedy Heaviest Observer SubTree)[9] protocol was introduced to reduce the probability of double–spend attacks when the network throughput gets higher and it starts to fork more often. In this protocol the "longest chain" rule is replaced by the "heaviest tree" rule. While enabling the network to scale in terms of block creation rate (so that the GHOST network can sustain much higher rate compared to the longest–chain one, while remaining secure), it does not increase the speed at a given block creation rate. In fact, it might be somewhat slower than the longest–chain at the same block creation rate. This protocol was initially adopted by Ethereum.

Later on, it was followed by SPECTRE[10] and PHANTOM protocols that further expanded that idea into the graph space.

2.6.1 Proof–of–Stake Consensus Protocols

It has been proven that it is impossible to construct a completely asynchronous protocol that reaches consensus in a distributed system with even a single faulty process, even if we assume a reliable network [11]. So, all the consensus protocols (both PoW and PoS) have to make certain synchronicity or time–keeping assumptions. However, while the first–generation PoW protocols relied on macroscopic time intervals and, in fact, it was the essence of the PoW idea, to make sure it takes time and effort to process the block, PoS protocols soon started the race toward lower latency.

Perhaps the first implementation worth mentioning was Practical BFT suggested in 1999[12], implemented for the NFS file sharing protocol. It was targeted towards local area networks with low latency and a small number of servers. In such environment, it was shown to be tolerant to defection of up to 1/3 of the network nodes into Byzantine coalition, providing correct data efficiently and with low overhead. However, it heavily relied on the timing assumptions behind the network, working well in the LAN environment and struggling once we got into WANs, such as Internet. Moreover, its scalability in terms of number of nodes was limited. As a result, it was considered a good starting point for blockchain/crypto protocols, but not the solution.

There were numerous attempts to improve pBFT in the pre–crypto era[13], however, none of them proved to be an ideal solution needed for a massively distributed high–load system such as a modern blockchain.

Tendermint[14] consensus mechanism makes reasonable assumptions on network latency (it might be high, but there is certain upper limit we know in advance), timers on the nodes and their ability to measure time with a certain reasonable accuracy. Those assumptions make it a BFT protocol quite resistant to double–spending problem. However, while solving the POW resource waste problem, in itself, a single Tendermint blockchain will not be scalable enough, both in terms of transactions per second and in terms of the number of nodes. Terra (ticker: LUNA) and Cosmos (ticker: ATOM) are based on the Tendermint consensus.

HoneyBadger[15] was among the early adoptions of the pBFT protocol specifically built for asynchronous communications (without making the network time assumptions) over high–latency low–quality network, such as Internet. The downside was that this protocol relied on the static node topology, that is: all the nodes should be known before we spin up the system, stale / inactive nodes count towards the Byzantine clique and there is no good way to mark the known malicious nodes and purge them from the system.

AlgoRand[16][17] takes that idea further and allows for the dynamic topology. In Algorand, a random subcommittee is selected at the beginning of every epoch from a large set of validators and that subcommittee proposes and validates transactions for the epoch. The selection mechanism is decentralised and is enacted via the use of a verifiable random function. The problem here lies in the lack of a built–in incentive mechanism. Validators are not (*) rewarded for supporting the network or stacking the coins and (**) are not punished for being offline. (*) might limit the number of validators (as it is economically inefficient to support a box without any reward) and the amount of stacked coins (as it might be economically inefficient to lock coins without getting any interest on them). This, in turn, might make it easier to form a majority during the voting for a malicious actor. (**), in turn, might result in problems with network liveliness and, as a result, the actual number of validators might be much lower than expected, so a malicious clique will have more significant voting power (assuming all of them have incentives to be online, as opposed to honest validators).

Ouroboros[18] was, perhaps, among the first protocols of the second generation of PoS consensus protocols. It solved the "stake grinding" problem where the attacker manipulates randomness in their own favour. It was followed by Ouroboros BFT[19], Ouroboros Praos[20] and Ouroboros Genesis[21], which refined the idea further. This set of protocol is (partially) adopted by Cardano (ticker: ADA) network. Ouroboros Hydra[22] is a notable extension for off–chain scaling. Ouroboros Chronos[23] protocol was suggested to establish global clock.

Ethereum 2.0 relies on "Gasper"[24] consensus mechanism that combines Casper FFG[25] proof–of–stake with the GHOST fork–choice rule.

Summary: there is a large number of reasonably good consensus protocols at the moment. The exact one has to be selected after extensive analysis. Requirements for the candidates include:

- BFT
- Solution to the nothing–at–stake problem
- Solution to the long–range attack problem
- Finality with flexible time–outs for asymmetric shards

2.7 Smart Contracts

While Vitalik Buterin was likely not the first to coin the term "smart contract"[26], he was probably the first who formally described it in a modern way[27] and implemented it within the Ethereum blockchain.

A smart contract, the way it is commonly understood in the blockchain context, is a small bit of software that is stored in the key–value storage (and so it part of the commonly shared "state of the world"). The software is written in a specific language — one well documented and understood by all participants. Blockchain users can invoke that bit of software; once it is invoked, it is executed and the correctness of its execution could be verified by all nodes in the network.

In a way, all blockchains have smart contracts, but some of them are very limited in scope, e.g. "send N tokens from account X to account Y" can be considered a smart contract with the following algorithm:

- Check the amount of tokens in account X.
- If that amount is larger than N, subtract N from account X and add N to account Y.
- If that amount is smaller, do nothing.

However, in the modern blockchains, "smart contract support" refers to a nearly Turing–complete bit of code. If we take a step away and remember that the "blockchain network" is a key–value storage (essentially, a database), we can say that *a smart contract is just a stored procedure working in a certain virtual machine.*

There are various metrics we could use and properties we could look at when we compare smart contracts in various blockchains, but we would stress two:

- **Permissiveness vs narrowness**. The wider the set of operations allowed withing the smart contract is, the easier it is to write the smart contract, but, at the same time, the slower is the execution. And vice versa, the narrower is the smart contract engine spec — the faster it is, but it makes harder for the developers to do something beyond trivial.

