CAMELINK - THE TRUSTLABEL PROOF OF STAKE BLOCKCHAIN FOR A FAIRTRADE WORLD

Bryan T. and the team
Digital Unicorn & Camelcoin

V0.0.3
December 2021

Abstract

In this paper, we present an overview of the architectural design of CameLink technology and solutions. CameLink is a Proof-of-Stake (PoS) blockchain that offers minimal transaction fees, quick confirmation times, fast validation, and randomization for further security. We propose a PoS-based blockchain protocol with a fair voting mechanism, strict security guaranteed, and quick finality in particular.

1. INTRODUCTION

Legal Disclaimer

Nothing in this White Paper is an offer to sell, or the solicitation of an offer to buy, any tokens. CameLink is providing this White Paper primarily to solicit public input and comments. Nothing in this White Paper should be construed or interpreted as a guarantee or promise as to how the CameLink business or tokens will evolve, or as to the tokens' utility or worth. This White Paper covers current goals, which CameLink reserves the right to change at any time, and the success of which will be determined by a variety of factors beyond CameLink's control, such as market-based factors and factors in the data and cryptocurrency industries, among others. Any predictions for the future are exclusively based on CameLink's examination of the concerns raised in this White Paper. That assessment could turn out to be erroneous.

2. MOTIVATION

The market for camel products has a huge potential. Unfortunately, there is a lack of public awareness and much of the potential of the world's herds remains and will remain untapped without structural change, alliance, universal promotion and official recognition.

CameLink's goal is to construct and develop a cryptocurrency (Camelcoin), design its own blockchain network (CameLink Blockchain), and make camel products more democratized and recognized around the world. This three-pronged approach will help to support the economic development of all activities involving camelids, stimulate innovation in all sectors of these activities (food, cosmetics, tourism, textiles, sports, and so on), improve the global position of these markets, and create new income-generating activities in areas where they are needed.

CameLink will ensure the establishment of a quality label as well as speedy advancement in research and innovation through the use of a trustworthy, up-to-date, and tamper-proof database. CameLink will also be an international laboratory that will make significant progress in the areas of camel milk preservation, therapeutic use of camel urine against cancer, and other autoimmune diseases in various processes (capsules, powder, creams, etc.), and research on the use of camel milk in the fight against autism in collaboration with the African Agricultural Research Centre. The usefulness of camel products against diabetes and cholesterol problems, as well as the usage of dheroua (camel hump fat) against all gastrointestinal and skin disorders.

The laboratory will become THE world reference. Every breeder or manufacturer of camel products on the planet will be able to have their products analyzed free of charge by DHL within 48 hours; the credit will go to the CamelCoin development funds to facilitate their commercial and international administrative procedures.

All results from international laboratories can be inserted, verified, validated, and approved in the CameLink blockchain to become irrefutable and absolute. The potential and virtues of camel products have been known for thousands of years. Unfortunately, today, this ancient knowledge is buried under a mass of data and, due to a lack of information and promotion, has remained cloistered within the herding communities and has spread very little to the informed Muslim community. We estimate that only 10% of camel milk production is used today for food and commercial purposes, the rest has been lost forever. IT IS TIME FOR A CHANGE!

CamelCoin and CameLink provide a platform for us to teach, educate, and communicate what blockchain technology is, how revolutionary it is, and how it will impact millions of people's daily lives. Many projects, fresh ideas, and useful information can be linked and therefore aided in their creation and launch to generate revenue-generating activities while giving medical, health, and wellness solutions to a large audience that does not find their needs met by the current market products.

3. CAMELINK

3.1. Overview

3.1.1. Blockchain networks

As shown in Fig. 1, transactions (data) are kept in blocks that form an ever-growing sequence (chain) that is shared across network participants. A blockchain's core units are transactions. When Alice wishes to send money to Bob, for example, she makes a transaction that includes her address as the input, her digital signature to verify that she is the one who performed the transaction, the amount of money to be sent, and Bob's address as the output. The transaction is subsequently broadcast to the rest of the network via Alice. After receiving the transaction, a miner, or consensus member, will validate it and incorporate it, together with other transactions received from other users, in a block. If the block is successfully mined, the miner will broadcast it to the network so that other nodes can verify it. This block will be merged into the chain and labeled as the latest block in the chain if it is successfully confirmed and identified as the first block mined after the last block in the chain. A block also contains a hash pointer, which is formed by hash functions and maps all of the block contents to the hash pointer. The hash functions' major characteristic is that they ensure that the chain is tamper-evident. It means that every change in past data will result in a different hash value in the next block, which can be tracked back to the genesis block, or the chain's initial block. Depending on the requirements of different consensus mechanisms, a block can also contain additional data. To reduce storage space, the transactions in a block can be stored in the form of a Merkle tree.

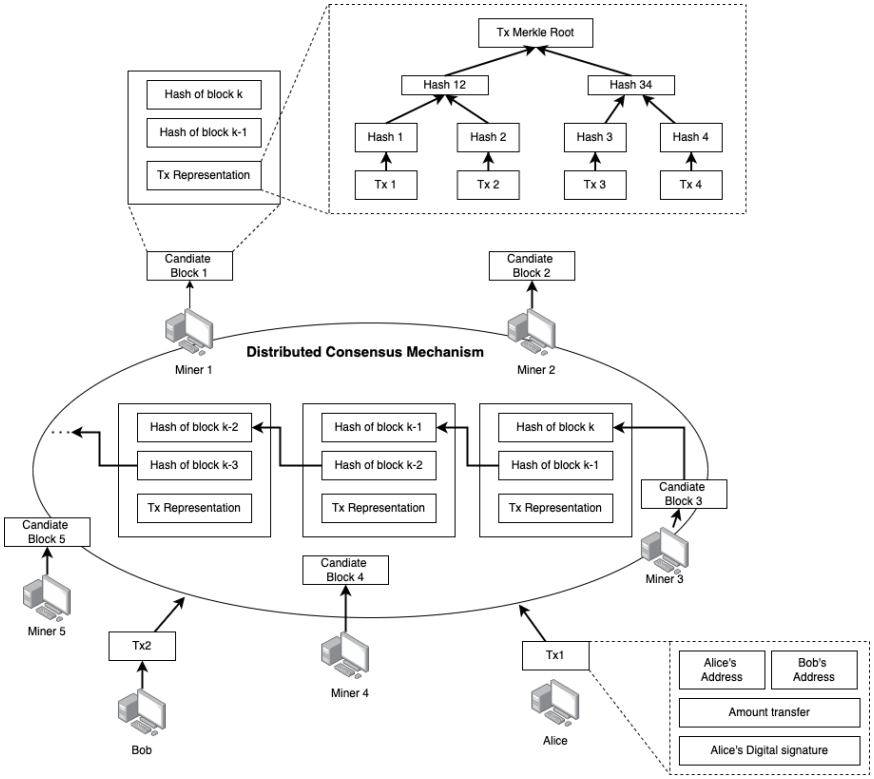


Figure 1: Blockchain networks

3.1.2. Benefits and applications

Although blockchain technology attracts a lot of attention due to the successful implementation of cryptocurrencies, its benefits extend far beyond. The key benefits of blockchain technology are as follow:

