TECHNICAL OVERVIEW

RTP Curve Launch

Bonding-curve token launches that graduate, at a continuous price, into a Uniswap v4 pool whose liquidity no one can remove.

Document	RTP Curve Launch: Technical Overview
Version	1.0-draft.26
Date	November 2026
Status	Security review in progress
Code reference	Curve Launch contracts v1.0.0-r4, commit 1f339ee
Deployment	GIWA Sepolia testnet, earlier build 8cae3fc; Robinhood Chain mainnet after internal review

CONTENTS

About this document

This technical overview explains RTP Curve Launch in plain, verifiable terms: what the contracts do, what they cannot do, and which rules a creator, a trader, a platform or a referrer can rely on.

Abstract

RTP Curve Launch creates a token with a fixed supply and sells part of it on a constant-product bonding curve with a virtual quote reserve. When the quote raised, net of fees, reaches a threshold set per quote asset, the curve's reserves become the seed of the token's own Uniswap v4 pool at the same price and in the same transaction, and the single full-range position is held by a contract that has no way to remove it. A decaying opening-buy tax raises the cost of buying in the opening seconds. Trade fees are booked to a pull-based ledger and split between the creator, the protocol, the integrating platform and an optional referrer; after the migration a share of the protocol's fees funds a permissionless buy-and-burn. This paper describes the contracts at version v1.0.0-r4.

01 Design goals

02 Architecture

03 Bonding curve and graduation

04 Opening-buy tax

05 Fees, the ledger and creator keys

06 Security model

07 Trust assumptions and known limitations

08 Deployment, status and roadmap

09 Appendix: notation and glossary

10 Appendix: references and revision history

How to read the figures

Values marked testnet are the published testnet configuration of v1.0.0-r4; other values are bounds compiled into the contracts. Formulas use the notation in the appendix. Tables of examples show how the formulas behave; they are not projections for any project.

Status at the time of writing

Curve Launch runs on a public EVM testnet with test tokens only; the deployed instance comes from an earlier build, as set out in the security model. GIWA stays on testnet. Security review in progress: internal review rounds are under way and an external audit is being arranged. Mainnet deployment follows the review rounds and one complete verification run; the external audit may not be complete by then, and its results are published when it is.

Related papers

Three papers are published together on rtp.sh: *RTP Protocol: Overview*; *RTP Curve Launch: Technical Overview*; *RTP Fixed-Price Launch: Technical Overview*. Where two papers describe the same rule, the technical overview of that launch path is the reference.

Not an offer. Nothing in this document is an offer to sell, a solicitation to buy, or investment, legal or tax advice. Participation in any token launch carries the risk of total loss.

Design goals

A token launch fails its holders when liquidity can be pulled, when insiders trade on different terms, when parameters move after the launch, or when the move from the curve to an open market opens a price gap. Curve Launch removes each of these in code or states the limit openly.

1B tokens
fixed supply per launch in the testnet configuration, minted once

80% / 20%
sold on the curve, and placed in the v4 pool at graduation (testnet)

16x
price rise from the curve's opening to graduation (testnet)

4.2 ETH
graduation threshold, net of fees (testnet)

90% or 50%
starting opening-buy tax of the testnet tiers, decaying to zero in 15 or 30 s

10%
most of the supply the launcher's initial buy can receive

7 days
public delay on treasury, attestor and verification-gate changes

0 proxies
no upgrade path; a defect is fixed only by a new deployment

What holders can rely on

- Terms are snapshotted when a launch is created: supply, fee and split, opening tax, threshold and virtual reserve. Later configuration changes apply to new launches only.
- The price is continuous at graduation, and the graduated liquidity has no removal path in code. It is locked, not burned.
- No one, the operator included, can pause trading or claims, move funds credited to others or upgrade a contract.
- Protections are stated with their limits. The opening tax raises the cost of sniping; it does not make sniping impossible.

Architecture

Curve Launch is a compact set of contracts. One core holds curve accounting and the fee ledger; graduation hands each token to its own Uniswap v4 pool through an immutable graduator and fee hook.

