

Building the Future of AI & Self-Custody

No intermediaries. No compromises.

We create open-source, AI-powered onchain solutions that empower users with true ownership of their digital assets.

STUDIO

Lunatic Wisdom
Labs

LIVE

HyperTrade
OrbCast
BuilderPad

CUSTODY

User-held keys

TOKEN

LWL only

SOURCE

Open

- Labs thesis: ship multiple products that keep users in control of keys and signatures, publish the work as open source, and use one token — LWL — wherever product revenue supports the token.

Contents

1. Vision and Goals	3
2. Products	3
3. Why Now: Global Stablecoins and Onchain Markets	4
4. HyperTrade: A Trading Interface for Every Level	5
5. HyperTrade Architecture Overview	6
6. Wallet, Custody, and Gasless Infrastructure	7
6.1 UR.APP Integration: External Wallet Access and IBAN Accounts	7
7. Markets and Execution	10
8. Revenue Model and Sustainable Growth	11
9. AI Agents and Market Intelligence	12
10. Community Rewards and Roadmap	14
11. LWL Tokenomics	14
12. Risk, Security, and Trust Model	15
13. Regulatory and Policy Landscape	16
14. Appendix: Factual Anchors, Audits, and Sources	17

1. Vision and Goals

Lunatic Wisdom Labs LLC (lunaticwisdomlabs.com) is a Wyoming studio that publishes open-source, onchain products. Users hold their own keys. The labs does not take custody of wallet funds in order to operate an interface. AI is part of the studio brief — used where a product actually needs it — not a slogan pasted onto every app.

The audience is global: people who want dollar-denominated balances and clear market access on a phone, and people who already know how signing, gas, and order books work. LWL ships more than one product so those needs do not have to live in a single binary or a single market type.

- **Self-custody:** embedded wallets, user-signed actions, no company balance sheet of user deposits.
- **Open source:** each live product is published so it can be inspected, forked, and improved.
- **AI where it fits:** HyperTrade includes market intelligence and optional trading agents.
- **One token:** LWL is the only token. Products do not get their own tickers (see §11).

- **The studio goal is not to put every market in one app. It is to keep shipping interfaces that stay non-custodial, remain open source, and feed a single token policy when they earn revenue.**

2. Products

Three products are live. Products share a custody posture and the LWL token; they do not share one app, one order path, or one compliance stack.

Product	What it is	Where to look
HyperTrade	Self-custodial mobile interface for onchain trading and optional stablecoin finance: perpetuals (including non-crypto HIP-3 books), optional UR.APP IBAN/card rails, and optional AI trading agents.	hypertrade.exchange · LWL-HyperTrade/hypertrade
OrbCast	Self-custodial interface for HIP-4 outcome / prediction markets (sports and event books). Showcase is sports. No perpetual futures, HIP-3 books, banking, or AI worker in v1.	orbcast.xyz · LWL-OrbCast/orbcast
BuilderPad	Self-custodial web console for branded Hyperliquid trading apps, with an optional creator coin. Shares HyperTrade infra (Privy, Bridge2, builder fees). Not a second mobile binary, and not HIP-4.	builderpad.xyz · LWL-HyperTrade/hypertrade

Technical chapters from §4 onward describe **HyperTrade** in depth (architecture, wallet, UR.APP, agents). OrbCast and BuilderPad are in scope as sibling products and as revenue sources for LWL — not as second architecture dumps.

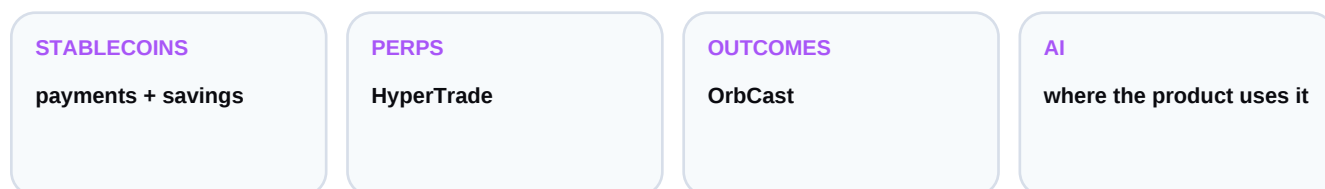
- **Shared rails today include Privy embedded wallets, user-signed gasless USDC where a product uses them, and transparent interface fees. What a given product lists, signs, or verifies is product-specific.**

3. Why Now: Global Stablecoins and Onchain Markets

Stablecoins have become one of crypto's clearest product-market-fit categories: dollar-like balances, global transferability, and fast settlement. At the same time, onchain market infrastructure is moving beyond crypto-only speculation into equities, commodities, FX, index exposure, and event / outcome books.

LWL sits at that convergence with more than one product. A user who wants always-on macro and perps uses HyperTrade. A user who wants event and sports outcomes uses OrbCast. A creator who wants a branded Hyperliquid app (and optionally a coin) uses BuilderPad. A HyperTrade user who starts with a USDC trading balance can later use portfolio tools, AI analysis, or — if they opt in — card and IBAN rails. Those paths are not copied onto OrbCast or BuilderPad.

This matters globally. In many markets, access to stable dollar balances, always-on markets, and event markets is more useful than another zero-utility token. The labs thesis is that useful financial interfaces will outperform purely narrative-driven crypto applications.



4. HyperTrade: A Trading Interface for Every Level

HyperTrade is an interface, not a traditional broker. The app helps users prepare, sign, and submit transactions while keeping funds in user-controlled wallets and routing execution to external onchain infrastructure.