  It is worth noting that both too permissive and too restrictive VMs tend to lead to higher security risks.
- **Parallelisation**. As in regular database applications, there is a striking difference in throughput between the mechanisms that allow for the parallel execution on different boxes and the ones that do not.

Summary: smart contract support is crucial for the network success. The blockchain should have the right balance between functionality and speed of the smart contracts, and parallel execution of smart contracts is, perhaps, the only way to ensure scalability.

2.8 Interoperability

Interoperability is the ability to communicate with other blockchains.

Note: it is worth mentioning a special case of interoperability when two instances of the same blockchain communicate with each other. E.g. private sub-chain talks to the main chain or one shard to another. Integral support of this special case is stipulated by sharding, as suggested in the section above.

Interoperability with other networks is an important extension to the base protocol. We see the need for such a mechanism in the network, but not necessary at a core level. E.g. there could be bridges or bridging shards to other networks.

Summary: for a while, interoperability was more of a gimmick or a way to transfer tokens from Ethereum to its own network once it it up and running. However, there is a chance that going forward it might become and important use case. While we can not predict exactly how this is going to work, it is important to have mechanisms in place to be future–proof.

2.9 Tokenomics

Finally, there is a question of token economics (tokenomics). Basically — what regulates token emission, how the new tokens are distributed and when the existing tokens are redistributed, how the transactions are priced, who and when gets and who and when pays the commissions and fees.

2.9.1 Proof–of–Work

In Bitcoin, and the majority of other Proof–of–Work networks, new tokens are generated in the process of mining, which involves finding solutions to computeation–intensive cryptographic problems. The node that finds a solution that meets certain criteria creates the blocks with the next batch of signed transactions and gets the newly–minted coin as a part of transactions; basically it is allowed to add a transaction adding one coin to its wallet. In addition, there might be a transaction fee going to a miner, as users willing to prioritise their transactions can add a small fee to make sure a miner picks their transaction ahead of others to put into the block. As we can see, when the network becomes congested, the fee can be very high prohibiting the use of the blockchain for regular payments.

2.9.2 Proof–of–Stake

In Proof–of–Stake networks, there is a need to lock capital to support network configuration (add trust to the nodes) and voting (add trust to the changes to the ledger). Therefore, there should be an incentive for the capital holders to lock their money in the network. This is usually done via the interest mechanism, whereby new coins are generated in each round and distributed among those who locked their capital for the round, proportionally to the amount locked.

In addition, there is usually a small transaction fee attached to each transaction. Depending on the network, it can be attached either only to user–generated transactions, or to all transactions including the voting ones. In the latter case, nodes participating in the voting process incur significant costs just for supporting the network. This leads to the **centralisation trap**: while it is technologically easy to set up a node, it is economically disadvantageous to be a part of the network, unless the user has a huge amount of capital stacked. This leads to **practical centralisation of the formally decentralised network**.

Summary: delegated / leased Proof–of–Stake with new coins generated as return on capital is optimal at the moment. It is going to take a long time for the inflation forces to prevail, and if we introduce a formula to dynamically adjust creation rate, it should be enough to keep inflation to a minimum. In addition, there should be fees for signing transactions which should go to node owners. If we charge only for the user–generated transactions, it should be cheap enough to maintain the node even with a limited amount of capital.

3 Market Problems

Most of the blockchains out there suffer from one or more of the following issues:

- Centralisation either on the technical, or on tokenomics level, or both
- Vulnerability to cryptographic attacks using the quantum computers

- Bottleneck in transaction processing leading to low throughput and/or high latency
- Unbalanced tokenomics
- Smart contracts that are difficult to write and prove

4 Solution

Our proposal for L1 blockchain is based on the following choices:

- **Cryptographic protocols**: based on post-quantum encryption.
- **Consensus mechanism**: based on delegated Proof–of–Stake.
- **Consensus protocol**: BFT–style.
- **Blockchain network**: asymmetric sharding to parallelise transaction processing.
- **Smart contracts**: asynchronous smart contracts.
- **Interoperability**: support for heterogeneous blockchains.
- **Tokenomics**: inflationary return on capital, flexible transaction fees going to the network nodes (the more complicated the transaction — the bigger the fee).

While there are a number of L1 blockchains endowed with one or a few of those characteristics, the QCoin Network (QCN) is currently the only blockchain in development combining all of them at the same time. We believe this combination is a key to bringing blockchain technology to the next level.

5 Executive Summary

We propose a new third–generation L1 blockchain that combines evolution across several major axes to create a revolutionary result. The key benefit for all participants of the QCoin Network (QCN) extends to a forward–looking approach regarding the development of blockchain technology as a preemptive step towards a post–quantum world.

The inevitable arrival of a quantum computer and a blockchain based on it will render all existing networks obsolete. The QCoin Network, however, will be resistant to such a development, providing consistently high throughput, low latency, and unprecedented security of user private data.

Key QCoin Network characteristics:

- **High throughput, low latency**. Throughput should be high enough to handle a reasonable part of financial transactions. Latency on the main chain should be low enough to provide seamless user experience. In addition, for DeFi (and other potential applications involving automated bidding), latency requirements should go into the sub–second range.

  Our target in terms of throughput is 100,000 TPS sustained, 500,000 peak. Our target in terms of latency is in the range of seconds on the main processing chain, with the ability to construct specialised processing chains with sub–second latency (e.g. for the DeFi market making).

- **Resistance to quantum hacking**. There is more and more evidence that large–scale quantum computing is an achievable goal, and that it could arrive in the next decade. Large–scale general–purpose quantum computers change complexity of many cryptographic calculations from exponential to linear, and many cryptographic schemes (including those that are implemented in the current blockchains) become unreliable.

  We are using post–quantum cryptography throughout the project to make sure the network is future–proof.

