# LQDEX

## Decentralized, Trustless, Cross-Chain Exchange

Sergey Nikitin
sergey@lqdex.com

Yogesh Srihari
yogesh@lqdex.com

White Paper

June 4, 2018
Version 0.9 (a working draft)

# Table of Contents

# Abstract

In the last few months, the blockchain community has seen a rapid proliferation of DEX, decentralized digital asset exchanges. The advantages of DEX over their centralized counterparts are:

1. Substantially higher security
2. No (or fewer) restrictions on the types of assets that can be traded and who can trade them

But, the modern DEX have major shortcomings. Typically:

- Only single blockchain assets can be traded
- High cost due to miner fees
- Trade execution is slow

As a result, most DEX suffer from low liquidity. Professional traders still, for the most part, prefer using centralized exchanges with larger trading volumes and lower fees, even though it means exposing themselves to the risk of losing funds.

In contrast, LQDEX is a decentralized trustless **cross-chain** digital asset exchange. It has all of the advantages of DEX without the typical shortcomings. The main features of LQDEX are:

- Cross-chain trading
- Low cost, no miner fees
- Trade execution speed comparable to centralized exchanges

LQDEX supports trading of digital tokens across multiple blockchains without counterparty risk. For example, Bitcoin can be traded for Ether. It does not use atomic swaps and does not require modifications to the existing blockchains. The system accomplishes these goals by using the following technologies:

1. Proxy token trading with smart contracts
2. Economically bonded collators providing cross-chain interoperability
3. A purposely built blockchain without miner fees and high execution speed

By solving the challenges of decentralized trading, LQDEX plans to attract institutional traders, which will bring high liquidity to the network.

Digital asset trading is a huge market. At the moment of this writing, the combined **daily** trading volume of all cryptocurrency exchanges is **$10 Billion**.

# The Challenge

Users of modern cryptocurrency exchanges demand risk-free trading, wide range of tradable assets, liquidity, and transparency. The market research indicates that the ideal exchange should have the following characteristics:

1. Is counterparty risk-free
2. Allows trading of assets on multiple blockchains
3. Has substantial liquidity
4. Is inexpensive to use
5. Has fast order execution
6. Is transparent and publicly auditable

Let us consider some of these requirements in more details and the technologies that can be used to implement them.

## Counterparty Risk

The main concern of using regular centralized cryptocurrency exchanges is the risk of losing funds. There were multiple precedents of centralized exchanges losing all or part of user funds. For example, in 2014, Mt. Gox, the Japanese-based exchange, lost almost half a billion US dollars of user funds. [1] The reason centralized exchanges are vulnerable is because they have a single point of failure. If the server hosting the exchange is compromised or shut down users lose their funds.

The risk of losing funds when trusting a third party, such as a centralized exchange, with holding user assets is called “counterparty risk.”

But, the counterparty risk is not just an exchange specific issue. In fact, it is an age-old problem humans had to deal with as early as trading was invented.

If two people traded with each other, they traditionally had to deal with counterparty risk. Let us say human A wants to trade chickens for lambs, and human B wants to trade lambs for chickens. Both are happy with the price the other person is offering. However, there is a problem. One of the two has to give his product to the other person first. Whoever, gives it first, has the risk of the other party not going through with the transaction and running away with both chick and lambs!

While the risk may be acceptable for smaller transactions, for larger deals, people invented a “trusted third party.” The trusted third party is a person or a company that both of them trust. In that situation, both humans give their goods to the third party. Only when both are received, the third party does the exchange and gives the good to the corresponding clients. This scenario has the risk of the trusted third party going bankrupt or running away with both chickens and lambs.

Exchanges, in particular, are more vulnerable to the counterparty risk issues. If an exchange is to have a substantial liquidity, it needs market makers to deposit large amounts of funds on an exchange. As a result, if something happens to an exchange, loss of funds can be substantial.

## Benefits of Decentralization

For the first time in human history, blockchain technology has solved the age-old counterparty risk problem. Theoretically, if done correctly, one person should be able to exchange digital assets with another person, without having to trust anyone.

What makes this possible is design of the blockchain itself. Instead of entrusting funds to a single entity, the asset ledger is held by a large number of blockchain nodes. Even if one node is attacked by hacker, or decides to run away with user’s assets, it would not make a difference, as consensus of the majority nodes is needed to

modify the asset ledger. The only way to compromise a blockchain is to attack the majority of its nodes. But, the nodes are hosted on many different servers, often in different parts of the world. Attacking the majority of the nodes is typically very expensive and impractical.

So, the solution to a trustless exchange problem is a decentralized exchange hosted on multiple servers without a single point of failure. If one server is hacked or runs away with user's funds, the entire system continues functioning and users, at least theoretically, do not lose their funds.

## The Technological Challenge

So, why has not anyone built a proper decentralized exchange yet? The reason is that there is one more technological challenge that have not been solved.

The problem is that it is easy to exchange digital tokens within the same blockchain. However, to exchange tokens generated by one blockchain for tokens generated by another blockchain, the two blockchains need to be able to somehow communicate and send commands to each other. But, a universal counterparty risk-free cross-chain protocol does not yet exit.

## Smart Contracts

Within the same blockchain, users can trade digital tokens risk-free by using "smart contracts." A smart contract is a computer program which executes certain predefined instructions on the blockchain. The benefit of a smart contract is that users do not have to trust each other, when performing a trade or another operation.

For example, a simple smart contract can perform the following instructions:

1. Wait until token X is received from user A and token Y is received from user B.
2. If both received, send token X to user B and token Y to user A (do the exchange).
3. If the condition 1 is not met, give the tokens back to the users (cancel transaction).

## Cross-Chain Interoperability

Unfortunately, the limitation of smart contracts is that they have to be able to manipulate the tokes, so they can only work with tokens of the same blockchain as the smart contract itself. For example, users can easily trade ERC20 tokens on the Ethereum blockchain with smart contracts.

But, what if someone wants to trade Bitcoin for Ether with a smart contact? If there was an algorithm that could accomplish that counterparty risk-free, it would be called a "cross-chain smart contract."

The difficulty with implementing cross-chain smart contracts has to do with how blockchains work. There are two features that all blockchains have in common.

## The Private Key Dilemma

First, to execute a transaction, someone has to sign it with a "private key." If a blockchain instruction does not get signed, it is not getting done, as simple as that. For example, you hold the private key for your Bitcoin wallet and do not show it to anyone. There is no way in the world that any person, computer, or anything else can generate a transaction which takes your bitcoin out of your wallet. If there was, there would not be much point of having Bitcoin, or any other cryptocurrency for that matter, to begin with.

Second, blockchains cannot conceal any information. Anything that gets put on any blockchain is instantly public knowledge. Can

someone develop a blockchain which can store private information? No! It will not be a blockchain then. Blockchains work by allowing independent and unknown parties to host the blockchain ledger on their computers. Anyone can install the Bitcoin core (or any other blockchain core) on his computer and get a copy of all Bitcoin data from the beginning of Bitcoin.

Now, it is clear why connecting two blockchains is difficult. To initiate a transaction on one blockchain, the other blockchain would have to have the private key for that blockchain and sign the transaction with the private key before transaction can be written on the blockchain after miner verification. But, no blockchain can be trusted with a private key, as everyone will instantly know what it is.

## Cross-Chain Trading

Let us see what solutions have been proposed for accomplishing cross-chain trading so far.

### Trusted Third Party

We will codename this solution “Mt. Gox.” It is a privately held server holding private keys for all blockchains it works with. When users send tokens to the server it performs some computations and sends them out to the appropriate blockchains. This is how all centralized exchanges work. This solution is not counterparty risk-free. Even the most secure server can be compromised or taken down.

### Trading within One Blockchain

Some exchanges decided not to bother with cross-chain trades and decided to work with only one blockchain. Examples are EtherDelta and 0X. We will talk about them more in details later. This solution does not work across multiple blockchains. It does not meet the requirement number 2 of the ideal exchange, being able to trade a wide range of assets on multiple blockchains.

### Proxy Tokens with Trusted Third Parties

Some exchanges developed a concept of “proxy tokens.” Proxy tokens are tokens on one blockchain that represents tokens on another blockchain. They are similar to plastic chips in a casino. Each corresponds to an amount of money, but they are not real money. They cannot be used outside of the casino that issued them. The usefulness of proxy tokens is that they can be exchanged risk-free with smart contracts within one blockchain even though they represent real tokens on other blockchains.

However, trusted third parties are still needed in order to convert these proxy tokens into the real tokens, once the user finishes trading and wants to cash out. The exchange that pioneered this model is BitShares. This solution is not counterparty risk-free. The trusted third parties can be compromised and the users will lose funds.

### Atomic Swaps

Atomic Swaps is the technology that does allow some limited cross-chain interoperability. To do a “swap” (an exchange), the two users have to manually sign their transactions with their private keys. A technology layer acts as an escrow, which ether performs the swap or returns funds backs to the users. Because each user holds the corresponding private key, a swap can only be executed when both users are online and sign within a specified time window. It does not allow automated order execution, when one of the users is offline.

