Cryptoeconomics of Revnets

The CEL Team

November 12, 2025

Abstract

Contents

1 Introduction

1.1 What is a Revnet?

Revnets are autonomous, tokenized financial structures that route revenue inflows and redemptions through immutable smart-contract rules. They are deployed using the Juicebox V5 protocol and governed entirely by code rather than DAO-style votes. A revnet is a digital vending machine for revenue: money goes in, tokens come out; tokens can later be redeemed for a programmatic share of the treasury. All core rules are *fixed at deployment* and define deterministic issuance, redemption, and distribution, with optional AMM integration and operation across multiple EVM-compatible chains. Revnets are *governance-free*, *deterministic*, *transparent/auditable*, and *management-scoped*, aligning contributors, builders, and customers around a tamper-resistant financial logic.

1.2 How does a Revnet work (at a glance)

1. **Pay in (issuance).** Anyone can pay the revnet in the accepted base asset(s) \$RES. The contract mints \$TOK at the current issuance price; an optional *split* routes a fixed percent of new tokens to preset recipients. Funds remain in the treasury.

2. **Cash out (redemption).** A holder burns \$TOK to reclaim treasury funds. A stage-defined *cash-out tax* keeps part of the redeemable value in the treasury, which raises the future floor for remaining holders.

3. **Borrow (loans).** Instead of cashing out, a holder can borrow from the treasury against their \$TOK. The borrowable amount is capped by that collateral's cash-out value.

1.3 The Economic Levers

Revnets operate in *stages*. A *stage* k is a time window during which the revnet follows a fixed configuration of parameters. Each stage hard-codes exactly seven parameters which are the only adjustable levers of a revnet:

$$\mathcal{S}_k = \big(t_k,\, P_{\text{issue},k,0},\, \gamma_{\text{cut},k},\, \Delta t_k,\, \sigma_k,\, r_k,\, \mathcal{A}_k\big). \tag{1}$$

Where:

1. **Stage start time** (t_k): When this stage takes effect.
2. **Initial issuance rate** ($P_{\text{issue},k,0}$): \$TOK per \$RES unit paid at the start of the stage (sets the initial price).
3. **Issuance cut percent** ($\gamma_{\text{cut},k}$): The fractional reduction in issuance each period (equivalently, the per-period price increase factor $1/(1-\gamma_{\text{cut},k})$).
4. **Issuance cut frequency** (Δt_k): How often the issuance cut applies (e.g., daily, monthly).
5. **Split percentage** (σ_k): Fraction of every new mint routed to predefined recipients (the rest goes to the payer).
6. **Cash-out tax rate** (r_k): Share of cash-out value retained in treasury (drives the floor and its growth).
7. **Auto-issuances** ($\mathcal{A}_k$): One-time, pre-authorized mints at stage start (chain, amount, beneficiary).

Once set, the tuple $\mathcal{S}_k$ is immutable. Stage transitions occur automatically by timestamp, so the current stage is uniquely determined by t. Within a stage, issuance, redemptions, and distributions follow the deterministic rules implied by these seven parameters.
Intuitively:

- The issuance levers (2–4) define a time-rising *issuance price*;
- The cash-out/tax lever (6) defines an activity-driven *price floor*;
- The split/auto-issuance levers (5,7) deterministically allocate new supply.

Section 2 formalizes these dynamics.

2 Mathematical Model of Revnet Economics

In the following section we model a Revnet mathematically, formalizing:

- Its state variables and their evolution in time;
- Its mechanisms such as issuance, cash-out and loan.

2.1 Parameters and State Variables

The economic behavior of a Revnet is determined jointly by:

1. The immutable *stage parameters* $\mathcal{S}_k$ (cf. Sec. 1.3), fixed at deployment for each stage k;
2. The evolving *state variables*, which track treasury balances, token supply, and loan positions over time.

Stage parameters. For reference, the parameter tuple for stage k is

$$\mathcal{S}_k = \big(t_k,\, P_{\text{issue},k,0},\, \gamma_{\text{cut},k},\, \Delta t_k,\, \sigma_k,\, r_k,\, \mathcal{A}_k\big).$$

State variables. The core dynamic variables are listed in Table 1.

Variable	Description
$B(t)$	Treasury balance at time t
$S(t)$	Outstanding token supply at time t
$S_{\text{collateral}}(t)$	Tokens burned as loan collateral at time t
$B_{\text{borrowed}}(t)$	Amount currently borrowed against collateral at time t
$P_{\text{issue},k}(t)$	Issuance price function under stage k at time t

Table 1: Core state variables of a Revnet protocol

At any time t, the state of the protocol is fully determined by the pair

$$\big(\mathcal{S}_k,\ \{B(t), S(t), S_{\text{collateral}}(t), B_{\text{borrowed}}(t)\}\big),$$

where $\mathcal{S}_k$ is the active stage (selected deterministically by t) and the second component evolves endogenously as users interact with the Revnet. The next subsections formalize how each mechanism (issuance, redemption, loans) updates these variables.

2.2 Pay in – Issuance

At any time t within stage k (defined by the parameter tuple $\mathcal{S}_k$), participants may pay the Revnet contract in the accepted base asset $RES. In return, the contract mints new $TOK at the issuance price $P_{\text{issue},k}(t)$. A fraction σ_k of the minted tokens is routed to pre-defined split recipients, while the remainder accrues to the payer. The payment remains in the treasury, increasing the backing of all outstanding tokens.

Issuance price (discrete cuts). Within stage k starting at time t_k, the issuance price evolves by discrete multiplicative jumps every Δt_k seconds:

$$P_{\text{issue},k}(t) = P_{\text{issue},k,0} \cdot \gamma_k^{\left\lfloor \frac{t - t_k}{\Delta t_k} \right\rfloor}, \qquad t \in [\, t_k,\, t_{k+1}\,), \tag{2}$$

where

$$\begin{aligned} \gamma_k &= \frac{1}{1 - \gamma_{\text{cut},k}} \quad \text{(per-interval price growth factor)}, \\ P_{\text{issue},k,0} &= \text{initial issuance price at } t = t_k, \\ \Delta t_k &= \text{issuance cut frequency for stage } k. \end{aligned}$$

Here $\lfloor x \rfloor$ denotes the *floor function* (greatest integer $\leq x$), which makes $P_{\text{issue},k}(t)$ a step function: the price is constant within each interval and jumps by a factor γ_k precisely at times $t = t_k + m\,\Delta t_k$ that lie in $[t_k, t_{k+1})$.
The variable t ranges over the stage window $[t_k,\, t_{k+1})$, where t_{k+1} is the (start) time of the next stage. The *length* of stage k is therefore

$$\text{duration}_k \;=\; t_{k+1} - t_k,$$

with the convention that the final stage has $t_{k+1} = +\infty$ (infinite duration). The number of scheduled issuance jumps within stage k is

$$N_k \;=\; \left\lfloor \frac{t_{k+1} - t_k}{\Delta t_k} \right\rfloor,$$

and the jumps occur at $t = t_k + m\,\Delta t_k$ for all integers m such that $t_k + m\,\Delta t_k \in [t_k, t_{k+1})$.

Minted quantity. For a payment amount x of $RES tokens at time t in stage k, the contract issues q_{issued} $TOK according to:

$$q_{\text{issued}}(t) = \frac{x}{P_{\text{issue},k}(t)}. \tag{3}$$

Token allocation. The minted tokens are split according to the split fraction $\sigma_k \in [0, 1]$:

$$\begin{aligned} q_{\text{payer}}(t) &= (1 - \sigma_k)\, q_{\text{issued}}(t), && \text{(Tokens allocated to payer)} \\ q_{\text{split}}(t) &= \sigma_k\, q_{\text{issued}}(t). && \text{(Tokens allocated to splits)} \end{aligned}$$

The effective *user issuance price* is therefore

$$P_{\text{issue}}^{\text{user}}(t) = \frac{P_{\text{issue},k}(t)}{1 - \sigma_k}. \tag{4}$$

