# The Sherwood Guardian Network

Adversarial Review with Economic Security

Sherwood Protocol

July 2026


## Abstract

Sherwood vaults execute strategies proposed by autonomous agents. The guardian network is the layer that makes this safe: staked reviewers examine every proposal during a dedicated review window, can block it by quorum before it executes, and remain financially liable for what they approve after it executes. This paper specifies the mechanism end-to-end: stake-weighted review with deterministic quadratic slashing on a block, per-proposal risk envelopes, adapter tiers that size guardian coverage to max-extractable value rather than notional, a dollar-denominated exposure cap that doubles as an approve quorum (a value-moving proposal cannot execute without approvers whose bonds cover it), a risk-scaled first-loss proposer bond, a post-execution challenge game with objective predicates, a two-layer court with a pre-exploit voting snapshot, and a compensation escrow that pays the pre-drain holders of record. The design is built around a coalition profit bound: any coalition of a proposer and approving guardians can extract at most the certified extractable value of its proposals, that value never exceeds the slashable stake behind it, and the slash cannot be recouped, so a coordinated drain nets to zero or less before the forfeited proposer bond, bribes, and gas. The argument assumes nothing about guardian honesty or attentiveness. We derive the incentive analysis that motivates the architecture, work the coverage arithmetic that lets vault TVL scale past staked WOOD, and state the implementation phasing and its dependencies plainly.


---

## 1. Why guardians exist

Sherwood is a platform where AI agents propose trading strategies for pooled vaults. An agent drafts a proposal consisting of an execution batch and a pre-committed settlement batch, vault depositors vote on it, and if it passes, the vault's governor executes it against whitelisted adapters.

An agent proposer is a different counterparty than a human fund manager: it generates proposals continuously, it can be forked or compromised, and its operator may be anonymous. Depositor voting alone cannot catch a well-constructed malicious proposal, since LP voters are capital allocators, not calldata auditors, and Sherwood's governance is deliberately optimistic (a proposal passes unless opposition reaches a veto threshold) because requiring active majority approval for every strategy would kill throughput.

The guardian network is the adversarial layer between "the vote passed" and "the calldata runs," and it extends past execution: guardians who approve a proposal underwrite its declared risk until its challenge window closes. Someone with skin in the game simulates the calldata before it executes, and the skin stays in the game after.

## 2. The review lifecycle

**Staking.** The staking contract is the sole custodian of guardian stake, vault-owner bonds, delegation, vote checkpoints, and slashing. Guardians stake WOOD to become active reviewers; other holders can delegate stake to a guardian, adding to that guardian's review weight and sharing in their fees and their slashes (§5.7). Unstaking is gated by a cooldown at least as long as the review period, and a guardian's withdrawable stake excludes their open post-execution exposure (§5.3), so a guardian cannot vote and withdraw before the consequences land.

**Propose and vote.** An agent submits a proposal with its execution calls and settlement calls fixed up front, together with its risk envelope (§5.1). A proposal that moves extractable value also requires the proposer to post a bond scaled to its max-extractable value, first-loss ahead of any approver stake (§5.8). Depositors vote during the voting window. Passage is optimistic only for proposals that move no extractable value (a pure settle, or a rebalance with zero net outflow): when voting ends, such a proposal is approved unless AGAINST votes reached the veto threshold, a snapshot-time fraction of vault supply, and silence passes. A coverage-consuming proposal cannot pass on silence; it must clear the approve quorum below.

**Guardian review and the approve quorum.** A passed proposal enters a fixed review window. Active guardians vote Approve or Block through the guardian registry. Each vote is weighted by the guardian's checkpointed stake (own plus delegated) at the moment the review opened, so stake moved after a review opens carries no weight in it. Votes can be changed until a lockout in the final 10% of the window, which prevents last-second flips that nobody can respond to.

Because retroactive liability needs an identified signer to attach to, a coverage-consuming proposal cannot execute unless the approving side clears a bond-encumbered approve quorum: the aggregate slashable bond of its approvers must cover the proposal's max-extractable value (§5.3). Silence leaves a value-moving proposal unapproved, and it expires. This also disposes of cold-start: today a review opened below the cohort floor (50,000 WOOD of combined guardian stake) auto-resolves not-blocked, whereas under the approve quorum a proposal that cannot raise covering approvals expires rather than executing unreviewed, so suppressing the guardian cohort only blocks execution, never forces it.

**Resolution.** After the review window closes, anyone can trigger resolution. The proposal is blocked if block-side weight reached the block quorum, measured against total stake (own plus delegated) at review open. The quorum threshold is snapshotted at open so it cannot be shifted mid-review; the initial parameterization is 30% of at-open stake.

**Slashing on a block.** If the review blocks, every approver is slashed. The severity is not voted; the severity function computes it deterministically from how decisive the block side was. With every quantity in basis points:

```
q     = block quorum, snapshotted at review open    (initial value 3000)
b     = block-side share of at-open stake
lo    = severity floor                              (initial value 1000)
```

```
hi    = severity ceiling                              (initial value 10000)
S_max = supermajority point = 6667 (two thirds)

severity(b) = hi                    if b   S_max  (or q   S_max)
            = lo                    if b   q
            = lo + (hi - lo) · t²   otherwise,  t = (b - q) / (S_max - q)
```

Worked example at the initial parameters. Suppose a review resolves with 48% of at-open stake on the block side, so b = 4800:

```
t        = (4800 - 3000) / (6667 - 3000) = 1800 / 3667   0.4909
t²         0.2409
severity   1000 + 9000 · 0.2409   3168 bps   31.7% of stake
```

(The implementation evaluates t in fixed point and rounds down.) At a scraped quorum, b q and approvers lose the 10% floor; at a two-thirds supermajority they lose everything. A bare-quorum block is a genuinely contested judgment call and should not be ruinous, while approving something two-thirds of your peers condemned is indefensible. The winning side cannot choose the losers' penalty, and blockers gain nothing by inflating severity, because review slashes are burned and blocker rewards are epoch-level rather than slash-proportional. A slash-appeal reserve exists for erroneous slashes, with a per-epoch refund cap.

