Discretized Liquidity with an Integrated Money Market and Progressive Liquidation

Qomira Labs

Version 1.0
September 2026

Abstract

Juncta is a discretized liquidity market maker in which the reserves held in inactive bins form the supply side of an integrated money market. On a conventional concentrated liquidity venue, only the bin containing the current price earns anything; every other bin the liquidity provider funded holds real capital that earns nothing until the price arrives. Juncta lends that capital out. A bin far from the active price lends most of what it holds, a bin near the price lends none of it, and the distance between them is priced as a term structure over distance rather than over time.

The capital does not move. Reclassifying a bin between trading and lending is a change of accounting rather than a transfer, so no external contract sits in the path and no wrapper token is issued.

The central difficulty is that a position whose reserves are lent out cannot also back a borrow, because the same dollars would stand behind two claims. We resolve this by restricting collateral capacity to the portion of a position that was never deployed, and show that this bounds total claims by total supply and makes seizure immediate by construction rather than by haircut. Liquidation proceeds in five levels, each closing only the amount needed to restore a target health factor, with recovery windows sized for collateral that prices on slow feeds, including tokenized real-world assets.

1 Introduction

A concentrated liquidity market maker lets a provider commit capital to a chosen price range instead of the whole curve. The capital efficiency this buys is real, and it is also narrower than it first appears. The provider commits across a range, the protocol divides that range into discrete bins, and at any moment exactly one bin contains the current price. That bin trades. The rest hold reserves and wait.

Measured across a live market, the share of deposited capital earning anything at a given moment sits in the low single digits. The rest is not misallocated and it is not idle by mistake. It is inventory, posted deliberately, against prices that may not arrive for days or at all. A

provider can narrow the range to concentrate capital where it trades, but that raises rebalancing frequency and exposure to divergence loss without changing the underlying fact: most of the capital in most positions is waiting rather than working.

Lending that inventory out is the obvious response, and in its naive form it is unsound. If a position's reserves are lent to a borrower, and the same position is then pledged as collateral for a second borrow, the same dollars stand behind two claims. No haircut corrects this, because a haircut adjusts the value assigned to collateral rather than the number of claims against a dollar. The failure appears again at liquidation, where seizing a position means withdrawing reserves that are not there to withdraw. Section 9 states the problem precisely and resolves it.

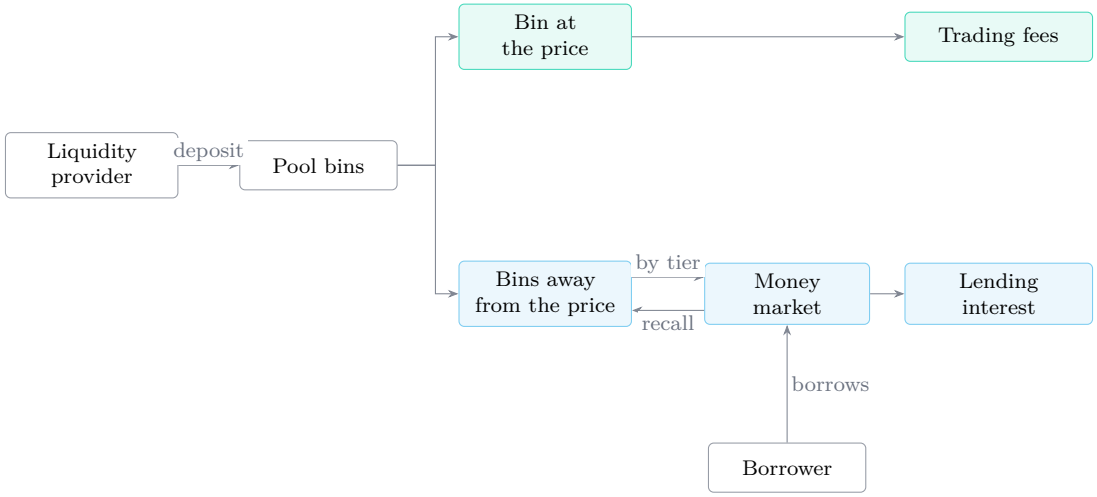

Figure 1: A single deposit is trading inventory and lending supply at the same time, and both returns accrue to the same position. The bin holding the current price earns trading fees. Bins away from it are lent to the money market in a proportion set by their distance, at a rate that rises with how likely they are to be recalled, and are called back as the price approaches. No capital leaves the pool at any point in this diagram.

Juncta is a discretized liquidity market maker and a money market sharing one pool of capital and one accounting layer. Reserves in bins far from the active price are lent; reserves near it are not; and what a borrower pays reflects how likely the capital behind them is to be called back. The remainder of this paper specifies the mechanism and the conditions under which it is solvent.

1.1 Contributions

1. **A deployment model priced by distance.** Bins are assigned to tiers by their distance from the active price. The fraction lent rises with distance and the premium paid by borrowers falls, because the two price opposite sides of the same fact. This is a term structure over distance rather than over time (Sections 6 and 8).

2. **A sharded global aggregate.** Supply and utilization are maintained per token across every pool holding it, without serializing the pools behind a single account. Risk gates read the aggregate conservatively, so that replication lag can close a door early but never

open one (Section 7).

3. **A collateral capacity rule with two proofs.** Capacity is restricted to the undeployed portion of a position. We show that total claims never exceed total supply, and that the seizable portion is by construction the portion that was never lent, so a seizure invokes no recall and is subject to no withdrawal band (Section 9).

4. **Progressive liquidation for slow-priced collateral.** Five levels, each closing only the minimum needed to restore the target health factor, with recovery windows between them. The schedule is built so that collateral valued on daily feeds, including tokenized real-world assets, is not closed out by a brief dip or a lagging oracle (Section 13).

1.2 What this paper does not cover

The protocol's account layouts, instruction signatures, error codes and test requirements are specified separately. This paper states what holds and why; it does not reproduce the implementation that enforces it. Where a claim rests on a control rather than on the model, we say so and name the control.

2 Background

We assume familiarity with automated market makers and with pooled lending, and define here only what later sections depend on.

2.1 Discretized liquidity

A discretized liquidity market maker partitions the price axis into bins. Each bin covers a fixed proportional width and holds reserves of one or both assets of the pair. Within a bin, trading follows a constant sum rule at a single price, so a trade that stays inside one bin has no price impact at all. Impact arises only when a trade exhausts a bin and moves to the next.

This differs from a continuous concentrated liquidity curve in two ways that matter here. First, price impact is a step function rather than a smooth one. Second, and more important for what follows, a bin is an addressable object with its own reserves. A protocol can reason about, and act on, the capital in one bin without touching the rest. That addressability is what makes selective deployment possible.

Away from the active bin, every bin is single sided. Bins above the active price hold only the base asset; bins below hold only the quote asset. The active bin is the sole two-sided bin, and it is the only bin whose composition changes continuously as trades arrive.

2.2 Money market accounting

A pooled money market records supply as shares against a growing reserve, borrowings against those reserves, and a utilization ratio that drives the interest rate. Withdrawal is possible

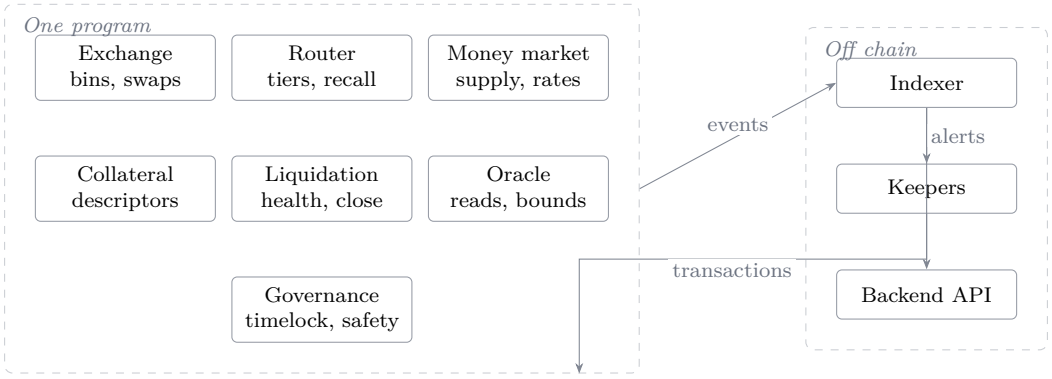

Figure 2: Seven on-chain modules in a single program, and three off-chain services. The off-chain side is not optional: liquidation and emergency recall have no in-protocol trigger, and the reconciliation checks that catch silent accounting drift run in the indexer.

only to the extent that reserves have not been lent. The reserve a protocol holds back against withdrawal, and the rate curve that discourages utilization from approaching one, are the two mechanisms that keep the market usable.

Juncta departs from this in one respect: there is no separate supply deposit. Providing liquidity to the exchange *is* supplying the money market. The consequence, stated here because everything in Section 8 follows from it, is that withdrawing liquidity inherits the money market's utilization.

2.3 Notation

Table 1 lists the symbols used in more than one section. A complete table appears in Appendix B.

Part I: The exchange

3 Bin mathematics

3.1 The geometric price grid

Bins are geometric. The ratio between one bin and the next is fixed by the pool's bin step s, expressed in basis points:

$$b = 1 + \frac{s}{10{,}000}, \qquad P(i) = b^i. \tag{1}$$

$P(i)$ is the *lower* bound of bin i. A bin covers the half-open interval $[P(i), P(i+1))$, so a price sitting exactly on a boundary belongs to the bin it opens and never to the bin it closes. Bin identifiers are signed 32-bit integers with no anchor offset, so bin 0 is price 1.0.

Symbol	Meaning
s	bin step, in basis points
b	geometric base of the price grid, $1 + s/10{,}000$
$P(i)$	lower bound of bin i
d_i	distance of bin i from the active bin
δ	fraction of pool reserves deployed to the money market
U	global utilization of a token's money market
CF	collateral factor
LT	liquidation threshold
h_k	the kth haircut, a multiplier in $(0, 1]$
V_{pos}	value of a liquidity position
V_{col}	collateral capacity of that position
HF	health factor

Table 1: Symbols used across sections.

Identifiers are bounded on both sides at $i_{\min} = -443{,}636$ and $i_{\max} = 443{,}636$. Bounding only the upper side leaves the lower side reachable at extreme prices, where b^i underflows to zero and a bin prices at nothing. Both bounds are checked on every conversion.

3.2 Recovering a bin from a price

The inverse of Eq. (1) is

$$i = \left\lfloor \frac{\ln P}{\ln b} \right\rfloor, \tag{2}$$

computed in fixed point rather than in floating point. Logarithm and exponential round independently, and a bin boundary is exactly where that matters, so the result of Eq. (2) is corrected against Eq. (1) in both directions before it is returned: downward if the recovered bin's lower bound exceeds the price, upward if the next bin's lower bound does not.

