LayerCover

Protocol Whitepaper

layercover.com

Abstract

LayerCover is a fully on-chain, parametric cover protocol inspired by the Lloyd's of London marketplace. It enables capital providers (underwriters) to deploy single-sided liquidity across multiple independent risk pools, earning premiums while maintaining full control over capital allocation and risk exposure. Cover is underwritten and settled entirely by smart contracts. Payouts are triggered deterministically by policyholders through a Hybrid Custody model: utilizing either a direct swap of distressed assets or, in the event of frozen protocols, an atomic escrow assignment. This ensures claims can be settled instantly without governance votes or subjective assessments, even if the underlying DeFi protocol has paused transfers. By combining modular risk assessment, tranched reinsurance protection, and salvage rights on payout, LayerCover delivers a capital-efficient alternative to mutualised risk pool designs, offering predictable and fully auditable protection for decentralised finance.

November 24, 2025

Contents

1 Introduction

The global insurance industry has, for centuries, operated as a central marketplace where risk is underwritten, capital is pooled, and claims are settled according to mutually agreed terms. Lloyd's of London, founded in 1688, remains the archetype of such a marketplace where a network of syndicates and underwriters, each providing capital to cover specific risks, coordinated through a central governance framework. In decentralised finance (DeFi), the need for an equivalent mechanism is both clear and urgent. Protocol hacks, smart contract failures, oracle malfunctions, governance exploits, and other tail events have caused billions of dollars in losses in recent years. Unlike traditional finance, DeFi has no central clearinghouse or government-backed safety net. When failures occur, users are typically left with few options: absorb the loss, hope for a socialised bailout, or pursue unenforceable legal remedies.

1.1 Why Now

Decentralised finance (DeFi) has matured from experimental protocols to an ecosystem securing hundreds of billions of dollars in total value locked (TVL). Institutional investors, professional market makers, and on-chain treasuries are now active participants, bringing with them higher expectations for risk management and capital protection. Yet despite this growth, the overwhelming majority of DeFi capital remains uninsured.

A purpose-built, on-chain insurance primitive is required: parametric triggers, deterministic settlement, and composable capital. *LayerCover* addresses this gap with a marketplace for specialised underwriting and predictable, rule based payouts.

1.2 Limitations of Current Models

Today, the on-chain cover market is dominated by **mutualised risk pool** models. In these systems, large, undifferentiated capital pools are used to cover many protocols simultaneously. While aggregation offers certain efficiencies, these models suffer from several structural drawbacks:

- **Slow, subjective claims:** Claims decisions often rely on governance votes, introducing delays, uncertainty, and the risk of politicised outcomes eroding trust in the system.
- **Capital denomination risk:** In mutualised risk pools, underwriters must contribute to a shared pool whose assets are often volatile or denominated in the protocol's governance token. The inability to provide single-sided liquidity in a preferred stable or blue-chip asset exposes underwriters to unwanted market risk, reducing institutional appeal.
- **Fixed-term, prepaid cover:** Current mutual cover providers require buyers to commit to multi-month cover and pay the full premium upfront, even if the risk exposure changes. This lack of flexibility discourages usage for dynamic or short-term DeFi positions and limits recurring adoption.

These structural limitations have prevented mutual-based cover models from achieving meaningful market penetration. Despite hundreds of billions of dollars locked in stablecoins and DeFi protocols, active on-chain cover accounts for only a fraction of one percent of the total addressable market.

2 Key Benefits of LayerCover

A better system is possible. LayerCover improves upon the aforementioned issues, by offering the following advantages:

2.1 Underwriter Advantages

Single-Sided Liquidity: Underwriters can pledge capital in one asset of their choice (e.g. USDC, ETH). Each pledge to a pool is made in a single asset, but an underwriter may open multiple positions across different pools for different assets. This enables diversification while preserving the benefits of single-sided liquidity.The result is a more predictable, controlled risk-return profile, suitable for both institutional and sophisticated retail participants.

Dual-Revenue Efficiency: LayerCover enables the same principal capital to be pledged across multiple independent pools, each assigned a risk rating and governed by a total risk-points budget. This structure allows leveraged premium generation while maintaining strict solvency limits, with mutex groups further limiting exposure to correlated risks. In addition to premiums, underwriters also earn yield from idle capital deployed through yield adapters, these are contract modules that place unallocated funds into low-risk external strategies (see Glossary), while allowing instant recall for claims. By combining premium income with yield from integrated capital strategies, LayerCover delivers one of DeFi's most competitive venues for yield generation.

Salvage Rights on Payout: Underwriters receive insured assets from claims, recovering potential residual value (see Sec. 4.6 for full mechanics).

2.2 Policyholder Advantages

Perpetual, Pay-As-You-Go Cover: Unlike fixed-term policies that require upfront payment, LayerCover offers a perpetual model in which cover remains active as long as the premium deposit sustains the pro-rated, per-second cost. Premiums dynamically adjust based on pool utilisation. Policies lapse automatically when deposits run dry, but holders can top up at any time, cancel for a refund of unused deposit, or request coverage increases, providing flexibility and cost efficiency.

Instant, Rules-Based Settlement: Claims paid automatically with no governance delay. A small claim fee is payable to deter frivolous claims. Full mechanics of the claim process are described in Sec. 3.2 and Sec. 4.6.

Protocol-native reinsurance protection: Backstop reserves ensure full claim settlement, during extreme liquidity events. This is a rare event, since solvency floors normally ensure primary pools remain capitalised. (see Sec. 3.3 for details).

2.3 Partner Advantages

Referral Program: Frontend operators such as DeFi protocols, aggregators, and wallets can join LayerCover's referral program by receiving unique referral codes to share with their users and communities.

When cover is purchased through these codes, the referring partner earns a percentage of the premiums while the user benefits from a policy discount. Rewards are distributed automatically and transparently on-chain, creating a recurring, low-overhead revenue stream for partners.

Mutually Reinforcing Value: The referral system aligns incentives across all participants. End-users gain access to flexible on-chain protection, referrers earn a share of the premiums, and LayerCover scales its reach through trusted community and platform partners. This creates a positive feedback loop: protection for users, revenue for referrers, and ecosystem growth for the protocol.

3 User Operations

3.1 Underwriter Operations

Underwriters participate in the system by depositing capital into the CapitalPool, an ERC-4626 vault that issues shares at the current price-per-share (PPS). These deposits form the underwriter's principal, which can then be allocated across multiple risk pools through the UnderwriterManager contract (UM). Allocation does not physically move assets; rather, it records that a portion of the underwriter's principal is pledged to back coverage in the chosen pools. Each pool consumes part of the underwriter's finite risk-points budget, and allocation rules such as mutex groups or maximum pool counts apply. Once allocated, underwriters begin earning rewards from premiums, but they also become responsible for bearing losses. Rewards accrue continuously and can be claimed at any time, while losses are applied in shares so that they remain neutral to changes in PPS.

If an underwriter wishes to reduce their exposure to a pool, they may submit a deallocation request for a chosen amount. This request enters a governance-defined notice period, during which the specified capital is locked and cannot be reallocated elsewhere. The notice period protects solvency by preventing underwriters from instantly withdrawing in response to an imminent claim. Once the notice expires, the underwriter may execute the deallocation, at which point the pledge to that pool is cut and the capital becomes unpledged within the vault. Unpledged capital remains in the system until it is withdrawn, but no longer backs coverage in that pool.

Separately, an underwriter may request to withdraw funds from the CapitalPool. Withdrawals also follow a notice period, after which the request can be executed to burn shares and return the underlying asset. Importantly, withdrawals automatically scale down the underwriter's pledges across all pools in proportion to their reduced principal, meaning it is not necessary to first deallocate a matching amount from individual pools. To safeguard solvency, the UM enforces a coverage floor that requires each pool to retain capital in excess of its sold coverage and pending losses. As a result, if a withdrawal request would breach this floor, the system executes it only up to the allowed amount, leaving the remainder pending until conditions permit.

3.2 Policyholder operations

Policyholders obtain coverage by purchasing a **policy NFT** from their chosen pool. Each policy represents a fixed coverage amount backed by a deposit that continuously pays premiums at the pool's current rate. As long as the deposit can fund accrued costs, the policy remains active. Policies are transferable: whoever owns the NFT controls its coverage, deposit, and the right to make claims.