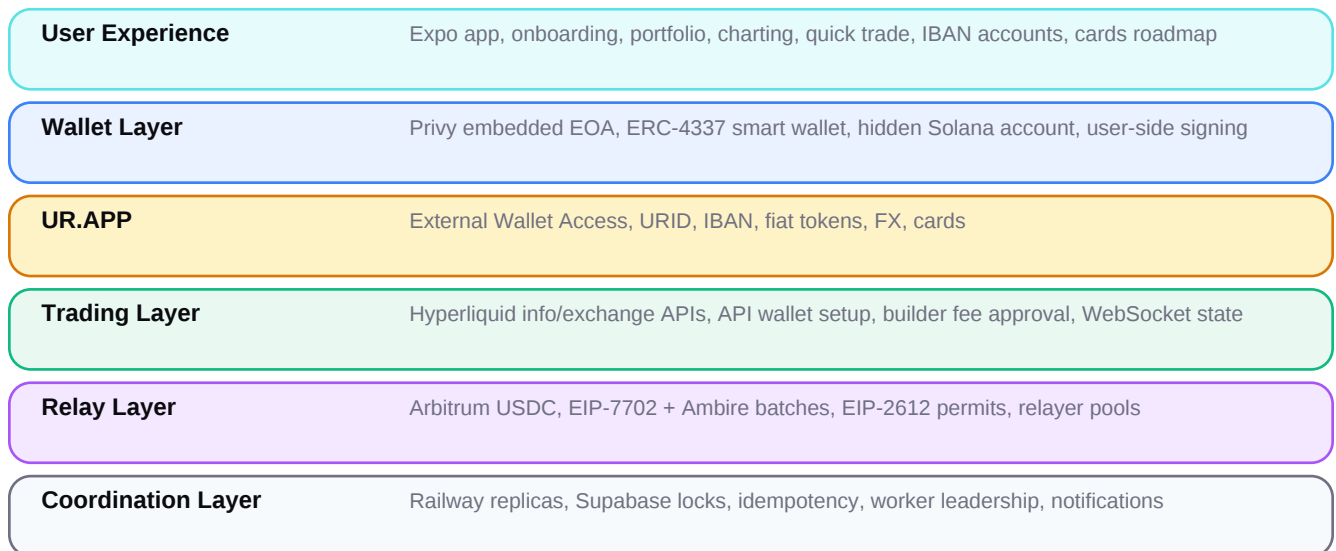
User Need	HyperTrade Response
Easy onboarding	Privy email, Google, and Apple login with automatic embedded wallet creation.
Simple balances	Wallet balance, trading balance, and portfolio screens with clear transfer states.
Fast trading	Hyperliquid API-wallet setup enables one-tap order signing after user approval.
Risk awareness	Margin previews, leverage controls, liquidation estimates, reduce-only guards, and education-first copy.
Future everyday finance	IBAN accounts, FX, and debit cards via UR.APP (External Wallet Access mode).

Experience principles

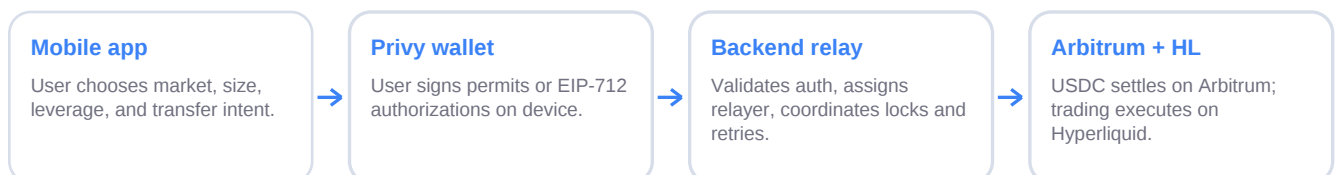
- **Start simple, reveal depth:** new users can place a straightforward trade, while advanced users still get leverage, margin mode, order editing, and portfolio controls.
- **Make balances legible:** wallet funds, trading funds, and future card balances should never be ambiguous.
- **Respect user control:** every sensitive action remains user-authorized, even when the app hides operational complexity.

5. HyperTrade Architecture Overview

HyperTrade's architecture is intentionally modular. User identity and wallet signing live in the mobile app. Backend services coordinate relays, market data, notifications, and operational state. Hyperliquid is the active execution venue for this product. Arbitrum USDC is the stablecoin settlement rail. IBAN-backed fiat accounts and card rails are delivered through **UR.APP** (docs.ur.app) via External Wallet Access — user-signed flows with regulated fiat-token rules on Mantle (see §6.1). HyperTrade remains a non-custodial interface for wallet keys and trading; it does not hold user funds.

