Universal Business Consensus

Author: Antonio “Wakuu” Walker
Created: November 5, 2020
Revised: December 11, 2023
Version 1.1

Abstract 3
Introduction 3
Market Automation 4
Consortium Environment 5
Proof of Activity 6
Merkle Space Verifier 10
Tamper-Proof Distributed Ledger 11
Muvor Virtual Machine 13
Muvor Protocol (Secure) 14
EGAHN ($EGN) – Error Generative Adversarial Hypergraph Network 15
Conclusion 18
Library 20

Abstract. Blockchain technologies are a major milestone in creating the next generation of communications and technology. These technologies apply modern standards of cryptography and encapsulation to generate a linked list of encrypted blocks. Of these standards, the most well-regarded and hard-tested method is SHA256. Said method is renowned such that any additional encrypted bits are considered overkill by experts, uncrackable to any single modern adversary. To break this standard, adversaries require multiple powerful machines operating at much higher rates than that of any home PC. Many are familiar with this procedure, called mining, and entertain such an investment as ASIC miners, to acquire even a sliver of marginal return, even as its algorithmic bases goes to exponential complexity with each newly mined block, in ordinance with the validator's prover. While most see nothing wrong with these very functional, hardened methods, there are various concerns that must be addressed with the original standard of combining hashed objects into a linked list to provide viable security to users' data. One clear concern is a rogue miners' attempt to brute force any part of our linked list. Now, modern blockchains, while encrypted, do not provide as much anonymity as once thought. These standards have been proven crackable using current technology and are now being found as such through various methods, one being originally noted in [10] under incentives. This is the most undiscerning part of blockchain, as chain control would scale directly with CPU power.

1. Introduction

In a Post Quantum Reality, it is very well agreed upon that, even the simplest iterations of Quantum Computing will crack current encryption algorithms on the order of milliseconds. This has allowed innovation in the cryptocurrency space, proving to be prepared for Quantum Computing with storage mediums such as Quantum-Resistant Ledger [11]. In the scale of Quantum Computing, there derives new demonstrations of cryptography, namely Lattice –based, Symmetric –based, Code –based, Isogeny –based, and Multivariate encryptions. While QRL uses Symmetric –based, Sphincs Algorithm, we know that not all cryptography is made equally. In this paper, and on the Muvor Platform, we will be utilizing various Post Quantum Cryptography Standards, to ensure data security.

Today's industry has created a dichotomy in the way economic prosperity scales disproportional to relative strength of monetary notes. In depression, interest rates fall, causing more demand; less supply, and in a bull, less demand – for the dollar per se – and more supply. This disproportion has toppled some of the greatest economies our world has ever seen, due to inflationary tolls, during the most testing times, where notes are needed, only to the effect of increased prices.

Cryptocurrency provides a viable riposte to this issue. Regarded as digital gold, investors can purchase these derivatives with plans of holding, confident returns are imminent given popularity. Built atop the blockchain, these derivative currencies can be mined like gold, and are therefore sought to be commoditized as such. The more blockchain mined, the less likelihood of a miner to mine to a new block.

This is a common standard of a Proof of Work (PoW) consensus algorithm, where validators look at Central Processing Unit (CPU) hashing power to determine which machine

deserves the reward for creating a new block; giving vulnerable opportunity to whomever consistently controls the fastest machine. This problem has been mitigated by way of new standardization using a validation committee – a regulating body – who may own a large majority of the coin and gives orders to validate ownership of each newly rewarded block. This standard, dubbed Proof-of-Stake (PoS) consensus is flawed to a similar order of PoW; whomever owns the most stake, may go rogue, invalidating previous transactions, only to the favor of ones-self.

Hybrid Consensus protocols were first proposed by Rafael Pass and Elaine Shi, dubbed "Snow White" [12], various groups continue to further revise this model. Proof-of-Activity is a flexible hybrid consensus proposed for a fork-free blockchain [13], whilst providing another layer for secure validations, mitigating rouge adversaries. The main idea is, the chain will be validated through both PoW and PoS consensus, meaning that blocks will first be mined, and then validated by a regulatory committee. This paper will reflect on how Muvor provides these validations, with Wakuu Enterprises, Inc. as their lobbying body, or host. The Muvor Exchange is comprised of 5.67 Trillion total tokens, which is released in dynamic fashion. As the regulatory body, Wakuu Enterprises, Inc. will notify all stakeholders prior to releasing any additional tokens to the Exchange.

2. Market Automation [19]

Muvor Exchange is a dynamic stable coin. Financially, the goal of this derivative is to give upside to any stagnation. In doing so, our coin provides an effective holding environment, allowing both liquidity and upside. To achieve this, we employ a decentralized version of easing to allow complete control over your assets without losing out on returns.

We provide streamlines to our users, ending prisoners' dilemma we are accustomed to across global markets. The dynamic aspect of the Muvor Xchange currency offers an environment such that holding is expected and apricated. This eliminates uncertainty of our liquid assets during uncertain times; ensures the trusted upside during times of prosperity. Muvor drives your portfolio upward in times where liquidity is essential and provides positive returns years over especially in times of prosperity.

Since our currency is dynamic, it has the power to self-regulate, rather than being dependent on other currencies, or uncertain institutions. The self-regulation is provided in the shim; an autonomous mining adapter, that awards all participants dependent on the individuals' score. This provides equilibrium, in the form of equity, across the exchange, optimizing our behaviors, using uncertainty.

There are various examples of financial pooling to one ultra-secure currency, even in present markets, including the US Dollar, the Euro, Bitcoin, ERC, and other derivatives. Often, these mediums drive times of peace and prosperity, the preceding apart often looming war.

Currency, being the key to market activity, must therefore receive some sort of automation, devoid borders, with consistently low inflation, actively searching for what is best for utilization efforts. We achieve this using various shim's [s9], both custom, and generic. These shims track the overall wellness of Muvor, and make the necessary steps to continue upgrading, and staying on the bleeding edge.