When a buyer purchases coverage, they select a pool, specify a coverage amount, and provide an initial premium deposit. The policy NFT is minted immediately, but coverage only becomes active after a governance-defined cooldown period. This prevents opportunistic behaviour, e.g. buying cover after an exploit has already begun. Both coverage and deposit must be non-zero at inception. Once active, the deposit is drawn down second by second. The policyholder may top up the deposit at any time; before accepting additional funds, the system first settles any premium owed to the current block and then updates the balance.

Coverage may also be adjusted over the life of a policy. If the owner wishes to increase their cover, they can submit a request that is placed into a pending queue. Each increase becomes effective only after its own cooldown, and is finalized automatically on the next interaction, such as a top-up, another increase request, or a claim. Multiple pending increases are allowed, subject to a bounded queue size to keep costs predictable. Conversely, reductions in coverage take effect immediately once premiums are settled. A partial reduction simply lowers the insured amount while leaving the deposit in place, whereas a full reduction closes the policy entirely and refunds the remaining deposit to the owner.

A policy can also be cancelled voluntarily. After activation, the holder may close their policy at any time; the protocol settles outstanding premium to the current block and refunds the unused deposit. Policies can also terminate automatically by lapse if the deposit is exhausted. Lapsed policies may be reactivated by topping up, but reactivation is subject to the cooldown period to prevent opportunistic behaviour during incidents.

If a covered event occurs, the policyholder may claim up to the current coverage amount. The claimant transfers any insured asset (if required) to the protocol, and in return receives an immediate payout from the pool's pledged capital, net of the pool fee. Claimants receive their payout in the pool's underlying asset (see Sec. 4.6 for salvage rights distribution).

Example:

- User purchases $100k cover with $5k deposit → NFT minted.
- Ten days later, user tops up $2k → deposit extended.
- User requests +$50k cover → enters queue, finalises after cooldown.
- User reduces $30k cover → effective immediately.
- User cancels → unused deposit refunded.

Restrictions:

- If a pool is paused due to an incident (see Section 4.5), new purchases and increases are disabled, but reductions, cancellations, and claims remain available.
- Each new policy and increase is subject to cooldown; reductions and cancellations require the policy to already be active.

3.3 Protocol-Native Reinsurance (Tranched Backstop)

The Reinsurance Layer (formerly the Backstop Pool) functions as a protocol-native safety net, ensuring claim settlement even during extreme liquidity events. Unlike traditional monolithic pools, LayerCover implements a structured **Shared Asset Controller** architecture. This design aggregates liquidity from multiple independent **Tranche Vaults**, enabling capital providers to select their preferred risk-adjusted return profile through specific priority tiers: Junior, Mezzanine, or Senior.

3.3.1 Capital Seniority and Waterfall Mechanics

The system enforces a strict seniority structure, known as the "loss waterfall," to determine the order in which capital is utilised for payouts. Each tranche vault t is assigned a priority index P_t, where a lower index corresponds to higher seniority (lower risk).

In the event of a solvency shortfall in a primary underwriting pool, the protocol triggers a drawdown from the Shared Asset Controller. Realised losses are allocated in *reverse priority order*:

- **Junior Tranche (First Loss):** This capital acts as the first line of defense. It absorbs claims immediately after primary pool covers are exhausted. Due to this high exposure, it commands the highest share of premiums.
- **Mezzanine Tranche:** Capital is drawn from this layer only after the Junior tranche has been fully depleted. It balances moderate risk with moderate rewards.
- **Senior Tranche (Last Loss):** This capital is the most protected layer in the system (P_0). It is utilised only in catastrophic tail events where both Junior and Mezzanine liquidity have been exhausted.

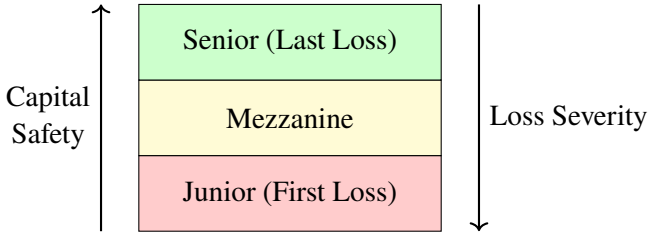

Figure 1: Reinsurance Capital Waterfall: Losses absorb from the bottom up (Junior → Senior), while capital safety increases from top to bottom.

3.3.2 Weighted Reward Distribution

To compensate for the asymmetric risk exposure defined by the waterfall, the "Reinsurance Tax" (the percentage of gross premiums routed to the backstop, see Sec. 4.5) is distributed differentially.

The Shared Asset Controller assigns a **Premium Weight** (W_t) to each tranche. Rewards are distributed pro-rata based on these weights rather than simple liquidity provision. This ensures that Junior depositors receive a significantly higher APY to compensate for their "First Loss" position, while Senior depositors earn a lower, more predictable yield reflecting their protected status.

3.3.3 Yield Efficiency via Shared Assets

While risk is segmented into tranches, liquidity is aggregated for efficiency. The Shared Asset Controller pools idle USDC from all tranches (Senior + Mezzanine + Junior) and deploys the aggregate balance into a single, whitelist-approved **Yield Adapter** (e.g., Aave V3 or Compound V3).

This architecture ensures that all reinsurance capital remains productive, earning external DeFi yields on top of their respective insurance premium shares, without fragmenting liquidity across multiple small deposit pools.

3.4 Managed Underwriting (Syndicates)

While individual underwriters can manage their own allocations, LayerCover also supports a delegated, professionally managed underwriting model known as *Syndicates*. A Syndicate is a specialized, manager-led

vault that aggregates user capital to deploy liquidity across LayerCover's risk pools.

3.4.1 Structure and Capital Flow

Syndicates function as ERC-4626 compliant vaults, serving as the sole entry and exit point for passive depositors. Instead of analyzing individual protocols and managing complex allocation weights personally, users deposit a single asset (e.g., USDC) into a Syndicate and receive syndicate shares. These shares represent their pro-rata ownership of the vault's underlying capital and accrued yield.

The Syndicate Manager, typically a professional risk assessor or a DAO retains the exclusive authority to allocate this pooled capital to specific underwriting pools via the `Underwriter Manager` contract. This structure allows the manager to actively adjust risk exposure, diversify across mutex groups, and manage leverage ratios on behalf of all depositors. Crucially, the underlying capital inherits the protocol's standard notice periods and loss settlement rules, ensuring that syndicate liquidity subject to the same solvency constraints as individual underwriters.

3.4.2 Fee Mechanics and Incentives

To align the interests of managers and depositors, Syndicates utilise a configurable, on-chain fee model. Managers may charge:

- **Performance Fees:** Calculated using a high-watermark formula based on the vault's share value appreciation (e.g., premiums earned minus realised losses).
- **Management Fees:** A streaming rate charged on the total assets under management.

Both fee types are capped at the factory level (e.g., 20%) to protect depositors. These fees are accrued automatically and claimed by the manager, ensuring that compensation is strictly tied to the successful preservation and growth of the Syndicate's capital.

3.4.3 Registry and Security

Syndicates are deployed via a `SyndicateFactory`, which automatically registers them within the protocol. The core `UnderwriterManager` checks this registry to enforce access controls, ensuring that only valid Syndicate contracts (or individual underwriters) can allocate capital. This architecture streamlines the flow from passive liquidity to active protection, removing the need for intermediate curator layers while maintaining full auditability.

Withdrawals follow a 30–day notice period. This period is enforced to prevent depositors gaming the system by withdrawing cover before expected claims are made. Depositors request withdrawal by specifying a share amount, which starts the notice clock. After the notice elapses, they redeem the exact shares requested and receive the corresponding amount of USDC at the prevailing PPS. Redemption first uses idle USDC and, if needed, pulls additional liquidity from the yield adapter.

During claims, asset may be drawn from the backstop to cover payouts. These draws reduce Net Asset Value (NAV) and thus price-per-share (PPS) in real-time, making Backstop Pool depositors the underwriters of system–wide residual risk. In exchange, they benefit from premium inflows and yield.