- **Tokenomics incentivising decentralisation**. Majority of PoS networks, while declaring reasonably low technical requirements for nodes, actually impose very high capital requirements for breaking even. Basically, it is cheap and easy to set up the node, but running it without attracting a large amount of capital is too expensive. This is a result of tokenomics (rules that lead to coin creation, burn and distribution). As a result, instead of becoming a truly decentralised network, such networks are controlled by a small number of participants.

  In PoW networks, the amount of resources used in combination with block frequency lead to a system where majority of nodes have to participate in mining pools (for instance, as of now the three largest Bitcoin mining pools control 68.9% of Bitcoin mining capacity).

  In the QCoin Network, we specifically target this problem by introducing tokenomics rules that do not give preferences to large stackers.

- **Flexible smart contracts support**. Currently, many blockchains come from "crypto–centric mindset", where a smart contract is a clever way to solve a mathematical problem, Olympiad–style: clever, elegant, cludgy and mind–twisting. As a result, smart contract engineers are among the highest–paid IT professionals, and smart contract security auditors are paid even more. Still, smart contracts are risky business, and now and then, we hear stories about money being stolen through them, just because there was a bug in that Olympiad–style solution.

  We stem from the "programming mindset", and one of the goals of the QCoin Network is to make the API, libraries and virtual machine targeted at regular IT developers, not mathematicians.

  The use of asynchronous smart contract model helps QCN achieve true parallelisation by processing various transactions on multiple nodes.

- **Sharding for parallelisation, cross–chain mechanisms for interoperability**. If there is no sharding, blockchain transaction performance is basically limited to one single computer, the other nodes just confirm and reproduce the results computed on that one. There is that much you can do to optimise their transactions, after that they hit the throughput wall. Currently most of the major blockchains are already struggling, a few highly–optimised are still fast enough, but considering the exponential growth in blockchain adoption, they will hit the wall soon.

To mitigate that, QCoin Network has integrated sharding capabilities and parachain mechanism that are well aligned with the asynchronous smart contract model.

6 Technical Functionality

6.1 Need for Speed

The effective speed of the blockchain depends on three components:

- Speed of the key–value storage.
- Speed of the logic that modifies the key–value storage ("smart contracts").
- Speed of the trust mechanism.

In turn, speed can be measured as throughput (number of transaction in a certain period of time) or as latency (time it takes from transaction initiation to transaction acceptance).

The difference between those measures for "blockchain solutions" is much more significant than in traditional databases. Consider an example of Bitcoin block size change. If we increase size of the block tenfold, throughput will increase tenfold, while latency will remain the same. Alternatively, if we change the mining parameters so the block time will be reduced to 5 minutes and at the same time reduce the block size accordingly, we will reduce the latency by 50% while remaining exactly at the same throughput.

There are, however, "natural limits", something that even the clever engineering will not overcome. For instance, if we say that all the processing is done just on one server, all our network power will be limited to just one box, which is usually a small server or a powerful desktop.

If we dive deep into some of the modern blockchain networks, we will see that at the end all the smart contract processing is done just on one single server, and the total speed of the network depends on that server, with all the other validators (hundreds, thousands) either duplicating the calculations or merely acknowledging the blocks.

The solution to the speed problem that we suggest is based on two principles:

- **Sharding**. It should be possiblt to split the network into a number of pieces operating independently between the rounds of synchronisation.

  Note on asymmetric sharding. Usually, shards are considered to be identical, symmetric, and share the same speed/finality/throughput/security properties. We think it might be interesting to consider an asymmetric approach where the shard characteristics are selected based on the applications running on that shard. E.g. if you we running an order–based DEX we might want to tune up performance, reduce the latency at the cost of finality. If you are running a SWIFT–style transaction processing, you can have higher latency, but finality is much more important. Moreover, there might be individual limits set on the shards, sort of smart contracts operating on top of the whole shard e.g. limiting maximum transaction size for customer–oriented payment processing to improve security.

- **Asynchronous operations**. Key–value storage and operations around it should be organized in such a way that the parallel processing of independent transactions (that is, transactions not trying to write to the same accounts) should be possible.

There is an additional positive side effect of sharding. If done properly, it allows the smaller nodes to be validators as well, increasing decentralisation of the network.

6.2 Blockchain network

The QCoin Network is a multi–chain network with several types of chains:

- **Backbone chain**. The chain that stores the information about the topology of the network together with the final hashes for all the blocks on the processing chains, providing a mechanism for finality. It does not store the regular user accounts or smart contracts themselves, focusing just on the support of the blockchain itself.
- **Processing chain(s)**. Processing chains store the regular user accounts and smart contracts. Blocks store information about the changes in the state of the accounts (via execution of smart contracts). Hashes of the processing chain blocks go into the backbone chain to fix the new state of the world.

  Details of the processing chains (native tokens used in the accounts, VM specs for the smart contracts etc.) can be different, provided that the clients and validators are aware of them. This leads two the next split in types of chains supported by the QCN network:

  - **Standard processing chain(s)**. Processing chains operating over native QCN QCoin accounts under one set of rules. The parameters of the chain are set by the Foundation to balance the latency, throughput and processing costs for the most common tasks.
  - **Custom processing chain(s)**. Chains with custom sets of rules operating either on a dedicated subset of QCN QCoin accounts (e.g. private chain for a DeFi exchange with locked in funds) or on accounts associated with alternative coins tapped into the network. Parameters of those chains could be set by the Foundation or by the owners of those chains to match domain–specific requirements.

6.2.1 Backbone Chain

The backbone chain contains a relatively small amount of information (network topology + combined hashes of the transactions from the toplevel processing chain), and there is no smart contract processing on it, so it remains a single blockchain with no sharding.

6.2.2 Standard Processing Chains and Sharding

Sharding is a technique used by QCN to increase the throughput of the network by processing different transactions on different machines in parallel (as opposed to processing the same transactions on different machines in parallel in the majority of currently existing blockchains).

Basic rules of sharding are:

- The network starts with just one shard processing all the transactions for all the accounts.
- If the load on a certain shard (including the initial shard) reaches certain level, it is split into two shards, each of them responsible for processing half of the smart contracts of the original shard.

