# LinqProtocol (LNQ) Token & Staking Economy: The Building Blocks for a Truly Decentralized Internet

*LinqAI Team*

***(V0.2, Dated: January 09, 2026)***

LinqProtocol is a decentralized cloud computing marketplace powered by its native token, LNQ. By tightly integrating LNQ into marketplace operations, LinqProtocol balances resource utilization and incentivizes participation thereby addressing common pitfalls of previous decentralized compute platforms. LNQ serves multiple functions, supporting network security, trust, and growth.

***Disclaimer****: This document represents a work in progress. It does not claim to be final and may be subject to change.*

## Abstract

This litepaper presents the tokenomics of LNQ, the native utility token powering LinqProtocol - a semi-verifiable decentralized compute marketplace designed to align incentives among requestors, providers, and governance participants in a trust-minimized environment. Amidst the burgeoning DePIN sector, valued at over $50 billion in market capitalization as of 2024 and projected to reach $3.5 trillion by 2028[1], existing solutions face persistent challenges including verification overheads, token volatility, and incentive misalignments that hinder enterprise adoption and network growth. LNQ addresses these through a multifaceted cryptoeconomic design: probabilistic semi-verifiability via watcher nodes (initially centralized for uptime and spot benchmarking, transitioning to provider opt-in services); antagonistic staking and slashing mechanisms ensuring honest provision (with reputation scores offsetting collateral requirements while bounding exploits); lease-based escrows for requestors (fixed-term rentals with renewal deposits and variable LNQ payout schemes for rate stability); and multi-tier dispute resolution for cases where automated decisions may not suffice.

Key innovations include Adaptive Unlock Emissions (AUE) modeling "as-if-mint" dynamics for the upfront 1 billion supply, bounding inflation to GMV growth; variable escrow payout cliffs to prioritize critical costs (such as electricity) which scale with provider stake at risk and reputation scoring; veLNQ governance for antagonistic dispute resolution (three-tiered: automated, juror, council), parameter adjustments; and GMV-tied deflationary burns (for instance 25% of fees) to foster scarcity. Developer grants, streamed via governance to qualifying providers, are capped by AUE-derived treasury inflation.

We demonstrate resilience across scenarios: define how we measure growth, supply-demand imbalances, and explore dispute equilibria. Under current assumptions we project up to 85% organic cost savings[2] from bidding and sufficient uptime under attacks. This framework positions LNQ as a sustainable backbone for decentralized compute, overcoming non-verifiability through economic security thresholds exceeding attack costs, while

enabling seamless Arbitrum integrations for scalability. Future extensions, including zk hybrids, and persistent storage on provider nodes underscore LNQ's potential to capture enterprise workloads in a maturing DePIN ecosystem.

# Section 1: Building Blocks of a Decentralized Compute Marketplace and Evaluation of Existing Solutions

Decentralized compute marketplaces represent a paradigm shift from traditional centralized cloud infrastructure, leveraging blockchain-enabled peer-to-peer networks to aggregate and distribute computational resources. These systems aim to democratize access to compute power, reducing costs through underutilized hardware while introducing cryptoeconomic incentives to ensure reliability in trust-minimized environments. However, they face unique challenges, including verification complexities, incentive misalignments, and scalability hurdles. This section delineates the core components of such marketplaces, evaluates prominent existing solutions through quantitative metrics and critiques, and positions LinqProtocol's LNQ token as a targeted response to identified market gaps.

## 1.1 Core Components of Decentralized Compute Marketplaces

At the foundation of decentralized compute marketplaces lie several interdependent components that facilitate efficient resource allocation and execution in a distributed setting. Requester-provider matching relies on bidding algorithms, where requestors post job specifications (e.g., CPU/GPU requirements, duration) and providers submit competitive bids, optimized via on-chain order books and automated matching engines. Off-chain execution is critical for performance, as blockchain throughput limitations necessitate external computation; in LinqProtocol, this is monitored through watcher nodes that perform probabilistic uptime checks and spot benchmarking to verify provider capabilities (initially centralized or elected for reliability, transitioning to opt-in services run by providers using Kubernetes stacks, rewarded from the L2 treasury funded by protocol fees). On-chain settlement occurs via escrows, where LNQ tokens are locked upon job acceptance and released upon completion.

Hybrid trust mechanisms blend cryptographic and economic primitives: staking provides collateral against dishonesty, while reputation scores (numerical, non-decaying, adjusted based on job outcomes and benchmarks) offset staking requirements for reliable providers. Mathematically, supply-demand equilibrium can be modeled using a production function adapted from storage analogs like Filecoin[3], where marketplace output $Y$ depends on provider count $L$ and gross marketplace value (GMV, $K$):

$$Y = A L^{\alpha} K^{\beta}$$

- $Y$ – Observed compute throughput (eg,. vCPU-hours cleared per epoch)
- $L$ – Active, verified providers (supply).
- $K$ – Gross Marketplace Value, denominated in LNQ (demand).

- $A$ – Efficiency factor uplifted by watcher coverage and network latency; a greater value for A results in squeezing more work out of the same inputs.

- $\alpha$ – Elasticity of supply (how aggressively extra machines boost throughput.)
- $\beta$ – Elasticity of demand (how readily fresh GMV turns into executed jobs.)

With $\alpha, \beta < 1$ and $\alpha + \beta < 1$, returns diminish. This is useful, because it keeps the network from overshooting into wasteful over-capacity.

- If $\alpha > \beta$ hardware is scarce; emissions and grants should target providers, not marketing.
- If $\beta > \alpha$ customers are the choke point; fee cuts or UX subsidies move the needle faster than more GPUs.
- Watchers shift A rightward (double the sampling rate, no extra token spend.)

***Note:*** *LinqProtocol does not inherently suffer from the same loss as Filecoin when excess idle hardware is part of the network due to the difficulty in monetizing idle storage capacity. With LinqProtocol In the case of* $\alpha > \beta$ *the possibility exists of allowing opt-in services - that do not explicitly require network demand - to run on provider machines such as Monero, Bitcoin, or other miners to generate value. Further discussion of this is deprioritized but noted for future consideration.*

In comparison to centralized systems like Amazon Web Services (AWS), decentralized models offer potential cost variability and savings. AWS EC2 instances charge approximately $0.05 per vCPU-hour for standard Linux on-demand pricing in 2025 [3], providing predictable but rigid scalability. Decentralized variability arises from bidding dynamics, potentially yielding up to 85% [2] savings through market competition, though offset by verification overheads and volatility risks. This cost-benefit analysis underscores the need for robust incentives to bridge the gap between centralized reliability and decentralized efficiency.

## 1.2 Comparative Evaluation of Existing Solutions

To contextualize LNQ's design, we evaluate key decentralized compute and storage projects using 2025 metrics, including token price, total value locked (TVL) or market capitalization as proxies for network value, active providers/miners, yields where applicable, and notable critiques. Data is drawn from real-time market analyses as of July 2025.

In-depth critiques reveal common pitfalls: Filecoin's over-collateralization fosters centralization, as evidenced by persistent miner concentration due to to very large up-front staking costs.[21]; Golem's cold-start problems manifest in low adoption rates, stemming from verification scalability and user friction.

## 1.3 Market Gaps and LNQ Positioning

The evaluation in **Table 1.1** exposes critical gaps in the DePIN landscape: high verification overheads in zk-proof systems (general latency increases of 20-40% in decentralized compute in the most optimistic cases, per 2025 analyses[5]), token volatility impeding enterprise uptake, and incentive misalignments leading to centralization or low adoption rate. These issues stifle the sector's potential, despite projections of exponential growth to $3.5 trillion by 2028[1].

LNQ positions itself to address these through semi-verifiability via watcher nodes (probabilistic

uptime/spot checks enabling efficient trust without zk overhead, showing resilient equilibria under attacks); antagonistic mechanisms (staking/slashing with reputation offsets to bound exploits); and volatility mitigations (variable payout lease escrows for predictable costs). Empirical grounding from Akash's volatility-driven barriers underscores LNQ's variable escrow payout mechanisms to facilitate enterprise workloads - for both enterprise requestors but also enterprise providers where fixed costs may not allow for volatility-related risks over a long period of time. Such innovations forster organic (up to 85%[2]) savings via bidding while aligning multi-role incentives for sustainable growth.

| Project | Token Price (USD) | TVL/Market Cap (est. USD) | Active Providers/ Miners | Yields (est.) | Key Critiques |
|---|---|---|---|---|---|
| Golem (GLM) | ~$0.26 | ~$260M (market cap) | ~5K | N/A | Cold-start issues persist, with years of slow adoption across the network. |
| Filecoin (FIL) | ~$2.62 | ~$1.5-2B (market cap) | ~3K | N/A | Fosters centralization, as evidenced by persistent miner concentration due to the very large up-front staking cost.[21] |
| Akash (AKT) | ~$1.39 | ~$330M (market cap) | N/A | ~15% | Volatility and payment barriers hinder enterprise adoption. |
| Render (RENDER) | ~$3.75 | ~$2B (market cap) | N/A | N/A | Emissions pitfalls lead to inflationary pressure, diluting value despite GPU focus.[6] |

