The Everything Protocol

Decentralized Finance, finally solved: *one reserve that prices every trade, backs every loan, and fills every order, with nothing idle*

Jean Rausis Eric Rabl Flavio Roth
Abdel Mecheri Quang Chuc Nguyen*

August 2026

Abstract

Decentralized finance keeps its three core venues apart: exchanges price assets, money markets lend them, and derivative venues host leverage, each locking its own capital. The Everything Protocol fuses the three into a single contract per token pair. One reserve simultaneously prices swaps on an amplified concentrated-liquidity invariant (a port of Curve's CryptoSwap), backs loans and leveraged positions, and hosts resting limit orders on a geometric tick grid; the same deposit is never idle. The protocol needs no external oracle: a lagged price band, advanced once per block by linear decay and a one-way clamp, bounds every credit decision, while the curve's own profit-gated repeg re-centers liquidity. Solvency is kept by construction rather than by trust: a two-ledger accounting separates the junior liquidity tranche from senior user escrow, every exit pays cash in full and never an IOU (not-yet-realized yield is fronted collectively, within a capacity gate), fill proceeds rank senior-most, and losses are written down junior-first so suppliers are never haircut. A liquidation cascade stabilizes the pool before every book-moving operation and terminates in a bounded number of steps per transaction; no reachable state bricks the pair. This paper presents the mechanism, its mathematics, and the invariants that hold it together.

*Author names appear in no particular order; the ordering carries no information about relative contribution.

1. Introduction

A decentralized financial system is usually assembled out of specialized protocols. An automated market maker (AMM) prices the asset [3, 4]; a money market lends it [5, 6]; a perpetuals venue carries leveraged exposure; an order-book layer, when it exists at all, lives off-chain. Each protocol holds its own capital, and each boundary between them is paid for twice: once in idle liquidity, once in composition risk. The same unit of value cannot simultaneously provide depth to traders, earn interest from borrowers, and rest at a limit price, even though those three uses are rarely in conflict at the same instant. A venue that lets one deposit serve all three at once therefore starts with a structural capital advantage that no amount of composition between silos can replicate.

The separation is most costly exactly where on-chain finance should have an advantage: the long tail of assets. A lending market for a small or mid-cap token cannot exist safely on a monolithic money market, because its liquidation flow would have to cross an external exchange whose depth the money market can neither observe reliably nor control. Lenders bear the risk that, at the moment collateral must be sold, the venue that was supposed to absorb it has moved or emptied. The conservative answer, and the prevailing one, is not to list such assets at all.

The Everything Protocol starts from the opposite premise: *if a single venue owns the pricing curve, the credit book, and the liquidation path, it can underwrite what siloed protocols cannot.* When the pool that lends is the pool that prices, borrowable capacity can be derived, price level by price level, from the depth that will actually absorb a liquidation; when the liquidation engine settles into its own curve, the collateral's exit liquidity is a protocol variable rather than an assumption about a third party. Unification is not a convenience here; it is the risk model.

Concretely, the Everything Protocol is one contract per token pair in which:

- **swaps** execute against an amplified, self-re-centering concentrated-liquidity invariant, a faithful port of Curve's CryptoSwap (*twocrypto-ng*) [1], retaining its dynamic imbalance fee and adding a routing-invariant fee resolution and a utilization surcharge;
- **loans and leveraged positions** borrow one token of the pair against collateral in the other, anchored to a geometric grid of price ticks, with capacity shaped per tick by a pluggable model of the curve's own depth;
- **limit orders** rest on the same tick grid, filled in-path by the protocol's own swaps at exactly their limit price, and may opt in to be lent while they wait, so that resting capital earns the borrowers' interest;
- **settlement** pays real tokens in full on every exit, never an IOU, fronting not-yet-realized yield from the collective liquidity within a capacity gate, and reserves filled-order proceeds senior-most so that collecting a fill is never blocked;
- **liquidation** is a permissionless cascade that runs before every book-moving user operation, seizes collateral in per-tick aggregates, writes losses down against the junior liquidity tranche, and terminates in a bounded number of steps without ever wedging the pair.

No external price oracle or keeper subsidy appears anywhere in the design. The protocol's only inputs are deposits, trades, and time; the attack surfaces that arrive bundled with an external price dependency, oracle manipulation and cross-venue depegs, are absent by construction rather than mitigated. (The tokens a pair lists carry their own risks, which Section 11 prices.)

The intellectual debt is acknowledged plainly: the pricing core is Curve's [1], the interest-rate shape is Aave's [5], the tick discretization echoes Uniswap v3 [4], and the hard per-tick liquida-

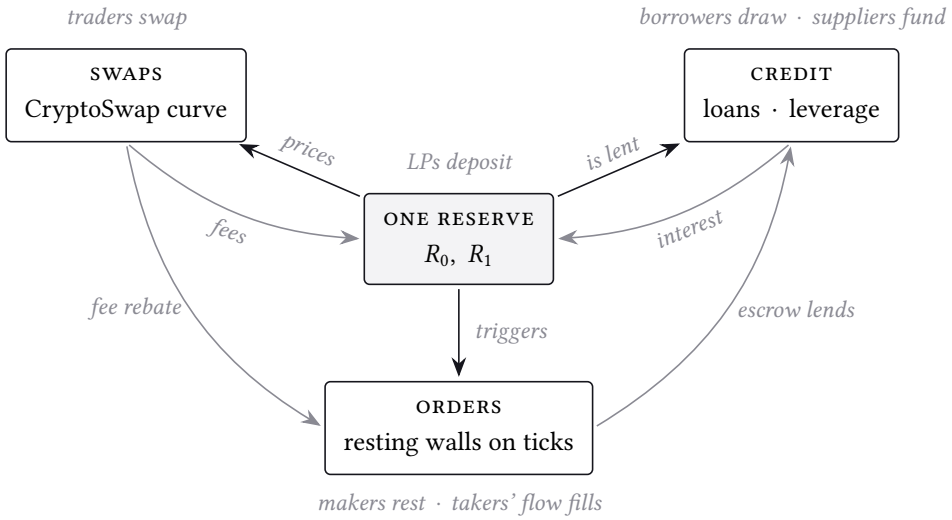


FIGURE 1. The unified pair, each market annotated with the roles that drive it. A single reserve, funded by liquidity providers, prices the swaps of traders and is lent to borrowers, and its price crossings trigger the orders makers rest; the LP share of fees and interest flows back into the same pool, a slice of each fill's fee is rebated to the filled maker, and opted-in order escrow deepens the credit book on the suppliers' account.

tion descends from the protocol's own first generation rather than from crvUSD's continuous LLAMMA [7]. What is new is the joint: a single solvency ledger under all three markets, a price band that replaces the oracle, an order engine whose escrow is also loanable supply, and a loss waterfall with an explicit seniority order and a liveness guarantee enforced by construction. The whole is deliberately more than the sum of the parts, because the parts share one balance sheet.

Section 2 states the model and the design principles. Sections 3 to 4 develop the pricing layer: the invariant, its solvers, the fee, the repeg, and the internal price band. Sections 5 and 6 develop the two markets built on the grid: resting orders and credit. Section 7 composes them into leverage. Sections 8 and 9 give the liquidation cascade and the solvency accounting, the heart of the protocol. Section 10 describes the on-chain architecture, Section 11 discusses risks honestly, and Section 12 concludes.

2. One Reserve, Three Markets

2.1. The model

A *pair* holds two tokens. Its state divides into two books kept under one roof:

- the *pricing reserve* (R_0, R_1): the balances the curve prices against, owned by liquidity providers (LPs) through pool shares;
- the *escrow*: user-owned deposits held by the same contract but never counted as pricing reserve, namely loan collateral, resting order funds, and standalone single-asset supplies.