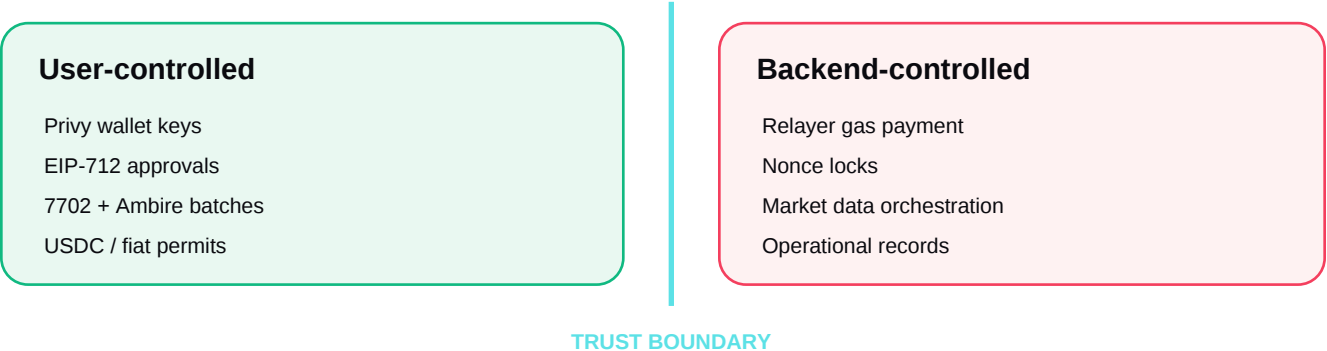


System flow: from user intent to execution



6. Wallet, Custody, and Gasless Infrastructure

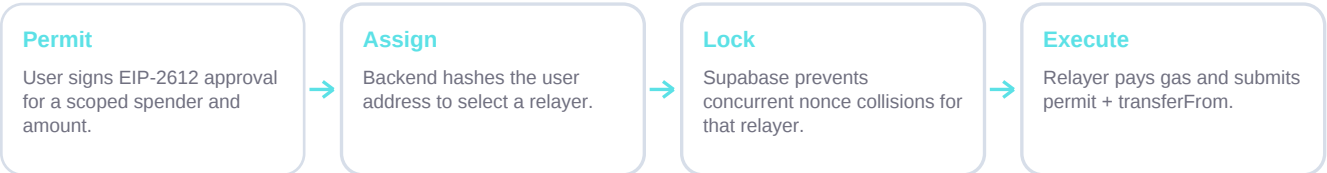
HyperTrade uses Privy to make wallet ownership accessible. Users can authenticate through email, Google, or Apple. The app creates an embedded Ethereum wallet for users without one, wires a Privy smart wallet for ERC-4337 account abstraction, and defensively creates a Solana wallet that remains hidden from the UI today. Arbitrum is the active EVM chain.



Gasless relayer design

Users sign EIP-2612 USDC permits on device. The backend cannot move funds by itself; it can only submit the exact action authorized by the user. A deterministic relayer assignment maps each wallet to one relayer, and Supabase locks serialize nonce-sensitive execution across Railway replicas.

Gasless USDC permit relay



6.1 UR.APP Integration: External Wallet Access and IBAN Accounts

UR.APP is the regulated banking infrastructure that powers HyperTrade's IBAN accounts, fiat-token balances, FX, pay-in/payout, and debit card rails. HyperTrade is a partner interface on top of UR.APP's APIs and onchain contracts — not a custodian. Developer documentation lives at [docs.ur.app](#); contract source and audit reports are in [ur-app/ur-contracts](#) (UR no longer publishes a public smart-contracts docs page).

How HyperTrade integrates with UR.APP

UR.APP integrations are built from three independent choices: **Account Mode** (who holds fiat and who signs), **Card Mode** (what funds card spend), and **KYC Mode** (how identity is verified). HyperTrade enables UR.APP by default; the same product can also run as trading-only — Privy, Arbitrum USDC, and Hyperliquid — without UR.APP credentials, Mantle banking contracts, or KYC (see §6.1 custody boundary).

UR.APP integration choice	What UR.APP offers	What HyperTrade uses
---------------------------	--------------------	----------------------

Account Mode	Managed Custody (partner signs fiat) · External Wallet Access (user wallet signs)	External Wallet Access — user signs; UR rules apply to fiat tokens
Card Mode	Fiat Only (spend fiat balance) · Crypto Backed (real-time auth against crypto)	Fiat Only — fiat-backed card (roadmap)
KYC Mode	UR.APP webview · Sumsb SDK · Sumsb reuse	Sumsb SDK in the HyperTrade mobile app

External Wallet Access (HyperTrade's Account Mode): the user's Privy embedded EOA owns the onchain **URID** (ERC-721 identity NFT) and receives **fiat-token balances** (e.g. USD24, EUR24) at the user's wallet address on Mantle. Banking actions require the user to sign — via EIP-712, EIP-2612 permits, or EIP-7702 batched execution — while HyperTrade sponsors gas where possible. This is not the same as holding raw USDC: once USDC is deposited into UR.APP contracts, credited balances are UR.APP-regulated fiat tokens subject to UR.APP's terms and compliance controls (see custody boundary below).

Managed Custody (not used by HyperTrade): in UR.APP's alternative Account Mode, fiat lives in a UR.APP-managed account and the partner backend signs fiat actions on the user's behalf. This suits consumer neobanks that hide wallet mechanics. HyperTrade deliberately chose External Wallet Access so users sign each banking action and URID ownership stays visible onchain at the user's EOA.

Fiat Only cards (HyperTrade's Card Mode): when card spend is enabled, the debit card draws from the user's UR.APP fiat balance (tokenized deposits). Users who want to spend crypto first off-ramp into fiat, then tap the card. **Crypto Backed** — UR.APP's alternative Card Mode where swipes settle against crypto via a partner prefund pool and real-time authorization webhooks — is not part of HyperTrade's design.

Design principle: UR.APP supplies regulated IBAN rails and onchain fiat tokens; HyperTrade supplies

- **UX, Privy wallet connectivity, and gasless relay infrastructure — without becoming a custodian of wallet keys or trading balances.**

Custody boundary: wallet USDC vs UR fiat tokens

HyperTrade's non-custodial posture applies to the core trading stack: Privy wallet keys stay on the user's device, Arbitrum USDC in the wallet before any deposit is user-controlled, and Hyperliquid trading balances move only under user-authorized API-wallet signatures. HyperTrade does not take custody of those assets.