The inability to execute orders automatically means that users can not place any type of conditional orders, such as limit orders, on the exchange. The lack of limit orders means the absence of market makers. The absence of market makers, in turn, means lack of liquidity on the exchange. In other words, a user would have to come to the exchange website and wait

for another user to appear in real time, who is willing to trade against him or her. As a result, a traditional exchange with liquidity and automated order execution cannot be built using the atomic swaps technology.

Another drawback of the technology is that it is limited to only Bitcoin clone blockchains or requires modifications to the blockchain core software (a hard fork) in order to work. While this solution comes closest to accomplishing cross-chain trading functionality, it is not universal and not suitable for an exchange in the traditional sense of the word.

## Summary

To summarize, there are only three possibilities of what type of entity can hold the private key. Each has its own pros and cons.

| Entity Type | Pros | Cons |
|---|---|---|
| The private key is stored on the blockchain | Cross-chain transactions are possible | Everyone will see the key and can steal user funds. This is not a viable solution. |
| The private key is stored on the private server (the “trusted third party”) | Cross-chain transactions are possible | The “trusted third party” can steal or inadvertently lose user funds (the Mt. Gox scenario). |
| The user (a human) has the private key | Some cross-chain risk-free transactions are possible | The user has to be online to sign each transaction. Not suitable for frequent trading or conditional orders. Requires modifications (hard forks) to blockchain cores. |

Each solution proposing any type of cross-chain functionality needs to be evaluated based on which entity has the private key. It must fall in one of the three categories above. Depending on which category it falls in, it will have its advantages and potential vulnerabilities.

## Other Decentralized Exchange Projects

A trustless decentralized cryptocurrency exchange with zero counterparty risk is the holy grail of digital asset trading. The company, which implements it, is very likely to achieve a high degree of commercial success. As a result, many startups attempted to tackle the problem. Some called their system a “decentralized exchange,” but as with anything else, the devil is in the details.

### BitShares

BitShares is a decentralized exchange. It uses proxy tokens for counterparty risk-free trading. It also uses Gateways that issue IOUs for trading on their platform [2]. However, the IOUs are not backed by any collateral, so if the gateway is hacked or shutdown, users lose their funds. The zero counterparty risk extends only to trading of proxy tokens. A user, holding a proxy token IOU, is not guaranteed to be able to convert it into an actual asset.

In addition to gateway IOUs, BitShares also has Market Pegged Assets (MPA), backed by a collateral in BitShares core currency, BTS [2]. However, they are not backed by the real asset. The user cannot convert an MPA directly to the corresponding real asset. He or she can either convert it to BTS and then convert BTS to the correct asset at a different exchange, or convert it to an IOU proxy token and convert that IOU to the asset at one of the gateways. Neither of these options are counterparty risk-free.

In short, BitShares cannot execute counterparty risk-free trades. It can only offer counterparty

risk-free trading of various types of native proxy tokens within its network.

### EtherDelta and RadarRelay

EtherDelta and RadarRelay are exchanges, which allow counterparty risk-free trading of ERC20 tokens on the Ethereum blockchain [3, 4]. Since trading is performed with tokens issued by only one blockchain, it can be performed with smart contracts without counterparty risk.

The limitation of these two platforms is that they only work with Ethereum tokens. They do not support Bitcoin for example. Another limitation is that trading on Ethereum network is slow and relatively expensive, as new blocks are produced only once every 14 seconds [5], and Ether has to paid for each transaction.

### ARK

ARK is a blockchain with the ability to execute transactions on other blockchains using its proprietary SmartBridge technology. It has something called “Encoded Listeners” to scan for SmartBridge data and execute cross-chain transactions. According to the ARK’s website:

“The Encoded listener node is a hub for listening to SmartBridge transactions. This transaction hub can be setup and run by anyone, Shapeshift, Changelly or even Coinbase. Anyone that wants to act as a medium to help the network can. And in exchange for providing this service they will be collecting transaction fees for passing data or exchanging currencies via SmartBridge.” [6]

As we know, to execute a transaction on a blockchain, one has to sign it with a private key. So, in this scenario, the Encoded Listener node would have to have the private key stored on it in order to execute transactions. This obviously makes the system vulnerable to hacking of the Encoded Listeners. It is not in any way a counterparty risk-free transaction.

In addition, if “anyone that wants to act as a medium” can set up an Encoded Listener node and send and receive cryptocurrency on behalf of ARK, there will probably be a lot of incidences of Encoded Listeners disappearing after receiving substantial amounts of user funds.

### Atomic Swaps

Atomic Swaps is the name of a promising technology that can allow counterparty risk-free transactions across certain blockchains [7]. It can work on Bitcoin clone blockchains such as Litecoin or Decred using their script language. Atomic swaps can also be performed with Lightning Network. However, the support for atomic swaps with scripting is not universal across different blockchains. It is also not clear which blockchains will support Lightning Network, as it requires a hard fork.

The limitation of this technology is that it only allows of exchanging funds when both users are online to sign the transactions with their private keys. It is more suitable for large OTC trades. It is not as convenient for frequent trading common on cryptocurrency exchanges. Neither, it allows execution of any automated orders, such as limit nor stop-limit orders.

# LQDEX Blockchain

Let us now see how the LQDEX blockchain works. The LQDEX Blockchain (LB) is a proof of stake blockchain. It features fast block creation, and it does not have miner fees. The network can issue native tokens, similar to Ethereum tokens.

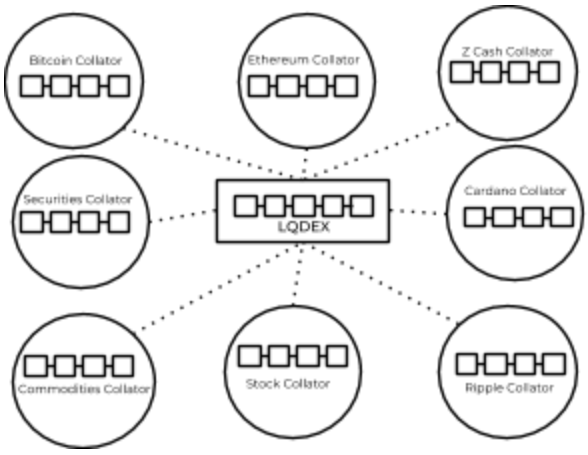


**Figure 1: High-level LQDEX blockchain architecture**

## LQD Token

The main currency of LB is called LQDEX (LQD), similar to Ether in Ethereum. LQD is traded on cryptocurrency exchanges and can be purchased for Bitcoin, Ether, USD, or other currency.

LQD token pays a reward. The LQDEX Network generates profits by charging commissions for trades and withdrawal fees. It distributes profits to LQD token holders as rewards.

## Proxy Tokens

LQDEX also issues proxy tokens, such as lq-BTC or lq-ETH. These correspond to digital tokens on external blockchains. There are always as many outstanding proxy tokens as assets on the balances of the Collators. When assets are deposited, the corresponding proxy tokens are created. When assets are withdrawn, the corresponding proxy tokens are destroyed.. For example, if LB has 1,000 lq-BTC tokens issued, it means that at the moment, there is exactly 1,000 BTC on the balances of all of its Collators.

## Trustless Cross-Chain Trading

LB can natively execute smart contracts with its native tokens, LQD or proxy tokens. All trades are performed as smart contracts using proxy tokens with zero counterparty risk.

Collators extend this functionality further and enable LQDEX Network to execute counterparty risk-free trades across multiple blockchains.

## Collators

A Collator is a server hosting an LB core and a core of another blockchain, such as Bitcoin core or Ethereum core ("external blockchain"). The server keeps the private key for accessing the external blockchain. A Collator performs the following functions:

1. Executes commands it receives from LB.
2. Provides information about external blockchain transactions to LB.

A Collator receives the share of profits from the exchange operations proportional to its LQD collateral deposit amount with LB. One way to think about Collators is as “miners” in a proof-of-work or proof-of-stake blockchain.

Here are examples of commands that can be executed by Collator hosing the Bitcoin core:

1. Deposit X amount of BTC.

2. Withdraw Y amount of BTC to an address.
3. Provide information about a Bitcoin transaction or wallet.

When Collator receives tokens, such as BTC, it stores them in its wallet. Upon detecting the transaction, LB creates new proxy tokens, such as lq-BTC, for the amount of external blockchain tokens received by the Collator.

Likewise, when LB instructs Collator to send tokens, such as BTC, Collator sends tokens from its wallet to the address specified by LB. LQDEX then destroys proxy tokens, such as lq-BTC, for the amount of external blockchain tokens sent by Collator.