Five roles interact with these books. *Traders* swap against the curve and, in passing, fill resting orders. *Liquidity providers* deposit both tokens into the pricing reserve and earn the swap fees plus a share of borrower interest. *Makers* rest limit orders on a price grid. *Borrowers* post

collateral in one token and draw the other, choosing the price at which they are prepared to be liquidated. *Suppliers* are the owners of any escrowed deposit that opted in to be lent: their capital joins the borrowable pool and earns the supplier slice of interest. Figure 1 annotates each market with the roles that drive it.

The unification thesis is a capital-efficiency identity: the very tokens that price a swap are simultaneously the inventory of the credit book, and the escrow that waits at a limit price is simultaneously loanable supply. Nothing is idle by construction (Figure 1). What elsewhere takes three protocols, two bridges, and a wrapper token happens here inside one contract, with one deposit, under one solvency ledger.

2.2. The state-update discipline

Every operation that moves the pool's books runs the same preamble before touching anything: accrue interest on both token sides, advance the price band (once per block, before any reserve mutation), then run the liquidation cascade until no liquidatable tick remains (an underwater tick the pool cannot yet price or absorb is deferred and retried; Section 8). Only then does the operation itself execute. The consequence is a strong ordering guarantee: *no user action ever executes against a stale or un-liquidated pool*; the flash loan, which touches neither book and repays within its transaction, is the one exception. A swap prices post-cascade reserves; a borrow is sized against post-accrual debt; an order fill sees the band the block opened with. Composition is protocol-controlled: the pair exposes curated high-level operations that compose the primitives internally, rather than letting callers assemble sequences that bypass the preamble (the flash callback may re-enter the pair's own operations, but each one re-runs it). What reads as a restriction is the guarantee's enforcement: a sequence that skips the preamble cannot be built. Sequencing attacks that exploit stale composite state, a standing hazard wherever separate protocols are glued together, have no equivalent here.

2.3. Design principles

Four principles recur throughout the design and are worth naming once.

P1. **No external inputs.** Prices, yield, and liquidation triggers derive from the pool's own state and the clock. There is no oracle to manipulate, no venue to depeg from, and no keeper to subsidize.

P2. **Solvency by construction.** Every credit to a user rounds down; every debt owed to the pool rounds up; every aggregate debit is clamped at zero. Claims are ordered senior to junior, and each payout base is a floor that reserves everything senior to it.

P3. **Failure as value.** Numerical routines that can fail on degenerate geometry return a classification instead of reverting, so that the protocol can act on a drained or extreme state rather than freeze in it. No reachable state bricks the pair.

P4. **Conservative pricing at every seam.** Where two subsystems meet (curve and band, orders and credit, liquidation and pricing), the protocol always adopts the bound that favors the pool, so cross-subsystem arbitrage buys nothing.

2.4. Notation

Amounts are in the tokens' native units scaled to eighteen decimals; prices are quoted as token 0 per token 1 unless stated otherwise. Throughout,

R_0, R_1	pricing reserves
R	one side's pricing reserve, R_0 or R_1
p_s	price scale (the curve's center of concentration)
D	the curve invariant
A, γ	amplification and damping parameters
o	the repeg oracle (an EMA of post-trade spot prices)
b^+, b^-	the price band's buy and sell anchors
M	a token side's borrow interest multiplier
L	a token side's supply index
B	total borrowed principal on a side
E	admitted lent escrow on a side
C	borrow capacity of a side
u	utilization, $u = B/C$
$P(i)$	the tick grid, $P(i) = 1.01^i$
B_i, A_i	tick i's execution and liquidation prices (Section 5; distinct from A, B)

3. The Pricing Curve

Swaps price against an amplified concentrated-liquidity invariant, a port of Curve's CryptoSwap in its two-coin form (*twocrypto-ng*) [1]. The port is faithful to the reference implementation's arithmetic, staged integer division included, so that a decade of production exposure to the math carries over; what the Everything Protocol adds around it is described in the sections that follow.

3.1. The invariant

Balances enter the curve in *value units*: after normalizing token decimals, $x_0 = R_0$ and $x_1 = R_1\, p_s$, where the price scale p_s is the pool's current center of concentration, quoted as token 0 per token 1. Define

$$K_0 \;=\; \frac{4\,x_0 x_1}{D^2}, \qquad K \;=\; A\,K_0\,\frac{\gamma^2}{\big(\gamma + 1 - K_0\big)^2}, \tag{1}$$

where A is the amplification coefficient and γ a damping range. The invariant D is the unique positive root of

$$KD\,(x_0 + x_1) \;+\; x_0 x_1 \;=\; KD^2 \;+\; \frac{D^2}{4}. \tag{2}$$

K_0 equals 1 when the pool is perfectly balanced in value units and decays toward 0 as it departs; K therefore interpolates the whole curve between a constant-sum regime near balance [2] (deep, nearly slippage-free liquidity around p_s) and a constant-product regime far from it (asymptotic safety of $xy = k$; see Figure 2). The parameters are bounded to $A \in [0.1,\ 1000]$ and $\gamma \in [10^{-8},\ 0.06]$ and may be re-tuned live along a time-linear ramp (at most a factor of ten per ramp, no ramp shorter than one day), which lets a pair change personality, from stable-pair tightness to volatile-pair breadth, without redeployment.

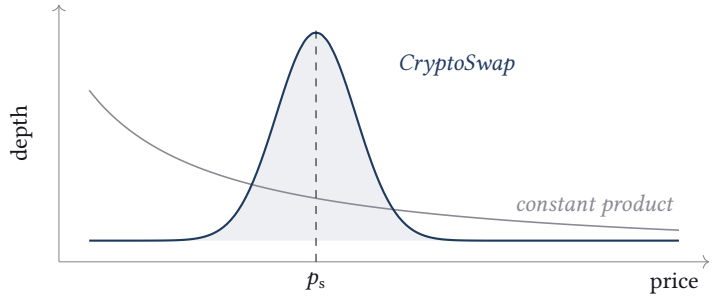


FIGURE 2. Liquidity depth by price. Depth here is the marginal reserve the pool trades per unit of price movement, $|dx_0/dp|$ evaluated along each invariant, drawn schematically for the same total capital on both curves. The CryptoSwap invariant concentrates the reserve around the price scale p_s and re-centers it as the market moves; the constant-product baseline spreads the same capital across all prices.

3.2. *Solving the curve*

Only D lacks a closed form: it is computed by Newton iteration, seeded conservatively. The output balance y at fixed D is the single real root of a cubic, evaluated analytically and polished by a short Newton run (with a cold Newton fallback where the analytic root is ill-conditioned). Both solvers run to a tolerance far below any economic quantum and are guarded by full-width intermediate arithmetic wherever a product could overflow. One engineering choice matters beyond the port itself: *failure is a value*, not an exception (P3). The solvers exist in paired forms: one that reverts, used where the caller requires an answer, and one that returns an explicit success flag. Degenerate geometry (an extreme value-unit imbalance, a collapsed invariant, an out-of-band post-solution check) is reported as a classification the caller can act on. This distinction, mundane as it looks, is what later allows a liquidation to recognize a drained pool and keep operating where a bare solver would have frozen the pair (Section 8).

The marginal spot price is exact rather than approximated: differentiating (2) implicitly yields a closed form for $p = -dx_0/dx_1$ in terms of (x_0, x_1, D), which the implementation evaluates directly. Exact-input quotes re-solve the invariant from the reserves they price on rather than trusting the cached value, and round the output down; exact-output quotes round the required input up. Every rounding in the pricing layer favors the pool (P2).