The elements of the Muvor Xchange currency allow it to be a dynamic, multifaceted, paradigm. This also promotes an environment with minimal price fluxuation, where the host network may promote any means or combination of backing. For instance, Wakuu Enterprises supports an at least 42:2 backing, in ordinance with present currency standards, markets, and

modalities. This not only provides us with plenty of scalability, but also liquidity for clients. As the blockchain becomes more secure, raising closer to its target, we also must regard node scaling, as more users create network nodes. Currently, we expand the number of available tokens (Figure 1); shrinking for less users, aiding stability, liquidity, consensus, and when the host is involved, lack-of prisoner's dilemma.

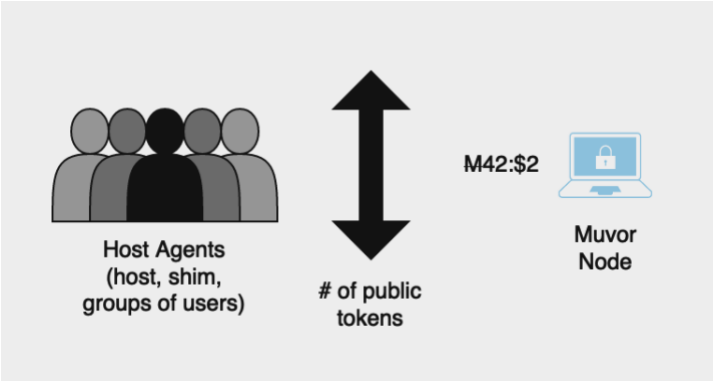

Figure 1: As more network nodes are added, hosts adjust the price targets for optimal consensus.

3. Consortium Environment [20]

Blockchain allows relative ease in the scope of data security, removing friction along key axes such as control, trust, and value. This not only opens secure infrastructure to easily share data, but also dilutes risk; since blockchain is immutable, all transaction logs and audits are tamper-proof. This simplifies many processes, by providing a single source of truth, from medical audits and secure documentation, to tracking music rights and licensure.

Muvor X currently consists of ***1,134,000*** total individual hard-tenant nodes, using various means of encrypted distributed databases, and communications, to enable fault-tolerant, consensus, preventing double spends, and other malicious behavior. Since these nodes are decentralized, each one must be known by the federation (Figure 2), resolving *Sybil* vulnerabilities. Even though the blockchain cryptographically signs each node in the chain, it is still a priority to maintain the highest level of security across the consortium.

Figure 2: Example communication between tenants on organization nodes.

In creating the consortium environment, your entity is the single source of truth for your data. This increases the tangibility of assets, meaning that clients who issue assets, licenses, contracts, among other agreements, will have more fine-grained control over how their asset propagates to its subscribers. While no single node in the mesh functions equivocally, each node does contain similar components that allow its function as a Muvor node (Figure 3). We implement various measures to ensure isolation, availability, scalability, and security within, and throughout the node's lifecycle. We found using homoiconic hash table lookups (Figure 4) to be the most efficient-effective means of discovery, so we should see great performance, even at scale.

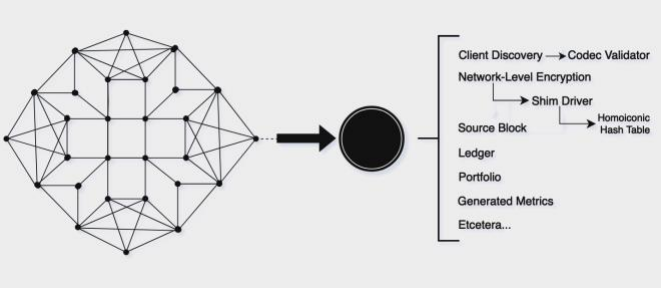

Figure 3: Example node mesh illustration

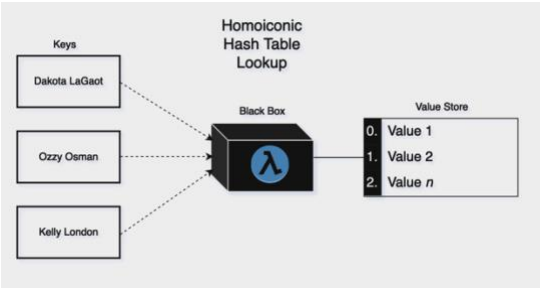

Figure 4: Hash lookup

Muvor provides necessary transparency while securing our data. This simplifies risk-analysis, audits, and asset propagation. In providing for the consortium, we remember the various real-time tasks that must take place, to continue operating as the highest levels. Things such as hard multi-tenancy, and hash lookup, allow our resources to be allocated elsewhere, allowing for more bandwidth, faster speeds, and cheaper fees.

4. Proof-of-Activity (PoA)

Muvor uses a variety of metaphysical rotating committees, to optimize its consensus. Each round, the block is submitted to the committee directly, and is validated at the bit level. Since committees for Proof-of-Work are purely digital, whenever a new block is mined, in example, through Grover's Algorithm [15] or Tani's Algorithm [14], the transaction will be submitted to our shim, where we run bit analyses, providing us with true Byzantine Fault Tolerance. Shuyang Tang et al [13] call this introduction of the mined block to a rotating committee, Generalized Proof-of-Work.

To increase the efficiency of our token, we introduce a Proof-of-Stake rotating committee within our host partnership network. These are individuals with whom the company entrusts with the decisive decision of when to release the next dynamic roll out, as well as how many tokens will be open to the public.

Proof-of-Stake is majority consensus. Our committee will require at least a 3/5 vote to consent to any dynamic movements on the exchange. These five (5) rotating entities extend a consortium from the host network or group, the shim registry, and three (3) randomly chosen consensus consorts. The shim again runs a variety of verification constructs, proving it to be safe, then validating with Proof of UTXO [17].

This concept is introduced originally by Tendermint [16] to efficiently process cross-chain transactions.

Proof-of-Activity brings the digital committee, from Proof-of-Work, and the rotating committee, from Proof-of-Stake, forming a hybrid consensus methodology that will offset the

vulnerabilities from each, and increase security. Multi-layered validators allow for much cleaner, fork-free consensus, with multiple correct answers. Then, at its truth, our shim verifies the block, and writes the hash. This can be repeated in many different fashions, as guest, client, organization, or an employee. We sign the transaction with its precise codec and pass it to the chain.