- Decentralisation: Blockchain networks are not controlled by a central controller. Thus, they do not have any single point of failure. Instead, all the nodes reach the agreement on the state of the network by participating in the distributed consensus mechanisms.
- Transparency: Data stored in a blockchain is visible to all network participants.
- Immutability: Once the data is stored in the blockchain, it is extremely difficult to be altered. Moreover, thanks to the distributed consensus mechanisms, the network can achieve consensus on the data even in a trustless environment.
- Security and Privacy: Using cryptographically secure mechanisms, the privacy and security of the network participants can be significantly enhanced. Users in the network use a pair of public and private keys for identification and verification. When a user makes a transaction, a digital signature is used, which can be easily verified but impossible to forge.

Blockchain technology, as we already know, has numerous uses in a variety of fields. Many businesses across a wide range of industries are attempting to integrate Blockchain technology into their existing systems, with the hope of reaping numerous benefits. The blockchain can be implemented in a variety of ways.

- Banking and Finance: Perhaps no industry stands to benefit from integrating blockchain into its business operations more than banking. Financial institutions only operate during business hours, five days a week. That means if you try to deposit a check on Friday at 6 p.m., you will likely have to wait until Monday morning to see that money hit your account. Even if you do make your deposit during business hours, the transaction can still take one to three days to verify due to the sheer volume of transactions that banks need to settle. Blockchain, on the other hand, never sleeps.
- Cryptocurrencies: Cryptocurrencies, e.g., Bitcoin, Ethereum, Cardano, are the most famous applications of blockchain technologies. By spreading its operations across a network of computers, blockchain allows Bitcoin and other cryptocurrencies to operate without the need for a central authority. This not only reduces risk but also eliminates many of the processing and transaction fees. It can also give those in countries with unstable currencies or financial infrastructures a more stable currency with more applications and a wider network of individuals and institutions they can do business with, both domestically and internationally.
- Supply Chain: Supply chain is the most important sector. Users can track the origin of products they bought. CameLink and Camelcoin as a currency take advanced in this industry

The use of blockchain technology is still in its early stages, but it is built on widely understood and sound cryptographic principles. Currently, there is a lot of hype around the technology, and many proposed uses for it. Moving forward, it is likely that the hype will die down, and blockchain technology will become just another tool that can be used [4].

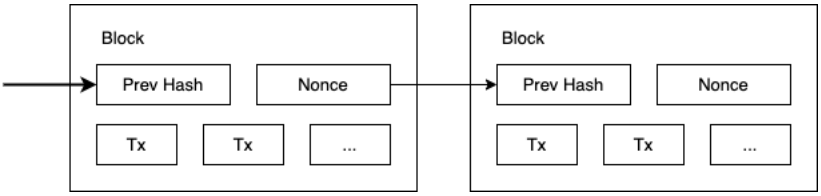
3.2. Consensus mechanism

Due to connection delay, i.e., Byzantine failures, nodes in a blockchain network can be faulty, exhibit arbitrary or malicious behaviors, or have disinformation. In such trustless contexts, the consensus process is the key component of a blockchain network, ensuring that every participant agrees on the state of the network. Other network functions, such as transaction addition and motivating players to behave appropriately, are governed by the consensus method.

3.2.1. Proof of work

We'll need to use a proof-of-work system similar to Adam Back's Hashcash, rather than newspaper or Usenet articles, to create a distributed timestamp server on a peer-to-peer basis. Scanning for a value that begins with a number of zero bits when hashed, such as SHA-256, is the proof of work. The average amount of work required is proportional to the number of zero bits required, and this may be verified by running a single hash. In our timestamp network, we use a nonce to increment until a number is found that gives the block's hash the requisite zero bits. The block cannot be modified without redoing the work once the CPU effort

has been consumed to meet the proof-of-work requirement. Because subsequent blocks are connected after it, changing it would necessitate reworking all subsequent blocks.



The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added. To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases [7].

3.2.2. Proof Of Stake Consensus

The proof of stake (PoS) paradigm is founded on the idea that the more invested a user is in the system, the more likely they are to want it to succeed and less likely they are to try to sabotage it. Stake is a cryptocurrency amount that a blockchain network user has put into the system (through various means, such as by locking it via a special transaction type, or by sending it to a specific address, or holding it within special wallet software). The coin can no longer be spent after it has been staked. The amount of stake a user holds is used to determine when new blocks are published in proof of stake blockchain networks. As a result, the likelihood of a blockchain network user publishing a new block is proportional to the quantity of staked cryptocurrency in the entire blockchain network.

With this consensus model, there is no need to perform resource intensive computations (involving time, electricity, and processing power) as found in proof of work. Since this consensus model utilizes fewer resources, some blockchain networks have decided to forego a block creation reward; these systems are designed so that all the cryptocurrency is already distributed among users rather than new cryptocurrency being generated at a constant pace. In such systems, the reward for block publication is then usually the earning of user provided transaction fees.

The methods for how the blockchain network use the stake can vary. Here we discuss four approaches: random selection of staked users, multi-round voting, coin ageing systems and

delegate systems. Regardless of the exact approach, users with more stake are more likely to publish new blocks. When the choice of block publisher is a random choice (sometimes referred to as chain-based proof of stake), the blockchain network will look at all users with stake and choose amongst them based on their ratio of stake to the overall amount of cryptocurrency staked. So, if a user had 42 % of the entire blockchain network stake they would be chosen 42 % of the time; those with 1 % would be chosen 1 % of the time. When the choice of block publisher is a multi-round voting system (sometimes referred to as Byzantine fault tolerance proof of stake) there is added complexity. The blockchain network will select several staked users to create proposed blocks. Then all staked users will cast a vote for a proposed block. Several rounds of voting may occur before a new block is decided upon. This method allows all staked users to have a voice in the block selection process for every new block.