3.3. *The fees*

A taker's total rate composes three parts, each with its own driver: a *dynamic curve fee* driven by pool imbalance, a *utilization surcharge* driven by the credit book, and a static *protocol fee*.

The dynamic curve fee interpolates between two bounds with pool imbalance. With x_0, x_1 the value-unit balances, define the balance measure and mixing weight

$$K_{\text{bal}} = \frac{4\,x_0 x_1}{(x_0 + x_1)^2}, \qquad g = \frac{\gamma_{\text{fee}}}{\gamma_{\text{fee}} + 1 - K_{\text{bal}}}, \qquad f \;=\; g\,f_{\text{mid}} + (1-g)\,f_{\text{out}}, \tag{3}$$

rounded up in the pool's favor. K_{bal} is 1 at perfect balance and tends to 0 under deep imbalance, so trades that worsen the pool's balance pay toward f_{out} while trades that restore it pay toward f_{mid}; γ_{fee} sets how sharp the transition is, and $\gamma_{\text{fee}} = 0$ is the static sentinel (a flat f_{mid}, no curve evaluation at all).

The rate is resolved *once per swap*, on the hypothetical post-trade state computed from the gross input and the pre-swap reserves. This breaks the circularity between rate and net amount, and it makes the fee a pure function of (pre-state, input): the rate cannot depend on how the swap is later routed between resting orders and the curve, a property the order engine's incentive argument relies on (Section 5). If the hypothetical post-state is unpriceable, the fee resolves to f_{out}: conservative, never a revert.

The *utilization surcharge* is the credit book's voice in the swap price. It reads the output side's credit utilization u (Section 6) and is kinked: zero while u stays below a configured kink, then rising linearly with the excess above it. A swap that consumes reserves nobody is borrowing against pays nothing extra; a swap that drains the very reserves the credit book is leaning on pays in proportion to how hard it leans, so the surcharge prices the externality a large outflow imposes on borrowers and lent suppliers, whose exit liquidity that reserve is. Like the dynamic fee, it is resolved once per swap on the pre-swap state.

The *protocol fee* is a static rate accruing to the protocol. The total of the three parts is capped, and a coupling constraint holds the worst-case fee below the liquidation penalty so that fees can never make a liquidation uneconomical.

3.4. Profit, and the price scale that follows the market

Concentration must follow the market or die by it. The pool tracks its own wealth with the *constant-product benchmark*

$$\text{xcp} \;=\; \frac{D}{2\sqrt{p_{\text{s}}}}, \qquad v \;=\; \frac{\text{xcp}}{S}, \tag{4}$$

where S is the LP token supply; v is the virtual price and grows only with retained fees. A profit accumulator χ compounds the growth factor of v across operations. The pool re-centers by moving p_{s} toward an internal EMA oracle o of post-trade spot prices,

$$o \leftarrow \alpha\, o + (1-\alpha)\min\!\big(p_{\text{last}},\, 2p_{\text{s}}\big), \qquad \alpha = e^{-\Delta t/T}, \tag{5}$$

with T the oracle's smoothing horizon, a per-pair lever on the order of minutes, and the $2p_{\text{s}}$ cap bounding what a single block can inject into the oracle. With $n = |o/p_{\text{s}} - 1|$ the oracle's relative distance and $\varsigma = \max(\sigma,\, n/5)$ the step, σ being the configured minimum step, the scale moves only when $n > \varsigma$, by

$$p'_{\text{s}} \;=\; p_{\text{s}} + \frac{\varsigma}{n}\,\big(o - p_{\text{s}}\big), \tag{6}$$

so a move never overshoots the oracle and each move covers at least a fifth of the gap. The move *commits only if the pool can afford it*: with v' the post-move virtual price, the move is attempted and then kept only under the double gate

$$\underbrace{\big(v-\varepsilon\big)^2 \;>\; \chi}_{\text{attempt}} \qquad \text{and} \qquad \underbrace{v' > 1 \;\text{ and }\; (v')^2 > \chi}_{\text{commit}}, \tag{7}$$

i.e. only when accumulated fees exceed the mark-to-market cost of moving concentrated liquidity (the buffer ε reserves a minimum retained profit): re-centering is funded by income, never by principal. The gate is evaluated on exact full-width products because the virtual price is unbounded above, and it is saturating below: after a loss re-baseline the pool simply stops re-centering until fees rebuild the buffer. Outside an explicit parameter ramp, the virtual price

is enforced non-decreasing on the trading path (a liquidation's checkpoint re-baseline may legitimately lower it; Section 8); the repeg is additionally frozen during liquidation cascades, on drained (unpriceable) reserves, and whenever the move would push the folded balances out of the solvable domain: each freeze exists so that only genuine, fee-funded trading can move the pool's center (Section 8 details the cases).

4. The Internal Price: a Band, not an Oracle

Credit needs a price that a trader cannot set. Oracle-fed designs answer by importing a dependency and its whole failure surface; the Everything Protocol answers by removing the need. The protocol's first generation answered with smoothed virtual reserves; the present design distills the idea to its minimal form: a stored *price band*, two anchors bracketing the spot,

$$b^+ \;\geq\; s \;\geq\; b^-, \tag{8}$$

where s is the curve's marginal spot price, tracked in the reciprocal orientation to Section 3 (token 1 per token 0); the band's algebra is orientation-symmetric. The band is advanced once per block, in the operation preamble, before any reserve moves: a linear decay of each anchor toward the spot, then a one-way clamp,

$$b \;\leftarrow\; b \pm |s - b| \cdot \frac{\Delta t}{\tau}, \qquad b^+ \leftarrow \max(b^+, s), \qquad b^- \leftarrow \min(b^-, s), \tag{9}$$

with τ the band's decay window, a per-pair lever on the order of minutes (Figure 3). Three consequences follow directly from (9) and carry the protocol's oracle-free security argument.

Property 1 (Same-block immunity). *Within a block the band is frozen: the decay runs on the first operation of the block, against the spot as the block opened. A flash-loan wick, however violent, cannot loosen any credit decision in its own block: no anchor moves, and the live spot enters each composition only on its conservative side, able to tighten borrow validation and to defer a liquidation but never to manufacture one (per-tick capacity shaping reads the live curve by design, strictly below the global envelope the pair enforces itself).*

Property 2 (Manipulation only widens). *The clamp in* (9) *is one-way: pushing the spot up drags b^+ up in full at the next advance but leaves b^- decaying on its own clock, and symmetrically for a downward push. An attacker pays curve slippage to move the spot and receives in exchange a* wider *protective margin working against them. In particular, the* sandwich attack *on credit pricing (push the spot, extract a mispriced borrow or a forced liquidation, pull the spot back) is unprofitable by construction: the attacker pays slippage on both legs and every credit decision in between reads the band edge that moved against them. At rest the band collapses to zero width and costs honest users nothing.*

Property 3 (Lag decays geometrically). *A pair left untouched for one window τ closes the band on the spot in a single advance. Under continuous activity the per-block decays compound multiplicatively, leaving at most $e^{-T/\tau}$ of a displacement after time T; either way the protective margin is transient by construction and never a standing spread.*

4.1. How the band is consumed

Every consumer selects the band edge that favors the pool (P4), composed with the live spot:

- **Credit pricing.** Borrow validation and liquidation pricing read the *lending price*: the composition $\min(b^-, s)$ or $\max(b^+, s)$, the edge dictated by the direction of the exposure and by the consumer's own conservatism. Borrow validation maximizes the loan's price, so debt is never under-stated at the open; liquidation and per-tick capacity minimize it, the lenient edge, so a spike can defer a liquidation but never manufacture one. A large swap can therefore move the swap price freely but moves the lending price only at the band's pace: mass liquidation by single-block price action is structurally excluded.
- **Execution basis.** While an anchor is displaced beyond the spot (the wake of a large move), the curve leg of a swap in that direction executes on reserves *reconstructed at the anchor* rather than at the raw spot. A back-run that tries to buy the crash in the same block pays a material premium over the raw curve price, decaying with τ; conversely, flow that waits out the window pays nothing. The dynamic fee, deliberately, stays on the real imbalance state.
- **Spread guard.** After every swap, the post-trade spot is compared against the lagged opposite anchor; a deviation beyond a configured tolerance reverts the trade. This is the outermost bound on how far any single transaction can carry execution away from the recent past.