***Table 1.1*** *- Comparative Analysis of Existing Decentralized Compute Marketplaces (from Data Collected in 2025)*

# Section 2: Basic Incentives Driving Semi-Verifiable Compute Provision via Antagonistic Mechanisms

In decentralized compute marketplaces, where full cryptographic verifiability often incurs prohibitive costs, incentive mechanisms must bridge the trust gap between participants. Semi-verifiability emerges as a pragmatic approach, relying on probabilistic checks rather than exhaustive proofs to deter dishonesty while maintaining efficiency. This section formalizes semi-verifiability in the context of LinqProtocol, emphasizing the role of watcher nodes, and delineates antagonistic mechanisms including staking, slashing, and reputation systems that drive honest behavior. Drawing on game-theoretic principles, we demonstrate how these elements achieve Nash equilibria for reliability, with ties to dispute resolution processes that enforce slashing and reputation adjustments in the ProviderRegistry. This design addresses core DePIN challenges, such as the verifier's dilemma in blockchain protocols, where participants may shirk verification to conserve resources.[7]

## 2.1 Defining Semi-Verifiability in Compute

Semi-verifiability in decentralized compute refers to a hybrid trust model where task outcomes are not fully proven cryptographically but are probabilistically validated through sampling and monitoring, balancing computational overhead with economic deterrence. This contrasts with deterministic verifiability, exemplified by zero-knowledge proofs (ZKPs), which provide irrefutable evidence of correctness but introduce significant latency and resource demands. For instance, ZKP-based verifiable machine learning frameworks often exhibit excessive overheads. In other mechanisms such as TEEs (Trusted Execution Environments) overheads are observed to be ~20%.[5] Such tradeoffs are in part due to the complexity of proof generation and verification in decentralized settings. These trade-offs are particularly acute in real-time or resource-constrained DePIN applications, where full ZKP deployment can hinder scalability and adoption.

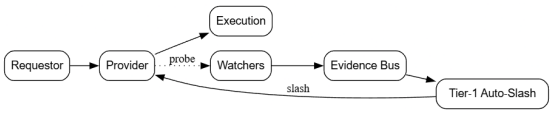


*Diagram 1.1 - Watcher-based semi-verifiability flow*

In LinqProtocol, semi-verifiability is achieved through watcher nodes that consume two complementary data streams from every provider cluster: (i) continuous telemetry of resource-level metrics and logs, and (ii) results from mandatory pre-deployment benchmarks plus opportunistic "random checks" executed at random whenever capacity sits idle. Together, these feeds let watcher nodes maintain a near-real-time performance profile, surfacing issues such as network outages, power loss, or ageing hardware before they impact workloads.

During the network's initial phases, watcher nodes are operated by a small, centrally elected set of nodes financed by the L2 treasury; over time, providers can **opt-in** to host the service themselves, earning additional rewards and progressively decentralising verification. Under current assumptions we expect that by sampling only ≈ 10–20 % of time windows and tasks, the system attains a misconduct-detection probability of ~0.8 without incurring the latency overhead of

deterministic proofs. Signed watcher attestations flow straight into the dispute engine, where they can trigger automated slashing or reputation adjustments in the ProviderRegistry, tightly coupling verification with economic enforcement and deterring rational misbehaviour.

### 2.2 Antagonistic Incentive Mechanisms

Antagonistic incentives in LinqProtocol leverage economic penalties and rewards to align provider behavior with network integrity, creating a self-enforcing system where the cost of deviation exceeds potential gains. Central to this is dynamic staking and slashing: providers must stake collateral proportional to job value, formalized as:

$$S_{\text{req}} = \alpha V_{\text{job}}$$

Where $\alpha$ is a risk factor (e.g., 1.5x for high-value tasks to account for variability) and $V$ is the total job value. To prevent exploits, such as partial completion where rewards outweigh penalties, the mechanism enforces that any partial reward is strictly less than the staked amount, bound by governance parameters to maintain exploit resistance.

Game-theoretically, this yields a Nash equilibrium for honest participation. Consider a provider's strategy space: honest execution yields the full job reward, while deviation (e.g., falsifying results) risks detection and slashing.

The expected loss for deviation is modeled as:

$$\mathbb{E}[\text{loss}] = p_{\text{detect}}\ \sigma\ S$$

- $p_{\text{detect}}$ – probability the watcher network catches a bad result (assumed to be ≈ 0.8).
- $\sigma$ – fraction of the stake burned when you're caught (0.5 by default).
- $S$ – the amount of LNQ the provider had to lock for the job.

To keep cheating irrational, we require this expected loss to outstrip the best payoff a provider could get by falsifying results, $G_{\text{misconduct}}$; in other words:

$$p_{\text{detect}}\ \sigma\ S\ \geq\ G_{\text{misconduct}}.$$

With today's baseline numbers, the cheater expects to forfeit **40% of their stake every time they deviate**. In practice, a profit-seeking provider quickly realises that honest execution is the only strategy that pays.

Sybil resistance is further bolstered by reputation-weighted staking, where a provider's numerical reputation score (adjusted via job outcomes and benchmarks) impacts staking offsets, for example higher scores reduce required collateral through a multiplier of 1 divided by reputation factor, with limits to cap reductions.

Reputation-weighted staking ties a provider's collateral directly to their track record. For any job, the stake required is:

$$C_{\text{req}}\ =\ \frac{S_{\text{base}}}{R}, \qquad \text{clamped to } C_{\min} \leq C_{\text{req}} \leq C_{\max}.$$

Here $S_{\text{base}}$ is the baseline stake and $R$ is the provider's current reputation score. A newcomer with $R \approx 0.5$ posts the full baseline amount, whereas a well-proven node with $R = 1$ locks only a fraction of it. Governance-defined floors $(C_{\min})$ and ceilings $(C_{\max})$ prevent either extreme under-collateralisation or punitive over-collateralisation, keeping sybil attacks costly while avoiding Filecoin-style over-staking spirals. Because reputation rises slowly with successful completions and falls sharply on disputes or uptime

failures, collateral automatically adjusts: good actors see their capital burden lighten, while inconsistent or pseudonymous actors face higher stakes until performance improves or they exit the market.

This integrates with the ProviderRegistry, where disputes trigger score deductions and slashes, preventing pseudonymous attacks by tying influence to verifiable history. Such mechanisms echo reputation-based consensus in PoS systems, enhancing sybil resistance without excessive centralization. Overall, these antagonistic elements ensure semi-verifiable compute remains robust, with watcher ties amplifying detection and enforcement.

# Section 3: Incentive Structures for Network Participants

Decentralized compute marketplaces function as multi-sided platforms, connecting requestors seeking resources with providers offering them, while governance participants underpin security and decision-making. Effective incentive design is crucial to align behaviors across these roles, mitigating issues like free-riding or adverse selection prevalent in blockchain ecosystems. In LinqProtocol, incentives are antagonistic, rewarding cooperation while penalizing deviations through staking, reputation, and governance mechanisms. This section details the structures for each participant group, quantifying yields where possible and modeling multi-sided equilibrium to ensure balanced growth. Drawing on game-theoretic analyses of blockchain incentives, such as collective punishment schemes that deter faults in decentralized settings, LNQ's design fosters a self-sustaining economy where participant utilities are interdependent.

## 3.1 Requestors (Clients)

Requestors, as compute consumers, are incentivized through mechanisms that ensure reliable access while minimizing abuse risks. The primary incentive for Requestors is the reduced cost of compute by means of antagonistic bidding between Providers as well as access to reputation/cost tradeoffs which can be made at the Requestors discretion. Anti-abuse is enforced via lease-based escrows: a lease represents a fixed-term compute rental (e.g., 30 days at 0.84 LNQ per hour, requiring full up-front deposit in LNQ. Renewal occurs on any day before the lease expires via another full-term deposit, tied to Variable Escrow Payout Cliffs in LNQ (see **Section 6**) for variable cost payouts to mitigate losses during periods of volatility. Boundaries are clearly defined in two cases where cancellations incur a small penalty (e.g., 5-10% of remaining escrow) and a portion of the penalty fee is directed to the Provider as a compensation for the opportunity cost.

**Case 1 - Escrow includes base cost:** In the case an escrow includes and immediate payout for base costs cancellation can be calculated as follows:

$$C = h_{\text{used}} \cdot c_{\text{hour}} - C_{\text{base}} + C_{\text{fee}}$$

- $C$ – Total cost charged after cancellation.
- $C_{\text{base}}$ – Fixed base cost applied regardless of usage.
- $h_{\text{used}}$ – Number of hours worked before cancellation.
- $c_{\text{hour}}$ – Cost per hour of work.
- $C_{\text{fee}}$ – Cancellation fee

In escrows which incur a base cost cancellations are only possible in the case that $C > C_{\text{base}}$ which implies that leases which include a base cost are fixed while $C <= C_{\text{base}}$.

