THE CIRCUS The last launchpad

PROGRAMMABLE AND CONFIGURABLE MARKET INFRASTRUCTURE FOR PERMISSIONLESS CAPITAL FORMATION

TECHNICAL WHITEPAPER

Version 0.1 | November 2026

Built on Elysium. Designed for Hyperliquid.

Written by Sunny K

Abstract

Token issuance is now computationally trivial. Market creation is not.

Contemporary token launch infrastructure largely reduces capital formation to a standardized bonding curve, a predetermined liquidity threshold and a limited set of creator-controlled parameters that puts us all to sleep. The result is a market of 100m doppleganger assets all with substantially identical underlying market mechanisms and similar outcomes. Zero. Crypto, once one of the most feared technologies in the world is now a clownshow. The laughing stock of the worlds financial markets.

Pumpfun made money by giving monkeys the ability to mint tokens. Since then the little ones started deploying their own launchpads, each as unremarkable as the other. The same curve, the same parameters and the same iQ behind the github. Its boring. Now you have 100m tokens with the same outcome and all the NPCs on ct whining about why crypto is broken totally missing the point entirely. 100M tokens isnt a problem. 100M identical replicas is. Without a variable there is nothing unique, its too easy to clock the game. 100M completely unique tokens each with unique market triggers creates 100M unique games with 100M unique outcomes. HE HE HE

Thats what Ive done. I took the training wheels off, I added some insane variables and optimised exclusively for extreme outcomes. You wanted to turn crypto into a clownshow, fine, now you must face the wrath of the clown.

The Circus introduces a different primitive: the programmable primary market.

Pumpfun gave monkeys the ability to mint tokens.
Im giving them the ability to play god and mint the entire market.

Want to deploy a token and see it go from $10 - $10m with your first $500 buy. Now you can. HE HE HE

Built on Elysium, Hyperliquid's emerging Layer 2 environment, Circus enables permissionless deployment of tokens into Uniswap v4 liquidity pools governed by creator-defined pricing bands, with 2 decimal configurations and $1-$1T valuation ceilings. Then I made everything else configurable to multiple the variable outcomes ie trading fees, automated vesting, dynamic anti-sniping mechanisms and deterministic fee distribution.

You can now design custom markets with over 10(120) possible market configurations from just the core controls meaning trillions of different tokens on different journeys with different outcomes.

Its long-term architecture extends beyond decentralized exchange liquidity. Following relevant infrastructure upgrades, Circus intends to connect primary token issuance to HyperCore's order books, oracle systems and execution infrastructure.

Core principle. The creator imagines a market. Smart contracts enforce its rules. Participants determine its price.

01 The Market Structure Problem

Most permissionless token issuance systems share three fundamental limitations.

Standardized market design. Creators receive a narrow selection of supply, liquidity and distribution parameters. The actual price-discovery mechanism is usually determined by the platform.

Misaligned execution economics. Automated sniping, concentrated early access, discretionary liquidity control and opaque fee arrangements can transfer value away from secondary-market participants.

Disconnected market stages. Issuance, AMM liquidity and exchange listing are often unrelated operational processes rather than a continuous rule-governed lifecycle.

Circus addresses these limitations by treating market structure as programmable infrastructure, exposing parameters historically reserved for exchanges, market makers and listing desks.

The fundamental unit is not simply a token. It is a token and its unique associated market rules, deployed together.

02 Protocol Architecture

Circus is organized into five functional layers.

Layer	Responsibility
Issuance	Token creation, supply, allocations and deployment
Market construction	Pricing bands, liquidity provisioning and initial execution
Execution economics	Trading fees, anti-snipe premiums and quote-asset settlement
Distribution	Creator revenue, protocol revenue, vesting and holder rewards
Exchange integration	Future oracle synchronization, HyperCore listing and order-book liquidity

The initial architecture uses Uniswap v4 hooks and components derived from Clanker V4.

2.1 Atomic market deployment

One launch transaction is designed to validate the token and market configuration, deploy and initialize the token, establish the Uniswap v4 liquidity position, assign locked allocations, and execute an optional creator purchase.

Atomic execution avoids partial launch states in which token issuance succeeds but required market initialization fails. The initial dev buy occurs within the deployment transaction, providing deterministic ordering relative to subsequent transactions, without eliminating broader transaction-ordering risk.