Contract	Role
RtpCore	Launches, bonding-curve trading, opening-buy tax, pull-based fee ledger and claims, launch tiers, graduation hand-off, per-launch buyback reserve and time-locked role changes.
RtpToken	Minimal ERC-20 minted once per launch. No owner, no later mint, no transfer tax, no blacklist. Holders may burn their own tokens.
RtpGraduator	Opens the token's v4 pool at the exact curve price and holds its single full-range position permanently. No owner, no setters.
RtpFeeHook	Uniswap v4 hook that charges the pool fee, one value fixed for the deployment (1%), on pool swaps and credits it to the core ledger. Only the graduator can initialise pools or add liquidity.
RtpV4Router · RtpV4Quoter	Router and quoter for hooked pools: exact-in and exact-out, minimum-out bounds and deadlines; quotes equal settled amounts.
RtpBuybackEngine	Permissionless buy-and-burn from a launch's buyback reserve, priced against a time-weighted average and rate-limited. No owner, no setters. Part of v1.0.0-r4; deployed with the next testnet instance.
RtpCoreLens	Read-only previews: predicted token address, reserves, opening tax, buy and sell quotes. Holds no funds. Part of v1.0.0-r4; deployed with the next testnet instance.
Uniswap v4 PoolManager	On mainnet the chain's canonical PoolManager must be wired; the deploy pipeline refuses a private copy unless a documented fallback is chosen.

Bonding curve and graduation

Each launch mints its entire supply S once and opens a constant-product curve against the approved quote asset. The curve trades against a quote reserve Q + vQ, the real quote held plus a virtual reserve vQ, and a token reserve R. The same rule governs the pool after graduation, so the price is continuous.

```
cs = floor( S × (vQ + T) / (2 · vQ + T) )    R₀ = floor( cs × (vQ + T) / T )
```

```
buy with net quote n:  out = floor( R × n / (Q + vQ + n) )
```

```
sell of t tokens:  gross = min( Q, (Q + vQ) × t / (R + t) ), less the trade fee
```

cs is the supply sold on the curve, R_0 the starting token reserve (larger than cs, so part of it is virtual) and T the graduation threshold in real quote, net of fees. The price on the curve is (Q + vQ) / R. A sell never pays out virtual quote. When Q reaches T, vQ is set to zero and R to the real remaining supply S − cs: the pool holds the raised quote T and the reserved tokens, and its price T / (S − cs) equals the last curve price exactly. From the opening of the curve to graduation the price rises by $((vQ + T) / vQ)^2$.

Testnet configuration	Value
Supply S	1,000,000,000 tokens
Virtual quote reserve vQ / threshold T	1.4 ETH / 4.2 ETH
Sold on the curve cs / seeds the pool	800,000,000 (80%) / 200,000,000 (20%)
Opening price / graduation price	1.3125×10^{-9} / 2.1×10^{-8} ETH per token

Quote raised on the curve, ETH net of fees	0	1.4	2.8	4.2
Price, multiple of the opening price	1x	4x	9x	16x

The virtual reserve and the threshold are set per quote asset and snapshotted per launch; values may differ on other deployments.

How graduation works

- The buy that reaches the threshold is clamped to it; any surplus is refunded inside the same trade.
- In that transaction the graduator opens the token's Uniswap v4 pool at the exact curve price and seeds it with the raised quote and the reserved supply, as one full-range position that it holds permanently. There is no LP token and no function to remove liquidity.
- If the hand-over fails inside the graduating trade, for example because the buy ran short of gas, trading continues on the in-core pool at a continuous price until anyone completes the migration.
- Supply can only fall afterwards, through holder burns and the buy-and-burn.

Opening-buy tax

For a short window after a launch, buys pay a tax on top of the trade fee. It decays with the cube of the time left, so it falls fast and reaches zero at the end of the window.

$$\text{tax} = s0 \times ((w - t) / w)^3 \text{ for } t < w, \text{ and } 0 \text{ afterwards}$$

s0 is the tier's starting tax, w its window in seconds and t the seconds since launch. Fee plus tax never exceeds 99% of the spend, so the buyer always keeps at least 1%. Compiled bounds: s0 at most 99%, w at most 60 seconds. Sells pay no opening tax.