Invariant 3.1. For every valid price P and bin step, the returned identifier i satisfies $P(i) \le P < P(i+1)$.

3.3 Constant sum within a bin

A bin trades at a single price until it is empty. This is the substantive difference from a constant product curve: price impact is a step function rather than a smooth one, and inside a single bin there is no impact at all. For an input Δ_{in} against bin i,

$$\Delta_{\text{out}} = \Delta_{\text{in}} \cdot \bar{P}_i, \qquad \bar{P}_i = \frac{P(i) + P(i+1)}{2}. \tag{3}$$

The midpoint is used rather than the lower bound because liquidity sits across the whole width

of the bin and not at its edge. Using the lower bound would systematically misprice every fill in one direction.

3.4 The direction of rounding is a safety property

Every protocol amount is computed through checked fixed-point helpers, and every rounding is chosen rather than inherited from the arithmetic.

Invariant 3.2. Every rounding in the protocol rounds against the acting party and in favour of the pool. Amounts out round down, amounts owed round up, shares minted round down, shares burned round up.

The reason this is stated as an invariant rather than left to convention is that a single call site rounding the other way is not a visible bug. It is a slow drain, indistinguishable from ordinary rounding noise until an accumulated deficit surfaces somewhere else entirely.

4 Pools and positions

4.1 Pool structure

A pool is identified by its asset pair in canonical byte order, together with the fee preset it was created against. Ordering is decided on the raw bytes of the two asset identifiers and a mis-ordered pair is rejected rather than silently corrected, so that a pair maps to exactly one pool per preset. A pool is created permissionlessly, but only against a fee and bin step combination that governance has whitelisted as a pair, never as two independently chosen values.

A pool account does not hold bin data. Bins live in separate fixed-size arrays, so a swap loads and locks only the arrays it actually crosses rather than the whole pool. On a chain where two transactions writing the same account cannot run concurrently, this is what allows independent trades in different regions of the same pool to proceed in parallel.

4.2 A position is a vector of per-bin shares

A position is not a single claim on a pool. It is a set of claims, one per bin in its range, and shares are minted per bin rather than per position. For a deposit of a_i into bin i holding liquidity L_i against S_i outstanding shares,

$$\text{shares}_i = \begin{cases} a_i & \text{if } L_i = 0, \\ \left\lfloor \dfrac{a_i \, S_i}{L_i} \right\rfloor & \text{otherwise.} \end{cases} \tag{4}$$

The per-bin construction is not merely a bookkeeping convenience. Away from the active bin, every bin is single sided: bins above the active price hold only the base asset and bins below hold only the quote asset. A deposit that funds one side while the share calculation credits a notional two-sided value would let the depositor redeem a side they never funded.

Invariant 4.1. Shares for a bin are computed against that bin's own asset, determined by the bin's position relative to the active bin. The active bin is the only two-sided bin, and its shares are computed against the value of both legs at the bin's single price, which is exact because a bin has one price.

Liquidity is distributed across a range by one of three shapes: uniform across the range, concentrated at the current price, or concentrated at the edges. The choice is the provider's and affects only the weights a_i.

4.3 Withdrawal

Removing liquidity settles accrued income before any share is burned. Fees and lending yield accumulate in checkpoint fields that are separate from bin reserves, and burning shares without first settling those checkpoints takes the income with them. This failure is invisible to the party it happens to until somebody reconciles independently, which is why the settled amount is reported in the withdrawal event rather than only being credited.

Two further conditions apply, and both are established later. A position pledged as collateral cannot be withdrawn at all, including partially, because a partial withdrawal from a locked position is a full withdrawal in instalments (Section 9). And because provided liquidity is also lending supply, a withdrawal is split against current utilization rather than paid in full on demand (Section 8).

5 Swap execution

5.1 The bin walk

A swap consumes bins outward from the active bin, in the direction the trade pushes the price. Selling the base asset moves the price down and consumes bins below the active bin; selling the quote asset moves it up. Within each bin the fill follows Eq. (3), and when a bin is exhausted the walk steps to the next.

Direction is worth stating explicitly because reversing it produces the one pricing error that never looks wrong on screen. A large sell executed in the wrong direction fills at prices above spot, and the resulting number is simply favourable. The property that catches it is that price impact must be non-negative for any size in either direction.

5.2 Available liquidity is not bin liquidity

This is the check that makes the lending design safe, and it is the first place the money market intrudes on the exchange.

A bin's recorded reserve includes capital that has been lent to a borrower and is not physically present. Paying a swap from the nominal reserve pays out money the pool does not hold, and the shortfall surfaces later as an unexplained vault deficit rather than as a failed trade. The

quantity a swap may take is therefore

$$\text{available}_X(i) = \text{reserve}_X(i) - \text{borrowed}_X(i), \tag{5}$$

computed strictly per side. The two sides of a bin are independent balances, and capping a quote-side payout against base-side deployment state is a straightforward way to get Eq. (5) wrong in a direction that is hard to detect.

Invariant 5.1. A swap that exhausts available liquidity fills what it can and reports the shortfall. It does not revert.

Invariant 5.1 is a deliberate choice about failure mode. Reverting would convert a localized liquidity shortage into a denial of service on the entire pool, which is a strictly worse outcome for every participant including the one whose trade is short filled.

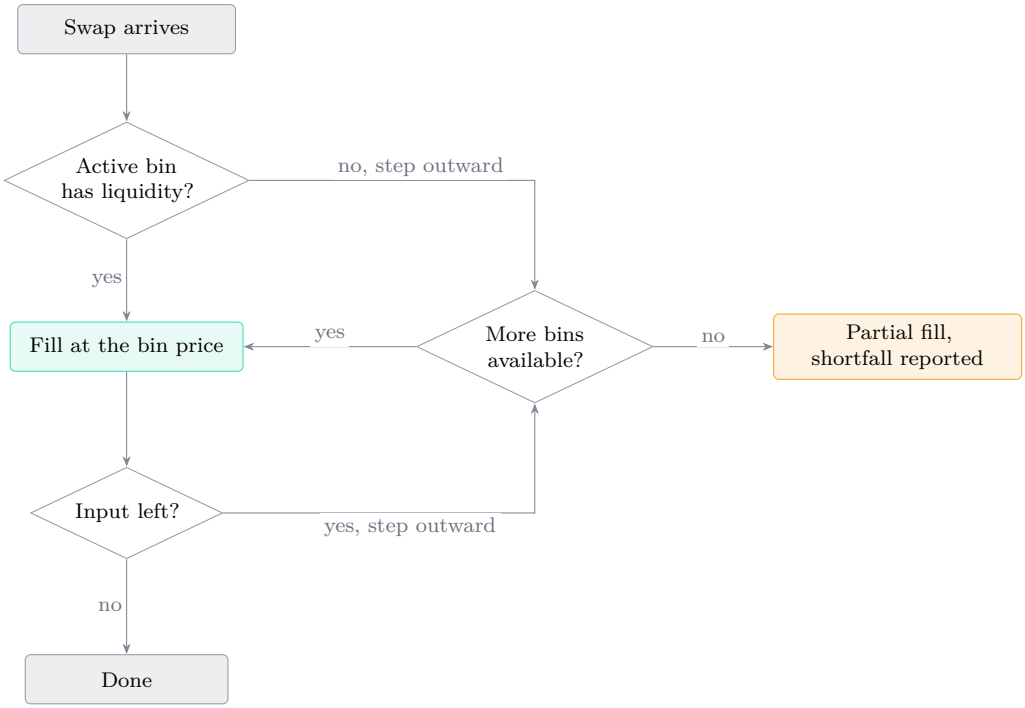

Figure 3: A swap consumes bins outward from the active bin. The test at each bin is availability, which is reserve less the amount lent out (Eq. (5)), not the recorded reserve. Exhausting availability produces a partial fill with the shortfall reported, never a revert.

5.3 Transfers are measured, not assumed

The pool fee is taken from the output. For assets that levy a transfer fee of their own, the amount that arrives differs from the amount sent, so every credit is computed from balance deltas measured across the transfer rather than from the requested amount. Any downstream accounting that uses the requested amount for such an asset is wrong by the transfer fee, every time.

5.4 Multi-hop routes

A swap carries between one and three hops. Beyond three, the accounts required by the hops and their bin arrays exceed what a single instruction can load.

Route continuity is proved before any hop executes rather than discovered partway through. A route whose second hop begins from an asset the first hop never produced will either fail after the first hop has already executed or, worse, succeed against a stale balance. Routes that revisit an asset are also rejected, since a cycle can only lose to fees.

5.5 The bin ceiling

Each bin array a swap crosses is an account it must load, and both the account count and the compute budget of a single instruction are bounded. A swap that would cross too many bins therefore cannot execute at all.

The ceiling is enforced across the whole route rather than per hop, because the constraint concerns what one instruction loads and a three-hop route loads all of its arrays at once. The precise value is a measured quantity rather than a derived one, and is set as a governed parameter against compute measurements taken under load.

One subtlety affects how the count is taken. An empty bin that the price moved through is still a bin the instruction had to load, so the count is the span from the active bin to the furthest bin touched, inclusive. Counting only bins that held something understates the account cost and admits routes that cannot execute.

Part II: The unified capital layer

6 Idle reserves as lending supply

The router decides which bins are lent and when the capital is called back. It runs inside the swap instruction, and only when the active bin changes. It touches nothing outside the pool it belongs to, because writing to a shared cross-pool account on every price move would serialize every pool trading a given asset behind a single lock. Aggregation across pools is handled separately, in Section 7.

6.1 Deployment tiers

Let $d_i = |i - i_{\text{active}}|$ be the distance of bin i from the active bin. Deployment is a function of distance alone.

Deployment rises with distance while the premium falls, because the two price opposite sides of the same fact. A bin the price can reach in three steps is likely to have its capital called back soon, so little of it is lent and whoever borrows it pays most for the privilege. A bin

Tier	Distance	Deployed	Premium	Interpretation
0	0 to 2	0%	n/a	Hard buffer, never lent
1	3 to 5	20%	+150 bps	Recall likely, borrowers pay most
2	6 to 10	50%	+75 bps	Recall plausible within a day of movement
3	11 to 20	70%	+25 bps	Reaching here takes a sustained move
4	21 and beyond	80%	base	A fifth stays behind as instant reserve

Table 2: Deployment tiers by distance from the active bin.

twenty steps away is unlikely to be reached, so most of it is lent at the base rate.

This is a term structure over distance from the current price rather than over time. It is the protocol's most distinctive economic object, and Section 8 gives the rate that results.