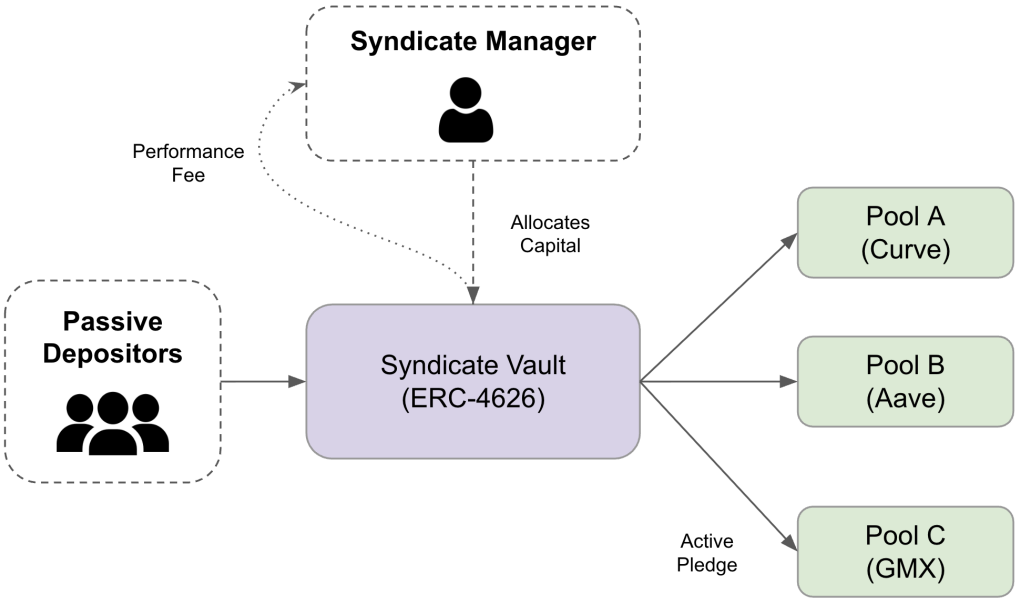

Figure 2: Syndicate Architecture: The Manager directs aggregated capital into diverse strategies, while Depositors retain passive exposure via a single vault token.

4 System Functionality

4.1 Vaults and Their Purpose

Vaults are lightweight escrow contracts that hold policyholder deposits on behalf of both the insured user and the underwriters backing their risk. Each vault is created deterministically for a specific policy and is responsible for maintaining balances, enforcing rights, and ensuring that obligations can always be settled in underlying units. By design, vaults guarantee that ownership and risk assignments can be updated without requiring tokens to move between external accounts.

The need for vaults arises because many DeFi protocols restrict or pause transfers of user positions during incidents or exploits. If assets were held directly in a user wallet, such restrictions could prevent the transfer of salvage rights from policyholders to underwriters, breaking LayerCover's parametric payout model. By routing deposits into vaults instead, the protocol avoids this problem. Ownership of the vault itself remains constant, but rights over its assets can be reassigned internally between policyholder and underwriters. No token transfer is required, so even if the underlying protocol halts withdrawals or freezes positions, the LayerCover contracts can still enforce salvage rights and pay out claims instantly.

In practice, a vault's balance may be split into two categories: *unassigned* funds, which remain under policyholder control, and *assigned* funds, which represent coverage pledged to underwriters. Policyholders may sweep unassigned funds, while underwriters may sweep assigned funds, ensuring both sides interact with the same escrow safely. If the vault cannot redeem all of the requested amount from a yield adapter or integration, it reports a shortfall, which the RiskManager contract settles through the insured withdrawal process. This maintains deterministic settlement for policyholders while keeping underwriter exposures accurate.

4.2 Hybrid Premium Pricing

LayerCover employs a dual-pricing architecture designed to balance immediate liquidity availability with price certainty for large hedging positions. This hybrid model consists of two distinct pricing tiers:

1. **Variable Rate (Pool-Based):** The core liquidity hub offering perpetual, pay-as-you-go coverage.
2. **Fixed Rate (Intent-Based):** An RFQ-style marketplace for fixed-duration, fixed-rate contracts.

4.2.1 Variable Rate Model (Adaptive Base Rate)

LayerCover's variable pricing is not static; it utilises an *Adaptive Base Rate (ABR)* mechanism that algorithmically adjusts the pricing curve based on the "implied risk" signal from the Fixed Rate (Intent) market. This ensures that the protocol's core liquidity pool never prices risk significantly below what professional underwriters are willing to accept in the open market.

The variable premium rate R_{var} is calculated dynamically at the time of a policy purchase:

$$R_{var}(U, R^{min}_{fixed}) = \max\left(R_{curve}(U), \quad R^{min}_{fixed} \times \alpha\right) \tag{4.1}$$

Where:

- $R_{curve}(U)$: The standard utilisation-based interest rate curve (calculated using the kinked model defined below).
- R^{min}_{fixed}: The lowest valid fixed-rate quote currently available in the active Intent Orderbook for the same risk pool.
- α: A governance-controlled dampening factor (e.g., 0.9 or 90%), allowing the variable pool to undercut the fixed market slightly to encourage liquidity usage, but preventing deep arbitrage.

The Utilisation Curve The underlying curve $R_{curve}(U)$ follows a dual-slope model to manage capital efficiency:

$$R_{curve}(U) = \begin{cases} R_{base} + \frac{U}{U_{opt}} \cdot S_1 & \text{if } U \leq U_{opt} \\ R_{base} + S_1 + \frac{U - U_{opt}}{1 - U_{opt}} \cdot S_2 & \text{if } U > U_{opt} \end{cases} \tag{4.2}$$

This hybrid approach creates a self-correcting market:

1. **Signal Propagation:** If professional underwriters perceive higher risk, they raise their fixed-rate quotes ($R^{min}_{fixed} \uparrow$).
2. **Algorithmic Response:** The `DynamicRateEngine` detects this shift and automatically lifts the floor of the variable pricing curve.
3. **Protection:** This prevents the "lazy" capital in the variable pool from being sold cheaply against high-risk events that active market makers have already priced in.

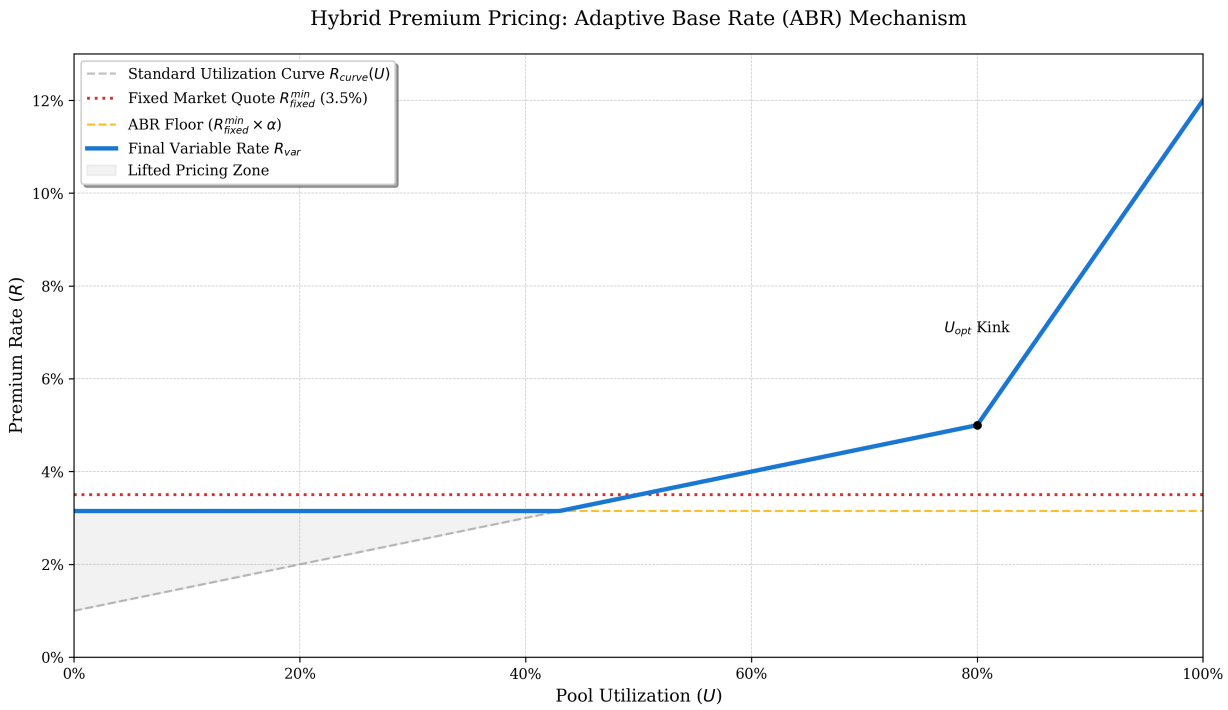

Figure 3: Utilisation–based premium curve

4.2.2 Adaptive Base Rate & Market Fallback

The `DynamicRateEngine` links the Variable Pool pricing to the Fixed Rate market to prevent arbitrage. However, to ensure continuous availability:

- **Market Signal:** When the Intent Orderbook is active, the Variable Rate is floor-bounded by the lowest valid fixed-rate quote (R^{min}_{fixed}).
- **Liquidity Failsafe:** In the event that the Intent Orderbook is empty or has insufficient volume to establish a reliable market signal, the engine automatically reverts to a governance-defined **Fallback Rate** ($R_{fallback}$).

This mechanism ensures the Variable Pool always generates a valid premium quote, preventing denial-of-service during periods of low market maker activity while protecting underwriters from mispricing.

4.2.3 Resilience Against Gamification and Market Manipulation

The Adaptive Base Rate (ABR) mechanism relies on open market signals to price risk accurately. This design necessitates specific defenses against manipulation strategies such as "spoofing" (posting fake low quotes) or "anchoring" (posting fake high quotes).

1. Rejection of High Outliers The ABR logic exclusively utilises the **lowest valid ask** (R^{min}_{fixed}) from the Intent Orderbook.

$$\text{ABR} = \min(Q_1, Q_2, \ldots, Q_n) \times \alpha$$

Quotes priced significantly above the market rate are automatically filtered out by this minimization function. They remain in the order book as valid future offers but have zero impact on the current Variable Pool pricing,

preventing underwriters from artificially inflating pool premiums to discourage competition.

2. The Lag vs. Spoofing Trade-off LayerCover deliberately utilises *active open quotes* rather than *filled historical orders* to determine the ABR.

- **Rationale:** Filled orders are lagging indicators. In a rapid de-pegging event, historical trades would price risk too low, allowing the Variable Pool to be drained cheaply before a new trade established a higher price.
- **Spoofing Defense:** To mitigate the risk of an underwriter posting an artificially low quote to suppress the variable rate (spoofing), the protocol enforces the **Arbitrage Trap**. Any quote posted to the order book is instantly executable. If an underwriter posts a quote below the true risk premium to manipulate the ABR, arbitrage bots and rational hedgers can atomically fill that quote, forcing the attacker to underwrite risk at a loss that exceeds their potential gain from the manipulation.

3. Utilisation Failsafe The ABR serves only as a pricing *floor*. It cannot override the core solvency logic of the utilisation curve.

$$R_{final} = \max(\text{ABR}, R_{curve}(U))$$

If manipulation successfully suppresses the ABR, the resulting low price stimulates demand, increasing the pool's Utilisation (U). As U rises beyond the optimal kink (U_{opt}), the steep slope S_2 engages, driving the premium rate up exponentially and independently of the order book signal.

4.2.4 Fixed Rate Model (Intent-Based)

For institutional hedgers, market makers, and large coverage amounts, the variable rate model introduces slippage and budgeting uncertainty. To resolve this, LayerCover introduces *Intent-Based Pricing*.

In this model, Underwriters (via Syndicates) sign cryptographically verifiable "Intents" off-chain. These are binding offers to provide specific coverage amount C at a fixed annualized rate r_{fixed} for a fixed duration T. Unlike the variable model, the premium cost P_{intent} is deterministic and paid upfront:

$$P_{intent} = \frac{C \cdot r_{fixed} \cdot T}{365 \times 10000} \tag{4.3}$$

Capital Efficiency and Locking: When a policyholder matches with an intent, the specific Underwriter's capital is atomically locked in their Syndicate vault for the exact duration T. This eliminates the "socialized" aspect of the variable pool; the Underwriter receives 100% of the underwriting premium share directly, rather than a pro-rata share of the general pool.

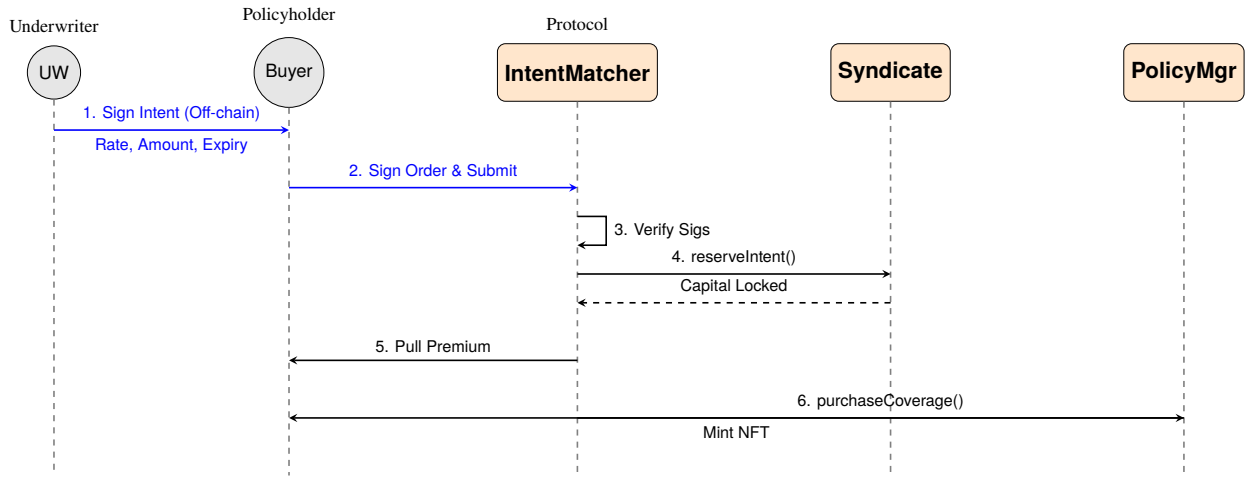

Figure 4: Intent Matching Lifecycle: An atomic RFQ process where off-chain signatures from the Underwriter and Buyer are matched on-chain, triggering simultaneous capital locking and policy issuance.

4.3 Loss Distribution

Summary. When a claim is paid, the resulting loss is prorated across all underwriters who had pledged capital to that pool at the time of the event. Each underwriter's share of the loss is proportional to their active pledge at that time. This ensures no one can avoid losses by withdrawing immediately after the event, and the system stays price–neutral and efficient even when many underwriters participate.

Objective. Allocate realised claim losses to underwriters in a way that is (i) pro–rata to the capital that actually backed the risk at the moment of the event, (ii) *price–neutral* with respect to pool PPS changes, and (iii) $O(1)$ to update regardless of underwriter count.

Snapshot and indexing. On a payout of total amount X units of the pool's underlying, the pool first takes a *price–neutral* snapshot by reserving shares corresponding to X using the current $valueToShares(\cdot)$ conversion (Def. 6.1):

$$\texttt{unsettledPayoutShares} \leftarrow \texttt{unsettledPayoutShares} + \texttt{valueToShares}(X).$$

This does not change NAV at the instant of snapshot; NAV moves only as assets are actually transferred (Sec. 6.1).