In using hybrid consensus, various codecs, and a variety of clients, we also use multiple algorithms to compute our consensus needs. In Proof-of-Work, we require an algorithm that will scale exponentially with each newly mined, loose block; this is hard-mined. We use a custom algorithm that increases its complexity with each additional block. This challenge was made to be very difficult, even for quantum computers, and even with multiple correct answers. Here's the penetration probability via Proof-of-Work shim:

Grover's Algorithm:

1. **Search Problem**: $P(success) = \sin^2(\frac{t}{2}\theta)$

   where t is the number of iterations and θ is the initial probability amplitude.

2. **Quadratic Speedup**: $P(success) = \sin^2(\frac{\sqrt{N}}{2}\ \theta)$

   This demonstrates the quadratic speedup of Grover's Algorithm in unstructured search.

3. **Probability of success over iterations**:

$$P(success) = \sum_{t=1}^{\infty} \sin^2(\frac{t}{2}\ \theta)$$

Pollard's Rho Algorithm:

1. **Random Walk**: $Expected\ time = O(\sqrt{N}\ )$

   The expected running time is sub-polynomial in the size of the factors due to the probabilistic nature of the algorithm.

2. **Probabilistic Nature**: The success of Pollard's Rho relies on a combination of random choices and the characteristics of the function f(x).

$$Expected\ time = \sum_{i=1}^{O(\sqrt{N}\ )} 1$$

Shor's Algorithm:

1. **Quantum Parallelism**: Shor's Algorithm leverages quantum parallelism to explore multiple possibilities simultaneously, providing an exponential speedup over classical algorithms.
2. **Quantum Speedup**: The time complexity of Shor's Algorithm is $O((logN)^3)$, showcasing its polynomial speedup over classical factoring algorithms.
3. **Superposition and Entanglement**: The algorithm uses superposition and entanglement to efficiently represent and manipulate quantum states during computation.

$$P(success) = \sum_{i=1}^{N} Complex\ Quantum\ Terms$$

Julia should be good for this…

```
# A challenge with increasing computational complexity
function pow_challenge(n)
    salt = rand(UInt8, 16)

    t_cost = 2^n  # Increase the number of iterations
    m_cost = 2^(16+n)  # Increase the memory cost
    parallelism = 1  # Keep parallelism at 1

    hash = challenge("Challenge Data", salt, t_cost, m_cost, parallelism)
    return hash
end
```

This is a very hard challenge that requires a large amount of compute to resolve.

In Proof-of-Stake, we require an algorithm we can run hyper-effectively, at any time, awaiting its success. We immediately assign these blocks to an organization where these blocks cannot become loose unless they are burned. In this consensus, we use Schnorr's Forgery to provide validations, as it scales with each new client. Here is an example of this staking:

1. **Given**:
   - A public key Y.
   - A message m.
   - A signature (s, e) where s is the partial signature and e is the challenge.
2. **Calculate**:
   - Compute u = e(G) − s(Y), where G is the base key.
   - Compute e′ using the hash of the public key, the message, and the base of u.
3. **Verify**:
   - If $e = e'$, the signature is valid. Otherwise, it's invalid.

Let's define some variables and equations:

- **Given**:

    - Public key: Y
    - Message: m
    - Signature: (s, e)
- **Intermediate Variables**:
    - u = e(G) – s(Y)
    - e' = H(Y, m, x(u))
- **Equations**:
    - u = e(G) – s(Y)
    - e' = H(Y, m, x(u))
- **Verification**:
    - If $e = e'$ the signature is valid.
    - $O(1)$ complexity validations

Proof:

$$P(Forgery) = \sum_{e=1}^{u} \frac{1}{u} \times Negligable$$

Converting to Julia…

```
function product(value1, value2)
    if length(value1) != length(value2)
        throw(ArgumentError("Values must have the same length"))
    end

    u = 0
    for e in 1:length(value1)
        u += value1[i] * value2[i]
    end

    return u
end
```

Here, we compute products to iterate over all possible values of the random nonce u (from 1 to u), where the term $\frac{1}{u}$ represents the probability of guessing u correctly. The term $Negligible$ accounts for the assumed negligible probability of successfully computing the discrete logarithm.

This captures the probabilistic nature of forging a Schnorr signature by considering all possible values of the random nonce. Keep in mind that the actual analysis involves more complex mathematical expressions, but this notation provides a concise representation of the overall probability.

As suggested in Pass and Shi's work, this Hybrid, Proof-of-Activity, Consensus combines the permissionless setting of mining, with the permissioned setting of Practical Byzantine Fault

Tolerance, to validate all block creations, ensuring fork-free consensus. This allows for multiple solutions to be acceptable, as well as allowing our payouts to be processed to many individuals at once, in real time. Our shim then assigns performance scores to the machines dependent on its hash rate, persists these scores to memory, and then can use these scores for future transactions, mining eligibility and fairness, among other usages.

5. Merkle Space Verifier

As the consensus has multiple acceptable answers, we must regard a similar modality for its verifier (Figure 5).

