

Levyx

A Clear Market for Trade Across Time

A trading and settlement platform for time-based contracts,
anchored on Arbitrum with selective zero-knowledge attestations.

White Paper | Version 1.1

7 October 2026

ON THE SHOULDERS OF GIANTS

Open a position today; let settlement carry it to its conclusion.

By

Levyx Labs



Levyx at a Glance

What it is	A venue and settlement layer for time-based crypto contracts.
Built on	Arbitrum, with a WASM margin engine and on-chain settlement.
Privacy	Zk-solvency attestation: aggregate solvency proven without exposing accounts.
Design	On-chain and off-chain parallelism with a canonical event record.
Scope	Perpetuals first, then futures, options, and time-window contracts.

Abstract

LEVYX is a trading platform for time-based contracts. Every agreement it supports is defined by a specification: an underlying, a size, a quote, a schedule of events, and a settlement rule. The platform records the trade at agreement time, manages collateral and risk for the duration of the position, measures the contract against a published reference at a defined observation, and settles it in cash. Time is not an implicit background condition; it is a first-class field of every instrument.

This white paper sets out the contract families Levyx supports, the platform architecture, the market and risk rules, and the roadmap through which the platform acquires further capabilities and reaches its target state. The platform is built on Arbitrum. Settlement is anchored on-chain, the margin engine runs on WASM execution for compute-heavy risk evaluation at a fraction of EVM cost, disputes resolve within a fixed bounded window, and soft confirmations reach participants in under two seconds. The platform also publishes a zero-knowledge solvency attestation: aggregate collateral, aggregate margin requirement, and the commitment to account state are verifiable on-chain, while individual account balances remain private.

Latency-sensitive work runs off-chain; finality-sensitive work is anchored on-chain. The two domains exchange only the minimal state needed to reconcile, so the platform applies modern parallel and proof-based technique without sacrificing auditability.

LEVYX provides shared rules and shared records so that participants know what they are trading, when each obligation falls due, and how the outcome will be calculated.

Contents

1	Executive Summary	3
1.1	Related Work	4
2	Contract Design and Position Accounting	4
2.1	The perpetual contract	4
2.2	Contract families	5
2.3	Position accounting	5
3	The Life of a Position	6
3.1	Order, matching, and reconstruction	6
4	Risk Engine	7
4.1	Computation	7
4.2	Mark price construction	8
4.3	Liquidation and the loss waterfall	8
5	System Architecture	9
5.1	Latency-sensitive path	9
5.2	Finality-sensitive path and reconciliation	10
6	Arbitrum as Execution and Settlement Layer	10
6.1	Stylus and bounded dispute resolution	10
6.2	Orbit as a scaling path and zero-knowledge attestation	11
7	Trust Boundaries	12
7.1	On-chain components	12
7.2	Off-chain components and reference data	13
8	Market Integrity and User Protection	13
8.1	Prevention and detection	13
8.2	Emergency controls and the market-rule handbook	14
9	User Journey and Governance	15
9.1	Account opening, trading, and close-out	15
9.2	Governance and change management	15
10	Composability, Compliance, and Operating Model	16
10.1	Composability with on-chain systems	16
10.2	Compliance and operating model	16
11	Capability Roadmap	17
11.1	Staged acquisition	17
11.2	Metrics and reporting	18
12	Conclusion	18
12.1	What Levyx is	18
12.2	What Levyx is not, and the path forward	19

1 Executive Summary

Perpetual futures let a trader hold a leveraged long or short position in an asset such as BTC or ETH without owning it and without a fixed expiry. That flexibility creates a demanding risk problem: positions are leveraged, collateral is shared across accounts, prices move continuously, and the venue must decide when a position is safe, when it must be reduced, and how losses are contained when a position can no longer be saved. The design challenge is not the existence of perpetuals; it is the quality of the rules that govern them.

LEVYX is a venue and settlement layer for crypto perpetual markets, built on Arbitrum. It combines a transparent order and position system with explicit collateral rules, funding payments, oracle controls, and on-chain settlement infrastructure. The design is intended for traders, market makers, and risk managers who need a clear answer to five questions: what position is open, what collateral supports it, which price is used for risk, what happens during liquidation, and when the result is final. Each of those questions has a published answer, and each answer is reconstructable from the event record.

The platform is designed around five direct principles:

1. **Continuous exposure without expiry confusion.** A perpetual contract has no delivery date, but it still has precise funding intervals, price observations, margin checks, and close-out rules. The absence of an expiry is not the absence of structure.
2. **Collateral before leverage.** Every position is backed by defined collateral, haircuts, maintenance requirements, and account-level risk limits. Leverage is a risk parameter, not a product promise; the platform publishes the maximum it will allow and the conditions under which that maximum is reduced.
3. **Funding keeps the market anchored.** A published funding mechanism transfers payments between longs and shorts so that the perpetual price remains linked to the reference spot market. Funding intervals, the calculation formula, and the settlement of each payment are part of the contract specification, not an operational detail.
4. **Liquidation is a controlled sequence, not an event.** The system cancels risk-increasing orders, reduces positions in a defined order, and applies a published loss waterfall when account equity is insufficient. Liquidation is a sequence of steps, each visible in advance.
5. **Fast trading, auditable settlement.** Matching and risk updates must be responsive, while the canonical ledger, oracle inputs, and settlement assertions remain independently verifiable. Speed on the trading path does not compromise the audit trail on the settlement path.