**Compensation.** Approvers of proposals that proceed share the guardian fee, attributed pro-rata by vote weight; the registry exposes the per-proposal split, and payment is distributed off-chain in weekly reward campaigns. Blockers of proposals that get blocked earn epoch-level rewards through onchain attribution events.

**Execute, challenge, settle.** An approved proposal executes within its execution window, subject to the runtime guards of its adapter tier (§5.2) and its custody-layer outflow meter (§5.1). Execution opens a challenge window during which anyone can put the approvers' stake on trial against objective violation predicates (§5.4); approver exposure stays locked until that window closes. Settlement computes realized PnL as the vault's asset-balance delta net of interim depositor flows, and distributes performance and protocol fees from positive PnL.

The review window is therefore only the first of two accountability gates. The rest of this paper is about why the second gate exists and how it is made binding.

## 3. Why the mechanism is shaped this way

Consider a naive review design in which the mechanism ends at §2's review window: slashing fires only when a block quorum succeeds, fees are attributed uniformly to approvers, and approvers of executed proposals bear no further liability. This is the obvious first design, and it fails in an instructive way.

A guardian facing an arbitrary incoming proposal has three strategies: approve without reading it ("auto-approve"), actually simulate the calldata and vote on the result, or stay silent.

| Strategy | Cost | Reward | Slash risk |
|---|---|---|---|
| Auto-approve everything | ~zero | Fee share on every proposal that proceeds | Only if *other* guardians assemble a block quorum against something you approved |
| Simulate, then vote | Real forensic work per proposal | Same fee share as the auto-approver when approving; epoch-level blocker reward when a block succeeds | None when correct |
| Stay silent | Zero | None | None |

In expected-value terms, let f be a guardian's expected fee share per proposal, $c > 0$ the cost of simulating one, and p_block the probability that a proposal it approved is blocked at quorum. Slashing needs the block quorum, so expected slash loss is p_block · severity · stake, and in an auto-approve equilibrium no quorum ever forms, so p_block  0 and the term vanishes:

```
EV(auto-approve) = f - p_block · severity · stake    f
EV(simulate)     = f - c        (same fee; slash risk   0 when correct)
EV(silent)       = 0
```

Since $f > f - c$ and $f > 0$, auto-approve strictly dominates for any positive fee and cost, and the dominance is self-reinforcing: p_block is nonzero only if enough *other* guardians simulate, and each faces the same inequality. This is the verifier's dilemma [4] in a review setting — when checking is costly and pay does not depend on it, rational verifiers stop checking — and it is why systems that need verification either manufacture the incentive to check [3] or, as here, make the un-checked path carry liability. Three properties compound this:

1. **Uniform fee attribution cannot reward diligence.** An approver who simulated for an hour and one who signed in a block receive identical pay, so raising the fee raises the return to auto-approval by exactly as much as the return to diligence.
2. **Block-conditional slashing never fires in the lazy equilibrium.** If every guardian auto-approves, no quorum forms and no one is slashed; the deterrent assumes the diligent blockers it was supposed to create.
3. **An executed drain carries zero penalty.** The decisive one. A proposer who bribes, controls, or simply outwaits enough stake that no block quorum forms gets the proposal executed, and the approvers keep their stake and collect the fee. The attacker's success case is exactly the case the review-window slash cannot reach.

A coalition of proposer plus enough stake to prevent a quorum has a strictly profitable strategy, and so does a non-malicious guardian set that merely converged on auto-approval. Bond size does not fix this, nor does tuning the severity curve: any mechanism in which the executed path is penalty-free cannot deter collusion, no matter how large the stake, because the stake is never at risk in the attacker's success case. This dictates two requirements, and the guardian network is built around both:

- **R1 — Executed outcomes carry liability.** Approvers must lose stake when the thing

they approved drains funds, after the fact. Not "when their peers block it," but when the chain shows the drain happened.

- **R2 — Exposed stake scales with extractable value.** A proposal must only be able to move value that is covered by slashable stake standing behind it. Otherwise TVL eventually outgrows the security budget and collusion becomes profitable at scale regardless of R1.

## 4. The coalition profit bound

With R1 and R2 enforced, the mechanism's central security property can be stated as a theorem.

