TECHNICAL WHITEPAPER · VERSION 0.1.0-TESTNET

AnyPerp

Any token. A perp. Today.

Permissionless Isolated Perpetual Markets
on Robinhood Chain

Product, Protocol, Risk & System Architecture

Date: 2026-07-22
Network: Robinhood Chain testnet (chain ID 46630)
Site: https://anyperp.fun
Source: https://github.com/AnyPerp/anyperp
Status: Unaudited testnet prototype — not for real funds

Contents

1. Abstract

AnyPerp is a factory for isolated, oracle-priced perpetual futures markets deployed on Robinhood Chain, an Ethereum-compatible Arbitrum-style L2. Any account may create a market candidate for an existing ERC-20 by posting a creator bond, selecting a registered oracle route, and choosing a conservative risk tier. Trading activates only after mechanical admission checks pass: supported collateral, risk envelope, seed liquidity, insurance capital, and multi-source oracle validity.

The protocol deliberately rejects shared LP vaults, coin margin, creator-controlled prices, and unrestricted custom oracles. Each active market owns its own collateral vault, liquidity vault, and insurance fund. Trader PnL, bad debt, and liquidation shortfalls are confined to that market's balance sheet. Execution prices are derived from a validated index plus a bounded skew-impact curve; funding is a zero-sum, checkpointed transfer between longs and shorts; liquidations are permissionless and partial-by-default.

This whitepaper describes the product thesis, economic model, oracle and risk controls, smart-contract modules, and off-chain services that compose the current testnet prototype (v0.1.0). The software is open source under the MIT License, unaudited, and explicitly unsuitable for real capital until independent security, economic, legal, and operational gates are cleared.

Core safety proposition
Permissionless creation is not permissionless liability. Anyone may deploy a candidate; the contracts activate trading only when oracle quality, capital buffers, and risk parameters meet mechanical thresholds.

2. Introduction & Problem

2.1 The listing gap

Centralized exchanges and mature perpetual venues list markets selectively because every listing consumes oracle capacity, insurance capital, monitoring, liquidation depth, and compliance attention. On a public chain, a new spot token can appear within minutes while a defensible derivative index, counterparty capital, and loss waterfall do not. The delay is partly gatekeeping and partly risk production.

Traders who receive or buy a long-tail token often cannot hedge without OTC credit or directional-only spot exposure. Token projects lack a neutral path to prove oracle quality and fund isolated liquidity. Liquidity providers are asked to underwrite unknown assets through shared pools that mutualize tail risk across unrelated markets.

2.2 Why long-tail perps are different

- Spot depth can vanish within blocks; a single pool may define the price.
- Supply may be concentrated; symbols can impersonate other assets.
- Token code may rebase, tax transfers, blacklist, pause, or callback.
- Price gaps can exceed any realistic maintenance margin.
- Architectures that assume deep external discovery and broad insurance systematically underprice these failures.

2.3 Network context

Robinhood Chain is documented as an Ethereum-compatible L2 with standard Solidity tooling, JSON-RPC/WebSocket endpoints, ETH gas, and ecosystem integrations including Chainlink feeds and an L2 sequencer-uptime check. Sequencer soft confirmation, first-come-first-served ordering (no priority-gas auction), and Arbitrum-specific block semantics require confirmation-aware indexing and no assumption that higher gas wins a liquidation race. AnyPerp targets testnet chain ID 46630 for the prototype.

3. Design Thesis

AnyPerp spends capital efficiency to buy explicit solvency boundaries. The design answers a single product question: how can market creation be open while insolvency remains named, local, and visible?

3.1 Decisions adopted

- Oracle-priced peer-to-isolated-pool execution (deterministic fills).
- One market instance + three vaults per base token / oracle route.
- USD-style collateral only in the MVP; no coin margin.
- Registered multi-source routes; no human price injection.
- Risk tiers that creators may tighten but never loosen.
- Skew impact, OI/skew/utilization caps, partial liquidation.
- Timelocked governance; guardian may only reduce risk (pause / reduce-only).

3.2 Decisions rejected

- Shared LP vault across markets (contagion).
- Pure constant-product vAMM as primary engine (repeg + unfunded PnL).
- CLOB-first launch (maker scarcity defeats immediate listing).
- Creator-controlled or authority-push oracles (unilateral PnL transfer).
- Cross-margin portfolios and stock/RWA session products in MVP.
- Protocol token requirement for security or fee capture.