- If the load of two adjastent shards goes below a certain level, they are merged back into a bigger shard.

6.2.3 Custom Processing Chains

The rules of sharding for custom processing chains are flexible and could be defined by the creators of custom chains.

6.3 Transaction Model

To support parallel operations, both within the shard and across shards, QCN employs the actor transaction model and an asynchronous approach to accounts and smart contracts.

The majority of the blockchains imply the synchronous approach to accounts and smart contracts (database and stored procedures). Basically, an account is a data–only record in the database and when a smart contract (or, generally speaking, any operation on the database, including externally originated transactions) tries to access an account, the execution is blocked until the the state of the account has been observed or changed.

Imagine it like having one server with the database and the business logic, with all the logic locking portions of the database (or, in many cases, the whole database) for their exeution. And, in turn, each and every access to the database results in the delays on the business logic side. All the nodes in the network just duplicate the job to confirm the transactions (so increase in the number of nodes does not mean increase in performance) or sign the blocks generated by other nodes.

This approach has its own pros and cons. It is simple and easy to understand for novices. It is closer to traditional banking accounts from the outside. It leads to low latency and high throughput while working on one single–core machine and, do a certain extent, it can be interpolated to a single multi–core machine. However, it struggles when we go beyond a single box and try to scale out processing and storage to multiple servers.

In our system, we use a different approach when all the queries to the database (access to the accounts on the blockchain) are asynchronous and do not necessary pause the execution of the smart contracts.

6.3.1 Accounts and Smart Contracts

In the QCN blockchain, there is no distinction between smart contracts and accounts. Basically, an account is a certain type of smart contract that has ID, state, balance, an API to transfer balance to another account, and a hook to call another smart contract if needed.

As opposed to the "passive" accounts in traditional blockchains, our accounts are "active": they could be invoked, aksed to do something, execute logic inside them (e.g. check the balance and state) and reply back. In a way, all accounts and other types of smart contracts are stateful microservices that could be called and executed on the network inside the virtual machine. As with the other microservices, execution of different smart contracts could be carried out on a single box or can be distributed among multiple nodes.

6.3.2 Message Passing and Smart Contract Execution

In the traditional synchronous approach, accessing an account or calling another smart contract is a database query or a stored procedure call. You ask the database to read the state, modity it or run a stored procedure, the request locks the database or part of the database, the state is read or modified (or a procedure is called on the server) and then the execution returns back to the calling smart contract.

Under our approach, accessing an account or calling another smart contract is closer to message passing: the caller forms and sends a message to the callee and, depending on the business logic, at some point, might expect to get an answer, again as a message, from a callee.

Our approach is inspired by the well–known and widely used actor model[28] to concurrent transaction processing. Reformulating a well–known quote by Clinger[29]: we were "motivated by the prospect of a highly parallel distributed computing engine consisting of dozens, hundreds, or even thousands of independent nodes, each with its own local memory and communications processor, communicating via a high–performance communications network".

Web2 can be considered an actor framework with multiple servers communicating with each other, we bring it to Web3.

Under the actor approach, accounts and other types of smart contracts could be invoked on a number of machines in the network (e.g. via sharding) or, in parallel, on different cores of the multi–core box. During its execution, a smart contract does not change the state of other accounts directly. Instead, it could send a message to another smart contract (remember, an account is just a smart contract) that could be invoked on the same box or on a different box.

Sunch anynchronous framework maps well on the sharding mechanism in QCN. If there are no inter–shard communications in the smart contract, there is no need to synchronize the shards while running the contract. If there is inter–shard communication, there is no need to stall smart contract execution.

6.3.3 Message Types

There are three types of messages related to the smart contract execution in the QCN network:

- Intra–chain message: a message sent by one smart contract on the standard processing chain (or the QCN custom processing chain), or one of its shards, to another smart contract on the standard processing chain (or the same QCN custom processing chain).

- Inter–chain message: a message sent by one smart contract on the standard processing chain or one of its shards to another smart contract on the custom processing chain or, reversely, the message coming from the custom processing chain to the standard processing chain.

- External message: a message originating from ourside the chain (e.g. via the oracle mechanism) going to one of the smart contracts on the processing chain, or a message sent by the smart contract on the processing chain, to outside world via the gateway.

6.3.4 Transactions

In the QCN network, a transaction is something that gets into the blockchain as a whole or does not get into the blackchain at all. Basically, a transaction cannot be split between two or more blocks going into the blockchain and cannot be reversed by the means of the network.

A transaction is a reaction of a smart–contract to a message (intra–chain, inter–chain or external) and has several phases:

- The smart contract is invoked and the inbound message is passed to the smart contract.
- The smart contract code is executed. The code can:
  - Conduct calculations inside the Virtual Machine.
  - Send messages to other smart contracts or external gateways.
  - Wait for the replies from other smart contracts or gateways.
  - Change the state associalted with the smart contract (e.g. account balance).
- When the smart contract finishes its execution, all the changes to the smart contract state are finalized, hashed and go into the next block of the blockchain.

The execution of a smart contract on the network results in a certain fee taken out from the balance of the smart contract. The fee depends on:

- The number and size of the outbound messages.
- The amount of code executed by the VM.
- The data stored within the smart contract.

Smart contract execution could be terminated prematurely (before if calls the FIN instruction) in two cases:

- The smart contract runs out of balance and could not pay for the execution. E.g., the user wanted to send money to another account but did not have enough to pay for the transaction.
- The smart contract consumes more than a certain predefined MAXFEE fee that should definitely exceed the expected maximum fee. This is done to make sure the smart contract terminates (this solving the termination problem) and, even if there is error in the code, will not consume the full balance. Basically, a bug in the smart contract will not stall the execution inside the VM and will not drain the account.

6.4 Cryptographic Protocols

One of the key features of the QCN network is integrated support for the post–quantum cryptography mechanisms. While some other blockchains could be upgraded to include post–quantum cryptography in theory, it has not been done yet, and its implementation might pose technical difficulties, such as fixed number of signatures that could be generated with a single key, increase in the size of the signature etc.