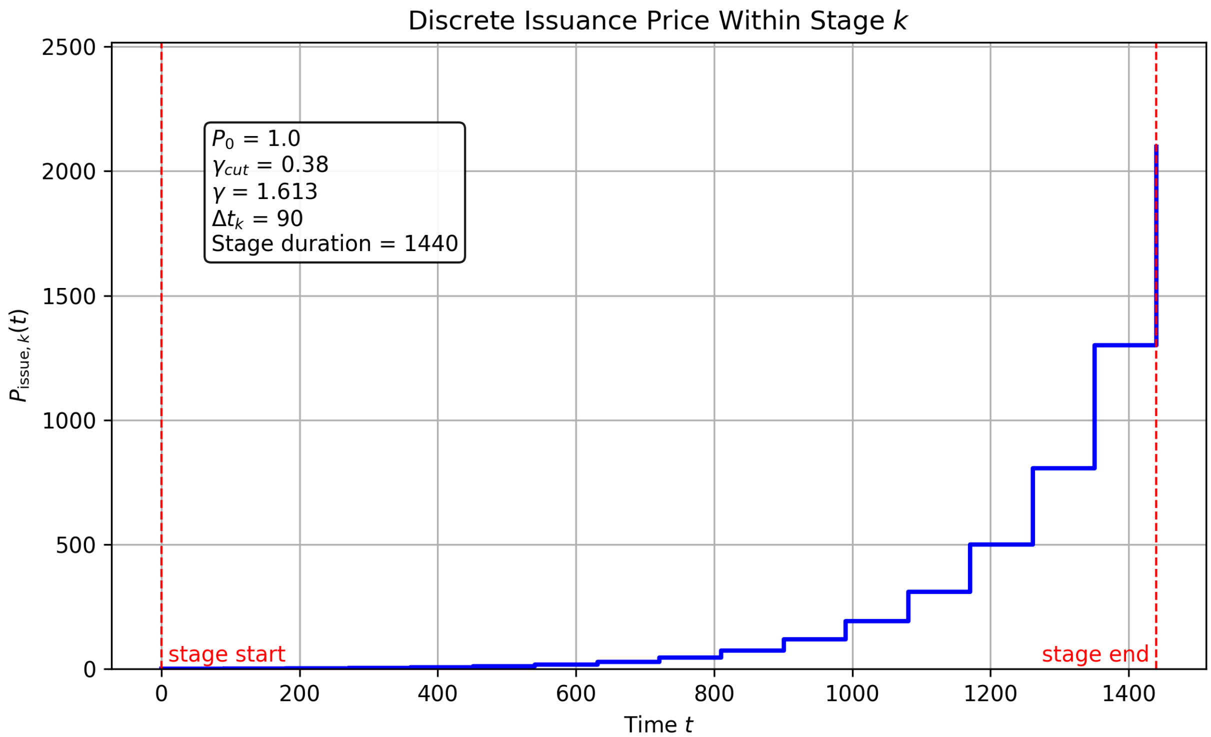

Figure 1: Discrete evolution of the issuance price $P_{\text{issue},k}(t)$ within a single stage k. The price increases in stepwise fashion every Δt_k according to $P_{\text{issue},k}(t) = P_{\text{issue},k,0}\,\gamma_k^{\lfloor (t-t_k)/\Delta t_k \rfloor}$, with parameters P_0, γ_{cut}, $\gamma = 1/(1-\gamma_{\text{cut}})$, and stage duration $(t_{k+1} - t_k) = \text{duration}$. Red dashed lines mark the stage boundaries at t_k and t_{k+1}.

State updates. Let $B(t)$ denote the treasury balance, $S(t)$ the circulating supply, and $U_i^{\text{ASSET}}(t)$ the balance of user i in a given asset. At the instant of an issuance event, the updates are:

$$B(t^+) = B(t^-) + x, \qquad \text{(Treasury balance)}$$
$$S(t^+) = S(t^-) + q_{\text{issued}}(t), \qquad \text{(Total token supply)}$$
$$U_i^{\text{RES}}(t^+) = U_i^{\text{RES}}(t^-) - x, \qquad \text{(Payer's base asset balance)}$$
$$U_i^{\text{TOK}}(t^+) = U_i^{\text{TOK}}(t^-) + q_{\text{payer}}(t), \qquad \text{(Payer's token balance)}$$

These update rules define a discrete-time dynamical system for $(B, S, \{U_i\})$.

2.3 Cash-out – Redemption

At any time t within stage k, a \$TOK holder may burn tokens to reclaim a share of the treasury in the base asset \$RES. Cash-out (also referred to as "redemption") is governed by a convex bonding curve, which ensures that partial cash-outs retain value in the treasury and gradually increase the floor price for remaining holders.

Redemption curve. Suppose a holder redeems q tokens at time t with circulating supply $S(t)$ and treasury $B(t)$. The reclaimable value (before fees) is

$$C_k(q; S, B) = \frac{q}{S} B \left[(1 - r_k) + r_k \frac{q}{S} \right], \tag{5}$$

where r_k is the cash-out tax for stage k.
For any $r_k > 0$, the curve is strictly convex:

$$\frac{d^2 C_k}{dq^2} = \frac{2 B r_k}{S^2} > 0,$$

so that:

- small redemptions receive worse effective prices (greater tax impact),
- large redemptions receive better effective prices (lower relative tax impact).

Cash-out fees: REV and NANA fees are applied in two stages:
Stage 1 - REV fee: Before applying the redemption curve, the REV fee is deducted from the token amount:

$$q_{nf} = (1 - \phi_{\text{REV}}) q = 0.975 q \quad \text{, where } \phi_{\text{REV}} = 0.025 \tag{6}$$

Stage 2 - NANA fee: After calculating the redemption value, the NANA fee is deducted:

$$C_{\text{gross}} = C_k(q_{nf}; S, B) \tag{7}$$

$$C_{\text{user}} = (1 - \phi_{\text{NANA}}) C_{\text{gross}} = 0.975 C_{\text{gross}} \quad \text{where } \phi_{\text{NANA}} = 0.025 \tag{8}$$

Both fees are redistributed as payments to their respective revnets, meaning that:
REV fee → Paid to the \$REV revnet → Issues \$REV tokens to the person cashing out;
NANA fee → Paid to the \$NANA revnet → Issues \$NANA tokens to the person cashing out.

User redemption price The effective price experienced by a user cashing out q \$TOK tokens:

$$P^{\text{user}}_{\text{cash-out}}(q) = (0.975)^2 \cdot \frac{B}{S} \left[(1 - r_k) + r_k \frac{0.975 q}{S} \right] \tag{9}$$

Fee recycling After the user redeems C_{user}, the updated supply and treasury are

$$S' = S - q_{\text{nf}},$$
$$B' = B - C_{\text{gross}}.$$

On this updated state (S', B'), the network fees are treated as separate redemptions :

$$C^{\text{REV}}_{\text{fee}} = C_k\big(q^{\text{REV}}_{\text{fee}}; S', B'\big),$$
$$C^{\text{NANA}}_{\text{fee}} = \phi_{\text{NANA}} C_{\text{gross}}.$$

These are forwarded as inbound payments into the REV and NANA Revnets, so the redeemer also receives X^{REV} \$REV tokens from $C_{\mathrm{fee}}^{\mathrm{REV}}$, and X^{NANA} \$NANA tokens from $C_{\mathrm{fee}}^{\mathrm{NANA}}$ [1].

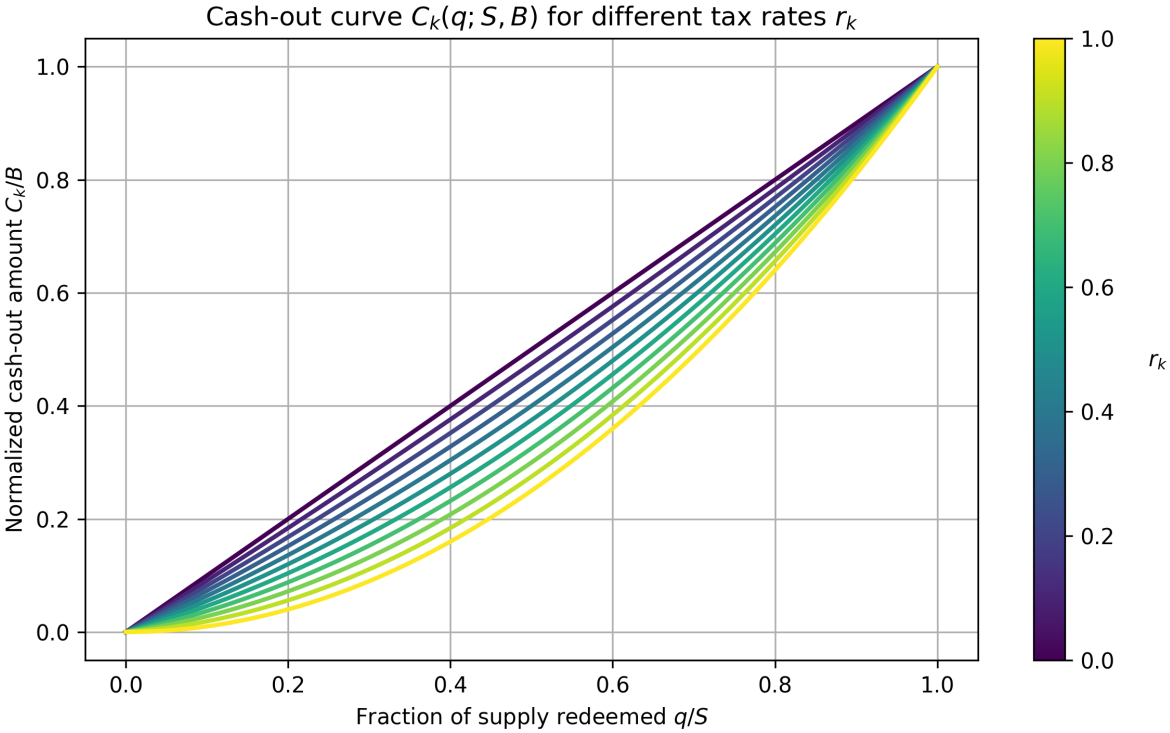

Figure 2: Cash-out function $C_k(q; S, B)$ for different cash-out tax rates $r_k \in [0, 1]$. The curve shows the normalized redemption amount C_k/B as a function of the redeemed fraction q/S for different cash-out tax values r_k. As r_k increases, the curve becomes more convex, indicating that higher taxes retain a larger share of the treasury value.

