Prop RFQ: Proprietary Request for Quote as Pressure-Aware Exit Pricing for Redeemable Real-World Asset Tokens

Marian-Daniel Rolník
Brno University of Technology
Faculty of Information Technology
Brno, Czech Republic
irolnik@fit.vut.cz

Ivan Homoliak
Brno University of Technology
Faculty of Information Technology
Brno, Czech Republic
Slovak University of Technology in Bratislava
Bratislava, Slovakia
homoliak@fit.vut.cz

Theodore Georgas
OnRe
Bermuda
theodore@onre.finance

Abstract

Redeemable real-world asset tokens can trade onchain faster than their backing assets can be sold or settled. An immediate-exit facility must therefore distinguish the token's reported net asset value (NAV) from the reserve available to pay sellers. Charging larger exits more can preserve that reserve, but a holder may avoid part of the charge by splitting the sale.

We propose Proprietary Request for Quote (Prop RFQ), an onchain exit facility that combines NAV and available liquidity with an order-size curve, decaying sell pressure, and a count-based cadence response. A model-specific identity-free exit-pricing trilemma explains why shared history also affects independent small sellers. We evaluate the Solana implementation using deterministic workloads, ablations, and a 15,120-configuration sweep. At the reference parameters, cadence reduces aggregate split advantage by 9.67% against matched pressure-only pricing, improves 34 of 40 split workloads, and raises worst-case reserve remaining from 10.49% to 17.66%. No compared simple policy or ablation dominates the reference configuration on common metrics. However, splitting can still improve payouts, and low-value sells enable cheap cadence griefing.

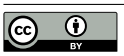
CCS Concepts

• **Applied computing** → **Electronic funds transfer**; • **Security and privacy** → *Economics of security and privacy*.

Keywords

request for quote, automated market maker, real-world assets, redemption liquidity, order splitting, decentralized finance

ACM Reference Format:
Marian-Daniel Rolník, Ivan Homoliak, and Theodore Georgas. 2026. Prop RFQ: Proprietary Request for Quote as Pressure-Aware Exit Pricing for Redeemable Real-World Asset Tokens. In *6th Workshop on Decentralized Finance and Security (DeFI '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3847164.3856385



This work is licensed under a Creative Commons Attribution 4.0 International License.
DeFI '26, The Hague, Netherlands
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-3025-2/2026/11
https://doi.org/10.1145/3847164.3856385

1 Introduction

Tokenization speeds up financial-claim transfers [6]. The backing loans, securities, and contractual cash flows may settle much more slowly than the token trades [9]. For redeemable real-world asset (RWA) tokens, reported net asset value (NAV) measures value rather than cash available for immediate exits.

Immediate exits must balance reserve preservation with prices for small holders. Paying at NAV can drain the reserve, while uniform large discounts penalize small holders. Discounts based only on each sale's size can invite splitting across transactions or accounts. The rule must therefore account for repeated selling without identifying owners.

We ask whether pricing can limit reserve depletion and splitting gains while preserving useful isolated small exits.

Our Work. We developed Proprietary Request for Quote (Prop RFQ) with OnRe, where it is deployed in production for the ONyc token. ONyc represents a proportional claim on a segregated portfolio used to underwrite short-duration insurance and reinsurance contracts [24]. In our setting, the issuer reports NAV and total value locked (TVL), while immediate exits draw on a separate settlement vault and primary redemption follows the slower issuer process. The experiments evaluate a frozen implementation under controlled workloads, not production trading outcomes.

Prop RFQ discounts the sale's NAV value using order size, available liquidity, and recent selling. Its shared state records sold value and successful sell count; splitting alone does not reset these signals.

Contributions. The paper makes three contributions:

- We model and implement pressure-aware RWA exits that distinguish reported NAV from funded settlement liquidity.
- We evaluate split mitigation, reserve effects, and cheap-sell griefing against simple policies and ablations under fixed adversarial workloads and a parameter sweep.
- We state a model-specific identity-free exit-pricing trilemma that explains the conflict between penalizing concentrated exits and protecting independent small sellers.

2 Background and Preliminaries

An RWA token links an onchain claim to an asset or contractual cash flow outside the blockchain [9]. We study fungible portfolio tokens for which the issuer publishes a NAV and allows holders to request settlement assets.

2.1 Redeemable RWA Tokens

NAV per token expresses the reported net portfolio value per outstanding token. Liquid reserves answer a different question: how much can the protocol pay before the remaining backing matures, settles, or is sold?

For a simplified example, suppose the reported net portfolio value is $10 million, including a $1 million liquid reserve, and there are 10 million tokens. The NAV per token is then $1, but only $1 million is available for immediate payment. A holder selling 1,000 tokens has a NAV claim of $1,000, whereas a sale of one million tokens would consume the entire reserve if paid at NAV. The portfolio need not have lost value for this liquidity constraint to bind; another immediate payment requires more liquid assets.

2.2 Immediate Exit Mechanisms

An RWA holder can wait for primary redemption through the issuer, sell to another holder, or use a protocol liquidity facility. The first route depends on the issuer's settlement process, the second on finding a buyer, and the third on settlement assets already held in the facility.

An automated market maker produces executable onchain quotes from a public pricing rule. A quote specifies the settlement amount offered for a given token sale; a haircut is the reduction from that sale's NAV value. For a constant-function market maker, reserves and a trading function determine the allowed exchanges [2]. In our RWA setting, the quote must relate the reported NAV to the smaller settlement reserve available for immediate exits.

2.3 Order Splitting and Shared State

If larger transactions receive larger haircuts, a seller may improve its total payout by splitting one exit into smaller calls. This incentive is especially direct when each call prices only its own size and forgets earlier sales. AMM research treats indifference to this form of transaction splitting as a separate property of the pricing rule [3].

Shared state counters this reset by letting earlier sales affect later quotes, although unrelated sellers then inherit part of that price impact. Yokoo et al. [28] show that one bidder can participate through several identifiers. The same ambiguity prevents our mechanism from assigning several accounts to one owner.

3 Problem Definition

We consider an issuer-managed RWA token with public supply, a periodically authorized NAV, and a funded reserve in one settlement asset. The pricing rule uses those inputs to serve immediate exits; it neither replaces primary redemption nor verifies portfolio value or solvency.

