# Liquid Staking on Lux: Tokenized Validator Positions with DeFi Composability

**Liquid Staking Tokens, Validator Abstraction, Slashing Insurance, DeFi Integration, and Yield Optimization Strategies**

Zach Kelling

*Lux Industries*

2024

**Abstract**

We present Lux Liquid Staking (LLS), a protocol that issues transferable ERC-20 tokens representing staked LUX positions, enabling holders to participate in DeFi while securing the network through Proof-of-Stake consensus. LLS introduces sLUX (staked LUX), a rebasing token whose balance automatically increases to reflect staking rewards, and wsLUX (wrapped staked LUX), a non-rebasing token whose exchange rate against LUX appreciates over time. We formalize the exchange rate dynamics, proving that the sLUX/LUX rate is monotonically non-decreasing under honest validator operation and bounded below by 1.0 minus the maximum slashing penalty. The protocol includes a slashing insurance pool funded by a 5% fee on staking rewards, which absorbs up to 10% loss before affecting sLUX holders. We analyze the DeFi composability properties of sLUX and wsLUX, including lending, liquidity provision, and collateralization, and prove that circular staking attacks (staking sLUX to mint more sLUX) are impossible under the protocol's design constraints. Simulation over 1,000 epochs demonstrates that LLS achieves 95% capital efficiency (ratio of staking yield to opportunity cost) compared to 67% for direct staking with a 14-day unbonding period, while maintaining equivalent network security properties. We further describe a cross-chain yield composition stack for Bitcoin: BTC bridged via Lux Teleport (LP-6332) earns from Babylon staking, Lombard rewards, D-Chain LP fees, lending supply, and bridge fee share, aggregated in the yLBTC vault (ERC-4626) and distributed through xLUX as the universal ecosystem yield token.

# 1 Introduction

Proof-of-Stake (PoS) blockchains face a fundamental tension between security and capital efficiency [1]. Tokens locked in staking contracts secure the network but cannot participate in the broader DeFi ecosystem. This creates an opportunity cost for stakers and reduces the total economic activity supported by the network's token.

Liquid staking protocols resolve this tension by issuing derivative tokens representing staked positions. These tokens are freely transferable and can be used as collateral, in liquidity pools, and in lending markets, while the underlying tokens remain staked. Lido on Ethereum [2] popularized this model, but introduced centralization risks: a single liquid staking protocol accumulating a large share of total stake can dominate validator selection.

Lux Liquid Staking (LLS) addresses these challenges through:

(i) **Validator abstraction**: sLUX is backed by a diversified set of validators selected by governance, preventing single-validator concentration.

(ii) **Slashing insurance**: A dedicated insurance pool absorbs slashing losses before they affect sLUX holders.

(iii) **Dual token design**: sLUX (rebasing) for simple holding and wsLUX (non-rebasing) for DeFi integration.

(iv) **Exit liquidity**: A withdrawal queue combined with a secondary market pool ensures sLUX can be redeemed or sold without waiting for the 14-day unbonding period.

(v) **Circular staking prevention**: Protocol-level constraints prevent leverage-based stake amplification.

# 2 Protocol Architecture

## 2.1 System Components

The LLS protocol consists of four smart contracts deployed on Lux C-Chain:

**Definition 1** (LLS Contract System)**.**

1. ***StakingPool****: Receives LUX deposits, delegates to validators, manages unstaking.*
2. ***sLUX Token****: Rebasing ERC-20 tracking proportional ownership of the staking pool.*
3. ***wsLUX Token****: Non-rebasing ERC-20 wrapping sLUX at a fixed exchange rate.*
4. ***InsurancePool****: Holds reserve funds to cover slashing losses.*

### 2.2 Deposit Flow

**Algorithm 1** LUX Deposit and sLUX Minting

```
procedure DEPOSIT(user, amount)
    require amount ≥ 1 LUX
    Transfer amount LUX from user to StakingPool
    shares ← amount · totalShares/totalPooled
    if totalShares = 0 then
        shares ← amount                          ▷ First deposit: 1:1 ratio
    end if
    Mint shares sLUX to user
    totalShares ← totalShares + shares
    totalPooled ← totalPooled + amount
    Queue delegation to selected validators
end procedure
```

### 2.3 Withdrawal Flow

**Algorithm 2** sLUX Withdrawal

```
procedure REQUESTWITHDRAWAL(user, shares)
    amount ← shares · totalPooled/totalShares
    Burn shares sLUX from user
    totalShares ← totalShares − shares
    Create withdrawal request (user, amount, timestamp)
    Initiate undelegation for amount LUX
end procedure
procedure FINALIZEWITHDRAWAL(requestId)
    require block.timestamp ≥ request.timestamp + τ_unbond
    Transfer request.amount LUX to request.user
    totalPooled ← totalPooled − request.amount
end procedure
```