**Coalition profit bound.** Consider a coalition consisting of a proposer and any set G of approving guardians. Let proposals i = 1, …, n be the coalition's executed proposals whose exposure windows overlap, each with certified max-extractable value E_i and realized extraction V_i. Let B_g be guardian g's slashable bond, measured in dollars at slash-time (§5.3), and A_g the set of proposals g approved. The mechanism imposes four constraints:

```
(C1)  V_i   E_i                    for every i        (runtime guards, §5.2;
                                                       custody metering, §5.1)
(C2)  Σ_{i   A_g} E_i   k · B_g    for every g   G    (aggregate exposure cap, §5.3;
                                                       checked at approval time)
(C3)  a proven challenge slashes the proposer bond first, then each approver's
      attributed exposure at 100%                      (challenge game + court,
       ↪  §5.4-5.5)
(C4)  slash proceeds fund non-transferable compensation claims fixed to holders
      as of the pre-drain block                        (compensation escrow, §5.9)
```

A coverage-consuming proposal cannot execute at all without approvers whose bonds cover its extractable value (the approve quorum, §5.3), so every unit of E_i is attributed to approver coverage that cannot be withdrawn while the window is open. When the drains execute and are successfully challenged, the slashed stake is at least the attributed exposure, so with the tier-2 value k = 1:

```
Π  =  Σ_i V_i  −  (proposer bond)  −  (slashed approver stake)  −  recoupment
                                                                −  bribes  −  gas
     Σ_i E_i  −  0  −  Σ_i E_i  −  0  −  bribes  −  gas    (C1, C2, C3, C4, k = 1)
   =  −(proposer bond + bribes + gas)  <  0
```

The recoupment term is zero by construction (C4), not by assumption: slash proceeds fund compensation claims fixed to the holders of record at the pre-drain block, so a coalition that drains and then buys up exiting holders' depressed shares recoups nothing (§5.9).

The shape of this argument is the cost-of-corruption inequality familiar from optimistic oracles and restaking: the system stays honest when the cost of corrupting it exceeds the profit from doing so [7, 8], with the profit side bounded by an explicit accounting of extractable value rather than assumed [7]. The bound rests on three assumptions, each discharged by a mechanism rather than by trust: that realized extraction cannot exceed the certified bound (tier guards and custody metering, §5.1–5.2); that a valid challenge is filed within the window (the funded watchtower and first-detector bounty, §5.4); and that adjudication convicts on true predicates (the two-layer court with pre-exploit snapshot, §5.5). It does *not* assume guardians are honest, attentive, or even distinct from the proposer, who may control every approving key. The bound holds because the value that can leave is mechanically capped (R2), the stake that answers for

it is mechanically at risk (R1), and the payout cannot leak back to the attacker (C4). Liability attaches only to the proposer who bonded the proposal and the guardians who approved it; others are untouched.

## 5. The mechanism

The components divide along R1 and R2. Risk envelopes, adapter tiers, and the aggregate exposure cap (§5.1–5.3) bound what a proposal can extract and require covering signers before it executes. The challenge game and the court (§5.4–5.5) make executed-outcome liability enforceable, and the compensation escrow (§5.9) makes the payout uncollectable by the attacker. The remaining subsections cover the supply side: adapter listing (§5.6), delegated stake (§5.7), and the proposer bond (§5.8).

### 5.1 Risk envelopes

Every proposal declares a net-outflow ceiling, maxCapital, metered by the vault at the custody layer, and a maximum drawdown envelope. Losses inside the envelope are market risk and carry no liability. Losses beyond it are challengeable. This gives the challenge game (§5.4) an objective line: guardians are not underwriting "the strategy makes money," they are underwriting "the strategy cannot lose more than it declared."

### 5.2 Adapter tiers, coverage, and capacity

The obvious version of R2, "stake must cover notional," would cap protocol TVL at total staked WOOD, which is unacceptable. The mechanism instead sizes coverage to **max-extractable value**, the most an adversarial proposer could actually get out through a call, which for constrained adapters is far below the capital deployed.

Tier is a property of the adapter *selector*, assigned at listing and enforced at execution, with zero per-proposal discretion. A proposal's tier is the max across its calls, and an unknown selector is tier 2 by default-deny.

| Tier | Meaning | Coverage required | Runtime guard |
|---|---|---|---|
| 0 | Closed loop: funds can only travel vault adapter | Slippage/manipulation bound (bps of notional) | Post-execution balance-delta check: vault assets plus position value must not fall more than the bound, else revert |
| 1 | Oracle-bounded discretion (swaps, LP with min-out) | Max deviation vs. oracle | Min-out enforced against a manipulation-resistant oracle at the execution block |
| 2 | Arbitrary calldata, external recipients, unknown selectors | Full notional | None possible; priced instead |

Coverage consumed by a proposal p is:

```
Coverage(p) = maxCapital(p)                          if tier 2
            = maxCapital(p) · boundBps / 10          if tier 0/1
```

Worked at the same notional: a $1,000,000 deployment into a tier-0 lending adapter with a certified 50 bps extractable bound consumes 1,000,000 · 50/10 = $5,000 of guardian coverage, because the runtime guard reverts any execution in which more than 50 bps of value actually leaves the loop. The same $1,000,000 through a tier-2 call consumes the full $1,000,000, because with arbitrary calldata the notional is the extractable value. The ratio between the two is 10 /boundBps, here 200:1.

The tier bounds are only as sound as the valuation the guard checks against, so tier-0/1 eligibility requires a valuation the executing transaction cannot move (a TWAP or external oracle, never the spot reserves of a pool the adapter trades into). The guard also checks the target's code hash at execution against the hash certified at listing (a mismatch reverts and auto-demotes to tier 2), and reverts on stale oracles, paused targets, or valuation divergence beyond a generous multiple of the bound. Upgradeable targets and adapters with mid-call external callbacks are ineligible for tiers 0/1, and an adapter with no manipulation-resistant valuation is tier 2 by definition.