Tier 4 lends 80 percent and never more. The remaining fifth is the instant reserve, and it is the same fifth that makes an ordinary withdrawal immediate below 80 percent utilization. These are one parameter with two names and cannot be tuned independently without breaking the withdrawal guarantee.

6.2 Buffer bins

Bins within a defined distance of the active bin are never lent under any circumstances. They are the reserve that lets a sudden multi-bin move execute without recalling capital in the middle of a transaction. The width is set by the pair's volatility class:

$$\text{bufferBins} = \left\lceil \frac{\text{safeDistance}_{\text{bps}}}{s} \right\rceil, \tag{6}$$

with safe distances of 50 basis points for stable pairs, 200 for blue chip, 300 for mid cap and 500 for volatile pairs. The division is a ceiling taken in integers. A floor would leave the last fraction of the safe distance unbuffered, and that fraction is the one nearest the price, which is the part that matters.

The buffer overrides the tier. A bin may be tier 4 by distance and still lend nothing because the pool's buffer reaches past it:

$$\text{deployment}(d) = \begin{cases} 0 & d \le \text{bufferBins}, \\ \text{tierDeployment}(d) & \text{otherwise.} \end{cases} \tag{7}$$

Figure 4 shows the resulting profile.

Invariant 6.1. A pool's buffer width is monotonically non-decreasing. A pool reclassified from volatile to stable keeps the wider buffer.

Narrowing a live pool's buffer would make already-lent bins buffer-eligible while capital is still outstanding against them, which is a state the rest of the design assumes cannot arise.

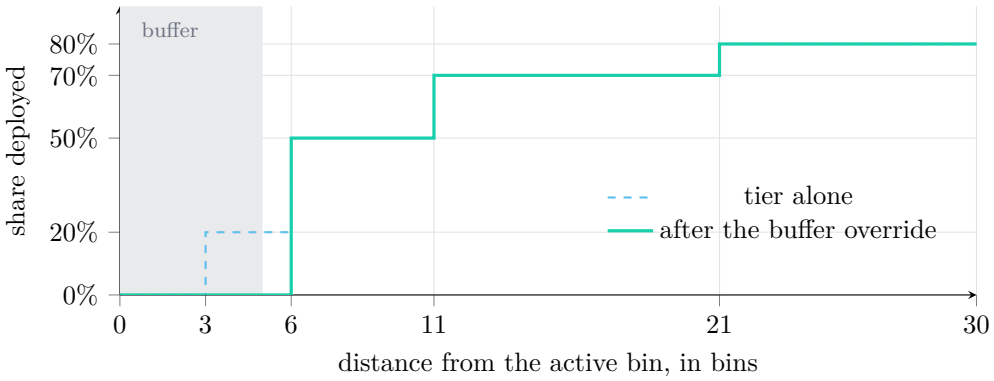

Figure 4: Share of a bin's reserves lent to the money market, as a function of distance from the active bin, for a pool with a five bin buffer. Tier 1 would deploy 20 percent at distances three to five, but the buffer of Eq. (6) overrides it to zero. The remaining 20 percent at tier 4 is the instant reserve of Section 8.

One consequence of Eq. (6) is reachable rather than theoretical. At a bin step of 1 basis point a blue chip buffer is 200 bins wide, which is 2 percent of price in each direction. A pool whose liquidity spans less than that has nothing outside its buffer and can never lend anything, whatever Table 2 says. Such a pool is legitimate and is permitted, but it is refused activation as a lending contributor, since a pool that structurally cannot lend should not be counted in a lending aggregate.

6.3 How much capital can actually earn

The tier ceiling and the buffer together bound the share of a pool's capital that can be earning lending interest at any moment:

$$\text{shareEarning} \;\le\; 0.8\left(1 - \frac{\text{bufferBins}}{\text{halfSpan}}\right). \tag{8}$$

A dollar-weighted figure above 80 percent is not reachable under this model, and a realistic pool lands between 40 and 70 percent. We state the bound because the comparison that matters survives it: a conventional concentrated liquidity venue earns on the active bin alone, which is a low single-digit share.

6.4 Deployment is tracked per side

A bin above the active price holds the base asset and a bin below holds the quote asset. When the price crosses a bin, that bin changes which asset it holds. Deployment state that is not tracked separately for each side drifts every time this happens, and the drift is silent.

Invariant 6.2. For each side of each bin: deployed $\ge$ borrowed, reserve $\ge$ borrowed, and pending recall $\le$ borrowed.

These are asserted on every mutation. Because drift of this kind is silent by nature, they are

also recomputed independently from bin state and compared against the router's own totals each epoch, so that a divergence is caught by comparison rather than by assertion alone.

6.5 Recall

Recall returns lent capital to a bin. There are three kinds.

Scheduled recall fires when the active bin moves and a bin enters the buffer zone. It is atomic, happens inside the swap, is scoped to the pool, and requires no external actor.

Withdrawal recall fires when a provider removes liquidity. If the instant reserve covers the request it is paid immediately; otherwise the remainder is queued and serviced as the backing bins are repaid.

Emergency recall fires when utilization crosses 90 percent. It is triggered externally, freezes new borrows of the affected asset, and pulls from the safest bins inward under a per-call bound.

Invariant 6.3. Recall never aborts. It takes back what is available and records the remainder.

What is available is the deployed amount less the borrowed amount: capital that was deployed but that no borrower has drawn. The remainder becomes a pending recall recorded against that specific bin, and it returns when the borrower who drew it repays. A recall that could fail would turn a routine price move into a failed swap, which is why Invariant 6.3 is stated as a property of the mechanism rather than as an error path.

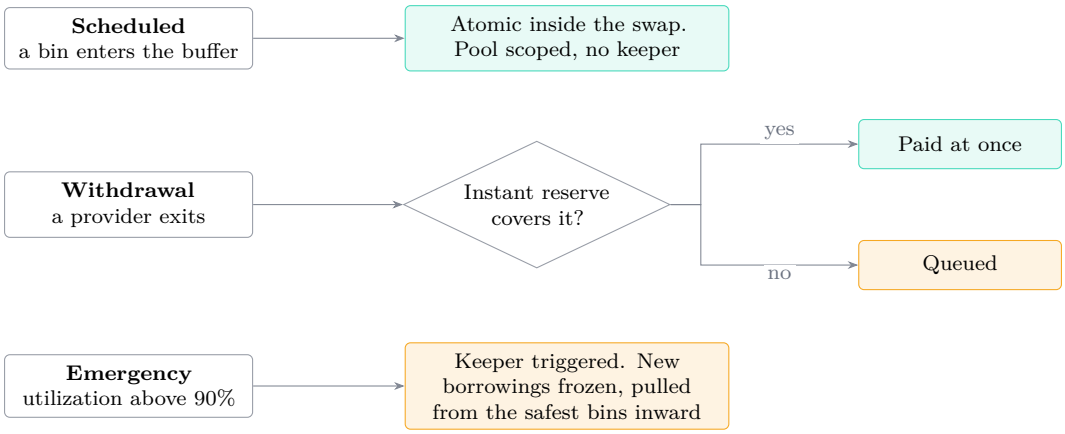

Figure 5: Recall never aborts (Invariant 6.3). It returns what no borrower has drawn and records the remainder against the specific bin, which is released when the borrower who drew it repays.

6.6 Cross-pool liquidity is not fully fungible

Supply and utilization are genuinely global across every pool holding an asset, and Section 7 shows how. Recall clearing is not. A repayment clears the pending recall on the bins the repaying position actually drew from, so one provider's capital returns because their particular borrower repaid while another's does not, though the protocol is receiving the same asset in volume from elsewhere.

These are different properties and conflating them is a real error. Describing the protocol as having one pool of liquidity is accurate about supply and inaccurate about recall, so interfaces must not present a global queue position to a waiting provider: under bin-scoped clearing there is no global queue, and a position in one is not a fact.

Pool-scoped clearing, ordered by urgency rather than by age, removes most of this unfairness at almost no cost, since the pool account is already write locked by the repayment. An asset-global queue folded asynchronously removes the rest. Both are specified; neither is required for the properties proved in Section 9.

7 The global aggregate

7.1 One market per asset

A borrower should face one rate and one depth for a given asset, not a separate market for every pool that happens to hold it. Liquidity fragmented across pools is the condition the protocol exists to remove, and reintroducing it at the lending layer would forfeit most of the benefit.

The obvious implementation is one account per asset, shared by every pool routing it and written by every swap that changes deployment. On a chain where two transactions writing the same account cannot execute concurrently, this makes every pool trading that asset serialize against every other. The protocol's throughput ceiling would then be set by its most popular asset rather than by the sum of its pools, and the more successful an asset became the worse its pools would perform.

7.2 Sharding the aggregate

Instead of one account per asset we keep N accounts, and each pool writes to the shard chosen deterministically from its own address:

$$\text{shard}(p) = \text{hash}(p) \bmod N, \qquad N = 16. \tag{9}$$

Write contention falls by a factor of N. A read loads all N shards and sums them. The trade is deliberate and it runs in the favourable direction: write contention is expensive and read cost is cheap, and the instructions that need a global figure are exactly the ones that can afford sixteen extra account loads. Borrow, repay, withdraw and liquidate need it. A swap never does.

Supply, borrowings and utilization are therefore real-time. There is no periodic job standing between a change and the figures that gate risk.

The proximity premium of Section 8 is a weighted mean, so it cannot be summed directly. It is sharded as a numerator and a denominator separately and divided at read time, which makes the aggregate exact rather than an approximation of a mean of means.

Invariant 7.1. A shard that has not been written within its staleness bound is treated as unknown and the read fails, unless the shard is empty.

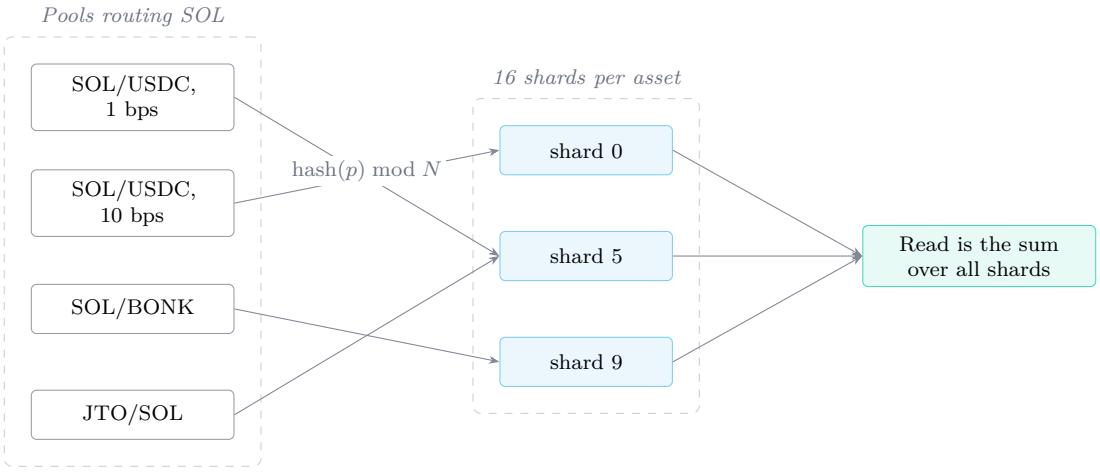

Figure 6: Each pool writes to one shard chosen from its own address, so pools trading the same asset do not contend for a single account. Reads load every shard and sum them, which is affordable because only borrow, repay, withdraw and liquidate need the aggregate. A swap never does.