Current candidates are the finalists of the round 3 of the NIST Post–Quantum Cryptography Standardisation study:

- CRYSTALS-Dilithium[30]

- CRYSTALS-KYBER[31]
- SPHINCS[32]

Currently, we are carrying out scalability and security research to choose the right implementation.

6.4.1 Cryptographic Protocols for Custom Processing Chains

The backbone chain and standard processing chain, together with their shards, shall rely on the post–quantum cryptography to make sure that attacks carried out with quantum computers will not pose significant security risks and will not undermine the foundation of the QCoin Network.

At the same time, thanks to the custom processing chains, if needed, parts of the QCoin Network could settle for the less demanding cryptographic protocols. For example, one could establish a custom processing chain for a DEX with limited scope in terms ot funds involved and its lifetime, and use a simpler cryptographic procedure to speed up transactions there.

6.5 Consensus Mechanism

QCN uses delegated Proof–of–Stake as a consensus mechanism.

6.5.1 Proof–of–Stake

The nodes in the QCN network (the backbone chain or one of the regular processing chains, while for custom chains rules could be different) should have a certain stake (amount of QCoins) locked on the associated account participaring in forming the consensus. The stake is used to vote for the next block (agree on the next batch of transactions), and check the block (see the protocol section below). It is used both as a way to democratically agree on the next block and a form of collateral, guaranteeing that the node acknowledging the block with the malicious transaction is punished accordingly.

Staked coins earn interest. The fee for transactions on the network is distributed between the nodes running and validating the transaction in proportion to their stakes.

6.5.2 Delegated Proof–of–Stake

Running a server for a backbone chain or one of the regular processing chains involves certain costs (basically, the server should be fast enough to support processing with sufficient speed, and the internet connection should have enough bandwith) and could be difficult for those who want to stake a small amount of coins. The return on the staked coins would not compensate the costs of running the server.

Therefore, there is a way for account holders to delegate the use of their QCoins for staking (and only for staking) via a special type of smart contracts. In this case, the delegated coins are used to boost the vote of the node that they were delegated to for a limited amount of time (that could be extended via the automatic rolls).

If the node does not participate in the network for a certain amount of rounds ("dead" / "malicious" node), the coins are automatically released from staking.

If the node tries to compromise the network (trying to insert malicious transactions, violate consistency, disturb the networking layer etc.), the staked coins go through the process of slashing, which involves part of the stake being taken from the malicious node and distributed between the other nodes in the network. To protect the delegated part of the stake (that is, coins belonging not to the node itself but to those who delegated their coins for staking), slashing first involves the node's own QCoin balance.

As with many Proof–of–Stake networks, if the number of validators gets too large, it becomes increasingly difficult to form a consensus because of the network's restrictions. Delegation of the stake is one of the mechanisms to combat such attempts.

6.6 Consensus Protocol

QC uses BFT algorithm for consensus with a few additions aligned with the proposed sharding mechanism.

6.6.1 Slashing

As with the majority of PoS networks, QCN uses slashing to discourage validators from malicious (e.g. agreeing on the invalid transactions) or careless (e.g. turning off the server) actions.

If a validator proposes or votes for a block with an incorrect transaction, part of its stacked funds is revoked from its account and gets distributed to the community.

If there is more than one validator peforming dodgy actions on the same block, the proportion of the slashed funds is higher.

6.6.2 Dormant Nodes

There could be a situation when the network connectivity or the datacenter fail and a large number of validators become unavailable for a period (e.g. fire in one of the major datacenters or traffic exchanges).

If the number of active validators drops below the certain level in one shard, other validators within the shard distribute the *halt event*, the blockchain pauses and the masterchain validators reorganize the topology of the network.

If it happens to the validators of the masterchain, all the other validators are automatically reassigned to the masterchain.

6.6.3 Two–step Finality Approach and Verification

Under the proposed algorithm, finality is reached in two phases (as opposed to single–phase finality in the majority of BTF based algorithms).

- **First phase: intra–shard voting.** During the first phase, all the validators in a given shard accept transactions, conduct the needed calculations and form the *candidate block.*

  Then the validators within the shard vote and agree on the *candidate block.* If they could not reach an agreement ("fork", "sleeping validators" etc.), they distribute the *halt event*, calculations on the network are paused and other validators step in.

If the validators withing the shard reach an agreement, then the candidate block is distributed to other shards.

- **Second phase: cross–shard verification and confirmation.** During the second phase, a number of randomly selected nodes from the other shards verify the candidate block generated at the first phase and either confirm its validity, or signal a potential issue with it.

  If all the verifiers confirm that the block is valid, it is considered the *final block* and its hash goes into the masterchain. After that, we assume that the finality has been reached, and all the changes would require external intervention.

  If any of the verifiers signals a potential issue, the *halt event* is distributed, all the calculations in the network are paused and all the validators work on the problematic set of transactions to either confirm or decline them.

7 Platform Economics

A large part of coins will be generated at launch and distributed among the founders, the team, initial investors and early adopters. Later, new tokens will be generated as a reward for stacking, and existing tokens will distributed towards node owners as part of transaction fees.

7.1 Initial Distribution

This section reflects our initial thoughts on the subject and should not be considered a final offer.

As in physics light can be viewed both as a wave and as a particle, QC can be viewed as a coin and as a project. As a project, it requires planning and financing. In this section we will discuss the project side.

We believe QC is primarily a public project. As a public project it is:

- Transparent: we disclose both technical and tokenomic detals, and the mechanisms behind them.
- Open: anyone reading this white paper can participate (subject to local laws).
- Public.

We tentatively propose the following initial coin distribution:

- Team: 15%.
- Initial financing: 30% including:
  - Strategic investors.

    First round of financing. Target: to find a few investors committed to the success of QC and secure money for development.
  - Venture capital.

    Second round of financing. Target: to broaden the circle of partners with the investors having relevant expertise and money to support the network growth.