**Case 2 - Escrow excludes base cost:** In the case an escrow does not incur a base cost, cancellation can be calculated simply as follows:

$$C = h_{\text{used}} \cdot c_{\text{hour}} + C_{\text{fee}}$$

This manages variable usage efficiently, as underutilized leases still allow pro-rata refunds minus penalties, aligning with economic models of resource allocation in multi-sided platforms where penalties deter wasteful behavior.

Savings for requestors are organic, derived from market competition in the bidding ecosystem, expected to yield up to 85%[2] reductions below centralized benchmarks (e.g., AWS vCPU-hour rates ~$0.05 in 2025)[3] through supply-demand dynamics. Users can select providers based on reliability-cost trade-offs, such as opting for lower-reputation nodes for greater savings, fostering a competitive equilibrium where bid spreads reflect risk-adjusted pricing. This mirrors incentive schemes in federated learning marketplaces, where client-side rewards encourage participation without direct yields.

### 3.2 Providers

Providers, as resource suppliers, are motivated by direct job fees with reputation and slashing ensuring accountability. Reputation is a numerical score (e.g., 0-100 scale) with no decay, increasing on successful job completions and decreasing on failures, benchmark shortcomings, or uptime check lapses irrespective of job state to capture holistic reliability. Staking offsets are straightforward yet bounded: collateral requirements are reduced via a multiplier equal to 1 divided by reputation factor as explained in **Section 2.2** with minimum/maximum limits (e.g., 0.5-2x base) to prevent over-reliance on reputation and bound exploits where partial rewards might exceed penalties.

This design integrates with the ProviderRegistry, where disputes trigger slashing (e.g., 20-90% of stake depending on the dispute tier) and score adjustments, updating eligibility for future jobs.

This aligns with reputation mechanisms in blockchain systems, where score-based weighting deters sybil attacks and promotes long-term honesty. Game-theoretic models of staking in proof-of-stake networks confirm that such offsets achieve equilibrium by making deviation costs (slashing) outweigh gains, with registry updates enforcing transparency.[9]

### 3.3 Governance Participants

Governance participants form the network's security and decision-making backbone with antagonistic elements driving engagement. There are two factions of governance participants in the ecosystem with separate and defined roles. **Protocol governance** voters can vote on protocol adjustments and are penalised for failure to form majority consensus only in the cases where governance proposals are passed. In **Section 3.4** we explore how proposals are created and the exact mechanics of protocol-level governance. Separately, **dispute governance** voters can risk a portion of their stake and vote on outcomes in dispute resolutions between Requestors and Providers. Dispute governance voters are penalized for failure to form a majority consensus by means of slashing minority consensus voters' stakes and redistributing them with the majority of the jury regardless of dispute outcomes. In **Section 8** we discuss the mechanics of dispute resolution in depth and explore the various dispute tiers. Protocol governance operates through stake-locked

voting contracts which mint veLNQ (vote-escrowed LNQ), enabling participation in protocol-level proposals. In the case of disputes jurors are randomly selected from governance participation for votes, with duties limited to availability for calls. This mirrors proof-of-reputation frameworks where minority penalties enforce alignment.[9]

**Watcher nodes**, as a separate service not directly participating in governance but serve an important role in dispute resolution. Watcher nodes are treasury-funded (from protocol fees) for network verification, with long-term opt-in by providers earning additional payment for running watcher nodes.

## Section 3.4: Protocol Governance

Protocol-level decisions such as parameter tuning, contract upgrades, and treasury actions are enacted through a stake-weighted ballot in which every participant locks LNQ to mint vote-escrowed tokens (veLNQ). The mechanism introduces two capital flows that jointly finance grant rewards and protocol revenue: an *irrevocable proposal fee*, and a *refundable proposal bond* that is conditionally redistributed once the vote finalises. Let

$$B > 0 \quad (\text{bond size}),$$
$$f \in (0,1)B \quad (\text{flat fee}),$$
$$\sigma_m, \sigma_p \in (0,1) \quad (\text{slash rates}),$$

where $\sigma_m$ governs minority penalties when a proposal *passes* and $\sigma_p$ governs proposer penalties when a proposal *fails* (default $\sigma_p = 1$). The execution logic is:

1. **Proposal submission**

   The proponent deposits $B$; a fraction $f$ is paid into the grant treasury immediately, rendering proposal spam economically costly even before voting begins.

2. **Ballot resolution**

   After the quorum-constrained window closes, votes are decrypted and tallied.

   ***If the proposal passes*** the bond is returned with a modest premium $\rho B (\rho \ll 1)$, financed by slashing the stake of voters on the minority side: each minority voter loses $\sigma_m s_i$, where $s_i$ is the LNQ weight of their ballot. The remainder of the minority pot is streamed pro-rata to majority voters, creating a direct economic signal to coordinate on outcomes likely to enjoy broad support rather than merely "any change".

   ***If the proposal fails*** the bond is confiscated: $\sigma_p B$ is credited to the grant treasury for future expansion funding, while the small residual $(1-\sigma_p)B$ is recycled to the treasury. Crucially, voters themselves are never slashed in a failed ballot, eliminating the perverse incentive to sabotage.

3. **Payout algebra**

   Denoted by $M$ and $\bar{M}$ the total stake on the majority and minority sides, respectively. For a passing proposal the expected net change for a representative majority voter staking $s_j$ is

   $$\Delta_j^{\text{maj}} = \rho \frac{B}{M} + \sigma_m \frac{s_j \bar{M}}{M} - c,$$

   where c is the opportunity cost of locking tokens during the voting window. Because $\rho \ll \sigma_m$ and $B$ itself is at risk, the payoff is dominated by coordination accuracy rather than by bond farming; conversely a minority voter expects

$$\Delta^{\min} = -\sigma_m s_i - c,$$

establishing a strict incentive to side with the most plausible majority while still allowing principled dissent when the expected social benefit exceeds private loss. For a failing proposal $\Delta_j^{\text{maj}}$ and $\Delta^{\min}$ collapse to $-c$ : turnout is rewarded only through the reputational externalities of active governance, not through zero-sum transfers.

4. **Parameter governance.**
   All coefficients $(B, f, \sigma_m, \sigma_p, \rho)$ reside in **ParamHub** and thus can themselves be tuned by the very mechanism they regulate, enabling reflexive calibration as network conditions evolve. Bounds are hard-coded in the protocol to preclude self-destructive oscillations.

By taxing *proposal quality* rather than *voter dissent*, the scheme preserves pluralism where minority voices pay only when proven wrong. Game-theoretic analysis mirrors Schelling-point courts in showing that, under honest-majority assumptions, the equilibrium strategy for rational token-holders is to submit only proposals whose expected social surplus outweighs the probabilistic bond loss, and to vote truthfully on anticipated majority sentiment. As such, governance becomes a capital-efficient public good rather than a rent-extraction vector, dovetailing with LinqProtocol's broader antagonistic, yet cooperative, incentive architecture. [10]

# Section 4: Token Distribution, Vesting Schedules, and Multi-Role Utility

Token distribution and vesting are foundational to cryptoeconomic credibility, preventing premature dilution, aligning long-term incentives, and mitigating risks such as insider dumps that have plagued early blockchain projects. In DePIN ecosystems, where network growth depends on sustained participation, adaptive vesting tied to milestones (e.g., GMV thresholds) enhances sustainability by correlating supply release with value creation. This section outlines LNQ's distribution and vesting framework, designed for transparency and growth alignment, before consolidating its multi-role utility as a medium of exchange, collateral, and governance instrument. Emissions are governed by Adaptive Unlock Emissions (AUE) to simulate controlled minting, ensuring inflation scales with network health.

## 4.1 Token Distribution and Vesting

LNQ's distribution prioritizes ecosystem sustainability, with allocations structured to bootstrap participation while vesting schedules enforce commitment. Token distributions for the project itself are emitted gradually as a percentage of AUE unlocks and tie rewards to performance. Higher GMV and network growth accelerate emissions, creating a direct incentive for development (e.g., more active leases/GMV = proportionally more tokens, bounding project inflation to network expansion). The treasury is allocated a portion for operational needs, such as developer grants and watcher node rewards, funded within AUE-derived limits to cap spending.

This adaptive approach mitigates risks observed in fixed-vesting models, where premature releases lead to volatility, and draws from milestone-based vesting in DePIN protocols, which correlate supply with utility to enhance token velocity and holder retention. Transparency is enforced via on-chain tracking, with relocks from Unicrypt to the L1 treasury ensuring emissions reflect real activity (see **Section 5.1**).

### 4.2 Multi-Role Utility Overview

LNQ's utility spans multiple roles, creating a cohesive economy where token demand scales with network usage, akin to multi-functional assets in layered blockchain designs that enhance composability and value accrual. **As a medium of exchange,** LNQ facilitates lease payments in the marketplace: requestors deposit LNQ into escrows for fixed-term rentals, with opt-in variable payout cliffs to providers to ensure protection from volatility. This role drives organic demand, as increased job volume directly correlates with token circulation.