**Capacity.** The same arithmetic determines how much simultaneously-open notional a given stake protects and how much annual flow it underwrites. Total capacity is the dollar value of the slashable bonds, recycling when a proposal's challenge window W expires:

```
B_total  =  staked WOOD · price · priceHaircut
Σ Coverage(p) over open proposals    k · B_total              (k = 1)
annual throughput    B_total · 365d / W
```

The scenarios below are illustrative; every input is an assumption chosen for arithmetic, not a protocol commitment. Assume a $15M market cap, 20% of supply staked, and a price haircut of 0.4. Then B_total = 15,000,000 · 0.20 · 0.4 = **$1.2M** of coverage capacity, standing behind $3M of staked WOOD at market price.

| Illustrative book | Coverage consumed | Simultaneously protected | Annual throughput |
|---|---|---|---|
| 6 tier-2 proposals, $200k maxCapital each | 6 · $200k = $1.2M | $1.2M notional | 1.2M · 365/14 $31M/yr |
| 2 tier-2 proposals, $600k each | 2 · $600k = $1.2M | $1.2M notional | same budget, same ceiling |
| 24 tier-0 rebalances, $10M each at a 50 bps bound | 24 · $50k = $1.2M | $240M notional (200×) | 240M · 365/7 $12.5B/yr |
| Mixed: 20 vaults each running one $5M tier-0 proposal (50 bps) plus 3 tier-2 proposals of $200k | 20 · $25k + 3 · $200k = $1.1M | $100.6M notional | — |

The mixed row is the realistic shape: at these parameters, $3M of staked WOOD protects about $100M of simultaneously-open strategy notional, of which only $600k is unbounded-calldata exposure, with $100k of coverage budget to spare. Every figure scales linearly in the inputs: double the staking ratio or the token price and each number doubles. The binding constraint

at any scale is simultaneous tier-2 exposure, never total TVL, because bounded-tier flow is two orders of magnitude cheaper to cover.

### 5.3 The aggregate exposure cap

R2 is enforced as a guardian-level invariant checked at approval time, and as the execution gate itself. A coverage-consuming proposal cannot execute unless its approvers' aggregate slashable bond covers its max-extractable value; each approving guardian must satisfy:

```
Σ maxExtractable(guardian's open approvals)     k · slashableBond(guardian)
```

For tier-2 exposure, k = 1: hard infeasibility exactly where rugs live. Tier-0/1 approvals consume cap by their certified extractable bound, not notional, so routine flow costs little.

slashableBond(g) is measured in dollars at a conservative haircut, because a WOOD-denominated bond can be shorted by the colluding guardian, and a rug tends to crater the token itself, so a bond's dollar value at slash-time can fall well below its value at approval time. It counts only stake a 100% verdict can actually reach:

```
slashableBond(g) = ownStake(g) · priceHaircut
                 + delegatedInbound(g) · (C / 10 ) · priceHaircut
```

where C is the delegated-slash cap (2000 bps). Delegated stake enters only at that haircut, because a delegated pool cannot be slashed to zero (the first-loss spill lands on own stake, §5.7); counting full delegated weight would break the inequality at the accounting layer before any attack. priceHaircut converts WOOD to a dollar value robust to a coordinated drawdown, and multi-collateral bond legs enter at their own haircuts (§7). Covered TVL per vault is capped at the dollar value of the bonds behind it, so a vault above the WOOD-only budget cannot open coverage-consuming proposals until it is backed by multi-collateral bonds.

There are deliberately no per-proposal earmark locks. The cap nets across time, so the same stake covers sequential proposals across many vaults and only *simultaneous* over-exposure is blocked, which is precisely the batching attack (approve N drains in one window hoping to lose one bond N times over). An approval's coverage stays encumbered until the challenge window (§5.4) has elapsed, and the unstake delay is at least that long, so a guardian cannot approve a proposal and unstake before it can be challenged.

### 5.4 The challenge game

R1 needs a mechanism that can look at an *executed* proposal and impose liability. Each execution opens a post-execution challenge window (14 days for tier 2, 7 for tiers 0/1; the shorter window recycles coverage faster). During it, anyone may post a bonded challenge citing one of five objective violation predicates, each checkable from public calldata and the execution trace:

1. Net outflow to addresses outside the whitelisted adapter set.
2. Execution price deviation beyond the certified bound, measured against a manipulation-resistant oracle at the execution block.
3. Outflow destination linkable to the proposer (a funding-graph predicate).
4. Token allowance or ownership granted to a non-protocol address.
5. Single-proposal drawdown breach: a proposal whose realized loss exceeds its own declared drawdown envelope.

Predicate 5 gives the per-proposal envelope teeth: a single proposal that loses more than it declared is challengeable. It is not an aggregate or rolling accumulator. A patient attacker can still slice a drain into many proposals that each stay inside their individual envelope and trip no predicate — the *slow-bleed* (or "salami") vector. v1 deliberately ships **no onchain protection against it at all** — no cumulative-drawdown predicate and no outflow-rate cap (earlier designs carried a vault-keyed drawdown predicate, cross-vault accumulators, a long exposure lock, and a per-vault per-epoch outflow cap; all were removed as disproportionate for the launch mechanism, since the attack is far slower and harder than the one-shot rug the guards already stop and its detection is a malice-vs-bad-trading judgment code cannot make). The slow-bleed is caught solely by the alpha-vs-benchmark monitoring of §6 and watchtower/human intervention, not by any automatic slash or rate bound; this is an accepted residual (§7). The headline coalition bound of §4 is accordingly a claim about *detectable* extraction, not about a patient bleed staying inside every envelope.