4.2. Inverting the curve

Band edges live in price space; the curve lives in reserve space. Bridging them requires the inverse problem: *given a target price p^*, which reserve pair on the live invariant trades at exactly p^*?* Under $xy = k$ this has the closed form $x = \sqrt{k/p}$; under (2) it has none. The reference solver is a bisection on x_0 over a bracket spanning the solvable domain, maintaining as a loop invariant that the target price stays bracketed: $p(\text{lo}) \le p^* \le p(\text{hi})$. The returned endpoint is then chosen by exposure: reconstructions that value *collateral* return the high endpoint (never over-valued), reconstructions that value *debt* return the low endpoint (never under-stated). Because either endpoint already satisfies its side of the bracket invariant, the conservative direction of every band-priced quantity is exact rather than approximate, independent of how far the bisection converged. Paired failure-as-value forms let a caller classify a state whose spot lies outside the bracket (a drained pool mid-cascade) instead of reverting on it.

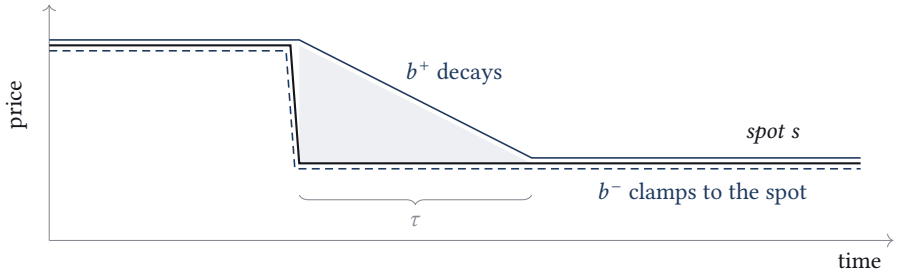
5. The Tick Grid: Orders

Everything that waits in the Everything Protocol waits on one ruler: the geometric grid

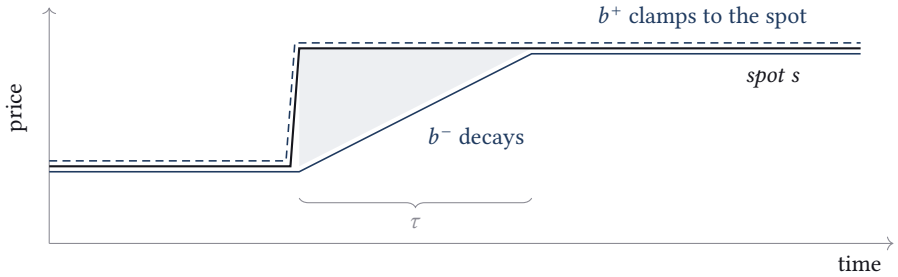
$$P(i) \;=\; 1.01^{\,i}, \qquad i \in [-3702,\ 13\,598], \tag{10}$$

adjacent levels one percent apart, spanning seventy-five orders of magnitude. The grid carries two prices per index, one per market:

$$\underbrace{B_i = P(i)}_{\text{execution ruler}} \qquad\qquad \underbrace{A_i(t) = P(i)\,/\,M(t)}_{\text{liquidation ruler}}. \tag{11}$$



(A) A downward move: b^- clamps to the new spot at once, b^+ decays toward it over one window τ.



(B) An upward move: b^+ clamps to the new spot at once, b^- decays toward it over one window τ.

FIGURE 3. The band after a price move, anchor lines drawn a hair apart for legibility (at rest all three coincide: the band has zero width). In each direction the anchor on the far side of the move clamps to the new spot at once, while the other decays toward it over one window τ (drawn idealized: one quiet advance after τ snaps the band shut, while repeated advances close the gap multiplicatively). Credit prices read the conservative edge for the whole shaded wake, and the execution basis charges back-runs the displaced anchor instead of the raw curve until the wake closes.

The *execution* price B_i is static: an order placed at tick i fills at exactly $P(i)$, today or in a year. The *liquidation* price A_i carries the borrow side's interest multiplier M (Section 6): as debt compounds, every loan's liquidation level drifts against the borrower in lockstep, which is precisely what lets the protocol liquidate a whole tick as one object, with no per-loan clock. Anchoring both markets to one grid is what makes their batch operations $O(1)$ per level: orders aggregate into per-tick buckets, loans aggregate into per-tick debt, and a price crossing settles each level in one step regardless of how many positions live there.

5.1. *Resting orders*

An order deposits an amount of one token at a tick and asks to become the other token at B_i. Placement enforces exact escrow custody (the deposit is held by the pair, never counted as pricing reserve), a dust floor, and a no-born-stale rule: the chosen tick must not already be crossed, so staleness can only arise from a later market move. At placement the maker makes one binding choice: whether the resting funds are *lent*. Lent escrow joins the borrowable pool (Section 6), earns the supplier slice of borrower interest (compounded into the order, increasing the amount that swaps at execution), and accepts capacity-gated entry and exit: an opt-in that the lend budget or the lent-order depth gate cannot admit in full reverts outright, never degrading silently to non-lent. Non-lent escrow earns nothing, is never lent, and exits unconditionally. The flag is frozen for the order's life, and each tick keeps the two populations in separate sub-buckets.

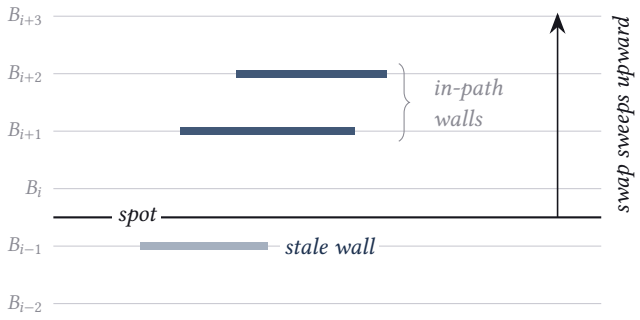


FIGURE 4. A buy-side swap crossing the execution grid. Stale walls fill first at their price without moving the curve; in-path walls fill as the price reaches them; only the residual trades on the curve. Makers receive exactly B_i; the taker's fee was already skimmed and is identical however the flow splits.

Internally a bucket never iterates its members. Each fill multiplies a per-bucket *remaining factor*; an order's live share is its entry snapshot of that factor against the current one, and a full fill bumps the bucket's version, invalidating every member in $O(1)$: a tick carrying ten thousand orders settles at the cost of a tick carrying one. The same normalized accounting runs the loan ticks. Orders are triggers against the AMM, not a matched book: no counterparty ever needs to be found, because the next taker's flow executes them at exactly their price, and a price crossing is the only event that matters.

5.2. *The sweep*

A swap's net input does not go to the curve first. It is offered, best price first, to every resting wall the trade path crosses (Figure 4):

1. the taker's total rate, the LP fee (3) plus the protocol fee and the utilization surcharge, is resolved once on the gross input and pre-swap state and skimmed up front; everything below runs fee-free;
2. *stale* walls (limit prices at or beyond the current spot, leftovers of earlier moves) fill directly at their B_i, off-curve, without moving the price;
3. *in-path* walls fill in crossing order: against a frozen copy of the pre-swap curve, the engine prices the input that would carry the price to each wall (by the inversion of Section 4); a wall within reach of the remaining budget fills at exactly B_i, and the first wall out of reach ends the pass;
4. the residual, if any, executes on the curve as one segment, and only this residual (plus the LP fee) ever touches the pricing reserve.