**In locking,** LNQ serves as a deterrent for bad actors trying to manipulate the network by allowing locked tokens to be slashed in the case of poor or dishonest behaviour. This punitive mechanism ensures a significant cost for dishonesty in the network which bad actors need to weigh against any benefits before acting dishonestly. **Governance utility** manifests through veLNQ: time-locked stakes in voting pools grant weighted votes for proposals (e.g., parameter adjustments) and juror roles in disputes, with antagonistic rewards (from successful/lost outcomes) and slashes (minority votes) fostering honest participation. This multi-role integration reduces token velocity by encouraging locks, as modeled in utility token frameworks where governance rights amplify holding incentives, ultimately supporting LNQ's deflationary trajectory as usage grows.

## Section 5: Special Network Requirements for LNQ, Including Adaptive Unlock Emissions Rate

Blockchain tokenomics often necessitate specialized mechanisms to balance supply dynamics with network growth, particularly in DePIN ecosystems where upfront minting can lead to inflationary pressures without adaptive controls. For LNQ, these requirements include Adaptive Unlock Emissions (AUE) to simulate controlled minting for the 1 billion upfront supply, ensuring inflation sustains but does not outpace utility. Additionally, deployment on Arbitrum as an optimistic rollup addresses scalability while prioritizing security through fraud proofs. This section formalizes AUE as an "as-if-mint" process and analyzes Arbitrum's Layer-2 (L2) integration, emphasizing Ethereum (ETH)/Arbitrum focus for low-friction user acquisition. Future extensions, such as LinqChain for a native hub, are noted but deprioritized to minimize initial complexity. These features draw from dynamic supply models in blockchain literature, where adaptive mechanisms mitigate dilution risks in utility-driven tokens.

### 5.1 Adaptive Unlock Emissions (AUE)

AUE functions as a closed-loop controller that aligns LNQ's fixed supply with real usage. Unlocks are held in Unicrypt and emitted only when the network’s composite growth signal warrants it, functioning as an "as-if-mint" mechanism to sustain proportional liquidity.

Emissions to the L1 treasury at block-time $t$ are computed via a dynamic feedback equation:

$$e_t \;=\; P\big(g_t - g_{\text{target}}\big) \;+\; I\int_0^t \big(g_\tau - g_{\text{target}}\big)\,d\tau \;+\; D\,\frac{dg_t}{dt}$$

- $e_t$ – Tokens emitted to the L1 treasury at block-time $t$.
- $g_t$ – Composite growth signal (leases + volume + locked wLNQ, individually normalised and weight-averaged).
- $g_{\text{target}}$ – Governance-set reference track for sustainable expansion (e.g., 5% MoM growth).
- $P, I, D$ – are tunable gains controlling real-time response, bias correction, and volatility dampening, respectively (adjustable on-chain via Paramhub to keep variance in check).

The controller borrows directly from classical $PID$ design[11]: the $P$ term reacts to the present gap, $I$ soaks up cumulative drift, and $D$ dampens sudden lurches. In practice this means:

- a **small network** (negative gap) retains 80-90% of tokens re-locked, curbing premature inflation.
- a **surging network** (positive gap) emits more tokens to support scaling, but derivative dampening prevents supply shocks from transient spikes.

**Decoding the growth signal**

$$g_t \;=\; w_L\,\hat{L}_t \;+\; w_V\,\hat{V}_t \;+\; w_S\,\hat{S}_t,$$

$$w_L + w_V + w_S = 1$$

Where:

$$\hat{L}_t = \text{leases}t/\text{leasesnorm},$$
$$\hat{V}_t = \text{volume}t/\text{volumenorm},$$
$$\hat{S}_t = \text{wLNQ}t/\text{wLNQnorm}.$$

Weights $w_\bullet$ encode strategic priorities (e.g. bias towards security early on ($w_S\uparrow$) or towards demand in expansion phases ($w_V\uparrow$).) Normalisers are rolling medians that keep the metric dimensionless and resistant to outliers.

**Governance levers**

- $PID$ **gains** – tuned quarterly; raising $P$ quickens response at the cost of volatility, upping $I$ closes long-term bias, $D$ dampens noise.
- **Signal weights** – re-balanced via DAO votes as the network graduates from boot-strap to scale.
- **Target track** – can be static (e.g., 5 % MoM) or algorithmic (e.g., trailing GMV exponential-moving-average).

Unlike conventional vesting, AUE introduces a cryptoeconomic reflex: LNQ's supply adapts to verifiable usage, not schedule. As long as $g_t$ mirrors genuine activity and the gains are tuned for stability, LNQ's supply remains both elastic and predictable, two traits that legacy DePIN tokens rarely achieve.

## 5.2 Other Requirements

LinqProtocol's deployment on Arbitrum as an optimistic rollup fulfills scalability needs while inheriting Ethereum's security, focusing on ETH/Arbitrum interoperability for low-friction user base expansion. Arbitrum's fraud proofs enable challenge-based validation: any participant can dispute invalid state transitions within a window (e.g., 7 days), with resolution on L1 Ethereum via interactive games that bisect execution traces until fraud is proven or disproven. With current and future proposed additions to Arbitrum it could reduce gas costs by 10-100x compared to L1[12],

facilitating high-throughput leases without compromising finality.

The ETH/Arbitrum focus prioritizes seamless onboarding, leveraging Ethereum's liquidity for LNQ bridging while minimizing cross-chain risks. Future potential includes LinqChain as a native hub for expanded services, but this is deprioritized to avoid friction in early adoption phases.

## Section 6: Variable Escrow Payout Cliffs for Volatility Mitigation

Token volatility represents a structural risk for decentralized compute providers, particularly where operating expenses (e.g. electricity, bandwidth, cooling) are incurred continuously while revenues are denominated in a volatile native asset. Unlike requestors, providers cannot always defer costs or tolerate prolonged price drawdowns without impairing service reliability. To address this asymmetry without resorting to stablecoins or external hedging instruments, LinqProtocol introduces Variable Escrow Payout Cliffs (VEPC).

VEPC is an escrow-native mechanism that selectively unlocks a bounded portion of job value immediately at lease start, allowing providers to front-load critical operating costs while preserving full cryptoeconomic alignment in LNQ. The amount eligible for immediate payout scales strictly with capital at risk (staked collateral) and historical trustworthiness (reputation score), ensuring that volatility protection is earned rather than granted.

### 6.1 Core Mechanism

For each lease, the requestor deposits the full job value in LNQ into escrow, as described elsewhere in this document. At job initiation, the provider may optionally declare a base cost component, representing near-term operating expenses they wish to cover immediately. This declaration does not guarantee payout; instead, it is evaluated against an algorithmic eligibility cap enforced by the protocol.

The maximum immediate payout a provider may request is bounded as a function of their reputation score and staked collateral:

$$\text{MaxImmediateCost} = f(R, S)$$

Where:

- $R$ is the provider's reputation score as tracked in the ProviderRegistry,
- $S$ is the amount of LNQ staked as collateral for the job,
- $f(\cdot)$ is a monotonic function governed by ParamHub parameters.

In abstract form, this may be expressed as:

$$\text{MaxImmediateCost} = \alpha \cdot g(R) + \beta \cdot h(S)$$

with:

- $g(R)$ increasing in reputation (higher trust → lower haircut),
- $h(S)$ increasing in stake (more capital at risk → higher eligibility),
- $\alpha, \beta$ governance-tunable weights.

Providers may declare any base cost up to this cap. Requests exceeding the cap are automatically rejected on-chain. This ensures that immediate payouts are hard-capped algorithmically, even though the scale and curvature of the function remain adjustable via governance.

## 6.2 Reputation-Weighted Haircuts and Capital at Risk

Reputation serves as a **haircut mitigator**, not a bypass. Higher reputation does not eliminate collateral requirements; it reduces the marginal stake required to qualify for a given immediate payout. This mirrors reputation-weighted staking offsets described in Section 2 but is applied specifically to escrow liquidity timing rather than eligibility to serve jobs.

Key properties:

- Low-reputation or new providers qualify only for minimal immediate payouts unless they over-collateralize.
- High-reputation providers can unlock a larger portion of escrow with the same stake.
- Governance-defined floors and ceilings prevent both under-collateralization and runaway privilege accumulation.

Crucially, **all immediate payouts remain subordinate to slashing**. The protocol enforces the invariant that the combined value of stake plus remaining escrow must always dominate the provider's extracted value under expected volatility conditions.

### 6.3 Volatility Risk Target and Drawdown Handling

VEPC does not attempt to eliminate volatility; it bounds its impact. The system is parameterized around a governance-defined downside risk target (e.g. protection against an assumed X% price drawdown during the lease period). This target is not a hard guarantee and may be adjusted via ParamHub as market conditions evolve.

Behavior under adverse price movements is explicitly defined:

- **Case 1 – Provider continues execution:** No protocol intervention occurs. The provider completes the job and receives the remaining escrowed LNQ at settlement, regardless of interim price movements.
- **Case 2 – Provider halts execution due to insufficient incentive:** The lease is terminated. The requestor is refunded the unused portion of escrow plus any applicable cancellation penalty, funded first from the provider's stake and then, if necessary, from the remaining escrow. Reputation penalties and slashing apply according to existing dispute logic.
- **Case 3 – Requestor cancels the lease:** Cancellation mechanics follow the rules defined in Section 3, independent of whether an immediate payout was taken.