4. System Overview

AnyPerp is composed of three layers that remain separable: (1) immutable per-market contracts and shared protocol modules on Robinhood Chain; (2) reorg-aware off-chain indexing, API, and keeper workers; (3) a wallet-connected web surface for discovery, creation, trading, and LP.

4.1 Roles

Role	Objective	Primary risk
Trader	Long/short with isolated margin	Oracle gap, insolvent LP
Creator	Launch a named market, earn fee share	Bond lock, rejected candidate
LP	Underwrite one market for fees + PnL	Trader profits, blocked withdrawals
Liquidator / Keeper	Restore health; earn penalties/fees	Invalid oracle, FCFS races
Governance	Timelocked parameter & adapter changes	Capture, bad parameters
Guardian	Emergency reduce-only / pause	Mistaken or delayed action

4.2 What the protocol does not claim

- That every token becomes safely leverage-tradable after creation.
- That mock oracles or faucet collateral imply mainnet readiness.
- That tests, smoke scripts, or compilation substitute for audit.
- Affiliation with Robinhood or endorsement of any listed asset.

5. Market Lifecycle

Markets progress through a strict state machine. Token name and symbol are never admission inputs — only addresses, routes, capital, and parameters.

State	Allowed	Forbidden
PendingValidation	Oracle/risk checks	Trading, vault withdrawals
Bootstrapping	LP & insurance seed	Trading
Active	Trade, LP queue, liquidate	Settlement claims
ReduceOnly	Reduce, deposit, liquidate	Increasing exposure
Paused	Governance incident work	Trade, funding, most exits
Settling	Claim at fixed price	Trading / funding
Closed / Rejected	Archival reads	Market activity

Activation path

- createMarket — pull bond; CREATE2 Market + CollateralVault + LiquidityVault + Insurance.
- validateMarket — RiskManager envelope + OracleRouter multi-source checks.
- seed — LP and insurance deposits (exact balance increase).
- activateMarket — same-transaction revalidation of capital and oracle; enter Active.

Candidates that cannot validate within the configured timeout enter Rejected. Bond return / slash policy is governance-configurable and must be fixed before mainnet.

6. Trading & Pricing Model

All contract accounting uses 18-decimal WAD normalization. Collateral transfers remain in native decimals. Liabilities round up; user credits round down — the protocol never rounds in the user's favor on obligations.

6.1 Price hierarchy

- Oracle price Ps — one adapter observation (price, time, confidence, depth, history).
- Index I — conservative aggregate (median / midpoint) after route validity checks.
- Execution E — index adjusted by integral average of the skew-impact curve.
- Mark M — risk price for margin/PnL; prototype uses validated I; full mark band is pre-mainnet work.

6.2 Core identities

$$\text{notional } N = |q| \times P$$
$$\text{unrealized PnL } U = q \times (M - E_{entry})$$

```
equity              = C + U − accruedFunding − accruedFees
initial margin      = ceil(N × IMR)
maintenance margin  = ceil(N × MMR)
trading fee         = ceil(|Δq| × E × feeBps / 10,000)
funding payment     = q × (F_cumulative − F_checkpoint)
```

6.3 Skew execution

Let base skew S = OI_long − OI_short, trade Δq, skew scale K, and max impact a:

```
averageSkew = S + Δq / 2
impactBps   = clamp((averageSkew / K) × a, −a, +a)
E           = I × (10,000 + impactBps) / 10,000
```

A long into positive skew pays above index; a short into positive skew receives a better sale price and reduces imbalance. An independent acceptable-price (slippage) guard remains on every trade. The curve is deterministic and cannot be altered by a keeper.

6.4 Worked example (illustrative)

Deposit $2,000. Buy q = 50 at execution $100 (notional $5,000). At 10 bps, entry fee is $5, leaving $1,995 margin. Close all at $112 with close fee $5.60. Gross PnL is 50 × (112 − 100) = $600; ending margin is approximately $2,589.40 before funding. The isolated LP loses $600 and receives fees before the configured split. The reverse move to $88 yields a $600 loss.

7. Oracle Architecture

Price integrity is the protocol's primary risk surface. Routes are composed only from governance-registered adapters. The router never accepts a transaction-specific human price.