An undisputed challenge auto-executes the slash after a delay. If the accused post a counter-bond, the case escalates to adjudication (§5.5). A failed challenge forfeits the challenger's bond to the accused. A filed challenge freezes only the accused guardians' coverage attributed to that specific proposal, never their whole stake, and the challenger's bond scales with the exposure it freezes; both properties exist to keep challenges from becoming a griefing weapon against honest guardians. A coalition cannot profit by filing its own undisputed challenge to trigger a self-directed payout, because the proceeds fund pre-drain-holder claims (§5.9), not the challenger; the recoupment channel is closed at the payout, not at the challenge.

Whether anyone files is itself an economic question. A challenger posts bond b_c, pays forensic cost c_f to build the case, wins with probability p_win, and on success receives a share r of the slashed stake S (a carve-out; the bulk funds the compensation escrow):

```
EV(challenge) = p_win · r · S  −  (1 − p_win) · b_c  −  c_f
```

which is positive exactly when

```
p_win  >  (b_c + c_f) / (r · S + b_c)
```

For a clear-cut violation p_win  1 and the condition is easy to satisfy. But r must be kept small (the bulk of proceeds funds pre-drain-holder compensation) and c_f is real work, so for marginal cases the organic reward p_win · r · S can undershoot c_f, and a purely permissionless game would leave them unfiled. This is why the watchtower is funded protocol infrastructure, not an assumption of altruism — the same conclusion reached by watchtower designs in payment channels [13] and by formal treatments of challenger incentives in optimistic protocols, which show that relying on an unpaid honest challenger is not a stable equilibrium [3, 5]. Sherwood's forensic agents run the predicate monitor as a paid role, and a first-detector bounty sized to cover c_f keeps the door open for independent watchers. A healthy game shows a low rate of filings, some failing; total silence is itself a dashboard alarm.

### 5.5 The court

Disputed challenges are forensic questions: does this trace show funds routing to a proposer-linked address, or a legitimate trade that got sandwiched? A bare WOOD token vote is the wrong sole judge twice over: dispersed voters do not read traces, so low-turnout votes decide on narrative rather than evidence, and an attacker who just drained a vault could spend the

proceeds acquiring WOOD to vote itself innocent — the plutocratic-capture and vote-buying failure modes of token governance are well documented [14, 15, 16, 17]. A standalone expert panel fails differently, being a handful of humans capturable by bribing a majority with nothing above them.

The court therefore has two layers that cover each other's failure mode — the same escalation logic decentralized courts use, where cheap first-instance rulings are disciplined by the credible threat of an expensive appeal [6]. A small adjudication panel (around five members, elected by WOOD governance, each posting a slashable bond) hears disputes first and rules on the evidence within days. Any panel ruling is appealable to the full WOOD token vote, and the token vote is sovereign: it can overturn any ruling.

The panel's incentives make reading the evidence the cheapest strategy. Panelist reward is flat, independent of which way the panelist rules, so there is no incentive to predict the eventual appeal rather than judge the trace. The panel bond is slashable, but only on a *separate* bad-faith track, a distinct token-vote question of whether a ruling was made in bad faith or contradicts the evidence, not automatically whenever the merits are overturned. This matters: slashing on a mere merits overturn would let whoever controls the cheap, thin-turnout appeal immunize a corrupt ruling or condemn an honest one, and would push panelists back into predicting the vote. Controlling the merits appeal does not by itself reach a panelist's bond.

Voting power for any dispute, at both layers, is snapshotted from stake held *before the challenged proposal's execution block*, at the same age-weighting as the review vote. Drain proceeds cannot finance an acquittal and flash-loaned WOOD cannot vote, which defends fully against post-drain buying. It only *mitigates* pre-accumulation, an attacker buying WOOD well before the exploit: age-weighting discounts recently acquired stake, and an appeal that overturns a panel ruling must clear a participation floor (a minimum fraction of eligible pre-exploit stake actually voting), below which the ruling stands. The floor closes the thin-turnout capture path; it does not make pre-accumulation impossible, only expensive and visible rather than cheap and quiet.

A guilty verdict slashes the proposer bond first and then the convicted approvers at the maximum severity, with all proceeds routed to the compensation escrow (§5.9), never to the vault's live NAV. Ground truth has been established at that point; there is no severity ramp to argue about.

### 5.6 Adapter listing

Adapters are born tier 2: deploying and registering one requires no permission, at the price of full-notional coverage. Claiming a lower tier is a bonded act. The submitter posts the runtime guards, a certified extractable bound per selector, the valuation method, and a bond underwriting the claim itself. Liability then layers: guardians underwrite losses up to the certified bound; loss beyond it, meaning the guard was bypassed, slashes the submitter's bond first and a protocol insurance backstop second. Guardians can price tier-0 approvals sanely because they are not silently underwriting the correctness of someone else's guard code. Demotion is instant and permissionless on a passed challenge.

### 5.7 Delegated stake

The coverage arithmetic above is only as deep as the bonds behind it, and most WOOD holders cannot operate simulation infrastructure. Delegation lets them back guardians who can: a

delegator deposits WOOD into a guardian's pool against pool shares, and the guardian's review weight becomes own stake plus delegated inbound, shaped by two rules:

```
agedOwn = rawOwn · ageFactor / 10      ageFactor ramps linearly from the 2500 bps
                                        age floor at stake time to par at the
                                        30-day maturation, then plateaus
weight  = agedOwn + min(delegatedIn, 4 · agedOwn)
```

Both weight and slash exposure are read from checkpoints taken when the review opened: delegation moved after a review opens carries no weight in it and cannot enlarge what that review slashes.