When a user **opts in** to UR.APP banking and runs Add Money, USDC on Arbitrum is approved and deposited into UR.APP's onchain contracts (e.g. Fiat24CryptoDeposit). Settlement credits regulated fiat tokens such as **USD24** or **EUR24** on Mantle. In External Wallet Access mode those tokens appear at the user's EOA address, but they are **UR.APP-issued fiat tokens** moving through UR.APP's contract system — not generic self-custodied USDC. IBAN pay-in, FX, payout, and card spend are governed by UR.APP's smart-contract permissions, API policies, and licensed compliance program.

That regulated layer is expected to carry restrictions that do not apply to raw wallet USDC: KYC-gated account status, sanctions screening, AML monitoring, and — where law or UR.APP's compliance program requires — **freezes, holds, or blocks** on token movement. Users can typically off-ramp back to USDC in their own wallet via signed withdraw flows, but while balances remain in USD24/EUR24 form, **UR.APP's terms of service and regulatory obligations apply**; HyperTrade cannot override them.

- **UR.APP IBAN, fiat-token, and card rails are optional. HyperTrade can ship as Privy + Arbitrum USDC + Hyperliquid trading only — without UR.APP API credentials, Mantle banking contracts, or KYC flows.**

Onchain identity and IBAN account model

- **URID (Fiat24Account):** each user receives an ERC-721 identity NFT minted to their Privy EOA on Mantle, gating access to IBAN accounts and card products. KYC status is recorded onchain in the contract's *status* mapping (e.g. Tourist → Live).
- **Fiat tokens at the EOA:** USD24, EUR24, CHF24, and related balances are credited to the user's Mantle address in External Wallet Access mode, but they are UR.APP-regulated ERC-20 fiat tokens — subject to UR.APP contract rules, URID status, and compliance controls (including freezes or holds where required).
- **IBAN provisioning:** after KYC reaches *Live*, UR.APP provisions SEPA/SWIFT IBAN details via the Profile API; inbound wires credit the user's fiat-token balance once settlement completes.

KYC and identity verification (Sumsub)

Identity verification for IBAN accounts is delivered through **Sumsub**, integrated via UR.APP's compliance stack using the **Sumsub SDK in your app** KYC mode. Users complete document capture, liveness checks, and identity verification inside the Sumsub mobile SDK surfaced from HyperTrade.

HyperTrade does **not** store user identity documents, biometric captures, or other sensitive KYC payloads on its own servers. The app embeds the Sumsub SDK for the capture UX and exchanges short-lived access tokens through UR.APP API endpoints (e.g. account-status and create-access-token flows). HyperTrade backends receive only the *outcome* needed to gate product features — verification status, chain eligibility, and compliance state — not copies of passports, selfies, or raw document files.

- **SDK on device:** document and liveness capture runs in Sumsub's SDK; sensitive media stays in Sumsub / UR.APP supervised infrastructure.
- **API orchestration only:** HyperTrade calls UR.APP endpoints to mint SDK tokens and poll verification status.
- **No doc vault:** HyperTrade operational databases hold wallet links, job state, and feature flags — not KYC document archives.
- **UR.APP-led retention:** document retention, AML obligations, and regulatory record-keeping are governed by UR.APP and Sumsub under their licenses.

Gasless flows: EIP-7702, Ambire, and permit relays

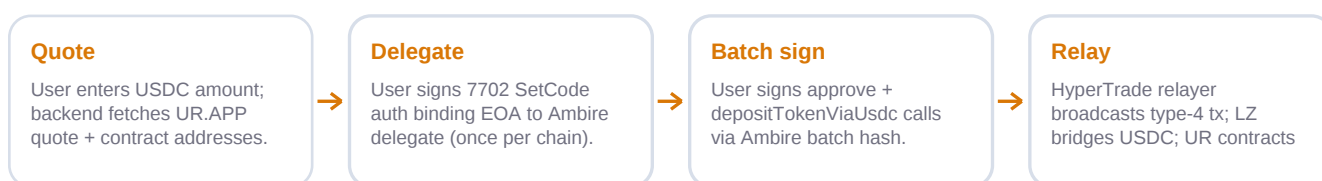
Several UR.APP banking actions require onchain authorization. HyperTrade sponsors gas so users never need native tokens for Mantle or Arbitrum operations. Two relay patterns cover the majority of flows:

Flow	Chain	Authorization pattern	What the user signs
Add Money (USDC → fiat)	Arbitrum → Mantle (LZ)	EIP-7702 + Ambire batched execute	7702 SetCode authorization (once per chain) + Ambire batch signature for approve + deposit
FX Convert (fiat ↔ fiat)	Mantle	EIP-7702 + Ambire batched execute	Same 7702 delegation + batch signature for approve + moneyExchangeExactIn
Withdraw / Payout	Mantle	EIP-2612 permit (REST settlement)	EIP-2612 permit authorizing UR.APP to debit fiat tokens
Hyperliquid deposit	Arbitrum	EIP-2612 USDC permit	Scoped USDC permit for relayer transferFrom

EIP-7702 delegation: on first use per chain, the user signs an EIP-7702 authorization that temporarily delegates their EOA to Ambire's audited AmbireAccount7702 implementation. The EOA's code slot becomes `0xef0100 + delegate address`, enabling batched execution with `_msgSender() == user EOA` so UR contracts (e.g. Fiat24CryptoDeposit) resolve the correct URID owner.

Ambire batch execution: the user signs an off-chain Ambire batch hash (raw secp256k1, no EIP-191 prefix). HyperTrade's dedicated UR.APP relayer pool broadcasts a type-4 (EIP-7702) transaction that attaches the authorization, invokes `EOA.execute(calls[], signature)`, and pays gas plus any LayerZero cross-chain fees. Relayer assignment is deterministic per wallet with Supabase locks to prevent nonce collisions across Railway replicas.