3.1 System Scope and Exit Options

Within this funded path, fixed-NAV pricing exposes the reserve to concentrated exits, uniform discounts also penalize small holders, and discounts based only on each sale's size can invite splitting. Accounting for recent flow offers another choice, but requires deciding how much earlier selling should affect an unrelated holder.

3.2 Identity-Free Exit-Pricing Trilemma

The trilemma concerns rules that give concentrated exits larger haircuts than they allow for each independent small owner. All exit sizes below are measured by their settlement value at NAV. Fix a small-exit threshold $q_0 > 0$, a price tolerance $0 \le \varepsilon < 1$, and an uninterrupted trace with no buys, parameter changes, resets, or time decay. An ideal immediate-exit mechanism would provide three guarantees:

(1) Identity-free permissionless access: any valid account may exit without identity verification, and neither eligibility nor pricing links accounts to economic owners.
(2) Split-order resistance: from the same initial state, any consecutive partition $Q = \sum_i q_i$ receives no more in total than one direct exit of size Q.
(3) Favorable small exits: each independent owner with total exit $q \le q_0$ receives at least $(1-\varepsilon)q$, even after earlier exits.

Suppose a concentrated direct exit $Q > q_0$ receives at most $(1-\eta)Q$, with $\eta > \varepsilon$. Now partition Q into parts $q_i \le q_0$. If those parts came from independent owners, favorable small-exit pricing would pay at least $(1-\varepsilon)Q$ in total. Without owner linking, a single holder can reproduce the same trace and receive more by splitting than by exiting directly. The absolute payout advantage is at least

$$(\eta - \varepsilon)Q. \tag{1}$$

The argument relies on the strict difference $\eta > \varepsilon$ and on the direct and partitioned exits being executable. It does not rule out uniform pricing, which gives up the distinction between small and concentrated exits. Under these assumptions, consider each pair of guarantees.

Permissionless access and split-order resistance are compatible with cumulative path-integrated pricing. Let x be cumulative sold value and $p(x)$ a common non-increasing marginal-price schedule for the uninterrupted trace. The payout for an exit of size q is

$$\Pi(x, q) = \int_x^{x+q} p(u)\,\mathrm{d}u. \tag{2}$$

For each consecutive part, let $x_i = x + \sum_{j<i} q_j$. The intervals meet without overlap, so $\sum_i \Pi(x_i, q_i) = \Pi(x, Q)$ in real arithmetic. However, a later independent small seller receives the lower marginal price created by earlier flow.

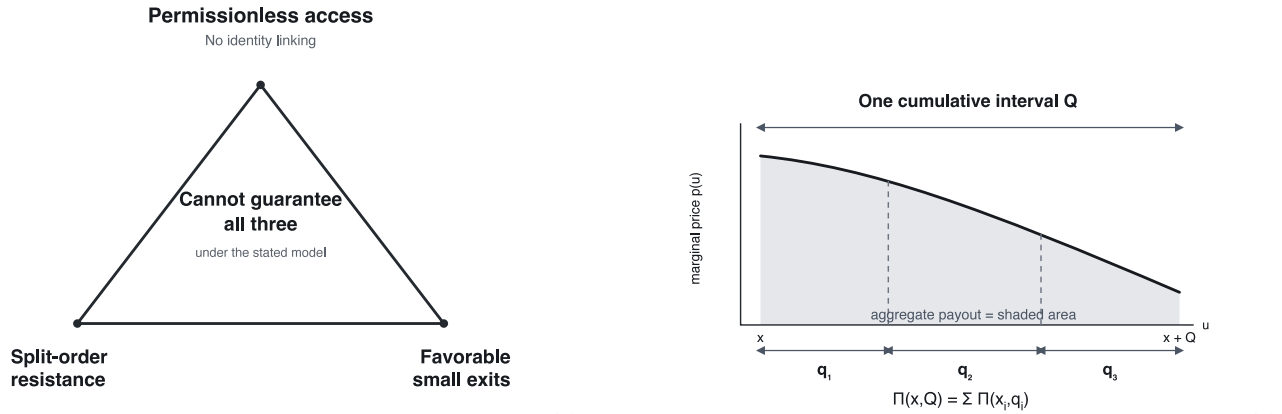
Permissionless access and favorable small exits protect both independent sellers and a holder reproducing their trace across accounts, violating split resistance when the direct exit receives a larger haircut.

Split-order resistance and favorable small exits require linking transactions to an order or economic owner, weakening identity-free access.

Position of This Work. The model-specific argument in Figure 1 is not a general impossibility result for all AMMs. Our mechanism keeps permissionless access and uses shared history to reduce split gains, guaranteeing neither split resistance nor near-NAV prices after every exit sequence.

3.3 Design Options and Objectives

We assess four objectives for shared state without owner linking:



Each pairwise choice weakens the third guarantee. Consecutive partitions cover the same area as one order.

Figure 1: Pairwise identity-free exit-pricing choices. Panel (a) summarizes the trilemma; Panel (b) shows why a cumulative integral neutralizes consecutive splitting in real arithmetic, subject to implementation rounding and discrete transitions.

(1) Funded execution: a successful sale never pays more settlement assets than the vault holds.
(2) Useful small exits: an isolated small sale should remain close to NAV both ordinarily and after stress.
(3) Reserve preservation: a concentrated exit should not consume the reserve at the price offered to a routine small sale.
(4) Split mitigation: splitting one exit should gain less than pricing that ignores earlier transactions.

Table 2 compares these guarantees with documented exit mechanisms and proprietary spot AMMs.

3.4 Threat Model

The adversary may create unlimited accounts, split an exit into any number of transactions, choose sizes and timing, cross epoch boundaries, and observe all public state. It may also use low-value sells to advance cadence or buy before selling to reduce recorded pressure. Transaction ordering and user-selected minimum outputs remain part of the environment.

We trust the authorized NAV and TVL inputs, the onchain vault balance, the settlement asset, Solana execution, and the parameters fixed for each experiment. Omniscient [14] provides an oracle architecture for delivering offchain API data to smart contracts; our model takes the authorized inputs as given. We do not assume that accounts can be linked to one economic owner. The model establishes neither issuer solvency nor portfolio value and does not coordinate queued redemptions or create liquidity.