7.1 Source topology

- Primary: verified Chainlink AggregatorV3 (or Data Streams) feed with correct denomination.
- Secondary: independent registered feed when available.
- DEX TWAP: counterfactual window from a verified pool / IVerifiedTwapSource boundary.
- Aggregation: median for 3+ sources; two-source midpoint only if deviation passes the tier.
- Sequencer gate: reject while down and through recovery grace.

7.2 Validity (conceptual)

```
valid_i = enabled AND price>0 AND fresh AND confidenceOK AND depthOK AND historyOK
P = median(valid prices)
routeValid = |sources| >= minSources
        AND range(P) <= maxDeviation
        AND sequencerUp AND recoveryGraceElapsed
```

7.3 Manipulation resistance principles

- Never use same-block AMM spot or freshly initialized cumulative ticks for admission.

- Cap OI relative to conservative spot depth and LP assets.
- Treat correlated pools as one economic source, not independent sources.
- Revalidate on every increase, activation, liquidation, and settlement.
- Experimental tiers force strict caps so extractable value cannot exceed manipulation cost under simulation.

Stock / RWA note
Robinhood stock-token feeds follow market hours and corporate actions. AnyPerp MVP excludes session-aware equity perps; those products need separate funding, settlement, and legal design.

8. Risk Engine & Tiers

Fully open creation cannot mean unrestricted liability. Each market stores a risk envelope. Creators may choose tighter parameters; they cannot choose looser ones than the tier allows. Values below are initial proposals for simulation — not approved mainnet production parameters.

Parameter	Blue-chip	Established	Emerging	Experimental
Max leverage	10×	5×	2.5×	1.5×
Initial margin	10%	20%	40%	66.67%
Maintenance	6%	12%	25%	50%
Max OI / LP	100%	65%	35%	15%
Max skew / LP	35%	25%	15%	8%
Max utilization	80%	70%	55%	35%
Max impact	1%	2.5%	5%	10%
Min sources	3	2	2	1 + strict caps
Withdraw delay	24h	48h	72h	7d

Automatic risk responses

- Reduce-only: OI near cap, utilization max, source count degraded, collateral warning, thin insurance.
- Pause: no safe price, sequencer down, hard deviation breach, token behavior change, invariant failure.
- Settlement: persistent oracle failure, abandoned spot, terminal exploit, governance delisting after notice.

9. Liquidation, Funding & Loss Waterfall

9.1 Liquidation

A position is liquidatable when conservative equity falls below maintenance. Longs use a lower price bound; shorts use an upper bound; collateral uses a lower collateral bound. The engine prefers partial close restoring a target margin band; full close applies on negative equity, dust remainder, or gap conditions that make partial liquidation unsafe.

- Permissionless liquidate(account, maxClose) via LiquidationEngine.
- Penalty capped by remaining margin and closed notional × penalty bps.
- On Robinhood FCFS sequencing, redundant keepers matter more than priority fees.

9.2 Loss waterfall (per market only)

1) Trader remaining margin
2) Market insurance fund
3) Capped protocol backstop (explicit grant; no automatic mutualization)
4) ADL against profitable high-leverage accounts (last resort)
5) Disclosed socialized loss within the same market only

9.3 Funding

Funding is event-driven and checkpointed. The UI may quote hourly rates while the contract accrues per second. Positive rates mean longs pay shorts. Premium and skew components are clamped; velocity limits prevent single-observation shocks. Aggregate payer and receiver legs are zero-sum within directional rounding. Funding does not advance on invalid oracle or during sequencer outage; catch-up is capped after recovery.

10. Liquidity & Creator Economics

Each market has one collateral-denominated LP vault. Shares are dilution-resistant (virtual share / asset defense against first-depositor inflation). Deposits are immediate when accepted. Withdrawals are two-step: request escrows shares; after the tier delay, execution requires free assets above reserved liabilities.

$$\Delta \text{LPAssets} \approx \text{traderRealizedLosses} - \text{traderRealizedProfits} + \text{LP fee share} - \text{uncovered bad debt} - \text{withdrawal frictions}$$

10.1 Fee proposal (testnet defaults)

Trading fee default: 10 bps of execution notional (ceil). Illustrative split used in implementation discussions: majority to LP vault, remainder to market insurance, protocol treasury, and market creator. Creation protocol fee is zero; creators pay gas and lock a bond. Funding is not protocol revenue.