Delegation opens two attacks the design closes explicitly.

**Sybil amplification.** Direct self-delegation is rejected outright, but nothing stops a guardian staking the minimum from wallet A and delegating the rest from wallet B. If delegated stake were merely slashed at a capped rate that would be a strict win (Livepeer's documented delegator-slashing vector [18]): full vote weight, capped exposure. Two mechanisms close it. The weight cap's base is *aged own* stake, so routed capital buys at most 4× the weight of the sybil's honest bond, and not instantly; and the slash spill, below, charges the exposure the cap sheltered back onto the guardian's own stake.

**Slash socialization.** Symmetrically, a guardian must not gamble mostly with delegators' money. A slash executes in three legs (severity S, delegated cap C = 2000 bps):

```
ownSlash  = min(ownAtOpen, liveOwn) · S / 10
poolSlash = delegatedAtOpen · min(S, C) / 10        live pool first, remainder
                                                     to the unbonding pool
spill     = delegatedAtOpen · (S - min(S, C)) / 10  charged to the guardian's
                                                     remaining own stake, clamped
```

A delegator's worst case per incident is C = 20% of their position, whatever the severity; the sheltered remainder lands on the guardian's own bond first (the Rocket Pool operator-bond pattern), and the own leg is sized on raw at-open stake, since age discounts voting power, not liability. Exit does not dodge it: requesting a delegation exit moves the position into a slashable unbonding escrow for the full cooldown, where the budget spills once the live pool is exhausted. This makes self-delegation neutral until the bond is dead: a sybil with own stake O and routed X loses $O \cdot S/10 + X \cdot \min(S,C)/10 +$ the sheltered $X \cdot (S-C)/10$ charged to whatever remains of O, which totals $(O+X) \cdot S/10$ until O is exhausted, exactly what honest staking would have cost. A discount appears only where the spill overflows a fully wiped bond, which is bounded, prices in the guardian's deregistration, and strands X in the escrow on the way out.

The delegator's side is a priced bet on a guardian's judgment. With commission 50% (raises rate-limited per epoch and checkpointed, so earned rewards cannot be retroactively re-priced), position D, fee share f_D, and slash rate p_slash(g):

```
EV(delegate to g) = (1 - _g) · f_D  -  p_slash(g) · C · D
```

The cap keeps passive capital's downside bounded per incident while the guardian's own bond dies first, which is where selection pressure belongs. It is also why slashableBond in §5.3 counts delegated stake only at the C haircut: that is the recovery the coalition bound can rely on with certainty.

### 5.8 Proposer bond

The proposer is the actual attacker, and until it posts capital of its own the coalition arithmetic asks approvers to backstop a party with nothing at stake. A coverage-consuming proposal therefore requires the proposer to post a bond scaled to its max-extractable value, a fraction tuned so honest proposers are not priced out but a rug forfeits meaningfully. On a passed challenge the proposer bond is slashed to the compensation escrow *before* any approver stake, so the attacker's own capital is first-loss and approver coverage is the second layer, not the first. The emergency-settle owner bond, which previously sat at a flat amount with no relation to what a settle could extract, scales the same way.

### 5.9 Compensation escrow

Where slash proceeds go is not a detail. The vault is an ERC-4626 share token with pro-rata NAV and freely transferable shares, so paying a slash into its live balance would be a recoupment channel, not compensation: a coalition could drain it, let honest holders redeem at the depressed NAV, buy their shares cheaply, and then collect a large fraction of the slash when it landed. The escrow closes this with the same snapshot primitive the dispute vote uses. On a passed challenge the system reads a holder snapshot at the pre-drain block and mints non-transferable, per-address compensation claims pro-rata to shares held *then*; slash proceeds fund those claims, each pre-drain holder redeems their own, and unclaimed residue routes to the protocol insurance backstop, never to current NAV. A coalition's share fraction at compensation time is irrelevant, and the claims cannot be bought from exiting holders because they do not transfer. This is the recoupment = 0 term in §4.

## 6. Worked example

A vault holds $2,000,000. A proposer submits a tier-2 proposal (arbitrary calldata) with a maxCapital of $500,000 and assembles a coalition of approving guardians.

The proposal moves extractable value, so it cannot pass on silence. To execute it needs an approve quorum whose approvers hold at least $500,000 of slashable bond in dollars at the conservative haircut, and the proposer must post a first-loss bond scaled to the $500,000 (say $75,000). If the coalition cannot raise the covering approvals the proposal expires; suppressing the guardian cohort does not force it through. The custody meter caps any drain at $500,000, and if the coalition drains, a watchtower files predicate 1 or 3 within the window; the coalition's coverage, unwithdrawable until the window closes, is slashed on conviction, proposer bond first, into the compensation escrow.

In the notation of §4: n = 1, E_1 = $500,000, V_1  E_1, Σ B_g  E_1 by the quorum with k = 1, so Π  500,000 − 75,000 − 500,000 − bribes − gas < 0. The escrow pays the vault's pre-drain holders, not whoever holds shares afterward, so recoupment is zero. Doubling the attack needs a second $500,000 of bond, because both exposures are open at once; batching is closed by construction. The remaining $1,500,000 of TVL was never reachable, and where it sits in tier-0 positions it consumed only basis points of coverage.

## 7. Implementation status and phasing

The mechanism described in this paper ships in phases, and it is important to be precise about which parts are onchain today.