State updates. Let $U_i^{\mathrm{ASSET}}(t)$ the balance of user i in a given asset, after redeeming q tokens:

$$\begin{aligned}
U_i^{\mathrm{TOK}}(t^+) &= U_i^{\mathrm{TOK}}(t^-) - q, && \text{(User's \$TOK balance)}\\
U_i^{\mathrm{RES}}(t^+) &= U_i^{\mathrm{RES}}(t^-) + C_{\mathrm{user}}, && \text{(User's base asset balance)}\\
U_i^{\mathrm{REV}}(t^+) &= U_i^{\mathrm{REV}}(t^-) + X^{\mathrm{REV}}, && \text{(User's \$REV balance)}\\
U_i^{\mathrm{NANA}}(t^+) &= U_i^{\mathrm{NANA}}(t^-) + X^{\mathrm{NANA}}, && \text{(User's \$NANA balance)}
\end{aligned}$$

[1] Which are computed as:

$$\begin{aligned}
X^{\mathrm{REV}} &= (1 - \sigma_k^{\mathrm{REV}}) \frac{C_{\mathrm{fee}}^{\mathrm{REV}}}{P_{\mathrm{issue},k}^{\mathrm{REV}}(t)} && \text{(in \$REV tokens)}\\
X^{\mathrm{NANA}} &= (1 - \sigma_k^{\mathrm{NANA}}) \frac{C_{\mathrm{fee}}^{\mathrm{NANA}}}{P_{\mathrm{issue},k}^{\mathrm{NANA}}(t)}. && \text{(in \$NANA tokens)}
\end{aligned}$$

While the aggregate system variables update as:

$$S(t^+) = S(t^-) - q, \qquad \text{(Total supply)}$$
$$B(t^+) = B(t^-) - C_{\text{user}} - C_{\text{fee}}^{\text{REV}} - C_{\text{fee}}^{\text{NANA}}. \qquad \text{(Treasury balance)}$$

2.4 Borrow – Loan

Instead of cashing out, a holder can borrow \$RES from the treasury against their \$TOK as collateral. The collateral's cash-out value caps the amount that can be borrowed. Collateral tokens are *burned* at origination (not locked) and are *reminted* pro rata as the loan is repaid. The system maintains overcollateralization by tying borrowable amounts to cash-out values, ensuring the revnet remains solvent even if all loans default.

2.4.1 Taking the loan

Borrowing capacity. The maximum borrowable amount with collateral q_c is determined by the cash-out function:

$$L_{\text{gross}}(q_c) = C_k\big(q_c; S_{\text{eff}}, B_{\text{eff}}\big), \tag{10}$$

with effective values that include outstanding loans:

$$S_{\text{eff}} = S(t) + S_{\text{collateral}}(t), \tag{11}$$
$$B_{\text{eff}} = B(t) + B_{\text{borrowed}}(t). \tag{12}$$

Here $S(t)$ and $B(t)$ represent the current circulating supply and treasury balance (after the loan has been issued).[2] These formulas treat burned collateral as if it were still in supply, and treat outstanding loans as if they remained in treasury. This pricing mechanism ensures:

- No sequencing advantages: later borrowers get almost the same terms as earlier ones against the next cash-out value.[3]
- The system maintains solvency: borrowers cannot extract more value than a direct cash-out would provide (See Sec. 4 for the detailed analysis on solvency).

Borrowing fees: When borrowing, fees are deducted from L_{gross} and are calculated as:

$$F = \frac{f \cdot L_{\text{gross}}}{1000 + f}, \qquad f \in [10, 500]. \tag{13}$$

In particular, three fees are applied:

1. **Protocol fee** (F_{NANA}): $f_{\text{NANA}} = 25$ charged by the underlying NANA Juicebox protocol.
2. **REV fee** (F_{REV}): $f_{\text{REV}} = 10$ charged by the REV revnet ecosystem.
3. **Prepay fee** (F_{prepaid}): Variable prepaid fee chosen by the borrower, ranging from $f_{\text{prepaid}} \in [25, 500]$. This fee purchases an interest-free repayment window.

[2] Equivalently, if S_0, B_0 are the pre-loan state:

$$S(t) = S_0 - S_{\text{collateral}}(t), \quad B(t) = B_0 - B_{\text{borrowed}}(t).$$

[3] Actually, later borrowers get a slight edge. The Prepay Fee is indeed used to buy tokens via the issuance mechanism. This slightly increaes the B/S ratio (See Sec. 3.2), thus increasing the accessible capital for the same borrowed amount.

Thus, the borrower receives:

$$L_{\text{net}} = L_{\text{gross}}(1 - F_{\text{NANA}} - F_{\text{REV}} - F_{\text{prepaid}}). \tag{14}$$

Fees are forwarded as inbound payments into the respective Revnets, so the borrower also receives X^{NANA} \$NANA tokens from F_{NANA}, X^{REV} \$REV tokens from F_{REV}, and X^{TOK} \$TOK tokens from F_{prepay}[4].

Prepaid duration. The source fee purchases a "fee-free" repayment period proportional to the amount prepaid:

$$T_{\text{prepaid}} = \frac{f_{\text{prepaid}}}{500} \cdot T_{\text{liquidation}} \tag{15}$$

Where $T_{\text{liquidation}} = 3650\text{days}$ (10 years).
During T_{prepaid}, the loan can be repaid without additional fees. For example, if $f_{\text{prepaid}} = 25$ ($\sim$ 2.5%) then the "fee-free" repayment period will be of 182.5 days (∼6 months).

State updates. Let $U_i^{\text{ASSET}}(t)$ the balance of user i in a given asset, after taking a loan with q_c tokens:

$$\begin{aligned}
U_i^{\text{TOK}}(t^+) &= U_i^{\text{TOK}}(t^-) - q_c + X^{\text{TOK}}, && \text{(Borrower's \$TOK balance)} \\
U_i^{\text{RES}}(t^+) &= U_i^{\text{RES}}(t^-) + L_{\text{net}}, && \text{(Borrower's base asset balance)} \\
U_i^{\text{REV}}(t^+) &= U_i^{\text{REV}}(t^-) + X^{\text{REV}}, && \text{(Borrower's \$REV balance)} \\
U_i^{\text{NANA}}(t^+) &= U_i^{\text{NANA}}(t^-) + X^{\text{NANA}}, && \text{(Borrower's \$NANA balance)}
\end{aligned}$$

While the aggregate system variables update as:

$$\begin{aligned}
S(t^+) &= S(t^-) - q_c, && \text{(Circulating supply (collateral burned))} \\
S_{\text{collateral}}(t^+) &= S_{\text{collateral}}(t^-) + q_c, && \text{(Tracked collateral)} \\
B(t^+) &= B(t^-) - (1 - F_{prepaid})L_{\text{gross}}, && \text{(Treasury balance)} \\
B_{\text{borrowed}}(t^+) &= B_{\text{borrowed}}(t^-) + (1 - F_{prepaid})L_{\text{gross}}. && \text{(Loan obligation)}
\end{aligned}$$

[4]Where:

$$\begin{aligned}
X^{\text{NANA}} &= (1 - \sigma_k^{\text{NANA}}) \frac{F_{\text{NANA}} L^{\text{gross}}}{P_{\text{issue},k}^{\text{NANA}}(t)}, \\
X^{\text{REV}} &= (1 - \sigma_k^{\text{REV}}) \frac{F_{\text{REV}} L^{\text{gross}}}{P_{\text{issue},k}^{\text{REV}}(t)}, \\
X^{\text{TOK}} &= (1 - \sigma_k) \frac{F_{\text{source}} L^{\text{gross}}}{P_{\text{issue},k}(t)}.
\end{aligned}$$

2.4.2 Repaying the loan

When repaying a loan, the borrower specifies how much collateral to retrieve (q_{return}) and a maximum repayment amount ($R_{\max}$). The system calculates the required payment based on the remaining loan value and elapsed time as follows.
Let the borrower have:

- Deposited q_c tokens as collateral at time t_{created}
- Borrowed gross amount L_{gross}
- Paid prepaid fee with parameter f_{prepaid}
- Current time t with elapsed time $\Delta t = t - t_{\text{created}}$

At repayment, the borrower wishes to retrieve q_{return} collateral, leaving:

$$C_{\text{new}} = q_c - q_{\text{return}}. \tag{16}$$

The system calculates the new borrowable amount based on remaining collateral:

$$L_{\text{new}} = C_k\big(C_{\text{new}};\, S_{\text{eff}}, B_{\text{eff}}\big). \tag{17}$$

The principal that must be repaid to support this collateral reduction is:

$$P_{\text{repay}} = L_{\text{gross}} - L_{\text{new}}. \tag{18}$$

This ensures the loan remains overcollateralized after the partial repayment.

Total repayment. The total amount due is:

$$R(t) = P_{\text{repay}} + F_{\text{time}}(t), \tag{19}$$

where $F_{\text{time}}(t)$ is a time-dependent fee.

The repayment amount is split into two operations:

1. **Principal** (P_{repay}): Returned to the revnet, restoring treasury balance without minting tokens.
2. **Source fee** (F_{time}): Paid to the revnet as a standard payment, which mints $TOK to the beneficiary and increases the revnet's balance.

Time-dependent fee. The additional time-dependent fee depends on when the repayment occurs:

Case 1: Within prepaid period ($\Delta t \leq T_{\text{prepaid}}$)

$$F_{\text{time}}(t) = 0. \tag{20}$$

Case 2: After prepaid period but before liquidation ($T_{\text{prepaid}} < \Delta t \leq T_{\text{liquidation}}$)
First, calculate the originally prepaid amount:

$$F_{\text{prepaid}} = \frac{f_{\text{prepaid}} \cdot L_{\text{gross}}}{1000 + f_{\text{prepaid}}}. \tag{21}$$

The unpaid portion of the loan is:

$$L_{\text{unpaid}} = L_{\text{gross}} - F_{\text{prepaid}}. \tag{22}$$

A time-based fee rate increases linearly from 0 to 1000 over the remaining loan period:

$$\phi(t) = \frac{t - t_{\text{created}} - T_{\text{prepaid}}}{T_{\text{liquidation}} - T_{\text{prepaid}}} \cdot 1000. \tag{23}$$

The full source fee (if entire loan were repaid at time t) is:

$$F_{\text{full}}(t) = \frac{\phi(t) \cdot L_{\text{unpaid}}}{1000 + \phi(t)}. \tag{24}$$

For partial repayment, the source fee is proportional to the principal being repaid:

$$F_{\text{time}}(t) = \frac{P_{\text{repay}}}{L_{\text{gross}}} \cdot F_{\text{full}}(t). \tag{25}$$

Case 3: After liquidation period ($\Delta t > T_{\text{liquidation}}$)
The loan can no longer be repaid; it must be liquidated (see below).