03 Programmable Price Discovery

3.1 The seven-band architecture

Circus replaces the conventional single-curve launch interface with up to seven configurable pricing bands. Each band specifies an allocated share of total token supply and a target fully diluted valuation. Creators can inspect the capital required to traverse each band before deployment.

For total supply S, define each band i as:

$$B_i = (w_i, V_i)$$

Here w_i denotes the band's allocated supply fraction, V_i its target fully diluted valuation, and $n \leq 7$ the number of active bands. The allocation constraint is:

$$\Sigma_{i=1}^{n} w_i = 1$$

For an indicative marginal price function P(q), the notional capital needed to traverse band i can be represented as:

$$C_i = \int [q_{i-1}, q_i] P(q)\, dq$$

This integral expresses the economics conceptually; the actual execution function follows the implemented Uniswap v4 liquidity-band and swap mathematics. It does not imply a continuous-curve implementation within every band.

3.2 Market templates

Template	Design intention
Normal	Conventional initial price discovery

Insane	More aggressive repricing
Low Lift	Lower initial capital barriers
Psychotic	Extreme sensitivity to early demand
Custom	Full control of supported band parameters

Templates represent configurations, not return predictions. Different inputs alter the relationship between initial purchase size, executable liquidity and price progression.

3.3 Supply-independent mathematics

The design supports supplies from 1 to 10^{30} tokens. Contract-level mathematics must account for integer precision, decimal normalization, rounding and overflow boundaries across that full range.

04 Dynamic Execution Taxes

4.1 Time-decaying anti-snipe fees

Creators may configure a launch-period buy fee as high as 99%, together with a decay window. One conceptual formulation is:

$$f_launch(t) = f_base + (f_initial - f_base)\, D(t)$$

D(t) is a bounded decay function moving from 1 toward 0. The exact timing, rounding and fee-precedence rules are defined by the deployed contract implementation.

The premium above the applicable normal fee is allocated to the creator rather than Circus. This changes the cost of early automated extraction; it does not guarantee protection against sniping.

4.2 Future reputation-aware execution

Planned HyperCore integration could enable differentiated anti-snipe fees using verifiable account history. Established eligible trading accounts could pay normal fees while new or unestablished wallets face the launch premium, subject to published qualification rules and defenses against reputation farming and wallet linking.

05 A Quote-Native Fee Economy

5.1 Protocol fees

Circus specifies a fixed 1% protocol fee on the defined qualifying trade notional N:

$$F_protocol = 0.01\, N$$

The intended allocation of protocol revenue is split evenly between platform-token rewards and the treasury:

$$F_holders = 0.5\, F_protocol$$

$$F_treasury = 0.5\, F_protocol$$

Recipient	Share of trade notional
Platform-token holder rewards	0.50%
Protocol treasury	0.50%
Total protocol fee	1.00%

Creator-selected fees are separate economic entitlements. The precise precedence and compounding among creator fees, protocol fees and launch-period taxes must be specified at the router and hook level.

5.2 Creator-controlled fees

Creators independently set buy and sell fees from 0% to 9%, in increments of 0.01 percentage points. Parameters permit different launch and ongoing-market incentives but provide no guarantee of favorable outcomes.

5.3 Quote-asset settlement

All protocol and creator fees settle in the quote asset. HYPE pairs accrue HYPE, USDC pairs accrue USDC, and future tokenized-equity pairs are intended to accrue the supported equity quote token. No compulsory fee inventory accumulates in the newly created token.

5.4 Permissionless collection

A creator-facing collection flow chains the required transactions, including wrapped HYPE unwrapping when appropriate. A caller can initiate collection but cannot redirect a creator's accrued funds to another beneficiary.

06 Locked Liquidity and Deterministic Vesting

6.1 Liquidity restrictions

Liquidity is contractually restricted against unauthorized withdrawal or reassignment. Future migration to HyperCore must be an explicitly authorized, narrowly scoped contract transition, not discretionary administrative custody. "Locked forever" and migration can coexist only when that allowed movement is specified and enforced by the contracts.