10.2 Creator bond

Creators post a bond and typically seed LP + insurance. Rewards should vest and be clawable under predetermined misconduct rules. Dead markets enter reduce-only then settlement after inactivity windows; bonds may cover cleanup costs. No market is silently deleted from the registry.

11. Smart-Contract Topology

Markets are non-upgradeable instances. New logic ships as a new factory version. Shared configuration (adapters, risk tiers, fees) mutates only through GovernanceTimelock. EmergencyGuardian can restrict but cannot unpause, settle, transfer funds, or set prices.

Module	Responsibility
MarketFactory	CREATE2 deploy; bond; validate; seed; activate
MarketRegistry	Immutable ID / address history
Market	Positions, OI/skew, PnL, funding, state machine

Module	Responsibility
CollateralVault	Isolated trader margin custody
LiquidityVault	LP shares, reserves, withdrawal queue
MarketInsuranceFund	Market-only bad-debt cover
OracleRouter + adapters	Registered multi-source validation
RiskManager	Tier envelopes; tighten-only candidate params
FundingEngine / FeeManager	Deterministic rates and fee splits
LiquidationEngine	Permissionless entrypoint; no custody
TriggerOrderManager	On-chain stops/limits with ETH fee escrow
ProtocolBackstop	Explicit capped grants; no silent mutualization
GovernanceTimelock	Delayed admin actions
EmergencyGuardian	Pause / reduce-only only

Trade path (summary)

- Approve + depositMargin → CollateralVault exact pull.
- executeTrade(delta, acceptablePrice, deadline) → oracle validate → funding settle → impact pricing → fee → PnL.
- Trader profit debits LiquidityVault; trader loss credits it; never debits another market.

12. Off-Chain Stack

On-chain state is authoritative. Off-chain services improve UX and keeper availability; they must not become a second source of truth without reorg safety.

- Indexer — raw log ingestion with canonical / orphaned events; rewind on reorg.
- PostgreSQL — migrations for events, projections, 78-digit fixed-point numerics.
- API (Fastify) — read/prepare endpoints, WebSocket streams, ops health.
- Keepers (BullMQ / Redis) — funding, liquidations, triggers, LP withdrawals, optional oracle push on testnet mocks.
- Frontend — multi-surface site (landing / docs / app) with wallet connect, create flow, trade UI.
- Ops — Docker Compose local stack, Prometheus skeleton, incident runbooks.

Hosted API outage must not brick contracts: users can still call RPC-facing functions directly. Projections lag is disclosed; finality labels distinguish soft confirmation from L1-posted and finalized.

13. Competitive Positioning

Model	AnyPerp stance
GMX-style oracle pool	Similar execution; reject cross-market shared LP
Hyperliquid HIP-3	Builder markets inspirational; AnyPerp uses smaller bonds + registered adapters, not validator-supervised oracles
Perpolator-style any-token	Isolation useful; reject coin margin & authority prices
vAMM (Perp v1 style)	Rejected as primary: virtual depth does not fund PnL

Model	AnyPerp stance
CLOB / RFQ first	Deferred: poor long-tail maker bootstrap
Shared insurance mega-pool	Rejected: permissionless long-tail contagion

Positioning summary: AnyPerp is not "the most capital efficient perp." It is a factory for named, isolated risk containers with mechanical admission control — optimized for early, transparent derivative discovery on Robinhood Chain without pretending every token is BTC.

14. Security Posture & Limitations

14.1 Known threat classes

- Thin-pool oracle manipulation and correlated "independent" sources.
- Jump risk beyond liquidation buffers; keeper / sequencer interruption.
- LP bank runs and utilization locks; stablecoin depeg / freeze.
- Malicious token behavior (rebase, tax, blacklist, callbacks).
- Implementation error in accounting, rounding, or access control.
- Governance key concentration; compromised guardian.
- Regulatory classification of perpetual derivatives and front-end access.

14.2 Prototype limitations (honest)

- Unaudited; no public bug bounty completed.
- Settlement observation design and full DEX TWAP binding still evolving.
- ADL / socialized-loss paths documented more completely than implemented.
- Indexer projections and historical scanners are partial.
- Testnet may use mock oracles and mintable collateral — flags that must be off for mainnet.
- Economic parameters are proposals pending calibration against real depth data.