  - Coin presale.

    Final round of financing before the launch of the open market operations and the mainnet. Target: to increase the involvement of the general public.

- Foundation: 32.5%.
- Initial validators support: 12.5%.
- Initial market liquidity support: 10%.

Most of the coins are vested. Specifically:

Target	Free float at start	Time for the full unlock
Team	10%	4 years
Initial financing	5%	varies
Foundation	5%	6 years
Initial validators	0%	locked
Market Liquidity	5%	s.t. Foundation decision

7.1.1 Initial Financing

Initial financing is based on three pillars:

- Strategic investors.
- Venture capital.
- Public coin presale.

7.1.2 Foundation

Foundation is the main governing body of QCN. While the team is responsible for the initial development of the technology and the depolyment of the testnet, after launch, the Foundation is going to be responsible for the success of QCN. On the technical side, the mandate includes the ongoing development of the QCN, support of projects and the ecosystem around it. On the governance side, the mandate includes proposals for policy changes and fine–tuning of tokenomic parameters (discussed in the next sections).

7.1.3 Initial Validators Support

While we believe in decentralization, and the technology behind the QC should prevent any abuse coming from the validators, when the network is large enough, during the first stage, when the market cap and public involvment might be relatively small, we require proof of authority. The validators that decide to participate in the initial validators program should pass KYC checks and meet certain technology thresholds.

7.1.4 Initial Market Liquidity Support

The liquidity support program is targeted at market makers and venues with the aim to gradually distribute the coins to the general public via CEX and DEX mechanisms and at the same time dampen price shocks. Market makers and venues that decide to participate in the liquidity support program should pass KYC checks and commit to certain rules and targets (primarily in terms of smoothing the shocks and preventing market abuse).

7.2 Ongoing Tokenomics

In the previous section we discussed QC as a project. In this section we discuss QC as money.

Going back to the properties of good money, two of those properties depend on and, in turn, define our tokenomic model:

- Medium of exchange. There should be enough coins to make support the economic and financial activities of the actors. Since we believe that crypto, and QC in particular, will gain a larger share in financial transactions in the future, and the world economy will grow, it means that the supply of QCoins should grow accordingly.
- Store of value. If the money (coin) supply grows too fast, exceeding current demand and capacity of the economy using it, inflation kicks in, and the money depreciates.

So, the optimal money supply is something flexible that depends on a number of factors and changes in dynamics.

As a result, we do not share the original Bitcoin vision that the supply of coins should have a fixed (and rather small) upper limit. While it might be good for those who want to capitalize on the asset appreciation, it limits its use as means of exchange. In the past, conventional monetary systems had to abandon the gold standard in part because of that. Nowadays gold remains a viable asset, but not good money.

We also believe that the simple approach to just increase the supply by N% every year, adopted by some of the blockchains, might be too rigid, since it does not reflect real coin usage. How can one say upfront that the amount of coins needed to support the operations is going to increase by 5% or 7% every year?

Note: while crypto tends to show considerable volatility, we believe that it might decrease over time for coins that become more integrated with conventional finance and economy. We can not guarantee that, but we understand the issues here and believe that a combination of short–term liquidity measures and a long–term money supply policy might dampen the shocks.

7.2.1 Transaction Fees

In the QCoin Network, there is a small fee attached to transactions that depends on the number of accounts used (account access fee), processing power required for the smart contract logic (smart contract processing fee) and amount of the on–chain storage used by the transaction (storage fee). The fee, expressed in terms of QCoins, is distributed from transaction originators (that is, users willing to make a transaction) to the nodes that process and sign the transaction within the shard of the processing chain and to the nodes supporting the master chain at the time the transaction was running.

7.2.2 Validators Tokenomics

Validators provide the technological backbone of the network, stakers provide the tokenomic one. There are two ways those QCoin Network participants are incentivized:

- Reward for staking — return on capital. In PoS networks, it is crucial to have certain capital locked for voting and validating the changes to the ledger.

  In the QCoin Network, those who decide to lock their capital for validating purposes receive interest on locked capital in form of newly minted QCoins. This creates incentives for holding the coins within the network and supporting it with voting capital.

  Since the QCoin Network uses delegated Proof–of–Stake, capital owners do not have to support the needed infrastructure themselves, they can delegate their voting power to validators. Their capital remains safe, since they do not make any wallet transfers.

- Reward for running a node — compensation for work. For a decentralized network, it is important to have a reasonably large number of nodes accepting the transactions and modifying the state of the world accordingly. So, there is an incentive for the ones who run the nodes.

7.2.3 Coin Minting and Burning

As we have discussed in the beginning of the section, the amount of coins in circulation is something that should change over time, and there should be a flexible mechanism for that.

Currently, in conventional monetary systems, central banks have that function and, to a certain extent, do it. However, central banks are centralized. We belive that there is a decentralized way of doing it with the Foundation having power and resources to formulate proposals so all the stakeholders could vote and, in a decentralized way, determine the policy.

We assume the QCoint Network is going to initially run on the inflation model, minting new coins to pay the reward for staking. The exact parameters of the inflation process will be subject to the decisions made by the Foundation with the aim of providing enough liquidity for the operations on the network without devaluing the coin.

It is worth noting that the inflation pressure during the first couple of years will be exacerbated by the linear release of the locked coins of the Foundation, the initial investors and the team. At the same time, we expect general public to get more involved and buy QCoins on the open market, thus creating the pressure on the demand side.

8 Marketing Model

The QCoin Network development team will employ a marketing model that relies on promotional tools that have proven to be effective in advancing the popularity of marketplace projects among active user audiences of centralized exchanges, decentralized exchanges and DeFi market services.

The main instruments the project intends to utilize are direct collaborations to ensure maximum audience reach through cross–integration to raise its ranking as a go–to solution connected to multiple services.