4 Proposed Approach

The mechanism combines a funded-liquidity bound with pressure-aware pricing and a recovery rule.

4.1 Overview

For each configured token pair, Prop RFQ starts from the sale's issuer-authorized NAV value and pays a discounted amount from the redemption vault. Figure 2 places this funded facility beside primary redemption and peer-to-peer sale. Its pricing scale, called the funded-liquidity wall, depends on the vault balance, a reserve target, and recent sell pressure. Dividing the current order value by the resulting effective liquidity gives its utilization.

Two shared signals carry history across transactions: sell pressure records sold value after funded buys, while cadence counts successful sells within a fixed time interval, or epoch. Since neither signal identifies owners, both also affect unrelated sellers who arrive later.

4.2 Funded Reserve

Let q denote the current sale's NAV value after the input fee, expressed in settlement-asset units, and let L_t be the vault balance. The configured reserve target R_t is the reported token TVL multiplied by the reserve fraction $\tau \in [0, 1]$ and converted into settlement-asset units. The hard reserve is

$$H_t = \begin{cases} L_t, & \tau = 0, \\ \min(L_t, R_t), & \tau > 0. \end{cases} \tag{3}$$

Thus, a \$1 million vault with a \$800,000 target gives $H_t = \$800,000$, but a \$600,000 vault gives only $H_t = \$600,000$. The target can restrict the pricing liquidity; it cannot supply missing cash.

4.3 Sell-Pressure State

To retain recent selling, each pair records current-epoch sell value S_t, funded buy value B_t, previous-epoch net sell pressure P_t, successful sell count c_t, and epoch start s_t. At quote time t, set $N_t = \max(S_t - B_t, 0)$ and $\Delta = t - s_t$. The quote uses pressure

$$V_t(q) = q + \begin{cases} N_t + P_t\left(1 - \frac{\Delta}{T}\right), & 0 \le \Delta < T, \\ N_t\left(2 - \frac{\Delta}{T}\right), & T \le \Delta < 2T, \\ 0, & \Delta \ge 2T, \end{cases} \tag{4}$$

where T is the epoch duration. The current order enters $V_t(q)$ before pricing, while older selling gradually loses its effect through pressure decay. Current-epoch net selling remains fully counted

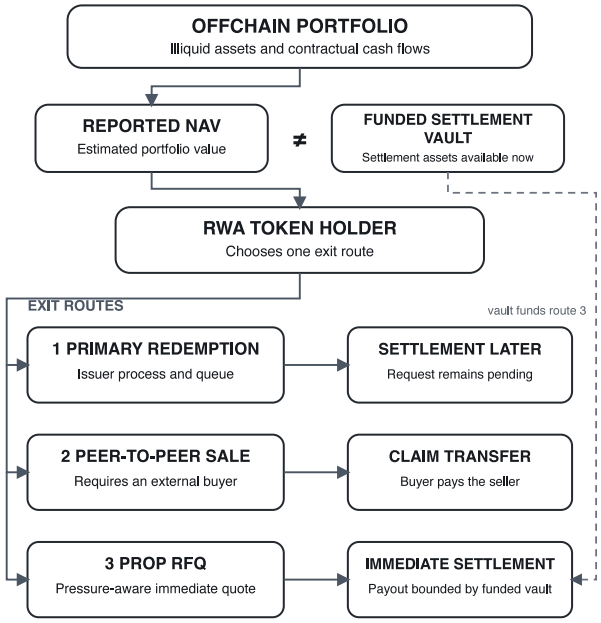


Figure 2: The proposed pressure-aware exit mechanism beside primary redemption and peer-to-peer sale.

until the next boundary and then decays linearly over the following epoch. This allows prices to recover without erasing the liquidity constraint: decay changes the pricing state but does not replenish the vault.

Pressure lowers the dynamic wall to

$$W_t(q) = \frac{L_t}{1 + \sigma V_t(q)/L_t}, \tag{5}$$

where σ is the wall sensitivity. We then use the smaller of this dynamic wall and the hard reserve as effective liquidity, $E_t(q) = \min(W_t(q), H_t)$, and compute utilization $u_t(q) = q/E_t(q)$. The wall controls the pricing scale rather than imposing a separate rejection threshold. For example, with $L_t = H_t = \$1$ million, $V_t(q) = \$100,000$, and $\sigma = 2$, the wall is approximately \$833,333. An order with $q = \$100,000$ then has utilization $u_t(q) = 0.12$ instead of 0.10 without the pressure adjustment.

The model assumes nonnegative real values, $t \geq s_t$, positive T and σ, and a valid sell with $q > 0$, $L_t > 0$, and $H_t > 0$. These conditions keep the pricing denominators positive; in the implementation, $2T$ and $s_t + T$ must also fit the signed timestamp representation.

4.4 Endpoint and Cadence Pricing

Having related the order size to available liquidity, we compute an endpoint haircut:

$$h_b(u) = h_{peg} u^e, \tag{6}$$

where h_{peg} is the haircut at unit utilization and e is the curve exponent. We call this endpoint pricing because one haircut, evaluated at the current order's utilization, applies to the whole order. It differs from the cumulative integral in Equation 2, which sums marginal prices across the sold interval. The reference exponent $e = 2.5$ gives a convex curve: with $h_{peg} = 0.07$, the endpoint haircut is about 0.022% at $u = 0.1$ but 7% at $u = 1$. This reference design choice keeps the size component small at low utilization; it is not an estimated economic constant. The configured domains are $h_{peg} \in [0, 1]$ and $e \in \{0.1, 0.2, \ldots, 10\}$.