Invariant 7.1 is stated because the alternative reading is available and wrong. A shard nobody has written recently carries no information, and treating its recorded total as current is equivalent to asserting that nothing has happened in the pools that write to it. Silence is not good news.

7.3 What remains asynchronous

Sharding removes the periodic job from the risk path but does not eliminate it. What remains is reconciliation and accrual, where seconds of lag genuinely do not matter: folding accumulated per-pool deltas into shard totals, advancing the yield index, and matching pending recalls.

Two refinements bound the worst case. A pool whose accumulated delta crosses a threshold signals for immediate folding rather than waiting for the next tick, which bounds lag for the deltas that matter without raising the base frequency for those that do not. And deltas that block a recall fold on a faster path than yield accrual, since the two have different urgency and should not share a schedule.

7.4 Every risk gate reads conservatively

Whatever lag remains must fail in the direction that restricts.

Invariant 7.2. A risk gate counts pending changes in the direction that restricts and never in the direction that permits. Pending borrowings are counted as already drawn; pending withdrawals are counted as already paid; supply from a pool that has not completed its seasoning period is not counted at all.

The consequence is the property we want from replication lag: staleness can close a door early, and it can never open one. A gate computed this way may refuse a borrow that a perfectly

current view would have allowed, which is a cost paid in occasional inconvenience. The opposite error is paid in solvency.

8 Interest rates and withdrawal bands

8.1 The rate model

The base borrow rate is a two-slope function of global utilization U with a kink at U^*:

$$r_{\text{borrow}}(U) = \begin{cases} r_0 + \dfrac{U}{U^*}\,(r_1 - r_0) & U \le U^*, \\ r_1 + \dfrac{U - U^*}{1 - U^*}\,(r_2 - r_1) & U > U^*, \end{cases} \tag{10}$$

with a hard cap on *new* borrowings at $U = 0.80$. Existing borrowings are unaffected by the cap, which restricts the creation of new obligations rather than the servicing of old ones.

What a particular borrower pays adds the proximity premium of the capital standing behind them:

$$r_{\text{paid}} = r_{\text{borrow}}(U) + \bar{r}_{\text{prox}}, \qquad \bar{r}_{\text{prox}} = \frac{\sum_{\text{shards}} \text{numerator}}{\sum_{\text{shards}} \text{denominator}}. \tag{11}$$

This is the term structure of Section 6 expressed as a price. Capital that may be recalled soon is paid more for the risk of being recalled, and the premium is an average across all the capital actually deployed rather than a property of any one lender's position.

A weighted mean with no per-contributor cap is manipulable: a sufficiently large contributor moves the protocol-wide rate by choosing where its bins sit. Each pool's weight in both the numerator and the denominator is therefore capped at the same share limit that bounds its contribution to supply, applied when the delta is folded rather than when it is recorded.

The supplier rate is the borrow rate net of utilization and the reserve factor:

$$r_{\text{supply}} = r_{\text{paid}} \cdot U \cdot (1 - \text{reserveFactor}). \tag{12}$$

8.2 Withdrawal bands

Because provided liquidity is the lending supply, withdrawing liquidity is gated by utilization. This is the single most important property for a provider to understand before depositing rather than after.

The middle band tapers linearly rather than stepping. Without the taper, one basis point of utilization would convert a fully immediate withdrawal into a fully queued one, and a provider watching the figure approach 80 percent would have every reason to withdraw pre-emptively. That behaviour is precisely what pushes utilization the rest of the way, so the cliff would manufacture the condition it was meant to signal.

Two properties of the band structure are worth stating plainly.

Global utilization	Withdrawal
Below 80%	Immediate, from the instant reserve
80% to 90%	Partly immediate, remainder queued, tapering across the band
Above 90%	Queued, released as the specific bins backing it are repaid

Table 3: Withdrawal availability by global utilization.

The instant reserve and the tier 4 deployment cap are the same 20 percent. They are one parameter with two names. Raising the deployment cap without lowering the instant band breaks the guarantee that a withdrawal below 80 percent utilization is immediate, and the two must move together or not at all.

A queued withdrawal is not a frozen position. It continues to accrue trading fees and lending interest until it is released. The distinction matters to the person waiting, for whom the capital being stuck and the capital still working and arriving later are very different situations.

Part III: Collateral and solvency

9 A dollar cannot back two claims

Sections 6 and 7 described how reserves in inactive bins are lent out. This section addresses what happens when the owner of those reserves also wants to borrow against them. The answer is not a valuation adjustment. It is a constraint on capacity, and it is the constraint the rest of the protocol is built around.

9.1 The arithmetic

Consider a position that supplies $1,000 of liquidity to a pool on deployment tier 4, which permits up to 80 percent of pool reserves to be lent to the money market. Suppose the position is also accepted as collateral at a collateral factor of 60 percent. Three things are then claimed to happen at once: the position earns swap fees, its deployed reserves earn lending interest, and it backs a borrow.

Supply contributed	$1,000
Deployed to the money market and drawn by a borrower	$800
Borrowed by the position owner against the same position	$600
Total claims against $1,000 of supply	**$1,400**

The same dollars are lent to an unrelated borrower and pledged against the owner's own borrow. No haircut corrects this. A haircut reduces the value assigned to collateral; it does not reduce

the number of claims on a dollar. The error is in the supply accounting, not in the valuation.

9.2 Seizure cannot take what is not there

The second face of the same problem appears at liquidation. Seizing a liquidity position means withdrawing its token balances from the pool vaults. If 80 percent of those balances have been lent out, the seizure must first recall them, and recall is governed by the withdrawal bands of Section 8. Above the utilization band ceiling, a recall queues.

A liquidation that queues is not a liquidation. The borrower's health continues to deteriorate while the seizure waits, and the protocol carries the difference. Pricing the expected delay as an additional haircut does not help: it makes the collateral worth less on paper while leaving the seizure just as impossible.

9.3 The capacity rule

Both failures resolve under a single restriction.

Definition 9.1 (Collateral capacity)**.** Let V_{pos} be the value of a liquidity position, $\delta \in [0, \delta_{\max}]$ the fraction of pool reserves deployed to the money market under the pool's tier, $\text{CF} \in (0, 1]$ the collateral factor for the position's asset pair, and $h_1, \ldots, h_m \in (0, 1]$ the applicable haircuts. The collateral capacity of the position is

$$V_{\text{col}} \;=\; V_{\text{pos}}\,(1-\delta)\,\text{CF}\prod_{k=1}^{m} h_k. \tag{13}$$

The factor $(1-\delta)$ is the portion of the position held in buffer bins and the instant reserve. By the construction of Section 6, that portion is never lent.

Proposition 9.1 (No double claim)**.** *Let a pool hold total supply V and deploy a fraction δ of it. Let B_{mm} be the outstanding money market borrowings drawn from that pool and B_{col} the total borrowed against positions in the pool pledged as collateral under Definition 9.1. Then $B_{\text{mm}} + B_{\text{col}} \le V$.*

Proof. Money market borrowings are drawn only from deployed reserves, so $B_{\text{mm}} \le \delta V$. Borrowings against collateral are bounded by total capacity. Writing CF_p and $h_{p,k}$ for the factors applying to position p, and using $\text{CF}_p \le 1$ and $h_{p,k} \le 1$,

$$B_{\text{col}} \;\le\; \sum_{p} V_{\text{pos}}(p)\,(1-\delta)\,\text{CF}_p\prod_{k} h_{p,k} \;\le\; (1-\delta)\sum_{p} V_{\text{pos}}(p) \;=\; (1-\delta)\,V.$$

Adding the two bounds gives $B_{\text{mm}} + B_{\text{col}} \le \delta V + (1-\delta)V = V$. □

Proposition 9.2 (Immediate seizability)**.** *Any amount seizable from a position under Definition 9.1 is withdrawable without invoking recall.*

Proof. By Eq. (13) the claim against a position is bounded by $(1-\delta)$ of its value, which is the buffer and instant reserve portion. The deployment tiers of Section 6 never lend that portion, so no part of it is outstanding with a money market borrower. Withdrawal bands gate the recall of deployed reserves only. A seizure confined to the undeployed portion therefore invokes no recall and is subject to no band. □

Corollary 9.3. *Haircuts on liquidity collateral price valuation uncertainty alone. They do not price withdrawal delay, because under Proposition 9.2 there is none.*

Figure 7 shows the two accountings side by side.

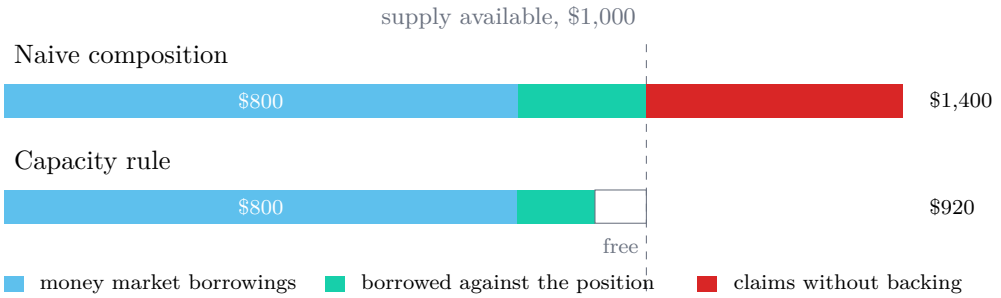

Figure 7: Claims against $1,000 of pool supply at 80 percent deployment and a 60 percent collateral factor. Applying the collateral factor to the full position value claims $1,400 against $1,000 of supply. Applying it to the undeployed portion, as Eq. (13) requires, claims $920 and leaves the remainder unencumbered.

The three properties the rule is meant to deliver hold together rather than in tension:

Property	**Why it holds**
Solvency	Each dollar backs exactly one claim (Proposition 9.1)
Liquidation is always immediate	The seizable portion was never lent (Proposition 9.2)
All three yields coexist	Fees throughout, interest on the deployed part, a borrow against the rest

9.4 Choosing the split

Capacity falls as deployment rises. A blue chip pool running at 88 percent deployment offers 12 percent of position value before the collateral factor is applied, which for many borrowers is too small to be worth the transaction. This is the honest price of the invariant, and it is paid in capacity rather than in solvency.