The main tools for promotion of the QCoin Network on the market will be:

- Formation of a community of like–minded people, ambassadors and people using the product.
- Transmission of project values and user pain resolution.
- Attraction of users to participate in the development of the product in exchange for rewards, leveraging the incentivization aspect as a bootstrapping instrument at early product stages.
- Conveyance of the practical usefulness of the product via word of mouth. The QCoin Network team aims to create functionality that simplifies user interactions in DeFi and brings them profit.
- Partnerships with Web3 teams, developers and influencers who help develop a community. The QCoin Network team is convinced that when market players are united by a common goal of transparency, decentralization and equality in rights, the industry will be able to develop in favor of the users, not a select group of founders.

Among the most effective instruments to be employed for promoting the QCoin Network social networks and online presence will be:

1. SEO promotion;
2. Keyword search optimization;
3. Advertising via bloggers on social networks YouTube and Instagram;
4. Targeted website banners;
5. Targeted email newsletters;
6. Outreach to Web3 services platforms, project teams, developers, and other market services;
7. Other marketing techniques —- engagement marketing via smart use of content, participatory, hands–on events, and engrossing branding material used for generating project value perception among potential users.

In addition, the development team plans to engage in active participation in leading events, conferences and exhibitions. Apart from constant displays of the QCoin Network in influencer feeds and social media channels, the team will be launching promotions and bonuses for both average users and businesses to maintain their engagement.

The development team is also currently working on establishing working relations with numerous potential partners to ensure the availability of instant applications and use cases for the QCoin Network upon launch, leveraging the partners' user bases.

9 Roadmap

In our view, a new L1 blockchain is not just a mathematical challenge, but also a business one, albeit risky, startup–like and with lots of good engineering under the hood. With that in mind, we thought it might be worth having in this memo our ideas on how we could push the project forward.

9.1 The Organization

QCoin Network [**QCN**] is the L1 blockchain. **QuarkCoin** [**QC**] is the native token of the Quark Project.

The founders form the **QCoin Network Company** that initially develops key technologies, algorithms and software, and holds the IP. Once the network reaches a certain stage, a company will be opened and the latter will delegate further development to the QCoin Foundation. The core company is financed in fiat or established cryptocurrencies by the founders and initial investors who, in return, get the initial pool of QCoins.

The QCoin Foundation [**QCF**] is a public not–for–profit organization that is aimed at developing, promoting and protecting the project. The Foundation has a governance mechanism based on stakes and contribution metrics to ensure that both the ones who decided to give their money and faith, and the ones that decided to give their time and labor to the project, have a say. The Foundation is initially financed by the QCoins public sale, and later by the QCoins that are minted during the regular course of network functioning.

9.2 The Plan

1. **Initial stage**

   The founders write technical specifications and detailed descriptions of the network. At this stage, it is crucial to understand that our goals and propositions form a technically consistent vision. Once this is done, initial investors and developers are invited to join the club.

2. **Private development stage**

   The QCN Company is registered. Money is raised using the traditional venture capital methods, both fiat and established cryptocurrencies are accepted here.

   The core development team is formed. The main task is to come up with the needed documentation (including white papers, as it is fashionable in the crypto world, but not limited to them). Research is carried out, algorithms and technologies are fixed (selection of the encryption algorithms, details of the consensus protocol etc.). The first version of the code is written, most of the functionality should be supported there. The first private testnet is deployed, developers port some of the popular dApps to BQN to see how they behave.

   At the end of this stage, we should have a working fully documented prototype of the QCN, something that we could show and that would inspire people to join The Project with money or effort. It is crucial not to fail here.

3. **Public development stage**

   QCN goes public. At this stage, in addition to the strategic investors, venture capital financing and, potentically, initial coin offering is organized and the QCoins start the limited circulation on Ethereum network.

   The documentation is made public, the public testnet is launched as the alpha prototype (basically everybody is invited to try it for free as clients, to experiment with dApps and see how it feels; however, at this stage there is just a fixed pool of servers, controlled by the QCN Company).

Security audit is ordered. Developers are hired to port dapps to BQN and market their experience to the community.

Agreements with the initial validators and liquidity providers are secured.

QCoin Foundation is founded. QCN Company slowly transfers auxiliary activities to QCoin Foundation. QCoin Foundation is becoming an active participant in the conferences around the world.

All the development is still done within the QCoin Network Company, and all the decisions are still made by a small group of founders.

At the end of this stage coin goes into presale to the general public on the Ethereum network.

4. **Launch**

   Champagne for the believers who invested in the QCoins, sour beer for the rest.

5. **Inflation stage**

   When there are quarks involved, there should be an inflation stage during which the universe expands rapidly.

   During the inflation stage, we expect to see a significant increase in the value of the QCoin and, at the same time, trading volume. Market makers work to guarantee the demand for the coin is met, and that it does not become a bubble. Small portion of the vested coins becomes free–floated.

   During this stage, we expect to encounter risky situations that require a rapid response, that would be easier to do through the small QCN Company and liquidity providers bound by their commitments, not through the Foundation that should target the longer–term goals.

6. **Sovereignty**

   The Quark Network becomes the main network for the circulation of the QCoin. Existing coins are transferred from the Ethereum network to QCN, securing network independence.

   At this stage, we expect to transfer network development and maintenance from the QCN Company to the QCoin Foundation.

7. **Steady stage**

   During the Steady Stage, the QCoin Network Company limits its involvement and works only through its votes and authority within the QCoin Foundation. The QCoin Foundation is responsible for the maintenance, development and amendments to the Project. We expect the Foundation to retain some wealth from the initial stages, but as it evaporates, we expect it to rely more on the network fees.

10 Team Information

11 Disclaimer

Nothing herein constitutes legal, financial, business, investment or tax advice. Neither the authors of this document, nor any of the entities mentioned in this document, their employers, founders, partners or any other involved parties shall be liable for any kind of direct or inderect loss or damage one might suffer in connection with this document.

The reader should seek independent legal advice before committing any action based on this document. The reader has the responsibility to comply with the local laws and regulations while reading this paper and taking any decisions based on the information in it.