Fills clamp and never revert: a wall short of deliverable budget fills partially and the remainder keeps resting; a per-swap tick cap bounds gas, and anything beyond it simply stays for the next trade. Budget conservation holds exactly: the net input equals the sum of maker outputs plus the curve residual, and the taker's output equals the sum of wall fills plus the curve output. And since every wall filled is a price at least as good as the curve at that depth, resting liquidity weakly improves taker execution, strictly whenever a wall prices inside the curve: the deeper the book, the better the venue quotes.

5.3. *Why nobody skips a wall*

The engine's incentive argument is a single sentence with teeth: because the taker's fee is a function of the gross input and the pre-swap state only (Section 3), identical whether flow lands on walls or curve, filling a wall at B_i weakly dominates trading the curve at or beyond B_i, for the taker as well as for the protocol's fee take. A resting maker cannot be deliberately skipped at a profit, and a stale wall is a strictly better price than the curve, so the sweep's best-price-first order is incentive-compatible rather than merely imposed. The maker's residual risk, resting behind the spot while the market walks away, is compensated by the free cancel, immediate and unconditional for non-lent escrow (Section 9), and by the maker rebate below.

5.4. *Maker economics*

Takers pay the same fee everywhere; makers exchange fee-free at exactly their price, with zero slippage by construction. On top of that, governance can route a share of the LP fee earned on a wall fill back to the filled maker, at a rate differentiated by lend status. The rebate is computed at the static mid rate, which bounds the sum of rebates by the LP fee actually collected on every path, dynamic fee included. The natural split reads directly off the roles: a *non-lent* maker is an active market maker providing pure, senior, always-exitable depth, and receives most of their fill's LP fee; a *lent* maker is a passive lender already earning borrow interest, and receives a small share. Wash trading buys nothing (the protocol fee, when set, is always lost), and an unfilled wall earns nothing, so the rebate cannot be farmed by resting alone.

5.5. *Fill claims*

A fill's output is not paid into the maker's balance; it is *reserved*: booked into a fixed claim pot that ranks senior-most against the pair's physical balance (Section 9). Collecting a fill is therefore never gated, in any pool state, at any utilization (the one departure is the liquidity order below, whose recycled fills leave this class): the tokens were set aside at fill time, no later event can re-spend them, and the one guard left on the payout is a solvency backstop the reservation makes unreachable. Partial fills settle pro-rata and re-anchor the order, so collection is idempotent and drift-free; cancellation is immediate and unconditional for non-lent escrow, while for lent escrow it is a voluntary lent exit (Section 9), block-and-retry until the pool can front its grown value.

5.6. *Liquidity orders*

One further order type closes a loop. A *liquidity order* is a non-lent wall whose fills recycle: the output, rebate included, re-rests immediately as a wall on the opposite grid at the same tick's price. Price crosses the level downward, the order buys; price crosses back up, it sells; the round trip is value-neutral at the single static price by construction, and the maker's return is the rebate earned on every fill. Recycled proceeds leave the fill-claim class for plain order escrow: they keep quoting instead of waiting senior-most, and redeeming them is gated like a cancel rather than like a fill claim. It makes a standing two-sided quote a resident primitive, with cohort accounting (a per-tick pot whose re-flips convert value at B_i with no share issuance, so round trips cost no dilution) and failure-as-value flips that skip rather than revert mid-sweep, parking value that cannot re-rest in an idle leg that the cohort redeems pro-rata. Where the classic AMM forces liquidity to quote everywhere at once, a liquidity order, enabled per pair by governance and then

placed permissionlessly, lets capital quote both sides of one level inside the same solvency ledger as everything else.

6. Credit

The credit book has no pool of its own. It lends the pricing reserve, plus whatever escrow has opted in, and its entire design reduces to one question answered conservatively at every step: *how much debt can this pool carry such that liquidating it into its own curve remains solvent?*

6.1. Who lends

Liquidity providers lend by construction: the reserve that prices swaps is the credit book's first inventory, and borrowing does not move the swap price (borrowed amounts remain counted in the pricing reserve, with the outstanding principal tracked alongside). Three escrow classes may join them, each by explicit opt-in: resting order funds, loan collateral, and standalone single-asset supplies (these lent by definition); the pair is thereby a full single-asset lending venue in its own right, before any order or loan is placed. One LP position thus earns both of the pair's income streams at once, swap fees and lending interest, where siloed designs force a choice. Admission is binary, first-come, first-served, against a per-token budget proportional to the reserve, lent escrow $\leq \lambda R$ with λ the lend factor; a lent order additionally clears the per-tick lent-order depth gate of Section 6.5. A non-lent deposit earns nothing, is never lent out, and is never gated at exit; a lent deposit earns the supplier slice of interest and accepts the exit discipline of Section 9. All lent positions are held as *supply shares*, units of a pool whose value grows with the supply index L.

6.2. The two-bucket capacity

A token side's borrow capacity is

$$C \;=\; \beta_R\, R \;+\; \beta_E\, E, \tag{12}$$

where R is the pricing reserve, E the admitted lent escrow, and β_R, β_E the pair's borrow haircuts, the same on both token sides. The haircuts are not safety theater; their complements are *physical swap floors*: token amounts held against borrowing and fronting but consumed freely by swaps. A pool at maximal utilization can still trade whatever reserve its debt has not claimed (the curve leg is bounded by that unattributed slice alone, and wall fills stay live even when it is exhausted), and that asymmetry is deliberate, because trading is how prices correct and how liquidations clear. For the same reason $B \leq C$ is *not* an invariant: swaps may draw the reserve down until utilization exceeds one, and the system is designed to resolve that regime through the interest rate rather than forbid it. Utilization is $u = B/C$ with the bare borrowed principal in the numerator; collective fronts (Section 9) consume capacity but are deliberately excluded from u, since the protocol liquidates on price, not on a health factor, and pricing them into the rate would compound the receivables of price-safe borrowers without forcing any resolution.

6.3. Loans

A borrower chooses the liquidation tick, and the tick prices the loan. Collateral is sized so that, at the tick's price, it covers principal plus the liquidation penalty:

$$c \;=\; \big(1+\pi\big)\, q\, A_i, \tag{13}$$

rounded up, with q the borrowed amount, A_i the chosen tick's deterministic price, and π the liquidation penalty. Four gates guard the open: the tick must clear the pool's current price, itself read at the band edge that disfavors the borrower (P4), by strictly more than a configured tick buffer, so no loan is born nearly underwater; the amount must fit the global envelope (12) and the per-tick and per-range capacity of Section 6.5; the principal must stay strictly below the borrowed side's pricing reserve, the well-formedness bound that keeps the next gate's quote finite; and the collateral posted must be at least the input the curve would charge to buy the same output,

$$c \;\geq\; \mathrm{swapIn}(q), \tag{14}$$

so borrowing is never cheaper than swapping and borrow-and-default is never a discounted trade. A small origination fee, denominated in collateral and converted at the live reserve ratio, accrues to the protocol.

Debt is tracked at *entry basis*. With M the side's interest multiplier, a loan opened at M_{entry} contributes q/M_{entry} (rounded up) of scaled debt to its tick and to the side, and owes $q{\cdot}M_{\text{now}}/M_{\text{entry}}$ at repayment. The entry multiplier is frozen at creation, per-tick scaled debt is the exact sum of its loans' bases, and this exactness is what liquidation later leans on: closing a tick removes precisely the debt its loans carried, interest and all, with nothing orphaned in the aggregates (Section 8). Loans are transferable and repayable by delegation under typed signatures, so a position is itself an asset: it can be sold, or serviced by a third party, without handing over keys or collateral.