By construction, the provider cannot externalize volatility losses onto the requestor beyond work already performed, and the requestor cannot claw back legitimately paid base costs absent provider fault.

### 6.4 Slashing and Payout Ordering

In the event of dishonesty, non-performance, or dispute-triggered penalties, losses are absorbed in the following order:

1. **Staked collateral** (primary loss absorber),
2. **Remaining escrow balance** (secondary, only if stake is insufficient),

This ordering preserves the core security principle articulated elsewhere in the paper: partial rewards

plus immediate payouts must never exceed capital at risk.

### 6.5 Governance Surface (ParamHub Controls)

Variable Escrow Payout Cliffs introduce several explicit governance levers, including but not limited to:

- Maximum immediate payout as a fraction of job value,
- Functional form and weights of the reputation and stake scaling function,
- Reputation multiplier bounds and haircut curves,
- Assumed downside volatility target used in eligibility calculations,
- Minimum stake floors for any immediate payout eligibility.

All parameters are initialized conservatively and are adjustable only through protocol governance, allowing VEPC to adapt as empirical volatility data and network maturity evolve.

### 6.6 Rationale and Enterprise Implications

VEPC shifts volatility management from **price anchoring** to **capital-at-risk alignment**. Instead of stabilizing prices via external assets, the protocol stabilizes *behavioral incentives*: providers with higher stake at risk and stronger track records receive greater protection against short-term price shocks.

This preserves LNQ as the sole unit of account, avoids stablecoin dependencies, reduces sell-pressure feedback loops, and creates a credible path for enterprise-grade providers whose cost structures cannot tolerate prolonged revenue uncertainty.

## Section 7: Provider Collateral Staking, Inflation Bounds, and Deflationary Economics

Staking in LinqProtocol is deliberately narrow in scope: it exists solely to provide **economic security through collateral**, not to generate passive yield. Unlike many blockchain systems that conflate staking with investment-like returns, LNQ staking is restricted to providers who actively supply compute and must post capital at risk to participate. This design eliminates passive rent-seeking, reduces systemic complexity, and ensures that all locked capital directly underwrites real economic activity.

In DePIN contexts, unchecked token emissions or yield-driven staking often lead to reflexive inflation and weak security guarantees. LinqProtocol instead separates concerns cleanly: **staking enforces honesty**, **emissions are governed independently via Adaptive Unlock Emissions (AUE)**, and **deflation is driven mechanically by usage**. This section formalizes provider-only collateral staking and its interaction with inflationary and deflationary controls.

### 7.1 Provider Collateral Staking Mechanics

All LNQ staking in the protocol is performed by **providers** and serves as collateral against non-performance, dishonesty, or dispute outcomes. There are no passive staking pools, no yield-bearing instruments, and no pooled insurance constructs. Capital is locked only when it is explicitly placed at risk.

For each job or lease, providers must stake LNQ collateral proportional to the value and risk profile of the workload:

- Baseline collateral requirements scale with job value (e.g., ≥ 1.5× expected payout).
- Reputation-weighted offsets reduce required collateral for proven providers, bounded by governance-defined multipliers (e.g., 0.5×–2×) to prevent under-collateralization.
- The invariant **partial rewards + immediate payouts < stake at risk** is strictly enforced, ensuring that deviation is always economically irrational.

Collateral remains locked for the duration of the lease and is subject to slashing in the event of:

- failed execution,
- watcher-detected underperformance,
- dispute resolution outcomes,
- or protocol-defined non-compliance.

This mirrors antagonistic security models in proof-of-stake systems, but with a crucial distinction: **only actors who can cause harm are required to stake**, and only in proportion to the harm they can cause.

### 7.2 Inflation Bounds and Deflationary Mechanisms

Inflation is governed dynamically to sustain proportional expansion without inducing dilution. Token emissions are modulated by the Adaptive Unlock Emissions (AUE) controller, which adjusts release rates in response to deviations between the observed composite growth signal and a governance-defined target trajectory (see **Section 5.1**).

Using a minimal computer simulation we illustrate what an optimal inflation/deflation curve looks like in both optimistic and pessimistic scenarios over 5 years as illustrated in **figure 1.1.**.

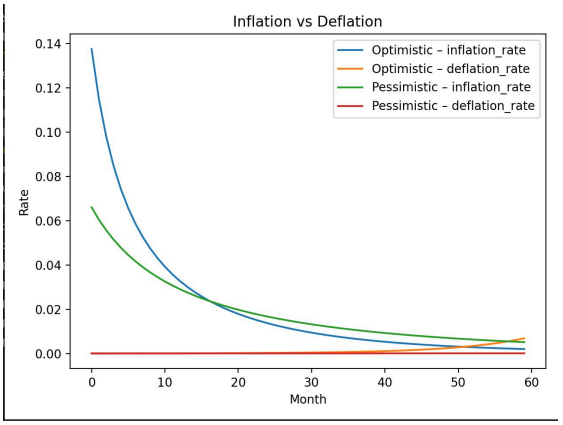


*Figure 1.1 - Optimal Inflation vs Deflation theoretical projection*

Deflationary counterbalance is achieved through proportional token burns, parameterized as:

$$\text{Burns}_t = \gamma \cdot \text{GMV}_t$$

where $\gamma = 0.25$ is a tunable protocol parameter and $\text{GMV}_t$ denotes the total value of completed jobs and leases in LNQ equivalents at time $t$. This linear relationship ensures that deflation scales dynamically with real economic throughput, introducing scarcity as a function of productive activity.

To prevent destabilizing contractionary effects in the long term, $\gamma$ is subject to governance calibration via ParamHub, allowing the protocol to progressively taper the burn coefficient as circulating supply declines or systemic liquidity conditions warrant adjustment. While no additional LNQ may be minted beyond the initial fixed supply of 1 billion tokens, governance maintains discretion over deflationary pressure through modulation of $\gamma$, enabling equilibrium between value accrual and long-horizon network participation.

Token velocity is indirectly governed via exchange-theoretic dynamics:

$$V_t = \frac{P_t \cdot Q_t}{M_t}$$

where $V_t$ is token velocity, $P_t$ the aggregate price level, $Q_t$ transaction volume, and $M_t$ the token supply. By coupling burns to transaction-linked GMV, the mechanism anchors monetary throughput to usage fundamentals, mitigating inflation/deflation spirals and aligning monetary contraction with utility rather than speculation.

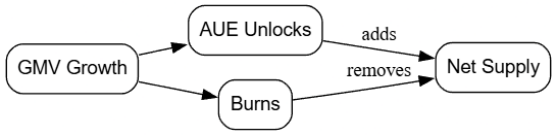


*Diagram 1.3 - Conceptual inflation vs deflation mechanism*

# Section 8: On-Chain Dispute Resolution via a Three-Tiered System

Dispute resolution is essential in decentralized systems to adjudicate conflicts arising from off-chain execution, such as failed tasks or misconduct, without relying on centralized authorities. In blockchain ecosystems, on-chain mechanisms leverage cryptoeconomic incentives to achieve fair outcomes, but they must balance efficiency, cost, and security to prevent capture or inefficiency. LinqProtocol's three-tiered system escalates from automated checks to community and council involvement, tying directly to participant roles (e.g., watcher nodes supply evidence, governance jurors vote via veLNQ) and antagonistic incentives (slashes/rewards) for alignment. This hybrid approach addresses the "oracle problem" in DePINs, where external data integration risks manipulation, by incorporating probabilistic verification and game-theoretic deterrence. Modeled after decentralized arbitration protocols like Kleros, it ensures resolution costs remain low while maintaining scalability. [15]

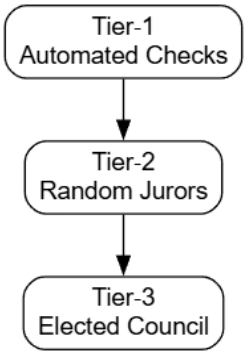


*Diagram 1.4 - Three-tier dispute-resolution ladder*

## 8.1 Tier 1: Automated

The first tier employs smart contract logic for rapid, objective resolution of straightforward disputes, minimizing human intervention. Triggers include timeouts (e.g., provider non-response beyond lease terms) and watcher-derived evidence: uptime proofs or spot benchmarks verify performance, with failures automatically slashing stakes (e.g., 20-50%) and adjusting reputation scores in the ProviderRegistry. Watcher nodes (short-term centralized/elected, long-term provider opt-in) supply on-chain attestations, tying directly to their treasury-funded role in semi-verifiability.

This automation draws from verifiable delay functions in blockchain consensus, ensuring deterministic outcomes with low latency. If unresolved (e.g., ambiguous benchmarks), escalation occurs, bounding costs to gas fees.