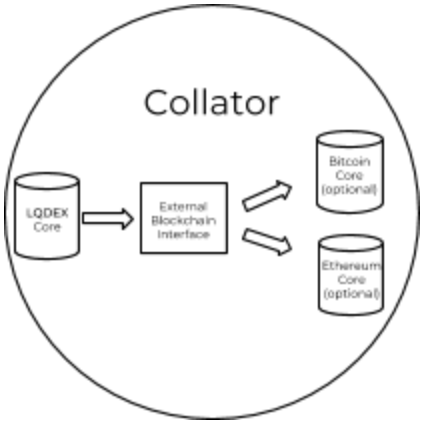


**Figure 2: Collator software architecture**

The LB software combines in one package the LB Core, which records LB transactions, and the Collator functionality. A participant of the network may install the core and only use it for recording transactions or use it for both, recording transactions and providing the external blockchain connectivity. In both cases, the participant would need to deposit an amount of LQD token, which serves simultaneously as both, the voting stake and a Collator's Collateral.

## The Collateral

A Collator must maintain a deposit with LB in LQD in the amount of 200% or more of its token balance. For example, if a Collator has a balance of 100 BTC, it must maintain a deposit of 200 BTC or more in LQD with LB. This deposit is used as collateral. The collateral makes it unprofitable for a Collator to "walk away" with user funds.

One way to think about a Collateral is as a Stake in a proof-of-stake blockchain. See the Proof of Stake vs. Proof of Work section for a more detailed explanation of what Stake is used for. The bottom line is to participate in any blockchain, be that proof-of-work Bitcoin or the next implementation of Ethereum Casper, a participant has to make a capital expenditure, which can be substantial.

In case of Bitcoin, a miner has to invest in expensive mining equipment and electricity lines, the cost of which can run in millions of dollars. In case of a PoW blockchain, such as Ethereum Casper, a miner needs to invest in the Stake. In ether case, an investment has to be made and participation is not free.

Since miners have to invest funds anyway in one way or another, why not use this investment for one additional purpose, which is a Collateral, and entrust these miners to hold a limited amount of user funds, not to exceed half of their Stake in the system? The use of the Stake as a Collateral to hold user funds in addition to voting power is just a natural extension of the PoW blockchain concept.

## Collator Transactions

### Deposit Transactions

There is no counterparty risk for deposit transactions, as the funds have already been deposited to LB as collateral. If Collator balance reaches 200% of the collateral or higher, LB stops sending deposit requests to that Collator

and informs the Collator to increase its collateral to resume processing deposits.

### Withdrawal Transactions

If a Collator fails to perform a withdrawal transaction, LB will first attempt to execute the command with a different Collator, supporting the same external currency. If all Collators fail to perform the transaction, LB will liquidate a part of the Collator's collateral proportional to the amount of the withdrawal. It will send that amount in LQD to the user instead of the external currency. The user will receive an amount of LQD larger than the amount of withdrawal requested. He or she can then convert that amount of LQD to the currency that the Collators failed to send at a different exchange.

### External Blockchain Reporting

Collators report information about transactions on external blockchains to LB. If LB needs information on a specific transaction, it broadcasts this query to all Collators that are known to with with a given external asset, such as Bitcoin.

For example, LB can request information of whether or not a deposit was made to specific Bitcoin address. After the query is made, the corresponding Collators submit their votes on whether or not they think the deposit was made. Once the certain time period passes, LB determines that the deposit was made if the supermajority of Bitcoin collators reported the deposit.

## Transaction Examples

Let us consider some example transactions on LQDEX Network in more details.

### Scenario 1: Account Funding

A user wants to start trading on the LQDEX Network. He first needs to deposit funds to LB. At the moment, he wishes to deposit 1 BTC. To process the deposit, the network will perform the following steps:

1. LB creates a wallet for the user on its blockchain.
2. LB finds the Collator, which supports Bitcoin and has the highest collateral ratio.
3. LB instructs the Collator to commence a deposit transaction for 1 BTC.
4. Collator provides a Bitcoin address to LB, to which the user needs to send his deposit.
5. LB informs the user to send 1 BTC to the Bitcoin address provided by Collator. LB also sets a time window, during which the user needs to perform the deposit.
6. LB keeps querying the provided Bitcoin address to see if a deposit of 1 BTC appears in it.

If the deposit appears:

1. LB creates lq-BTC tokens for 1 BTC. It places the lq-BTC tokens for 1 BTC in the user's wallet.

If the deposit does not appear:

1. If the deposit does not appear during the time window, the receive transaction is cancelled.

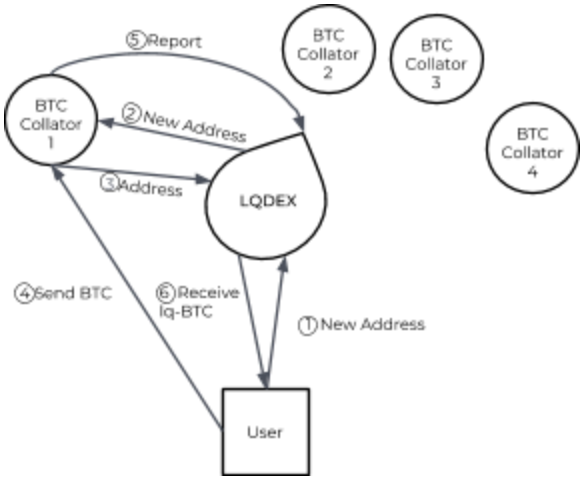


**Figure 3: Deposit transaction flow**

## Scenario 2: Trading

The user wants to exchange his lq-BTC tokens for lq-ETH tokens. LB can natively trade its tokens with smart contracts. Users can trade any internal tokens on LB without counterparty risk.


**Figure 4: Trade transaction flow**

## Scenario 3: Withdrawal

The user finished trading and now wants to withdraw his Ether. At the moment, he has 10 ETH.

1. The user provides an Ethereum address where he wishes to withdraw the funds to.
2. LB finds the Collator, which supports Ethereum with the sufficient balance and the lowest collateral ratio.
3. LB instructs that Collator to send 10 ETH to the address provided by the user (the transaction can be broken up among multiple RBGs).
4. The Collator executes the transaction and return the blockchain transaction number to LB.
5. LB confirms that the transaction number is valid and the transaction has been executed.

If the transaction has been executed:

1. LB destroys 10 lq-ETH tokens and removes them from the user's wallet.

If the transaction has not been executed:

1. LB finds another Collator, which supports Ethereum and has the sufficient balance and continues with step 3.

If no suitable Collators found:

1. LB calculates an amount of LQD held on deposit by all Collators proportional to 10 ETH. Let us say it is 20 ETH equivalent in LQD.
2. LB takes an amount of LQD worth 20 ETH from the Collator's collaterals and sends it to the user.
3. The user can optionally convert LQD to at least 10 ETH at a different exchange.

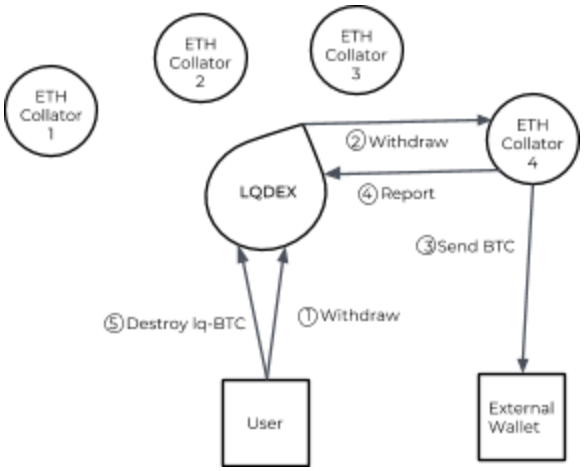

**Figure 5: Withdraw transaction flow**

## Collateral Management

Collators must maintain a sufficient collateral to be able to perform certain operations, such as accepting deposits from users. As the value of the external asset fluctuates, the ratio of the amount of the collateral to the value of the external asset held by a Collator changes.

A Collator may also withdraw all or part of its collateral, but only so that the remaining amount is at least 200% of the value of the external asset held by the Collator.

LB keeps balance information of all assets, for which it issued proxy tokens, of all Collators. It computes the value of each asset held by a Collator and the value of its collateral. If the ratio of the value of the collateral and value of the asset, held by a Collator becomes low, LB can increase that ratio by sending withdrawal requests only to that Collator until the ratio becomes at least 200%.

However, there may be situations, when all Collators do not maintain sufficient ratios for a particular asset. In this case LB can perform the following actions:

| Collateral Ratio | Action |
|---|---|
| **> 200%** | The Collator can process withdraws.<br>The Collator can process deposits for amounts that would not bring the collateral amount to below 200% of the Collator's balance. |
| **<= 200%** | The Collator can process withdrawals only.<br>LB sends a notice to Collator to increase the collateral. |
| **<=175%** | LB sends notices to all users holding the corresponding proxy tokens informing them that the support for the external asset will be soon discontinued.<br>Users may optionally withdraw or exchange the external asset. |
| **<=150%** | LB liquidates the Collator's collateral.<br>LB cancels proxy tokens of the asset held by the users.<br>LB sends LQD tokens in the amount of ~150% of the value of the cancelled tokens to the users. |

## Collator Loss Recovery

So far, we have discussed what happens when everything is running smoothly, also known as the "happy path." Let us now consider a scenario, when a collator is compromised.