6.4. Interest

Each token side accrues independently through a multiplicative index,

$$M \;\leftarrow\; M\Big(1+\rho(u)\,\frac{\Delta t}{1\,\mathrm{yr}}\Big), \qquad \rho(u) = \begin{cases} \rho_0 + \rho_1 \dfrac{u}{u^*}, & u \leq u^*, \\[2ex] \rho_0 + \rho_1 + \rho_2 \dfrac{u-u^*}{1-u^*}, & u > u^*, \end{cases} \tag{15}$$

the familiar kinked utilization curve [5], ρ_0 a base rate, ρ_1 the climb to the kink u^*, ρ_2 the slope beyond it: cheap credit until the pool is leaned on, then a steep climb that makes over-utilization self-correcting, since past the kink repayment and fresh supply are both strongly paid to arrive. The rate is clamped at a hard ceiling, and pinned there when capacity is zero, so the violated regime prices fiercely but finitely. Accrued interest is split by a fixed order: a protocol slice φ is top-sliced, and the remainder credits the supply index,

$$\Delta L \;=\; (1-\varphi)\,\frac{\Sigma\,\Delta M}{S_{\text{sh}}}, \tag{16}$$

where Σ is the side's scaled debt and S_{sh} the total supply shares; LPs earn through the reserve's shares, lent escrow through its own. User credits round down at their final, per-share step and debts toward the pool round up (P2); L is monotone non-decreasing by construction, and M resets to one exactly when a side's debt clears, which doubles as the in-protocol recovery path after a defaulted side is swept clean.

6.5. *Shaping capacity across ticks*

Where (12) bounds the book, a *capacity model* shapes its distribution across price. The model is a plugin: a pure, stateless contract that each pair selects for itself, consulted by one static call at borrow time (and its twin at lent-order placement), returning gross per-tick and per-range capacity; the pair nets existing debt itself. Several models exist as drop-in alternatives, among them a curve-exact CryptoSwap model that reads the invariant's true depth, a conservative constant-product model, and an inverse model; the launch default is the CryptoSwap model. The constant-product model, the simplest to state, illustrates the shape: it prices depth as

$$\delta(p) = \sqrt{k/p}, \qquad \mathrm{cap}_i \;=\; \left(1 - \tfrac{1}{\sqrt{1.01}}\right) \delta\big(A_i\big) \cdot m_{\mathrm{tick}}, \tag{17}$$

with the per-range cap telescoping δ across each ten-tick group, truncated at the current price: debt may concentrate where the curve is deep enough to absorb its liquidation and must thin out where it is not. The tick and range multipliers are asymmetric on purpose: a single tick may lean far on the local depth, but a whole range cannot, so debt cannot form a uniform wall of adjacent maxed ticks. The model is governance-swappable atomically for both domains, and it sits inside a strict trust envelope: it can only shape distribution below the global envelope, which the pair enforces independently, and no exit path ever consults it, so a broken or malicious model can pause new borrowing but can neither mint capacity nor trap a single user. The constant-product depth deliberately under-states CryptoSwap's true depth near the peg, which is why the curve-exact model is the default; a pair that prefers the extra margin can swap the conservative model in atomically. This shaping is what turns the introduction's premise into arithmetic: the pair can underwrite the long tail because its credit book never promises more than its own curve stands ready to absorb.

7. Leverage

Leverage in the Everything Protocol is not a separate market; it is a composition. To lever long the collateral token: borrow the other token, swap it into collateral, post the proceeds, borrow again, all in one atomic transaction. The liquidation tick chosen at the open sets the effective leverage, the kinked interest rate (15) plays the role of a funding rate, and reaching the tick is the stop-out.

Two flash primitives fund the loop without upfront capital:

- **The pair's own flash facility.** Any caller may flash-borrow the pair's physical balance, fee-free, within one transaction. The interface is deliberately narrower than the ERC-3156 standard it resembles [8]: there is no receiver parameter (loan and callback bind to the caller, removing the standard's confused-deputy surface), and repayment is pulled by the pair rather than checked as an inflow, so ambient transfers cannot spoof it. The facility composes: the callback may swap, borrow, and repay on the same pair (only a nested flash is refused, and each composed operation keeps its own re-entrancy guard). What makes the open composition safe is the accounting: every payout base reads the physical balance *plus* the outstanding flash, so a flash in flight moves no solvency decision anywhere in the pair.

- **The router's flash macros.** A command router in the style of the universal dispatcher composes multi-step actions: multi-hop swaps, liquidity management, and one-click leverage that flash-borrows the debt leg externally (at zero fee) [6], swaps it through an external aggregation route where that is cheaper, opens the loan with the pair pulling collateral from the router, and repays the flash with the borrowed tokens, with the deleveraging mirror running the loop backwards under typed-signature delegation. The net effect is that a leveraged position opens and closes in one transaction, from margin alone, at the best execution available.

No hard leverage cap is imposed; the bound is economic. Each iteration of the loop must post collateral sized by the penalty (13) and displaced by the tick buffer, with the swap-cost floor beneath it ruling out any cheaper synthetic route, and the resulting geometric series exhausts itself after a handful of turns of the trader's own capital, the exact multiple set by the penalty, the buffer, and the swap route. The cap emerges from the same inequalities that price everything else, which is the design's aesthetic in miniature: no special case, just the same solvency arithmetic seen from another side.

8. Liquidation

Liquidation is not an auxiliary process that hopefully runs; it is the pool's homeostasis, executed at the top of every book-moving operation. The cascade design answers three questions: how to close many positions at once, who absorbs a shortfall, and how to guarantee the process itself can never wedge the pool.

8.1. The cascade

Loans are indexed by liquidation tick, so detection is geometric: every populated tick at or below the current lending price is underwater, no per-loan check required. The cascade walks the tick bitmap from the far end, closing tick after tick, with both sides' thresholds frozen at the entry of the pass; a second full pass then absorbs whatever the first pass's seizures shifted, and two passes suffice, an argument the specification makes explicit. Two details carry the correctness. The current tick is computed from the *band-priced* lending price, rounded down, so a loan whose liquidation level sits strictly above the price is never touched; and each closed tick removes its debt at *entry basis*, the exact scaled principal its loans carried, so the side's aggregates never orphan a closed loan's interest, and the protocol's fee pot is clamped down to its realized cash plus the cut on surviving debt, leaving no phantom income behind.

Closing a tick is $O(1)$ in its population: the tick's version is bumped (instantly invalidating every loan keyed to the old version), its collateral moves from escrow into the pricing reserve (non-lent collateral as a fresh share-backed inflow, lent collateral by re-homing its existing shares, which is what carries the forfeited yield to the liquidity providers), and its principal leaves the borrowed reserve. Termination is a construction, not a hope: the stabilization is a bounded sweep over a finite bitmap whose every closed tick clears its bit, and a permissionless bounded entry point drains arbitrarily large backlogs in capped, gas-safe chunks. The engine's cost profile is flat in a crash, precisely when per-position liquidation engines degrade and cascades stall elsewhere.