Testnet tier	s0	w	tax at 5 s	10 s	20 s
Strict tiers, the default included	90%	15 s	26.7%	3.3%	0%
Standard tier	50%	30 s	28.9%	14.8%	1.9%

Where the tax goes

The opening tax is fixed by the protocol per tier and is booked to the protocol treasury only. It is not shared with the creator, the platform or a referrer, so it cannot be recycled by self-dealing.

The launcher's initial buy. A buy made inside the launch transaction itself pays no opening tax (the trade fee still applies) and can receive at most 10% of the supply. It is the only exemption in the system.

A cost, not a prohibition. A same-block buyer who accepts the cost can still buy. The threat model quantifies what the tax does and does not prevent.

Fees, the ledger and creator keys

Every trade pays a basis-point fee. On the curve the fee and its split are snapshotted from the launch tier; in the pool the fee hook charges one fee fixed for the deployment and books it to the same split. Nothing is pushed during a trade: every share is credited to a ledger in RtpCore and claimed by its owner.

Item	Rule enforced by contract
Trade fee on the curve	Set per tier, more than 0 and at most 5%; 1% on every testnet tier.
Fee in the pool	One value for the deployment, at most 1%; 1% on testnet.
Split	Protocol, platform and referrer shares are set per tier and add up to at most 100%; the creator takes the rest, so no dust leaks. Each launch keeps the split of its tier at launch. Tier values are readable on-chain.
Missing recipients	With no platform payee, or no valid referrer, that share goes to the protocol treasury.
Launch fee	Paid at launch to the treasury, at most 1 ETH.

Referral integrity

A referrer equal to the trader, the recipient, the launcher or the creator key is treated as no referrer. In pool trades the hook only sees the router, so the trader is taken to be the transaction's origin.

Creator keys

Creator fees accrue to an address key, or to a social-handle key that is later bound to a wallet through an attested proposal with a 72-hour public notice, during which the owner or the treasury can cancel it. A balance on an unbound handle can be claimed once the handle is bound, with no deadline; there is no sweep.

Verified tiers

A tier can require the creator key to carry an identity attestation, read from the chain's attestation registry at launch and again when the creator key is transferred.

Buy-and-burn reserve

After a launch migrates to its pool, a share of every fee, taken from the protocol's own share, accrues to that launch's buyback reserve: 5% of the fee on testnet, and a tier's protocol share can never be smaller. RtpBuybackEngine turns the reserve into buys in the pool whose whole output is burned in the same transaction. Anyone can call it; it has no owner, no setter and no pause, and holds nothing between calls.

Price and size protection, fixed at deployment	Testnet
Reference price: mean tick over at least this window, from the fee hook's oracle	30 minutes
Band: a buy may push the price at most this far above the lower of the reference and the spot price	100 ticks, about 1%
Size per call: share of the reserve, and share of the pool's quote depth	10%; 0.5%
Rate limit: a token bucket of the pool's depth, refilled over	0.5%; 30 minutes
Cooldown between calls, scaled by how much of the size was filled	600 s
Caller's tip, out of the quote spent (compiled maximum 0.5%)	0.1%

A front-run that moves the price past the band leaves no fill, and the call reverts without spending anything. Buybacks are rate-limited and not guaranteed. The engine is part of v1.0.0-r4 and is deployed with the next testnet instance.

SECURITY MODEL

What is enforced, and by whom

Enforced by contract

- No upgrade path: no proxy, no delegatecall, no self-destruct, no initialiser to front-run.
- The graduator has no owner and no setter, and no code path removes liquidity; the fee hook lets only the graduator initialise pools or add liquidity.
- Supply minted once; RtpToken has no owner, no later mint, no transfer tax and no blacklist.
- Terms snapshotted per launch; no setter reaches an existing launch.
- Pull payments behind a re-entrancy guard; trades never push value to third parties.

Powers, in full (v1.0.0-r4)

Owner	Add launch tiers and enable or disable them; approve quote assets and set their curve values; set the launch fee, at most 1 ETH. These take effect at once and apply to new launches only. Revoke the attestor at once.
Owner, with a delay	Treasury, attestor and verification-gate changes wait 7 days in public. The owner or the current treasury can cancel a treasury or attestor change; the owner can cancel a verification-gate change.
Owner or treasury	Cancel a pending social-handle binding during its 72-hour notice.
No one	Pause trading or claims, move funds credited to others, touch a pool or upgrade a contract.