It is also a choice the provider can make for themselves. A position may declare a fraction of itself collateral eligible, and the bins covering that fraction are forced to tier 0 for as long as the position is locked. They are never deployed, therefore never lent, therefore always seizable, and Proposition 9.2 continues to hold over them by the same argument. The provider gives up lending interest on that portion and receives borrowing capacity against it.

Two conditions bound the choice. The declared fraction cannot fall below the instant reserve plus the position's share of the buffer, since that portion was never going to be deployed in any case. And a request that would reduce the eligible fraction below what current debt requires is refused rather than clamped to the minimum, because silently granting less than was asked for is how a position ends up undercollateralized without anyone having made a decision.

Invariant 9.1. For every pool, the eligible collateral declared across its positions plus the borrowings attributable to that pool never exceed its reserves:

$$\sum_{p} \text{eligible}_p \cdot V_{\text{pos}}(p) \;+\; B_{\text{attr}} \;\le\; R_{\text{pool}}.$$

This is the per-pool form of Proposition 9.1, and it is asserted in execution rather than left as a consequence, because its violation produces no symptom at the moment it occurs.

9.5 A locked position cannot be withdrawn

Withdrawal from a position pledged as collateral fails, including partial withdrawal. The restriction has to cover the partial case, because a partial withdrawal from a locked position is a full withdrawal performed in instalments, and a rule that permits it in small enough pieces does not exist.

The lock is written when the borrow is taken and lifted on repayment. The position does not move to an escrow account in the meantime; it stays where it is and continues to earn trading fees, since relocating it would break the per-bin share accounting of Eq. (4) for no gain in safety.

10 Valuing a liquidity position

10.1 A claim on two amounts, not an asset with a price

This framing is the principal defence in this part of the paper. Most published failures of liquidity positions used as collateral come from pricing the position rather than its contents.

A position is valued as the sum of its reserves at independent prices:

$$V_{\text{pos}} = \sum_{i=\text{lower}}^{\text{upper}} \left(R_{X,i}\, P_X + R_{Y,i}\, P_Y\right), \tag{14}$$

where P_X and P_Y are governance-approved oracle prices obtained independently of the pool.

The pool's own price never enters Eq. (14). It determines only how the position's reserves divide between the two assets, and that division changes only through real trades that pay real value into the pool. An attacker who moves the pool price therefore changes the composition of the position and not its valuation, and the value they have to supply to move it is the value they transfer to the position's owner.

10.2 Never price an asset from the pool it lives in

Invariant 10.1. An asset with no approved independent oracle is valued at zero for collateral purposes. There is no pool-derived fallback and no estimate, including for display.

Invariant 10.1 is what defeats the fabricated pool attack. An attacker who mints an arbitrary quantity of a new asset, creates a pool for it, and deposits holds a position containing a great deal of that asset and nothing else of value. With no approved oracle, the fabricated leg values at zero and the position values at its other leg alone, which is exactly what the attacker actually deposited. They can borrow against what they contributed and nothing more.

Concentrated liquidity makes spot manipulation cheaper than constant product rather than dearer, because less capital is required to move the price a given distance. This is why a pool-derived price is prohibited here rather than discouraged.

10.3 Accrued income is excluded

Trading fees and lending interest accumulate in checkpoint fields and never increase the bin reserves that enter Eq. (14). The collateral is the reserves alone.

The reason is an attack rather than an accounting preference. Accrued income is claimable at any moment. If it counted toward collateral, a borrower could shrink their effective collateral at will by harvesting between health checks, and the protocol would register no change at the moment the collateral fell.

10.4 Requirements on the price source

Requirement	Setting	Reason
Smoothing	TWAP or EWMA, never spot	Manipulation must be sustained
Minimum window	30 minutes for collateral	A short window is one trade
Confidence	Reject beyond the asset's bound	A wide interval is a refusal to quote
Staleness	Reject past the age bound	Fail closed
Second source	Required for every collateral asset	Divergence pauses it

Table 4: Price source requirements for collateral valuation.

Divergence between two sources does not select a winner. When two sources disagree beyond the configured threshold, the honest position is that the price is unknown, and collateral operations for that asset pause until they agree again.

10.5 Haircuts and factors

Haircuts apply multiplicatively after the collateral factor, and each prices a named risk.

Single-asset collateral carries one risk, which is price. Liquidity collateral carries three: price, divergence loss eroding value independently of price, and an unwind step at liquidation. The

Haircut	What it prices	Range
Concentration	A narrow range loses value faster when price leaves it	5 to 25%
Utilization	High utilization delays any recall	0 to 15%
Price impact	Unwinding moves the price against the liquidator	2 to 20%
Withdrawal	Residual queue risk after Proposition 9.2	0 to 10%
Oracle quality	Feed confidence and update frequency	0 to 10%

Table 5: Haircuts applied to liquidity collateral.

collateral factors reflect that difference.

Pair type	Collateral factor	Liquidation threshold
Stable against stable	75%	83%
Blue chip against stable	60%	68%
Mid cap against stable	45%	53%
Volatile pair	35%	43%

Table 6: Collateral factors and liquidation thresholds by pair type.

11 Eligibility and isolation

Sections 9 and 10 establish how much a position can support and how it is valued. This section covers which pools are eligible at all, and how the consequences of getting that judgement wrong are contained.

11.1 Liquidation must be possible before the borrow is allowed

A borrow that cannot be unwound is worse than a borrow refused, because the protocol discovers the problem only at the moment it needs the collateral. The unwind is therefore simulated when the borrow is taken, and the borrow is rejected if the simulation does not complete.

Three conditions cause it to fail: insufficient seizable value at the liquidation threshold, an unwind whose price impact exceeds the haircut that was supposed to price it, or an asset-level transfer restriction that would block the seizure. The first is a sizing question, the second says the haircut is wrong for this position, and the third says the asset should not have been approved.

11.2 Two-stage approval

Eligibility is granted per pool, in two stages.

The first stage produces a validation record. The pool address, both assets and the owning program are read from chain state rather than from the arguments of the approving instruction. Both assets must have approved oracles meeting Table 4. Pool depth must exceed a floor, sustained over a window rather than sampled once. Pool age must exceed a minimum. Asset-level transfer extensions are enumerated and checked against policy. Unwind feasibility is demonstrated at the configured maximum position size, within the price-impact haircut.

The second stage is a governance action on the critical timelock path, and it commits to the hash of that validation record. Governance therefore approves exactly what was reviewed, and any later change to the reviewed material invalidates the approval rather than being silently inherited by it.

The resulting descriptor holds every risk parameter for that pool in one place: factors, thresholds, the five haircuts, depth floors, position and debt ceilings, the isolation bucket, and the two hashes that make the approval verifiable after the fact. Nothing in it is ever taken from an instruction argument.

Revocation blocks new borrowings while preserving repayment and liquidation. A revocation that stopped repayment would convert a risk decision into a forced default.

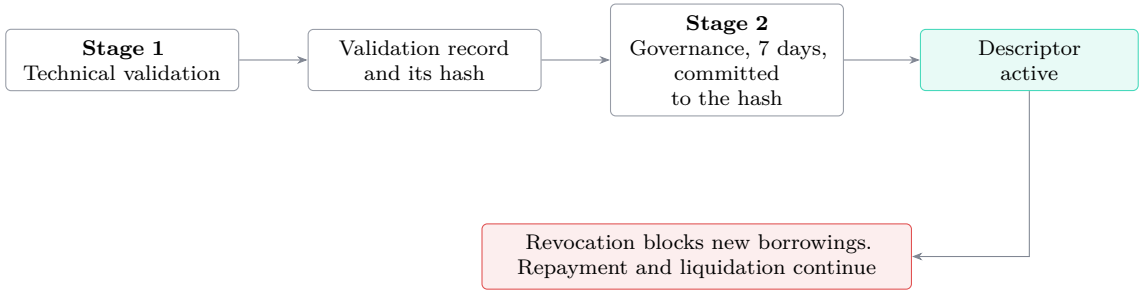

Figure 8: Governance approves the exact material that was reviewed, because the approving operation commits to its hash. A later change to any of it invalidates the approval instead of being inherited by it.

11.3 Isolation

Each approved pool has its own loss bucket with independent debt and loss caps. A loss in a bucket is absorbed by that bucket's own insurance reserve first, it cannot consume the liquidity of suppliers outside the bucket, and it cannot reach the protocol backstop unless governance explicitly extends coverage.

This is what makes the fabricated pool question survivable even if every other control fails. An attacker who manufactures bad debt manufactures it inside a bucket that touches nobody else.

A bucket is a loss-absorption boundary and not a supply boundary. It decides which insurance reserve pays and how far a loss may travel. It says nothing about where an isolated pool's capital is lent, which folds into the same aggregate as any other pool's. The two readings are easy to conflate because isolation suggests segregation, and the only thing segregated here is a loss.

Caps are enforced at several levels: per position, per user across all liquidity collateral, per pool, and protocol-wide. The level that matters most is the last one, a cap per wallet across every collateral type. A cap checked per type is bypassed by splitting a position across types, and correlated wallets are monitored off chain because the program can only enforce what it can see.

11.4 Tokenized real-world assets

Positions in pools holding tokenized real-world assets are eligible under the same machinery, with additional requirements layered on top. The requirements are additive: none of the base conditions is relaxed, and two issuer powers disqualify an asset outright rather than being priced.

Beyond the general conditions, an RWA pool requires a recorded legal enforceability review of the relationship between the token and the asset, verification of custody and title, a documented redemption path with its duration, a full record of the issuer's powers, and a net asset value feed with its own staleness bound measured in days. The last point is not a relaxation. A staleness bound written for a market data feed marks every real-world asset permanently stale, so the bound has to be stated in the units the asset actually prices in.

Issuer power	**Effect on eligibility**
Clawback	Disqualifying
Force transfer	Disqualifying unless the protocol is the only destination
Freeze or blacklist	Collateral factor capped at 40%, with a liquidation path that survives a freeze
Pause transfers	Haircut proportional to the maximum documented pause
Supply change	Haircut, plus a monitor that pauses on an unexpected change

Table 7: Issuer powers and their treatment.

An asset whose issuer can unilaterally reclaim it is not collateral. It is an obligation of the issuer, and for this purpose it is worth nothing.

Redemption illiquidity is the residual risk and it is not solvable in code. On chain liquidity can disappear in minutes while redemption takes days. It is bounded instead: borrow caps set well below on-chain depth, a withdrawal queue isolated to the bucket, a state in which new borrowings stop and only repayment and liquidation remain once depth falls below a floor, and monitoring on depth, queue length and feed age.