When the choice of block publisher is through a coin age system referred to as a coinage proof of stake, staked cryptocurrency has an age property. After a certain amount of time (such as 30 days) the staked cryptocurrency can count towards the owning user being selected to publish the next block. The staked cryptocurrency then has its age reset, and it cannot be used again until after the requisite time has passed. This method allows for users with more stake to publish more blocks, but to not dominate the system – since they have a cooldown timer attached to every cryptocurrency coin counted towards creating blocks. Older coins and larger groups of coins will increase the probability of being chosen to publish the next block. To prevent stakeholders from hoarding aged cryptocurrencies, there is generally a built-in maximum to the probability of winning.

When using a delegate system to choose a block publisher, users vote for nodes to become publishing nodes, allowing them to create blocks on their behalf. The voting power of blockchain network users is proportional to their stake, so the higher the stake, the more weight the vote has. The nodes with the most votes become publishing nodes, allowing them to validate and publish blocks. Users on the blockchain network can also vote against a well-established publishing node in order to have them removed from the list of publishing nodes. Voting for publishing nodes is ongoing, and it might be difficult to stay on as a publishing node. Publishing nodes are constantly threatened with losing their status as publishing nodes, and thus their incentives and reputation, thus they are driven to not act maliciously. Users of the blockchain network can also elect delegates to participate in the blockchain's governance. Delegates will make suggestions for improvements and adjustments, which will be voted on by the blockchain network users.

Existing PoS protocols select staking nodes proportionally to their stake to form block-creating committees. Yet, they do not guarantee that selected committees will create blocks, since consensus may fail due to accidental or adversarial behavior. Thus, the perceived fairness in the distribution of rewards in proportion to the stake of participating nodes is actually violated. [13].

It is worth noting that a problem known as “nothing at stake” [9] may arise from some proof of stake algorithms. If numerous rival blockchains emerge at some point in the future, a staked user could act on any of them because it is virtually free to do so. This could be done by the staked user to improve their chances of receiving a payout. For long periods of time, this can result in numerous blockchain branches continuing to grow without being reconciled into a single branch.

The "rich" can more easily stake more of the digital assets under the proof of stake systems, earning themselves more digital assets; however, obtaining the majority of digital assets within a system to "control" is generally cost prohibitive. [8]

3.2.2.1. Fundamental background

Proof-of-Stake (PoS) protocols were created as a low-energy alternative to Proof-of-Work (PoW). Leaders are chosen based on their stakes, or contributions to the blockchain network, rather than computational capacity. The stake of a node, especially in the PoS consensus mechanism, is the number of digital tokens it owns or deposits, such as coins in cryptocurrencies. Instead of wasting a lot of energy searching for a new block like in PoW, a leader will be chosen based on its stakes to complete the mining operation and add a new block to the chain, as seen in Fig. 2. Many PoS-based blockchain networks, such as Cardano, Sp8de, and Tezos, have implemented the Follow-the-Satoshi (FTS) algorithm to imitate the stake-based leader selection process. In these networks, all the tokens are indexed. The FTS algorithm is a hash function that takes a seed (i.e., a string of arbitrary lengths such as the previous block’s header or a random string created by some other selected nodes) as the input. The FTS algorithm then outputs a token index. Using the index, the algorithm searches the transaction history to find and select the current owner of that token to be the leader. Therefore, the probability *pi* that node *i* is selected to be the leader in a network of *N* participants is:

$$p_i = \frac{S_i}{\sum_{j=1}^{N} S_j},$$

where S_i is the stake of participant i. This means that the more stake a node holds, the higher chance it is selected to be the leader.

PoS systems have a faster transaction confirmation speed than PoW processes, in addition to the advantage of low energy consumption. The confirmation of a transaction in a blockchain network is determined by two key factors: transaction throughput and block confirmation time. The transaction throughput of a network is the number of transactions per second (Tx/s) that it can process, which is critical to the network's performance, especially when there are numerous pending transactions. Tx/s can be calculated by:

$$Tx/s = \frac{Block_{Size}}{Tx_{size} \times Block_{time}}$$

3.2.2.2. Stake pools

Stake pools and stakeholders

In PoS networks, the chances of an individual stakeholder with a small stake size being chosen as the leader are slim. Furthermore, a node must be connected to the network at all times in order to participate in the consensus process, incurring an operational cost. As a result, minor stakeholders frequently pool their bets to boost their chances of winning blocks and share operational costs, leading to the establishment of stake pools. A stake pool is treated as a single node, similar to mining pools in PoW networks, and hence poses a concern of centralization in PoS networks. In particular, stakeholders in IoV networks, such as RSUs, are frequently required to perform additional responsibilities, such as processing carpooling information and responding to vehicle trust rating enquiries. As a result, RSUs in these networks may be more likely to join stake pools in order to lower their operational costs. In this section, we examine the stake pools from a game theoretical perspective to determine the strategic decisions of the stakeholders, and how these decisions affect the decentralization of the PoS networks.

System Model

Consider N stakeholders with stakes S = (s_1, . . . , s_N) and M stake pools with costs c = (c_1,...,c_M) and fees α = (α_1, . . . , α_M) in the network. The pool costs are charged for joining the pool and maintaining its operations. The pool's fee is the profit margin of the pool's owner, which is usually 3% in real-world stake pools, e.g Stakecube. When the stakeholder i invests an amount s_i^m in the pool m, the expected reward r_i^m is given by:

$$r_i^m = \rho_m \varphi_i^m (1 - \alpha_m) R - c_m e^{-s_i^m},$$

where ρ_m is the proportion of pool m's stake in the total network stake, φ_i^m is the proportion of player i's stake in the total stake of pool m, and R is the block reward. The pool charges a fee of α_m percentage from each stakeholder's reward and a cost of $c_m e^{-s_i^m}$. It is worth noting that the cost is inversely proportional to s_i^m, which incentivizes the stakeholders to invest more stake into the pool. Let N_{-i} denote the set of all the stakeholders except stakeholder i, the stake proportion of pool m is:

$$\rho_m = \frac{s_i^m + \sigma_m + \sum k \in N_{-i} \; s_k^m}{\tau}$$

where $\tau = \sum_{i=1}^{N} \sum_{m=1}^{M} s_i^m$ is the total stake of the network, $\sum k \in N_{-i} \; s_k^m$ is the stakes invested in pool m by all the other stakeholders except stakeholder i, and σ_m is the current stake of pool m. Thus, ρ_m is the chance that the pool m is selected to be the leader and can receive the block reward *R*. When pool m receives the reward, it calculates each stakeholder's share based on how much the stakeholder invested in the pool, which is:

$$\varphi_i^m = \frac{s_i^m}{s_i^m + \sigma_m + \sum k \in N_{-i}\ s_k^m},$$

for stakeholder i. The cost and fee of the pool are then deducted from each stakeholder's share before it is finally delivered to each stakeholder.