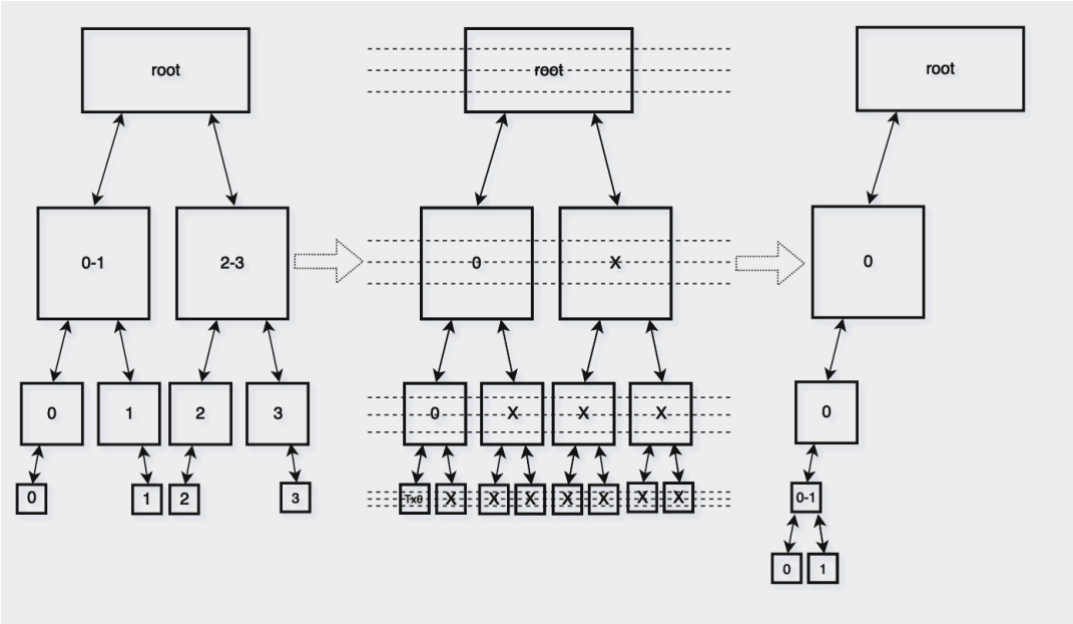

Figure 5: Merkle Tree space reclaim process

To do so, we implement a variety of verifiers from Schnorr's to Proof of UTXO to quickly assess the block in coordination with the chain, run our various provers, and amend the chain.

The hash tree is a tree structure in cryptography where each leaf node is a hash of a data block, and each non-leaf node is a hash of its children. We pair and hash the leaf nodes to create the first level of non-leaf nodes. After validating, the resulting `Root` is the topmost hash, often referred to as the Merkle Root. It uniquely represents the entire set of data blocks.

Merkle trees are widely used in blockchain technology to efficiently verify the integrity of data. If any leaf node is altered, it will change the hash of its parent, affecting the hash of the root. This property makes it easy to detect any tampering in the data.

The variety of verifiers allow us to detect these change attempts and remediate that attack. This also reduces the likelihood of forks substantially, being 100% fork-free with the help of the host.

6. Tamper-Proof Distributed Ledger [21][22]

Blockchains using classical computing, have been held in the highest regard for data security for a long standing. Classical hashing methods mainly include some form of RSA encryption, where data is hashed and digested to 256 logically outputted bits. To complete the computation, a quantum computer would need at least as many qubits as bits outputted. Each qubit is then composed of some large quantity of physical qubits, whereas, in the current state, quantum computers are composed of a small quantity of physical qubits.

Qubits to base d are known as qudits. Qudits are a form of qubit with d greater than 2 total states. This grants each qudit an infinitesimal number of phases and shifts applicable to each photon. This not only means that there are multiple acceptable answers, but also that 0 and 1 are not the only applicable bits in superposition. Qudits, can assume more than 10 states in superposition which can be entangled, modulated, and then read for verification using detectors looking for coincidences.

Coincidences are matching modulations or pulses of a frequency, which are used establish secure channels, for transmitting data. This process utilizes polarization currents to control frequency [18]. These frequencies use qudits to encode information, where each photon acts as a qudit, and thus may undergo some basic logic gate operation using optics and modulators to be detected using single-photon counters.

Device Independent QKD was introduced as a new standard to improve the transmission distances and key generation speeds. In this disclosure, we consider Twin-Field Quantum Key Distribution, where the square root of channel transmittance is proportional to the key rate, a form of Device Independent QKD. This way we utilize TF-QKD to transmit a pair of optical fields generated from two remote parties to meet in the untrusted center (e.g., Charlie) to implement the single-photon detection.

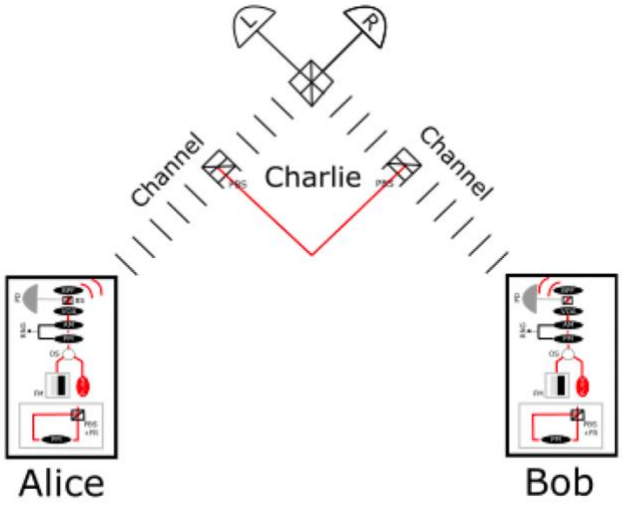

Figure 6: Example DIQKD Transceiver

At this untrusted center, the transceiver-device (Figure 6) does some calculation at Charlie, to validate the connection. TF-QKD inherits unconditional security over large distance, entanglement purification, information theory analytics, and entropic uncertainty relation features

from DI-QKD to increase the key rate substantially to break through linear key rate bounds. We can also exploit analytics like the Gottesman-Lo-Lütkenhaus-Preskill (GLLP) tagging methodology [3] by combining the decoy-state method [8], with our key rate, BB84 encoding [5], six-state encoding [6], and reference-frame-independent (RFI) [9] schemes to distill our secret key on a randomized *Z* phase basis.