The endpoint curve alone can favor splitting, so cadence adds a second haircut that grows with the count of prior successful sells. Let the positive integer k be the count at which this response reaches its maximum strength $y_{max} \in \{0, 0.1, \ldots, 5\}$. Before the current sell, the wave height is

$$y_t = y_{max} \min\left(1, \frac{c_t}{k}\right), \tag{7}$$

For $x = \min(u, 1)$, the cadence target is

$$g(x) = \frac{8x}{1 + 7x}, \qquad h_c(u, c_t) = \min\left(1, \frac{g(x) y_t}{3}\right). \tag{8}$$

The final output is

$$h(u, c_t) = \max(h_b(u), h_c(u, c_t)), \qquad a_t(q) = q \max(0, 1 - h(u, c_t)). \tag{9}$$

For $k = 20$ and $y_{max} = 1$, the cadence strength rises from zero to 0.5 after ten successful sells and reaches one after twenty. Because the current sale increments the count only after execution, the first sale in an epoch has no cadence haircut. The concave function $g(x)$ rises steeply from $g(0) = 0$ to $g(1) = 1$, making this response relevant below unit utilization; division by three sets the uncapped target at $u = 1$ to $y_t/3$. These fixed shape choices are visible in Panel (a) of Figure 3; the sweep varies cadence strength and threshold without claiming that the shape is optimal.

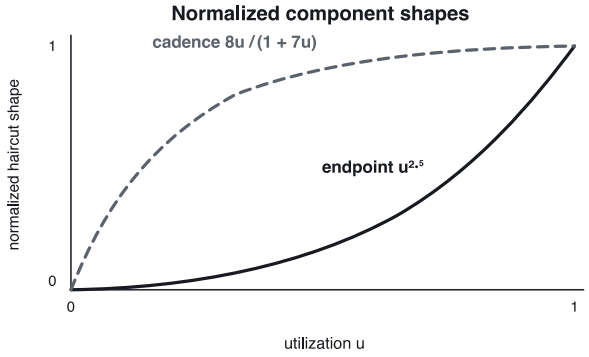
The maximum in Equation 9 selects the stronger haircut rather than adding the two discounts. When utilization exceeds one, the cadence input stays at $x = 1$ while the endpoint term continues to grow; an endpoint haircut of at least one reduces output to zero.

4.5 State Transitions and Recovery

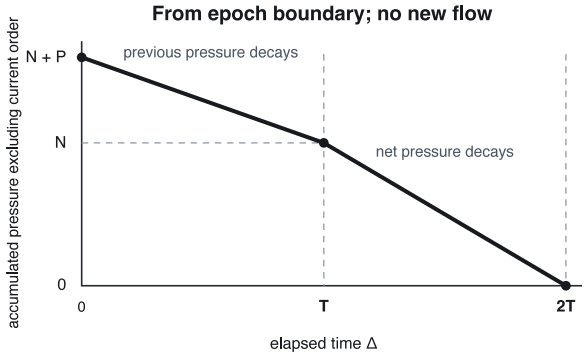
A successful sell increases the raw sell value and count, then removes the payout from L_t. A buy increases B_t only by the amount added to the vault, and net pressure never becomes negative. At the first epoch boundary, current net pressure moves into previous pressure and the current fields reset. That pressure decays during the second epoch and then reaches zero. Panel (b) of Figure 3 plots recovery without new flow; Section 6 checks that delayed quotes and executed trades follow this rule.

4.6 Adversarial Selling and Griefing

A successful sell always satisfies $0 \leq a_t(q) \leq q \leq L_t$. This bound concerns the funded vault; it says nothing about issuer solvency or NAV correctness. With liquidity, hard reserve, current order, and cadence fixed, more effective sell pressure cannot improve the quote. The count-based cadence signal is also a griefing surface because cheap sells can alter shared state before a later order. We measure attacker pricing loss and later quote harm.



(a) Endpoint and cadence shapes.



(b) Pressure recovery without new flow.

Figure 3: Pricing shapes and recovery used by the proposed mechanism. Panel (a) shows the normalized endpoint and cadence components before their scales, caps, and maximum are applied. Panel (b) shows pressure falling from $N_t + P_t$ to N_t during the first epoch and reaching zero at $2T$.

5 Implementation

The mechanism is an Anchor program in Rust on Solana, with its underlying pricing implementation documented in the public OnRe program [23]. For reproducibility, the reported results refer to the frozen paper artifact R1, whose program, tests, and evaluation harness are available from the research artifact repository. The cited public revision and live deployment need not have identical code or configuration.

Each token and settlement-asset pair has a program-derived account for curve parameters, epoch data, sell and funded-buy values, previous pressure, and successful sell count. The protocol governor controls the NAV schedule, pair parameters, sell fee, vault target, vault withdrawals, emergency administrators, and mint-authority transfer. Emergency administrators may pause the program, while a separate upgrade authority may replace it. Anyone may refresh TVL from the current NAV and circulating supply, but a sell quote does not require a fresh cache.

A sell quote first checks that the vault can cover the raw NAV value, then follows the pricing sequence in Section 4: hard reserve, pressure, dynamic wall, utilization, haircut, and payout. Execution records the sale's pressure only after success, and a user-supplied minimum output rejects a price made unacceptable by an intervening state change.

The input charge is the larger of a rounded-up percentage fee and an absolute minimum; both are zero in our experiments. Quotes reject a zero raw payout, an empty vault, or a positive reserve target that converts to zero. Runtime utilization, powers, and haircuts use scale 10^{12}. Exponent, cadence-strength, and wall-sensitivity fields use scale 10^4, while reserve and peg fractions use basis points. Integer division rounds down unless noted otherwise. Checked arithmetic rejects overflow, and the payout factor is clamped to $[0, 1]$. The dynamic wall has a minimum of one base unit. A timestamp before the stored epoch start gives an empty preview state and resets the tracker during transition.

The buy path uses the regular offer price and refills the redemption vault before routing any remainder, so only the amount actually funded reduces recorded pressure.

For speed, the parameter sweep imports the production Rust functions. An independent 100-digit decimal Python oracle checks 34 pricing and state vectors and five conversion vectors against rebuilt Solana Bytecode Format (SBF) in LiteSVM. The harness writes event-level CSV files, rankings, revision metadata, and figures. Its endpoint-only ablation sets wall sensitivity directly to zero even though production configuration requires a positive value.

Across the fixed 100-sample rebuilt-SBF profile, the maximum costs are 60,578 compute units for a sell quote, 247,320 for a sell, 27,277 for a buy quote, and 205,828 for a buy. The pair state occupies 460 bytes and requires 4,092,480 rent-exempt lamports. With lookup tables, the tested version-0 sell and buy transactions are 424 and 430 bytes.

