Proof of Loyalty

Mr Raymond Ng & Mr Yuan Hang
raymond@katalystcoin.com
https://katalystcoin.org
Telegram : @raymondngkh

5 November 2017

License of White Paper

Abstract

This white paper details on the procedure on how to make use of existing blockchain platform to innovate a safe and secure implementation of a proof-of-loyalty offchain blockchain explorer. Proof-of-loyalty (PoL) allows any new blockchain platform to define the economic activity other than just purely computing hashes.

The primary purpose of computing hashes as a proof-of-work mechanism is because it is difficult, other than being difficult being an economic incentive to protecting blockchain transactions, there are practically no other well defined economic incentives.

Proof-of-loyalty allows a specific blockchain platform to define an economic activity so as to encourage its implementation. For example, filecoin proof-of-storage-over-time rewards miners for providing diskspace online. Proof-of-loyalty however extends the concept even further - including activity like “reading updates”, “sharing updates” or any other activities that can be measured objectively and represented correctly on a blockchain platform.

Present Problems & Needs

Blockchain technology rewards community participation via issuing its own tokens. For example Bitcoin rewards the miners 12.5 Bitcoins for implementing the **proof-of-work** (PoW) consensus algorithm - in the process, securing the Bitcoin transactions.

A **proof-of-work** (PoW) system (or protocol, or function) is an economic measure to deter denial of service attacks and other service abuses such as spam on a network by requiring some work from the service requester, usually meaning processing time by a computer. For doing that work, the service requester is rewarded with crypto tokens as a result.

In the case of Bitcoin, the “work” is to find a hash with leading zeros - traditionally referred to as mining. After doing that, the “worker” would be rewarded with 12.5 BTC.

Already in the last few years, there has been evolution of 1 other protocol. The **proof-of-stake** (PoS) protocol has been used in Waves, Peercoin, Dash, PIVX Coin or Bitshares. Very soon, it would be implemented for Ethereum via the Metropolis update as well.

Proof-of-stake (**PoS**) is a type of algorithm by which a cryptocurrency blockchain network aims to achieve distributed consensus. In PoS-based cryptocurrencies the creator of the next block is chosen via various combinations of random selection and wealth or age (i.e. the stake).

Proof-of-stake has its own criticism as well, including the criticism of centralization. That is the user with the most tokens can control the whole network. It is basically against the spirit of decentralization.

The other problem is the “nothing at stake” problem, where (in the case of a consensus failure) block-generators have nothing to lose by voting for multiple blockchain-histories, which prevents the consensus from ever resolving. Because there is little cost in working on several chains (unlike in proof-of-work systems), anyone can abuse this problem to attempt to double-spend (in case of blockchain reorganization) "for free".

Putting the inherent flaws aside, there is still an issue that conventional proof-of-work does not define an economic measure other than the traditional economic measure of computing hashes being expensive (in time & money).

While the conventional bitcoin **proof-of-work** (PoW) and **proof-of-stake** (PoS) protocols have given rise to the whole blockchain revolution, there are further protocols / methods we can define to further reflect the economic measure.

Proof of Loyalty

Here comes **proof-of-loyalty** (PoL). Proof-of-loyalty mainly combines 3 mechanisms;

1) Proof-of-work (PoW)
2) Proof-of-stake (PoS)
3) Proof-of-time/age (PoT)

We would have a special focus and innovation of what constitutes work in the **proof-of-work** (PoW).

In the context of various blockchain platform, “work” can constitute the following activities;

1) Increasing awareness of blockchain platform
2) Eliminating foul play
3) Voting / contributing to blockchain platform.
4) Getting more users to use platform
5) Getting merchants as service providers
6) Serving as offchain loyalty-explorer.
   See next section for technical requirement

These economic activities are inherently useful for the blockchain platform more than just providing hashing capabilities.

To continue on the tradition of transparency and decentralization spirit of blockchain, there is a need for a blockchain explorer to audit and keep transparent of the economic activities that are recorded that form the basis of token distribution.

For most blockchain platforms, there is little allowance for data you can put within a single blockchain transaction that allows you to keep track of the economic activities to be recorded for the purpose of rewards disbursements of tokens.

There is a need for an offchain explorer that allows greater amount of data to be recorded.