The repeg is frozen throughout (Section 3.4): a seizure is a discrete reserve move, not a trade along the invariant, and it must not re-center the price scale, or triggering liquidations would

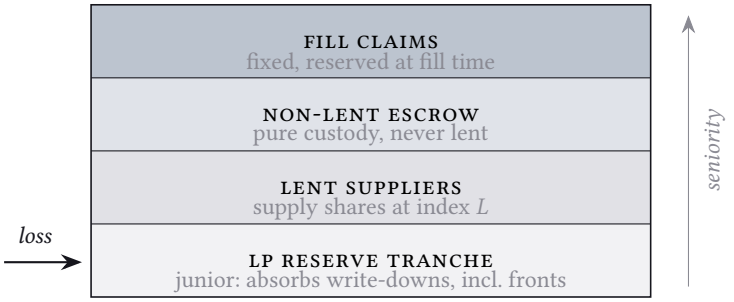


FIGURE 5. The loss waterfall. A liquidation shortfall is written down against the junior LP reserve tranche first, outstanding collective fronts included. The supply index L is never haircut, non-lent escrow is untouchable by construction, and fill claims are reserved senior-most at fill time. A write-down the junior tranche cannot absorb never settles: the tick defers instead, so no loss is ever booked against a senior class.

become a lever on the peg. Instead the cascade re-baselines the curve's profit checkpoint (a liquidation may legitimately lower the virtual price, so the non-decrease gate is suspended for this transition); the EMA integrates the elapsed window at the pre-seizure held price before that held price is re-anchored to the post-seizure spot, so the seizure never rewrites oracle history: the elapsed window is credited to the pre-seizure price, and the new spot weights only the windows after the commit. Price discovery resumes with the next real trade.

8.2. The waterfall

Seized collateral rarely covers grown debt exactly; the difference must land somewhere, and the EVERYTHING Protocol fixes where by construction rather than by discretion (Figure 5). After a cascade, the borrowed side's reserve claim is re-based to what physically backs it: with $\tilde{R}$ the post-seizure pricing reserve net of the fronts still outstanding (an advance is written off first when the debt behind it has died) and $I_{\text{sup}} = (1-\varphi)\, I$ the supplier slice of the interest still owed by surviving loans, the LP reserve shares are written down to

$$\text{kept} \;=\; \min\Big(\text{shares},\; \big\lceil \big(\tilde{R} + I_{\text{sup}}\big) / \, L \big\rceil \Big), \tag{18}$$

and the burned difference is the loss, absorbed by liquidity providers. The supply index L is never reduced: lent suppliers and filled makers do not pay for bad debt through their index, ever. The tranche is also the hard boundary of what settles at all: the cascade closes only the *absorbable prefix* of the underwater book, and a tick whose write-down would exceed what the junior tranche can absorb is deferred, not booked (Section 8.3). There is no unbooked bad debt and no senior deficit, by construction rather than by disclosure: a loss either fits the junior tranche and is written down, or the tick that would create it stays pending. In the wake of a large default the buy price decays toward the spot, which such a default leaves extremely low; the decay works as a slow on-curve auction of the seized collateral, the pool offering it at a price that improves for takers block by block, so liquidity providers sell the collateral at the best price the market will pay, and the proceeds replenish the reserve that absorbs the ticks still pending. Collective fronts outstanding at liquidation time are junior too: the advance the pool made against unrealized yield is written off at the LP tranche's expense, tranche-neutrally for everyone senior.

8.3. *Never bricked*

A liquidation engine that reverts is a denial-of-service against the whole pair, because every book-moving operation runs it first. The failure mode is subtle: a defaulted side can leave reserves so drained that the price solve itself has no answer, or a write-down larger than the junior tranche can absorb, and a naive implementation reverts in the preamble forever, bricking the pool permanently. The Everything Protocol's answer is *liquidation gating* (P3): the cascade never reverts on its own state. If the pool is too drained to price a tick, or the reserve cannot absorb its write-down, that tick is skipped and left in place, and every later transaction retries it in its preamble until it clears. While such ticks are pending, the operations that heal keep working: swaps, repayments, order cancels and fill collection all run, and their inflow is what refills the reserve. Mint, burn, borrow, leverage and new lent deposits revert with a typed error instead, so no one can enter or exit the junior tranche against an unsettled book. There is no terminal state, no timer, and no stored backlog: the pool heals through its own trading, or waits. Meanwhile the buy price decays toward the spot (Section 8.2), so the collateral behind the pending ticks is on offer to the market at a price that improves until someone takes it. The only accepted dead end is a token that stops transferring, which freezes the pair; such a pair is replaced by a fresh deployment, never reset in place. No panic, no unbounded revert, no permanent freeze: the pool's worst reachable state is one that can still be traded out of and healed.

9. Settlement and Solvency

The previous sections each produced obligations: LP claims on the reserve, suppliers' grown shares, makers' escrow and fill proceeds, borrowers' collateral. This section is the protocol's balance sheet: who is paid, from what, in which order, and what happens when the tokens are not all home at once.

9.1. *Cash first*

The Everything Protocol refuses the IOU as a settlement instrument. Every closing operation, repaying a loan, canceling an order, withdrawing a supply, burning LP shares, pays *real tokens, in full, immediately* (un-lending collateral crystallizes its grown value in escrow under the same physical-presence rule). The difficulty is that a lent deposit's yield may not be physically present at exit time (it is owed by borrowers who have not yet repaid). Rather than minting a claim on the future, the pool *fronts* the missing slice from its collective liquidity, exactly as a money market pays a withdrawal out of pooled cash, and unwinds the advance as interest is actually paid. A new advance ΔF, on top of the fronts F already outstanding, is admitted only under three simultaneous ceilings:

$$\Delta F \;\le\; \min\Big(\; \underbrace{C-(B+F)}_{\text{capacity headroom}}\;,\quad \underbrace{I_{\text{sup}}-F}_{\text{supplier-net receivable}}\;,\quad \underbrace{\Phi}_{\text{physical collective}}\;\Big), \tag{19}$$

i.e. the advance must fit the same envelope as a borrow (the swap floors are never consumed by an advance), must never exceed the interest actually receivable net of the protocol's already-promised slice, measured on a cash-anchored basis with the aggregate debt rounded down (deliberately a rounding below the accrual basis that (18) reads as I_{sup}, so an advance is never sized against a

wei no repayment will deliver), and must be physically present after reserving every non-lent floor and senior claim. An exit whose front cannot be admitted reverts intact, to be retried when liquidity returns: *block-and-retry*, the direct price of refusing the IOU. The worst case is a delay, never a devaluation. Designs that instead mint a claim socialize the gap silently; here the cost of illiquidity is explicit, temporary, and interest-bearing while it lasts.

The capacity-gate structure is deliberately asymmetric, and the asymmetry is the user-facing contract:

OPERATION	GATED?	FAILURE MODE
collect a fill claim	never	none: reserved at fill time
non-lent exit (any kind)	never	none: pure custody
liquidation	never	none: bypasses every gate
in-sweep fill of a lent wall	clamped	partial fill, never a revert
voluntary lent exit	capacity-gated	revert, retry when healed

A voluntary lent exit (un-lending collateral, withdrawing a supply, canceling a lent order, repaying a loan whose collateral is lent, burning LP shares) must leave the post-exit book inside the two-bucket capacity (12), debt and fronts included, the burn checking the same invariant in an algebraically rearranged form; a worst-case exit-capacity view lets a user size a partial exit that fits. In the violated regime, when swaps have drawn the reserve below the outstanding book, every lent exit blocks until capacity returns, and the escape hatches are the economic ones: the kinked rate (15) is already far past its kink, paying repayment and fresh supply to arrive; anyone may lend or mint to unblock a side, including the exiting user (self-rescue); and liquidation, never gated, keeps clearing the book meanwhile.