Finally, a real-world asset position may not collateralize a borrowing of another real-world asset. Two illiquid legs with correlated redemption risk and independent issuer powers produce a position nobody can unwind under stress.

Part IV: Liquidation

12 Health

A position's health factor is the ratio of its threshold-weighted collateral to its debt:

$$\mathrm{HF} = \frac{\sum_j C_j \, \mathrm{LT}_j}{\sum_k D_k}, \tag{15}$$

where C_j is a collateral position's value after factors and haircuts, as computed in Section 10, and LT_j its liquidation threshold.

Two properties of Eq. (15) are worth stating because both have straightforward implementations that are wrong.

Invariant 12.1. A position holding collateral and no debt has an undefined health factor, not a health factor of zero.

This is a legitimate and common state, and it is the state of every position that has never borrowed. Returning zero for it, which follows naturally from a guard against division by zero that yields a default, marks every such position as maximally unhealthy and liquidates it.

Invariant 12.2. The health factor governing a liquidation is recomputed from live reserves and current oracle prices at the moment of liquidation, never read from a stored figure.

A stored health factor is stale by construction, and staleness is bidirectional: it is how a solvent position gets liquidated and how an insolvent one does not.

13 Progressive liquidation

13.1 Why a single threshold fails

The conventional design liquidates at one threshold and closes a fixed fraction of the debt, commonly half. It has three problems.

There is no warning. A position is healthy until the instant it is not, and the borrower's first indication is the liquidation itself.

The amount closed is a parameter rather than a computation. A position a few basis points below the threshold and one deeply underwater are treated identically, and the first is closed far beyond what restoring it requires.

And every eligible liquidation is a race. Liquidators compete on speed for the same opportunity, bidding up priority fees, and the cost of that competition is paid out of the borrower's collateral.

13.2 The level schedule

Liquidation proceeds in five levels, with the bonus rising as health falls.

Level	Health factor	Closes	Bonus	LP bonus
1	1.00 to 0.95	to the target	0.25 to 0.50%	0.50 to 0.75%
2	0.95 to 0.90	to the target	0.50 to 1.50%	0.75 to 2.00%
3	0.90 to 0.85	to the target	1.50 to 3.00%	2.00 to 4.00%
4	0.85 to 0.80	to the target	3.00 to 5.00%	4.00 to 6.50%
5	below 0.80	as required	6.00 to 8.00%	7.00 to 10.00%

Table 8: Liquidation levels, with bonuses for single-asset and liquidity collateral. The target health factor is 1.20.

Liquidity collateral carries the higher bonus because it requires an unwind step that single-asset collateral does not. Without the difference, liquidity positions are the liquidations keepers decline, and the liquidations keepers decline are the ones that become bad debt.

Recovery windows separate the levels. A position that has been partly closed at one level is not immediately eligible at the next, which gives the borrower an interval in which to add collateral or repay. The windows are what make the schedule progressive rather than merely graduated.

13.3 How much is closed

At each level other than the fifth, the liquidation closes the minimum needed to restore the position to a target health factor $T = 1.20$. That minimum has a closed form.

Proposition 13.1 (Required close fraction)**.** *Consider a position at health factor h with a single collateral type of liquidation threshold* LT*, liquidated at bonus b. The fraction of debt that must be closed to restore health to T is*

$$f = \frac{T - h}{T - (1 + b)\,\mathrm{LT}}, \tag{16}$$

and this is feasible, meaning $f \le 1$, if and only if $h \ge (1 + b)\,\mathrm{LT}$.

Proof. Let C be collateral value and D debt, so $h = C\,\mathrm{LT}/D$. A liquidator repaying Δd seizes collateral worth $\Delta d\,(1 + b)$. Requiring the resulting health factor to equal T gives

$$\frac{(C - \Delta d\,(1 + b))\mathrm{LT}}{D - \Delta d} = T.$$

Expanding and collecting Δd,

$$C\,\mathrm{LT} - TD = \Delta d\big[(1 + b)\mathrm{LT} - T\big], \qquad \text{so} \qquad \frac{\Delta d}{D} = \frac{T - C\,\mathrm{LT}/D}{T - (1 + b)\mathrm{LT}} = \frac{T - h}{T - (1 + b)\mathrm{LT}}.$$

The denominator is positive whenever $T > (1 + b)\mathrm{LT}$, which holds for every combination in Tables 6 and 8. Then $f \le 1$ reduces to $T - h \le T - (1 + b)\mathrm{LT}$, that is $h \ge (1 + b)\mathrm{LT}$. □

Proposition 13.1 has a consequence that shapes the schedule. Restoration to the target is not always possible. For collateral with a high liquidation threshold, the feasibility condition $h \geq (1+b)\mathrm{LT}$ binds while the position still has collateral remaining. A stable pair at $\mathrm{LT} = 0.83$ becomes infeasible below a health factor of roughly 0.834, which is inside level 4.

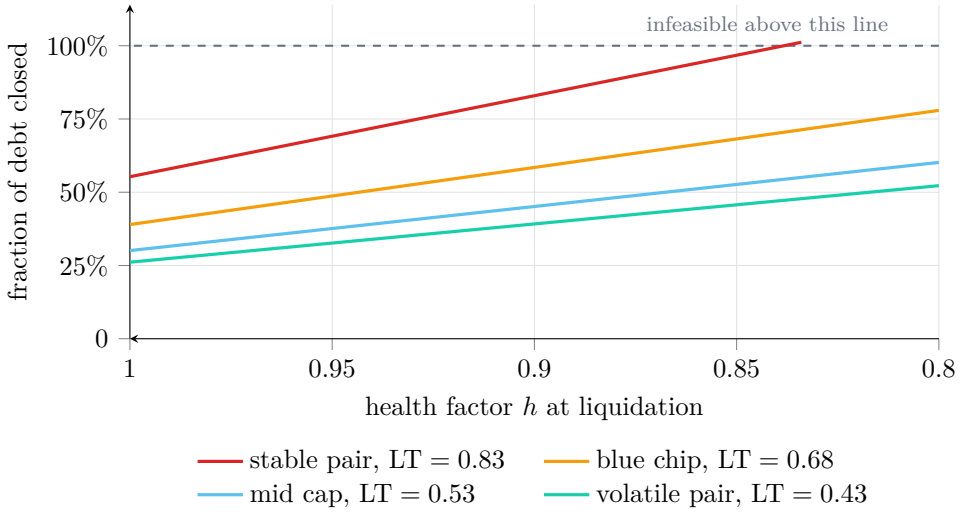

Figure 9: Fraction of debt that must be closed to restore a position to the target health factor of 1.20, from Eq. (16), at a liquidation bonus of one percent. Health deteriorates left to right. Collateral with a higher liquidation threshold requires a larger close at the same health factor and reaches infeasibility sooner: the stable pair curve terminates at $h = 0.834$, inside level 4, beyond which no partial close restores the target.

This is why level 5 closes as required rather than to a target. Below the feasibility bound, no partial close restores health to T, and the correct action is to close the position down rather than to compute a fraction that does not exist. Figure 9 shows where each collateral type crosses the bound.

Equation (16) also explains a counterintuitive property of the schedule: safer collateral requires a larger fraction of the debt to be closed at the same health factor. A high threshold means the collateral was permitted to support more debt per unit of value, so restoring a given ratio moves less per unit closed.

13.4 Collateral that prices slowly

The recovery windows exist for the general case, but their width is set by the hardest one. Collateral valued on a feed that updates daily cannot respond to a liquidation within a window measured in blocks, and a borrower holding it has no way to add collateral priced on the same feed in time to matter.

Two properties follow. Windows for positions holding slowly priced collateral are set against the update interval of the slowest feed in the position, not against a protocol-wide constant. And a health factor that has deteriorated because a feed has not updated is distinguishable from one that has deteriorated because value has fallen, since the first leaves the feed's age outside its bound and Table 4 already requires that case to fail closed rather than to price.

A brief dip in a market that reprices continuously and a stale reading on a feed that reprices weekly produce the same number in Eq. (15). They are not the same event, and a schedule that treats them identically liquidates real-world asset positions for the wrong reason.

13.5 Settlement

Liquidators receive underlying assets, never the position itself. Nobody wants to hold an illiquid position, and a liquidation that pays in something the liquidator cannot sell will not be performed, which is operationally the same as having no liquidation.

The seizure withdraws from the never-deployed portion, which by Proposition 9.2 is immediate regardless of utilization. This is the step that Section 9 exists for. Without the capacity rule it queues, and a liquidation that queues is not a liquidation.

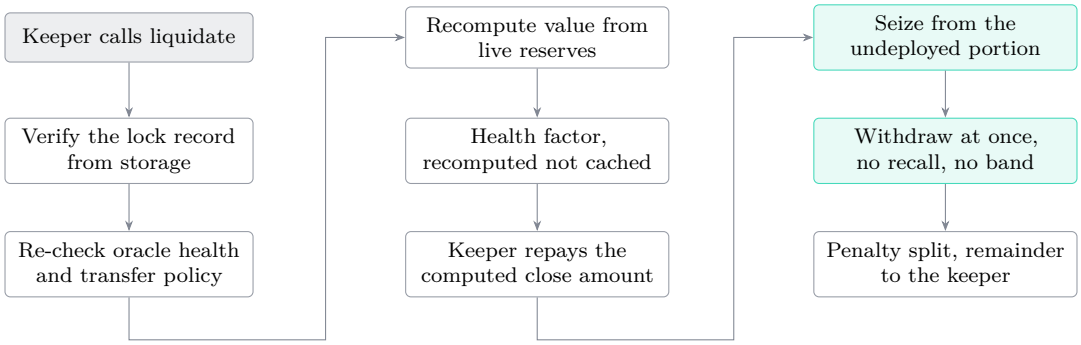

Figure 10: The unwind, read down each column in turn. The step that Section 9 exists for is the withdrawal: because the seizable portion is by construction the portion that was never lent, it completes immediately whatever the utilization (Proposition 9.2). Liquidators receive underlying assets, never the position itself.

14 Bad debt

14.1 Reporting

A level 5 liquidation settles a position whose collateral may not cover its debt. The deficit is bad debt, and the first requirement is that it be reported as whatever it actually is.

Invariant 14.1. A settlement reports the realized deficit. Zero is reported only when the deficit is genuinely zero.

A settlement path that always reports zero is not a rounding choice. It makes the protocol believe it is solvent while a real deficit accumulates somewhere that nobody is funding, and because every individual settlement looks clean, the condition is discovered only when a reserve is drawn against and found short. Unreported bad debt is bad debt nobody funds.

14.2 Absorption order

A realized deficit is drawn against four layers in order.