The deployed testnet instance. The instance on the public testnet comes from an earlier build (source commit 8cae3fc, deployed 2026-10-22). Until the next instance replaces it, its owner can pause new launches and change the treasury at once, an attestor change waits 2 days, unbound social-handle balances can be swept after 60 days, the owner and the treasury are one address, and the buy-and-burn engine and the lens are not deployed.

SECURITY MODEL

Trust assumptions and known limitations

These are the places where the contracts depend on something outside themselves, or where a protection has a stated limit. The threat model treats each of them in detail.

Topic	What holds, and what does not
Opening tax	It raises the cost of buying in the opening seconds. A buyer who pays it can still buy in the launch block.
One party, several roles	On-chain rules cannot tell two addresses of one person apart. The same party can be creator, platform and referrer, and self-trading economics are disclosed rather than prevented.
Referrer check in the pool	The hook sees the router, so pool trades compare the referrer with the transaction's origin.
Verified tiers	They read an external attestation registry. A change in that registry affects new verified launches and creator-key transfers, not existing balances.
Social-handle bindings	They depend on the attestor key and on someone watching the 72-hour notice; the owner or the treasury can cancel a binding in that window and the owner can revoke the attestor.
Deferred migration	If the hand-over fails in the graduating trade, trading stays on the in-core pool until anyone completes it.
Buy-and-burn	Rate-limited and price-banded; it may not run when the market is moving, and it is not a price support.
Sequencer ordering	Outside the contracts' control on every network; the design assumes worst-case ordering.
Deployed instance	The testnet instance is an earlier build with the controls set out in the security model.
Software risk	Smart contracts may contain undiscovered defects. Internal review rounds are under way and an external audit is being arranged; it may not be complete at mainnet deployment.

Deployment, status and roadmap

Deterministic, verifiable deployment

Every chain-specific value lives in configuration rather than code. Addresses are predicted before broadcast and checked afterwards. Every contract is deployed through CREATE3 except the fee hook, which goes through CREATE2 with a salt mined so that its address carries the Uniswap v4 hook permission flags; that address depends on the chain's PoolManager, so it differs between chains. The official deployment registry on rtp.sh lists every address, and the reference SDK refuses unlisted core addresses by default. The runtime codehashes and the full deployment manifest are published in a public deployment repository that holds no source code; the SDK is published when the code is made public. On mainnet the chain's canonical PoolManager must be wired; the deploy pipeline refuses a private copy unless a documented fallback is chosen.

Where it stands

- Testing covers properties, permissions, re-entrancy and heavy invariants (102,400 calls), with mutation testing of every line changed in this version: 152 mutants, 143 killed, the other 9 equivalent.
- 473 passing tests across 107 suites, plus fork suites run separately; every one of the 88 static-analysis findings triaged.
- Security review in progress: internal review rounds are under way and an external audit is being arranged; a bug bounty programme follows the external audit.
- Testnet source verified on the explorer; runtime codehashes recorded in the deployment manifest, to be published in a public deployment repository that holds no source code.

Roadmap

Stages are sequential milestones rather than calendar commitments, so no dates are given.

Testnet	Next	In progress	Next	Being arranged
GIWA Sepolia Source-verified instance with test tokens; GIWA stays on testnet	**Next testnet instance** Deploys v1.0.0-r4: adds the buy-and-burn engine and the read-only lens	**Internal review and verification** Review rounds, then one complete verification run	**Robinhood Chain mainnet** On the chain's canonical Uniswap v4	**External audit** Results published when complete

APPENDIX

Notation and glossary

Notation

Symbol	Meaning
S	Total supply of a launch, minted once
Q	Real quote held by the curve or the pool
vQ	Virtual quote reserve; zero after graduation
R, R_0	Token reserve of the curve, and its starting value
T	Graduation threshold, in real quote net of fees
cs	Supply sold on the curve at graduation
s0, w, t	Starting opening-buy tax, its window, and the seconds since launch