Add Money: EIP-7702 gasless deposit



Operational verbs enabled today and on roadmap

- **Pay-in:** user wires fiat to their IBAN; balance credits on settlement.
- **Payout:** user signs and sends fiat from IBAN balance to an external beneficiary via SEPA/SWIFT.
- **On-ramp / Add Money:** USDC on Arbitrum deposits into UR.APP contracts; Mantle balance credits as USD24/EUR24 (regulated fiat tokens, not raw USDC).
- **Off-ramp / Withdraw:** fiat tokens convert back to USDC in the user's wallet via permit-authorized settlement.
- **FX:** gasless fiat-to-fiat conversion on Mantle via 7702 + Ambire.
- **Debit card (roadmap):** Fiat Only mode — card spend draws from UR.APP fiat balance; card credentials revealed in UR.APP secure webview.

Withdraw and payout flows use UR.APP's permit REST API: the user signs an EIP-2612 permit on device, and UR.APP submits and pays for onchain settlement — HyperTrade's relayer does not move funds the user did not authorize.

7. Markets and Execution

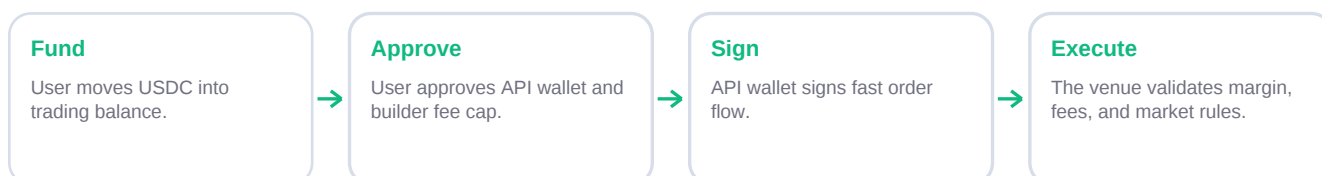
HyperTrade's live trading path uses Hyperliquid: info APIs, exchange APIs, WebSocket state, API-wallet order signing, builder fee approvals, and environment-scoped caches for live vs test.

Onchain perpetual markets are expanding beyond crypto. Public reporting in early 2026 cited Hyperliquid HIP-3 markets reaching roughly \$1.4B in open interest, with oil and other non-crypto markets driving significant demand. HyperTrade can list HIP-3 providers such as XYZ Markets and other deployers as they bring more of those books onchain — oil, gold, forex, large equities, and major indices — rather than only crypto-native tickers.

Outcome / prediction markets are a separate LWL product. **OrbCast** (orbcast.xyz, LWL-OrbCast/orbcast) is the HIP-4 interface: binary and multi-outcome books, sports as the showcase, self-custody, and no perps, HIP-3, banking, or AI worker in v1. HIP-4 is not a HyperTrade roadmap item and is not folded into the HyperTrade tab. **BuilderPad** (builderpad.xyz) is a separate web surface on the same HyperTrade stack: creators publish a branded perps/spot app (and optionally a coin). It is not an outcomes product.

Perpetuals, non-crypto HIP-3 books, and outcome books all point at the same user demand: trading views about the world, not only crypto token prices. LWL meets that demand with more than one product.

HyperTrade trading setup



8. Revenue Model and Sustainable Growth

LWL products are designed as interface layers. They do not need to operate a matching engine, custodian, market maker balance sheet, or exchange back office to begin monetizing. HyperTrade sits on Privy, Arbitrum USDC, optional UR.APP IBAN rails, Supabase, Railway, and Hyperliquid execution. OrbCast uses the same class of wallet and interface-fee model on HIP-4 books. BuilderPad uses the HyperTrade web stack for branded apps.

Where a product takes a transparent builder or interface fee, users approve a fee cap and orders can include that product's builder code. Revenue is usage-based. **Any** realized LWL product revenue — HyperTrade, OrbCast, BuilderPad, and future revenue-generating projects — can fund LWL buybacks and burns. The share applied is dynamic and never below 70% (see §11). Alignment is simple: revenue grows with real product use, not with custody or hidden spreads.

Revenue Stream	Product	Why It Scales
Builder / interface fees	HyperTrade (active)	Usage-based order flow through the interface, with user-approved fee caps.
Builder / interface fees	OrbCast	Same fee class on HIP-4 outcome flow as that product earns it.
Builder / interface fees	BuilderPad	Creator-set builder fee on fills through their branded app; optional one-time activate fee.
Cards and payments	HyperTrade (roadmap)	Possible partner revenue from card activity, IBAN top-ups, FX/spend, or negotiated share with UR.APP.
AI-assisted agents	HyperTrade (active)	House LLM + market-data pipeline; monetization aligned with fees on agent-originated flow.
Future products	LWL	Any later revenue-generating LWL project uses the same single-token buyback and burn policy.

Cost structure

- **Low fixed cost:** Railway, Supabase, Privy, RPC providers, notifications, analytics, and standard monitoring are manageable compared with operating exchange or custody infrastructure.
- **Usage-aligned upside:** builder revenue increases as real trading usage increases.
- **Operational leverage:** more users do not require a proportional increase in headcount or servers because core coordination is stateless and lock-driven.
- **Runway discipline:** modest infrastructure costs reduce pressure to over-monetize early users before the product and regulatory environment are ready.