6 Evaluation

We use deterministic workloads to isolate how each pricing component affects splitting, reserve depletion, and later sellers. Simple policies provide comparison points, ablations remove individual components, and a parameter sweep measures how the resulting trade-offs change.

6.1 Experimental Setup

Each run begins with 1,000,000,000 settlement base units, unit NAV, and zero fees so that all measured discounts come from pricing. We use a small order of 0.1% of initial liquidity to measure the price available to a later seller; we call this the protected order, without assuming a guaranteed payout.

The reference configuration uses a 7% peg haircut, exponent 2.5, wall sensitivity 2.0, cadence threshold 20, cadence strength 1.0, and a 24-hour epoch. These values were fixed before the sweep to exercise every component, rather than calibrated to observed OnRe trading. As the examples in Section 4.4 show, they combine a small endpoint haircut at low utilization with a cadence response that reaches full strength after twenty sells. Wall sensitivity 2.0 supplies the pressure adjustment illustrated in Section 4.3. The sweep varies these five pricing parameters while keeping the 24-hour epoch as a

fixed time unit for boundary and recovery tests, not an estimate of portfolio settlement time.

The comparison includes fixed-NAV payout, static haircuts, an integrated cumulative curve, endpoint-only Prop RFQ pricing, pressure without cadence, and the full mechanism. For the integrated baseline, Equation 2 uses $p(x) = 1 - 0.07x^{2.5}$, where x is cumulative submitted value divided by initial liquidity. Its payout is floored to settlement base units. To measure splitting, we submit stress exits of 10%, 25%, 50%, or 90% of initial liquidity, either directly or in 2, 5, 10, 20, or 100 parts within one epoch or across its boundary. Separate traces test low-value cadence manipulation, ordinary bursts of sales, funded buys that relieve pressure, changes in order sequence, and two epochs of recovery. Nexus RAMM remains qualitative because its asset model, state, and pricing objectives differ from those of the mechanism evaluated here.

6.2 Metrics and Baseline Gate

We measure splitting advantage against the same exit executed in one order. If A_1 is the direct payout, A_n the payout for n parts, and Q the submitted NAV value, then

$$G_n^{NAV} = \frac{\max(A_n - A_1, 0)}{Q}, \qquad G_n^{rel} = \frac{\max(A_n - A_1, 0)}{A_1}. \tag{10}$$

G_n^{NAV} expresses the extra payout as a fraction of submitted value, while G_n^{rel} compares it with the direct payout and is defined only for $A_1 > 0$. The latter can be large when the direct exit is heavily discounted, even if the extra payout is a smaller share of Q. To capture the other costs of the pricing rule, we also report the protected-order discount $1 - a/q$ and reserve remaining L_{end}/L_0, with all ratios expressed as percentages.

For cadence-strength setting y, let G_y denote NAV-normalized split gain and G_0 the gain of the matching pressure-only configuration. We define aggregate cadence mitigation as

$$M_y = \frac{\sum G_0 - \sum G_y}{\sum G_0}. \tag{11}$$

The sum covers the 40 combinations of four exit sizes, five nontrivial partition counts, and two timing patterns. A positive value means that cadence removes some aggregate split advantage; when the pressure-only sum is zero, the metric is not applicable.

We compare configurations on four dimensions: aggregate cadence mitigation, worst NAV-normalized split gain, worst protected-order discount, and minimum reserve remaining. A configuration is non-dominated when no alternative is at least as good on every dimension and strictly better on one. For simple policies and ablations, this comparison uses only the last three metrics because cadence mitigation requires a matched pressure configuration.

We separately compare the attacker's pricing loss with the harm to a later quote, excluding fees, capital cost, and minimum-output failures. Every nonzero cadence row fails this strict price-only grief diagnostic, so a favorable pricing comparison does not imply resistance to manipulation.

Table 1: Worst results over the fixed workloads. Small-sale price is the lowest NAV percentage paid to a 0.1% exit after an earlier stress sale. Split gain is the largest partition advantage, and vault remaining is the lowest final balance. The static range covers 1%, 3%, 5%, 7%, 10%, 20%, and 30% haircuts; the separate 7% row matches the Prop RFQ peg haircut. Arrows indicate the preferred direction.

Pricing rule	Small sale (% NAV) ↑	Split gain (% NAV) ↓	Vault left (%) ↑
Fixed NAV	100.00	0.00	10.00
Static range	99.00 → 70.00	0.00	10.90 → 37.00
Static 7%	93.00	0.00	16.30
Integrated curve	94.61	0.00	11.38
Prop RFQ endpoint	99.99	5.37	10.00
Prop RFQ pressure	99.99	70.01	10.49
Full Prop RFQ	99.93	62.05	17.66

6.3 Reference Configuration

Table 1 compares the later small-sale price, split gain, and final vault balance. The three Prop RFQ rows add the endpoint curve, shared sell pressure, and cadence in sequence.

Adding cadence reduces the summed NAV-normalized split advantage by 9.67% relative to pressure-only pricing: 34 workloads improve, six are unchanged, and none worsen. This is an aggregate reduction in splitting advantage, not a 9.67 percentage-point increase in seller payouts. The minimum reserve remaining also rises from 10.49% to 17.66%, or from 104.9 to 176.6 million of the initial billion base units. These minima need not occur in the same trace as the worst small-sale price or split gain.

The individual traces show both the benefit and its limits. For a 50% exit in twenty same-epoch parts, relative split gain is 1.25% for endpoint-only, 7.51% for pressure-only, and 0.73% for full pricing. Crossing an epoch boundary raises the full mechanism's gain to 4.53%, while a 90% exit in twenty same-epoch parts still gains 196.19% relative to its heavily discounted direct payout. Later independent sellers also bear a cost: the worst honest-burst discount rises from 0.01% under pressure-only pricing to 0.36% with cadence. Endpoint-only, integrated, and static policies remain more split-resistant in these workloads, so the full design offers a reserve and pricing trade-off rather than split neutrality.