1. The bucket's own insurance reserve.
2. The protocol backstop, for shared buckets only.
3. A governance decision.
4. Socialization across suppliers, on the critical timelock path.

The fourth is never automatic. Socialization spreads a loss across suppliers who did not choose the position that caused it, so it is a decision rather than a mechanism, and it is made on the slowest path the protocol has.

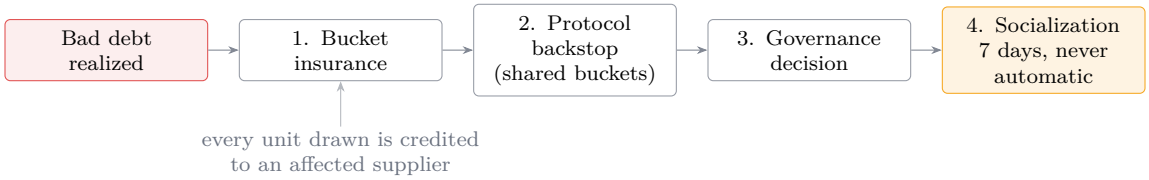

Figure 11: A realized deficit draws on four layers in order. An isolated bucket's loss stops at the first, and the fourth is a governance decision on the slowest path rather than a mechanism that fires on its own.

Invariant 14.2. Every unit drawn from an insurance reserve appears as a credit to an affected supplier. The amount drawn and the amount credited are always equal.

Invariant 14.2 closes a failure that is easy to write and hard to notice. A draw that reduces the recorded deficit without crediting the suppliers who absorbed it leaves the protocol's books correct and the suppliers short. They lose twice: once to the bad debt, and once to the insurance that was supposed to cover it. The two figures are reconciled independently each epoch rather than relying on the settlement path to be right.

14.3 Isolated buckets do not socialize outward

A loss in an isolated bucket is absorbed by that bucket's insurance and then stops. It does not reach the shared backstop and it does not reach suppliers outside the bucket unless governance explicitly extends coverage through the same slow path.

This is the containment property that Section 11 relies on. It is also the property most easily lost by an implementation that treats a bucket as an accounting label rather than as a boundary the absorption path actually checks.

14.4 The safety module

Pause controls are mandatory rather than optional, and they fail closed. A safety check that reports no pause when its configuration is absent or unrecognized fails open, which is the

opposite of what a pause exists to do.

Authority over pauses is deliberately asymmetric. Setting a pause is available to governance and to an emergency authority. Clearing one is available to governance alone and is timelocked, with a single narrow exception: a pause triggered by an unusable price feed clears automatically once that feed has been healthy for its full cooldown, because the condition that caused it is objectively observable and its resolution needs no judgement.

Invariant 14.3. A pause may only remove risk, never add it. Repayment, collateral top-up and bad debt absorption remain available under every pause state.

A pause that blocks repayment prevents borrowers from curing their own positions while their health continues to deteriorate, which converts a safety mechanism into a liquidation engine pointed at the users it was meant to protect.

Part V: Analysis

15 Risk analysis

Each risk below is stated as the mechanism, the bound that contains it, and what that bound depends on. Where a bound rests on an off-chain control rather than on the model, we say so.

15.1 A withdrawal run at the deployment ceiling

Mechanism. Providers withdraw while utilization is high. The instant reserve is consumed, the remainder queues, and the queue itself becomes a reason to withdraw.

Bound. The instant reserve is 20 percent of supply and is never lent, so withdrawals below 80 percent utilization are unaffected. Between 80 and 90 percent the split tapers, which removes the discontinuity that would otherwise reward withdrawing first. Above 90 percent, emergency recall freezes new borrowings of that asset, so utilization cannot continue to rise from the numerator while the denominator falls.

Depends on. The identity between the instant reserve and the tier 4 cap holding as a single parameter, and on emergency recall being triggered. The second is an external action and is the weakest link in this chain.

15.2 Depeg of an asset priced as stable

Mechanism. An asset classified as stable loses its peg. A stable pair pool becomes a volatile pair pool while its collateral factor still reads 75 percent and its liquidation threshold still reads 83 percent.

Bound. Every asset priced as a stable is monitored against a peg band. An asset outside its band is not a stable for risk purposes, and the collateral factor reads the asset's current

classification rather than the one it was listed under. Pools containing a depegged asset enter a state in which only repayment and liquidation remain.

Depends on. The band being narrow enough to trigger before the mispricing is material, and on the classification being read at every valuation rather than cached at listing.

15.3 Oracle failure

Mechanism. A feed goes stale, widens its confidence interval past usefulness, or disagrees with a second source.

Bound. Every price passes through one path that fails closed at four points: an adapter that is not wired, an age past the staleness bound, a confidence interval wider than the asset permits, and divergence between two sources beyond a threshold. Divergence does not select a winner. When two sources disagree, the price is unknown and collateral operations for that asset pause.

Depends on. A second source existing for every collateral asset, and on the smoothing window being long enough that manipulation must be sustained. Section 10 sets that window at 30 minutes for collateral valuation, and notes that concentrated liquidity makes short-window manipulation cheaper rather than dearer.

15.4 Keeper inaction

Mechanism. Liquidations and emergency recall require an external party to act. Positions that are unprofitable to liquidate are not liquidated, and the ones nobody liquidates are the ones that become bad debt.

Bound. Bonuses rise with each level, so a position that is unprofitable at level 2 becomes more profitable at level 4. Liquidity collateral carries a higher bonus than single-asset collateral at every level, which prices the unwind step that would otherwise make it the collateral keepers decline.

Depends on. The bonus schedule being adequate under the conditions where it matters, which are the conditions where gas is expensive and many positions are eligible at once. This is the risk in the protocol least addressable by design alone.

15.5 Shard skew and replication lag

Mechanism. The aggregate of Section 7 is assembled from shards written at different times. A gate that reads a lagging view could permit an action the current state would refuse.

Bound. Invariant 7.2 requires every gate to count pending changes in the restricting direction, so lag can close a door early and cannot open one. Invariant 7.1 refuses a read containing a shard past its staleness bound rather than treating its last written total as current.

Depends on. Nothing external. This bound is structural.

15.6 Recall unfairness

Mechanism. Under bin-scoped clearing, a provider's queued withdrawal is released when their specific backing bins are repaid, not when the protocol receives that asset generally. Two providers in identical circumstances can wait for very different periods, and neither can see or influence which they are.

Bound. None, in the first version. This is accepted rather than resolved, and the consequence is disclosure: an interface must not present a global queue position, because under bin-scoped clearing there is no global queue and a position in one is not a fact.

Depends on. Pool-scoped and asset-global clearing are specified and remove most and then nearly all of this, in that order. Neither is required for the solvency results of Section 9.

15.7 Real-world asset redemption illiquidity

Mechanism. On-chain depth for a tokenized asset can disappear within minutes while redemption from the issuer takes days. A liquidation that must sell into that gap cannot complete at the modelled price.

Bound. Not solvable in code, and bounded instead: borrow caps set well below on-chain depth, a withdrawal queue isolated to the bucket, a state that admits only repayment and liquidation once depth falls below a floor, and monitoring on depth, queue length and feed age. Section 11 also prohibits one real-world asset collateralizing a borrowing of another.

Depends on. The depth floor being set against stressed depth rather than observed depth, and on the monitoring being live. Both are operational.

16 Security considerations

This section covers the decisions that change how the protocol should be trusted, rather than the full catalogue of controls.

16.1 One program, one upgrade authority

The protocol is a single program with module separation rather than several programs. The reason is on the hot path: a swap that changes the active bin triggers the router, which touches routing state and bin arrays. Within one program that is a function call. Across programs it is a cross-program invocation, which costs compute on every price-moving swap and turns the routing state layout into a public interface that cannot then be changed.

The cost is that one upgrade authority governs everything. Three controls address it. The loader authority is the governance account rather than a key. Upgrades run on the critical timelock path and the queued operation commits to the exact program hash to be deployed. And the actual on-chain loader authority is compared against the expected account continuously, because an upgrade path that does not run through the protocol's own timelock makes

every other timelock decorative: whoever holds the upgrade authority can replace the timelock itself.

16.2 Two timelock delays

Governance operates on two delays. Non-risk parameters, fee presets, keeper incentives and metadata move on 72 hours. Feed identity, collateral factors, liquidation thresholds, caps, buffer rules, treasury limits, program upgrades, authority changes, loss socialization, enabling a collateral class, and anything else that increases risk move on seven days.

Both minimums are fixed at initialization rather than being themselves governable, since a delay governance can set to zero is not a delay, only a delay one vote further away. A queued operation commits a hash over the program, the instruction, the full arguments, every account with its flags, a nonce, the authority epoch and the target configuration version. Executed operations are terminal and cannot return to the queue.

The corresponding implementation hazard is a fast configuration path that reaches a risk parameter through a shared internal setter. Where that happens, the timelock is decorative for precisely the parameters it exists to protect, so the two paths do not share code and each risk field is written in exactly one place.

16.3 Authority is resolved once

Every role resolves through a single authority record keyed by an epoch, and no module holds its own authority field. Transferring governance increments the epoch, which invalidates every cached authority simultaneously. The alternative, per-module authority fields updated individually, fails by leaving one behind.

16.4 Asset-level transfer behaviour

Both token programs are pinned when a pool is created, so an asset cannot later be presented under a different program than the one it was reviewed under. Transfer extensions are enumerated at pool creation rather than at first deposit, so an incompatible asset never gets a pool instead of getting one nobody can withdraw from.

Where an asset levies its own transfer fee, every credit is computed from balance deltas measured across the transfer. An extension that could block a transfer is checked before a borrow is permitted, since an extension that blocks seizure makes the position unliquidatable at exactly the moment that matters.

17 Related work

17.1 Concentrated liquidity

Concentrated liquidity market makers let a provider commit capital to a chosen price interval rather than to the whole curve, which raises the depth a given amount of capital offers near the current price [1]. The capital efficiency is real and the idle inventory problem is a direct consequence of it: capital committed to an interval the price is not in provides depth that nobody is using, and earns nothing while it does so.

Juncta inherits this structure and addresses the consequence. We do not claim a better allocation of liquidity across prices. We claim that the capital sitting at prices the market is not visiting can earn a second return without leaving the venue.

17.2 Discretized and bin-based market makers

Bin-based designs replace the continuous curve with a partition of the price axis, trading at a constant sum within each bin. This gives zero price impact inside a bin and makes a bin an addressable object with its own reserves.

That addressability is the property Juncta depends on, and it is why the design is built on a discretized market maker rather than a continuous one. Selective deployment requires being able to reason about, and act on, the capital at one distance from the price without touching the capital at another. On a continuous curve there is no such object to act on.

17.3 Pooled lending markets

Pooled lending markets accept supply against shares, lend against collateral, and set rates from a utilization curve with a kink [2, 3]. Equation (10) is this model, and we do not claim novelty in it.