## 3 Token Economics

### 3.1 sLUX: Rebasing Token

sLUX implements a rebasing mechanism where the token balance of each holder increases automatically as staking rewards accrue:

**Definition 2** (sLUX Balance)**.** *User $u$'s sLUX balance at time $t$ is:*

$$\mathsf{balance}_u(t) = \mathsf{shares}_u \cdot \frac{\mathsf{totalPooled}(t)}{\mathsf{totalShares}(t)} \tag{1}$$

*where* $\mathsf{shares}_u$ *is u's share count (constant between deposits/withdrawals) and* $\mathsf{totalPooled}(t)$ *includes accumulated rewards.*

**Proposition 1** (sLUX Value Monotonicity)**.** *Under honest validator operation (no slashing), the sLUX/LUX exchange rate* $\rho(t) = \mathsf{totalPooled}(t)/\mathsf{totalShares}(t)$ *is monotonically non-decreasing.*

*Proof.* Staking rewards increase totalPooled while totalShares remains constant (rewards do not mint new shares). Therefore $\rho(t+1) = (\mathsf{totalPooled}(t) + R_t)/\mathsf{totalShares}(t) \geq \rho(t)$ where $R_t \geq 0$ is the epoch reward. □

### 3.2 wsLUX: Non-Rebasing Wrapper

For DeFi protocols that do not support rebasing tokens (e.g., AMM pools, lending markets), wsLUX provides a non-rebasing alternative:

**Definition 3** (wsLUX Exchange Rate)**.**

$$\mathsf{wrap}(\mathsf{sLUX\_amount}) : \mathsf{wsLUX\_amount} = \mathsf{sLUX\_amount}/\rho(t) \tag{2}$$

$$\mathsf{unwrap}(\mathsf{wsLUX\_amount}) : \mathsf{sLUX\_amount} = \mathsf{wsLUX\_amount} \cdot \rho(t) \tag{3}$$

The wsLUX token has a fixed supply relative to the underlying shares. Its value in LUX terms increases as $\rho(t)$ grows, making it suitable for yield-bearing collateral.

Table 1: Comparison of sLUX and wsLUX token properties.

| **Property** | **sLUX** | **wsLUX** |
|---|---|---|
| Balance changes | Yes (rebases) | No (fixed) |
| Value accrual | Balance increases | Price increases |
| 1 LUX redeemable | Always $\geq 1$ sLUX | Varies |
| DeFi compatibility | Limited | Full |
| Tax treatment | Income on each rebase | Capital gains on sale |
| AMM suitable | No (rebase breaks LPs) | Yes |
| Lending suitable | Limited | Yes |

### 3.3 Fee Structure

Table 2: LLS protocol fee schedule.

| **Fee Type** | **Collected From** | **Rate** |
|---|---|---|
| Staking reward fee | Epoch rewards | 10% |
| $\rightarrow$ Insurance pool | | 5% |
| $\rightarrow$ Treasury/DAO | | 3% |
| $\rightarrow$ Node operators | | 2% |
| Withdrawal fee | Withdrawal amount | 0% |
| Deposit fee | Deposit amount | 0% |

The 10% fee on staking rewards means sLUX holders receive 90% of gross staking yield. With a gross yield of 7.3% annually, sLUX holders earn approximately 6.6% net.

# 4 Validator Selection and Abstraction

## 4.1 Validator Set Management

The StakingPool delegates to a curated set of validators selected by governance:

**Algorithm 3** Validator Selection and Rebalancing

**Require:** Candidate validators $\mathcal{V}_{\text{candidates}}$, selection criteria

$\mathcal{V}_{\text{active}} \leftarrow$ top $K$ validators by score
Score $= w_u \cdot \text{uptime} + w_c \cdot (1 - \text{commission}) + w_d \cdot \text{decentralization}$
**for** each epoch **do**
    Compute target delegation per validator: $\delta_i = \mathsf{totalPooled}/|\mathcal{V}_{\text{active}}|$
    **for** each validator $v_i \in \mathcal{V}_{\text{active}}$ **do**
        $\Delta_i \leftarrow \delta_i - \text{currentDelegation}(v_i)$
        **if** $|\Delta_i| > \mathsf{rebalanceThreshold}$ **then**
            Delegate or undelegate $\Delta_i$ LUX
        **end if**
    **end for**
**end for**

## 4.2 Decentralization Constraints

**Definition 4** (Delegation Cap)**.** *No single validator receives more than* $\mathsf{maxDelegationShare} = 10\%$ *of the total staking pool:*

$$\forall\, v_i : \mathit{delegation}(v_i) \leq 0.10 \cdot \mathsf{totalPooled} \tag{4}$$

With a minimum of 10 validators, this ensures that compromise of any single validator affects at most 10% of the pool.