Losses are then accounted for with a *shares–per–pledge* index $D_i^{(sh)}$ maintained per pool i (Defs. 6.3–6.4). Let L be the realised loss value (in underlying units) attributable to the pool after any backstop injection (if applicable). Define

$$\texttt{lossShares} = \texttt{valueToShares}(L), \quad D_i^{(sh)} \leftarrow D_i^{(sh)} + \frac{\texttt{lossShares} \cdot \texttt{PRECISION}}{\texttt{totalPledged}_i^{\star}},$$

where $\texttt{totalPledged}_i^{\star}$ is the *eligible pledge base* at the event block: pledged capital after applying notices/locks and any solvency floor adjustments (§3.1). This freezes the loss base against subsequent reallocations.

Settlement at underwriter level. Each underwriter u in pool i tracks a per–pool snapshot $d_{u,i}^{(sh)}$. Their pending loss (in *shares*, not value) updates lazily on interaction:

$$\texttt{pendingLossShares}_{u,i} = \max\left(0,\ \left\lfloor \frac{P_{u,i}^{\text{active}} \cdot \Delta D_i^{(sh)}}{\texttt{PRECISION}} \right\rfloor - d_{u,i}^{(sh)}\right),$$

where $P_{u,i}^{\text{active}}$ is the underwriter's *eligible* pledge at the event block. Settlement burns these shares at current PPS, ensuring price–neutrality across time:

$$\text{burn}(\texttt{pendingLossShares}_{u,i}), \qquad d_{u,i}^{(sh)} \leftarrow d_{u,i}^{(sh)} + \text{increment}.$$

Edge cases and invariants.

- **Concurrent claims:** multiple events simply add to $D_i^{(sh)}$; order does not matter.
- **Zero eligible base:** if $\texttt{totalPledged}_i^{\star} = 0$, no index update occurs; the claim is fully met by the BackstopPool (Sec. 3.3), and backstop NAV absorbs the loss.
- **Withdrawal/deallocation after event:** attempts to avoid losses via post event actions are inert; the loss base is fixed at the event block by $\texttt{totalPledged}_i^{\star}$.
- **Contagion (optional):** if governance defines dependency edges between pools, a loss in pool A may emit synthetic losses into pools $\{B\}$ via pre–declared weights; each affected pool applies its own $D^{(sh)}$ tick with its local eligible base.
- **Rounding:** all divisions use flooring with PRECISION scale; any dust remains inside the pool and is socialised over time, consistent with fixed point rules (Ref. [19]).

This design yields constant time pool updates and strictly pro–rata underwriter attribution, while remaining robust to PPS drift between event and settlement.

4.4 Reward Distribution

Summary. Premiums and incentives are streamed continuously to underwriters in proportion to pledged capital. This ensures fairness and prevents underwriters from avoiding losses by withdrawing post-event. Settlement is atomic and safe from transaction interruption, eliminating this as an attack surface..

Sources. Underwriter rewards comprise (i) streaming premiums from active policies, (ii) pool incentive tokens (if configured), and (iii) strategy yield rebated to the pool. All rewards are accounted *per token* τ via an $R_{i,\tau}$ index (Sec. 6.2).

Streaming and indexing. Premiums accrue continuously per second from each active policy and are periodically consolidated into the pool's reward stream. On each credit of amount Q of token τ to pool i:

$$R_{i,\tau} \leftarrow R_{i,\tau} + \frac{Q \cdot \texttt{PRECISION}}{\texttt{totalPledged}_i^{\dagger}},$$

where $\texttt{totalPledged}_i^{\dagger}$ is the *current* eligible pledge base at the time of distribution. For an underwriter u:

$$\texttt{pendingReward}_{u,i,\tau} = \max\left(0,\ \left\lfloor \frac{P_{u,i}^{\text{active}} \cdot \Delta R_{i,\tau}}{\texttt{PRECISION}} \right\rfloor - d_{u,i,\tau}\right),$$

with $d_{u,i,\tau}$ the underwriter's last reward snapshot. Rewards settle (transfer to u) on any user or pool interaction; settlement also advances $d_{u,i,\tau}$.

Multi–token and fees. The mechanism supports multiple τ concurrently (e.g., USDC premiums, incentive ERC–20s). Claim fees (if set) are split according to governance parameters among (i) active underwriters, (ii) the BackstopPool, and/or (iii) a treasury, each via its own index.

Fairness and robustness.

- **Price neutrality:** rewards are paid in tokens, not shares; no implicit mint/burn of pool equity occurs.
- **Eligibility symmetry:** the same notion of $P_{u,i}^{\text{active}}$ used for losses is used for rewards, so capital that backs risk also earns the stream that compensates it.
- **DoS resistance:** index updates are $O(1)$, independent of the number of underwriters or active policies.
- **Rounding:** flooring at distribution ensures the pool never over pays; unallocated dust accumulates to the pool and is socialised.

4.5 Premium Splits & Reinsurance Tax

Gross premiums paid by policyholders are split at the source to ensure continuous capitalization of the reinsurance layer.

$$P_{total} = P_{underwriter} + P_{backstop} \tag{4.4}$$

Where:

- **Backstop Share ($P_{backstop}$):** A fixed percentage (Default: 20%) is routed directly to the Backstop Pool. This acts as a protocol-native reinsurance tax, incentivizing passive liquidity providers to cover tail risks.
- **Underwriter Share ($P_{underwriter}$):** The remaining 80% is streamed to the specific risk pool and distributed to active underwriters pro-rata to their pledged capital.

4.6 Oracle–free design and unit–based accounting

A defining feature of the protocol is its independence from external price oracles. Core functions including policy management, premium accrual, and claim settlement are executed deterministically on–chain, without reliance on off–chain feeds. The sole oracle function is circuit breaking: for example, the protocol may pause new policy issuance if an oracle reports that an asset has de–pegged beyond a set threshold. This safeguard prevents policyholders from purchasing cover only after a loss event has already occurred.

This design removes a major DeFi attack vector: stale/manipulated prices and governance exploits. By avoiding oracle dependencies, LayerCover ensures predictable execution and minimizes systemic risk.

Unit–based accounting reinforces this robustness: all liabilities and payouts are denominated in the underlying asset itself. Underwriters and policyholders share a common unit of account, eliminating valuation disputes and ensuring transparency of obligations.

In contrast, many on–chain insurance models rely on continuous oracle inputs or governance votes to adjudicate claims mechanisms that introduce delay, discretion, and manipulation risk. LayerCover's oracle–free, unit–based approach guarantees that claims resolve instantly and mechanically.

4.7 Flexible Custody & Salvage Mechanics

LayerCover employs a **Hybrid Custody Architecture** that decouples financial settlement (payouts) from asset recovery (salvage). This allows the protocol to support both censorship-resistant assets (via non-custodial

swaps) and complex, potentially pausable DeFi positions (via escrow vaults) without fragmenting the underwriting capital. In all cases, salvage rights are distributed to underwriters pro-rata to their liability at the precise block of the claim (t_0).

4.7.1 Model A: Direct Swap (Put Option Mechanics)

For immutable assets (e.g., WETH, RAI), the claim mechanism operates effectively as a **perpetual American-style Put Option**. To execute a claim, the policyholder approves and transfers the insured asset to the protocol.

Validation is purely procedural: The RiskManager verifies only that the policy is active, the premium is paid, and the user holds the insured asset. No external oracle or governance vote is required to confirm a "loss event." If the user chooses to exercise their claim—regardless of market conditions—the protocol accepts the asset as salvage and pays out the covered amount from the Capital Pool instantly.

4.7.2 Model B: Escrow Vault (Pre-Funded Failsafe)

For yield-bearing positions, lending market deposits, or vault shares (e.g., **aUSDC on Aave, Gauntlet Vault shares, or Compound-V3 positions**) where the underlying protocol may pause transfers during an exploit, reliance on an external transfer is an unacceptable risk. If the external protocol (e.g., Aave) pauses the token contract, a Direct Swap would revert, leaving the user unpaid. LayerCover solves this via **Escrow Vaults**.