Given the quantum bit error rate (QBER):

$$e_{ZZ} = 1 - \frac{\langle ZAZB \rangle}{2}$$

We collect a data set complementary to Z, thus Alice and Bob form a medium C. Using our analyses, we can calculate C:

$$C = (1 - 2e_{XX})2 + (1 - 2e_{XY})^2 + (1 - 2e_{YX})2 + (1 - 2e_{YY})^2$$

This procedure prepares an entanglement state on the Z basis for Alice, and then Alice (Bob) can compute $\alpha(\chi)$. After obtaining C and e_{ZZ}, we can estimate an Eavesdropper's information I_E. Thus, we can define a secret key rate:

$$R = 1 - h(eZZ) - I_E$$

We then define the number of vacuum events in Z_A:

$$S_{zz,0} \geq max\left[\tau_0 \frac{(\nu n^-_{zz,\omega} - \omega n^+_{zz,\nu})}{\nu - \omega}, 0\right]$$

We also define τ_n; the probability that Alice sends an *n*-photon state:

$$\tau_n = \sum_{k \in \kappa} e^{-k} k^n p_k / n!$$

We define the number of single-photon events in Z_A:

$$S_{zz,1} \geq \max\left\{\frac{\tau_1 \mu}{\mu(\nu - \omega) - \nu^2 + \omega^2}\left[n^-_{zz,\nu} - n^+_{zz,\omega} - \frac{\nu^2 - \omega^2}{\mu^2}(n^+_{zz,\mu} - S_{zz,0}/\tau_0\right], 0\right\}$$

QBER e_{ZZ} for single-photon events in Z_A:

$$e_{ZZ} \leq min\left[\tau_1 \frac{m^+_{zz,\nu} - m^-_{zz,\omega}}{(\nu - \omega)S_{zz,1}}, \frac{1}{2}\right]$$

Vacuum events as $S_{k\zeta}, 0$, and single-photon events as $S_{k\zeta}, 1$.

We calculate by optimizing the secret key rate to $R = \vartheta/N$ over our open data set $\{Pr^A_Z, p_u, p_v, \mu, \nu\}$. We set $\omega = 0$ to assume the vacuum state; as our message(s), we let $m^+_{zz,\nu} - m^-_{zz,\omega} = 0$, to assume its neutrality. Since X and Y are symmetric, we treat them similarly, $Pr^A_X = Pr^B_X = Pr^A_Y = Pr^B_Y$, and expect $Pr^A_Y = 0$.

This low-cost Unconditional Security [23] encourages completely secure interactions medially amongst clients, at any scale. The blockchain then uses these secure channels for bit level validations, which then grant authorization for writing to the ledger.

The blockchain consists of a multitude of hashed blocks of information, chained together using a linked list. These linked lists consist of a nonce value to keep as an index. These indexes can be used for many things from block placement to a cache for specific blocks to deciding how and who to reward for the creation of certain blocks. As we know, blockchain is immutable,

therefore any changes to the underlying structure of the chain will cause fatal errors across the chain.

There is also a timestamp, and transaction id within the underlying structure. These values are then hashed together, along with the previous hash, to form the linked list i.e., the previous hash links the arrayed blocks together. These blocks can be encrypted using a variety of cryptographic hashing algorithms and are considered immutable by relation. It is very common to scale the hash function proportional to the wherever your nonce is referencing in the chain. Blockchain is well known as a great medium to scale its chain with O(n) complexity. In using qudits for communication, this allows us to retain similar performance to that of classical computing, with more secure, and harder, connections.

7. Muvor Virtual Machine (MVM)

To utilize these techniques most efficiently, we decided to create and implement our own Request-Response methodology. Our goal, in doing so, is to promote rails for effective Block-Node lifecycles. Accordingly, we created and added two new Request Methods, reforming classical *CRUD* into a Quantum-Block *CREB* (Figure 7) for **CREATE, READ, EMIT**, and **BURN**.

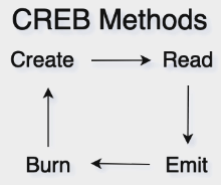

Figure7: Permissible MVM Request Methods

From CRUD, we inherit Create and Read methods, which are immutable, and therefore can replicated here for simplicity. For the Emit method (Figure 8), we denote the **Emitting** of a block as ***an expected expression that updates block function(s) rather than block structure***. **Burning** a block can therefore be deduced as ***the reverting of a block to base structure, allowing for reuse without mutability***.

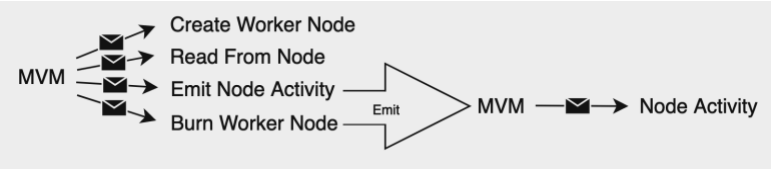

Figure 8: Node Lifecycle

Somethings to note here include that while when a node is burned, it does retain partials of its previous history "where it's been" and therefore does not necessarily mean it will reincarnate even remotely like its prior emitter phase. For instance, the shim may reincarnate a dormant block as a loose block, while creating new clients from an older loose block.

After burning to a loose block, these blocks retain integral structure including nonce, creation time, reward amount, transaction history, hash, previous hash, and its overall shim scoring components. Upon becoming available, a block can be revived in several ways, including Loose Block, Decentralized User, Decentralized Organization, and ERP Organization.

A loose block is the simplest, and smallest version of a Muvor block. These blocks cannot exceed 1MB in size and are only available within the Muvor Network. These are the blocks that will go through lifecycle transformations, hosting clients' offerings, portfolios, wallets, and organization data. When a miner succeeds via Proof-of-Work consensus, that mined block immediately becomes a loose block, and while these blocks cannot be any further burned, the shim can also find these, rewarding all clients, in its shared success.

A Decentralized User is a pseudonymous client, who is the owner of their node. These users have full access to the Muvor Community Platform, and those clients in its network. These blocks receive real-time scores from the shim as it tracks all on-chain activity for expected behaviors. Generally, the better the behavior the better score received from the shim. When a loose block emits a Decentralized User, or one is mined via Proof-of-Stake, this <10MB fully extensible client at its highest permissibility operates as a native cloud user. With every ability from firewall, CI/CD, DNS, etc., all the way to creator functions like Streaming, Licensing, Content Management, etc., these clients operate as hard tenants, completely isolated from each other, while gaining the ability to communicate, roam, and function as a decentralized entrepreneur.

A Decentralized Organization is only to be emitted from a Decentralized User, after this, said client can add more clients, invite peers, and other nodes. Now this Organization can emit further operations to pool resources and conduct various consensus doctrines that are 100% blockchain verifiable. These blocks are <15MB, and allow everything that of a Decentralized User, and that of a Community Organization, as previously discussed.