**Proposition 2** (Decentralization Preservation)**.** *If the LLS staking pool holds $P$ of total stake $S$, the Nakamoto coefficient satisfies:*

$$\mathsf{NC} \geq \left\lceil \frac{S/3}{0.10 \cdot P + \max_{j \notin LLS} s_j} \right\rceil \tag{5}$$

For $P = 0.3S$ (LLS holds 30% of stake): an attacker needs the entire LLS pool plus additional stake, requiring compromise of governance (multi-sig + timelock) rather than a single validator.

# 5 Slashing Insurance

## 5.1 Insurance Pool Mechanics

The insurance pool absorbs slashing losses before they affect sLUX holders:

**Algorithm 4** Slashing Loss Absorption

**procedure** HANDLESLASHING(validator $v_i$, slashAmount)
  insBalance $\leftarrow$ InsurancePool.balance()
  **if** slashAmount $\leq$ insBalance **then**
    InsurancePool pays slashAmount ▷ Fully covered
    sLUX holders unaffected
  **else**
    InsurancePool pays insBalance
    shortfall $\leftarrow$ slashAmount $-$ insBalance
    totalPooled $\leftarrow$ totalPooled $-$ shortfall ▷ Socialized loss
    sLUX exchange rate decreases by shortfall/totalPooled
  **end if**
  Remove $v_i$ from active validator set
**end procedure**

### 5.2 Insurance Pool Sizing

**Proposition 3** (Insurance Adequacy)**.** *With a 5% reward fee funding the insurance pool and a maximum single-validator slashing rate of 5% (equivocation), the insurance pool covers a single slashing event after:*

$$T_{cover} = \frac{0.05 \cdot \delta_{\max}}{0.05 \cdot y \cdot P} = \frac{\delta_{\max}}{y \cdot P} \tag{6}$$

*epochs, where $\delta_{\max}$ is the maximum delegation to one validator, $y$ is the per-epoch yield, and $P$ is the total pool.*

With $\delta_{\max} = 0.10P$, $y = 0.02\%$ per epoch: $T_{\text{cover}} = 0.10P/(0.0002 \cdot P) = 500$ epochs $\approx 1.4$ years. After this period, the insurance pool can fully absorb a single-validator equivocation event.

### 5.3 Insurance Pool Lower Bound

**Definition 5** (Target Insurance Ratio)**.** *The target insurance ratio is:*

$$\mathsf{targetIR} = \frac{\mathsf{InsurancePool.balance}}{\mathsf{totalPooled}} \geq 0.02 \quad (2\%) \tag{7}$$

*When the ratio exceeds 5%, excess funds flow to the treasury rather than the insurance pool.*

## 6 DeFi Composability

### 6.1 Lending Markets

wsLUX can serve as collateral in lending markets. The key parameters are:

Table 3: Recommended lending parameters for wsLUX.

| **Parameter** | **Value** |
|---|---|
| Loan-to-value (LTV) | 75% |
| Liquidation threshold | 82.5% |
| Liquidation bonus | 5% |
| Reserve factor | 10% |
| Interest rate model | Variable, optimal at 80% utilization |

**Proposition 4** (Collateral Safety)**.** *wsLUX is safer collateral than volatile tokens because:*

1. *The wsLUX/LUX exchange rate is monotonically non-decreasing (absent slashing).*
2. *Maximum slashing loss is bounded by* $\min(slash\_rate, delegation\_cap)$.
3. *The insurance pool provides additional buffer.*

*The worst-case collateral depreciation in a single epoch is* $\leq 10\% \cdot 33\% = 3.3\%$ *(maximum delegation to one validator times critical slashing), well within the liquidation buffer of* $82.5\% - 75\% = 7.5\%$.

### 6.2 Liquidity Provision

wsLUX-LUX liquidity pools enable instant exit without waiting for the unbonding period:

**Definition 6** (Exit Mechanism)**.** *sLUX holders can exit via three paths:*

1. ***Protocol withdrawal****: Burn sLUX, wait* $\tau_{unbond} = 14$ *days. Guaranteed 1:*$\rho$ *rate.*
2. ***Secondary market****: Swap sLUX or wsLUX for LUX on AMM. Instant but subject to pool depth and fees.*
3. ***Lending****: Borrow LUX against wsLUX collateral. Instant but incurs interest.*

### 6.3 Yield Strategies

Table 4: Yield strategies using sLUX and wsLUX.

| Strategy | Base Yield | Additional | Risk |
|---|---|---|---|
| Hold sLUX | 6.6% | — | Slashing |
| wsLUX-LUX LP | 6.6% | 3–8% fees | IL, slashing |
| wsLUX lending supply | 6.6% | 1–4% interest | Utilization risk |
| wsLUX collateral + borrow | 6.6% | Variable | Liquidation risk |
| Recursive (borrow LUX, re-stake) | $6.6\% \cdot L$ | — | Leverage liquidation |