While it is economically unprofitable for a Collator to walk away with user funds, it is still possible of a Collator to be compromised due to hacker attacks, mismanagement, hardware malfunction, and other events. Since each individual Collator is a centralized server holding a private key to its assets, a hacking attack on a Collator can potentially lead to a complete loss of funds held by that Collator.

LQDEX Blockchain must be designed in way that a loss of any Collator does not lead to loss of user funds. In fact, LB counts on the fact that some collators will be compromised periodically, as all centralized entities holding private keys usually are.

## Withdrawal Recovery

So how does that work? One misconception is that if a Collator (let's say for Bitcoin) goes down, LB blockchain distributes its Collateral to the users, whose Bitcoin was held in custody of that Collator. But, this is not how it usually

works. Instead, users would typically be completely unaffected and not even aware of one Bitcoin Collator going down.

When a Collator becomes unavailable for any reason, the following steps happen:

1. LB sends a command to the Collator to perform another withdrawal.
2. The Collator does not respond within a given time period.
3. LB sends the withdrawal command to the next Bitcoin collator.

So, the user withdrawal is performed either way, at least in most cases. It this scenario, the user does not notice and does not care if one Bitcoin collator went offline. The system continues to function, as if nothing happened.

However, what if in step 3, there is no other Collator available to process the withdrawal command for the requested amount? It can potentially happen, if the user requested to withdraw are relatively rare asset, held by a small number of Collators. Only in this case, LB would have to liquidate the unavailable Collator's Collateral and distribute it to the users affected by the loss of that Collator.

## Asset Recovery

In step 3 of the algorithm above, LB simply sends the command to the next Collator able to process the withdrawal. This allows completing the withdrawal operation without any friction of the user. However, now there is a situation that the total amount of asset (in our example it was Bitcoin) held by all Collators is smaller than the amount of the corresponding proxy token in the system.

If the asset deficit is left uncorrected, at some point in the future, a situation may arise that a user will try to withdraw Bitcoin, and no Collator can process the withdrawal request, as the total amount of Bitcoin held by all Collator is smaller than the amount of the corresponding proxy token. However, at that point in the future the system may no longer be able to compensate users for the loss of Bitcoin with the lost Collator's Collateral. Imagine, if the Collator went down a year ago, but the situation, when the withdrawal could not be processed, occurred just now. The exchange rate of LQD token against BTC may have changed substantially, and the Collateral may no longer be enough to compensate users for the loss of Bitcoin.

To prevent this situation from occurring, LB attempts to recover the lost Asset and restore the one-to-one correspondence of the actual asset and the corresponding proxy token in the system.

The asset recovery works as follows. When a Collator becomes unavailable and does not respond within a specified period of time, LB assumes that the Collator is now permanently offline. LB now offers the Collateral of that Collator to other Collators in exchange for the asset held by the lost Collator.

The Collateral-asset exchange is structures as an auction. LB broadcasts the intent to acquire the exact amount of the lost asset to all Collators known to hold that asset. The Collators submit bids with LQD token to provide the asset. For example, let us say LB needs to acquire 100 BTC. Collators bid how much LQD they would like to receive for providing 100 BTC to LB. These amounts can be calculated as the current market rate of BTC vs. LQD plus a premium for providing the service.

In a sense, it is an opportunity for Collators to acquire more LQD token at a discount. How much of a discount? The lost Collator's Collateral was at least double the amount of the asset held, so LB has a lot of room to offer a discount for selling its Collateral in exchange for the lost asset.

After the bids are submitted, LB simply selects the lowest bid and executes the exchange of LQD for the real asset.

The mechanics of the exchange is very simple, no real assets, such as BTC or LQD are actually moved anywhere. LB simply increases the amount of Bitcoin held by that Collator and its Collateral by the amount of LQD it received from the exchange in the Collator ledger. It will now count on that Collator to process Bitcoin withdrawals up to the new amount of Bitcoin.

# LQDEX Blockchain Architecture

To satisfy the requirements 4 and 5 of the ideal exchange, being inexpensive with fast order execution, the LQDEX Blockchain has to have high transaction throughput and no miner fees. These requirements are accomplished by the "proof of stake" (PoS) blockchain architecture.

## Proof of Stake vs. Proof of Work

In a PoS blockchain, nodes generate blocks based on how many tokens they have, not on their computational power. This is different from a proof of work (PoW) blockchain, where nodes need to solve a computation puzzle to produce new blocks. The main disadvantage of a PoW blockchain is that substantial time and energy is required to produce a block. So, the recording of the transactions is slow and requires miner fees to pay for the expense of the nodes (also called "miners"), which produce blocks.

For example, Ethereum, at the moment, uses the PoW block creation mechanism. As a result, EtherDelta and 0X decentralized exchanges, which work on Ethereum, suffer from slow speed and having to pay miner fees for each transaction. These miner fees are paid to the miners. They do not generate profit for the exchanges.

Another drawback of the PoW system is the higher potential vulnerability to the 51% attack discussed later in more detail. Some coins, such as Bitcoin, have limited supply. At some point in the future, when all coins are mined, miners will no longer receive reward for mining in new coins. They will only be rewarded with transaction miner fees. It is possible that fewer miners will operate at that time, as the reward will be lower. With fewer miners, a situation is possible, when someone compromises 51% or more of the nodes, thus launching the 51% attack. If someone gains control of 51% or more of the nodes, he, she, or they can send fraudulent transactions to the network and steal other users' funds.

PoS blockchain implementations also have the advantage of being able to come to consensus on block finality (discussed later), an important feature for an exchange.

LQDEX blockchain is a PoS blockchain. We selected the PoS blockchain model, because of the benefits listed above. This implementation of the PoS blockchain prioritizes consistency over availability.

## The Stake

LB uses its LQD token as a stake to record transactions or to supply external blockchain information. Each node needs to deposit an amount of LQD in order to produce blocks on LB blockchain. When a new transaction happens on the network, all nodes record it, when they receive it. The recording of new transactions occurs instantly (with the speed of the network), as no solving of a computational puzzle is required.

## The Reward

No miner fees are paid for recording a transaction on LB. Instead, the nodes get rewarded with reward paid by the LQD token. The more LQD token a node deposits with the network, the higher its stake and the larger share of the total network reward it receives.

The reward comes from the commissions charged for trades and withdrawal of funds. It is fundamentally different from the miner fees. An Ethereum dapp, such as EtherDelta, pays miner fees to third parties, the miners. They, in turn, spend the miner fees on their electric bills. These miner fees do not contribute to

EtherDelta's profit. In contrast, commissions and fees paid on LB are distributed to the prospective stakeholders as rewards, so they contribute to the participants' profits.

## Functional Requirements

Because this blockchain application is an exchange, it has a set of unique requirements than other blockchains, such as Bitcoin or Ethereum, may not implement.

### Fast Execution

Users expect trades to be executed quickly. Execution time of one or two seconds or even less is ideal. In contrast, on Bitcoin blockchain, blocks are mined on average once every ten minutes, and even, when a block is mined, the transaction is not confirmed. Users will not wait that long to perform an operation.

### High Transaction Volume

High transaction volume is essential for an exchange. If the volume is low, the spreads are getting larger and trading becomes less attractive for most users.

### Finality

Most commonly used blockchains do not have the finality problem solved. In other words, the users never receive a 100% guarantee that the acceptance of their transaction by the blockchain is final. For example, on Bitcoin or Ethereum (the current version), even after many confirmation, there is still a chance, although it becomes smaller as the time goes by, that a transaction can still be rejected by the blockchain.

Not having finality is unacceptable for an exchange type of application. If there is no finality, a situation is possible, when by the time a transaction can be found invalid, the user has already withdrawn all funds from the exchange, so the blockchain state cannot be reversed to the correct one.

### Fairness

On an exchange the trade execution order is important. Other blockchains, Bitcoin or Ethereum, group several transactions in one block. After the transactions are arranged in a block, their order is not saved. The exchange blockchain needs to keep track of the trade timestamps and also reach consensus on that order.

### High Security

The exchange blockchain needs to have a an efficient mechanism for rewarding and penalizing the nodes for recording transactions. It needs to be Byzantine fault tolerant and resilient to DDOS attacks, long range attacks, and other possible attacks, as long as the number of malicious blockchain nodes is no higher than one third of all nodes (the theoretical limit of proof of stake blockchain resilience).