ERP Organizations are organizations built for the Muvor ERP. These blocks are <25MB in size and boast a rich environment for thriving using blockchain technology. One notable service is the Product Lifecycle. When an ERP Organization is ready to launch a product from the PLM service, they can launch to the entire Muvor Community seamlessly, as well as various sharing means, including affiliate programs, reliant on MVM for all the heavy lifting.

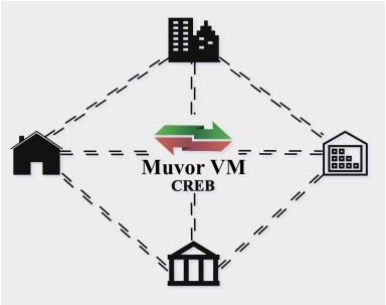

Figure 9: MVM at scale

This Master-Emitter environment (Figure 9) allows MVM to both have complete control over the propagation of the blockchain, while also therein becoming part of the blockchain. This grants us decentralized requests to Quantum-Block *CREB*, without the need for an extension or daemon. This also lowers gas fees substantially, and increases bandwidth, across the chain.

8. Muvor Protocol (Secure) - mvrp(s):/ /Wakuu-Enterprises

All Muvor Xchange blocks are capable of securely running Muvor Protocol, which describes the set of instructions that allow Muvor Networks to reach the internet. Without this, MVM *CREB* would not be feasible, and would likely require the aide of an extension or daemon. We utilize these instructions to dispatch any recognized activity, validate its host, and respond

accordingly. We separate these operations into a User and Kernel Modality such that only permitted users may access the shim. To do so, a user must first verify themselves cryptographically, a task that is proven impossible by classical brute force [24]. Then pass through our shim validation, which will expose the host system from Mac Address to GFlops allocated to platform and more, scoring the client, and then either removing them or permitting their kernel usage.

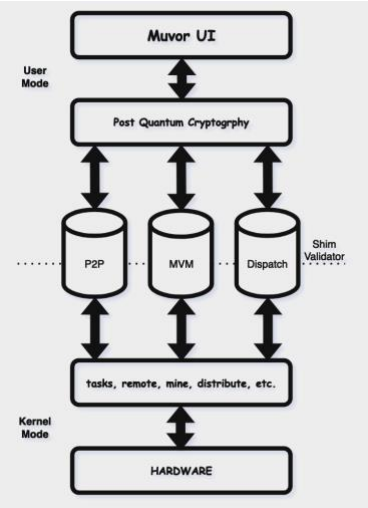

Figure 10: The complete Muvor Protocol

Muvor Protocol (Figure 10) can also utilize standard TLS 1.3 for secure connections. **On-chain network requests may also be made via the mvrn(s):// prefix.** This is a low-latency, high-availability means to beat common traffic, and take advantage of being in the Muvor Ecosystem.

Our secure protocol allows our client seamless internet-intranet communications, where on-chain provides various advantages, and off-chain does not require the use of extension or daemon. Secure Muvor Protocol also leverages MVM for bit validations, which keeps everything secure.

9. EGAHN ($EGN) - Error Generative Adversarial Hypergraph Network

EGAHN is the Error Generative Bases for Muvor's Adversarial Hypergraph Neural Network. This is a series of convolutions to predict an adversarial output. We use the lattices from block creations, and the shim, to map vertices to edges, and then graph them into Hypergraphs. For this, we integrated a proprietary Deep Learning Algorithm that compares the edges of each node, in the consortium, and predicts the requested outcome.

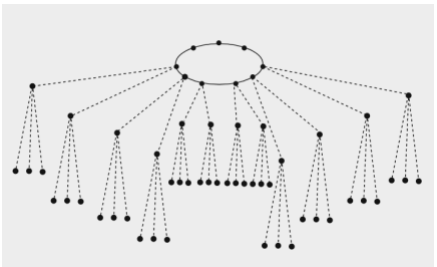

Figure 11: Node Vectors form "perfect lattices" into Homoiconic Endomorphism Ring

We call these similarities from an endomorphism ring at the top level of the network, calculated from "perfect lattices" (Figure 11). The endomorphism ring contains information about each node, which aides in the tracing and discovery of edges. We call it $EGN as it is having been generated from a set of eigenvectors contained within our map, which superimpose an endomorphism ring [25]. To find this, we define a finite field $\mathbb{F}_q$ for $q = p^t$ and p prime. Knowing the decryption undoes encryption, we can denote a Homomorphic Cipher (E, D) over our field $\mathbb{F}_q$ representing data set $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. The cipher is said to be homomorphic if for each key k $\in \mathcal{K}$ and plaintexts $m_1, m_2 \in \mathcal{M}$, it is: $E(k, m_1) *_C E(k, m_2) = E(k, m_1 *_M m_2)$ i.e., decryptable. This allows us to securely generate our lattices over our field from hard problems and endomorphisms such that in the ring of integers $\mathbb{Z}[i] = \{a + bi; a, b \in \mathbb{Z}$, we then impose a lattice structure $\rho_2 : \mathbb{Z}[i] \to \mathbb{Z}^2$ superimposed as $a + bi \mapsto (a + b, a - b)$. We can then define a parallelogram from a lattice $(\Lambda, \rho) \in \mathbb{F}_q$ of basis $\mathcal{B} = \{v_1, \dots, v_n\}$:

$$\mathbb{F}_q(B) = \left\{ \sum_{i=1}^{n} \lambda_i v_i : with\ 0 \leq \lambda < 1 \right\}$$

This allows us to construct higher-order interactions from group monomorphisms using our previously discussed hard problem sets. In doing so, we've defined a set of m nodes $i_1, \dots, i_m$ with hyperedges size m. We use this data recursively and denote the number of hyperedges connecting nodes $i_1, \dots, i_m$ as $A_{i_1,\dots,i_m} \geq 0$.