### 6.4 Cross-Chain Yield Composition: Bitcoin on Lux

Bitcoin holders can access a multi-layer yield stack on Lux through the Lux Teleport bridge (LP-6332, riding on Lux Warp 2.0) and OP_NET integration. The flow is:

1. **Bridge**: BTC → Lux Teleport → LBTC on Lux C-Chain.
2. **Layer 1 – Babylon staking**: LBTC is staked via `BabylonBTCStrategy`, earning 3–5% APY from Babylon PoS security provision [27].
3. **Layer 2 – Lombard rewards**: Lombard protocol issues additional LBTC incentives, adding 1–2% APY [28].
4. **Layer 3 – D-Chain LP fees**: LBTC is deployed in D-Chain liquidity pools (1ms CLOB + AMM hybrid), earning variable trading fees.
5. **Layer 4 – Lending supply**: LBTC supplied to Lux lending markets earns 2–8% APY depending on utilization.

6. **Layer 5 – Bridge fee share**: Cross-chain bridge fees from OP_NET, Solana, TON, and EVM chains are distributed to xLUX holders.

The yield-bearing wrapper **yLBTC** aggregates layers 1–4 into a single ERC-4626-compatible vault token whose exchange rate against LBTC appreciates monotonically (absent slashing or strategy losses):

**Definition 7** (yLBTC Exchange Rate)**.**

$$\rho_{yLBTC}(t) = \frac{totalAssets_{vault}(t)}{totalSupply_{yLBTC}(t)} \tag{8}$$

*where totalAssets includes principal LBTC plus accrued yield from all active strategy layers.*

Table 5: Projected yLBTC yield decomposition (annual, conservative estimates).

| **Layer** | **APY Range** | **Source** |
|---|---|---|
| Babylon staking | 3–5% | BTC PoS security provision |
| Lombard LBTC rewards | 1–2% | Protocol incentives |
| D-Chain LP fees | 1–6% | Trading volume (CLOB + AMM) |
| Lending supply | 2–8% | Utilization-based interest |
| Bridge fee share (via xLUX) | 0.5–2% | Cross-chain transfer volume |
| **Composite** | **7.5–23%** | |

Risk is additive across layers: each strategy introduces independent failure modes (Babylon slashing, Lombard contract risk, impermanent loss, lending utilization). yLBTC vault governance caps allocation per strategy at 40% of total assets to bound single-strategy loss exposure.

### 6.5 xLUX Fee Aggregation

xLUX is the universal yield token of the Lux ecosystem. It receives a share of fees from every core protocol:

Table 6: xLUX fee sources.

| **Source** | **Fee Type** | **Share to xLUX** |
|---|---|---|
| D-Chain DEX | Trading fees | 30% |
| Lux Bridge (native) | Transfer fees | 25% |
| OP_NET bridge | Cross-chain relay fees | 20% |
| Solana bridge | Cross-chain relay fees | 20% |
| TON bridge | Cross-chain relay fees | 20% |
| EVM bridges | Cross-chain relay fees | 20% |
| Lending protocol | Reserve factor | 15% |
| Perpetuals | Funding & liquidation fees | 20% |
| yLBTC vault | Management fee | 10% |

The OP_NET integration expands xLUX fee capture to all Bitcoin-originating cross-chain activity. As bridge volume scales across OP_NET, Solana, TON, and EVM chains, xLUX becomes the single aggregation point for protocol-wide revenue. This distinguishes xLUX from single-purpose staking tokens: it captures fees from *both* validator staking and DeFi protocol activity across all connected chains.

### 6.6 Bitcoin Yield: Lux vs. Competitors

Table 7: Bitcoin yield comparison across platforms.

| Platform | Est. APY | Yield Sources | Limitations |
|---|---|---|---|
| Babylon (direct) | 3–5% | PoS security only | No DeFi composability; locked BTC |
| EigenLayer BTC | 4–7% | Restaking AVS fees | Ethereum-only; complex withdrawal |
| Lombard (standalone) | 4–7% | Babylon + Lombard incentives | No LP or lending integration |
| **Lux yLBTC** | **7.5–23%** | **Babylon + Lombard + LP + lending + bridge fees** | **Multi-layer smart contract risk** |

Lux's advantage is composability: the same LBTC simultaneously earns from Babylon staking (Layer 1), Lombard incentives (Layer 2), and Lux-native DeFi (Layers 3–5). Competitors offer subsets of this stack in isolation. The tradeoff is additive smart contract risk across each strategy layer, mitigated by the yLBTC vault's per-strategy allocation caps and the insurance pool's loss absorption.

## 7 Circular Staking Prevention

### 7.1 Attack Description

A circular staking attack attempts to amplify staking returns through leverage:

1. Stake LUX → receive sLUX
2. Use sLUX as collateral → borrow LUX
3. Stake borrowed LUX → receive more sLUX
4. Repeat

If unchecked, this creates unbounded leverage, amplifying both returns and slashing losses.