6.2 Team allocations

Creators may establish locked team allocations with rule-based release conditions. Allocation and vesting data are intended to be visible on the token page and share card. Visibility supplements, but does not replace, enforceable lock and vesting contracts.

6.3 Governance minimization

The initial owner role is controlled by the deployer, with a planned transfer to a 2-of-6 multisignature. Governance and participant entitlements remain conceptually separate. The production threat model must document upgrades, allowlisting, emergency actions and every other privileged capability.

07 Multi-Asset Quote Infrastructure

The launch infrastructure treats quote currency as a configurable supported asset. HYPE and USDC are implemented now, with additional quotes intended to be introduced through controlled allowlisting rather than rebuilding the issuance engine.

7.1 Expansion assets

Under verification: kHYPE, KNTQ and tokenized NVDA, SPY and QQQ via xStocks/EQX infrastructure. Production enablement depends on verifying contract addresses, transfer restrictions, decimals, issuer characteristics and reference feeds.

7.2 Equity-denominated markets

A token quoted in a supported NVDA token expresses its market value in NVDA units. Its indicative dollar-equivalent price is:

$$P^{T}_USD \;=\; P^{T}_NVDA \times P^{N\square DA}_USD$$

Oracle reference prices do not guarantee executable depth or redemption value. Tokenized-equity markets add issuer, custody, tracking, market-hours and oracle risks. The design uses issuer price feeds in the interim and aims to integrate exchange-native oracles when available.

08 HyperCore Convergence

The intended long-term architecture links permissionless primary issuance with Hyperliquid exchange infrastructure. These capabilities remain upgrade-dependent rather than operational today.

8.1 Rule-based graduation

Creators could predefine valuation thresholds or other eligibility rules that trigger a permitted migration from AMM liquidity to a supported HyperCore order book:

$$Graduate(T) \Leftrightarrow V_T \geq V_trigger$$

Production rules must define price authority, observation windows, liquidity requirements and defenses against transient manipulation.

8.2 Exchange-native prices

Future contracts may consume Hyperliquid oracle feeds to reference quote asset prices. Such logic requires data freshness, availability and manipulation checks.

8.3 Working liquidity

After graduation, eligible locked liquidity could support rule-constrained order placement, inventory rebalancing and market making. Such strategies require strict execution permissions, inventory limits, accounting reconciliation and recovery provisions. Locked liquidity is not risk-free liquidity.

Lifecycle. Issuance → price discovery → exchange trading. The current implementation covers issuance and AMM trading; native order-book integration remains planned.

09 The Mad Lab: Pre-Deployment Simulation

The Mad Lab is a creator-facing simulator that estimates how a chosen market configuration behaves under specific hypothetical transactions before launch. Scenarios include purchases of 10, 50, 100 and 1,000 HYPE, price effects, tax charges at minutes 0, 7 and 15, band traversal and net creator proceeds under immediate-sale assumptions.

The simulator should be execution-consistent with the deployed contract logic. It models mechanical outcomes for a defined starting state, not future sentiment, trader flow, transaction ordering or external liquidity.

Operating principle. Before the market exists, its rules should already be observable.

10 Protocol Rewards and Participation

10.1 Holder distributions

Half of protocol fee revenue is intended to accrue to epoch-based rewards pools for platform-token holders, claimed through on-chain proofs. An indicative distribution rule is:

$$r_(u, e) \;=\; R_e \times h_(u, e) \;/\; H_e$$

Here R_e is an epoch's distributable revenue, h_(u,e) the account's eligible weight and H_e the total eligible weight. The final system must define eligibility, snapshots, denominated assets, proofs and claim rules.

10.2 The Big Top

The Big Top measures realized and unrealized trader PnL across defined epochs, supporting rankings and prizes. Robust scoring requires consistent valuation and accounting for transfers, multiple wallets and capital flows.

10.3 Trading points

Trading activity accrues points intended to inform a platform-token distribution. Eligibility, anti-Sybil and anti-wash-trading rules remain essential. Points are not themselves an unconditional claim to a fixed number of tokens.

11 Security Engineering and Verification

The current engineering record reports 383 automated tests, alongside fuzz testing, invariant checks, mutation testing, Slither and Mythril. Security work concentrates on contracts and state transitions that calculate, hold or move value.