State updates. Let $U_i^{\text{ASSET}}(t)$ the balance of user i in a given asset, after repaying a loan with q_c tokens:

$$\begin{aligned} U_i^{\text{RES}}(t^+) &= U_i^{\text{RES}}(t^-) - R(t), && \text{(Borrower's base asset balance)} \\ U_i^{\text{TOK}}(t^+) &= U_i^{\text{TOK}}(t^-) + q_{\text{return}}, && \text{(Borrower's \$TOK balance)} \end{aligned}$$

While the aggregate system variables update as:

$$\begin{aligned} B(t^+) &= B(t^-) + R(t), && \text{(Treasury balance)} \\ S(t^+) &= S(t^-) + q_{\text{return}}, && \text{(Circulating supply)} \\ S_{\text{collateral}}(t^+) &= S_{\text{collateral}}(t^-) - q_{\text{return}}, && \text{(Tracked collateral)} \\ B_{\text{borrowed}}(t^+) &= B_{\text{borrowed}}(t^-) - P_{\text{repay}}. && \text{(Loan obligation)} \end{aligned}$$

The returned collateral is reminted to the borrower, and the loan obligation is reduced by the principal repaid.

2.4.3 Liquidation

If a loan remains unpaid beyond the liquidation period, it becomes liquidatable by anyone. A loan can be liquidated if:

$$\Delta t = t - t_{\text{created}} > T_{\text{liquidation}} = 3650 \text{ days}. \tag{26}$$

For each liquidated loan:

1. The loan NFT is burned
2. Accounting is updated:

$$S_{\text{collateral}}(t^+) = S_{\text{collateral}}(t^-) - q_c, \tag{27}$$
$$B_{\text{borrowed}}(t^+) = B_{\text{borrowed}}(t^-) - L_{\text{gross}}. \tag{28}$$

3. The actual token supply $S(t)$ and treasury balance $B(t)$ remain unchanged

3 Price Dynamics and Arbitrage Mechanisms

The user issuance (Eq. 4) and cash-out prices (Eq. 9) define the ceiling and floor prices of \$TOK, respectively. To show this, let us assume that at time t^* an AMM emerges with price P^{AMM}.

3.1 Definition of the Price Corridor

Price Ceiling

If $P^{\text{AMM}} > P^{\text{user}}_{\text{issue}}$, an arbitrageur would:

1. Buy q \$TOK in the revnet at current issuance price in exchange for x^{in} \$RES, i.e. $q = \frac{x^{in}}{P^{\text{user}}_{\text{issue}}}$;
2. Sell all q tokens for x^{AMM} base tokens at current AMM price, i.e. $x^{out} = P^{\text{AMM}} q$
3. Since $P^{\text{AMM}} > P^{\text{user}}_{\text{issue}}$, then $x^{out} > x^{in}$.

Thus, if $P^{\text{AMM}} > P^{\text{user}}_{\text{issue}}$ an arbitrageur buys \$TOK through the Revnet, selling them on the AMM for \$RES.

This demonstrates that $P^{\text{user}}_{\text{issue}}$ defines the *price ceiling* of \$TOK:

$$\boxed{P^{\text{ceil}}(t) = P^{\text{user}}_{\text{issue}}(t) = \frac{P_{\text{issue},k}(t)}{1 - \sigma_k}} \tag{29}$$

Price Floor

If $P^{\text{AMM}} < P^{\text{user}}_{\text{cash-out}}$, an arbitrageur would:

1. Buy q \$TOKS in exchange for x^{in} \$RES in the AMM at current AMM price, i.e. $q = \frac{x^{in}}{P^{\text{AMM}}}$;
2. Cash-out the q \$TOKs for x^{out} at current revnet cash-out price, i.e. $x^{out} = P^{\text{user}}_{\text{cash-out}} q$
3. Since $P^{\text{AMM}} < P^{\text{user}}_{\text{cash-out}}$, then $x^{out} > x^{in}$

Thus, if $P^{\text{AMM}} < P^{\text{user}}_{\text{cash-out}}$ an arbitrageur buys \$TOK through the AMM, cashing them out in the Revnet for \$RES.

Thus, the user's cash-out price for redeeming q tokens $P^{\text{user}}_{\text{cash-out}}(q)$ defines the effective floor price of \$TOK for a redemption size q, i.e. the *redemption-dependent price floor*:

$$\boxed{\tilde{P}_{\text{floor}}(q) = (0.975)^2 \frac{B}{S} \left[(1 - r_k) + r_k \frac{(0.975)q}{S} \right] = P^{\text{user}}_{\text{cash-out}}(q)} \tag{30}$$

However, since $P^{\text{user}}_{\text{cash-out}}$ is an increasing function of the redeemed quantity q, larger redemptions yield progressively higher per-token payouts. To obtain an absolute floor that is independent of the cash-out size, we consider the marginal redemption price at infinitesimal q, which defines the effective *redemption-independent price floor*:

$$\boxed{P_{\text{floor}} = \lim_{q \to 0} \tilde{P}_{\text{floor}}(q) = (1 - r_k)(0.975)^2 \frac{B}{S} \approx (1 - r_k) \cdot 0.951 \cdot \frac{B}{S}} \tag{31}$$

This marginal value represents the lowest attainable redemption price and therefore constitutes a strict lower bound for the rational secondary-market price of \$TOK.

Price Corridor

The arbitrage opportunities define a price corridor for the \$TOK price, i.e. at any time t, this holds

$$P^{\text{floor}} \leq P^{\text{AMM}} \leq P^{\text{ceil}} \tag{32}$$

This is the window for the emergence of a market price P^{AMM}.

3.2 Price Corridor Dynamics

The **price ceiling** P^{ceil} increases automatically over time, since issuance price schedules are hard-coded to rise independently from the network activity (See Eq. 2).

The **price floor** P^{floor} is proportional to the ratio between the current treasury state B and the current circulating tokens S, i.e. $P^{\text{floor}} \propto \frac{B}{S}$. For this reason, P^{floor} evolves as a function of network activity.

To analyze how the price floor changes, we take the differential of $P^{\text{floor}} = k\frac{B}{S}$:

$$dP^{\text{floor}} = k \cdot d\left(\frac{B}{S}\right) = k\frac{S\,dB - B\,dS}{S^2} = P^{\text{floor}} \cdot \frac{S\,dB - B\,dS}{BS} \tag{33}$$

The floor price increases when $dP^{\text{floor}} > 0$, which requires:

$$S\,dB - B\,dS > 0 \quad \Leftrightarrow \quad \frac{dB}{dS} \begin{cases} > \dfrac{B}{S}, & \text{if } dS > 0, \\ < \dfrac{B}{S}, & \text{if } dS < 0. \end{cases}$$

Equivalently, for discrete events at time t^+ causing changes Δb and Δs:

$$\frac{\Delta b}{\Delta s} \begin{cases} > \dfrac{B(t)}{S(t)}, & \text{if } \Delta s > 0, \\ < \dfrac{B(t)}{S(t)}, & \text{if } \Delta s < 0. \end{cases} \tag{34}$$

Price floor during issuance: During an issuance, the revnet receives x base assets, minting new tokens at the ceiling price:

$$\Delta b = x > 0$$
$$\Delta s = \frac{x}{P^{\text{ceil}}} > 0$$

Therefore:

$$\frac{\Delta b}{\Delta s} = P^{\text{ceil}}$$

The condition in Eq. 34 holds if:

$$P^{\text{ceil}} > \frac{B}{S} \approx P^{\text{floor}}$$

Thus, if the price ceiling exceeds the price floor, an issuance event increases the price floor. Otherwise, the price floor decreases.

Price floor during a cash-out: During a cash-out, the revnet burns q circulating tokens, redeeming $C_{\text{tot}}(q)$ base assets to the user:

$$\Delta b = -C_{\text{tot}}(q) < 0$$
$$\Delta s = -q < 0$$

The price floor increases if the condition in Eq. 34 holds, requiring:

$$\frac{\Delta b}{\Delta s} = \frac{C_{\text{tot}}(q)}{q} < \frac{B}{S}$$

By definition of the cash-out function (See Eq. 5), this condition is always satisfied:

$$\frac{C_{\text{tot}}(q)}{q} = \frac{B}{S}\left[(1 - r_k) + r_k \frac{q}{S}\right] < \frac{B}{S}$$

where the inequality holds since $(1 - r_k) + r_k \frac{q}{S} < 1$ for $q < S$. Thus, cash-outs always increase the price floor.

Price floor during a loan: During the **issuance of a loan**, the user burns q_c circulating tokens, borrowing L_{gross} base assets from the revnet:

$$\Delta b = -L_{\text{gross}}(1 - F_{\text{prepaid}}) < 0$$
$$\Delta s = -q_c < 0$$

By construction (See Eq. 10), borrowers cannot extract more than they could through direct redemption:

$$L_{\text{gross}}(q_c) = C_{\text{tot}}(q_c, S_{\text{eff}}, B_{\text{eff}}) \leq C_{\text{tot}}(q_c, B, S)$$

Therefore, the condition in Eq. 34 holds:

$$\frac{L_{\text{gross}}(1 - F_{\text{prepaid}})}{q_c} < \frac{L_{\text{gross}}}{q_c} \leq \frac{C_{\text{tot}}(q_c)}{q_c} < \frac{B}{S}$$

Thus, loan issuances always increase the price floor.