OffChain Proof of Loyalty Blockchain Explorer

The Waves blockchain transaction only allows the attachment of data of 140 characters. Barely enough to contain any meaningful indication of what economic activities have been carried out during that blocktime.

However, 140 characters is sufficient to store a sha256 hash.

For example, this proof of loyalty explorer containing the following data;

Fan ID	Activity	Media identifier
fan1	Reading	id1
fan2	Sharing media	id2
fan1	Online node	nil
fan3	Install Wallet	nil
fan2	Online node	nil

Equivalent sha256 hash :

996E8CAFD0D4CCF07DE70A8DB235C241C578DCAF5F6380F92923A4C7DEC65B01

One of the advantages of using the computing hashes to store on the main waves blockchain is that on 1 hand it does not tax the main blockchain which is often used by many users. At the same time the offchain records has no such arbitrarily defined block size of 1 MB like Bitcoin. So this proof-of-loyalty is intended to be scalable right from the start.

After computing its sha256 hash the record is practically untamperable. As a single modification of the recorded economic activities would change the hash drastically and in the process protects the authenticity of the record of economic activities.

Hashing case study generated from

http://passwordsgenerator.net/sha256-hash-generator/

Formatting the text as follows;

fan1,Reading,id1
fan2,Sharing media,id2
fan1,Online node,nil
fan3,Install Wallet,nil
fan2,Online node,nil

To leverage on the waves blockchain (although the principles here are applicable to any blockchain platform that allows the recording of data on a per blockchain transaction standpoint) to record this **proof-of-loyalty** economic activities.

You can designate a waves address, for example;

3P2UGNo2PRM7NoXE23hJezjEsAPFQbHmM2x

Technically, this waves address can be any other waves address designated for each specific token representing a specific blockchain platform intending to leverage on **proof-of-loyalty**.

The token that would then be used for the purpose of recording on the waves blockchain to achieve the purpose of blockchain - untamperable and totally visible record of activity.

For example, LiberaCoin is a coin that cuts the red tape, bringing back the passion for artistic creators. LiberaCoin would be sent to the above designated waves address with the following Attachment:

"<blockheightofproofofloyalty>, 996E8CAFD0D4CCF07DE70A8DB235C241C578DCAF5F6380F92923A4C7DEC65B01"

Without the quotes.

The offchain Proof of Loyalty chain would be kept as decentralized records by willing provider of that diskspace and also web / network services.

The Offchain Proof of Loyalty Blockchain Explorer would access this record via a pointer as follows;

https://<offchainurlprovider>/<blockheight>

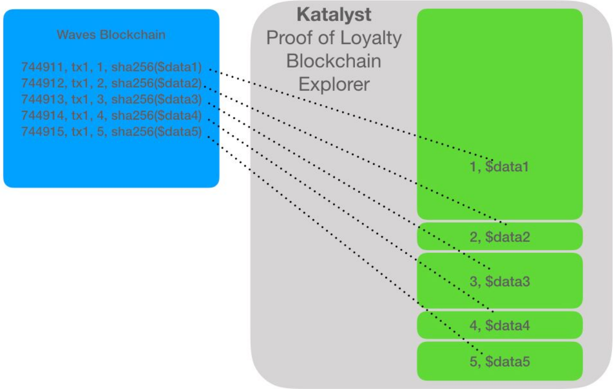
Which would of course display this record;

Fan ID	Activity	Media identifier
fan1	Reading	id1
fan2	Sharing media	id2
fan1	Online node	nil
fan3	Install Wallet	nil
fan2	Online node	nil

At the end of displaying that record a **sha256 hash** would be generated for the above mentioned record. Which is in this case;

996E8CAFD0D4CCF07DE70A8DB235C241C578DCAF5F6380F92923A4C7DEC65B01

This generated hash is then compared to the hash recorded on the waves blockchain. If the hash matches, then the offchain proof-of-loyalty (PoL) blockchain explorer display the right record and that record has never been modified or tampered with before.