Glossary

Term	Meaning
Bonding curve	A pricing function that sets the token price from the reserves on each trade; here constant-product with a virtual quote reserve.
Graduation	The hand-over of a launch from the curve to its own Uniswap v4 pool, at the final curve price, normally within the same transaction.
Launch tier	A configuration the operator enables: supply, fee and split, opening tax and window, and whether a verified creator is required.
Hook	A Uniswap v4 contract invoked by the pool on swaps and liquidity events; RtpFeeHook charges the pool fee.
Locked liquidity	A pool position held by a contract with no withdrawal function. The liquidity exists and is tradable, but no one can remove it.
Creator key	The key creator fees accrue to: an address, or a social handle later bound to a wallet.
Pull payment	A design in which recipients withdraw amounts credited to them, instead of contracts sending funds during other operations.

APPENDIX

References and revision history

References

Key	Reference
[UNI-V4]	Uniswap Labs, Uniswap v4 Core: the singleton PoolManager and hooks.
[EIP-7825]	Transaction Gas Limit Cap: at most 16,777,216 (2^{24}) gas per transaction.
[ERC-20]	Token Standard.
[EIP-712] [ERC-1271]	Typed structured data signing; signature validation for contracts.
[CREATEX]	CreateX factory, CREATE3 deployment to the same address on every chain.
[BUSL]	Business Source License 1.1.
[ROBINHOOD-CHAIN]	Robinhood Chain, an Ethereum Layer 2; mainnet chain id 4663, testnet 46630.
[GIWA]	GIWA Chain, an Ethereum Layer 2 on the OP Stack; testnet GIWA Sepolia, chain id 91342.
[DOJANG]	The attestation registry read by verified launch tiers on GIWA chain.
[CODE-CL]	RTP Curve Launch contracts, v1.0.0-r4, commit 1f339ee. The source is published with the public release; the source of the deployed testnet build is verified on the explorer.

Revision history

Version	Date	Change
draft 18	2026-11-03	Initial draft, in two documents.
draft 19	2026-11-04	Every statement checked against the contracts and corrected where it differed.
draft 20	2026-11-04	Names: RTP Protocol, RTP Curve Launch, RTP Fixed-Price Launch. Material not yet public is no longer described as published.
1.0-draft.21	2026-11-04	Three documents: the protocol overview and a technical overview of each launch path. Document control, notation, trust assumptions and known limitations, references.
1.0-draft.22	2026-11-04	Security status stated as it stands: internal review rounds under way, external audit being arranged. Deployment plan: GIWA stays on testnet; both launch paths go to Robinhood Chain mainnet after the review rounds and one complete verification run.
1.0-draft.23	2026-11-05	Document identifiers removed; the papers refer to each other by title. Runtime codehashes and the deployment manifest are published in the code repository; rtp.sh lists the addresses.
1.0-draft.24	2026-11-05	References name GIWA and Robinhood Chain. The Fixed-Price Launch targets Robinhood Chain mainnet directly.
1.0-draft.25	2026-11-05	Fixed-Price Launch also targets Base mainnet. Deployment manifests go to a public repository that holds no source code. RTP.fun named for the platform layer; the Curve Launch fee hook stated as CREATE2.
1.0-draft.26	2026-11-05	Fixed-Price Launch: the success fee in this deployment is 1%; the worked examples show it.

Licensing

RtpCore, RtpCoreLens, RtpGraduator, RtpFeeHook and RtpBuybackEngine are licensed under the Business Source License 1.1 (Licensor RTP.fun, change date 2029-10-25, then MIT); RtpToken, the interfaces, the router, the quoter, the scripts and the SDK are MIT. The licences grant no trademark rights.

Verification and contact

Only addresses listed in the official deployment registry on rtp.sh are official, whatever name a front end presents. Security reports: security@rtp.fun, by private disclosure. Licensing enquiries: licensing@rtp.fun.

Important information. This document describes software under development and testnet deployments that carry no real value. Parameters, designs and timelines may change before any mainnet deployment, and a mainnet deployment may come before the external audit is complete. Digital assets are volatile and may lose all value. Smart contracts may contain undiscovered defects despite review. Availability may be restricted in certain jurisdictions. This document is not an offer, a solicitation or advice of any kind.