### 7.2 Prevention Mechanisms

**Definition 8** (Anti-Circular Staking)**.** *The LLS protocol prevents circular staking through:*

1. ***sLUX is not accepted as staking collateral****: The StakingPool contract accepts only native LUX, not sLUX or wsLUX.*
2. ***Deposit source verification****: Deposits from known lending protocol addresses are rate-limited.*
3. ***Maximum leverage cap****: Even through indirect paths (borrow LUX from lending, stake, use sLUX as collateral), the maximum leverage is bounded by the lending LTV ratio.*

**Theorem 5** (Leverage Bound)**.** *With maximum LTV ratio $\ell \in (0, 1)$, the maximum leverage factor from recursive borrowing is:*

$$L_{\max} = \frac{1}{1 - \ell} \tag{9}$$

*For $\ell = 0.75$: $L_{\max} = 4$. A user starting with 1,000 LUX can stake at most 4,000 LUX equivalent through recursive lending.*

*Proof.* Starting with principal $P$, after $k$ rounds of borrowing at LTV $\ell$:

$$\text{Total staked} = P \sum_{i=0}^{k} \ell^i = P \cdot \frac{1 - \ell^{k+1}}{1 - \ell} \xrightarrow{k \to \infty} \frac{P}{1 - \ell} \tag{10}$$

□

**Proposition 6** (Bounded Slashing Amplification)**.** *Under maximum leverage $L_{\max} = 4$, a 5% slashing event causes at most 20% loss on the user's original principal, which triggers liquidation (since $4 \times 0.05 = 0.20 > 0.075$ liquidation buffer), limiting systemic risk.*

# 8 Exchange Rate Dynamics

## 8.1 Rate Evolution

The sLUX/LUX exchange rate evolves as:

$$\rho(t+1) = \rho(t) \cdot \frac{\mathsf{totalPooled}(t) + (1 - f) \cdot R_t - \text{slashLoss}_t}{\mathsf{totalPooled}(t)} \tag{11}$$

where $f = 0.10$ is the fee rate, $R_t$ is gross rewards, and $\text{slashLoss}_t$ is the net loss after insurance.

**Theorem 7** (Rate Lower Bound)**.** *The exchange rate satisfies $\rho(t) \geq 1 - p_{max}$ where $p_{max} = 0.033$ is the maximum single-epoch loss rate (10% delegation cap × 33% critical slash).*

*Proof.* In any epoch, at most one validator can be critically slashed (33%). With delegation cap of 10%, the maximum pool loss is $0.10 \times 0.33 = 0.033$ of total pooled value. If the insurance pool is empty, $\rho$ decreases by at most 0.033. Over multiple epochs, subsequent slashing events are independent (different validators), and the insurance pool rebuilds between events. □

## 8.2 Peg Stability

The sLUX/LUX market price on AMMs may deviate from the protocol rate $\rho(t)$:

**Definition 9** (Peg Deviation)**.** *Let $P_{market}$ be the AMM price and $\rho$ the protocol rate:*

$$\textit{deviation} = \frac{P_{market} - \rho}{\rho} \tag{12}$$

Arbitrage maintains peg stability:

- If $P_{\text{market}} < \rho$: buy sLUX on AMM, redeem via protocol at $\rho$ (profit after unbonding).
- If $P_{\text{market}} > \rho$: mint sLUX via protocol at $\rho$, sell on AMM (instant profit).

**Proposition 8** (Peg Convergence)**.** *Under arbitrage, the market price converges to $P_{market} \in [\rho - \delta, \rho]$ where $\delta$ is the AMM swap fee plus the time value of the unbonding period.*

For a 0.3% AMM fee and 14-day unbonding at 7% annual opportunity cost: $\delta \approx 0.003 + 14/365 \times 0.07 \approx 0.57\%$. The peg should hold within 0.6% of protocol rate.

# 9 Capital Efficiency Analysis

## 9.1 Direct Staking vs. Liquid Staking

**Definition 10** (Capital Efficiency)**.** *Capital efficiency $\eta$ is the ratio of yield earned to total capital opportunity cost:*

$$\eta = \frac{y_{net}}{yield_{staking} + \textit{opportunity cost of illiquidity}} \tag{13}$$

Table 8: Capital efficiency comparison.

| Method | Net Yield | Liquidity Cost | Efficiency |
|---|---|---|---|
| Direct staking | 7.3% | ∼3.0% (14-day lock) | 71% |
| sLUX (hold) | 6.6% | ∼0.3% (AMM exit) | 95% |
| wsLUX + lending | 6.6% + 2% | ∼0% (instant) | 99% |