Waves Blockchain
744911, tx1, 1, sha256($data1)
744912, tx1, 2, sha256($data2)
744913, tx1, 3, sha256($data3)
744914, tx1, 4, sha256($data4)
744915, tx1, 5, sha256($data5)
Katalyst
Proof of Loyalty
Blockchain
Explorer
1, $data1
2, $data2
3, $data3
4, $data4
5, $data5

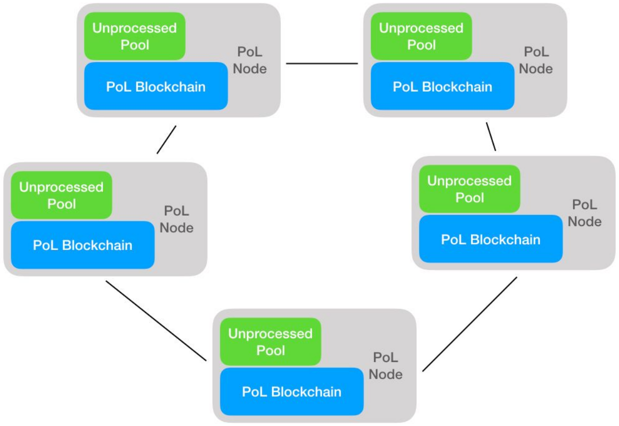
Above figure is a simplified guide to understanding how the hash of the Proof-of-Loyalty (PoL) blockchain. The blocksize in the Proof-of-Loyalty (PoL) blockchain does not have to be limited to a small size like Bitcoin.

Implementing this way allows the implementation of proof-of-loyalty by leveraging on the strong protection afforded by an underlying blockchain platform. Although waves is denoted as an example in this white paper, this principle should be cross applicable in all blockchain platforms.

As far as as theoretical flaw of this method, the only potential criticism is hash collision. That is, 2 totally different record of economic activities producing the same hash. In the first ever [1] cryptographic hash collision of SHA-1, it would take 6,610 years of processor time, and SHA256 would take exponentially longer time than that. If utilizing longer hashes like SHA512, would mean that the computing needed to repeat a collision is in the order of millions if not billions of years. In fact, it is probably multiple orders of magnitude greater than billions of years.

This part completes the theoretical part why the Proof-of-Loyalty (PoL) would function as a untamperable record of activities. The next step is to detail the software framework and the hardware infrastructure that would realize the Proof-of-Loyalty (PoL).

Specific Hardware Infrastructure & Software Framework of Proof-of-Loyalty (PoL)



Each Proof-of-Loyalty (PoL) node is intended to run on an extremely stable Ubuntu Linux systems. The Proof-of-Loyalty (PoL) software is going to be developed in Java programming / C programming language, so the node can technically run on Mac OSX as well.

While Java runs also on Windows, we recommend against running the Proof-of-Loyalty (PoL) software on a Windows machine as Windows machines are not as stable as Ubuntu Linux or Mac OSX.

The full Proof-of-Loyalty (PoL) node would perform the following functions;

1) Election of a Leader Node to Generate the next block in the Proof-of-Loyalty (PoL) blockchain.

2) Leader node to add Unprocessed Pool to the Proof-of-Loyalty (PoL) Blockchain.

3) Broadcast to other Proof-of-Loyalty (PoL) Node that block creation has completed

4) Sychronization of Proof-of-Loyalty (PoL) Blockchain from Leader Node to other Nodes

5) Sychronization of Unprocessed Pool between all the Nodes so that they can be created in the next Block time.

6) Fee Accounting

7) Wait for Leader Election Protocol, go to Step 1.

The detailed description of every step is as follows;

1) Election of a Leader Node

Each node would be elected via a Proof-of-Stake (PoS) consensus to generate the next block of activities to be recorded for the Proof-of-Loyalty (PoL).

The consideration of stake in the Proof-of-Stake (PoS) consensus would be primarily Katalyst & OToken the node has. The node's balance of Katalyst or OToken would be recorded in the Proof-of-Loyalty (PoL) blockchain - when it also records activities of token owners outsourcing their tokens to our nodes for block generation.

The following pseudo code is learned from various other Proof-of-Stake (PoS) blockchain projects;

Leader election protocol

- Nodes willing to participate in the election must declare themselves as "active"
- Based on the last block you get **BASE_TARGET** and **GENERATION_SIGNATURE**.
- For each Active Node calculate SHA256 (**GENERATION_SIGNATURE**, **PUBLIC_KEY**) as **HIT** value.
- For each Active Node calculate **BASE_TARGET** * **ACCOUNT_BALANCE** as **STATIC_TARGET** value.
- Node with lowest **HIT**/**STATIC_TARGET** ratio will forge the next block.