9.2. Two ledgers, one balance

Each token side keeps two parallel books. The *junior* book is the pricing reserve R mirrored by a share ledger at the supply index L; the *senior* book is user escrow: collateral, resting order funds, supplies, each held at explicit floors, with fill claims senior-most above them all. The bridge between books is a family of one-sided inequalities, of which three carry the architecture:

INVARIANT 1 (Wedge). *On each side, the reserve share ledger covers the pricing reserve:* $\text{shares} \cdot L + \text{dust} \geq R$. *Swap inflows credit shares rounding down, outflows burn rounding up, so the wedge survives every trade, mint, burn, and seizure.*

INVARIANT 2 (Fill seniority). *The fill-claim payout base is the physical balance minus only the netted reserve claim and realized protocol fees; it reserves no escrow above it. In every reachable state it covers the outstanding fill claims, so collecting a fill can never compete with, or lose to, any junior class.*

INVARIANT 3 (No unbooked bad debt). *A liquidation settles only what the junior tranche can absorb: a tick whose write-down would exceed the tranche is deferred, never booked against a senior class (Section 8.3). Senior obligations stay covered by physical balance in every reachable state, and deferred ticks clear as trading inflows rebuild the reserve.*

Every payout base in the system is built the same way: the physical balance minus everything senior to the claim being paid, floored at zero. Paying any class can therefore never dip into a senior class's backing, by arithmetic rather than by policy; even the permissionless protocol-fee

distribution is bounded the same way, transferring only the realized, cash-anchored fee slice, never the booked-but-uncashed cut, and never user backing. The rounding doctrine (P2) orients every division so that dust accumulates toward the pool, and dedicated dust ledgers fold it back into the junior tranche rather than letting it leak.

10. Architecture

10.1. One contract per pair

A pair is a single contract: it owns the two books, every entry point, and the LP-token surface, and it is itself the LP token, with a minimum liquidity burn at genesis. Its storage evolves append-only across upgrades, so an upgrade can add state but never reshape what already sits under users' positions.

10.2. Factory and governance

A factory, itself upgradeable, deploys every pair behind one shared implementation: a single upgrade migrates every pair atomically. The factory owns every economic lever, batched per-pair: fees and the dynamic-fee shape, maker rebates, the interest model, the two-bucket fractions, the lend budget, liquidation penalty and tick buffer, per-tick borrow multipliers, curve parameter ramps, repeg parameters, band decay, order knobs, the spread-guard tolerance and the borrow kill-switch, and the capacity-model slot, whose hot-swap re-prices the borrow and order domains in one atomic step. Setters that change a rate denominator carry a settle-first flag that closes the pending accrual window at the old capacity before the change lands. The emergency stop is architectural: upgrading the factory to an empty implementation freezes every parameter and pair-creation path (the pairs' setters answer only to the factory) while user funds and exits keep working directly at the pair; the pair implementation's and the factory's own upgrade hatches necessarily stay live. Pair creation is permissioned at this stage of the protocol's life, so that every market opens with parameters already tuned to it, not tuned after the fact.

10.3. Periphery

A command router composes user journeys (multi-hop swaps across pairs, liquidity management, flash-funded leverage and deleverage, token plumbing) as byte-coded command sequences with per-command revert tolerance, holding no custody between transactions. A stateless quoter reuses the execution libraries rather than re-implementing them: an exact-input quote is bit-identical to the swap it predicts on unchanged state, and exact-output or multi-hop quotes are curve-only bounds.

10.4. Engineering discipline

The specification names its invariants, per subsystem, as first-class objects (the wedge, the budget conservations, the seniority floors, the termination and never-brick guarantees among them). The mathematical *why* lives in documents like this one; the code carries the *what*, and the invariants are the contract between them.

11. Risks and Honest Limitations

A whitepaper that lists only strengths is marketing. The following properties are deliberate trade-offs, and users should price them.

- **Lent exits can wait.** A voluntary exit of lent funds is capacity-gated with no guaranteed unblocking time. The interest curve makes prolonged blockage expensive for borrowers and lucrative for rescuers, but the guarantee is economic, not temporal, per (12) and (19). Non-lent deposits never share this risk, which is precisely why the lend flag is an explicit opt-in.
- **LPs are the insurance fund.** Bad debt, including written-off fronts, lands on the junior LP tranche (18). There is no external backstop, and a catastrophic default defers the liquidation ticks the tranche cannot absorb, gating entry to and exit from the junior tranche until trading inflows rebuild it. LPs are compensated for this junior position by earning the LP share of the pair's swap-fee and borrower-interest streams.
- **Liquidation marks at the band, not at realizable value.** Seized collateral is valued at the band price, not at the price its own sale into the curve will realize; the capacity model (Section 6.5) exists to keep that gap small, and the post-cascade decay of the buy price toward the spot (Section 8.2) lets the pool sell what it seized at the best price the market will pay, but the residual slippage of a large seizure remains junior-tranche risk.
- **Governance can change the rules.** The factory, the pair implementation, and the modules are upgradeable, and parameters are live levers. The protocol's trust model at this stage includes its operator; an immutable mode is explicitly not offered in the current version. The exposure is bounded by the architecture rather than by promise: the emergency stop freezes every lever while user funds and exits keep working directly at the pair, and no exit path consults the one pluggable module.
- **Parameters are levers, not constants.** Every economic parameter is a per-pair governance setting, tuned by simulation per market; no argument in this paper depends on a particular tuning, but several parameters interact: the penalty-versus-worst-case-fee coupling enforced as a hard bound at the factory, the penalty-versus-buffer relation left as an economic tuning choice.
- **The band trades latency for safety.** Credit prices lag the market by up to the decay window under fast moves. This is the choice that makes single-block manipulation worthless, and its cost is that honest liquidations also wait out the window.
- **No automated position exits.** A leveraged position closes manually or by liquidation, with its penalty; the protocol attaches no stop-loss and no take-profit to loans (the upside exit is a manual close, or a resting limit order the owner collects).
- **Token assumptions.** Fee-on-transfer and rebasing tokens are unsupported in the current version. Rebasing support is planned, alongside the real-world-asset listings that motivate it; fee-on-transfer tokens remain out of scope.

12. Conclusion

The premise of the Everything Protocol is that an exchange, a money market, and a leverage venue are not three businesses but three uses of one balance sheet, and that the boundaries

between them, in today's DeFi, exist for the protocols' convenience rather than the capital's. Merging them is easy to say and unforgiving to do: the whole difficulty concentrates in the solvency accounting, where one reserve must simultaneously honor a pricing curve, a credit book, an order escrow, and a queue of claims, under adversarial sequencing, with no oracle to arbitrate.

The design answers with structure rather than trust: a price band that cannot be moved faster than it decays; a tick grid that makes both debt and orders aggregate objects; a capacity rule derived from the depth that will absorb its own liquidation; settlement that pays cash now and ranks every claim explicitly; losses that fall junior-first by arithmetic; and a liquidation engine built to survive its own worst case. Each mechanism is small enough to state in a formula, and the protocol is the closure of those formulas under composition.

One pool, priced by its own trades, lending its own depth, filling its own orders, absorbing its own failures: liquidity with nothing idle and nothing hidden. The name is the thesis: everything, from one reserve.

References

[1] M. Egorov and Curve Finance. *Automatic market-making with dynamic peg.* Curve Finance, 2021. Reference implementation *twocrypto-ng*.

[2] M. Egorov. *StableSwap: efficient mechanism for stablecoin liquidity.* Curve Finance, 2019.

[3] H. Adams, N. Zinsmeister, and D. Robinson. *Uniswap v2 Core.* 2020.

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

[5] Aave. *Aave Protocol Whitepaper.* 2020.

[6] Morpho Labs. *Morpho Blue.* 2023.

[7] M. Egorov and Curve Finance. *Curve stablecoin design (LLAMMA).* 2022.

[8] A. Cuesta Cañada, F. Kobayashi, fubuloubu, and A. Williams. *ERC-3156: Flash loans.* Ethereum Improvement Proposals, 2020.