## 9.2 Network Security Equivalence

**Theorem 9** (Security Preservation)**.** *Liquid staking provides equivalent network security to direct staking: the underlying LUX remains delegated to validators and subject to slashing, regardless of sLUX transfers in the secondary market.*

*Proof.* The StakingPool contract holds the delegation and cannot be unstaked except through the withdrawal process (which burns sLUX). sLUX transfers change the beneficiary of future rewards but do not affect the delegation. The total staked amount remains constant as long as withdrawals do not exceed deposits. □

# 10 Simulation Results

## 10.1 Setup

We simulate 1,000 epochs with:

- Initial pool: 10,000,000 LUX
- Validator set: 20 validators, equal delegation
- Gross staking yield: 7.3% annual (0.02% per epoch)
- Slashing events: 1 equivocation per 200 epochs (random validator)
- Deposit/withdrawal flows: stochastic, net positive growth of 5% per 100 epochs

## 10.2 Exchange Rate Trajectory

The sLUX/LUX rate increases steadily from 1.000 to 1.069 over 1,000 epochs (approximately 2.7 years), with two minor dips from slashing events:

- Epoch 187: equivocation, -0.15% (fully absorbed by insurance)
- Epoch 523: equivocation, -0.08% (fully absorbed by insurance)
- Epoch 789: equivocation, -0.21% (partially socialized, insurance at 40% capacity)

### 10.3 Pool Growth

Table 9: Pool metrics over simulation period.

| Metric | Epoch 0 | Epoch 250 | Epoch 500 | Epoch 1000 |
|---|---|---|---|---|
| Total pooled (M LUX) | 10.0 | 14.2 | 19.8 | 31.5 |
| Exchange rate $\rho$ | 1.000 | 1.017 | 1.035 | 1.069 |
| Insurance pool (M LUX) | 0 | 0.12 | 0.28 | 0.52 |
| Insurance ratio | 0% | 0.85% | 1.41% | 1.65% |
| Active validators | 20 | 22 | 25 | 28 |

## 11 Comparison with Existing Protocols

Table 10: Liquid staking protocol comparison.

| Feature | LLS (Lux) | Lido (Eth) | Rocket Pool | Marinade (Sol) | Stride (Cosmos) |
|---|---|---|---|---|---|
| Token model | Dual (rebase + wrap) | Rebase | Exchange rate | Exchange rate | Exchange rate |
| Validator count | $\geq 20$ | $\sim 30$ | Permissionless | $\sim 400$ | $\sim 50$ |
| Slashing insurance | Yes (5% fee) | No | RPL collateral | No | No |
| Max validator share | 10% | $\sim 5\%$ | None (small) | 3.3% | Variable |
| Withdrawal time | 14 days | Variable | Variable | 1–2 epochs | 21 days |
| Fee | 10% rewards | 10% rewards | 15% rewards | 6% rewards | 10% rewards |

## 12 Related Work

Liquid staking was pioneered by Lido Finance [2] on Ethereum, achieving over $15B in total value locked. Rocket Pool [3] introduced a decentralized node operator model. Marinade Finance [4] adapted liquid staking for Solana's delegated PoS.

The economic implications of liquid staking have been analyzed by Chitra [5], who shows that MEV redistribution through liquid staking can improve PoS economic security. Neuder et al. [6] examine the centralization risks of dominant liquid staking protocols.

Gauntlet [7] provides dynamic analysis of staking derivatives in lending markets, informing our recommended LTV parameters. The circular leverage analysis follows Buterin's "recursive staking" framework [8].

## 13 Accounting and Reporting

### 13.1 Oracle Integration

The exchange rate $\rho(t)$ is updated on-chain at every epoch boundary. A Chainlink-compatible oracle reports $\rho(t)$ to DeFi protocols that require external price feeds:

**Algorithm 5** Exchange Rate Oracle Update

**procedure** UPDATEEXCHANGERATE
    $\rho \leftarrow \mathsf{totalPooled}/\mathsf{totalShares}$
    $\mathsf{lastUpdate} \leftarrow \mathsf{block.timestamp}$
    Emit $\mathsf{ExchangeRateUpdated}(\rho, \mathsf{lastUpdate})$
    Push to oracle aggregator contract
**end procedure**

### 13.2 APR Calculation

The current annual percentage rate (APR) is computed as:

$$\text{APR}(t) = \frac{\rho(t) - \rho(t - 365)}{365} \cdot 365 = \frac{\rho(t)}{\rho(t - 365)} - 1 \tag{14}$$

For real-time estimates (when 365 epochs of history are not available), the protocol extrapolates from the last 7 epochs:

$$\text{APR}_{\text{est}} = \left(\frac{\rho(t)}{\rho(t - 7)}\right)^{365/7} - 1 \tag{15}$$

### 13.3 Tax Reporting Support

The protocol emits structured events for tax reporting:

1. **Deposit event**: Records amount deposited and sLUX received.
2. **Rebase event**: Records per-share value change at each epoch (for income-based jurisdictions).
3. **Withdrawal event**: Records sLUX burned and LUX received (for capital-gains-based jurisdictions).
4. **wsLUX wrap/unwrap**: Records exchange rate at time of conversion.