2) Leader node to add Unprocessed Pool to the Proof-of-Loyalty (PoL) Blockchain

Upon leader election and broadcasting to the rest of the nodes on the election result. This election result would also be put to the Unprocessed Pool;

For example in a 3 node setup (for illustrative purposes)

Node ID	Hit / Static	Blockheight
Node 1	0.04	11456
Node 2	0.57	11456
Node 3	0.12	11456

For the election of Node 1, one more transaction would be added to the Proof-of-Loyalty (PoL).

Transaction ID	Node ID	Blockheight
Leader Selection	Node 1	11456

The original Unprocessed Pool together with the the records here would be added formally to the Proof-of-Loyalty (PoL) Blockchain.

After the successful addition, add to the Unprocessed Pool.

Transaction ID	Node ID
Block Generation	Node 1

The steps in this action must be completed in 1 atomic step. The programmer do note when implementing this part must program in such a way that if these steps do not get completed it must rollback.

This is to take into account if there is a massive disaster concerning the node (hardware breakdown, asset seizure, nuclear attack / explosion in the vicinity of the data centre containing the node), then it must rollback upon switching on again.

3) Broadcast to other Proof-of-Loyalty (PoL) Node that block creation has completed

After completion of block, it has to broadcast to all the nodes that it has completed block generation, and note the steps down in the Proof of Loyalty.

The other nodes upon receiving the notification from the leader node shall make a note in its Unprocessed Pool that it has received a notification on the block generation from the leader node.

However, if the leader node is destroyed in a nuclear attack / explosion or asset seizure, then the decentralized nodes all over the world must be able take over.

Programmatically, the leader node is assumed to be taken out if after being elected as leader node, and did not reply on block generation completion in a designated time.

Given the notification of block generation completion does not by itself require huge amount of data to be transferred (typically it would be less than 100 bytes). The size of communication would not increase even if the future block size is tremendously huge (e.g. > 100mb).

Allocating a time limit of 5 minutes for Leader node to feedback completion of block generation should be sufficient.

Programmatically though, this time limit should be set as a macro and is changeable (pending voting rules, etc) to reflect future changes.

Beyond the time limit when the leader node is not able to feedback on the completion, the nodes participating in that round's election must decide (depending on stakes, in the case some nodes receive notification and some nodes do not receive notification) to invalidate the previous election and goes back to step 2) to elect another leader and start a new round of block generation and validation.

4) Sychronization of Proof-of-Loyalty (PoL) Blockchain from Leader Node to other Nodes

After broadcast has been made on the success of block generation, the next step is the synchronize the new addition to the rest of the nodes. The synchronization to the rest of the nodes must be further confirmed that the block generated and sha256 hash generated tally with the hash stored on the Waves blockchain (or any other blockchain).

This ensures the synchronization is validated to be the actual data added during that said blockheight.

5) Sychronization of Unprocessed Pool between all the Nodes so that they can be created in the next Block time.

The Unprocessed Pool shall be synchronized before it is time for leader election again. This is to ensure whatever nodes being elected to generate the next block, all the activities in the Unprocessed Pool shall be processed in the least time possible.

The synchronized activity shall still note the originating node so that the node can be rewarded via Proof-of-Loyalty (PoL). Recorded activity shall be removed from the Unprocessed Pool.

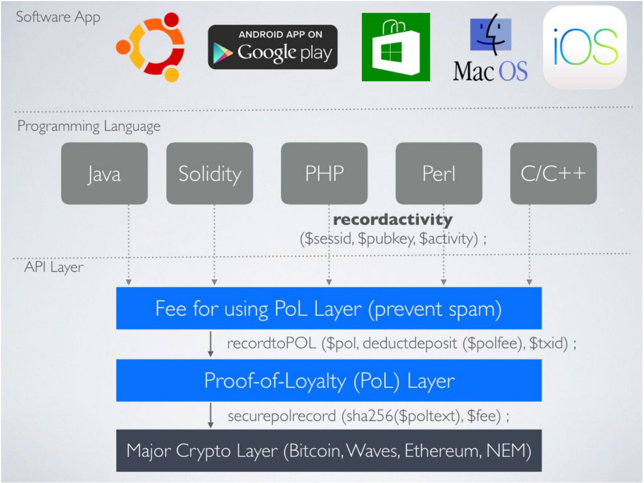
6) Fee Accounting

Fee Accounting is to be done for users of the Proof-of-Loyalty blockchain. For a start, it should be set as free for testing purposes. As it gets more mature, fee can be adjusted based on voting. The fee could be an established crypto or any token as a fee, as long as the node owners are willing.