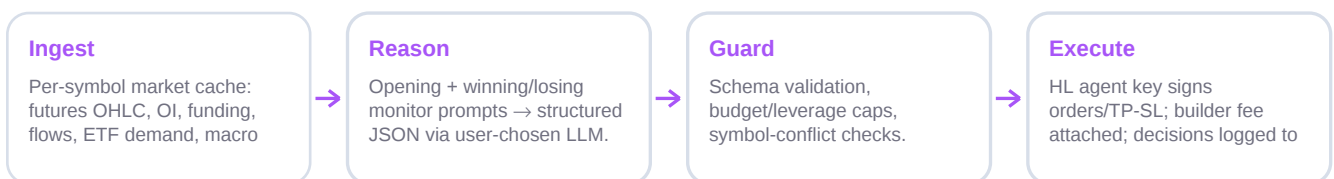
- **Business model principle:** monetize the interface layer transparently while keeping custody, settlement, and market execution in specialized infrastructure.

9. AI Agents and Market Intelligence

AI is a HyperTrade surface, not an OrbCast v1 feature. HyperTrade combines **in-app market analysis** (Gemini-backed research for human traders) and **AI Trading Agents** — autonomous agents that monitor Hyperliquid perps, reason over structured market context, and place orders through HL named-agent keys the user explicitly approves.

The agent stack is split into a **control plane** and an **execution plane**. The mobile app and FastAPI backend handle agent CRUD, Privy JWT auth, wallet-ownership checks, and the approve-agent ceremony. A dedicated **ai-agent-worker** service (Node/TypeScript on Railway) runs the trading brain and HL order flow — it holds no public HTTP surface and talks to Supabase with the service role only.

AI agent runtime (hourly cycle)



Agent modes and custody boundary

Agents run in two modes. **Shared** trades from the user's main unified Hyperliquid balance, bounded by per-agent **max_capital_usd**, optional per-position caps, leverage limits, and a symbol allowlist. **Dedicated** trades from a segregated HL sub-account (volume-gated), with the same named-agent signing model and a hard isolation boundary at the clearinghouse. Each agent receives its own HL keypair; the user approves it as a **named agent** (*approveAgent*) alongside the device's manual trading agent. HL protocol guarantees agent keys can sign L1 order actions only — not withdrawals — even if backend data were tampered with.

Agent private keys are generated in the worker, encrypted with **AES-256-GCM** under a Railway-only **AGENT_KMS_KEY**, and stored as ciphertext in Supabase. The backend never holds LLM API keys; model inference runs on the worker using house provider keys (Gemini, Grok, OpenAI, DeepSeek, Claude). Users can pause, stop, or revoke agents at any time.

Brain: data → flags → prompts → validation

Market context is assembled in two phases each cycle. **Phase 1** builds a shared per-symbol snapshot once per coin — aggregated futures and spot history, open interest, funding, taker flow, liquidations, premium/basis vs spot, plus symbol-specific overlays: Deribit DVOL options context for BTC/ETH, spot-ETF flow history for BTC/ETH/SOL/XRP, and a macro calendar slice (US holidays + high-impact CPI/FOMC windows). Slow-moving context is persisted in a **global_context_cache** table so all agents and worker restarts share one refresh, not N duplicate pulls.

Phase 2 fans out per active agent. Raw series feed **computeScalperFlags** and a composite long/short score. The brain then branches:

- **Opening path** — for flat symbols: opening prompt with session context, conviction bands, planned stop anchors, and macro/ETF sections. Responses are JSON-validated; sub-conviction setups are skipped until market context updates.
- **Winning monitor** — for profitable positions: hold / add / trim / exit with stop-management rules (breakeven, tighter) that never loosen protective stops.
- **Losing monitor** — for underwater positions: hold / cut / exit with stored opening thesis and invalidation criteria replayed into the prompt.

LLM output is treated as **untrusted input**. Validators clamp leverage, size, symbol allowlists, and reduce-only integrity before the HL adapter sees an order. Live mids from Hyperliquid (cycle-cached) anchor decisions; cached OHLC snapshots are fallback when mids are unavailable. Opening decisions for the same symbol + model dedupe within a cycle so parallel agents do not burn duplicate LLM calls.

Execution adapter and conflict guards

The HL execution adapter is a server-side port of HyperTrade's signing semantics: unified-account free margin, position-linked TP/SL via *positionTpsl*, reduce-only IOC exits with in-cycle retries, and HyperTrade's builder fee on every agent order. Budget headroom for new opens uses live mark notional so inflated winners consume cap honestly.

Guard	Behavior
User conflict	Skip opening a coin where the user already holds a manual position the agent does not own.
Peer conflict	Shared agents on the same master wallet run sequentially and claim symbols so two agents cannot race the same coin. Dedicated agents are isolated on their own sub-account.
Manual close	If the user closes or reduces an agent position externally, the worker adopts reality and does not re-open.
Order tagging	Agent orders carry a deterministic <i>cloid</i> prefix so portfolio UI can badge bot-managed coins.

Every cycle writes an audit trail: **ai_agent_decisions** (structured decision + reasoning JSON), **ai_agent_positions** (thesis, stops, PnL state), and **ai_agent_runs** (equity snapshot for dashboard charts). Supabase RLS is deny-all on agent tables — only the backend and worker service role can read or mutate rows, scoped by *privy_user_id* at the API layer.

- **Product principle: AI agents automate research and execution within explicit user-approved bounds — named HL agent keys, not custodial trading accounts.**

10. Community Rewards and Roadmap

HyperTrade rewards active users through a loyalty tier system. Higher tiers unlock trading fee discounts tied to real usage — trading activity, referrals, and product engagement — so incentives stay aligned with authentic participation rather than passive farming. OrbCast has its own rewards surface; the two programs are not the same ledger.