### 8.2 Tier 2: Community Jurors

For disputes too complex for automated resolution (e.g., subjective assessments of task quality), Tier 2 engages veLNQ-staked community jurors. When a Tier 2 dispute arises, the protocol randomly samples a governance-defined fraction (default 10%) of veLNQ stakers. From this sample pool, the first 6–8 jurors (number defined by ParamHub) who explicitly "lock-in" their commitment occupy seats; others may freely ignore the summons without penalty. Jurors bond a portion of their veLNQ stake upon lock-in, but voting commences only after all juror seats are confirmed filled, at which point a 48-hour window opens.

Votes remain encrypted until all jurors have submitted ballots, eliminating the risk of early-majority signaling and ensuring unbiased decision-making. Minority voters subsequently forfeit a configurable fraction of their bonded stake ($\mathrm{minoritySlashRate}$); these slashes are directly redistributed pro-rata to jurors on the majority side. If the jury's verdict is unanimous, no redistribution occurs, emphasizing accuracy over arbitrary financial incentives. Jurors choosing non-participation see neither reward nor punishment, ensuring that absence due to external circumstances, such as unexpected absence, does not lead to unintended penalties.

Expected value for a juror who locks in can now be written in closed form.

Let

- $s$ be the juror's bonded stake,
- $\sigma$ the dispute minority-slash rate set in **ParamHub**, and
- $p_{\mathrm{maj}}$ the juror's subjective probability of landing on the majority side (a function of evidence and coordination heuristics).

Because a passing verdict transfers the forfeited minority bonds to the majority, while a failing verdict transfers an identical fraction from the juror's own bond, the net payoff over the 48-hour window is

$$\mathbb{E}[R] \;=\; \sigma s\,(2p_{\mathrm{maj}} - 1) \;-\; c_{\mathrm{lock}},$$

where $c_{\mathrm{lock}}$ captures the opportunity cost of immobilising capital (foregone earnings and liquidity premium). A rational actor engages only if $\mathbb{E}[R] > 0$, which reduces to

$$p_{\mathrm{maj}} \;>\; \frac{1}{2}\Big(1 + \tfrac{c_{\mathrm{lock}}}{\sigma s}\Big).$$

With representative values $\sigma$ = 0.10 and $c_{\mathrm{lock}}/s \approx 0.01$ for a two-day lock. The break-even accuracy threshold is ~0.55. Thus a juror who is at least 60 % confident in their assessment (empirically $p_{\mathrm{maj}} \approx 0.7$ in well-signalled disputes) expects a positive return, while anyone less certain rationally declines the summons rather than risk loss by potentially siding with the minority. The expression formalises the intuitive Schelling incentive already embedded in the mechanism: lock-in makes economic sense only when a juror believes they can coordinate on truth more often than chance, ensuring that inattentive or uninformed participants stay on the sidelines and that the seated panel converges on high-confidence voters.

### 8.3 Tier 3: Council

Tier 3, reserved for highest-stake or deadlocked cases, invokes a separately elected council whose

members serve fixed terms after governance ballot election. Dispute council size (3, 4, 6, or 8 members) is a ParamHub variable. Disputes escalated to Tier 3 activate only after every council seat is explicitly acknowledged, ensuring confirmed availability and engagement from proven, governance-elected members. Council candidates provide deeper contextual understanding through their prior governance participation.

Council decision-making protocols mirror Tier 2 rules, including a 48-hour voting window, encrypted ballots until all council members vote, and the redistribution of minority slashes to majority voters. However, initial stakes and slashing rates at Tier 3 are significantly elevated compared to Tier 2, proportionate to the higher gravity and financial implications of escalated disputes.

### 8.4 Incentives/Analysis

Incentives within LinqProtocol's dispute-resolution mechanism are structured to align juror behavior with truthful, accurate outcomes through cryptoeconomic incentives. Jurors who voluntarily participate by staking their veLNQ tokens face potential slashing if voting in the minority position, effectively penalizing incorrect or out-of-consensus judgments. Conversely, jurors voting with the majority position receive proportionate rewards, funded directly from the slashed stakes of minority jurors. This internal redistribution ensures a closed-loop incentive system, promoting rational engagement and minimizing malicious or arbitrary voting.

Opt-in participation ensures that jurors explicitly accept these potential risks and rewards, thus self-selecting for confidence and availability. Jurors unwilling or unable to commit face neither rewards nor penalties, preventing punitive consequences for non-participation due to external factors such as temporary unavailability.

Through this game-theoretic approach, LinqProtocol ensures that jurors' incentives remain strongly aligned with truthful consensus outcomes, reinforcing the integrity and efficacy of the decentralized dispute-resolution framework.

## Section 9: Role of Governance in Relation to Previous Sections

Governance in decentralized systems orchestrates parameter adjustments and dispute oversight, ensuring adaptability while preventing capture in trustless environments. For LinqProtocol, governance integrates prior elements including antagonistic incentives, AUE, and dispute tiers through a DAO framework that evolves from centralized controls to community-driven decisions. This role mitigates risks like parameter rigidity or adversarial exploits, drawing from blockchain governance models where voting mechanisms balance efficiency and decentralization. Analyses of DAO vulnerabilities emphasize the need for incentive-aligned structures to sustain long-term viability on LinqProtocol.

### 9.1 Framework

The governance framework centers on a DAO utilizing veLNQ for voting power (see **Section 3.4**): stakes time-locked in pools enabling proposals and resolutions. Key parameters include AUE rates (emissions thresholds), burn gamma (deflation fraction), VEPC limits and multipliers, staking mins (collateral floors), and developer grants which are proposed and passed via votes, streamed to qualifying providers (e.g., meeting

registry criteria), with treasury funding limited to AUE-derived inflation to cap expenditures. See **Glossary** for details on ParamHub adjustable parameters.

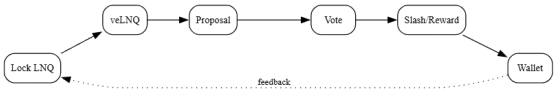


*Diagram 1.5 - Governance & veLNQ feedback loop*

### 9.2 Decentralization Roadmap

The roadmap progresses from an initial centralized ParamHub (team-managed for launch stability) to full DAO handover over time. Early phases focus on security audits and parameter bootstrapping, transitioning via phased veLNQ activation to decentralize control.

This staged approach aligns with blockchain decentralization metrics, where gradual shifts enhance resilience without early vulnerabilities, ensuring governance evolves with network maturity.

## Section 10: Data Storage On- and Off-Chain

Data storage in blockchain systems must navigate the tension between on-chain immutability and off-chain scalability, as full on-chain persistence incurs prohibitive costs while off-chain solutions risk centralization or data loss. In DePIN ecosystems, hybrid models optimize for efficiency, leveraging decentralized protocols like IPFS for durability. LinqProtocol employs a phased approach: essential data on-chain for transparency, with off-chain storage transitioning from IPFS to provider-hosted solutions. Cost models, informed by gas analyses on rollups like Arbitrum, project minimal gas costs per transaction, bounding overheads. This section delineates on/off-chain essentials, trade-offs, and security, drawing from blockchain storage literature where hybrid designs reduce latency by 20-50% while preserving integrity. [16]

### 10.1 On-Chain

On-chain storage in LinqProtocol is reserved for essentials requiring immutable auditability: escrows (lease deposits/payouts), votes (governance proposals/juror decisions), and the ProviderRegistry (reputation scores, staking details). Deployed on Arbitrum, transactions incur small gas costs, enabling low-friction updates without L1 Ethereum's significantly higher fees. This selective approach aligns with cost-efficient designs in rollup ecosystems, where data availability layers minimize expenses for verifiable state.

### 10.2 Off-Chain

Off-chain storage begins with IPFS for initial durability, where content-addressed files (e.g., job logs, benchmarks) are pinned across nodes to mitigate the risk of data loss. The system transitions to persistent, redundant storage provided by LNQ providers via opt-in Kubernetes-managed services, with data replicated across multiple nodes (typically 3–5) to enhance resilience. Storage operations are funded by the L2 treasury (derived from protocol fees), with metadata and consensus verifications (e.g., Merkle root hashes) persisted in Postgres databases and referenced on-chain to maintain agreement and integrity.

To prevent indefinite repository growth and manage storage efficiency, a garbage collection mechanism is implemented. Data types such as logs, metrics, and benchmark results from random

watcher checks are compressed and stored in object storage for approximately one year before automatic deletion. Persistent storage backups (e.g., database snapshots and state files) are rotated monthly, with older backups purged after this retention period. This selective data lifecycle management ensures availability of essential historical metrics and audits without accumulating redundant or unnecessary data, supporting a sustainable off-chain storage approach.

### 10.3 Trade-Offs/Security

Hybrid costs blend on-chain gas (for essentials) with off-chain pinning, yielding enormous savings compared to full on-chain data storage. Security relies on checksums and pinning (IPFS/provider redundancy against failures).

## Section 11: Known Limitations, and Incentive Mitigation

No system is without constraints, and LinqProtocol's design, while innovative, acknowledges inherent limitations in decentralized compute. This section enumerates key limitations with empirical grounding, and articulates how antagonistic incentives mitigate these issues. Extensions outline pathways for evolution, ensuring LNQ's adaptability in the DePIN landscape.