Safety boundary
Local engineering demo: go (mocks only). Public testnet: conditional on verified routes and fresh roles. Mainnet / real funds: no-go until audits, legal review, economic proofs, and operational drills pass.

15. Roadmap & Go / No-Go Gates

15.1 Near-term engineering

- Verified live oracle routes for launch assets; remove mock dependence.
- Complete projections, scanners, and operator UI coverage.
- Foundry invariant / fuzz suites and static analysis in CI.
- Economic calibration: manipulation cost vs extractable LP/insurance.

15.2 Mainnet gates

- At least two independent smart-contract audits + retest of critical findings.

- Economic / risk review of tier parameters and loss waterfall.
- Legal and jurisdictional policy for hosted front-ends.
- Multisig + timelock ownership transfer; key ceremony documentation.
- Independent keeper and RPC redundancy; incident drills (RPO/RTO).
- Public vulnerability disclosure program.

15.3 Explicit no-go conditions

- Only one manipulable pool defines price and EV(manipulation) > buffers.
- Any path for one market to debit another market's vault.
- Settlement depends on creator discretion or unpublished admin price.
- Operators cannot detect or respond to source failure.

16. Conclusion

AnyPerp proposes a disciplined answer to permissionless perpetual creation: open the factory, close the balance sheet. By isolating capital, requiring registered multi-source oracles, enforcing tiered risk envelopes, and making loss waterfalls market-local, the protocol aims to convert listing demand into transparent, bounded risk containers rather than a single shared catastrophe.

The testnet prototype demonstrates the engineering surface — contracts, simulations, indexing, keepers, and UI — needed to evaluate that thesis in public. It does not yet demonstrate safe mainnet parameters or universal token eligibility. Treat the code as research infrastructure; treat real funds as out of scope until the gates in Section 15 clear.

Open source repository: https://github.com/AnyPerp/anyperp
Product surface: https://anyperp.fun
Tagline: Any token. A perp. Today.

Appendix A — Glossary & Parameters

Term	Meaning
Index price I	Oracle-validated reference, independent of perp OI
Execution price E	Index ± bounded skew impact for a fill
Mark price M	Conservative price for margin and liquidation
IMR / MMR	Initial / maintenance margin rates
Skew	OI_long − OI_short in base units
Isolated margin	Collateral pledged to one account on one market
LP vault	Market-specific counterparty capital pool
Insurance fund	Market-specific residual loss buffer
Backstop	Optional capped protocol grant; not automatic mutualization
ADL	Auto-deleveraging of profitable counterparties last
CREATE2	Deterministic deployment addressing for market instances
Timelock	Delayed governance execution for parameter changes
Guardian	Emergency actor limited to risk-reducing actions
WAD	1e18 fixed-point unit used in accounting

Illustrative testnet fee defaults

- Trading fee: 10 bps of notional (ceil).
- Experimental preset bond / seed magnitudes are configuration, not promises.
- Liquidation penalty order of magnitude: up to a few percent of closed notional, margin-capped.

Appendix B — Testnet Deployment Pointers

Public address manifests for Robinhood Chain testnet (46630) are published in the open-source repository under deployments/ (for example 46630-latest.json). Core modules include OracleRouter, RiskManager, FundingEngine, FeeManager, LiquidationEngine, TriggerOrderManager, MarketRegistry, MarketFactory, and related vault deployers. Always verify bytecode and roles against the chain before integrating.

Representative suite roles from a published testnet manifest include governance timelock, emergency council, and treasury addresses. These are public on-chain identifiers, not secrets. Private keys, database URLs, and host credentials are never part of the protocol specification and must not appear in distribution copies of this whitepaper.

References (selected)

- Robinhood Chain documentation — network, finality, oracles, EVM differences.
- Chainlink AggregatorV3 and L2 sequencer uptime patterns.
- GMX — oracle-priced execution, funding, liquidations, pool risk.
- Drift — partial liquidation and constrained PnL settlement lessons.
- Hyperliquid HIP-3 — builder-deployed perps with isolation requirements.
- AnyPerp open-source tree — contracts, simulations, services, scripts.

End of document. © 2026 AnyPerp contributors. Released for technical communication alongside the MIT-licensed software. Parameters and implementations may change without notice as the testnet evolves.