7) Wait for Leader Election Protocol, go to Step 1.

The block time can be arbitrarily set to 10 minutes for the time being.

Proof-of-Loyalty (PoL) as Building Blocks for Blockchain Community & Business Users


Software App
ANDROID APP ON Google play
Mac OS
iOS
Programming Language
Java
Solidity
PHP
Perl
C/C++
recordactivity
($sessid, $pubkey, $activity) ;
API Layer
Fee for using PoL Layer (prevent spam)
recordtoPOL ($pol, deductdeposit ($polfee), $txid) ;
Proof-of-Loyalty (PoL) Layer
securepolrecord (sha256($poltext), $fee) ;
Major Crypto Layer (Bitcoin, Waves, Ethereum, NEM)

The white paper so far has mentioned only about the API Layer and fee collection. Proof-of-Loyalty (PoL) blockchain recorder is intended to be a very light layer with no other validation software routines that may tax the server's Central Processing Unit and also the hard disk accesses.

The main reasons are that to keep the Proof-of-Loyalty full node to be resource light as possible. At the same time, designing it that way also allows the blockchain community and business users to be as flexible as possible as how they want to use the Proof-of-Loyalty (PoL) for business application.

The Proof-of-Loyalty (PoL) layer functions on the API Layer. This layer can be designed as the building block of all existing software applications, and all future and potential software applications. Existing and to be developed applications can just call the functions defined in the simple API developed by Katalyst and leverage of Proof-of-Loyalty (PoL) in less than a few hours of software development.

An existing ride sharing app (e.g. Uber, Didi) for example could add on our software routine and record activities on Proof-of-Loyalty (PoL) to instill greater consumer confidence by the transparency of how it gives out discount or driving vouchers.

It could also instill more partner confidence by how their driving record is objectively stored on the Proof-of-Loyalty (PoL) blockchain. Upon disbursement of rewards or realization of stock options, the Proof-of-Loyalty (PoL) functions as an untamperable record of contribution.

A blockchain platform like OToken could use Proof-of-Loyalty (PoL) to keep track of how often are users online and whether they are keeping themselves updated with the news shared. Also additional rewards can be given for sharing of news on various social media channels. For example, upon detection that an update has been shared on facebook and clicked upon additional rewards can be issued to the user sharing.

Other than allowing other software applications to leverage on the Proof-of-Loyalty (PoL) blockchain, we ourselves are also developing some software that leverage on our own Proof-of-Loyalty (PoL) blockchain.

For example, we are developing the following software components that are crucial for the ecosystem of Katalyst. They are as follows;

1) **Decentralized Voting** - We believe a healthy blockchain community must have a mechanism in which votes are gathered from the community and recorded in the Proof-of-Loyalty blockchain.

Decentralized voting is detailed in another specific white paper about the subject matter. If interested to know more you may proceed to https://katalystcoin.com

2) **Zerotrust Cryptographic Gateway** - As there are more ICO projects launched on multiple blockchains (like Waves, NEM & Ethereum), we believe we have got to develop protocols and software for the gateway so that such exchanges can be done via the

gateway which would be recorded on the Proof-of-Loyalty blockchain.

Zerotrust Cryptographic Gateway is detailed in another specific white paper about the subject matter. If interested to know more you may proceed to https://katalystcoin.com

3) **Immutable File System** - The immutable file system would store files, images, videos that are meant to be immutable. Apps can be programmed / website can be setup to display the files and then checked with the Proof-of-Loyalty blockchain to verify authenticity.

Where applicable, this system can be used to store more complicated records like images, audio files, etc. It can be used for the purpose of storing scanned contracts, certificates, land certificates, deed of guarantee. Important paperwork that needs the untamperability of the Proof-of-Loyalty blockchain.

The detailed description of Immutable File System would be published in due time.