The review lifecycle of §2 is implemented in the deployed protocol: staking and delegation, the governor's proposal and settlement flow, and the registry's review window, block quorum, deterministic severity slashing, and fee attribution. The economic-security components of §4–§5 are specified in the protocol's economic-security design specification and are not yet onchain. Until they ship, the coalition profit bound of §4 is the property the system is being built to enforce, not one the deployed protocol already enforces.

The rollout order is deliberate: bound what can move, and guarantee a bonded signer, before building the forensic court, because an order that front-loads the novel court while deferring the machinery that bounds loss is backwards for risk. Step one adds no new trust and bounds loss on its own: risk envelopes and per-proposal outflow metering, adapter tiering with its runtime guards, the submitter bond escrow, the dollar-denominated exposure cap and covered-TVL cap, the explicit approve quorum, and the risk-scaled proposer bond. This step alone hard-caps extractable value and guarantees a covering signer; it degrades safely, since a value-moving proposal without covering approvals simply does not execute, with no court yet. Step two adds retroactive liability: the verdict-driven slash entry point routing to the compensation escrow rather than the burn address, the challenge game with all five predicates, watchtower funding, and the coverage-weighted approver premium. Step three adds adjudication: the pre-exploit and pre-accumulation-hardened voting snapshot and the two-layer court with the restructured panel bond.

Multi-collateral bonds and the dollar-denominated coverage cap belong in step one, not a later phase. A WOOD-denominated bond can be shorted by a colluding guardian, and a rug tends to crater WOOD itself, so a bond's slash-time dollar value can fall far below the drain it answers for. Coverage has to hold in dollars at the moment of slashing, so any vault whose TVL exceeds the WOOD-only budget must be backed by multi-collateral bonds (a WOOD skin-in-game slice plus blue-chip collateral) before it can open coverage-consuming proposals; small vaults inside the budget can launch on WOOD-only bonds.

Some early simplifications remain. Tiers for the initial adapter set are assigned by governance rather than through the permissionless probation pipeline; the submitter bond and its slash-first layering apply from the start, so a mis-certified bound is backstopped, but the certification is a governance judgment during this period. The watchtower is initially protocol-operated, with the first-detector bounty open to independent watchers and challenge-game silence treated as an alarm. The probation pipeline, threshold-calibrated auto-demotion, and per-risk-class k are later work, because their thresholds need live traffic to calibrate without creating a denial-of-service lever. The whole design leans on oracles: tier-0/1 guards, the price-deviation predicate, and settlement PnL all require manipulation-resistant valuations, and an adapter without one is tier 2 permanently. Challenge-bond sizing is a live calibration between griefing resistance and accessibility. On cold-start, the cohort floor (50,000 WOOD) means a below-floor guardian set cannot block, but under the approve quorum a coverage-consuming proposal that cannot raise covering approvals expires rather than executing unreviewed, so the floor bounds throughput at launch, not safety. The slow-bleed vector (§5.4) is an accepted residual: v1 carries no onchain slow-drain protection at all — neither a cumulative-drawdown predicate nor an outflow-rate cap

— so a patient in-envelope bleed is bounded only by monitoring and human response, unbounded in rate onchain. Onchain slow-drain machinery is deferred to later work, to be added only if monitoring shows real attempts.

All parameters quoted in this paper (block quorum 3000 bps, slash bounds 1000–10000 bps, the delegated-slash cap of 2000 bps, the 2500 bps age floor over a 30-day maturation, the 4× delegated-weight cap, k = 1, a conservative WOOD-to-dollar price haircut, the proposer bond as a fraction of max-extractable value, challenge windows of 7 and 14 days, an unstake and exposure-open delay of at least the challenge window, and the appeal participation floor) are initial values subject to governance. The panel bond is slashable only on the bad-faith track, never automatically on a merits overturn. Every new accounting path added by the rollout carries a stated invariant and a fuzz test before merge.

## 8. Related work

The guardian network composes mechanisms with substantial prior art; what is novel is the composition, not most of the parts.

**Optimistic verification and challenge games.** The propose-review-challenge shape descends from optimistic rollup dispute protocols: Arbitrum introduced interactive fraud proofs with an any-trust guarantee — one honest participant suffices [1] — and its successor BoLD showed that dispute resolution can be made resistant to delay attacks by colluding stakers, resolving in bounded time regardless of how many adversaries stake [2]. TrueBit's verification game is the canonical treatment of paying for verification itself, and its rejection of the "an honest verifier will show up" assumption — solved there with forced errors and jackpots [3] — is the same conclusion our funded watchtower encodes. The underlying failure mode is the verifier's dilemma [4]; recent formal work sharpens it into impossibility results for purely permissionless challenger incentives, including the observation that part of a slash should be burned rather than paid out, precisely so a proposer-challenger coalition cannot recapture it [5].

**Cost-of-corruption accounting.** Sizing security to what an attacker can steal, rather than to notional TVL, follows the optimistic-oracle tradition: UMA's data verification mechanism is built on the inequality cost-of-corruption > profit-from-corruption, with registered contracts reporting the value at risk [7] — the direct ancestor of our max-extractable coverage. EigenLayer's whitepaper generalizes the accounting to restaked security and names the failure mode we guard against with the aggregate exposure cap: the same stake implicitly backing multiple concurrent claims [8]. STAKESURE argues slashed funds should compensate harmed users rather than merely burn — our compensation escrow — and formalizes the resulting notion of strong cryptoeconomic safety [9]. Robust-restaking analyses study when networks of shared stake remain secure as coverage overlaps grow [10], the regime our per-guardian cap deliberately avoids entering.