### 13.4 Emergency Withdrawal

In case of protocol emergency (critical vulnerability, governance attack), an emergency withdrawal mechanism allows users to exit at a guaranteed minimum rate:

**Algorithm 6** Emergency Withdrawal

**procedure** EMERGENCYWITHDRAW(user)
    **require** $\mathsf{emergencyMode}$ = true (set by governance)
    $\mathsf{amount} \leftarrow \mathsf{shares}_{\text{user}} \cdot \mathsf{totalPooled}/\mathsf{totalShares}$
    $\mathsf{guarantee} \leftarrow \mathsf{shares}_{\text{user}} \cdot 0.95$ ▷ 95% floor
    Transfer $\max(\mathsf{amount}, \mathsf{guarantee})$ to user
    Burn user's sLUX shares
**end procedure**

The 95% floor ensures users can recover at least 95% of their principal even under worst-case conditions (e.g., simultaneous slashing of multiple validators while insurance pool is depleted).

### 13.5 Restaking Integration

sLUX and wsLUX are compatible with restaking protocols that provide additional yield by securing additional services:

**Definition 11** (Restaking)**.** *Restaking allows sLUX holders to opt into additional slashing conditions in exchange for rewards from auxiliary protocols (oracles, bridges, data availability layers):*

$$\textit{Total Yield} = y_{\textit{staking}} + \sum_{j \in \textit{AVS}} y_j - \textit{Risk Premium}_j \tag{16}$$

*where AVS denotes Actively Validated Services.*

Table 11: Projected restaking yield by AVS type.

| AVS Type | Additional Yield | Additional Slash Risk | Net Premium |
|---|---|---|---|
| Oracle network | 1.5% | 2% | +1.0% |
| Cross-chain bridge | 2.0% | 5% | +1.2% |
| Data availability | 1.0% | 1% | +0.8% |

## 14 Governance and Upgradability

### 14.1 Governance Structure

LLS is governed by a multi-sig DAO with the following powers:

- Adding/removing validators from the active set
- Adjusting fee splits (within bounds: total fee $\leq 15\%$)
- Setting the delegation cap and insurance target ratio
- Emergency pause in case of critical vulnerability

All governance actions are subject to a 48-hour timelock, allowing users to exit before changes take effect.

### 14.2 Upgrade Path

Contracts are upgradeable via a transparent proxy pattern with timelock. Major upgrades (changing core logic) require:

1. DAO proposal with 7-day voting period
2. Supermajority (66%) approval
3. 48-hour timelock after approval
4. External audit for changes exceeding 500 LOC

# 15 Conclusion

Lux Liquid Staking achieves the following properties:

1. **Capital efficiency**: 95% efficiency versus 71% for direct staking, unlocking staked capital for DeFi participation.
2. **Security preservation**: Underlying stake remains delegated and slashable regardless of sLUX transfers.
3. **Slashing insurance**: 5% reward fee funds an insurance pool absorbing up to 3.3% losses before affecting holders.
4. **Decentralization**: 10% validator cap and governance-curated validator set prevent stake concentration.
5. **DeFi composability**: Dual token design (sLUX/wsLUX) supports lending, LPing, and collateralization.
6. **Circular prevention**: LTV-bounded leverage limits maximum recursive staking to 4x.

The cross-chain yield composition stack (Section 6.4) extends the liquid staking thesis to non-native assets: BTC holders access a 5-layer yield stack via yLBTC, while xLUX (Section 6.5) aggregates fees from DEX, bridge, lending, perpetuals, and cross-chain activity into a single ecosystem yield token.

Future work includes cross-appchain liquid staking (sLUX representing stake on multiple appchains), restaking integration for shared security, and on-chain governance migration from multi-sig to token-weighted voting.

## 15.1 Governance and Upgrade Path

The liquid staking protocol is governed by a combination of on-chain and off-chain mechanisms during its phased rollout:

**Phase 1 (Current): Multi-sig Governance.** A 4-of-7 multi-sig controls protocol upgrades, validator set changes, and fee parameter adjustments. Multi-sig holders are selected from independent entities to minimize collusion risk.

**Phase 2: Token-Weighted Voting.** sLUX holders gain voting rights proportional to their holdings. Proposals require 5% quorum of total sLUX supply and 66% supermajority. A 48-hour timelock on approved proposals allows users to exit before changes take effect.

**Phase 3: Immutable Core.** The core deposit/withdrawal and exchange rate logic is made immutable. Only peripheral parameters (fee rates, validator caps, insurance thresholds) remain governable. This minimizes the governance attack surface while preserving operational flexibility.