- **Architecture:** Each policy is linked to a deterministic `EscrowVault` contract deployed at purchase. This vault custodies the insured DeFi position (e.g., the aToken or Vault Share) and maintains an internal ledger separating beneficial ownership into two states:
  - `unassignedUnderlying`: Owned by the **Policyholder** (sweepable anytime).
  - `assignedUnderlying`: Owned by the **Underwriters** (locked for salvage).
- **The Atomic Assignment (The "Ledger Flip"):** In the event of a claim—even if the underlying DeFi platform has completely halted withdrawals—the `EscrowClaimManager` does *not* attempt to move tokens on the blockchain. Instead, it calls `assignToUnderwriters(amount)` on the vault. This function executes an $O(1)$ state update:

$$\text{unassigned}' = \text{unassigned} - \text{claimAmount} \tag{4.5}$$

$$\text{assigned}' = \text{assigned} + \text{claimAmount} \tag{4.6}$$

- **Result:** The policyholder receives their payout immediately from the capital pool. The underwriters instantly acquire the *legal and technical title* to the insured positions inside the vault. Because this logic is internal to the `EscrowVault` contract, it **cannot be blocked** by the external protocol's pause state.

4.7.3 Salvage Distribution Mathematics

Regardless of the custody model (Swap or Escrow), the accounting for underwriter entitlements is identical. We define q_a as the quantity of salvage asset a acquired by the pool (either received via swap or assigned in escrow).

Unlike premium income which accrues continuously via an index, salvage rights are **fixed at the moment of the claim (t_0)**. This ensures that underwriters who backed the risk at the time of the loss retain their recovery rights indefinitely, even if they subsequently withdraw their capital from the pool.

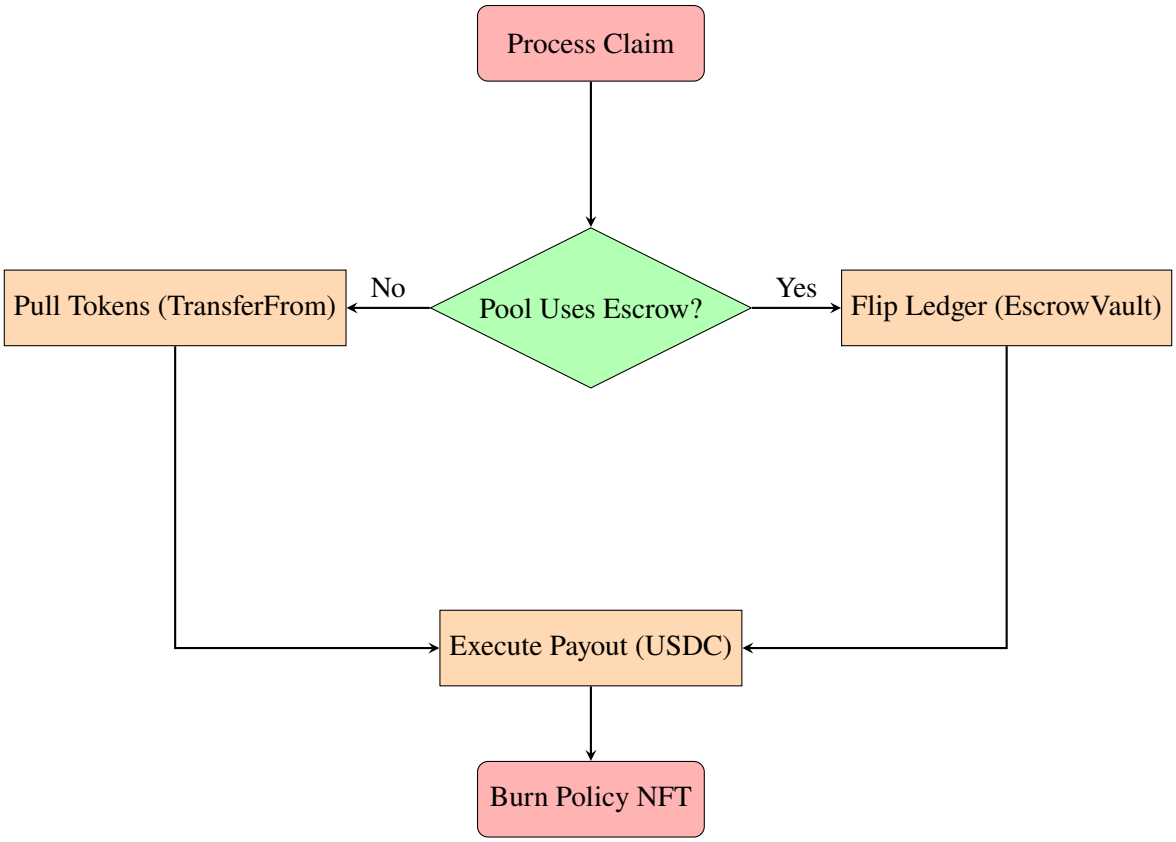

Figure 5: Logic flow for Claim Settlement showing the bifurcation between standard Token Salvage and Escrow Assignment paths.

4.7.4 Discrete Accounting

Instead of a global pool index, salvage is tracked per individual policy claim. When a claim is initialized for Policy k at block t_0:

1. The protocol records the **Total Capital Provided** ($Cap_{total}^{t_0}$) by all underwriters at that specific block.
2. The insured assets (quantity q_a) are deposited into a distinct salvage vault for Policy k.

4.7.5 Claiming Entitlements

An underwriter u can claim their share of the salvage at any time. Their entitlement is static and proportional to their pledge *at the time of the loss*, ensuring no dilution from new capital entering the pool post-event.

$$Entitlement_{u,k} = q_a \times \frac{Pledge_u^{t_0}}{Cap_{total}^{t_0}} \tag{4.7}$$

Where:

- $Pledge_u^{t_0}$ is the underwriter's active pledge at block t_0 (historically queryable via `UnderwriterManager`).
- $Cap_{total}^{t_0}$ is the total eligible underwriting capital at block t_0.

4.7.6 Resolution of Frozen Assets

If the salvage assets are "frozen" inside an Escrow Vault (Model B), underwriters do not receive the raw token immediately. Instead, they receive a **transferable claim right** to the `assignedUnderlying` balance.

- Underwriters may call `transferAssignedPosition()` to trade these rights on secondary markets (OTC) while the asset remains frozen.
- This creates a liquid market for "insured salvage" without requiring the underlying protocol to unpause, ensuring underwriters can exit positions and recycle capital efficiently.

5 Governance and Risk Management

The **LayerCover** protocol is designed to minimise discretionary governance intervention. Governance primarily sets static parameters such as pool risk ratings, mutex group definitions, and pricing curves. Execution of underwriting, loss distribution, and payouts is fully automated on–chain.

5.1 Risk Points System

The *risk points system* provides a quantitative budget that constrains how much leverage an underwriter can take across multiple pools.

Definition. Each pool i is assigned a risk cost c_i in *risk points*, proportional to its perceived underwriting risk. An underwriter u has a maximum budget TOTAL_RISK_POINTS, enforced at allocation time:

$$\sum_{i \in A_u} c_i \;\leq\; \text{TOTAL\_RISK\_POINTS}. \tag{1}$$

Purpose.

- Prevent concentration of exposure across many high–risk pools.
- Enable differentiated leverage: lower–rated pools consume fewer points.
- Allow governance to tune systemic risk without micromanaging capital flows.

[tikz,border=10pt]standalone tikz

5.2 Pool Ratings and Constraints

Each pool has a *risk rating* and associated *constraints* recorded in the `PoolRegistry`:

- **Risk Rating:** Discrete labels (AAA, AA, A, BBB, BB, B, C) reflect the perceived risk of the protocol or asset.
- **Mutex Groups:** Exclusion sets prevent underwriters from allocating to correlated pools (e.g., DAI and USDC).
- **Capacity Limits:** Optional per–pool caps in absolute terms or as a % of total NAV, preventing over–concentration.
- **Fee Parameters:** Governance–set minimum/maximum premium rates and payout fee bps.

Ratings and constraints are updated via governance proposals, subject to timelocks to avoid disruption. Changes to mutex groups or capacity limits affect only new allocations.