### Simple Implementation

Simple implementation is also an important requirement. Not only complex algorithms and systems take longer to implement, they are more difficult to test and audit. A probability of a vulnerability or an error goes up substantially with implementation complexity. In addition, the exchange blockchain needs to be predictable and publicly auditable. If the inner workings are overly complex, they public may not be able to trust the exchange completely.

### Low Equipment Cost

Low barriers to entry for businesses that wish to run the blockchain nodes is important. If business are required to spend substantial amounts on equipment alone, running a node will be less attractive. So, the number of nodes

will be smaller, and the network will be less stable.

## Byzantine Fault Tolerance

Byzantine fault tolerance was named after the Byzantine Generals' Problem. Imagine that you are a member of a group of generals in the ancient Byzantine Empire. Each of you commands a portion of the army. You need to come to an agreement on whether to attack an ancient city or to retreat. You need to make sure that whatever the decision is made it is a coordinated attack or a retreat, as in the worst case scenario, the entire army can be defeated.

Imagine now, that some generals are traitors and may cast malicious votes for a suboptimal strategy, but you do not know which ones.

Also imagine, that all generals are located far away from each other and send messages via couriers, who may or may not deliver a message, and who may also forge messages.

In a blockchain architecture, each node can be thought of as a Byzantine general. The nodes need to come to a decision, called a consensus, on whether or not to keep or to discard a block of data.

Byzantine fault tolerant systems are networks that function in an environment, where it not known whether or or not each node is malicious or not.

Byzantine fault tolerance (BFT) usually refers to a measure of what fraction of nodes in the network can be malicious before the network seizes to properly function. For example, a 33% BFT system means that the network can function correctly, if up to 33% of its nodes are compromised.

## General Assumptions

LQDEX network is first and foremost a blockchain. As such, it has the advantages of using the blockchain technology, discussed earlier.

Likewise, however, it also has the disadvantages inherent to all blockchains. When we discuss LB architecture and how it functions in different usage scenarios, it is important to understand the limitations of the blockchain technology.

For example, Bitcoin would not function well, if the entire network consisted of just one node, controlled by one person. It could possibly function, but then it would not be trustless. The users would have to trust the owner of the only Bitcoin node to correctly record transactions. That node operator could double spend his payments to the users. (Although, it is debatable whether or not she would benefit from defrauding its users. Various adversarial scenarios are discussed later in this paper.)

It is a known fact that blockchains become vulnerable if the number of nodes is small.

Blockchains, both PoW and PoS, are also vulnerable to 51% attacks. In other words, if a malicious actor takes over a majority number of nodes or stake in the network.

Furthermore, it can be shown that PoS blockchains remain Byzantine fault tolerant if only no more than 1/3 of total stake involved in malicious voting in the network environment with bounded, but unknown network latency. [17]

For the purpose of this white paper, unless stated otherwise, the following assumptions will apply:

1. There is a very large number of nodes, all operated by different entities
2. There is a very substantial amount of total stake in the system

3. For each asset, there is a very large number of Collators, which work with that asset

In this context, “very large” means a number that would make it impractical to mount a coordination attack, where a substantial number of nodes collude with each other.

A “substantial amount” means a large enough amount, which makes it prohibitively expensive for any malicious actor to invest in a Stake equal to 1/3 of that amount in order to attempt an attack on the network.

We understand that these assumptions will not always be true in the real world, and we will go over possible attack scenarios later in this paper.

However, if someone asks us what will happen if 51% of all nodes (or total Stake) become malicious or what if all Bitcoin Collators collude with each other and decide to misbehave, we have to say that LB is by design not equipped to handle these types of attack scenarios. But, neither does Bitcoin or Ethereum. This is a limitation of all blockchains, not just LQDEX.

## Block Content

Each block contains the following types of data:

1. Funding requests (deposit and withdrawals)
2. Trade orders
3. Trade executions
4. Trade settlements (asset account balance changes)
5. Available Nodes table
6. Submitted votes
7. Finalization decision on a prior block
8. Node ID, which proposed the block

| LQDEX Block |
| --- |
| Header |
| Funding requests |
| Trade orders |
| Trade executions |
| Trade settlements |
| Submitted votes |
| Available nodes |
| Finalization or previous block |
| Node ID |

**Figure 6: LQDEX block contents**

Only the changes to the data needs to be stored in each block. For example, if only one new trade order came in during that block generation time, only that information is stored in the block. So, to fetch the entire order book, the client needs to collect the data from multiple blocks.

## LB Performance

As we saw earlier, speed of execution is one of the top requirements for a successful exchange. LB is designed to accomplish the highest possible execution speed. In fact, as we will see below, the order processing speed of LQDEX is comparable to a regular centralized exchange processing speed.

Most POS blockchains utilize node voting in one form or another to reach consensus. As a result, their processing speed is limited by the internet round-trip delay time also known as network latency. The consensus on a block cannot be reached quicker than the majority of the nodes could cast their votes over the network.

In addition, if a particular application required computation-intensive calculations, the nodes are typically not rewarded for computing the tasks quicker than the other nodes. And there is no reason for it, as even if one nodes completes the computation quicker, it still needs to wait for

the other slower nodes to reach consensus on the results. So, most POS systems, although a step up from PoW blockchains, are still inherently slow.

In contrast LB has the following features:

1. Performs as many operations as possible without a voting consensus.
2. Rewards and takes advantages of the nodes with higher computing power.

## Lead Node

To accomplish these objectives, LB introduces a concept of a Lead Node. Simply speaking, the Lead Node is the most powerful node among all nodes in the network, capable of executing the highest number of transactions per second. The Lead Node receives the highest reward out of all LB nodes, so LB encourages the competition on computation power among its nodes.

This concept is somewhat similar to a PoW blockchain such as Bitcoin, where a more powerful node would mine more blocks and receive more reward. However, the difference is that in case of Bitcoin, the nodes compute hashes of no intrinsic useful value. In LB, the nodes compute the actual exchange application payload such as trade matching and settlements.

The Lead Node is selected by voting. All nodes in the system compute new blocks. When a block is computed the time of the computation is recorded as part of the block. The node that computed all blocks faster than the current Lead Node for a period of time, let us say 100 blocks in a row, is voted to become the new Lead Node.

## Finality

The trading within LB is implemented with a use of proxy tokens. A user at his or her discretion can convert his proxy tokens into a real underlying asset via a process we discussed in details earlier. In order to make this possible, we need to introduce a concept of finality.

Consider blockchains similar to Bitcoin or the current version of Ethereum. On these blockchains the user never knows when the transaction is finalized and can never be reversed. Instead, the user is given the number of confirmations or a “block height” of his transaction. The number of confirmation indirectly indicates the likelihood of his transactions being ever reversed. The more confirmations, the lower the probability.

Let us say that a user obtained a particular proxy token as a result of a trade, and the user converts his proxy token into a real asset. If LB was like Bitcoin or Ethereum, there would always be a small probability that the trade would be found invalid and the proxy tokens received would be cancelled. But, the user has already received the real asset, that LB cannot cancel or reverse a withdrawal!

Therefore, it becomes clear that LB needs to implement a concept of finality. If a block is considered final, the transactions contained in that block and the resulting proxy tokens can never be cancelled. The user can safely convert these “finalized” proxy tokens into the corresponding real asset without any risk to the network.

While the above conclusion is fairly obvious, the reverse is slightly less intuitive. Specifically, the finality is only required for the conversion of proxy tokens into real assets. In other words, a user can potentially perform any number of operations with proxy tokens only. As long as nothing is withdrawn, it is not important whether these transaction are finalized or not. At worst, the proxy token transactions can be reversed and executed again at a slightly different price or order. But no real assets or money will be lost. (We will also see later that since Collators get their stake slashed for performing incorrect

operations, it is also very unlikely that even the unfinalized proxy token transactions would be reversed.)

To summarize the finality is required, but it is not really important to users up until they decide to withdraw their funds.

## Trade Execution

With that in mind, LB is structured in way that allows it to perform proxy token trades at the same speed as a regular centralized exchange. The consensus is only required prior to performing withdrawal operations. It functions as follows.

The Lead Node receives trade orders from Clients. It executes all orders with the maximum speed it can. The Lead Node does not wait for any other nodes. Therefore, the internet round-trip delay time is no longer a factor in trade speed execution.

The Lead Node arranges transactions in blocks and makes them immediately available to clients and other nodes. End users see their trades near instantly executed.

The only limitation is that users cannot withdraw their funds until the consensus is reached.

This limitation, however, is not a major drawback. Sending Bitcoin, Ether, or other assets is a lengthy process, much longer than the internet network latency time. The additional minor delay required for consensus will not significantly affect the user experience. (Also, centralized exchanges, even though they do not wait for consensus, often implement various time consuming security checks prior to fund withdrawal.)

## Block Arrangement

Transactions are arranged in blocks based the time when they were received by a node. So, one block can have any number of transactions in it. However, if there were no transactions received during the block time period, no block is created. For example, a block time interval can be precisely ten seconds.