Purpose of the white paper. This white paper has only informational purposes and should be treaded as such. In particular, all the technical details described here including but not limited to the QCN network, cryptographic algorithms, blockchain configuration, consensus mechanism, consensus protocol, accounts and smart contract architecture, all the tokenomic details including the initial distribution of tokens, tokenomic model, fees involved, are provided solely for informational purposes and do not constitute any binding commitment. All the information here is subject to change without the prior notice and with no obligation to update the white paper itself or in any form reach out to any person with an update.

Not a public offer. This white paper is not and does not contain a public offer.

Forward–looking statements. All the information in this document is tentative, is not the final proposal, and should not be treated as such. All statements here should be treated as forward–looking, including all the statements about the technical and tokenomical details of the QCN network.

Distribution. No part of this document is to be copied, modified, reproduced, distributed or used in any way without the prior written consent of the authors of this document and the Foundation.

Regulatory approval. This document serves for information purposes only and thus should not be and has not been analysed and approved by any regulatory authority in any jurisdiction.

12 Final remarks

We are open to discussion of potential partnerships.

Quark team

References

[1] A. Fox, E.A. Brewer. “Harvest, yield, and scalable tolerant systems”. *Proceedings of the Seventh Workshop on Hot Topics in Operating Systems (1999).*

[2] Quantum–Resistant Ledger (QRL) whitepaper. *October’2016.*

[3] Aggarwal D., Brennen G., Lee T., Santha M., Tomamichel M. “Quantum attacks on Bitcoin, and how to protect against them”. *Ledger, vol.3, 2018.*

[4] Lamport L., Shostak R., Pease, M. “The Byzantine Generals Problem”. *ACM Transactions on Programming Languages and Systems, 1982.*

[5] John R. Douceur. “The Sybil Attack”. *IPTPS 2002.*

[6] “Idena Concept Paper”. *Whitepaper draft, September 20, 2019 .*

[7] King S., Nadal S. “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake” *Whitepaper. August 19, 2012.*

[8] Joshua A. Kroll, Ian C. Davey, E. Felten. “The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries”. *(2013).*

[9] Sompolinsky Y., Zohar A. “Secure High-Rate Transaction Processing in Bitcoin”. *FC 2015*

[10] Sompolinsky Y., Lewenberg Y., Zohar A. “SPECTRE: A Fast and Scalable Cryptocurrency Protocol”. *Cryptology ePrint Archive: Report 2016/1159 (2016 / 2018 versions).*

[11] Fischer M, Lynch N., Paterson M. “Impossibility of Distributed Consensus with One Faulty Process”. *April 1985, Journal of the ACM 32(2).*

[12] Miguel Castro, Barbara Liskov. “Practical Byzantine Fault Tolerance”. *OSDI.*

[13] Gang Wang. “SoK: Understanding BFT Consensus in the Age of Blockchains”. *Report, July 4, 2021*

[14] Kwon J., “Tendermint: Consensus without Mining”. *Draft (v0.6) whitepaper, 2014.*

[15] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, Dawn Song. “The Honey Badger of BFT Protocols”. *2016.*

[16] Jing Chen, Silvio Micali. “Algorand”. *Whitepaper, published on arXiv in 2016, current version dates back to 2017.*

[17] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, Nickolai Zeldovich. “Algorand: Scaling Byzantine Agreements for Cryptocurrencies”. *Proceedings of the 26th Symposium on Operating Systems Principles. October 2017.*

[18] Kiayias A., Russell A., David B., Oliynykov R. (2017) “Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol”. *Katz J., Shacham H. (eds) Advances in Cryptology – Crypto 2017.* Springer, Cham. 27 July 2017.

[19] Kiayias A., Russell A. (2018) “Ouroboros-BFT: A Simple Byzantine Fault Tolerant Consensus Protocol”. *Cryptology ePrint Archive: Report 2018/1049.*

[20] David B., Gaži P., Kiayias A., Russell A. (2018). “Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain.” *Eurocrypt 2018.*

[21] Badertscher C., Gaži Peter., Kiayias A., Russell A., Zikas V. (2018). “Ouroboros Genesis: Composable Proof-of-Stake Blockchains with Dynamic Availability.” *2018 ACM SIGSAC Conference.*

[22] Chakravarty M., Coretti S., Fitzi M., Gaži P., Kant P., Kiayias A., Russell A. (2021) "Hydra: Fast Isomorphic State Channels." *Financial Cryptography 2021.*

[23] Badertscher C., Gaži P., Kiayias A., Russell A., Zikas V. "Ouroboros Chronos: Permissionless Clock Synchronization via Proof-of-Stake." *Eurocrypt 2021.*

[24] Vitalik Buterin, Diego Hernandez, Thor Kamphefner, Khiem Pham, Zhi Qiao, Danny Ryan, Juhyeok Sin, Ying Wang, Yan X Zhang. "Combining GHOST and Casper". *ArXiv, March 6 — May 11 2020.*

[25] Buterin V., Griffith V. "Casper the friendly finality gadget". *arXiv preprint, October 25, 2017.*

[26] Szabo N. "Smart Contracts". *Keynote speech, IEEE International Workshop on Electronic Contracting (WEC), 2004.*

[27] Buterin V. "A Next-Generation Smart Contract and Decentralized Application Platform". *GitHub, April 9, 2014.*

[28] Carl Hewitt, Peter Bishop. "A Universal Modular ACTOR Formalism for Artificial Intelligence". *Proceedings of the 3rd international joint conference on Artificial intelligence, January 1973.*

[29] William Douglas Clinger. "Foundations of Actor Semantics". *Doctoral dissertation, MIT, 1981.*

[30] National Institute of Standards and Technology. "Module-Lattice-Based Key-Encapsulation Mechanism Standard". *NIST FIPS 203, August 24, 2023.*

[31] National Institute of Standards and Technology. "Module-Lattice-Based Digital Signature Standard". *NIST FIPS 204, August 24, 2023.*

[32] National Institute of Standards and Technology. "Stateless Hash-Based Digital Signature Standard". *NIST FIPS 205, August 24, 2023.*