3.2.2.3. Smart contract

The term smart contract dates to 1994, defined by Nick Szabo as "a computerized transaction protocol that executes the terms of a contract. The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries.".

Smart contracts [6] extend and leverage blockchain technology. A smart contract is a set of code and data (also known as functions and state) that is distributed on the blockchain network via cryptographically signed transactions (e.g., Ethereum's smart contracts, Hyperledger Fabric's chaincode). The smart contract is executed by nodes in the blockchain network; all nodes that execute the smart contract must get the same outcomes, and the results are recorded on the blockchain.

Users of the blockchain network can create transactions that convey data to a smart contract's public functions. The smart contract performs a service by executing the right method with the data provided by the user. Because the code is stored on the blockchain, it is tamper obvious and resistant, allowing it to be utilized as a trustworthy third party, among other things. A smart contract can execute calculations, store data, reveal properties to represent a publicly visible state, and, if necessary, transmit funds to other accounts automatically. It isn't even required to carry out a financial function. The authors of this document, for example, have constructed an Ethereum smart contract that creates reliable random numbers in the public domain. It's worth noting that not all blockchains can support smart contracts.

The smart contract code can represent a multi-party transaction, typically in the context of a business process. In a multi-party scenario, the benefit is that this can provide attestable data and transparency that can foster trust, provide insight that can enable better business decisions, reduce costs from reconciliation that exists in traditional business to business applications, and reduce the time to complete a transaction.

Smart contracts must be deterministic, in that given an input they will always produce the same output based on that input. Additionally, all the nodes executing the smart contract must agree on the new state that is obtained after the execution. To achieve this, smart contracts cannot operate on data outside of what is directly passed into it (e.g., smart contracts cannot obtain web services data from within the smart contract – it would need to be passed in as a parameter). Any smart contract which uses data from outside the context of its own system is said to use an 'Oracle'.

For many blockchain implementations, the publishing nodes execute the smart contract code simultaneously when publishing new blocks. There are some blockchain implementations in which there are publishing nodes that do not execute smart contract code but instead validate the results of the nodes that do. For smart contract enabled permissionless blockchain networks (such as Ethereum) the user issuing a transaction to a smart contract will have to pay for the cost of the code execution. There is a limit on how much execution time can be consumed by a call to a smart contract, based on the complexity of the code. If this limit is exceeded, execution stops, and the transaction is discarded. This mechanism not only rewards the publishers for executing the smart contract code but also prevents malicious users from deploying and then accessing smart contracts that will perform a denial of service on the publishing nodes by consuming all resources (e.g., using infinite loops).

There may not be a requirement for users to pay for smart contract code execution in permissioned blockchain networks that support smart contracts, such as those that employ Hyperledger Fabric's chaincode. Other means of avoiding undesirable behavior can be used in these networks, which are based on having known participants (e.g., revoking access).

CameLink supports ordinary Solidity-based smart contracts via the Ethereum virtual machine at the moment (EVM). On CameLink, developers can design a smart contract using Solidity:

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.8.4;
contract Camelcoin {
    // The keyword "public" makes variables
    // accessible from other contracts
    address public minter;
    mapping (address => uint) public balances;

    // Events allow clients to react to specific
    // contract changes you declare
    event Sent(address from, address to, uint amount);

    // Constructor code is only run when the contract
    // is created
    constructor() {
        minter = msg.sender;
    }

    // Sends an amount of newly created coins to an address
    // Can only be called by the contract creator
    function mint(address receiver, uint amount) public {
```

```
        require(msg.sender == minter);
        balances[receiver] += amount;
    }

    // Errors allow you to provide information about
    // why an operation failed. They are returned
    // to the caller of the function.
    error InsufficientBalance(uint requested, uint available);