Because of the network latency and unequal transaction propagation paths, all nodes receive transactions at slightly different times. In borderline cases, two nodes may decide to put the same transaction in two different blocks.

## Block Validation

While the Lead Node executes all trades on its own, all other nodes perform validation of its work. They iterate over the following steps:

1. Fetch the blocks produced by the Lead Node, that have not been finalized.
2. Out of these nodes, select a random block that has not been yet validated by the current node.
3. Compute that block based on the prior state of accounts and new orders received by the current node.
4. Vote yes on the block, if the result matches. Vote no otherwise.
5. Continue with step 1.

When comparing a computed block result with the block generated by the Lead Node, the nodes need to take into an account that transaction timestamps may be slightly different. As a result, some transactions may not be included in the block or an extra transaction maybe included. The node needs to verify the order of the transactions and take into the account the borderline inclusion or exclusion cases.

## Consensus

Each block has a timestamp of when it was generated by the Lead Node. After a certain time period called “Voting Time Threshold”, the Lead Node tallies up all votes for a particular block.

If the node super majority voted Yes, the validation is considered successful. The Lead Node records the validation result in the current block.

If, however, the supermajority did not vote Yes, the block is discarded. Also discarded are all blocks generated after this block. The Lead Node starts computing blocks all over again beginning from the discarded block.

When the current block, in turn, is successfully validated by other nodes, it is ready to be finalized. As soon as all blocks are finalized prior to this block, it also becomes finalized.

## Collator Deficit

LQDEX is a great solution for a fast blockchain that allows nodes to generate blocks quickly and reach consensus even if they have different computational power. However, the system relies on a large enough number of nodes to be able to function. The availability of a large number of nodes is a typical requirement for all types of blockchains.

However, what can happen, if the number of available Collators is not as high as would be ideal. Imagine a situation, when a lead node generates so quickly, that a few other nodes struggle to keep up with block validation. Only a small number, let us say two or three nodes, are able to submit votes for each block. In that situation, we would not be able to reliably conclude that a particular block is a valid block.

To avoid the situation of nodes being “spread too thin” over a high number of blocks requiring validation, LB requires a minimum number of “yes” votes to be submitted before a block is considered to be validated. This minimum number has to be a statistically significant sample, let us say a minimum of 31 nodes.

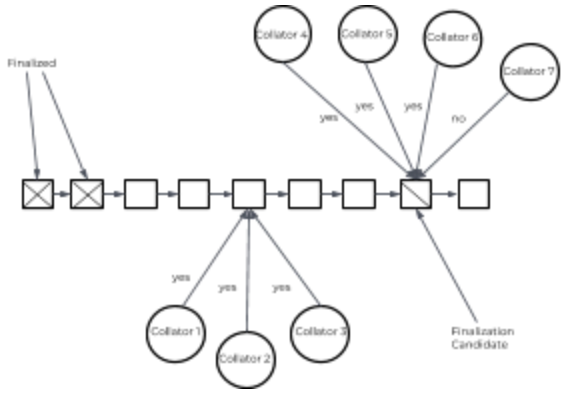


**Figure 7: LQDEX blockchain consensus**

If a situation arises, when newly generated blocks accumulate too quickly, and the nodes cannot keep up with the minimum validation rules, the Lead Node must slow down to match the speed of the remaining nodes.

This situation, however, should not occur, if the number of available Collators is sufficiently large.

## Gossip Protocol

To further increase the network speed, LB uses a modified version of a gossip protocol between its nodes and client connectivity.

In a typical blockchain, each node connects to a number of other random nodes. Let us say that this number is eight nodes. Likewise, end user client applications transmit their transactions to eight random blockchain nodes.

In LB, the node connectivity protocol is similar. But, a node or a client app, always attempts to secure at least one connection directly to the Lead Node. The remaining connections are to the seven random other nodes. If the Lead Node refuses a connection, the node or a client app

seeks a connection to another node closely connected to the Lead Node.

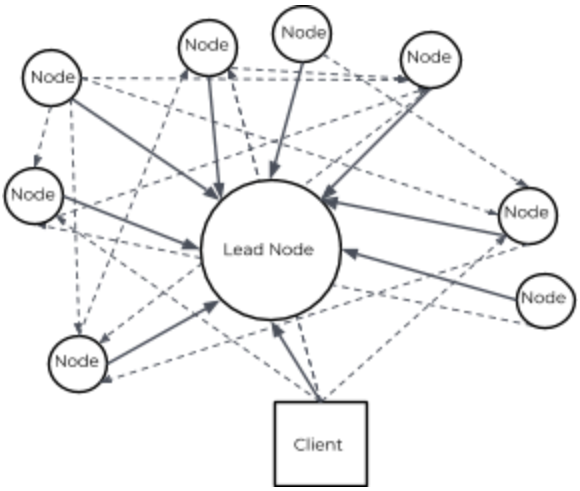


**Figure 8: LQDEX modified gossip protocol**

This modified gossip protocol shortens the signal travel time between the Lead Node and the other nodes and client applications.

## Reward and Penalty

The reward is paid to each node for a generated block or for a block the node voted on. The amount of the reward is computed with the formula:

$$R = K \cdot S$$

K - is the reward coefficient determined by LB
S - is the amount of collateral or stake deposited by the node

The reward coefficient depends on the network profits, the size of the stabilization fund, and the overall collateral ratios as discussed in the Monetary Policy section.

The reward is paid to the node in two cases:

1. A block generated by the Lead Node is successfully validated.
2. A node voted correctly on a block.

On the other hand, if the Lead Node generates an incorrect block or a regular node votes incorrectly on a block, the amount equal to a reward is taken out of the node's stake on LB as a penalty.

## Cryptographic Primitives

LQDEX uses standard cryptographic primitives that are proven to be scalable with networks like ethereum. Two important choices with respect to cryptographic primitives in a design of any blockchain would be following:

1. Hashing Algorithm.
2. Digital Signature Algorithm.

### Hashing Algorithm

Ethereum uses a variation of SHA3 which is called KECCAK-256 differs in the padding as described below.

SHA3-256(M) = KECCAK [512] (M || 01, 256)

We will be using KECCAK-256 as in ethereum standard implementation for our block hashing and computation of merkle hashes.

### Digital Signatures

Cryptographic Signatures have been instrumental in the success of blockchain and verification of the fact that the message was actually sent by the claimed sender.

Ethereum uses the ECDSA of the SECP-256k1 curve and using hash of the datum to sign.

We will be using exactly the same implementation as ethereum for signing trade transactions on LQDEX blockchain.

## Governance

LQDEX is governed by Collator Users. Collators Users can submit various proposals on how the LQDEX system can be modified. The issues for modification include:

1. Commission and withdrawal fees
2. Collator reward
3. Stabilization fund amount

When a proposal for modification is submitted, other Collator Users can optionally vote on the proposal. The voting is weighted with Collateral. If supermajority approves the proposal, the modifications of the system take effect.

More substantial system modifications are also possible. The Collators Users can also collaborate offline to agree on greater modifications of the system that require programming changes in LQDEX core software.

# Price Volatility of LQD Token

The LQD token is used as a collateral by Collators. In the Collateral Management section, we talked about what actions the network will take if the collateral ratio of a given Collator decreases. However, if the price of the LQD token drops very rapidly, it may still be possible for Collators to become undercollateralized.

To prevent the price of LQD token to fluctuate significantly against the value of other assets traded on LQDEX exchange, the network implements the token monetary policy. Similarly, to the monetary policy of the Federal Reserve, this policy is designed to keep the price of the LQD token as stable as possible.

## Token Value Model

Collators holding LQD tokens receive rewards for performing operations on the network, such as block confirmations. The present value of this token can be calculated with the formula:

$$PV = \frac{R}{(1+i)^1} + \frac{R}{(1+i)^2} + \frac{R}{(1+i)^3} + \dots + \frac{F}{(1+i)^n}$$

R - reward payment
F - price at the time of sale
i - inflation rate or opportunity cost
n - the number of time periods the tokens are held for

Example: suppose a Collator plans to hold tokens for one year. It estimates its opportunity cost to be 15% on annual basis, the future sale price to be the same as the buy price. The reward paid by the network is currently set at 15% on annual basis. The current price of tokens is $1,000.

In that case, the present value of the token to that Collator is exactly $1,000. In other words, the Collator is indifferent on whether to sell tokens or buy more tokens.

Let us consider another case. Now, the reward is set at 30%. With everything else staying the same, the price of the tokens is now $1,130. The Collator has an incentive to purchase tokens at the current price of $1,000 to make a profit of $130.

On contrary, if the reward decreases to only 10%, the present value of the tokens decreases to $957. The Collator has an incentive to sell the tokens to avoid losses.

If all market participants are rational, they buy assets when they can generate profit and sell assets when holding them becomes unprofitable. When demand for the asset increases, the price goes up to point when it becomes indifferent for the market players to buy more of that asset or to sell it. Likewise, when the demand drops, the price of the asset decreases to the point of equilibrium. While in practice, these price movements can take some time, we will assume here for simplicity that the market for crypto assets is very efficient and any changes to the reward are instantly reflected in the price of the token.