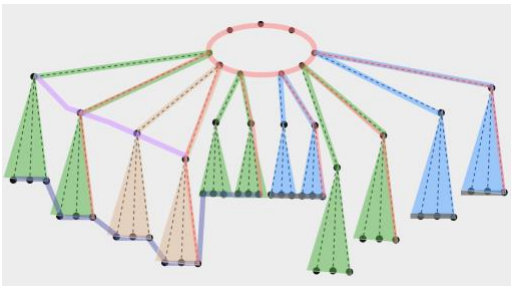

Figure 12: Example Hypergraph permutations

We can therefore deduce a hypergraph H from some network data G, and use our generative model to compute a probability:

$$P(H|G) = \frac{P(G|H)P(H)}{P(G)}$$

Here, we create a projection component $P(G|H)$. $P(H)$ denotes our generative data, while $P(G) = \sum_H P(G|H)P(H)$ functions as a constant evidence variable i.e., data from the network. We can then further express:

$$P(G|H) = \begin{cases} 1 & if\ G = \mathcal{G}(H), \\ 0 & otherwise. \end{cases}$$

Here, we use $\mathcal{G}(H)$ to handle the projection of H such that, by use of $G = \mathcal{G}(H)$, H inherently projects to G. With $A_{i_1,\dots,i_m}$ as our invariant, we can define various hyperedges:

$$E_m = \sum_{i_1,\ldots,i_m \in C_m^N} A_{i_1,\ldots,i_m}$$

This involves the eigenvalues and eigenvectors of a Laplacian matrix. The Laplace Eigenmap edge detection probability using Generalized Eigenvalue Problems (GEP) thus can be defined:

$$P_{edge} = \sum_{i=1}^{n} \lambda_i \cdot \| v_i \|^2$$

Here, *n* represents the number of eigenvalues (i.e., dimensionality of the Laplacian matrix), and $\| v_i \|^2$ denotes the squared Euclidean norm of the *i*-th eigenvector. Most importantly, we can therefore rewrite our edge probability as:

$$P\left(X_{i_1,\ldots,i_m} = 1 \middle| G\right) = \frac{1}{n} \sum_{\ell=1}^{n} X_{i_1,\ldots,i_m}(H_\ell)$$

Where $H_1, \ldots, H_n$ are *n* hypergraphs defined from $P(H|G)$, and where $X_{i_1,\ldots,i_m}(H) = 1_{A_{i_1,\ldots,i_m} \geq 1}$ is equal to 1 denoting presence of a hyperedge connecting nodes $i_1, \ldots, i_m$ in hypergraph *H*.

We can now compute binary cross-entropic loss:

$$S(p') = -p' log_2 p' - (1 - p') log_2 (1 - p')$$

This (re)constructs our hypergraph[26], in preparation to be trained, and aggregated, from a ternary store, to produce high-quality predictions.

More specifically we've created an unweighted and undirected graph $\mathcal{G} = (V, E)(n = |V|)$, where *V* represents a set of vertices, and *E* denotes found edges. From here, our network data will undergo a series of convolutions, and we'll invoke a version of DnnGAN's minimax game [27]:

$$\min_{\phi} \max_{\theta} O(G, D) = \sum_{c=1}^{n} \left(E_{v \sim p_t(\cdot|v_c)} [\log D_\theta (v, v_c)] + E_{v \sim G_\phi(\cdot|v_c)} [\log (1 - D_\theta (v, v_c))] \right).$$

Which will produce our prediction.

Figure 12 highlights the various ways clients may interpret Muvor as multiple series of Hypergraphs from a consortium representation. We named ours EGAHN, denoted $EGN. It calculates an average price based upon the shim scoring mechanism across the available nodes. We can simply represent this as:

$$Market\ Capitalization = Price\ per\ Unit \times Total\ Circulating\ Supply$$

This grants all clients the opportunity to leverage these techniques to not only create, produce, and publish high-quality data sets, but also have a means of funding your cause. For instance, in a medical setting, we can leverage the internal structure of the blockchain, and the prediction loop of EGAHN to create a publicly accessible, and admissible database, where clients can collaboratively train models, and even get funded while doing.

EGAHN stands for Error Generative Adversarial Hypergraph Network. When discussed in correlation to its derivation, a popularity-based currency, we denote this paradigm as $EGN. This neural network derives a consortium of hypergraphs from nodes, to create prediction loops that will optimize our behavior(s). These derivations come from errors and form perfect lattices into an endomorphism ring which helps us detect edges more efficiently for faster predictions.

10. Conclusion

This proposed system standardizes a means of public access, while also reducing friction on key axes of control, trust, and value. We start by dynamically automating the market to ensure our currency remains stable by similar standards to that of standard exchange banks. We then create a secure consortium environment, where we create client nodes that are cryptographically signed to the blockchain. To do this, we utilize Proof-of-Activity algorithms, defined as Proof-of-Work with rotating committees, and employ versions of Proof-of-Work and Proof-of-Stake to sign each block with low-latency, and track its activity throughout its lifecycle. For the signing, we use Post-Quantum Cryptography paradigms we also use in verifying and saving Merkle space. Doing this, means we've transmitted some data to-from our ledger, that requires tamper-proofing. In thus, we've created a set of acceptable answers, suspended in superposition, that reduce space upon computation or observation. We utilize the Muvor Virtual Machine to do this heavy lifting for us, which searches for *CREB* requests, to which it can respond. MVM talks to the Secure Muvor Protocol, which is an integrated set of instructions, that allow internet-intranet access to the Muvor Network. We can use their higher-order interactions to form morphisms we can use to create perfect lattices and create hypergraphs. Clients can then leverage these hypergraphs, with public data, to train the IHGANN, produce similarities, and even derive their own Post-Quantum Cryptocurrency.

Library

[1] Boaron, Alberto, et al. “Detector-Device-Independent Quantum Key Distribution: Security Analysis and Fast Implementation.” *Journal of Applied Physics*, American Institute of Physics (AIP), 9 Aug. 2016, www.osti.gov/pages/servlets/purl/1295142.

[2] Cao, L., et al. “Chip-Based Measurement-Device-Independent Quantum Key Distribution Using Integrated Silicon Photonic Systems.” *Physical Review Applied*, American Physical Society, 17 July 2020, journals.aps.org/prapplied/abstract/10.1103/PhysRevApplied.14.011001.