    // Sends an amount of existing coins
    // from any caller to an address
    function send(address receiver, uint amount) public {
        if (amount > balances[msg.sender])
            revert InsufficientBalance({
                requested: amount,
                available: balances[msg.sender]
            });
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit Sent(msg.sender, receiver, amount);
    }
}
```

Solidity is a popular programming language among developers, hence CameLink supports it as a smart contract standard.

3.2.2.4. Security

Distributed ledgers are known for their high levels of security. To construct transactions, participating agents use cryptography encryption. The integrity of transacting agents is ensured by public and private keys, as well as validation mechanisms that prevent manipulation. Cryptographic hashing functions, which generate unique identifiers with a regulated length independent of the input, are the foundations of blockchain security. Each hash is assigned to a block as an identifier and corresponds to the previous block's hash value. The hash function is also used in a consensus method to ensure that ongoing transactions are verified.

3.2.3. Comparison

Table 1: Consensus Mechanism Comparisons

	PoW	**PoS**	**Hybrid**
Leader selection	Based on hash rate	Based on stake	Depends on varant
Energy consumption	Significant	Negligible	Medium to negligible
Hardware requirement	High	None	Medium to none
Block generation speed	Slow	Fast	Medium to high
Transaction confirmation speed	Slow	Fast	Medium to high
Applications	Bitcoin, Ethereum, etc.	Cardano, **CameLink**, Algorand, etc.	Casper, Peercoin, etc.

The security of PoS protocols is influenced by a number of factors. Because the leader selection processes are emulated by voting rounds, when voters communicate their ballots to other participants, network synchronization is critical to the security of many PoS protocols. Because network delay and connection complexity make it impossible to guarantee that all messages are sent correctly in practice, network synchrony must be considered while assessing the protocol's security. Some PoS protocols have been shown to be secure when the network is partially synchronous, meaning that messages sent will arrive at their destinations within a given amount of time, or asynchronous, meaning that messages may not arrive at all.

The incentive mechanism, in addition to network synchrony, is critical to the security of a PoS consensus method. On the one hand, the payment scheme must reward block makers and validators in order to encourage consensus involvement. On the other hand, it must penalize harmful behavior and avoid different PoS-specific attacks, such as those that entail building a huge number of blocks because creating blocks in PoS is significantly easier.

Below, we discuss in more detail some emerging PoS-based protocols which have been widely implemented in practice, namely Ouroboros, Chains-of-Activity, Casper, Algorand, and Tendermint. Their core components, namely the consensus processes, and the protocols are then compared in Table 1.

The proof of stake (PoS) [1] consensus protocol was developed as an alternative to the proof of work (PoW) protocol in order to address scalability and environmental sustainability

problems. Furthermore, the energy usage from proof of work can be made to be 95-99% lower, resolving the environmental concern with proof of work. [3]

3.3. CameLink protocol

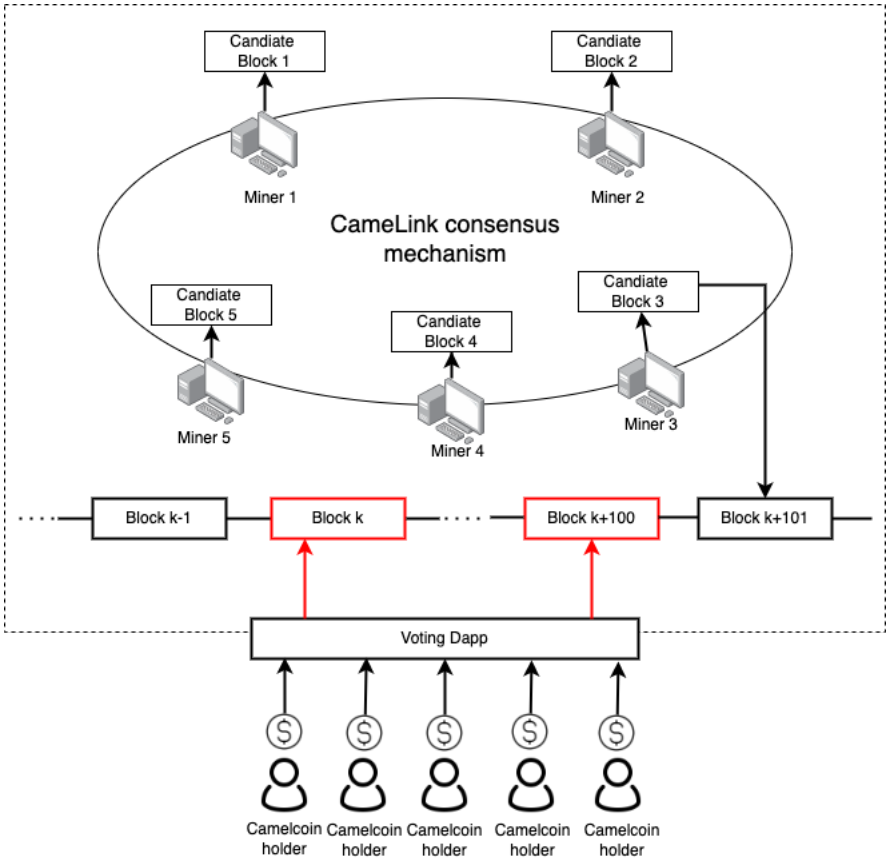


Figure 2: CameLink consensus mechanism

Camelcoin will be awarded to nodes that work hard in the system to create and verify blocks. Holders who vote for these incentivized nodes will additionally get Camelcoin in proportion to the amount of Camelcoin they deposited via ballots, as shown in Figure 2. The list of node candidates is constantly arranged based on the currencies that have been voted on. The nodes' performance will be monitored and reported to coin holders using three major metrics: CPU/Memory charts, which show the nodes' workload, the number of signed blocks, which shows their work performance, and the last signed block, which shows their most recent activity. Coinholders can unvote nodes with poor performance at any moment and allocate their votes to nodes with better performance. Because their voted coins are viewed as an investment by their supported nodes, coin holders have an incentive to do so. They should adopt a voting method to maximize their benefit from the investment. This simple approach keeps the system healthy by forcing nodes to compete for their place, resulting in the elimination of all weak nodes. As a result, only the most powerful nodes are voted on and have the opportunity to thrive.

We begin with Slush, a non-BFT protocol, then continue to Snowflake and Snowball, all of which are based on the same majority-based metastable voting mechanism. These protocols are more robust single-decree consensus mechanisms.

3.3.1 Slush: Introducing Metastability

A single-decree consensus protocol, inspired by epidemic or gossip protocols, lies at the heart of our method.

```
1: procedure ONQUERY( v, col')
2:     if col = ⊥ then col := col'
3:     RESPOND(v, col)
4: procedure SLUSHLOOP(u, col_0 ∈ {R, B, ⊥})
5:     col := col' // initialize with a color
6:     for r ∈ {1 . . . m} do
7:         // if ⊥, skip until onQuery sets the color
8:         if col = ⊥ then continue
9:         // randomly sample from the known nodes
10:        K := SAMPLE(N \u, k)
11:        P := [query(v, col) for v ∈ K]
12:        for col' ∈ {R, B} do
13:            if P.COUNT(col') ≥ α · k then
14:                col := col'
15:    ACCEPT(col)
```

Figure 3: Slush protocol. Timeouts elided for readability [12]

Slush, the most basic metastable protocol, lies at the heart of this family, as shown in Figure 3. Slush is not immune to Byzantine flaws, but it does serve as a model for the BFT [16] protocols that follow. For the sake of clarity, we'll describe Slush's operation by choosing between two opposing colors: red and blue.

A node in Slush starts out in an uncolored state. An uncolored node receives a transaction from a client and updates its own color to the one carried in the transaction before starting a query. A node sends a query message after selecting a tiny, constant-sized (k) sample of the network evenly at random. An uncolored node inherits the query's color, answers with that color, and launches its own query when it receives a query, whereas a colored node merely responds with its current color. If k responses are not received within a time bound, the node picks an additional sample from the remaining nodes uniformly at random and queries them until it collects all responses. Once the querying node collects k responses, it checks if a fraction ≥ αk is for the same color, where α > 0.5 is a protocol parameter. If the αk threshold is met and the sampled color differs from the node's own color, the node flips to that color. It

then goes back to the query step, and initiates a subsequent round of queries, for a total of m rounds. Finally, the node decides the color it ended up with at time m.

This straightforward approach has some intriguing characteristics. To begin with, it is nearly memoryless: a node keeps no state between rounds other than its current color, and no history of interactions with other peers in particular. Second, unlike traditional consensus procedures, which poll every member, each round includes randomly choosing a small, constant-size slice of the network. Second, even if the network starts in the metastable state of a 50/50 red-blue split, random sampling perturbations will cause one hue to gain a modest advantage, which will be built upon and amplified by subsequent samplings.

1: **procedure** SNOWFLAKELOOP(u, $col_0 \in \{R, B, \perp\}$)
2: col := col_0, cnt := 0
3: **while** undecided **do**
4: **if** col = $\perp$ then **continue**
5: K := SAMPLE($N \setminus u$, k)
6: P := [QUERY(v, col) for $v \in K$]
7: **for** col' $\in \{R, B\}$ do
8: **if** P.COUNT(col') $\geq \alpha \cdot k$ **then**
9: **if** col' $\neq$ col **then**
10: col := col', cnt := 0
11: **else**
12: **if** ++cnt > β then ACCEPT(col)

Figure 4: Snowflake [12]

Finally, if m is chosen high enough, Slush ensures that all nodes will be colored identically whp. Each node has a constant, predictable communication overhead per round, and we will show that m grows logarithmically with n.
The adversary can meddle with decisions if Slush is deployed in a network containing Byzantine nodes. If the correct nodes develop a predilection for one color in particular, the adversary can try to flip nodes to the opposing color in order to keep the network balanced. Because Byzantine nodes lack state, the Slush protocol lends itself to study but does not provide a strong safety guarantee by itself in the presence of Byzantine nodes. This is something we address in our first BFT protocol.

3.3.2 Snowflake: BFT

Snowflake augments Slush with a single counter that captures the strength of a node's conviction in its current color. This per-node counter stores how many consecutive samples of the network have all yielded the same color. A node accepts the current color when its counter exceeds β, another security parameter. Figure 4 shows the amended protocol, which includes the following modifications:

1. Each node maintains a counter cnt;

2. Upon every color change, the node resets cnt to 0;
3. Upon every successful query that yields ≥ αk responses for the same color as the node, the node increments cnt

When the protocol is correctly parameterized for a given threshold of Byzantine nodes and a desired ϵ-guarantee, it can ensure both safety (P1) and liveness (P2). As we later show, there exists a phase-shift point after which correct nodes are more likely to tend towards a decision than a bivalent state. Furthermore, there is a point-of-no-return beyond which a decision is unavoidable. The Byzantine nodes lose control after the phase shift, and the correct nodes begin to commit to adopting the same color, whp, past the point-of-no-return.

3.3.3 Snowball: Adding Confidence

The status of Snowflake is ephemeral: with each color change, the counter is reset. While the protocol can theoretically generate strong promises with minimum state, we will now strengthen the protocol by integrating a more persistent notion of belief to make it tougher to attack.