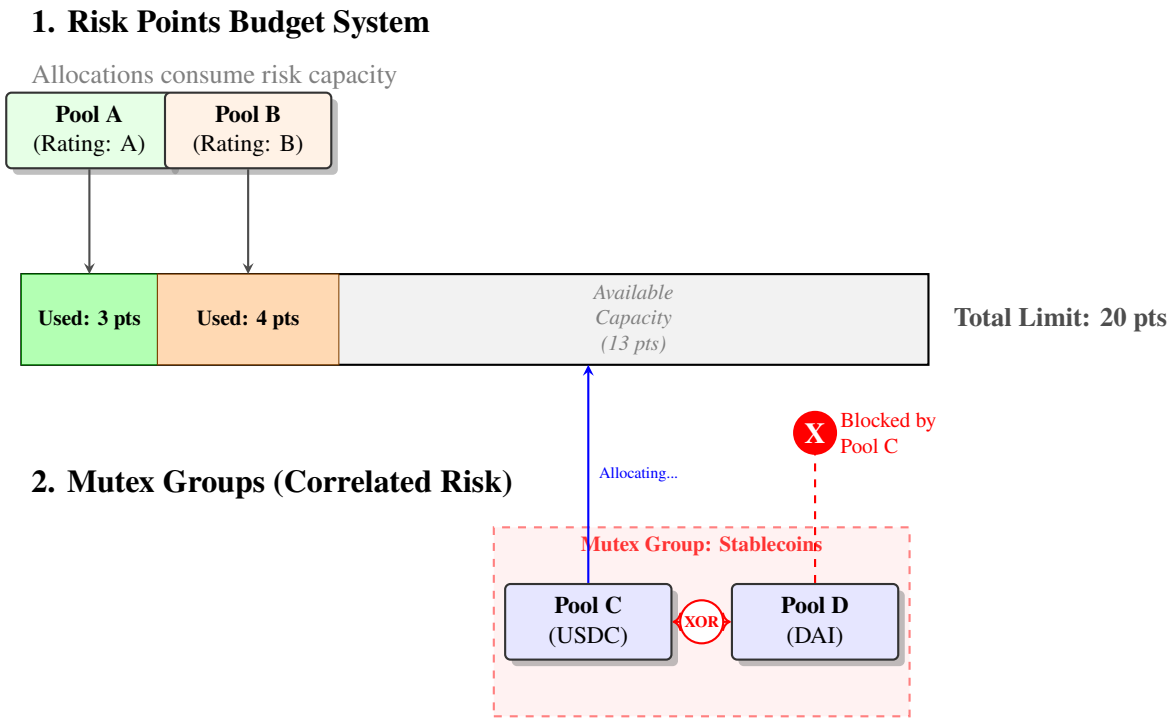

Figure 6: Risk Points and Mutex Group Architecture

5.3 Risk Framework Alignment with Traditional Capital Models

Although **LayerCover** is DeFi–native, its risk ratings, leverage constraints, and pool–level controls can be mapped to established solvency frameworks used by insurers and banks. This alignment helps institutional participants translate on–chain exposures into familiar capital adequacy metrics.

5.3.1 Mapping to Solvency II (Insurance)

The EU Solvency II regime requires holding capital to survive a $1/200$ year event (the Solvency Capital Requirement, SCR). LayerCover parallels this structure as follows:

- **Pool Ratings ⇒ SCR calibration.** Ratings (driven by protocol risk, liquidity, volatility, resilience) correspond to Solvency II standard formula or internal models for capital requirements.
- **Risk Points ⇒ concentration controls.** A portfolio–wide cap on risk points mirrors SCR concentration add–ons and dependency modelling.
- **Mutex Groups ⇒ dependency structure.** Exclusion sets replicate correlation constraints in SCR aggregation.
- **Parametric Payouts ⇒ predictable loss quantification.** Deterministic triggers reduce model risk and shorten settlement, improving capital modelling.

5.3.2 Mapping to Basel III/IV (Banking)

Basel standards focus on risk–weighted assets (RWA), leverage, and liquidity coverage. Analogues include:

- **Pool Ratings ⇒ risk weights.** Each pool's rating maps to a risk weight for pledged exposure, yielding an RWA–like measure.

- **Leverage via Pledges ⇒ leverage ratio.** Total pledged exposure relative to principal acts as a non–risk–based leverage cap.
- **Stress Testing ⇒ correlated scenarios.** Historical exploit data, TVL drawdowns, and oracle disruptions can emulate Basel stress tests.
- **Liquidity Coverage.** Pool buffers and backstops emulate a Liquidity Coverage Ratio (LCR)–style requirement.

5.3.3 Institutional Integration Benefits

- **ERM Reporting:** Immutable on–chain records support export into enterprise risk management and compliance systems.
- **Auditability:** Public, append–only state creates a native audit trail for supervisors and internal audit.

5.3.4 At–a–Glance Mapping

LayerCover Feature	Solvency II Analogue	Basel III/IV Analogue
Pool risk rating	SCR calibration (standard formula / internal model)	Risk weight assignment (RWA)
Risk points budget	Concentration limits in SCR aggregation	Portfolio–level leverage constraints
Mutex groups	Dependency / correlation structure	Correlation caps across asset classes
Parametric claim triggers	Predictable loss quantification	Deterministic loss recognition
Liquidity/backstop config	Liquidity planning for claims	LCR/NSFR–style liquidity buffers
Pledge leverage vs. principal	Capital add–ons for risk	Non–risk–based leverage ratio
On–chain audit trail	Auditability for supervisors	Data lineage for internal control

This modal enables institutional capital to flow on–chain without breaking off–chain governance, risk, and compliance (GRC) guardrails.

6 Core Economics & Accounting

LayerCover's economics boil down to a few composable rules:

- Shares price the pool by dividing total asset value (NAV) over an effective supply that excludes payout-reserved shares, so deposits and withdrawals are always fair.
- Payouts are price-neutral at the moment they're declared—NAV doesn't jump, we just reserve shares—so no one is diluted or advantaged mid-event.
- Rewards accrue continuously via simple indices, with each underwriter's stake adjusted for any notices or locks.
- Losses are realised by burning shares, with a shares-per-pledge index ensuring fairness regardless of PPS changes.
- Premiums are set by a transparent utilisation curve, making pricing predictable under load.

The formulas below are the minimal math to understand value, yield, and risk transfer; operational details (notices, deallocations, coverage floors) live in the appendix and don't change these core incentives.

6.1 Shares, NAV, and Price Neutrality

Effective circulating supply.

$$\text{EffShares} \equiv \text{TotalShares} - \text{unsettledPayoutShares}.$$

Conversions.

$$\text{valueToShares}(V) = \begin{cases} V, & \text{if NAV} = 0 \text{ or EffShares} = 0 \\ \left\lfloor \dfrac{V \cdot \text{EffShares}}{\text{NAV}} \right\rfloor, & \text{otherwise} \end{cases} \tag{6.1}$$

$$\text{sharesToValue}(S) = \begin{cases} S, & \text{if EffShares} = 0 \\ \left\lfloor \dfrac{S \cdot \text{NAV}}{\text{EffShares}} \right\rfloor, & \text{otherwise.} \end{cases} \tag{6.2}$$

Payout snapshot (price-neutral). On a payout of total amount X the pool increases

$$\text{unsettledPayoutShares} \leftarrow \text{unsettledPayoutShares} + \text{valueToShares}(X),$$

so snapshotting does not change NAV at that instant; NAV moves as assets are actually transferred.

6.2 Rewards & Losses (High Level)

Reward indexing. When amount Q of reward token τ is distributed over pool i,

$$R_{i,\tau} \leftarrow R_{i,\tau} + \frac{Q \cdot \text{PRECISION}}{\text{totalPledged}_i}.$$

An underwriter's claimable reward is

$$\text{Pending}_{u,i,\tau} = \max\left(0,\ \left\lfloor \frac{P_{u,i}^{\text{active}} \cdot R_{i,\tau}}{\text{PRECISION}} \right\rfloor - d_{u,i,\tau}\right),$$

where $P_{u,i}^{\text{active}}$ is the user's *effective* pledge after notices/locks

Loss indexing (shares-based). On a realised loss L in pool i:

$$\text{lossShares} = \text{valueToShares}(L), \tag{6.3}$$

$$\mathcal{D}_i^{(\text{sh})} \leftarrow \mathcal{D}_i^{(\text{sh})} + \frac{\text{lossShares} \cdot \text{PRECISION}}{\text{totalPledged}_i}. \tag{6.4}$$

Users' pending loss shares are computed from their pledge and the change in $\mathcal{D}_i^{(\text{sh})}$ since their last snapshot; these shares are burned at settlement, reducing value at the then-current PPS.