During the **repayment of a loan**, the user pays P_{repay} base assets to reclaim q_c tokens. For full repayment:

$$\Delta b = L_{\text{gross}} > 0$$
$$\Delta s = q_c > 0$$

Since:

$$\frac{\Delta b}{\Delta s} = \frac{L_{\text{gross}}}{q_c} \leq \frac{C_{\text{tot}}(q_c)}{q_c} < \frac{B}{S}$$

the condition in Eq. 34 is not satisfied, and the price floor decreases.

During the **liquidation of a loan**:

$$\Delta b = 0$$
$$\Delta s = 0$$

Therefore, $\Delta P^{\text{floor}} = 0$.

Price floor during auto-issuance: During an auto-issuance event:

$$\Delta b = 0$$
$$\Delta s = \mathcal{A}_k > 0$$

Since $\Delta b = 0$ while $\Delta s > 0$, we have $\frac{\Delta b}{\Delta s} = 0 < \frac{B}{S}$, so the condition in Eq. 34 is never satisfied. Thus, auto-issuances always decrease the price floor.

3.3 Summary

The arbitrage mechanisms between the Revnet and external AMMs establish fundamental properties of the $TOK price dynamics:

- **Price Corridor**: Arbitrage opportunities create a well-defined price corridor where $P^{\text{floor}} \leq P^{\text{AMM}} \leq P^{\text{ceil}}$ at any time t. This bounds the market price between the cash-out value (floor) and user issuance price (ceiling).
- **Ceiling Dynamics**: The price ceiling P^{ceil} increases monotonically over time according to the hardcoded issuance schedule, independent of network activity.
- **Floor Dynamics**: The price floor $P^{\text{floor}} \propto \frac{B}{S}$ evolves based on network activity as detailed in Table 2.
- **Value Accrual**: The increasing price floor mechanism ensures that value accrues to $TOK holders through most user interactions with the Revnet, creating a positive feedback loop between network usage and token value.
- **Loan**: The complete loan cycle (origination followed by full repayment within prepaid period) has a non-zero effect on the price floor, with the increase in the floor price caused by the prepaid fee.
- **Autoissuance**: Auto-issuance events have a negative effect on the price floor, as they increase supply without a corresponding increase in the treasury balance.

Table 2: Effect of Network Events on Price Floor

Event	Δb	Δs	**Price Floor Effect**	**Condition**
Token Issuance	> 0	> 0	**Increases**	if $P^{\text{ceil}} > P^{\text{floor}}$
Cash-out	< 0	< 0	**Increases**	Always
Loan Issuance	< 0	< 0	**Increases**	Always
Loan Repayment	> 0	> 0	**Decreases**	Always
Loan Liquidation	0	0	**No Change**	–
Auto-issuance	0	> 0	**Decreases**	Always

These dynamics create a system where the token price has both upward pressure from the rising ceiling and support from an activity-driven floor, establishing a sustainable value appreciation mechanism tied to network utilization.

4 Loan Solvency Analysis

In this section, we prove that the Revnet loan system is *always* solvent, meaning that the system can honor all obligations to both loan holders and circulating token holders, regardless of loan activity or defaults.

4.1 The Solvency Guarantee

A loan system is **solvent**: *if all circulating token holders can redeem their proportional share of the treasury at any time, regardless of outstanding loan activity or defaults.*
In the context of Revnets, solvency requires that the circulating token supply $S(t)$ can fully redeem the treasury balance $B(t)$ at all times.

Theorem 4.1 (Loan Solvency)**.** *The Revnet loan system maintains solvency for any number of loans, any loan sizes, and any sequence of operations, including defaults.*

Proof. By the functional form of the redemption curve (Eq. 5), for any state (S, B):

$$C_k(S; S, B) = B. \tag{35}$$

Consider issuing n loans sequentially:

$$S(t) = S(t_0) - \sum_{i=1}^{n} q_i,$$
$$B(t) = B(t_0) - \sum_{i=1}^{n} L_i.$$

where $L_i = C_k(q_i; S_{\text{eff}}(t_i), B_{\text{eff}}(t_i))$ is determined by the effective state at loan issuance.

Applying Eq. 35 to the current state:

$$C_k(S(t); S(t), B(t)) = B(t).$$

Therefore, the remaining circulating supply can always fully redeem the remaining treasury. This ensures that each token holder can always access the backing capital of their tokens, maintaining solvency regardless of outstanding loan activity. □

4.2 Overcollateralization

The loan system not only maintains solvency but is *overcollateralized*: borrowers receive strictly less than their proportional share of the treasury. Indeed, since:

$$C_k(q; S, B) \le \frac{q}{S} \cdot B, \forall r_k \in [0, 1]$$

Then we can define the overcollateralization margin as:

$$\text{Margin} = \frac{q}{S}B - C_k(q; S, B) = \frac{q}{S}B \cdot r_k \left(1 - \frac{q}{S}\right). \tag{36}$$

As a percentage of the fair share:

$$\text{Margin } \% = r_k \left(1 - \frac{q}{S}\right) \times 100\%. \tag{37}$$

4.3 Solvency Under Default

In Revnet loans, ”liquidation” is purely administrative: collateral is burned at loan creation (not at liquidation), and liquidation after $T_{\text{liquidation}} = 3650$ days only updates accounting variables $S_{\text{collateral}}$ and B_{borrowed} without changing the actual state (S, B).

Corollary 4.2 (Solvency Under Default)**.** *If all outstanding loans default, the system remains solvent.*

Proof. Upon default, the actual state $(S(t), B(t))$ remains unchanged since collateral was already burned and treasury funds were already subtracted from the treasury at loan creation. Therefore:

$$C_k(S(t); S(t), B(t)) = B(t),$$

and solvency is maintained. □

Unlike traditional lending systems where liquidation involves selling collateral to recover funds, Revnet loans are ”pre-liquidated” at issuance. The collateral is immediately burned, and $RES tokens are lent to the borrower in an amount that always guarantees system solvency. What is called ”liquidation” in Revnets is merely an accounting update that occurs after the repayment window expires, with no actual assets changing hands. Moreover, defaults improve the backing ratio: as proven in the price floor dynamics (Sec. 3.2), loan issuance increases B/S while repayment decreases it. When a loan defaults, the B/S increase from issuance becomes permanent, forever improving the backing for remaining token holders.

4.4 Summary

The Revnet loan system achieves guaranteed solvency through its mathematical construction:

1. **Total redemption property:** The redemption curve satisfies $C_k(S; S, B) = B$ for any state, ensuring that remaining tokens can always redeem the remaining treasury regardless of loan activity.

2. **Effective values accounting:** The state variables S_{eff} and B_{eff} ensure all loans are priced against a consistent reference state, preventing sequencing advantages.

3. **Pre-liquidation at issuance:** By burning collateral and borrowing overcollateralized funds at loan creation, the system eliminates traditional liquidation risks and ensures defaults cannot compromise solvency.

Moreover:

- **Unconditional solvency:** The system remains solvent for any number of loans, any loan sizes, and any default scenario

- **Overcollateralization:** Each loan is automatically overcollateralized by margin $r_k(1-q/S)$, providing additional safety

- **Defaults improve backing:** When loans default, remaining token holders benefit. Indeed, both the treasury and the supply remain reduced, but the since the

- **No liquidation risk:** The pre-liquidation design eliminates misliquidation risk, oracle dependence, and liquidation cascades

This solvency guarantee emerges directly from the mathematical properties of the redemption curve, without requiring external oracles, active liquidation mechanisms, or governance intervention.

5 Rational Actor Analysis: Cash-out vs Hold vs Loan Decision

A token holder with q tokens at time t can sell these tokens through:

- Direct cash-out: $P^{\text{sell}}(t_0, q) = \tilde{P}^{\text{floor}}(t_0, q)$ (See Eq. 30) accounting for network (REV and NANA).
- AMM sale: $P^{\text{sell}}(t_0, q) = P^{\text{AMM}}(t_0, q)$

The holder thus faces three fundamental strategies:

Strategy A: Exit immediately
Redeem tokens at time t_0 for immediate liquidity:

$$X(t_0) = qP^{\text{sell}}(t_0, q) \tag{38}$$

Strategy B: Hold
Maintain token position until future time t_1 and exit:

$$X(t_1) = qP^{\text{sell}}(t_1, q) \tag{39}$$

Strategy C: Loan
Borrow against tokens at time t_0, repay at t_1, then exit:

$$\begin{aligned}
\text{Borrow at } t_0 : X_l &= aC_k(q; S_{\text{eff}}(t_0), B_{\text{eff}}(t_0)) \\
\text{Repay at } t_1 : L_{\text{gross}} &= C_k(q; S_{\text{eff}}(t_0), B_{\text{eff}}(t_0)) \\
\text{Exit at } t_1 : L_{\text{gross}} &= qP^{\text{sell}}(t_1, q) \\
\text{Cost} : L_{\text{gross}} - X_l &= (1-a)C_k(q; S_{\text{eff}}(t_0), B_{\text{eff}}(t_0))
\end{aligned}$$

where a represents net loan proceeds after fees:[5]

$$a = \begin{cases} 0.945, & \text{if } f_{\text{prepaid}} = 25 \text{ (6-month window)} \\ 0.625, & \text{if } f_{\text{prepaid}} = 500 \text{ (10-year window)} \end{cases} \tag{40}$$

In this section, we analyze these strategies from two perspectives:

1. **Non-forward-looking investors**: Compare only immediate payoffs (Exit vs loan at t_0), treating holding as not an option
2. **Forward-looking investors**: Compare all three strategies over time horizon $[t_0, t_1]$, accounting for price appreciation and investment opportunities

[5] $L_{\text{gross}} = C_k(q; S_{\text{eff}}, B_{\text{eff}})$ and $L_{\text{net}} = L_{\text{gross}}(1 - F_{\text{NANA}} - F_{\text{REV}} - F_{\text{prepaid}}) = C_k(q)a$, where $F^{\text{NANA}} = \frac{25}{1025} \approx 0.025$, $F^{\text{REV}} = \frac{10}{1010} \approx 0.01$, and $F^{\text{prepaid}} = \frac{f_{\text{prepaid}}}{1000 + f_{\text{prepaid}}}$ with $f_{\text{prepaid}} \in [25, 500]$, giving $a \approx 1 - 0.035 - F^{\text{prepaid}}$. In particular $a \in [0.632, 0.941]$.