4) **Block Rewards System** - The Bitcoin proof-of-work protocol rewards the miner 12.5 BTC on average about every 10 minutes. Plus the transaction fees that are to be earned every 10 minutes from new transactions. That elegant Proof-of-Work (PoW) rewarding system drives the Bitcoin protocol.

Since the Proof-of-Loyalty (PoL) Blockchain is supposed to allow different users to be able to define their own specific rewards system. We have plans to make it easy for people to create their own rewards plan by minimal programming, and hopefully with just a clicking on a few buttons and guided according to their preferences.

For those blockchain projects / business users who have more programming capabilities we also provide them with an API that allows them to do recording easily. While they can just focus on designing their own blockchain platform with their own customized rewards system.

Taking for example, OToken. For example, if the following activities are recorded;

Fan ID	Activity	Media identifier
fan1	Reading	id1
fan2	Sharing media	id2
fan1	Online node	nil
fan3	Install Wallet	nil
fan2	Online node	nil

Rewards can be given every blocktime. Allowances may be made by only giving out the reward if it supersedes certain value for some blockchain platform to save on transaction fees.

For illustrative purpose, let’s say we determine the rewards to be given out as follows;

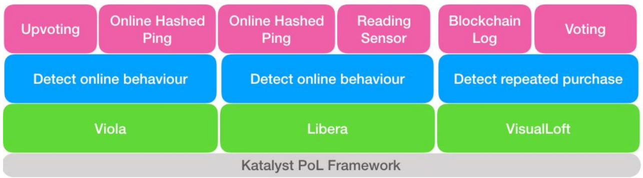
Activity	Rewards in OTokens
Reading	A number of OTokens
Sharing media	B number of OTokens
Online node	C number of OTokens
Install Wallet	D number of OTokens

So the reward given to the above different users during that blockheight would be as follows;

Fan ID	OToken Rewards
fan1	A + C
fan2	B + C
fan3	D

This is one example of how this can be designed. As to how it all can be designed, it is only limited by creativity.

Software Design Framework



This is a software stack to various other coins in the plans using Proof-of-Loyalty (PoL) blockchain. By structuring a Proof-of-Loyalty (PoL) blockchain that focuses on recording and allowing the rest of the stack to focus on their specific data / record validation and even the various methods of collecting information it allows for greater innovation and also allows more people to be a part of this process.

Innovation - Distributed Exchange

With Proof-of-Loyalty (PoL) blockchain giving its users' tokens value and utility, it would be demanded by their respective users and users of other blockchain to trade it to benefit from its value and utility.

We are developing a distributed exchange by the name of Barty Exchange. It would be used worldwide by many users to trade their blockchain inventories.

In Singapore, the crypto inventory (often misnamed as cryptocurrency by popular media and the public) exchange, Barty is already legally allowed to run basing on prevailing laws.

Innovation - Staking as a Business Opportunity

There are many investors who are investing in the conventional Proof-of-Work (PoW) bitcoin or altcoins mining. This method of mining is capital intensive and energy inefficient.

Basing on present estimates (25 Nov 2017) 0.13% of the global energy supply is going to support bitcoin mining. That is a lot of energy resources being used to support just 1 singular economic activity.

Not to mention to be effective in Proof-of-Work (PoW) mining you have to make capital expenses in highly specialized hardware to speed up your mining. Not to mention that mining in general is competitive in nature and the more miners are in the market, the lesser rewards in Bitcoin you may receive. Unless the price of Bitcoin continues to increase, the Bitcoins the miners would get decreases by the month and it would make it economically untenable to carry on.

Proof-of-Loyalty (PoL) mining by comparison is energy efficient. It is only dependent on your stake. This absolves the mining to be high on capital expenses - there is no expensive hardware to buy to make Proof-of-Loyalty (PoL) mining more effective and efficient.

Buying the stake may be also expensive but during the process of staking and end of staking, you can easily recover the cost of buying the tokens by selling it on our Barty Exchange. Compare this to Proof-of-Work (PoW) mining, the hardware's resale value would not be high

Resources

[1] 'First ever' SHA-1 hash collision calculated. All it took were five clever brains... and 6,610 years of processor time
https://www.theregister.co.uk/2017/02/23/google_first_sha1_collision/