6.4 Parameter Sweep

To test how much the reference setting determines these results, the sweep exports all $10 \times 9 \times 7 \times 4 \times 6 = 15{,}120$ parameter combinations, including their feasibility, recovery parity, and comparison results. Simple-policy dominance excludes 541 configurations, leaving 14,579, including the reference; after the baseline and recovery checks, 1,651 retained rows are non-dominated. Figure 4 isolates cadence's role by showing 24 configurations with the reference haircut, exponent, and wall sensitivity but different cadence strengths and thresholds.

The strongest mitigation extreme uses a 1% peg haircut, exponent 10.0, wall sensitivity 0.5, cadence threshold 5, and cadence strength 5.0. It removes aggregate split advantage across the fixed matrix, preserves at least 30.23% of initial liquidity, and gives a

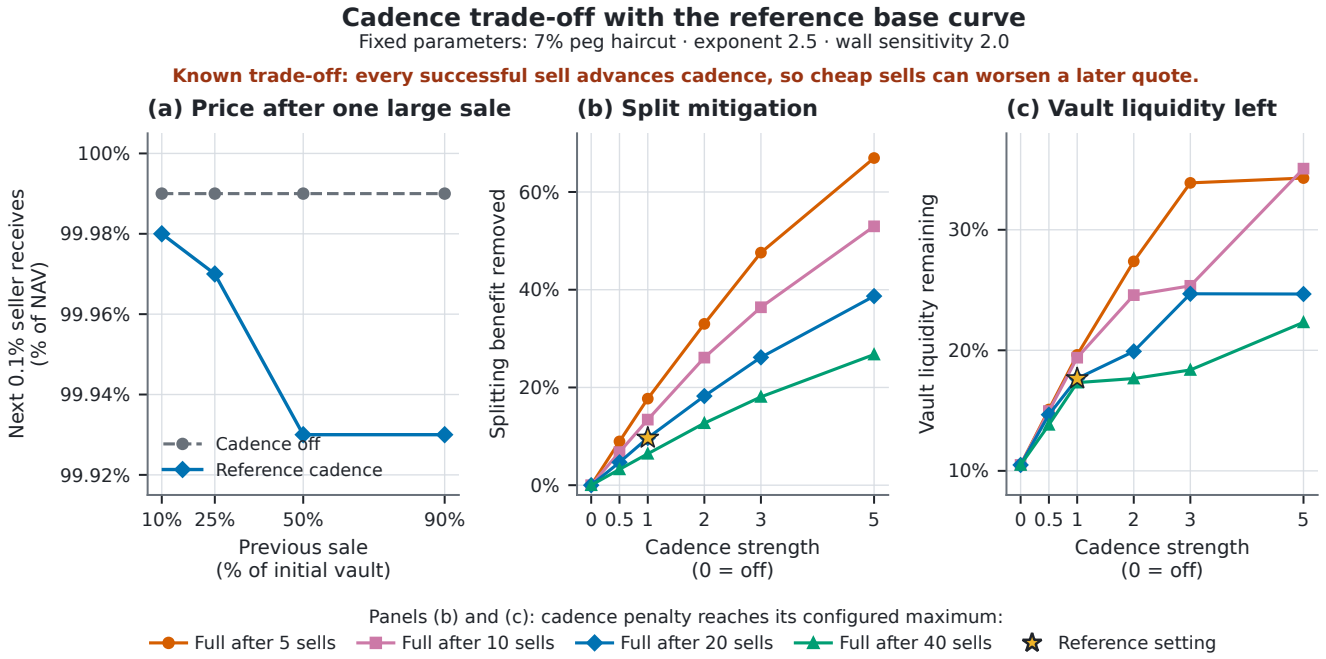


Figure 4: Cadence trade-off under the reference base curve. Panel (a) gives the next 0.1% seller's NAV percentage after prior sales of 10%, 25%, 50%, or 90% of initial liquidity. Panels (b) and (c) vary cadence strength and threshold and report aggregate split mitigation and minimum remaining liquidity. For every nonzero cadence-strength setting, the strict diagnostic finds a cheap-sell trace whose pricing loss is smaller than the later harm.

worst protected-order discount of 3.17%. Only 6 workloads improve because the other 34 are unchanged. This row is a Pareto extreme, not a selected parameter set.

6.5 Adversarial and Parity Results

The grief diagnostic asks whether very small, or dust, sells can cheaply worsen a later seller's quote. A trace passes only if it causes no victim harm or costs the attacker more than the harm imposed; only 2,484 configurations pass, all with cadence disabled. At the reference parameters, twenty dust sells cost 19 base units but reduce the protected payout by 2,653 base units, making the attacker's pricing loss only 0.71% of the victim's harm.

The funded buy-then-sell traces test a different attack: paying to reduce pressure before selling again. None is profitable; for the full mechanism, the closest nonzero trace still loses 553,032 base units. Order sequence nevertheless matters for ordinary sellers: after a 50% sale, the 0.1% seller receives 999,309 base units instead of the 999,999 received by moving first.

The first run revealed that delayed quotes restarted second-epoch decay at quote time rather than at the epoch boundary. We corrected the preview and state transition, added a regression test, and reran the protocol from implementation revision R1 described in Section 5. Across the shared-code sweep, implementation quotes now equal the explicitly rolled model at every recovery point for all 15,120 configurations. All 34 independent pricing vectors pass against rebuilt R1 SBF. The largest gap is 20 settlement base units under tolerance $\max(3, \lceil Q/50{,}000{,}000 \rceil)$. Five conversion vectors with 0, 2, 6, 8, and 9 settlement decimals match exactly. The pricing set covers elapsed times 0, $T/2$, $T-1$, T, $3T/2$, and $2T$; a backward timestamp; boundary order and liquidity values; selected parameter extremes; buy relief; and recovery. For comparison, rebuilt pre-fix SBF misses the second-epoch midpoint vector by 3,672 base units against a tolerance of five.

7 Discussion

Our reference setting retains more reserve than endpoint-only pricing and gives isolated small orders a better price than static policies with similar reserve preservation. Endpoint-only and linear policies resist splitting better, so the non-dominated configurations describe competing objectives rather than one best operating point. Without owner linking, identical traces from independent sellers and a splitting holder receive the same treatment.