```
1: procedure SNOWBALLLOOP(u, col0 ∈ {R, B, ⊥})
2:     col := col_0, lastcol := col_0, cnt := 0
3:     d[R] := 0, d[B] := 0
4:     while undecided do
5:         if col = ⊥ then continue
6:         K := SAMPLE(N \u, k)
7:         P := [QUERY(v, col) for v ∈ K]
8:         for col' ∈ {R, B} do
9:             if P.count(col') ≥ α · k then
10:                d[col0 ]++
11:            if d[col'] > d[col] then 1
2:                 col := col'
13:            if col' ≠ lastcol then
14:                lastcol := col' , cnt := 0
15:            else
16:                if ++cnt > β then ACCEPT(col)
```

Figure 5: Snowball

Snowball adds confidence counters to Snowflake, which track the number of queries that have returned a threshold result for their related color (Figure 5). A node chooses a color if its confidence in that color exceeds that of other colors after a specified number of successive searches. The following are the differences between Snowflake and Snowball:

1. Upon every successful query, the node increments its confidence counter for that color.
2. When the confidence in a node's current color drops below the confidence value of the new color, the node changes colors.

Snowball is not only harder to attack than Snowflake, but is more easily generalized to multi-decree protocols. [13]

```
1: procedure CAMELINKLOOP
2:     while true do
3:         find T that satisfies T ∈ T ∧ T ∉ Q
4:         K := SAMPLE(N \u, k)
5:         P := ∑ v ∈K QUERY(v, T)
6:         if P ≥ α · k then
7:             c_T := 1
8:             // update the preference for ancestors
9:             for T' ∈ T : T' ∗← T do
10:                if d(T') > d(PT 0 .pref) then
11:                    PT 0 .pref := T'
12:                if T' 6= ρ_T'.last then
13:                    ρ_T' .last := T' , ρ_T' .cnt := 0
14:                else
15:                    ++ρ_T' .cnt
16:        // otherwise, c_T remains 0 forever
17:        Q := Q ∪ {T} // mark T as queried
```

Figure 6: CameLink: the main loop

Each node executes the protocol main loop, as shown in Figure 6. The node tries to select a transaction T that hasn't been queried yet in each cycle. If no such transaction exists, the loop will pause until T is updated with a new transaction. It then picks k peers and asks them questions. If more than α of those peers return a favorable response, the chit value is set to 1. The preferred transaction of each conflict group of transactions in its ancestry is then updated. T is then added to the set Q, ensuring that it is never requested by the node again. For simplicity, the code that selects additional peers if some of the k peers are unresponsive has been omitted.

3.3.4 CameLink: Adding a DAG

CameLink, our final protocol, generalizes Snowball and keeps track of all known transactions in a dynamic append-only Directed Acyclic Graph (DAG). The genesis vertex of the DAG is a single sink. Maintaining a DAG has two important advantages. First, because a single vote on a DAG vertex indirectly votes for all transactions on the path to the genesis vertex, it improves efficiency. Second, because the DAG, like the Bitcoin blockchain, intertwines the fate of transactions, it increases security. This makes it difficult to reverse previous decisions without the permission of correct nodes.

When a client generates a transaction, it specifies one or more parents, which are inextricably linked to the transaction and serve as the DAG's edges. The parent-child links encoded in the DAG may or may not match to application-specific dependencies; for example, a child transaction does not have to spend or have any relationship with the cash received in the parent transaction. All transactions reachable via parent edges back in history are referred to as ancestor set, and all children transactions and their offspring are referred to as progeny.

CameLink embodies a Snowball instance for each conflict set. Whereas Snowball uses repeated queries and multiple counters to capture the amount of confidence built in conflicting transactions (colors), CameLink takes advantage of the DAG structure and uses a transaction's progeny. Specifically, when a transaction T is queried, all transactions reachable from T by following the DAG edges are implicitly part of the query. A node will only respond positively to the query if T and its entire *ancestry* are currently the preferred option in their respective conflict sets. If more than a threshold of responders vote positively, the transaction is said to collect a *chit*, $c_{u\,T}$ = 1, otherwise, $c_{u\,T}$ = 0. Nodes then compute their confidence as the sum of *chit* values in the progeny of that transaction. Nodes query a transaction just once and rely on new vertices and *chits*, added to the *progeny*, to build up their *confidence*. Ties are broken by an initial preference for first-seen transactions.

4. USE CASES

Decentralized Finance (Defi)

Defi is quickly expanding beyond the scope of a single chain. With quicker speeds, higher throughput, and reduced fees, CameLink is entirely interoperable with Ethereum assets, apps, and tools.

- Asset Issuance
- Automated Market Makers (AMMs)
- Borrowing & Lending
- Decentralized Exchanges (DEXs)
- Derivatives
- Insurance
- Peer-to-Peer Payments
- Prediction Markets
- Stablecoins

Institutions, Enterprises, and Governments

CameLink is the most trustworthy platform for businesses, governments, and institutions. With compliance, data security, and other rulesets integrated into the foundation, you can launch assets, develop apps, and construct subnets with complete control over your implementation.

- Asset Issuance & Trading
- Debt Financing
- Digital Identity
- Document Tracking
- Fund Management
- Insurance
- Intellectual Property

- Lending
- Real Estate
- Supply Chain
- Trade Finance

Non-Fungible Tokens (NFTs)

For less than a penny, you can create your own NFTs in seconds. Prove ownership digitally and improve value flow. Create and share art, collectibles, and other items with all of the advantages and none of the drawbacks.

- Art
- Certifications and Licences
- Collectibles
- Credentials
- In-game Items
- Music

We have different kinds of products and projects built on CameLink Blockchain to meet the needs of the Camel-ecosystem, such as Camelcoin, which is a reward and use on the Camel-ecosystem, THE HUMP, which can track the origin of products, and also a staking application, CamelZone, an e-commerce platform, and so on.

5. DIVERSE ECOSYSTEM

5.1. CameLink explorer