Security property	Required outcome
Asset conservation	Balances and fee liabilities reconcile to authorized transfers
Fee integrity	Every fee goes to the specified beneficiary
Vesting integrity	Locked allocations obey release rules
Collection integrity	Callers cannot redirect another beneficiary's proceeds
Market integrity	Swaps respect configured bands, precision and fee limits
Authorization integrity	Privileged operations respect declared access controls

11.1 Wei-level reconciliation

Testnet flows for launch, initial purchase, taxed buy, sell and fee collection are reconciled to contract state at native integer precision. This verifies recorded financial behavior but cannot replace independent adversarial review.

11.2 Audit and release gate

Salus has been identified for external review of the money path prior to mainnet release. The audit must not be represented as complete until the relevant final report and revision scope are available. The gate also includes deployment verification and operational checks.

12 Development and Deployment Status

Two engineering lanes—smart contracts and application—operate in parallel under coordinated oversight, producing tests, screenshots and change records. The team reports three testnet deployments across four days, same-day testnet implementation of a major fee-model revision and stock-quote contract verification at both 8- and 18-decimal precision.

Capability	Status, November 2026
Multi-band configuration	Mainnet
Token launch / dev buy	Mainnet
Dynamic launch tax	Mainnet
Quote-native fee collection	Mainnet
HYPE, USDEE and USDC markets	Implemented / testnet
Team vesting	Current design
Tokenized equity quotes	In verification

Platform-token rewards	Testnet
HyperCore graduation	Upgrade-dependent
HyperCore market making	Upgrade-dependent
V2. Hyper clown	Pending external auditor re-check

Relevant mainnet contracts have been reported as verified and the deployment runbook prepared. Neither statement implies that the entire production protocol is deployed or independently certified.

13 Risk and Trust Assumptions

Smart-contract risk. Hooks, routers, fee logic or vesting failures can lead to losses or inaccessible funds.

Market-design risk. Configurable bands and charges can generate severe slippage, volatility and unfavorable trading conditions.

Quote-asset risk. Markets inherit volatility, restrictions and counterparty characteristics of their quote asset.

Oracle risk. Incorrect, unavailable, stale or manipulated prices can cause unexpected execution.

Governance risk. Privileged capabilities and upgrade paths require explicit boundaries and monitoring.

Migration risk. AMM-to-order-book transitions introduce custody, accounting and inventory complexities.

Manipulation risk. Wash trading, point farming and manufactured PnL can undermine reward integrity.

Regulatory risk. Token rewards and tokenized equity markets may create securities, derivatives and market-access implications.

The protocol's security objective is not to eliminate price risk. It is to reduce hidden discretion, unauthorized fund movement and unverifiable economic behavior.

14 The Protocol Thesis

Crypto has optimized the issuance of assets. It has devoted less effort to making issuance markets individually programmable, transparent and integrated with mature trading systems.

Circus approaches token creation as a market-engineering problem. It gives creators control over distribution, price bands, transaction costs and launch mechanics while designing economic entitlements to be enforced through deterministic smart contracts.

Its proposed HyperCore integration extends that model toward exchange-native trading, creating the possibility of a token lifecycle specified at issuance rather than assembled through discretionary interventions after launch.

The ambition. Continue the grand experiment and make programmable market creation a first-class financial primitive.

15 A Note from Sunny K

They built machines that could issue ten thousand tokens per hour.

Very dull engineering.

Then they gave every single one the same fucking market and added stock pairs… How pedestrian.

Same curves. Same tricks. Same extraction. Same little ones pretending to be market architects because they have a $20 claude sub.

I spent years studying markets as mathematical systems. The order books. The incentives. The invisible machinery deciding who gets paid and who gets carried out.

And then I watched an entire industry replace market structure with sub room temp iq KOLs.

Something inside of my broke. I snapped.

Fine.

You wanted a circus.

I built one.

Only this time the clown understands the matching engine.

And now you must suffer me

— Sunny K | King of fools | Solo Dev & Ringmaster, The Circus

THE LAST LAUNCHPAD.

Technical Whitepaper v0.1 • November 2026

This document describes current architecture and intended roadmap. Future capabilities are distinguished from deployed functionality. It is not an independent audit, proof of production deployment or guarantee of investment performance.