6.3 Premium Pricing and Cost

Available capital and utilisation. Let

$$\text{availableCapital} = \max\big(0,\ \text{totalPledged} - \text{pendingWithdrawals}\big).$$

Utilisation in basis points is:

$$\text{utilBps} = \begin{cases} \dfrac{\text{sold} \cdot \text{BPS}}{\text{availableCapital}}, & \text{availableCapital} > 0 \\ \text{BPS}, & \text{otherwise.} \end{cases}$$

Capacity checks for new or increased coverage also account for any pending policy increases.

Rate curve (piecewise linear).

$$\text{Rate}_{\text{bps}} = \begin{cases} \text{base} + \dfrac{\min(\text{utilBps}, \text{kink}) \cdot \text{slope1}}{\text{BPS}}, & \text{utilBps} \le \text{kink} \\ \text{base} + \text{slope1} + \dfrac{(\text{utilBps} - \text{kink}) \cdot \text{slope2}}{\text{BPS}}, & \text{otherwise.} \end{cases}$$

Policy cost.

$$\text{PremiumCost} = \left\lfloor \frac{\text{Coverage} \cdot \text{Rate}_{\text{bps}} \cdot \text{ElapsedSeconds}}{\text{SECS\_YEAR} \cdot \text{BPS}} \right\rfloor.$$

6.4 Backstop (One-Paragraph Economics)

Premiums sent to the BackstopPool increase NAV without minting shares. Operational details of the Backstop mechanism are described in Sec. 3.3.

$$\text{NAV}' = \text{NAV} + \text{Premium}, \qquad \text{TotalShares}' = \text{TotalShares}.$$

6.5 Worked Examples

To illustrate how capital flows through the system, consider the following scenario with real numbers.

Scenario Setup

- Total value locked (TVL) in a DeFi protocol: **$10m**.
- Policyholders purchase cover totaling **$2m** from a LayerCover pool.
- The pool has **100 underwriters**, each pledging **$100k** in USDC, so pledged capital = **$10m**.
- Premium rate = **10% annualized**, utilisation-based.
- Coverage sold = **$2m**, so pool utilisation = **20%**.

Premium Flow

- Annual premium for $2m coverage at 10% = **$200k/year**, or $\approx \$16.7k$/month.
- This premium accrues per-second and is distributed pro-rata to underwriters.
- Each underwriter with a 1% share of the pool ($100k / $10m) earns:
  - **$2k/year** in premium income ($200k \times 1\%$).

Claim Event

- Six months later, the protocol suffers an exploit; insured assets fall in value.
- Policyholders trigger a claim for the full **$2m** coverage.
- The pool pays out **$2m** in USDC to claimants instantly.
- Losses are allocated proportionally:
  - Each underwriter's share of pool = 1%.
  - Each bears **$20k** of the $2m loss.
  - Their principal reduces from $100k $\rightarrow$ $80k.

Salvage Rights

- Claimants transfer insured assets (e.g., protocol tokens now worth **$500k** in secondary markets) into the pool as salvage.
- These are distributed to underwriters pro-rata to their loss share.
- Each underwriter receives **$5k worth** of insured tokens.
- Effective net loss per underwriter: $20k payout – $5k salvage = **$15k**.

Outcome After 6 Months

- Premiums earned before the claim: $2k $\times$ 0.5 years = **$1k** per underwriter.
- Net result per underwriter:
  - Initial capital = $100k.
  - Loss = –$15k.
  - Premiums = +$1k.
  - Ending position = **$86k** value ($80k USDC + $5k insured assets + $1k premium).

System-Level Observations

- Policyholders are fully compensated in stable USDC.
- Underwriters absorb the loss but partially offset it with premiums and salvage rights.
- Salvage creates asymmetric recovery: if insured assets recover above $500k in value, underwriters' realised losses shrink further.

7 Glossary of Key Terms

Active Pledge (P^{active}) The amount of an underwriter's capital currently backing coverage in a pool, after applying notice locks, cooldowns, and solvency floor constraints.

Backstop Pool An ERC-4626 vault that provides protocol-native reinsurance. It covers claim shortfalls when underwriting pools are temporarily illiquid or undercapitalised.

Claim Fee. A small fee payable by the policyholder when raising a claim. Designed to deter frivolous claims while keeping settlement predictable.

Cooldown Period A governance-defined delay before new or increased coverage becomes active. Prevents policyholders from purchasing cover opportunistically after a loss event has begun.

Coverage Floor The minimum capital each pool must retain, enforced by the `UnderwriterManager`, to ensure solvency when withdrawals or reallocations are requested.

Effective Shares (EffShares) The circulating pool shares used for NAV calculations. Defined as:

$$\text{EffShares} \equiv \text{TotalShares} - \text{unsettledPayoutShares}.$$

Loss Index ($D_i^{(\text{sh})}$) A per-pool shares-per-pledge index tracking realised claim losses. Ensures losses are attributed pro-rata and price-neutral regardless of PPS changes.

NAV (Net Asset Value) The total value of assets held in a pool or vault, denominated in the underlying token.

Notice Period A governance-defined waiting period before underwriters can deallocate or withdraw pledged capital. Protects solvency by preventing pre-claim withdrawals.

Oracle-Free Design Core functions (premium accrual, policy management, claim settlement) operate deterministically on-chain without external price feeds. The sole oracle use is circuit-breaking (pausing new policies on de-peg events).

Pending Loss Shares The shares an underwriter must burn to settle their portion of a claim, calculated from the change in the pool's loss index since their last snapshot.

Policy NFT A non-fungible token representing an active policy. Encodes coverage amount, deposit balance, and the right to claim. Transferable between users.

Premium Rate Curve A piecewise linear function defining how premium rates increase as pool utilisation rises, with separate slopes before and after a kink point.

Salvage Index ($S_{i,a}$) A per-pool, per-asset index tracking insured assets received during claims. Updates in raw units (not valuations) and allocates salvage entitlements pro-rata.

Salvage Rights Entitlements of underwriters to insured assets surrendered by policyholders during claims. Provides potential recovery of residual value.

Solvency Floors Capital requirements applied to ensure that total pledged assets in a pool always exceed coverage sold and pending losses.

Unsettled Payout Shares Shares reserved at the moment of a payout snapshot. NAV does not change until assets are actually transferred.

Utilisation The ratio of total coverage sold to available capital in a pool. Drives dynamic premium pricing.

`valueToShares` / `sharesToValue` Conversion functions between underlying token amounts and pool shares, ensuring price neutrality for deposits, withdrawals, and losses.

References

[1] Nexus Mutual, *Nexus Mutual Documentation*, ongoing. https://nexusmutual.io

[2] Sherlock, *Sherlock Protocol Documentation*, ongoing. https://docs.sherlock.xyz

[3] Unslashed Finance, *Unslashed Protocol Documentation*, ongoing. https://docs.unslashed.finance

[4] Rekt.news, "DeFi Exploit Postmortems (Euler, Curve, Mango, etc.)," ongoing. https://rekt.news

[5] Chainalysis, "Crypto Crime Report 2022," ongoing. https://blog.chainalysis.com/reports

[6] Gauntlet, "Risk Assessment for On-Chain Insurance Markets," Gauntlet Research, 2023.

[7] Bank for International Settlements, "DeFi Risk and Regulation," BIS Quarterly Review, 2022.

[8] Lloyd's of London, "About Lloyd's: History and Market Structure," overview page, ongoing. https://www.lloyds.com

[9] Basel Committee on Banking Supervision, "Basel III: A Global Regulatory Framework for More Resilient Banks and Banking Systems," Bank for International Settlements, 2011 (rev. updates). https://www.bis.org/bcbs/basel3.htm

[10] European Parliament and Council, "Directive 2009/138/EC (Solvency II)," Official Journal of the European Union, 2009 (as amended). https://eur-lex.europa.eu/eli/dir/2009/138/oj

[11] Centre Consortium, "USDC: Technical and Policy Resources," ongoing. https://www.circle.com/en/usdc