[3] Gottesman, Daniel, et al. *Security of Quantum Key Distribution with Imperfect Devices*. 3 Sept. 2004, arxiv.org/pdf/quant-ph/0212066.pdf.

[4] Lo, Hoi-Kwong, et al. *Measurement Device Independent Quantum Key Distribution*. Arxiv, 30 May 2012, arxiv.org/pdf/1109.1473.pdf.

[5] Ma, Xiongfeng. *Unconditional Security at a Low Cost*. 31 July 2006, arxiv.org/pdf/quant-ph/0608001.pdf.

[6] Pereira, Margarida, et al. “Quantum Key Distribution with Flawed and Leaky Sources.” *Nature News*, Nature Publishing Group, 26 July 2019, www.nature.com/articles/s41534-019-0180-9.

[7] “Quantum Resistant Ledger (QRL).” *QRL_Whitepaper*, GitHub, Oct. 2016, raw.githubusercontent.com/theQRL/Whitepaper/master/QRL_whitepaper.pdf.

[8] Wang, Xiang-Bin. *A Review on the Decoy-State Method for Practical Quantum Keydistribution*. 19 Sept. 2005, arxiv.org/pdf/quant-ph/0509084.pdf.

[9] Yin, Hua-Lei, and Yao Fu. “Measurement-Device-Independent Twin-Field Quantum Key Distribution.” *Nature News*, Nature Publishing Group, 28 Feb. 2019, www.nature.com/articles/s41598-019-39454-1.

[10] Nakamoto, Satoshi. “Bitcoin: A Peer-to-Peer Electronic Cash System.” *Bitcoin.pdf*, Satoshin@Gmx.com, 31 Oct. 2008, bitcoin.org/bitcoin.pdf.

[11] info@theqrl.org. “Quantum Resistant Ledger (QRL).” *QRL Whitepaper*, Theqrl.org, Oct.

2016, raw.githubusercontent.com/theQRL/Whitepaper/master/QRL_whitepaper.pdf.

[12] Pass, Rafael, and Elaine Shi. “Hybrid Consensus: Efficient Consensusin the Permissionless Model.” *917.Pdf*, CornellTech, Cornell, Initiative for CryptoCurrency and Contracts (IC3), eprint.iacr.org/2016/917.pdf.

[13] Liu, Zhiqiang, et al. “Fork-Free Hybrid Consensus WithFlexible Proof-of-Activity.” *liu2017forkfree*, Shanghai Jiao Tong University, 2017, allquantor.at/blockchainbib/pdf/liu2017fork-free.pdf.

[14] Samuel Jaques and André Schrottenloher. Low-gatequantum golden collision finding. InSelected Areasin Cryptography - SAC 2020 (to appear), 2020.http://eprint.iacr.org/2020/424.

[15] Singhal, Akanksha & Chatterjee, Arko. (2018). Grover's Algorithm. 10.13140/RG.2.2.30860.95366.

[16] Kwon, Jae. “Tendermint: Consensus without Mining.” *Tendermint.pdf*, yk239@Cornell.edu/Tendermint, 2014, tendermint.com/static/docs/tendermint.pdf.

[17] Bentov, Iddo, et al. “Proof of Activity: Extending Bitcoin’s Proof of Work via Proof of Stake.” *452.Pdf*, Http://Www.scipr-Lab.org/, eprint.iacr.org/2014/452.pdf.

[18] Carlson, Erika K. “Emitting Radio Waves with Polarization Currents.” *Physics*, American Physical Society, 15 Dec. 2020, physics.aps.org/articles/v13/s160.

[19] King, Mervyn A. *The End of Alchemy: Money, Banking, and the Future of the Global Economy*. W.W. Norton & Company, 2017.

[20] “Building Enterprise-Grade BlockchainDatabases with MongoDB.” *mongodb_blockchain.Pdf*, MongoDB, Inc, Nov. 2017, webassets.mongodb.com/_com_assets/collateral/mongodb_blockchain.pdf.

[21] Guy Harrison ·, et al. “Sealing MongoDB Data on the Blockchain - DZone Security.” *Dzone.com*, 2 Jan. 2018, dzone.com/articles/sealing-mongodb-data-on-the-blockchain.

[22] Anderson, Ross. *Security Engineering: a Guide to Building Dependable Distributed Systems*. John Wiley & Sons, Inc., 2020.

[23] Liu, Hongwei, et al. “Reference-Frame-Independent Quantum Key Distribution Using ... - Arxiv.Org.” *Reference-Frame-Independent Quantum Key Distribution Using Fewer States*, School of Science and State Key Laboratory of Information Photonics and Optical

Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China, 9 Nov. 2018, arxiv.org/pdf/1811.03244.

[24] Kumar, Rohan "Seirdy." "Becoming Physically Immune to Brute-Force Attacks." *Seirdy's Home*, Gemini Capsule, 2 Dec. 2021, seirdy.one/posts/2021/01/12/password-strength/.

[25] Chacón, Iván Blanco. "Ring Learning with Errors: A Crossroads between Postquantum Cryptography, Machine Learning and Number Theory." *arXiv.Org*, Science Foundation Ireland 13/IA/1914 and MTM2016-79400-P.., 2 Aug. 2020, arxiv.org/abs/1902.08551.

[26] Young, Jean-Gabriel, et al. "Hypergraph Reconstruction from Network Data." *Nature News*, Nature Publishing Group, 15 June 2021, www.nature.com/articles/s42005-021-00637-w.

[27] Zhao, Ming, and Yinglong Zhang. "Gan-Based Deep Neural Networks for Graph ... - Wiley Online Library." *GAN-Based Deep Neural Networks for Graph Representation Learning*, Headmaster Fund of Minnan Normal University, Grant/Award Number: KJ19009; National Natural Science Foundation of China, Grant/Award Number: 61762036; Natural Science Foundation of Fujian Province of China, Grant/Award Numbers: 2021J011007, 2021J011008, 29 Apr. 2022, onlinelibrary.wiley.com/doi/full/10.1002/eng2.12517.