The difference is the supply side. In a pooled lending market, supply is a deliberate deposit made by a participant who has chosen lending over other uses of the capital. In Juncta the supply is inventory that was already committed to a different purpose and is unused at this moment, so the capital has no opportunity cost while it is lent and the rate it requires reflects only the risk of being recalled. That is why the rate in Eq. (11) carries a term structure over distance rather than over duration.

17.4 Isolated lending

Isolated markets bound the consequences of listing a risky asset by confining its losses to a market that only its own participants joined. Juncta's buckets are this idea, applied per approved pool.

One distinction matters and is easy to lose. Section 11 isolates losses, not supply. An isolated pool's capital contributes to the same asset-level aggregate as any other pool's, because

fragmenting supply is the condition the protocol exists to remove. The bucket decides which insurance reserve pays and how far a loss may travel, and nothing else.

17.5 Liquidity positions as collateral

Accepting a liquidity position as collateral is not new, and the published failures have a common shape: the position is priced as an asset rather than valued as a claim on its contents, which makes the valuation a function of a price the borrower can move. Invariant 10.1 is the response, and it is a restatement of a lesson rather than a new idea.

What we believe is new is the interaction covered in Section 9. Where a protocol both lends out the reserves backing a position and accepts that position as collateral, valuation is not the binding constraint. Supply accounting is. We are not aware of a prior treatment that states the resulting capacity restriction and proves that it makes seizure immediate rather than merely cheaper.

References

[1] H. Adams, N. Zinsmeister, M. Salem, R. Keefer, D. Robinson. *Uniswap v3 Core*. 2021.

[2] R. Leshner, G. Hayes. *Compound: The Money Market Protocol*. 2019.

[3] *Aave Protocol Whitepaper*. 2020.

18 Conclusion

A concentrated liquidity position is mostly inventory. The capital is committed, posted against prices the market may not reach, and earning nothing while it waits. Juncta lends that inventory into a money market that shares the same custody and the same accounting, so that a single deposit earns trading fees while its bin is active and lending interest while it is not. The fraction lent rises with distance from the current price and the premium paid falls, which makes the rate a term structure over distance rather than over duration.

The difficulty this creates is not a valuation problem. If a position's reserves are lent and the position is also pledged, the same dollars stand behind two claims, and no haircut reduces the number of claims on a dollar. We restrict collateral capacity to the portion of a position that was never deployed. That bounds total claims by total supply (Proposition 9.1), and it makes a seizure immediate by construction rather than by estimate, because the seizable portion is exactly the portion no borrower ever held (Proposition 9.2). The cost is paid in borrowing capacity on heavily deployed pools, and a provider who wants more of it can move part of their position out of deployment and take it.

Liquidation closes the minimum that restores a target health factor, at bonuses that rise with severity, with recovery windows between levels. Proposition 13.1 gives that minimum in closed form and shows that it is not always attainable: above a threshold set by the collateral's own

liquidation threshold, no partial close restores the target, which is why the deepest level closes as required instead. Collateral priced on slow feeds is the case the windows are sized for.

What we set out to establish is narrow. Idle inventory can earn a second return without leaving the venue, and it can do so while remaining usable as collateral, provided the protocol never lets the same dollar answer for two obligations.

19 Future work

19.1 Recall fairness

Bin-scoped clearing is the version that requires no additional contention, and it is also the least fair. Pool-scoped clearing, ordered by urgency rather than by age, removes most of the unfairness at close to no cost, since the pool account is already write locked by the repayment that triggers it. An asset-global queue folded asynchronously removes nearly all of what remains.

Both are specified. Neither is required for the results in Section 9, which is why the protocol can ship without them, and both should be built in that order rather than attempting the harder one first.

19.2 Emergency recall ordering

Emergency recall currently pulls from bins ordered by distance from the active price. A pool's active bin is set by whoever trades it, so a pool can place itself at either end of that ordering by moving its own price, which is usable either to shield its own capital or to force churn on another pool's.

Ordering by tier and then by contribution share removes the manipulable input, since tier is a property of the deployment decision rather than of a price a pool can choose, and contribution share makes the largest supplier bear recall proportionally. This needs a design pass of its own before implementation. Until then the per-pool contribution cap bounds how much any single pool can gain.

19.3 Measured parameters

Several parameters in Appendix A are bounds rather than derived values, and must be set against measurements taken under load rather than reasoned to. The swap bin ceiling is the most consequential: it decides which routes are executable at all, and setting it from a compute estimate rather than a measurement produces either routes that fail or a ceiling lower than necessary.

19.4 Beyond a single chain

The design assumes one execution environment. The sharding of Section 7 is a response to a specific concurrency model, and the account and compute bounds throughout Section 5 are

that environment's limits rather than general ones. A second deployment is not a port of the accounting; it is a second instance with its own aggregate, and the question of whether supply should be shared across instances is open and is not answered here.

A Parameters

This appendix lists the parameters the arguments in this paper depend on. It is not the protocol's full configuration surface: implementation limits, revenue splits and operational thresholds are published with the protocol documentation, where they can be revised without reissuing this paper.

Values marked *measured* are bounds to be set against load testing rather than derived quantities.

Parameter	**Value**	**Change path**
Price grid (Sections 3 and 5)		
Bin identifier bounds	±443,636	Immutable
Maximum hops per route	3	Critical
Maximum bins per route	*measured*	Standard
Deployment (Section 6)		
Tier deployment	0 / 20 / 50 / 70 / 80%	Critical
Proximity premium	0 / 150 / 75 / 25 / 0 bps	Critical
Buffer safe distance	50 / 200 / 300 / 500 bps	Critical
Money market (Sections 7 and 8)		
Global shards per asset	16	Critical
Shard staleness bound	150 slots	Standard
New borrowing cap	80% utilization	Critical
Emergency recall trigger	90% utilization	Critical
Instant withdrawal band	below 80% utilization	Critical
Collateral and liquidation (Sections 10 and 13)		
Collateral factors	75 / 60 / 45 / 35%	Critical
Liquidation thresholds	83 / 68 / 53 / 43%	Critical
Target health factor T	1.20	Critical
Minimum smoothing window	1,800 s	Critical
Liquidation proceeds	keeper 90 / backstop 7 / insurance 3	Critical
Governance (Section 16)		
Standard delay	72 h	Immutable minimum
Critical delay	7 d	Immutable minimum

Table 9: Parameters the paper's arguments depend on, and the governance path required to change each.

Two entries carry a dependency worth restating. The instant withdrawal band and the tier

4 deployment cap are the same 20 percent of supply and cannot be changed independently (Section 8). And the target health factor interacts with the liquidation thresholds through Eq. (16): raising T or the thresholds raises the fraction of debt a liquidation must close, and past $h < (1 + b)$LT restoration to T stops being attainable at all.

B Notation

Symbol	Meaning	Introduced
s	bin step, in basis points	Eq. (1)
b	geometric base, $1 + s/10{,}000$	Eq. (1)
$P(i)$	lower bound of bin i	Eq. (1)
$\bar{P}_i$	midpoint price of bin i	Eq. (3)
i_{active}	identifier of the bin holding the price	Section 6
d_i	distance of bin i from the active bin	Table 2
L_i, S_i	bin i's liquidity and outstanding shares	Eq. (4)
$R_{X,i}, R_{Y,i}$	bin i's reserves of each asset	Eq. (14)
δ	fraction of pool reserves deployed	Definition 9.1
U	global utilization of an asset's market	Eq. (10)
U^*	utilization at the rate kink	Eq. (10)
r_0, r_1, r_2	rate at zero, at the kink, and at full	Eq. (10)
$\bar{r}_{\text{prox}}$	weighted proximity premium	Eq. (11)
CF	collateral factor	Definition 9.1
LT	liquidation threshold	Eq. (15)
h_k	the kth haircut, in $(0, 1]$	Definition 9.1
V_{pos}	value of a liquidity position	Eq. (14)
V_{col}	collateral capacity of a position	Eq. (13)
V	total supply of a pool	Proposition 9.1
B_{mm}	money market borrowings from a pool	Proposition 9.1
B_{col}	borrowings against positions in a pool	Proposition 9.1
HF, h	health factor	Eq. (15)
T	target health factor after liquidation	Proposition 13.1
b	liquidation bonus	Proposition 13.1
f	fraction of debt closed	Eq. (16)

Table 10: Complete notation.

C The close fraction with multiple collateral types

Proposition 13.1 treats a position holding a single collateral type. The general case is worth stating, because it has a consequence for which collateral a liquidator chooses to seize.

Proposition C.1 (Close fraction, general form)**.** *Consider a position with collateral values $C_1, \dots, C_n$ carrying liquidation thresholds $\mathrm{LT}_1, \dots, \mathrm{LT}_n$, total debt D, and health factor $h = \sum_j C_j \mathrm{LT}_j / D$. A liquidation that seizes collateral of type m at bonus b_m must close a fraction*

$$f = \frac{T - h}{T - (1 + b_m)\,\mathrm{LT}_m} \tag{17}$$

of the debt to restore health to T.

Proof. A liquidator repaying Δd seizes collateral of type m worth $\Delta d\,(1 + b_m)$, which reduces C_m by that amount and leaves every other C_j unchanged. Requiring the resulting health factor to equal T,

$$\frac{\sum_j C_j \mathrm{LT}_j - \Delta d\,(1 + b_m)\,\mathrm{LT}_m}{D - \Delta d} = T.$$

Multiplying out and collecting Δd,

$$\sum_j C_j \mathrm{LT}_j - TD = \Delta d\big[(1 + b_m)\mathrm{LT}_m - T\big].$$

Dividing by D and substituting h gives Eq. (17). Setting $n = 1$ recovers Eq. (16). □

Corollary C.2. *The fraction of debt that must be closed depends on the liquidation threshold of the collateral actually seized, not on the position's weighted average threshold. Where bonuses are equal, seizing the collateral with the lowest liquidation threshold closes the least debt.*

Proof. Equation (17) is increasing in LT_m, since the denominator $T - (1 + b_m)\mathrm{LT}_m$ decreases as LT_m rises while the numerator is independent of m. □

Corollary C.2 has a practical reading that runs in the protocol's favour. A liquidator choosing between collateral types will prefer the one with the lowest liquidation threshold, which is the riskiest collateral the position holds. The position is therefore left holding its safest collateral, and the protocol's remaining exposure improves as a by-product of the liquidator acting in their own interest.

The effect is partly offset where the riskiest collateral also carries the highest bonus, as it does between single-asset and liquidity collateral in Table 8. The ordering in any particular case follows from Eq. (17) with the applicable b_m and LT_m, and is not uniform across all pairs of collateral types.