5.1 Non-Forward-Looking Investor: Loan vs Cash-out

A non-forward-looking investor prioritizes immediate liquidity access at time t_0, disregarding future token appreciation potential. For such an investor, **holding is not considered**. The decision reduces to comparing the immediate payoffs from cash-out versus a loan.
We consider $S_{\text{eff}} = S$ and $B_{\text{eff}} = B$ i.e., there are no outstanding loans. Then, cash-out dominates a loan when:

$$X_{\text{exit}} > X_{\text{loan}} \quad \Rightarrow \quad 0.975 C_k(0.975q; S, B) > a C_k(q; S, B)$$

Substituting the redemption curve (Eq. 5) and defining $x = q/S$ as the fraction of total supply:

$$(0.975)^2 [(1 - r_k) + r_k \cdot 0.975 \cdot x] > a \cdot [(1 - r_k) + r_k x]$$

Rearranging gives the threshold condition:

$$\boxed{a < a^*(x, r_k) = (0.975)^2 \frac{(1 - r_k) + r_k \cdot 0.975 \cdot x}{(1 - r_k) + r_k \cdot x}} \tag{41}$$

For $a < a^*$, cash-out dominates; for $a > a^*$, loan dominates.
The threshold a^* depends on:

- **Position size** $x = q/S$: Larger positions benefit more from the convex redemption curve
- **Cash-out tax** r_k: Higher taxes favor loans over cash-outs
- **Prepaid-fee** f_{prepaid}: Increasing the prepaid fee rapidly reduces the immediate payoff of loans of any size. For $f_{\text{prepaid}} > 4\%$, cash-outs always access more liquidity than loans.

Figure 3 shows the decision regions. Key findings:

- **If** $r_k < 39.16\%$: Even with minimal loan fees ($a = 0.945$, shortest repayment window), cash-out always provides more immediate liquidity. *Cash-out dominates for all myopic users.*
- **If** $r_k > 39.16\%$: Loans become competitive for larger positions (x high) with short repayment windows (low f_{prepaid}, high a). The region where loans dominate shrinks as repayment windows lengthen.

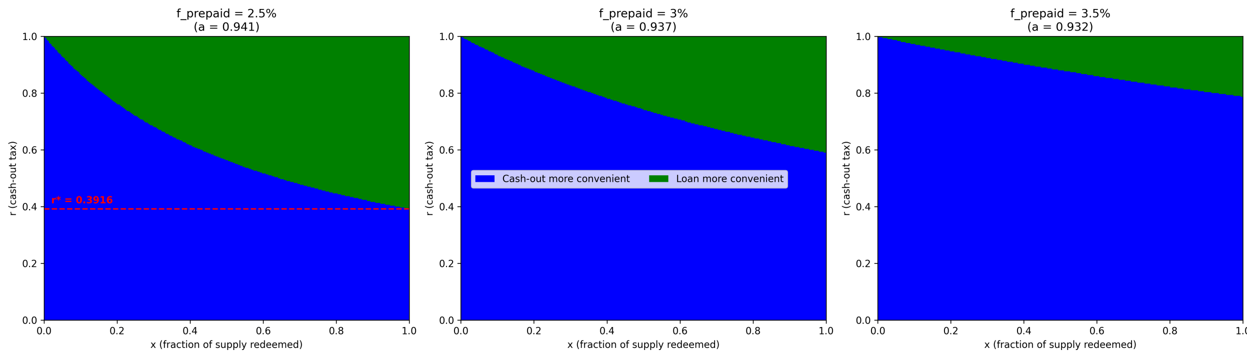

Figure 3: Decision regions for non-forward-looking investors comparing immediate payoffs. Blue: cash-out dominates; green: loan dominates. Panels show different prepaid fee levels: $f_{\text{prepaid}} = 2.5\%$ (6 months), 3% (7 months), and 3.5% (8 months). The red dashed line in the first panel marks the critical tax rate $r_k^* \approx 39.16\%$, above which loans become competitive even for short repayment windows.

Analytic confirmation: Differentiating Eq. 41 shows that

$$\frac{\partial a^*}{\partial r_k} = 0.975^2 \frac{-0.025x}{[1 + r_k(x-1)]^2} \leq 0 \quad \forall x, r_k,$$

so higher r_k reduces a^*, expanding the region where loans dominate.

For myopic investors focused solely on immediate liquidity:

1. Low cash-out tax regimes ($r_k < 39\%$) favor direct redemption.
2. High cash-out tax regimes ($r_k > 39\%$) can make loans attractive, especially for large holders with short repayment windows.
3. Longer prepayment periods (higher fees) progressively favor cash-outs over loans.

In this analysis we assumed $S_{\text{eff}} = S$ and $B_{\text{eff}} = B$ (no outstanding loans). When loans exist, $S_{\text{eff}} > S$ and $B_{\text{eff}} > B$, in which case $C(q, S, B) > C(q, S_{\text{eff}}, B_{\text{eff}})$, narrowing the parameter space where loans are preferred.

5.2 Forward-Looking Investor: Exit vs Hold vs Loan

A forward-looking investor evaluates strategies over time horizon $[t_0, t_1]$, accounting for expected price appreciation and investment opportunities. Now **holding becomes a viable option**, expanding the decision space to three strategies.
Let R denote the return rate on alternative investments (e.g., DeFi yields, stablecoin lending, ...) between t_0 and t_1.

Strategy A: Exit now Sell immediately and invest proceeds:

$$W_A(t_1) = P^{\text{sell}}(t_0, q) \cdot q \cdot (1 + R) \tag{42}$$

Strategy B: Hold Maintain position and sell at t_1:

$$W_B(t_1) = P^{\text{sell}}(t_1, q) \cdot q \tag{43}$$

Strategy C: Loan Borrow, invest proceeds, repay, then sell:

$$W_C(t_1) = aC_k(q)(1 + R) - C_k(q) + P^{\text{sell}}(t_1, q) \cdot q \tag{44}$$

where $C_k(q) = C_k(q; S(t_0), B(t_0))$ abbreviates the borrowable amount.

5.2.1 Exit vs Hold

Exit dominates when $W_A > W_B$:

$$\boxed{(1 + R) > \frac{P^{\text{sell}}(t_1, q)}{P^{\text{sell}}(t_0, q)}} \tag{45}$$

Exit if investment returns exceed expected token appreciation. This naturally sorts by conviction: pessimists exit, optimists hold.

5.2.2 Loan vs Hold

Loan dominates when $W_C > W_B$:

$$aC_k(q)(1 + R) - C_k(q) + P^{\text{sell}}(t_1, q) \cdot q > P^{\text{sell}}(t_1, q) \cdot q$$
$$C_k(q)[a(1 + R) - 1] > 0$$

Thus:

$$\boxed{R > R^* = \frac{1 - a}{a}} \tag{46}$$

Critical thresholds:

- 6-month loan ($a = 0.941$): $R^* = 5.8\%$ (11.6% annualized)
- 10-year loan ($a = 0.632$): $R^* = 60\%$ (6% annualized)

When productive investment opportunities exist ($R > R^*$), holding idle tokens is suboptimal. Loans enable capital deployment while maintaining exposure, strictly dominating holding.

5.2.3 Loan vs Exit (Baseline: $R = 0$)

Consider the case without investment opportunities. Loan dominates exit when $W_C > W_A$ with $R = 0$:

$$P^{\text{sell}}(t_1, q) \cdot q - (1-a)C_k(q) > P^{\text{sell}}(t_0, q) \cdot q$$

Rearranging:

$$\boxed{\Delta P^{\text{sell}} > (1-a)\tilde{P}^{\text{floor}}(t_0, q)} \tag{47}$$

where $\tilde{P}^{\text{floor}}(t_0, q) = C_k(q, S(t_0), B(t_0))/q$ is the redemption-size-dependent floor price (Eq. 30). Thus, without investment opportunities, loans are justified only by expected price appreciation exceeding the loan cost $(1-a)$. For 6-month loans, this requires 5.5% appreciation ($\sim$ 11 % annual growth); for 10-year loans, 37.5% ($\sim$ 3.7 % annual growth).