Roadmap themes

- **LWL:** keep publishing open-source products, keep a single token, and apply the buyback and burn policy in \$11 to any product revenue.
- **HyperTrade:** improve the core trading interface for casual and advanced users.
- **HyperTrade:** expand optional stablecoin utility through UR.APP IBAN accounts, Fiat Only card partnerships, and everyday finance use cases.
- **HyperTrade:** operate AI Trading Agents (Shared and Dedicated) with worker-side brain, global market cache, and named-agent signing.
- **OrbCast:** grow the HIP-4 outcome catalog and sports/event UX. No perps, banking, or AI worker in v1.
- **BuilderPad:** grow branded Hyperliquid apps (and optional creator coins) on the existing web stack. No HIP-4, banking, or AI worker in v1.
- Maintain non-custodial architecture as each product expands.
- Deepen loyalty-tier fee discounts as usage and product surface area grow.

11. LWL Tokenomics

LWL is the token of **Lunatic Wisdom Labs LLC**. It is the **only** LWL token. HyperTrade, OrbCast, BuilderPad, and any later LWL product do not get their own tickers. LWL launched via a **fair launch** on Uniswap through **pools.trade** on Robinhood Chain.

Max supply is **1,000,000,000 LWL** (1 billion). No additional mint is described in this document.

Allocation	Share	Tokens	Terms
Public	55%	550,000,000	Fair-launch float on Uniswap via pools.trade (Robinhood Chain).
Ecosystem	20%	200,000,000	Locked until November 2026.
AI agents	10%	100,000,000	Locked until January 2027. Not for sale — see note below.
Team	15%	150,000,000	Locked until August 2027.

AI agents allocation: the 10% LWL reserved for AI agents is not for sale and can be treated as outside active circulation. A time lock is not the same as a burn — those tokens remain in existence until a future burn mechanism, if introduced, removes them.

Buybacks and burns

Product activity supports LWL through on-market **buybacks and burns**, not by selling reserved allocations. The policy applies to **any** realized LWL revenue: HyperTrade, OrbCast, BuilderPad, and future revenue-generating projects.

- **Single token:** only LWL. The labs will not issue a separate token per product.
- **Source:** realized product revenue (interface / builder fees and any later LWL revenue lines), plus 100% of pools.trade creator fees from the LWL fair-launch pool.
- **Share:** the portion of realized revenue used for buybacks and/or burns is **dynamic** and depends on market conditions. It is **never less than 70%**.
- **Form:** the split between buyback and burn inside that share can move with conditions; both are in scope.

Reserved AI-agents allocation

A future mechanism may still retire the locked 10% AI-agents allocation through a burn rather than a sale. That path is not live in this draft. If introduced, it would be additive to the revenue-funded buyback and burn policy above — not a second token.

On-chain references

Locks and the token itself are on Robinhood Chain. Readers can verify supply and lock state on the explorer rather than taking the table on trust:

- **LWL token:** [0x7bb3E171EC502F65C08D38a61D51B9841524A72D](https://explorer.robinhoodchain.com/address/0x7bb3E171EC502F65C08D38a61D51B9841524A72D)
- **Lock contract:** [0x20f0137CD23411bc165BE14eA0a4F8D59E59C505](https://explorer.robinhoodchain.com/address/0x20f0137CD23411bc165BE14eA0a4F8D59E59C505)

LWL is not required to use HyperTrade, OrbCast, BuilderPad, hold a wallet, or trade. This section describes token supply, locks, and revenue-linked buybacks and burns only. It is not an offer, solicitation, or investment advice.

12. Risk, Security, and Trust Model

Each LWL product is a user interface, not a custodian. The table below is HyperTrade-weighted because that is where banking, 7702 relays, and AI agents live. OrbCast inherits the wallet, relayer, and interface-fee rows that apply to it. BuilderPad inherits those same wallet and interface-fee rows on the HyperTrade web stack. Neither inherits UR.APP, Sumsb, 7702 banking, or AI-agent key rows.

HyperTrade's security model is narrow by design. User wallets sign. Relayers relay. The trading venue executes. Supabase coordinates operational state. This separation reduces the blast radius of any one component and keeps wallet-key custody with users for the trading stack; optional UR.APP fiat-token flows add a regulated compliance layer governed by UR.APP (\$6.1).

Risk	Current Control
User key custody	Privy device-side embedded wallet model; backend never stores user private keys.

Gasless relay abuse	User-signed permits, deterministic relayer assignment, replay checks, rate limits.
Replica nonce collision	Supabase relayer locks with TTL recovery.
Setup incompleteness	Setup considered complete only when agent is active and builder fee is approved.
Third-party dependency	Clear dependency on Privy, Arbitrum, Mantle, Hyperliquid, UR.APP, oracles, and card partners.
7702 relay abuse	User-signed 7702 authorizations and Ambire batches; relayer bound to URID owner; rate limits and inflight deposit guards.
Regulatory uncertainty	Non-custodial interface posture; UR.APP-led KYC/AML for regulated IBAN and card rails.
UR fiat-token compliance	USD24/EUR24 balances use UR.APP contracts; lawful freezes, holds, and KYC-gated restrictions are expected on the regulated rail — not applicable to raw wallet USDC or Hyperliquid-only usage.
AI agent key custody	Per-agent HL keys encrypted AES-256-GCM; user approves named agent; HL blocks withdrawals from agent keys.
LLM output abuse	Structured JSON validators + adapter hard caps on size, leverage, symbols; untrusted model output never sent raw to HL.
KYC data handling	Sumsb SDK on device; HyperTrade stores verification outcomes only — no identity documents or biometric archives on HyperTrade servers.