Choosing Parameters in Practice. Operators must specify the funded reserve, expected ordinary sale sizes, and the maximum acceptable discount for a later small seller. For fixed utilization below one, increasing the endpoint exponent lowers the size haircut, while a lower cadence threshold makes the count response reach full strength sooner. Stronger cadence can retain more reserve but worsen later quotes, so split mitigation alone cannot guide parameter choice.

Epoch duration should account for timing and reserve replenishment: longer epochs extend the count window and pressure decay

without accelerating backing-asset settlement. The fixed 24-hour tests establish neither an optimal duration nor a safe production configuration; every tested nonzero cadence setting remains cheaply manipulable.

Cadence Griefing. Low-value sells can worsen a later quote at little protocol-level cost because cadence counts successful sells without identifying owners, as shown in Section 6.5. Ignoring small sells could restore a splitting strategy, while a minimum fee could harm legitimate small sellers. Account limits can be bypassed with several accounts, and identity gating would change the permissionless threat model [28]. We leave these variants unevaluated.

Scope and Residual Risks. The recovery defect and cadence limitation were disclosed to the program maintainers before submission.

Fixed synthetic workloads isolate behavior at one reserve scale; they do not represent production demand. They omit real arrivals, portfolio liquidation costs, changing or stale NAV, stale TVL, and external liquidity. Governance, emergency, mint, and upgrade authorities are trusted without establishing issuer solvency.

Earlier sales can change later quotes; minimum output lets sellers reject a price but cannot prevent reordering or delay. Borrowed inventory may increase attack size, while adaptive ordering, lending constraints, transaction fees, failed transactions, and capital cost remain outside the fixed traces. The results cover immediate funded exits, excluding ordering and funding of queued primary redemptions.

8 Related Work

The closest work spans AMM order splitting, flow-dependent redemption pricing, and tokenized-RWA liquidity controls. Table 2 compares their guarantees with Section 3.2.

Automated Market Makers and Order Splitting. Constant-function market makers define trades through reserves and a trading function [2]. Bichuch and Feinstein propose fees that make an AMM indifferent to transaction splitting [3]. Curve CryptoSwap instead centers its invariant on an internal price scale and changes fees with pool imbalance [7]. False-name mechanism design shows why separate accounts need not represent separate economic actors [28]. Our mechanism applies these concerns to a NAV-anchored exit with a separate reserve.

Flow-Dependent Redemption Pricing. Swing pricing uses flow-dependent NAV adjustments to allocate liquidation costs [4, 12, 15], while Prop RFQ prices transactions against an onchain reserve using order size and cadence. Nexus RAMM carries sell pressure through virtual pools and time-dependent ratchets [17]. Its economics, state, and objective differ from this RWA exit problem.

Tokenized RWA Liquidity. Documented RWA exits use periodic settlement, liquidity or solvency limits, funded capacity, or external venues [5, 8, 11, 21, 25, 27]. Prop RFQ prices immediate sales from funded liquidity and public order-flow state.

Identity-free concerns holder or taker access, excluding liquidity provision and router integration. Split resistance compares an executable Q with every executable consecutive partition from the same initial state, without buys, time passage, resets, or parameter changes. A half circle marks conditional or partial support, including configurable pool logic, batch allocation, and liquidity-dependent small-exit pricing. The proprietary-AMM survey does not establish identity-free taker access [16]. Nonzero USTB fees make strict split resistance conditional on rounding [27].

Table 2: Documented support for three identity-free exit-pricing properties.

Mechanism	Identity-free access	Split-order resistance	Favorable small exits
Hookless Uniswap v4 pool [1]	●	◐	○
Solana proprietary AMMs [16]	○	○	○
Nexus RAMM [17–19]	○	◐	○
Re Protocol reUSD [25]	○	◐	◐
Goldfinch Senior Pool [11]	○	◐	◐
Ondo OUSG [20–22]	○	●	◐
Superstate USTB [27]	○	◐	◐
Prop RFQ (this work)	●	◐	◐

○ Not established ◐ Conditional/partial ● Established

No compared external mechanism supports all three scoped properties. Redemption mechanisms provide full or partial split support through identity-linked limits, linear within-capacity pricing, or pro-rata batches; batch mechanisms do not establish the immediate-exit guarantee. Prop RFQ provides identity-free access and measured partial support for the other two properties, but guarantees neither across all traces.

Lending and Capital Efficiency. Košťál et al. [13] describe a pool-based lending protocol with fungible deposit tokens, borrowing, withdrawals, and liquidation. Smehyl and Homoliak [26] propose deploying idle lending liquidity into external yield strategies, with recall and loss-allocation concerns. Findra et al. [10] compare borrowing capacity, liquidation behavior, rates, and call-path gas under matched Ethereum fork tests. These studies concern capital allocation, borrowing, and collateral liquidation, whereas Prop RFQ prices immediate RWA token sales from an already funded reserve.

9 Conclusion

Prop RFQ prices immediate RWA exits using NAV, funded liquidity, and recent selling without linking accounts to owners. Cadence reduces aggregate split gain and retains more reserve in the fixed workloads, but can worsen later prices and allows cheap count manipulation. Rebuilt-SBF pricing and conversion checks pass under the stated tolerances, although extreme splits retain large gains. The model-specific trilemma explains why shared history cannot distinguish split sellers from independent holders.

Acknowledgments

The authors declare no competing interests. The authors used generative AI (OpenAI ChatGPT-5 and Google Gemini 2.5 Pro) to assist with language polishing, rephrasing, and LaTeX formatting. The authors take full responsibility for the manuscript. This work was supported by the Science Grant Agency VEGA under project 1/0771/26, the Slovak Research and Development Agency under the Contract no. APVV-25-0804, and by the BUT internal project FIT-S-23-8151.