No AMM (direct cash-out only). When no AMM is present, then the user can exit only through the cash-out mechanism, thus $P^{\text{sell}}(t, q) = P^{\text{floor}}(t, q)$, and the condition in Eq. 47 becomes:

$$q\tilde{P}^{\text{floor}}(t_1, q) - q\tilde{P}^{\text{floor}}(t_0, q) > (1-a)q\tilde{P}^{\text{floor}}(t_0, q)$$

Substituting:

$$0.975C_k(0.975q; S(t_1), B(t_1)) - 0.975C_k(0.975q; S(t_0), B(t_0)) > (1-a)C_k(q; S_{\text{eff}}(t_0), B_{\text{eff}}(t_0))$$

Assuming no outstanding loans initially ($S_{\text{eff}} = S$, $B_{\text{eff}} = B$), this becomes:

$$0.975C_k(0.975q; S(t_1), B(t_1)) > (1-a)C_k(q; S(t_0), B(t_0)) + 0.975C_k(0.975q; S(t_0), B(t_0))$$

Expanding the redemption curves we reach the condition:

$$0.975^2 \frac{B(t_1)}{S(t_1)} \left[(1-r_k) + r_k \frac{0.975q}{S(t_1)}\right] > (1-a)\frac{B(t_0)}{S(t_0)} \left[(1-r_k) + r_k \frac{q}{S(t_0)}\right] + 0.975^2 \frac{B(t_0)}{S(t_0)} \left[(1-r_k) + r_k \frac{0.975q}{S(t_0)}\right]$$

Case 1: No cash-out tax ($r_k = 0$)

When $r_k = 0$, Eq. 5.2.3 simplifies to:

$$\frac{B(t_1)}{S(t_1)} > \frac{1 - a + 0.975^2}{0.975^2} \frac{B(t_0)}{S(t_0)} \quad \Rightarrow P^{\text{floor}}(t_1) > \gamma^* P^{\text{floor}}(t_0)$$

The required floor price growth factor is:

$$\gamma = \frac{P^{\text{floor}}(t_1)}{P^{\text{floor}}(t_0)} > \gamma^* \tag{48}$$

For a 10-year loan, $a = 0.632$, $\gamma^* \approx 1.368$, meaning a 36.8% floor price appreciation over the loan period is required to justify taking a loan. This translates to approximately 3.07% annual growth. For a 6-months loan $a = 0.941$, thus $\gamma^* \approx 1.06$ In this case, 6% floor price appreciation is needed, which is about 12% annually.

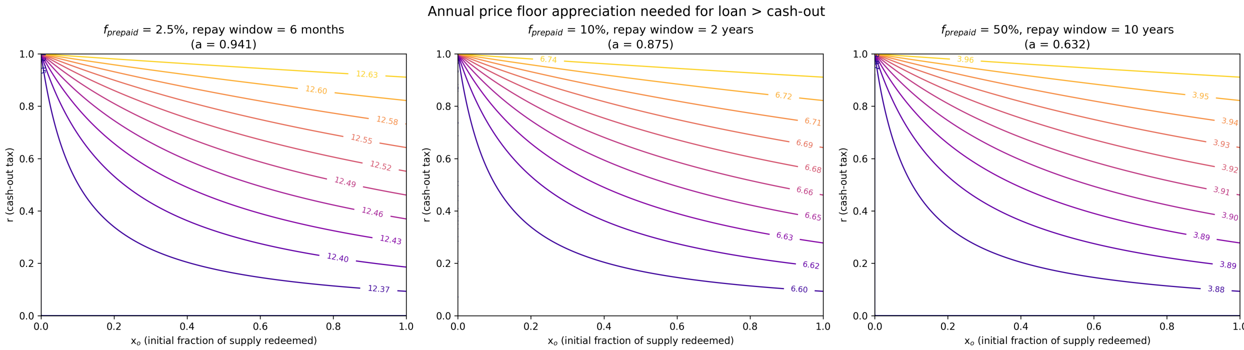

Figure 4: Annual price floor appreciation rate (in %) required for the loan strategy to outperform immediate cash-out, as a function of position size (x_0, fraction of total supply) and cash-out tax (r). The three panels correspond to different prepaid fee levels: $f_{\text{prepaid}} = 2.5\%$ (6-month interest-free window, $a = 0.941$), $f_{\text{prepaid}} = 10\%$ (2-year window, $a = 0.875$), and $f_{\text{prepaid}} = 50\%$ (10-year window, $a = 0.632$). Higher contours indicate parameter regions requiring stronger floor price growth to justify loans over immediate cash-out. The plots show that higher cash-out taxes *increase* the required growth threshold, and that shorter repayment windows (lower f_{prepaid}) increase the annualized rate requirement.

Case 2: With cash-out tax ($r_k > 0$)

For small positions where $q/S \to 0$, the condition approximates to:

$$\gamma^* \approx 1 + \frac{1-a}{0.975^2}$$

This is identical to the $r_k = 0$ case, showing that cash-out tax has minimal impact on small positions.
For larger positions, defining $x_0 = q/S(t_0)$ and assuming $x_1 \approx x_0$ (position size remains small relative to total supply changes): For larger positions, defining $x_0 = q/S(t_0)$ and assuming $x_1 \approx x_0$ (position size remains small relative to total supply changes), the full condition from Eq. 5.2.3 gives:

$$\gamma^* = \frac{(1-a)[(1-r_k) + r_k x_0] + 0.975^2[(1-r_k) + 0.975 r_k x_0]}{0.975^2[(1-r_k) + 0.975 r_k x_0]} \tag{49}$$

Simplifying:

$$\gamma^* = 1 + \frac{(1-a)[(1-r_k) + r_k x_0]}{0.975^2[(1-r_k) + 0.975 r_k x_0]} \tag{50}$$

As it is shown in Fig. 4 as r_k increases, the price floor appreciation γ^* *increases*, making loans *less* attractive relative to immediate cash-out.[6] Economically, higher cash-out taxes create two opposing effects: they penalize immediate redemption but also reduce the effective value obtained from loans (since loans are priced against the full redemption curve). The net result is that higher taxes require *more* future price growth to justify the loan strategy over immediate exit.

[6] This occurs because:

- The numerator $(1-a)[1 - r_k(1-x_0)]$ decreases at rate $(1-x_0)$
- The denominator $b[1 - r_k(1-bx_0)]$ decreases at rate $(1-bx_0)$
- Since $b < 1$ and $x_0 > 0$, we have $(1-bx_0) > (1-x_0)$
- Therefore the denominator shrinks faster, causing the fraction to increase

With AMM When an AMM exists, arbitrage ensures $P^{\text{AMM}} \geq P^{\text{floor}}$, so the AMM always provides a weakly better exit price than direct redemption. Therefore, $P^{\text{sell}}(t, q) = P^{\text{AMM}}(t, q)$ and Eq. 47 becomes:

$$\Delta P^{\text{AMM}} > (1-a)\tilde{P}^{\text{floor}}(t_0, q)$$

Normalizing by the floor price:

$$\frac{\Delta P^{\text{AMM}}(q)}{\tilde{P}^{\text{floor}}(t_0, q)} > (1-a) \tag{51}$$

This reveals a key insight: the loan strategy is profitable when the AMM price appreciates relative to the initial size-dependent floor price by more than the loan fee percentage $(1-a)$.

- For $a = 0.941$ (6-month repayment window): $(1-a) = 5.9\%$ total, or ∼11.8% annualized
- For $a = 0.632$ (10-year repayment window): $(1-a) = 36.8\%$ total, or ∼3.68% annualized

5.3 Summary

Rational actor behavior in revnets divides into two main regimes:

Myopic users. For immediate liquidity seekers, cash-out strictly dominates when the cash-out tax $r_k < 39.16\%$. Loans can become preferable for immediate liquidity access only when $r_k > 39.16\%$, for large positions, short repayment windows (high a), and few outstanding loans. Thus, low taxes drive exits while high taxes internalize liquidity through loans.

Forward-looking users. Strategic participants compare all three options:

- Exit instead of holding if external returns exceed expected token appreciation: $(1+R) > \frac{P^{\text{sell}}(t_1)}{P^{\text{sell}}(t_0)}$.
- Borrow instead of holding if investment opportunities justify the fee: $R > \frac{1-a}{a}$.
- Between exit and loan, choose loan when expected appreciation exceeds its cost: $\Delta P^{\text{sell}} > (1-a)P^{\text{floor}}(t_0, q)$.

Where a represents the net loan proceeds after fees (See Eq. 40), and R denote the return rate on alternative investments.
With an AMM, the last condition depends on the AMM price growth ΔP^{AMM}; without one, it depends on floor price appreciation ΔP^{floor}. Higher cash-out taxes r_k increase the required growth threshold, needed for loans to be more convenient.

6 Price Ceiling Runaway

Revnet issuance schedules encode a monotonically increasing ceiling price $P^{\text{ceil}}(t)$, independent of market activity. When the secondary market price P^{AMM} remains persistently below the ceiling ($P^{\text{AMM}} \ll P^{\text{ceil}}$), inbound payments are routed to the AMM instead of minting new tokens.
In this *runaway regime*, new issuance halts: inbound $RES flows no longer directly increase the treasury B, and circulating supply $S(t)$ changes only through redemptions, loan activity, and automints. Since the price floor $P^{\text{floor}} \propto B/S$, this routing effect can temporarily stall floor growth coming from issuance.

6.1 The Runaway Regime

During a runaway, the system's behavior depends critically on the type of demand sustaining token activity.

Service-coupled demand. When token purchases are tied to consuming a real service (e.g., API credits, access fees, ...), payments continue regardless of short-term market price. These inflows route to the AMM, buying tokens and reducing AMM-side supply, which gradually pushes P^{AMM} upward.
As the secondary price approaches the issuance ceiling, the routing logic reverses: new buyers again mint through issuance rather than the AMM, increasing B and expanding the treasury. This transition restores the normal issuance regime, an endogenous *recovery path* in which sustained service payments realign market and issuance prices and resume floor growth.
However, if the ceiling schedule rises too quickly relative to service inflows, $P^{\text{ceil}}(t)$ may keep escaping upward faster than the market can follow. In that case, the system remains trapped in AMM routing: inflows continuously buy from the pool, reducing its token reserves and increasing price slippage. Eventually, issuance will resume once trades become large enough to make AMM execution less favorable than direct minting, but by then secondary liquidity may have deteriorated: spreads widen, volatility rises, and the user experience worsens.
Upward price pressure also makes loans more attractive. As P^{AMM} appreciates toward P^{ceil}, the expected price appreciation ΔP^{sell} increases relative to the redemption baseline, improving the condition for taking a loan (Eq. 47). The prepaid loan fee F_{prepaid} is treated as an inbound payment to the source Revnet and, under runaway routing, is used to buy tokens from the AMM. These fee purchases support the AMM price, while completed loan cycles preserve circulating liquidity since collateral is reminted upon repayment.
In service-driven systems, even during runaway, continuous activity thus reinforces itself: service payments and loan fees supply buy pressure that lifts the market price, restores issuance when feasible, and strengthens the price floor. Yet, if ceiling growth is excessive, this recovery can occur at the cost of degraded secondary market quality.