## Reward and Inflation

LB generates profits by charging users fees such as trade commissions and withdrawal fees. These profits are used for payments of the reward to the Collators. For a given time period a Collator receives a reward, which is proportional to the Collators stake (the collateral amount) and the number of operations performed by that Collator on the network, such as block creation, etc. As we saw earlier, if the network could increase the amount of the reward, it could increase the price of its token.

Another way for LB to affect the price of its token is to create new tokens (print money). When new tokens are created, a “token inflation”

occurs. If the demand for tokens stays constant the price of one token decreases due to the inflation. You can see in the formula above, if R (reward) stays the same, but F (future value) decreases, then the P (current price) also decreases.

In the following sections, we will show how both of these instruments, controlling the reward amount and issuing more tokens, can be effectively used to control the price of the token.

## A “Black Swan” Event

A “black swan” event is an event when an unexpected and unusual financial calamity occurs. In regards to our application, the only such event would be if the price of LQD token fell so rapidly that all or many Collators would become undercollateralized quicker than the network could perform collateral liquidations at the rate of 100% or higher of the asset value held by the Collators. If this event was to occur, the users of the network could potentially lose their funds.

We also have to take into account that if such event was to occur the profits generated by the network would not potentially be enough to increase the reward sufficiently to increase the price of LQD token. Creating new tokens, as we saw earlier, would also be ineffective.

## Stabilization Fund

To address the potential black swan event problem, the network maintains a Stabilization Fund. The Stabilization Fund is simply an amount of LQD token held by the network. The fund is used to stabilize the price of the token quickly if it starts declining unexpectedly and a risk of under collateralization increases. Just like anything else, the Stabilization Fund is managed by a smart contract (a predetermined formula). The algorithm is as follows.

Each reward payment period, the network performs the following steps:

1. The network analyzes the collateral ratios of all Collators.

**IF ratios are too high (LQD appreciated):**

2. The network decreases the reward to Collators.

3. If the reward is already near zero, and the price of LQD keeps increasing, it signals a spike in the demand for LQD token due to more users or Collators joining the network. In that case, the network creates more tokens (prints money) to increase the supply.

4. The surplus tokens (all that are not paid to Collators) are added to the Stabilization Fund.

**IF ratios are too low (LQD depreciated):**

2. The network increases the reward to Collators. The reward is taken from the network profits and from the Stabilization Fund.

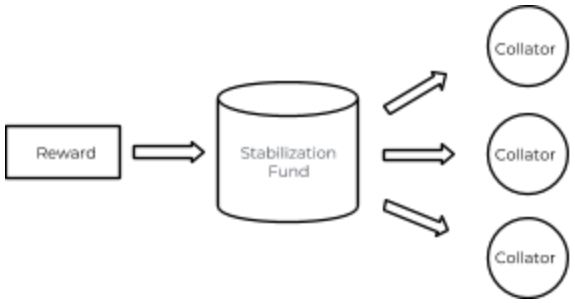


**Figure 9: Reward payment flow**

How large should the Stabilization Fund be? It is easy to compute the size of the Stabilization Fund that is required to prevent the price of token falling by a given number of percentage points.

For simplicity of calculation, let us assume that Collators plan to hold the token for an indefinitely long time. In that case, the token become an annuity. Its pricing is now determined by a simpler formula:

$$PV = \frac{R}{(1+i)^1} + \frac{R}{(1+i)^2} + \frac{R}{(1+i)^3} + \ldots = \frac{R}{i}$$

R - reward payment
i - current inflation rate or opportunity cost

We can see that in order to increase the price of the token by a factor of two, we need to increase the reward by a factor of two.

Let us also assume that currently all Collators are well collateralized, meaning that each has a collateral ratio of 200% or higher. In order for the users to be at risk of losing their funds, the collateral ratio has to fall by a factor of two or more to 100% or lower.

Let us say that the current reward rate is 15% on annual basis. That means that if the total amount of all tokens held as collateral is 1 million, the network pays out 150,000 as rewards per year to all Collators.

How much token should be kept in the Stabilization Fund to prevent Collators to become undercollateralized, if the price of token decreases by a factor of two?

To keep the price of the token constant, we will need to double the reward. Instead of 150,000, the network will need to pay out 300,000 in rewards per year to keep the price of the token on the same level. That means that the network will need an additional 150,000 to pay a total of 300,000. That extra 150,000 will need to come from the Stabilization Fund. So, we computed that in this scenario, the size of the Stabilization Fund should be 150,000 or about 15% of the total collateral amounts held by all Collators.

*A good rule of thumb then is to keep the size of the Stabilization Fund equal to the amount of the current reward that needs to paid to all Collators for the period of one year. This way, if the price of LQD token drops by up to 50% (factor of two), it can be quickly brought up to the original level by doubling the reward and sustaining the larger reward payments for at least one year.*

If the low demand for LQD token and the sale pressure persist for longer than one year, the Stabilization Fund will be depleted, the reward payment will drop, and as a result, the price of LQD will drop as well.

However, one year is usually plenty of time for the effects of the financial calamity to subside or to address any fundamental problems with the network, if any. In the worst-case scenario, if nothing can be done, one year is enough time to liquidate the Collator collaterals and reimburse all users of the network, so no one loses their funds.

# Adversarial Models

## Nothing at Stake

The nothing at stake problem occurs when a node is incentivized to vote “yes” for a block, but is not penalized in any way for incorrect voting. The lack of penalty is referred to as “nothing at stake.”

In this situation, the optimal strategy of each network participant is to vote “yes” regardless of the actual block content. If a fork is created accidentally or maliciously on the network, the nodes will always vote “yes” for both versions. This creates a possibility when funds can be double spent, or other conflicting type of transactions can exist on the blockchain.

To address this problem, LB has a penalty in the amount equal to the reward for incorrect block generation or voting.

If a node votes “yes” on two conflicting blocks, it will receive the reward for the correct block and lose exactly the same amount for the incorrect block. The total gain in this case is zero.

Similarly, if a node votes “yes” for three (or more) conflicting block, the total gain becomes negative, as only one block will be correct.

As a result, in no situation, the node has an incentive to vote “yes” on more than one block.

## Sybil Attack

In a Sybil attack, the attacker creates a large number of malicious nodes controlled by the attacker. The nodes then cast incorrect votes in the system. If misbehaving nodes voting power is higher than the voting power of the the good nodes, the attacker can potentially perform incorrect transactions, such as steal funds from other users.

This attack method is named after the subject of the book Sybil, a case study of a woman diagnosed with dissociative identity disorder. [16]

Some PoW blockchains can be vulnerable to Sybil attacks. PoS blockchains are typically not susceptible to attacks of this type.

Specifically, LB is not vulnerable to a Sybil attack, because the node’s voting power is proportional to its Stake in the network. In other words, it is not enough to create malicious nodes in the system. The attacker would also have to make a substantial investment in the staking token to have a voting power in the network.

In fact, the actual number of voting nodes does not make a difference. To launch a Sybil-style attack, the attacker would need only one node with a Stake amount sufficient to provide a Supermajority vote in the system. In this case, the attack becomes the 51% attack, discussed below.

## 51% Attack

A 51% attack is the scenario when a malicious actor takes over a substantial portion of the network. In case, of PoW blockchains, that usually means the number of nodes or the hashing power. In case of PoS blockchains, that means that it controls the majority of voting stake.

All blockchains are vulnerable to 51% attacks. Specifically, PoW blockchains can tolerate up to 1/3 of malicious actors in an environment with bounded, but unknown network latency.

Since LB is a PoS blockchain, the same limitations apply to LQDEX.

If a malicious actor takes over 34% of the stake in the system, it can prevent new blocks from being finalized. Since the supermajority vote is required for block finalization, the misbehaving node or nodes can prevent the network from finalizing blocks. While the trades would still be performed, the users would not be able to withdraw funds.

If a misbehaving actor takes over 67% of the stake, it can effectively generate malicious blocks and defraude other users.

The same scenario can occur on smaller scale, if a malicious actor takes over a subset of Collators that process transactions for a given asset, such as Bitcoin. It can then either prevent Bitcoin withdrawals with 34% stake or perform fraudulent withdrawals with 67% stake.

The only way to protect from the 51% attack is to have the total stake in the system large enough and spread over a large number of Collator Users.

The total Stake should be large enough to make it prohibitively expensive for any single entity to acquire a stake equal to 34% of the total Stake on the network.

Similarly, the number of Nodes and Collators for any particular asset should be large enough to make impractical to launch a coordination attack on all nodes.