### 11.1 Limitations

Watcher gaps represent a core limitation: as probabilistic monitors (uptime/spot benchmarks), they do not achieve full verifiability, leaving room for undetected edge-case failures in complex tasks, akin to oracle dilemmas in DeFi where partial checks suffice but risk incompleteness. Volatility persists despite variable escrow payout cliffs, as token prices can fluctuate up to 99% in market downturns, impacting budgeting even with mitigations. Cold-start problems hinder initial adoption where low network effects lead to user churn in early DePINs.

### 11.2 Incentive Mitigation

Bounds ensure economic security: stake requirements exceed rewards/exploits rendering attacks uneconomical (Filecoin parallels show similar thresholds deter majority of rational threats)[17] antagonistic design (slashes/reputation) overcomes non-verifiability by raising deviation costs above benefits.

### 11.3 Extensions

Zk hybrids could augment watchers for selective full proofs, reducing probabilistic gaps; LinqChain offers native scaling as a future hub for integrated services.

## Section 12: Arbitrum Ecosystem Integrations and Risks

Layer-2 (L2) solutions like Arbitrum are pivotal for scaling blockchain applications, addressing Ethereum's throughput limitations while inheriting its security. In DePIN contexts, L2 integrations enable high-volume transactions with reduced costs, but introduce trade-offs in decentralization and risk exposure. LinqProtocol's Arbitrum focus leverages optimistic rollups for efficient compute leasing, emphasizing low-friction ties to the Ethereum base layer. This section explores Arbitrum-specific integrations and analyzes risks/benefits through value flow models, with future LinqChain as a potential extension for native scaling.

### 12.1 Arbitrum-Specific Integrations

Arbitrum's optimistic rollup architecture provides scaling benefits by batching transactions off-chain and posting summaries to Ethereum, reducing gas fees by 10-100x compared to L1[12]. For LinqProtocol, this enables low-cost escrows and disputes, with throughput supporting ~640 to 6000 transactions per second depending on complexity of transactions,[18] far exceeding Ethereum's theoretical limit of 140TPS[19] (17.2TPS at time of writing),[20] as demonstrated in ZK rollup evaluations adaptable to optimistic models. Ties to the ETH base layer ensure settlement finality, with fraud proofs allowing challenges to invalid states, inheriting Ethereum's economic security for DePIN resilience.

### 12.2 Risks and Benefits

Sequencer risks include centralization, enabling censorship or front-running, as sequencers (often single entities) control transaction ordering; analyses bound these via incentives and bonds, but highlight liveness threats in delayed disputes. Fraud proofs mitigate invalid states but depend on honest challengers, with dynamic periods optimizing detection as in security frameworks. Benefits outweigh risks for low-friction hubs. In the future the option exists for LinqChain to extend as a native layer for seamless scaling.

## References/Bibliography


[1] **World Economic Forum & Capgemini.** (2025, June 3). Technology Convergence Report 2025. World Economic Forum. "Amidst the burgeoning DePIN sector, valued at over US $50 billion in market capitalization as of 2024 and projected to reach US $3.5 trillion by 2028." Retrieved from https://reports.weforum.org/docs/WEF_Technology_Convergence_Report_2025.pdf

[2] **Akash Network**, "The Rise of Decentralized Compute," *Akash Network Blog*, published circa early 2024. Claim: "On Akash, prices for cloud computing services are up to 85% lower than their centralized counterparts." Retrieved from http://akash.network/blog/the-rise-of-decentralized-compute/

[3] **Amazon Web Services**, "On-Demand Pricing – T2/T3/T4g Unlimited Mode CPU Credit Rates," *Amazon EC2 Pricing*, 2025. Key detail: "For T2 and T3 instances in Unlimited mode, CPU Credits are charged at: $0.05 per vCPU-Hour for Linux, RHEL and SLES, and $0.096 per vCPU-Hour for Windows and Windows with SQL Web." Retrieved from https://aws.amazon.com/ec2/pricing/on-demand/

[4] **Protocol Labs, Filecoin Project**, "Engineering Filecoin's Economy," *Filecoin Technical Report*, August 27 2020. Key statement: "In general, markets balance supply and demand… In the case of Filecoin, if you have extra storage and… those seeking data storage may be willing to pay… Aligning these two… results in a deal (market clearing)." Retrieved from https://filecoin.io/2020-engineering-filecoins-economy-en.pdf

[5] **Shelven Zhou (Phala Network)**, *"Performance Benchmark Results: SP1 zkVM on TEE H200 GPUs Hits <20% Overhead,"* Phala Network Blog, March 21 2025. Key claim: "The TEE overhead primarily comes from memory encryption … with overall overhead less than 20%". Retrieved from https://phala.network/posts/performance-benchmark-results

[6] **Gate,** "Render Deep Dive: Tokenomics, Adoption, Solana Move & Price Outlook," published July 24 2025. Key claim: "The community passed RNP-001… shifting RENDER from a fixed supply to a managed emissions system… allocated a 20% inflation pool… ~644 M max supply (up from 536 M). About 9.13 million RNDR tokens are being minted in the first year of BME as rewards." Retrieved from https://www.gate.com/crypto-wiki/article/render-deep-dive-tokenomics-adoption-solana-move-price-outlook

[7] **M. Wang and Q. Wu,** "Practical and Fast Intensive Validation on Blockchain," SSRN, 2023. Key point: "When the overhead becomes increasingly high, rational nodes choose to skip or refuse validation, which leads to the Verifier's Dilemma." Retrieved from https://papers.ssrn.com/sol3/Delivery.cfm/eb9eefb6-ad76-403d-93e6-7151a28f182b-MECA.pdf?abstractid=4601578

[8] **Eric Budish**, *"The Economic Limits of Bitcoin and the Blockchain,"* Working Paper (University of Chicago / NBER), June 2022. Key statement: "A necessary condition for no player to have a profitable attack is $\mathbf{p \geq V\_attack}$," Retrieved from https://ericbudish.org/wp-content/uploads/2022/06/

Economic-Limits-of-Bitcoin-and-Anonymous-Decentralized-Trust-on-the-Blockchain.pdf

[9] **Sreeram Kannan & Soubhik Deb (a16z Crypto)**, "The Cryptoeconomics of Slashing," a16z Crypto, January 25 2023. Key insight: "Comparing a lower-bound on the minimum cost for any adversary to mount an attack (cost-of-corruption) against an upper-bound on the maximum profit … indicates when it is economically profitable to attack the protocol. … Slashing can increase the cost-of-corruption, reducing or eliminating the total profit." Retrieved from: https://a16zcrypto.com/posts/article/the-cryptoeconomics-of-slashing/

[10] **Kleros Core Team**, *"Kleros: A Decentralized Justice Protocol for the Internet,"* Kleros Whitepaper, v1.0 (June 2018). Key insight: "Tokens are redistributed from jurors who voted incoherently to jurors who voted coherently. … Thomas Schelling … described 'focal point(s) for each person's expectation of what the other expects him to expect…' We expect agents to vote the true answer because they expect others to vote the true answer …" Retrieved from: https://kleros.io/whitepaper.pdf

[11] **Wikipedia**, *"Proportional–integral–derivative controller."*, Key detail: "The proportional (P) component responds to the current error value by producing an output that is directly proportional to the magnitude of the error. … The integral (I) component … considers the cumulative sum of past errors to address any residual steady-state errors. … The derivative (D) component predicts future error by assessing the rate of change of the error, which helps to mitigate overshoot and enhance system stability." Retrieved from: https://en.wikipedia.org/wiki/Proportional%E2%80%93integral%E2%80%93derivative_controller

[12] **Jesse Pollak (Coinbase Engineering),** "Supporting EIP-4844: Reducing Fees for Ethereum Layer 2 Rollups," Coinbase Blog, October 6, 2022. Key insight: "Enter EIP-4844: an upgrade to the Ethereum network that will reduce the cost of layer 2 rollups by 10-100x … With this change in place, we expect fees on layer 2 rollups to decrease by 10-100x." Retrieved from: https://www.coinbase.com/en-gb/blog/supporting-eip-4844-reducing-fees-for-ethereum-layer-2-rollups

[13] **Y. Sun, K. Li, W. Shi, and F. Yang,** "Blockchain-Based Crowdsensing: A Survey from a System Perspective," arXiv preprint arXiv:2309.12330, Sep. 2023. Available: https://arxiv.org/abs/2309.12330

[14] **M. T. C. Chiu, S. Mahajan, M. C. Ballandies, and U. V. Kalabić,** "DePIN: A Framework for Token-Incentivized Participatory Sensing," arXiv preprint arXiv:2405.16495, May 26, 2024. Available: https://arxiv.org/abs/2405.16495

[15] **C. Lesaege, N. Ast and S. Badreddin, Kleros,** Short and Long Whitepaper – Decentralized Justice Protocol, Kleros, Version 2.0, Nov. 2020. Available: https://kleros.io/whitepaper_long_en.pdf