CameLink Explorer works as a data search engine within a CameLink network. Users can examine different details relating to transactions on certain wallet addresses and blockchains using tools like Ethereum Explorer or Etherscan in a cryptocurrency environment. The sums exchanged, the sources and destinations of funds, and the status of numerous transactions are all examples of this. Users can access almost any data relating to transactions, wallets, and blockchains with the CameLink explorer, including rich lists and secret messages on the CameLink Blockchain.

On a technical level, blockchain explorer software uses an application programming interface (API) and blockchain node to draw various bits of information from a network. The software then uses a database to arrange the extracted data and presents it to the user in a searchable format. The explorer performs searches through an organized table on the database in response to user input.

CameLink Explorer's core functionalities allow users to search and examine data about recently mined blocks or transactions that have occurred on the blockchain recently. Some programs provide a screen that shows a live feed of blocks being mined as well as the data associated with each block.

5.2. Validator

Proof of stake takes a different approach to security by ensuring trust in a more old-fashioned currency: money.

Users build a node to participate in the blockchain verification process in proof of stake, which can be run by a single individual or a group of people working together. A node can be compared to a computer. The node must demonstrate its trustworthiness by storing a specified number of cryptocurrencies of the same sort as the blockchain it is confirming. Consider putting money in escrow or securing it with a security bond. Staking is the term for the process of locking away.

An algorithm selects one node for each block of transactions that needs to be verified, taking into account a variety of parameters to both reward those who have staked more bitcoin and prevent one node from gaining too much control over the process. That node is in charge of verifying the block and publishing or adding it to the chain.

After that, all of the other nodes are given some time to ensure that everything is in order. If a mistake or fraud occurs, the node that published the problematic block will have part or all of its piled coins erased. However, if everything appears to be in order, that node will be awarded extra bitcoin. This is the blockchain's security mechanism as well as a motivator for involvement.

Validators are users with accounts that have coins locked in a bond deposit by posting a bond transaction. These validators participate in the consensus protocol by broadcasting cryptographic signatures, or votes, to agree upon the next block. [2]

Blockchain technology's decentralized structure makes it so attractive and promising that more and more people are using it. The building blocks of a blockchain are known as nodes. They are in charge of data storage, however, this data must first be confirmed or verified on the blockchain network. This is when a validator is useful.

A validator verifies each incoming transaction in the same way as a banker verifies a transaction before it is processed.

A transaction can only be completed and its record can be added to the blockchain once its accuracy and legal authenticity are checked by a validator.

In the Proof-of-Stake mechanism, a validator determines whether or not a transaction conforms to the rules that deem it as valid. The entire process makes a blockchain network secure and transparent.

5.3. CameLink ecosytem

CameLink provides a platform for businesses to expand rapidly, safely, and efficiently by empowering entrepreneurs to create breakthrough solutions. Projects will be supported as part of the CameLink ecosystem to help them develop collaboratively throughout the ecosystem of connected products.

On the Camelink blockchain, Camelink is an open and efficient ecosystem. CameLink is dedicated to collaborating with ecosystem actors to solve global problems and add value to the community.

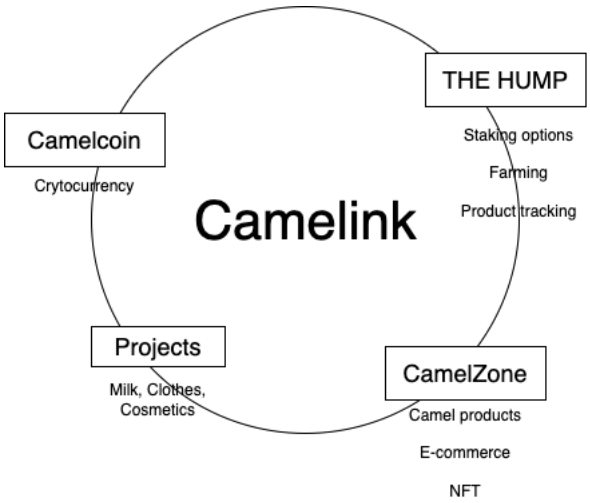


Figure 3: Current CameLink ecosystem

- **Camelcoin:** a way to finance projects worldwide in order to improve the economy of nomad's people and others. Camelcoin will provide the opportunity to acquire all camel related products such as as milk, cosmetics, jam, sheets etc... in a fast, cheap and secure way. Those products will be inscribed on the CameLink Blockchain (QR code, RFID etc...) To be traceable and secure, their method of production and fulfillment will incorporate a trust label for border and hygiene policies in order to ensure 100% reliability.
- **The HUMP:** on the Hump wallet and staking application, you can hold your Camelcoin and get rewards in CML and also track your orders.
- **CamelZone:** The products will be sold on peer-to-peer on the camelZONE e-commerce platform (first worldwide crypto-friendly e-commerce platform), and NFT property acts will be provided for valuables products (like whole race camels or others). We will find everything around camels on the platform, also for brewers, veterinary, racing clothes, etc... Also, all benefits of the camelZONE platform will be shared between camelcoins holders and a foundation.
- **Projects:** only 200 projects will be allowed to join the CameLink ecosystem, which will be for similar projects with similar values (economy, ecology, animal and human

respect, natural products, etc...) Programs include bees and honey, natural oils, botanicals, and so forth...

CameLink Blockchain will act as a trustlabel for those projects, and all products will be sold on the camelZONE platform, making the CamelCoin project the largest worldwide cooperative, completely decentralized, and independent, with all credit driving the CamelCoin price action to grow and assisting millions of people to better lives.

6. SUSTAINABILITY

Thousands of computers around the world are humming away at any one time, crunching complicated math problems that create and sustain bitcoin. This network is what makes bitcoin so appealing: it's decentralized, constantly on, and easy to trade. However, this means that the network is always consuming energy, something many bitcoin skeptics and detractors find objectionable. It's not only a bitcoin issue, either. Ethereum, like other cryptocurrencies and blockchains, has similar difficulties.