The 51% attack vulnerability is inherent to all blockchains. That said, we should also note that even if one actor takes over the substantial part of the network, it may not be necessarily motivated to defraud the users. Satoshi made in interesting point in Bitcoin White Paper:

> *If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.* [18]

Intuitively, the rationale is clear. Let us say you were able to acquire 51% of your neighborhood Walmart Supercenter. You now have full control and can do what you want with it. You have a choice to make. You can:

a) Steal everyone's wallet now and disappear with the money.
b) Continue operating the store and receive profits on continuous basis.

Chance are if someone has enough money to acquire a substantial portion of the Stake in the network, they might be more motivated to play by the rules and make continuous profits as opposed to defrauding the users.

However, predicting human behaviour is beyond the scope of this white paper. So, technically, we have to say that LB is, in fact, vulnerable to the 51% attack, just the same as other blockchains are.

## Non-Best Price Execution

Centralized exchanges may have algorithms in place that do not offer the Best Price Execution. For example, an exchange may engage in practice of trading against its users. When a new order comes in, the exchange may attempt to match this order with its own, trade against the user, as opposed to matching the order with the best price order available in the system. When a trade against the user is made, the exchange benefits not only from the trade commission, but also earns money of the spread. This practice is commonly used by "Dark Pools."

Can a Collator receive an order and attempt to match it with its own order at a price, which is not the best price?

This situation is obviously not possible on LQDEX. The Collator software is programmed to always execute order with the best price. If one Collator modifies its software to match an order with another order, no matter its own or not, which does not have the best price, the remaining "honest" nodes will not validate this trade.

## Front-Running

Front-running is a practice of exploiting information about an order before it enters the exchange order book.

For example, a trader places an order to buy 100 BTC with USD at market price i.e. a market order. Currently, there is a limit order in the order book that offers 100 BTC for sale at 10,000 BTC/USD. There is also another limit order to sell another 100 BTC at 10,100 BTC/USD.

The "front-runner" has somehow obtained information that the market order to buy 100 BTC is about to enter the exchange. This front-runner now has an informational advantage. He or she can quickly purchase 100 BTC for 10,000 BTC/USD and offer it for sale at a higher price, but not higher than the next limit order of 10,100 BTC/USD.

He now buys 100 BTC at 10,000 BTC/USD and places a limit order to sell 100 BTC for 10,099 BTC/USD.

When the market order for 100 BTC finally enters the order book, it gets matched at best price with the front-runner's order. The trader pays 10,099 USD for each BTC.

The front-runner earns 99 x 100 = 9,900 USD from the transaction with no risk! The trader, however, loses 9,900 USD, as he could have bought at 10,000 BTC/USD, if there was no front-runner.

Can someone engage in front-running on LQDEX? While front-running is theoretically possible at LQDEX, given the large number of Collators, the infrastructure investment required to gain an informational advantage is cost prohibitive.

Each trade order is submitted simultaneously to multiple Collators. They, in turn, transmit the order via gossip protocol to other Collators and so forth. The order propagates through LQDEX network. When an order is received by a Collator, it records the order time stamp. Collators would then have to reach the supermajority voting consensus on what sequence the orders were received in.

If an aspiring front-runner sets itself up on the same network as the Collators, it sees the orders at about the same time as the other Collators see it. Therefore, it has no informational advantage.

But, what if a front-runner finds a way to connect to one Collator, let us say the Lead Node, with a very fast network connection? The same front-runner, also found a way to connect with very fast connections to two thirds of all other nodes. He can now receive new order information from the Lead Node and send its own orders to the two thirds of the other nodes quicker than the trader's order propagates through the network.

While this scenario is theoretically possible, given a large number of Collators, the infrastructure investment to perform this type of front-running will most likely be cost prohibitive. In addition, new Collators periodically enter the network in various geographical locations. Being able to build and maintain fast network connections to two thirds of all Collators is unrealistic.

# Glossary

| | |
|---|---|
| **LQDEX Blockchain** | (Or “LB” for short), a blockchain consisting of LQDEX Nodes. |
| **External Blockchain** | (Or “EB”), any blockchain, except LB, such as Bitcoin or Ethereum. |
| **LQD Token** | The main token on the LQDEX blockchain, similar to ETH on Ethereum. |
| **Proxy Token** | A token on LQDEX Blockchain, corresponding to another asset such as Bitcoin or Ether. It is similar to an ERC20 token on Ethereum. |
| **LQDEX Core** | Software, which performs and records operations of the LQDEX blockchain, similar to Bitcoin Core or Ethereum Core. |
| **EB Core** | Software, which performs and records operations on an External Blockchain such as Bitcoin Core or Ethereum Core. |
| **EBI** | (Stands for “External Blockchain Interface”), a software that relays commands from LQDEX Core to an EB Core and relays information from an EB Core to LQDEX Core. |
| **Collator** | (A.k.a. “LQDEX Node” or simply “Node”), a device, hardware and software, running a LQDEX Core and optionally an EBI and one or more of EB Cores. |
| **Collateral** | (A.k.a. a “Stake”), a sum of LQD Token a Collator needs to deposit to LB in order to perform operations and optionally accept deposits of other assets such as Bitcoin or Ether. |
| **Reward** | A sum of LQD Token paid to a Collator for performing operations on LB. |
| **Node Supermajority** | 2/3 of all Collators which submitted a vote, or 2/3 of all Collators running a specific EB which submitted a vote. |
| **End User** | (Or a “User” or “Trader”), a person or an organization using LQDEX services such as trading of digital assets. |
| **Collator User** | a person or an organization operating a Collator. |
| **Client Application** | (Or a “Client”), a software providing access to LB running on the End User’s device such as a personal computer or a smartphone. |
| **Market Order** | A trade order without any price conditions. |
| **Limit Order** | A trade order with a condition to execute only at (or better than) a specified price, called “limit price.” |

| | |
|---|---|
| **Market Maker** | A trader, which placed one or more limit order. |
| **Market Taker** | A trader, which placed a market order. |
| **Liquidity** | The total amount of all limit orders (converted to the same currency, such us US dollar) at a given moment. |
| **Exchange Facility** | (Or a “Exchange”), a device, hardware and software, that performs trade order execution, including execution of automated orders such as limit orders. |
| **Best Price Execution** | Matching of a specific order with other orders to accomplish a trade in a way that yields the most favorable price. |
| **Dark Pool** | (For our purposes), an Exchange, which does not guarantee Best Price Execution. |

# References


[1] *Wikipedia.org*, Mt. Gox (https://en.wikipedia.org/wiki/Mt._Gox)

[2] *Bitshares.org*, BitShares 2.0: Financial Smart Contract Platform, Fabian Schuh, Daniel Larimer, 11/12/2015 (http://docs.bitshares.org/_downloads/bitshares-financial-platform.pdf)

[3] *Bitcointalk.org*, EtherDelta - Decentralized Token Exchange (https://bitcointalk.org/index.php?topic=2018051.0)

[4] *0xproject.com*, (https://0xproject.com)

[5] *Medium.com*, Ethereum Block Production Continues to Slide, Thomas Jay Rush (https://medium.com/@tjayrush/ethereum-block-production-continues-to-slide-1b74a2123e3f)

[6] A*rk.io*, What is the ARK SmartBridge, and How Does it Work?, Travis W. (https://blog.ark.io/what-is-the-ark-smartbridge-and-how-does-it-work-1dd7fb1e17a0)

[7] *Bitcointechtalk.com*, Atomic Swaps, Jimmy Song (https://bitcointechtalk.com/atomic-swaps-d6ca26b680fe)

[8] *The Dai Stable Coin System* (https://makerdao.com/whitepaper/DaiDec17WP.pdf)

[9] *Atomic Swaps* (https://en.bitcoin.it/wiki/Atomic_cross-chain_trading)

[10] *PolkaDot* (https://github.com/w3f/polkadot-white-paper/blob/master/PolkaDotPaper.pdf)

[11] *Ethereum White Paper* (https://github.com/ethereum/wiki/wiki/White-Paper)

[12] *BaseCoin White Paper* (http://www.getbasecoin.com/basecoin_whitepaper_0_99.pdf)

[13] *Omega One* (https://omega.one/static/media/whitepaper-eng-1.25.f65dc5a4.pdf)

[14] *Decentralized Token Exchange* (https://bitcointalk.org/index.php?topic=2018051.0)

[15] *CDO,* Vitalik Buterin (https://ethresear.ch/t/collateralized-debt-obligations-for-issuer-backed-tokens/525)

[16] *Wikipedia.org,* Sybil Attack (https://en.wikipedia.org/wiki/Sybil_attack)

[17] Consensus in the Presence of Partial Synchrony, Cyntyha Dwork, Nancy Lynch, Larry Stockmeyer (http://groups.csail.mit.edu/tds/papers/Lynch/jacm88.pdf)

[18] Bitcoin White Paper, Satoshi Nakamoto (https://bitcoin.org/bitcoin.pdf)

[19] NIST SHA3, (https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf)

[20] Ethereum Yellow Paper , DR. Gavin Wood (http://gavwood.com/paper.pdf)