13. Regulatory and Policy Landscape

LWL products should be understood as user interfaces and wallet-connected applications, not as custodians. Market type still differs by product: HyperTrade is a perpetuals and optional-banking interface; OrbCast is an outcome-markets interface; BuilderPad is a branded Hyperliquid trading interface (web), not a new venue. Those are different legal and store surfaces. This distinction is increasingly important as regulators clarify how self-custodial trading interfaces fit within existing frameworks.

In April 2026, SEC Division of Trading and Markets staff issued a statement on broker-dealer registration for certain user interfaces used to prepare crypto asset securities transactions. The statement is not a Commission rule and has no independent legal force, but it is directionally important: it describes conditions under which staff would not object to certain covered user interface providers operating without broker-dealer registration, including self-custodial wallet contexts, objective parameters, educational material, disclosures, and no custody of user funds.

LWL product design is consistent with several principles highlighted in that staff statement: users control wallets, the app prepares transaction parameters, fees and limitations should be disclosed, and the interface should avoid discretionary control over user funds or execution decisions. This is not legal advice, and product rollout must continue to be reviewed jurisdiction by jurisdiction.

Separately, the Hyperliquid Policy Center launched in February 2026 as an independent research and advocacy organization focused on decentralized market infrastructure, perpetual derivatives, and practical regulatory frameworks. Policy clarity around decentralized markets can reduce uncertainty for builders, partners, and users of interfaces in that ecosystem.

14. Appendix: Factual Anchors, Audits, and Sources

Lunatic Wisdom Labs

- **Labs:** lunaticwisdomlabs.com — Lunatic Wisdom Labs LLC, Wyoming, USA.
- **HyperTrade:** hypertrade.exchange · LWL-HyperTrade/hypertrade
- **OrbCast:** orbcast.xyz · LWL-OrbCast/orbcast
- **BuilderPad:** builderpad.xyz · LWL-HyperTrade/hypertrade (web/ in the HyperTrade tree)

Regulatory and ecosystem references

- SEC Division of Trading and Markets staff statement, April 13 2026: covered user interfaces, self-custodial wallet contexts, objective parameters, disclosures, and no custody of user funds. The statement is staff view only and not a Commission rule.
- Hyperliquid Policy Center launch announcement, February 18 2026: independent 501(c)(4) research and advocacy organization focused on decentralized finance, perpetual derivatives, and blockchain-based financial infrastructure.
- Hyperliquid Policy Center About page: describes Hyperliquid as a public, permissionless blockchain and decentralized exchange known for perpetuals, transparency, accessibility, performance, and non-custodial use.
- Public market reporting in 2026 cited HIP-3 open interest around the \$1.4B range and highlighted oil / non-crypto perpetuals as major drivers of growth.

UR.APP developer documentation

- **Developer docs home:** docs.ur.app
- **Integration guide:** Account Mode, Card Mode, and KYC Mode choices — [integration guide](#)
- **External Wallet Access:** HyperTrade's Account Mode — [External Wallet Access docs](#)
- **Managed Custody:** alternative Account Mode (not used by HyperTrade) — [Managed Custody docs](#)
- **Contract source & audits:** [ur-app/ur-contracts](#) · [audit reports](#) (UR removed the public smart-contracts docs page)

Third-party security audits

HyperTrade relies on audited infrastructure from wallet, execution, relay, and UR.APP partners. Independent audit reports for key dependencies:

- **Ambire EIP-7702 delegate (Hunter Security):** audit of AmbireAccount7702 used for gasless batched execution — [view audit report](#)
- **UR.APP onchain contracts (Blocksec):** Fiat24 deposit, relay, on-ramp, card authorization, and Mantle contract suites — [view audits on GitHub](#)
- **LayerZero (Endpoint V2, OFT, index):** cross-chain messaging and omnichain token bridges for Add Money (Arbitrum → Mantle) — [Endpoint V2](#), [OFT](#), [full index](#)

- **Arbitrum native USDC (Trail of Bits):** audit of Arbitrum USDC custom gateway and ArbOS upgrade — [view audit report](#)
- **Hyperliquid (Zellic):** audit of Hyperliquid core infrastructure — [view audit report](#)
- **Privy Shamir Secret Sharing (Zellic):** audit of Privy embedded-wallet key-sharding architecture — [view audit report](#)

HyperTrade codebase anchors: Privy embedded EOA + ERC-4337 smart wallet; External Wallet Access (URID, IBAN, balances, history); gasless EIP-7702 + Ambire Add Money and FX on Mantle; EIP-2612 permit relays for Hyperliquid deposits and UR.APP withdraw/payout; dedicated UR.APP relayer pool with Supabase nonce locks; Hyperliquid API-wallet + builder-fee setup; frontend margin and liquidation guards; ai-agent-worker (leader-gated cycles, global market cache, multi-LLM brain, HL named-agent adapter); Sumsb KYC via mobile SDK with verification outcomes only — no document storage on HyperTrade servers.

OrbCast codebase anchors: Privy embedded EOA; HIP-4 trader client (venue-generic); EIP-2612 permit relays for USDC deposits; builder-fee setup; sports chrome is fixtures/score only — not venue odds. No perps, HIP-3, UR.APP banking, or AI worker in v1.

BuilderPad codebase anchors: Vite + Privy web console in the HyperTrade tree (*web/*); dual EOAs (trade vs builder); preview vs own-builder activation; branded apps at {slug}.builderpad.xyz; optional Pons v2 creator coin on Robinhood Chain. No HIP-4, UR.APP banking, or AI worker in v1.

This document is not legal, investment, tax, or trading advice. Product roadmap items are forward-looking and subject to provider, regulatory, technical, and market constraints.