Pure speculative demand. When demand is primarily speculative, AMM routing during runaway only shifts where traders acquire tokens, and it does not guarantee persistent buy pressure. Without service-linked inflows, price movements depend solely on traders' expectations.
In this context, the feedback loop can become unstable. As confidence weakens, rational holders may prefer to cash out rather than lend or hold. Cash-outs *raise the floor* by reducing S faster than B, increasing the redeemable value per remaining token. However, they simultaneously deplete the circulating supply and possibly the liquidity on the AMM, leading to higher slippage and volatility. Thin liquidity discourages new entrants and amplifies price swings, reinforcing outflows.

Moreover, a higher floor raises the benchmark for new loans: since each token now represents a higher redemption value, the opportunity cost of using it as collateral increases. Unless users expect meaningful appreciation or yield, loans become less attractive relative to immediate redemption. This shift in incentives can extend the cascade until circulating supply stabilizes at a much lower level or the market price reconverges to issuance.

6.2 Design Implications

Runaway dynamics are not a failure but a structural regime. Their persistence and impact can be tuned through design levers:

- **Ceiling slope calibration.** The rate of increase of $P^{\text{ceil}}(t)$ should roughly match the expected organic inflow rate from service demand. If ceiling growth is too steep relative to activity, issuance remains inactive for long stretches, draining AMM liquidity and worsening market quality. A slower slope maintains proximity between market and issuance prices, keeping arbitrage pathways active and secondary trading healthy.
- **AMM deployment and depth.** Deep AMM liquidity should be bootstrapped only after a baseline of recurring service inflows is established. Prematurely deep AMMs can trap the system in a runaway phase where issuance never reactivates, while too little liquidity amplifies volatility.
- **Cash-out tax and redemption policy.** A redemption tax that retains part of the payout within the treasury (as in $r_k > 0$) ensures that every exit strengthens the floor, providing stability even during heavy redemptions.
- **Communication and interface design.** Displaying both $P^{\text{ceil}}(t)$ and the realized $P^{\text{floor}}(t)$ might help users understand that floor growth is activity-driven, not speculative. This discourages panic selling and reinforces the perception of structural value accumulation over time.

6.3 Summary

Price ceiling runaway is a natural regime that occurs when issuance pricing outpaces effective demand. Its consequences depend on demand composition and liquidity conditions:

- **Service-coupled systems:** Runaway is typically temporary and self-correcting. Continuous inflows and fee routing lift the AMM price, trigger arbitrage, and restore issuance-driven treasury growth. If the ceiling schedule rises too quickly, the recovery still occurs but at the cost of reduced secondary liquidity and trading quality.
- **Speculative systems:** Runaway can evolve into a quality-degrading cascade with shrinking liquidity and rising floor. The process remains bounded by solvency and naturally halts as supply contracts or prices reconverge to issuance.

From a systemic viewpoint, loans outperform cash-outs by preserving liquidity and recycling fees into AMM support. Yet, individual rationality during speculative runaway may favor immediate redemptions. Balancing this tension through calibrated ceiling dynamics, AMM staging, and redemption parameters is central to sustaining healthy Revnet evolution and long-term market quality.

7 Conclusions and Practical Implications

This paper has formalized the cryptoeconomic mechanisms of Revnets: autonomous, tokenized financial structures governed by immutable smart contracts. Through mathematical analysis of issuance, redemption, and loan operations, we have derived price dynamics, proven solvency guarantees, and characterized rational actor behavior under varying parameter configurations. This section synthesizes these findings with practical implications for investors and protocol designers.

Value Accrual Mechanisms

Revnets operate through a dual-price system consisting of a deterministic ceiling and an activity-driven floor. The price ceiling $P^{\text{ceil}}(t) = P_{\text{issue},k}(t)/(1-\sigma_k)$ increases according to predetermined schedules, rising by factor $\gamma_k = 1/(1-\gamma_{\text{cut},k})$ at intervals Δt_k regardless of network activity. The price floor $P^{\text{floor}}(t) \approx 0.951(1-r_k)B(t)/S(t)$ responds to user interactions through changes in the backing ratio $B(t)/S(t)$.

Section 3.2 demonstrates that the price floor increases through three primary mechanisms. First, token issuances increase the floor when the ceiling price exceeds the backing ratio ($P^{\text{ceil}} > B/S$), a condition typically satisfied during normal operation. Second, redemptions unconditionally increase the floor, as the cash-out tax r_k ensures that treasury depletion occurs slower than supply reduction. Third, loan originations increase the floor through mandatory overcollateralization margins of $r_k(1-q/S)$ and fee-payment. Conversely, auto-issuances decrease the floor by increasing supply without corresponding treasury contributions, while loan repayments reverse the initial collateralization effect.

Investment Analysis Framework

The rational actor analysis in Section 5 establishes quantitative thresholds for optimal investment decisions. An investor holding q tokens at time t_0 should exit immediately when external investment returns (R) exceed expected token appreciation: $(1+R) > P^{\text{sell}}(t_1)/P^{\text{sell}}(t_0)$. For investors maintaining exposure, loans become preferable to holding when $R > (1-a)/a$, where a represents net proceeds after fees. With $a = 0.941$ (six-month repayment window), this threshold equals 6.3% over the loan period, meaning $\sim 12.6\%$ annually. With $a = 0.632$ (ten-year window), the threshold rises to 58.2%, meaning $\sim 5.82\%$ annually.

The cash-out tax parameter r_k critically influences investor behavior. When $r_k \lesssim 39.16\%$, direct redemption provides superior immediate liquidity compared to loans for all position sizes. Above this threshold, loans become increasingly attractive for large positions, particularly with shorter repayment windows. This behavioral shift affects system liquidity dynamics during market stress.

Section 4 proves that the loan system maintains unconditional solvency. The redemption curve property $C_k(S; S, B) = B$ ensures that circulating tokens can always redeem the full treasury regardless of outstanding loans. Loan defaults strengthen the backing ratio for remaining holders, as borrowed funds remain in the treasury while collateral tokens are permanently burned. This solvency emerges from the mathematical structure rather than external insurance or governance mechanisms.

Protocol Design Parameters

The analysis reveals specific parameter constraints and their systemic effects. The cash-out tax r_k determines the boundary between exit-dominant and loan-dominant regimes at approximately 19%. Below this threshold, redemptions dominate during liquidity events; above it, loans can become viable but require stronger growth expectations. Higher values of r_k increase per-redemption floor appreciation but also raise the required price appreciation for loans to dominate exits, as shown in Section 5.2.3.

The ceiling growth rate requires calibration against expected demand. Excessive growth rates relative to organic demand create "runaway" conditions where $P^{\text{AMM}} \ll P^{\text{ceil}}$, routing purchases through secondary markets rather than primary issuance. Section 6 demonstrates that service-coupled demand provides natural recovery mechanisms through sustained buy pressure, while purely speculative systems risk liquidity degradation cascades.

Auto-issuances create immediate floor dilution proportional to the ratio of minted tokens to existing supply. The jump condition $S(t^+) = S(t^-) + a$ with unchanged treasury $B(t^+) = B(t^-)$ mechanically reduces the backing ratio. Optimal scheduling requires preceding periods of organic floor appreciation sufficient to absorb this dilution.

System Properties and Boundaries

The mathematical framework establishes several invariant properties. Theorem 4.1 proves that solvency holds for any sequence of operations, loan sizes, and default scenarios. The overcollateralization margin maintains itself through the bonding curve structure without external intervention. Price discovery operates entirely through internal state variables (B, S), eliminating oracle dependencies and associated manipulation vectors.

The system exhibits clear operational boundaries. During runaway phases analyzed in Section 6, secondary market quality depends critically on demand composition. Service-coupled systems experience temporary disruption with eventual recovery as usage drives price convergence. Purely speculative systems face potential liquidity spirals where redemption cascades reduce both supply and AMM depth, increasing volatility and discouraging new entrants.

Parameter immutability, while ensuring predictability and elimination of governance risk, prevents adaptive responses to changing conditions. Founders must encode appropriate parameters at deployment based on realistic projections rather than ex-post adjustments.

Synthesis

Revnets implement a deterministic approach to tokenized value coordination through immutable rules rather than governance processes. The mathematical analysis confirms that these systems maintain solvency, accrue value through quantifiable mechanisms, and provide calculable downside protection. The price corridor structure bounds the price for secondary markets.

Successful deployment requires alignment between parameter selection and realistic demand projections. The ceiling growth trajectory must approximate expected organic growth to avoid extended runaway phases. The cash-out tax must balance between encouraging retention (high r_k) and maintaining liquidity options (low r_k). Auto-issuances should be minimized and strategically timed. Most critically, sustainable value accrual requires genuine economic activity (e.g. service revenues, product fees, or other non-speculative demand sources) to drive the mechanical appreciation processes identified in this analysis.

For investors, Revnets offer quantifiable value propositions with explicit mathematical boundaries on risk and return. For founders, they provide frameworks for creating autonomous economic systems, contingent on careful parameter calibration aligned with the constraints derived herein. The elegance of the model resides in its mechanical simplicity: predetermined rules that generate complex but analyzable dynamics, creating systems that operate independently of human intervention once deployed.