References

[1] Hayden Adams, Moody Salem, Noah Zinsmeister, Sara Reynolds, Austin Adams, Will Pote, Mark Toda, Alice Henshaw, Emily Williams, and Dan Robinson. 2024. *Uniswap v4 Core*. White Paper. Uniswap Labs. Retrieved 2026-07-29 from https://app.uniswap.org/whitepaper-v4.pdf

[2] Guillermo Angeris and Tarun Chitra. 2020. Improved Price Oracles: Constant Function Market Makers. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*. 80–91. doi:10.1145/3419614.3423251

[3] Maxim Bichuch and Zachary Feinstein. 2022. Axioms for Automated Market Makers: A Mathematical Framework in FinTech and Decentralized Finance. (Oct. 2022). arXiv:2210.01227 [q-fin.MF] doi:10.48550/arXiv.2210.01227

[4] Agostino Capponi, Paul Glasserman, and Marko Weber. 2023. Swing Pricing: Theory and Evidence. *Annual Review of Financial Economics* 15, 1 (Nov. 2023), 617–640. doi:10.1146/annurev-financial-110921-100843

[5] Centrifuge. 2026. Centrifuge V2. Retrieved 2026-07-22 from https://docs.centrifuge.io/getting-started/legacy/centrifuge-v2/

[6] Committee on Payments and Market Infrastructures. 2024. *Tokenisation in the Context of Money and Other Assets: Concepts and Implications for Central Banks*. CPMI Papers 225. Bank for International Settlements. https://www.bis.org/cpmi/publ/d225.htm

[7] Michael Egorov. 2021. *Automatic Market-Making with Dynamic Peg*. White Paper. Curve Finance. Retrieved 2026-07-29 from https://docs.curve.fi/assets/pdf/whitepaper_cryptoswap.pdf

[8] Ensuro. 2022. Liquidity Pools. Retrieved 2026-07-22 from https://docs.ensuro.co/ensuro-docs/smart-contracts/liquidity-pools

[9] Financial Stability Board. 2024. *The Financial Stability Implications of Tokenisation*. Technical Report. Financial Stability Board. https://www.fsb.org/2024/10/the-financial-stability-implications-of-tokenisation/

[10] Jan Findra, Zdeněk Lapeš, and Ivan Homoliak. 2026. A Controlled Case Study of Design Trade-Offs in DeFi Lending Protocols. In *Proceedings of the 7th International Conference on Mathematical Research for Blockchain Economy (MARBLE 2026) (Lecture Notes in Operations Research)*. Springer, Limassol, Cyprus. https://eprint.iacr.org/2026/1728

[11] Goldfinch. 2022. Liquidity & LP Withdrawals. Retrieved 2026-07-22 from https://docs.goldfinch.finance/goldfinch/goldfinch-v1/protocol-mechanics/liquidity

[12] Dunhong Jin, Marcin Kacperczyk, Bige Kahraman, and Felix Suntheim. 2022. Swing Pricing and Fragility in Open-End Mutual Funds. *The Review of Financial Studies* 35, 1 (Jan. 2022), 1–50. doi:10.1093/rfs/hhab022

[13] Kristián Košťál, Muhammad Nasim Bahar, Richard Gazdík, and Michal Ries. 2023. Advancing Decentralized Finance: A Comprehensive Pool-Based Liquidity Protocol. In *2023 Fifth International Conference on Blockchain Computing and Applications (BCCA)*. IEEE, Kuwait, Kuwait, 158–165. doi:10.1109/BCCA58897.2023.10338926

[14] Kristián Košťál, Igor Ďurica, and Michal Ries. 2023. Omniscient: The Universal Blockchain Oracle. In *2023 IEEE International Conference on Blockchain (Blockchain)*. IEEE, Ocean Flower Island, Hainan, China, 113–120. doi:10.1109/Blockchain60715.2023.00028

[15] Ulf Lewrick and Jochen Schanz. 2017. *Liquidity Risk in Markets with Trading Frictions: What Can Swing Pricing Achieve?* BIS Working Papers 663. Bank for International Settlements, Basel. Retrieved 2026-07-29 from https://www.bis.org/publ/work663.pdf

[16] Lostin. 2025. Solana's Proprietary AMM Revolution. Retrieved 2026-07-29 from https://www.helius.dev/blog/solanas-proprietary-amm-revolution

[17] Reinis Melbardis, Anatol Prisacaru, and Miljan Milidrag. 2023. *The Ratcheting AMM (RAMM) Whitepaper*. White Paper Version 1.0. Nexus Mutual. Retrieved 2026-07-22 from https://docs.nexusmutual.io/assets/files/nexus-mutual-ramm-whitepaper-v1-2023-80163504ce80b4c5342f4ee0a073656e.pdf

[18] Nexus Mutual. 2026. Token Model. Retrieved 2026-08-09 from https://docs.nexusmutual.io/protocol/nxm-token/token-model/

[19] Nexus Mutual. n.d.. Membership. Retrieved 2026-07-29 from https://docs.nexusmutual.io/overview/membership/

[20] Ondo Finance. 2026. Eligibility & Onboarding. Retrieved 2026-08-09 from https://docs.ondo.finance/qualified-access-products/ousg/eligibility-and-onboarding

[21] Ondo Finance. 2026. Instant Minting and Redemption Limits. Retrieved 2026-07-22 from https://docs.ondo.finance/qualified-access-products/ousg/instant-limits

[22] Ondo Finance. 2026. OUSG Overview. Retrieved 2026-08-09 from https://docs.ondo.finance/qualified-access-products/ousg/overview

[23] OnRe. 2026. OnRe Solana Program. Software repository. Retrieved 2026-07-16 from https://github.com/onre-finance/onre-sol/commit/fc11256d9f411494f849931dcd7ae801a50ca29c

[24] OnRe. 2026. OnRe Tokenized Reinsurance (ONyc). OnRe documentation. Retrieved 2026-07-16 from https://docs.onre.finance/introduction/onre-tokenized-reinsurance-onyc

[25] Re Protocol. 2026. Redemptions. Retrieved 2026-07-22 from https://docs.re.xyz/minting-and-redemptions/redemptions

[26] Adam Smehyl and Ivan Homoliak. 2026. Enhancing Capital Efficiency in DeFi Lending and Liquidity Provision. In *Proceedings of the 2026 IEEE Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE, Florence, Italy. https://eprint.iacr.org/2026/1727

[27] Superstate. 2026. Smart Contracts. Retrieved 2026-07-22 from https://docs.superstate.com/investors/smart-contracts

[28] Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. 2004. The Effect of False-Name Bids in Combinatorial Auctions: New Fraud in Internet Auctions. *Games and Economic Behavior* 46, 1 (2004), 174–188. doi:10.1016/S0899-8256(03)00045-9