A POW architecture, which is currently used by both Bitcoin and Ethereum, involves a consensus method that necessitates a tremendous amount of computer power. Miners compete to solve complex arithmetic problems using computer components to validate electronic transactions on the blockchain, such as ASICs (application specific integrated circuits) in Bitcoin mining and graphics cards in Ethereum mining.

Proof-of-work (PoW) [5] requires the node to provide the computing power to solve a mathematical problem in order to append a new block to the blockchain. The most well-known user of PoW consensus is Bitcoin where miners are incentivized to stack up hardware and solve as many hash computations as possible to reap rewards in the form of coins. Because it demands massive amounts of computational power, cryptocurrency mining is most popular in countries where energy is cheap and accessible.

Proof of Stake (PoS) technology allows miners to avoid the energy-intensive cryptographic problem solving required in Proof of Work (PoW) systems. It allows owners to stake their tokens as collateral in exchange for incentives in order to validate transactions on the network by consensus in exchange for rewards, which is commonly done in big public pools.

In practice, this means that PoS does not require additional energy to verify trustworthiness, resulting in a significant reduction in the network's overall energy consumption.

7. RELATED WORKS

Bitcoin [7], also known as the first cryptocurrency on Blockchain, is a cryptocurrency that uses a blockchain based on proof-of-work (PoW). Following Bitcoin, a slew of alternative projects have emerged with the goal of addressing bitcoin's shortcomings, such as speed, gas fees, and so on. Bitcoin, unlike more typical BFT systems, offers a probabilistic safety guarantee

and presupposes honest majority computational power rather than a known membership, allowing it to scale to the internet. Bitcoin has low throughput (3 tps) and significant latency (5.6 hours for a network with 20% Byzantine presence and 232 security guarantees), despite being permissionless and resistant to adversaries. Furthermore, PoW necessitates a significant amount of processing power that is solely used for the purpose of ensuring security.

The Directed Acyclic Graph (DAG) is a DLT variation that has been presented as a blockchain alternative. Because they are arranged in a directed graph, the cooperating nodes in a DAG can cross-verify each other [10]. Inclusive blockchains [17] apply the Nakamoto consensus to DAG and define a framework for incorporating off-chain transactions in a consistent manner. To prune potentially harmful blocks in PHANTOM, participating nodes first identify an approximate k-cluster solution for their local block DAG, then topologically sort the remaining blocks to produce a total order. Conflux, like PHANTOM, uses PoW to finalize a linear order of transactions in a DAG structure, although it is more resistant to liveness attacks [15]. Even if all honest nodes are entirely synchronous, attackers with limited processing capacity can postpone transaction confirmation forever with high probabilities.

To reach consensus, several protocols use a Directed Acyclic Graph (DAG) structure rather than a linear chain [14], [11]. Hashgraph [14] is a leaderless system that uses randomized gossip to generate a DAG. It is a PBFT-variant that requires quadratic messages in expectation and demands full membership information at all times.

10. CONCLUSION

The architecture and foundation of the CameLink Blockchain were addressed in this study. The CameLink is lightweight, quick, scalable, secure, and efficient as compared to competing platforms that either use classical-style consensus protocols, which are intrinsically non-scalable or use Nakamoto-style consensus, which is inefficient and has significant running costs.

To be honest, this isn't the most impressive whitepaper we've ever seen. However, with the help of the CameLink Blockchain development team, we can create CameLink and turn it into the greatest Blockchain.

Apart from the consensus, CameLink also includes CameLink-related applications such as Explorer, Staking, and smart contracts written in Solidity, allowing developers to easily build on the CameLink.

11. ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to the Digital Unicorn team as well as Camelcoin's leader for providing us with the wonderful opportunity to be a part of this wonderful project in order to create sustainability and growth for the ecosystem of camel products, as well as for assisting us in conducting extensive research on camel products, for which we are extremely grateful. Second, we'd want to express our gratitude to Alex D. and other friends

and Bryan's former teachers, who assisted us greatly in completing this whitepaper within the time constraints.

REFERENCES:

[1] Jake Frankenfield, *"Proof of Stake (PoS)" on Investopedia* 2021

[2] Jae Kwon, *"Consensus without minting"* 2014

[3] Vitalik Buterin, *"Slasher: A Punitive Proof-of-Stake Algorithm" 2014*

[4] Dylan Yaga, Peter Mell, Nik Roby, Karen Scarfone, *"Blockchain technology overview"* 2018

[5] Wai Yan Maung Maung Thin, Naipeng Dong, Guangdong Bai, and Jin Song Dong, *"Formal Analysis of a PoS Blockchain"* 2018

[6] Nick Szabo, "*Smart contracts"* 1994

[7] Satoshi Nakamoto, *"Bitcoin: A Peer-to-Peer Electronic Cash System"*

[8] Nguyen, C. T., Hoang, D. T., Nguyen, D. N., Niyato, D., Nguyen, H. T., & Dutkiewicz, E. "*Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities.*" 2019

[9] Community, *"Ethereum Wiki"*

[10] Bahareh Lashkari And Petr Musilek, *"A Comprehensive Review of Blockchain Consensus Mechanisms"* 2021

[11] Imran Bashir, *"Mastering Blockchain Third Edition"* 2020

[12] Team Rocket, *"Snowflake to Avalanche: A Novel Metastable Consensus Protocol Family for Cryptocurrencies*" 2018

[13] Stefanos Leonardos∗, Daniel Reijsbergen∗, Georgios Piliouras∗, ∗Singapore University of Technology and Design, *"Weighted Voting on the Blockchain: Improving Consensus in Proof of Stake Protocols"* 2021

[14] Leemon Baird, *"The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance"* 2016

[15] Li, C., Li, P., Xu, W., Long, F., And Yao, A. C., *"Scaling Nakamoto Consensus to Thousands of Transactions per Second"* 2018

[16] Ethan Buchman, Jae Kwon and Zarko Milosevic, *"The latest gossip on BFT consensus"* 2018

[17] Yonatan Sompolinsky and Aviv Zohar, "*Phantom: A Scalable BlockDAG protocol"* 2021