Table 12: Liquid staking protocol comparison.

| Feature | Lux sLUX | Lido stETH | Rocket Pool rETH | Marinade mSOL |
|---|---|---|---|---|
| Token model | Rebasing + wrapped | Rebasing | Non-rebasing | Non-rebasing |
| Validator selection | Governance curated | Governance curated | Permissionless | Score-based |
| Min deposit | 1 LUX | 0.01 ETH | 0.01 ETH | 0.01 SOL |
| Withdrawal time | 14 days | Variable | Variable | 1–2 epochs |
| Insurance fund | 5% of rewards | No | RPL collateral | No |
| Validator cap | 10% max | 1% target | Self-bonded | Score-capped |

The phased governance approach ensures that the protocol progressively decentralizes control while maintaining the ability to respond to security incidents during the early deployment period. Each phase transition requires a successful audit and a minimum operational period of 6 months at the current phase.

### 15.2 Risk Factors

The liquid staking protocol faces several categories of risk:

**Smart Contract Risk.** Bugs in the deposit, withdrawal, or exchange rate contracts could lead to loss of funds. Mitigated by multiple audits, formal verification of the exchange rate invariant, and a bug bounty program.

**Validator Concentration Risk.** If a small number of validators attract the majority of delegated sLUX stake, the effective decentralization of the underlying consensus degrades. The 10% cap per validator bounds this risk.

**Depegging Risk.** Under extreme market stress, sLUX may trade below the exchange rate on secondary markets. The withdrawal mechanism guarantees par redemption at the protocol level, but the 14-day unbonding period creates temporary illiquidity that can amplify depegging in volatile markets.

## References


[1] F. Saleh, "Blockchain without waste: Proof-of-stake," *The Review of Financial Studies*, vol. 34, no. 3, pp. 1156–1190, 2021.

[2] Lido Finance, "Lido: Liquid staking for Ethereum," *Lido Documentation*, 2021.

[3] Rocket Pool, "Rocket Pool: Decentralised Ethereum staking protocol," *Rocket Pool Documentation*, 2021.

[4] Marinade Finance, "Marinade: Liquid staking for Solana," 2021.

[5] T. Chitra and K. Kulkarni, "Improving proof of stake economic security via MEV redistribution," *arXiv:2208.12311*, 2022.

[6] M. Neuder et al., "Low-cost attacks on Ethereum 2.0 by sub-1/3 stakeholders," *arXiv:2102.02247*, 2021.

[7] Gauntlet Networks, "Dynamic analysis of staking derivatives," 2022.

[8] V. Buterin, "Endgame," *Ethereum Blog*, 2021.

[9] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[10] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2014.

[11] A. Kiayias et al., "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *CRYPTO*, 2017.

[12] G. Fanti, L. Kogan, and P. Viswanath, "Economics of proof-of-stake payment systems," 2019.

[13] B. Biais et al., "The blockchain folk theorem," *The Review of Financial Studies*, vol. 32, no. 5, pp. 1662–1715, 2019.

[14] P. Daian et al., "Flash boys 2.0: Frontrunning in decentralized exchanges," in *IEEE S&P*, 2020.

[15] T. Roughgarden, "Transaction fee mechanism design for the Ethereum blockchain," *arXiv:2012.00854*, 2020.

[16] H. Adams et al., "Uniswap v3 core," 2021.

[17] Aave, "Aave Protocol V2," *Aave Documentation*, 2020.

[18] R. Leshner and G. Hayes, "Compound: The money market protocol," *Compound White Paper*, 2019.

[19] S. Kannan et al., "EigenLayer: The restaking collective," 2023.

[20] J. Kwon and E. Buchman, "Cosmos: A network of distributed ledgers," 2016.

[21] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," 2016.

[22] G. Huberman et al., "Monopoly without a monopolist," *The Review of Economic Studies*, vol. 88, no. 6, pp. 3011–3040, 2021.

[23] I. Roşu and F. Saleh, "Evolution of shares in a proof-of-stake cryptocurrency," *Management Science*, vol. 67, no. 2, pp. 661–672, 2021.

[24] K. John, M. O'Hara, and F. Saleh, "Bitcoin and beyond," *Annual Review of Financial Economics*, vol. 14, pp. 95–115, 2022.

[25] L. W. Cong, Y. Li, and N. Wang, "Tokenomics: Dynamic adoption and valuation," *The Review of Financial Studies*, vol. 34, no. 3, pp. 1105–1155, 2021.

[26] M. Sockin and W. Xiong, "Decentralization through tokenization," *The Journal of Finance*, vol. 78, no. 1, pp. 247–299, 2023.

[27] Babylon Labs, "Babylon: Bitcoin staking protocol," *Babylon Documentation*, 2024.

[28] Lombard Finance, "Lombard: Liquid Bitcoin staking," *Lombard Documentation*, 2024.