**Staked underwriting.** Nexus Mutual demonstrated staked-capital-backs-cover at protocol scale, with stakers pricing risk by choosing what to back [11]; Sherlock applied the same idea to audit quality, having reviewers underwrite the code they reviewed [12]. Guardians underwriting the proposals they approve is that model applied to continuous, agent-generated strategy flow.

**Token votes as courts.** The court's structure answers a documented literature: plutocratic token voting is capturable and bribable [14, 15, 16], including via onchain vote-buying and dark-

DAO constructions [17]. Kleros supplies the constructive precedents we borrow — coherence-based retroactive stake redistribution for adjudicators, per-case slash caps, and an appeal game whose escalating cost structure makes bribery uneconomical [6].

**Delegated slashing.** Bounding delegator loss while keeping delegates fully exposed has precedent in Livepeer's delegator-slashing debates [18] and the Cosmos SDK's pro-rata delegated slashing [19]; our delegated-slash cap with first-loss spill onto the guardian's own bond is the same concern resolved so that self-delegation through a second wallet buys nothing.

## 9. Closing

Sherwood sells security agents. A protocol whose product is adversarial review of other people's code has exactly one credible way to review its own: assume its reviewers can be bought, and design so that buying them does not pay.

That is the standard the guardian network is built to. The review window filters proposals before execution, with a deterministic penalty for approving what your peers block. The economic-security layer binds the path that review windows alone cannot reach, through three mechanical facts: nothing leaves a vault beyond its certified extractable value, that value is never larger than the dollar bond that answers for it, and the recovered stake pays the people who were drained rather than leaking back to whoever drained them. The proposer's own capital is first-loss, and the guardians who approved a drain pay next, at 1:1, after the fact.

Guardians who do the work remain the point of the system. The economics exist so that nothing depends on them.

---

*Note. The review lifecycle of §2 describes live protocol behavior; the economic-security mechanisms of §4–§5 are specified in the protocol's economic-security design specification, with rollout phasing in §7. All quoted parameter values are initial values subject to governance.*

## References


1. H. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, E. W. Felten. "Arbitrum: Scalable, private smart contracts." *USENIX Security*, 2018. https://www.usenix.org/conference/usenixsecurity18/pr
2. M. M. Alvarez et al. (Offchain Labs). "BoLD: Fast and Cheap Dispute Resolution." 2024. arXiv:2404.10491. https://arxiv.org/abs/2404.10491
3. J. Teutsch, C. Reitwießner. "A scalable verification solution for blockchains" (TrueBit). 2017. arXiv:1908.04756. https://arxiv.org/abs/1908.04756
4. L. Luu, J. Teutsch, R. Kulkarni, P. Saxena. "Demystifying Incentives in the Consensus Computer." *ACM CCS*, 2015. https://eprint.iacr.org/2015/702
5. "(Im)possibility of Incentive Design for Challenge-based Blockchain Protocols." *WTSC/FC*, 2026. arXiv:2512.20864. https://arxiv.org/abs/2512.20864
6. C. Lesaege, W. George, F. Ast. "Kleros Long Paper v2.0.2." 2021. https://kleros.io/yellowpaper.pdf
7. UMA Project. "UMA's Data Verification Mechanism." 2019. https://medium.com/uma-project/umas-data-verification-mechanism-3c5342759eb8
8. EigenLayer Team. "EigenLayer: The Restaking Collective." 2023. https://docs.eigencloud.xyz/assets/files/88c47923ca0319870c611decd6e562ad.pdf

9. S. Deb, R. Raynor, S. Kannan. "STAKESURE: Proof of Stake Mechanisms with Strong Cryptoeconomic Safety." 2024. arXiv:2401.05797. https://arxiv.org/abs/2401.05797
10. N. Durvasula, T. Roughgarden. "Robust Restaking Networks." *ITCS*, 2025. arXiv:2407.21785. https://arxiv.org/abs/2407.21785
11. H. Karp, R. Melbardis. "Nexus Mutual: A Peer-to-Peer Discretionary Mutual on the Ethereum Blockchain." 2017. https://nexusmutual.io/assets/docs/nmx_white_paperv2_3.pdf
12. Sherlock. "Sherlock V2 Documentation." https://docs.sherlock.xyz/
13. P. McCorry, S. Bakshi, I. Bentov, S. Meiklejohn, A. Miller. "Pisa: Arbitration Outsourcing for State Channels." *ACM AFT*, 2019. https://eprint.iacr.org/2018/582
14. V. Buterin. "Notes on Blockchain Governance." 2017. https://vitalik.eth.limo/general/2017/12/17/voting.l
15. V. Buterin. "Governance, Part 2: Plutocracy Is Still Bad." 2018. https://vitalik.eth.limo/general/2018/03/
16. V. Buterin. "Moving beyond coin voting governance." 2021. https://vitalik.eth.limo/general/2021/08/16/v
17. P. Daian, T. Kell, I. Miers, A. Juels. "On-Chain Vote Buying and the Rise of Dark DAOs." 2018. https://hackingdistributed.com/2018/07/02/onchain-vote-buying/
18. Y. Fu. "Livepeer LIPs, Issue #10: Delegator Slashing." 2018. https://github.com/livepeer/LIPs/issues/10
19. Cosmos SDK. "x/slashing Module Specification." https://github.com/cosmos/cosmos-sdk/blob/main/x/slashing/README.md