[16] **A. Liu, J. Chen, K. He, R. Du, J. Xu, C. Wu, Y. Feng, T. Li, and J. Ma,** "DynaShard: Secure and Adaptive Blockchain Sharding Protocol with Hybrid Consensus and Dynamic Shard Management," arXiv preprint arXiv:2411.06895, Nov. 2024. Available: https://arxiv.org/abs/2411.06895

[17] **Protocol Labs (Juan Benet et al.),** Filecoin: A Decentralized Storage Network, July 19, 2017. Key detail: "Storage Miners pledge their storage to the network by depositing collateral … If some proofs of storage fail, a proportional amount of collateral is lost" Retrieved from: https://filecoin.io/filecoin.pdf

[18] **Offchain Labs, Arbitrum Chain Launch FAQ & Troubleshooting: Building the Arbitrum Chain,** "For LinqProtocol, this enables low-cost escrows and disputes, with throughput supporting ~640 to 6000 transactions per second depending on complexity of transactions." Retrieved from: https://docs.arbitrum.io/launch-arbitrum-chain/faq-troubleshooting/troubleshooting-building-arbitrum-chain

[19] **Ethereum, Wikipedia,** The Free Encyclopedia, "As of March 2025, the maximum theoretical throughput for Ethereum is 142 TPS (given its current 36M gas limit, 12s blocks, and 21k gas cost for ETH transfers)." accessed July 24, 2025. Retrieved from: https://en.wikipedia.org/wiki/Ethereum

[20] **Etherscan,** *Ethereum (ETH) Blockchain Explorer*, "Transactions … 2,905.01 M (17.3 TPS)", accessed July 24, 2025. Available: https://etherscan.io/

[21] **Protocol Labs**, Filecoin Project. (2020, August 27). Engineering Filecoin's Economy (Technical Report). Key point: "high collateral creates barriers to miners joining the network," Retrieved from: https://filecoin.io/2020-engineering-filecoins-economy-en.pdf

# Appendix

## Glossary of Terms

This glossary defines key terms used throughout the litepaper, providing clarity on LinqProtocol's cryptoeconomic concepts and mechanisms.

- **AUE (Adaptive Unlock Emissions)**: A mechanism simulating controlled minting for LNQ's upfront 1 billion supply, unlocking tokens from locked pools based on network metrics like active leases, job volume, and staked wLNQ, with aggressive relocks for small networks to bound inflation.
- **DePIN (Decentralized Physical Infrastructure Network)**: A blockchain-based network that incentivizes decentralized provision of physical resources, such as compute power in LinqProtocol.
- **Escrow**: On-chain locked funds (in LNQ or hedged equivalents) deposited by requestors for leases, released upon completion or refunded with penalties on cancellation.
- **GMV (Gross Marketplace Value)**: The total value of jobs and leases processed in the network, denominated in LNQ equivalents, used to bound inflation, deflation, and emissions.
- **Lease**: A fixed-term rental of compute resources (e.g., 30 days at a set LNQ rate), requiring full up-front escrow deposits, renewable with another full-term deposit, and tied to hedges for rate stability.
- **LinqChain**: A potential future native hub for LinqProtocol, enabling expanded services with reduced friction, though

deprioritized in favor of ETH/Arbitrum focus.

- **LNQ**: The native utility token of LinqProtocol, serving multi-role functions including exchange, staking, and governance.
- **Paramhub**: An initial centralized module for managing adjustable parameters, transitioning to DAO control over time.
- **ProviderRegistry**: An on-chain database tracking providers' reputation scores, staking details, and eligibility, updated via disputes, slashes, and benchmarks.
- **Reputation Score**: A numerical, non-decaying metric for providers, increasing on successful jobs and decreasing on failures, benchmarks, or uptime lapses, used to offset staking requirements with bounded multipliers.
- **Variable Escrow Payout Cliffs (VEPC):** An escrow mechanism that allows a bounded portion of a lease's value to be paid out immediately to a provider at job start, with eligibility strictly determined by the provider's staked collateral and reputation score. VEPC mitigates token volatility for providers by front-loading critical operating costs while preserving full LNQ-denominated settlement and cryptoeconomic security.
- **Immediate Payout (Base Cost Payout):** The portion of escrowed LNQ released to a provider at lease initiation to cover near-term operating expenses such as electricity or bandwidth. Immediate payouts are optional, hard-capped algorithmically, and subordinate to slashing, ensuring extracted value never exceeds capital at risk.
- **Maximum Immediate Cost (MaxImmediateCost):** The enforced upper bound on the immediate payout a provider may request for a given lease, computed as a function of the provider's reputation score and staked collateral. This value is governed by ParamHub parameters and ensures that volatility protection scales only with demonstrated trust and capital commitment.
- **Reputation-Weighted Haircut:** A risk adjustment applied to immediate payout eligibility that reduces the amount a provider can extract upfront to account for uncertainty and downside price risk. Higher reputation scores lower the effective haircut by reducing the collateral required to qualify for a given immediate payout.
- **Downside Volatility Target:** A governance-defined risk parameter representing the assumed magnitude of adverse token price movement used when bounding immediate payout eligibility. The downside volatility target is not a guarantee, but a configurable safety margin that guides how conservatively VEPC caps upfront value extraction.
- **Provider Base Cost Declaration:** A provider-submitted statement of near-term operating expenses requested for immediate payout at lease start. Base cost declarations do not guarantee payout and are accepted only if they fall within the protocol-defined MaxImmediateCost bound.
- **Capital at Risk:** The combined value of LNQ staked as provider collateral and any remaining escrow balance that may be slashed or reclaimed in the event of

non-performance, dishonesty, or dispute resolution. VEPC ensures that immediate payouts never exceed capital at risk under governance-defined volatility assumptions.

- **Semi-Verifiability**: Probabilistic validation of compute tasks via watcher nodes' uptime checks and spot benchmarks, rather than full cryptographic proofs, balancing efficiency and trust.
- **veLNQ (Vote-Escrowed LNQ)**: Time-locked staked LNQ granting amplified voting power in governance pools, used for proposals, disputes, and juror qualifications.
- **Watcher Nodes**: Nodes performing uptime monitoring and spot benchmarking; short-term centralized/elected, long-term opt-in by providers via Kubernetes, funded by L2 treasury for network verification.
- **wLNQ (Wrapped LNQ)**: A bridged version of LNQ on Arbitrum L2, used in staking metrics for AUE calculations.

## Paramhub Adjustable Parameters

The Paramhub initially manages key tunable parameters to ensure network flexibility, with values set conservatively at launch and transitioned to DAO governance. Below is a non-exhaustive list of adjustable parameters, including initial defaults (hypothetical based on simulations) and rationale:

- **AUE Factors (PID Coefficients)**: Proportional (P=0.5), Integral (I=0.1), Derivative (D=0.2) for emissions control; adjustable to fine-tune inflation response to GMV/leases/staking deviations.
- **Burn Gamma**: 0.25 (25% of fees/GMV burned); tunable to modulate deflationary pressure, e.g., increase to 0.3 for scarcity in high-growth phases.
- **Staking Mins (Collateral Floors)**: Provider base = 100 LNQ, juror pool entry = 500 LNQ; modifiable to reflect risk levels or prevent under-collateralization.
- **Slash Fraction**: 0.2-0.5 for providers/jurors; adjustable per tier (e.g., minority vote slashes at 0.1) to calibrate antagonism.
- **Insurance Buffer Allocation**: 0.05-0.1 (5-10% of fees); tunable to hedge coverage based on volatility simulations.
- **Reputation Multiplier Bounds**: Min 0.5x, Max 2x offsets; gov-set to bound exploits while rewarding high scores.
- **Lease Penalty Rate**: 0.05-0.1 (5-10% on cancellations); adjustable to deter abuse without stifling usage.
- **Juror Quorum Threshold**: 10% veLNQ participation; modifiable for dispute efficiency.
- **Developer Grant Caps**: Limited to 10% of AUE-derived treasury inflation per cycle; tunable to prioritize ecosystem growth.
- **Proposal Bond Size:** $B$ - 1 000 LNQ (baseline). Capital the proposer must lock when submitting a protocol vote; at risk if the motion fails.
- **Proposal Fee Fraction:** $f - 0.05 \times B$. Non-refundable slice of the bond that funds the insurance buffer and deters spam.
- **Minority Slash Rate:** $\sigma_m - 0.10$. Portion of each minority voter's stake burned and redistributed to the majority when a proposal passed.

- **Proposer Slash Rate:** $\sigma_p$ – 1.00. Fraction of the proposer's bond confiscated when a proposal fails (full loss by default).
- **Proposer Premium:** $\rho$ – 0.02. Interest multiplier on the refunded bond when the proposal passes; paid from slashed minority stakes.
- **Dispute Council Size:** The size of the council formed from elected jurors to vote on Tier 3 dispute resolution. (eg. 4, 6 or 8)
- **Dispute Minority Slash Rate**: The portion of a stake that is slashed in the case of a Tier 2 minority.