The quality of the market depends on liquidity, oracle reliability, conservative risk parameters, secure infrastructure, and compliance with the rules of each jurisdiction in which it operates.

1.1 Related Work

Perpetual futures were introduced by BitMEX in 2016 as a crypto-native instrument [1]. The funding-rate mechanism that anchors a perpetual to spot is described in the BitMEX documentation and has been adopted, with variations, by nearly every crypto derivatives venue since. Decentralized perpetual venues such as dYdX [2], GMX [3], and the Aave-based GH0 market [4] have each taken a different position on the trade-off between matching speed and settlement certainty, but all of them operate the same basic instrument.

Academic work on decentralized exchange liquidity shows that professional liquidity providers dominate retail participants in return and skill, a pattern that has repeated across every on-chain venue studied [5]. Bounded liquidation designs have been proposed to replace abrupt close-outs with continuous, granular reductions, but these designs are typically tightly coupled to stablecoin mechanisms and carry their own oracle dependencies [6]. Zero-knowledge proofs of solvency for centralized exchanges have advanced considerably, with constructions that handle the cryptographic mismatch between secp256k1 asset keys and pairing-friendly SNARK curves [7], and compound-statement techniques that prove both asset ownership and liability coverage in one proof [8]. Manipulation-resistant oracles have moved from arithmetic-mean TWAPs to piecewise-parabolic median estimation with constant storage, reducing both error and delay while using less gas than TWAP [9].

LEVYX is not the first venue to offer perpetuals, and it does not claim to reinvent the instrument. Its contribution is architectural: an explicit on-chain and off-chain split, a WASM margin engine that runs the risk computation off the EVM hot path, a bounded dispute protocol for settlement finality, and a zero-knowledge solvency attestation that lets the operator prove aggregate solvency without exposing individual accounts. The paper also addresses a class of contract that perpetual venues typically do not: bounded, specifiable agreements with a defined observation window and a defined settlement event, of which the perpetual is one special case.

2 Contract Design and Position Accounting

This section defines the instruments LEVYX supports and the accounting rules that govern them. The perpetual is treated as one family within a broader class of time-based contracts, all of which share a common specification format and a common margin engine. Position accounting then provides the notation and formulas used throughout the rest of the paper.

2.1 The perpetual contract

A perpetual contract is an agreement between two parties to exchange the profit or loss on an underlying asset, with no expiry date. Positions are opened by matching buy and sell orders, held for as long as the participant chooses, and marked to a reference price at regular intervals. To keep the perpetual price tethered to the underlying, the venue periodically transfers a funding payment between longs and shorts. When a position closes, the difference between entry and exit is settled in the quote currency, typically USDT or USDC.

A perpetual is not a delivery contract. There is no underlying to receive, no delivery location, no inspection, and no rejection process. The obligation is denominated in cash

and settles in cash. The venue’s obligations are entirely financial: collateral, margin, funding, and close-out.

Even without an expiry, a perpetual position has several distinct moments: the **order** (the instruction to open, reduce, or close); the **match** (the execution that creates or modifies a position); the **funding** payment (the periodic transfer between longs and shorts); the **mark update** (the risk price used to compute margin); the **margin check** (the evaluation of account equity against maintenance requirement); the **liquidation** (the forced reduction of a position that no longer meets its margin requirement); and the **close** (the final settlement of the position’s economic result). Each of these moments has a precise definition in the contract specification, a precise timestamp in the event record, and a defined effect on the participant’s account.

2.2 Contract families

LEVYX treats the perpetual as one family of a broader class of time-based instruments. Every family shares the same underlying structure: an underlying, a size, a quote, a schedule of events, and a settlement rule. The families differ in what happens at expiry and how the position is measured along the way.

Perpetual. No expiry. Funding payments anchor the price to spot. Close is initiated by the participant or triggered by the margin engine.

Futures-style. Fixed expiry. Positions are marked to market and settled at expiry against a published settlement price, or closed earlier by the participant.

Option-style. Fixed expiry. The buyer holds a right, not an obligation. The buyer pays a premium at open; the seller posts collateral because the seller may have to perform.

Time-window. No fixed price. The contract’s value is derived from an interval — an average price over a day, a maximum over a week, a settlement against a specific index at a specific timestamp. The sampling schedule and the missing-observation rule are part of the specification.

All four families share the same specification format, the same collateral and margin engine, the same reference-data pipeline, and the same settlement process. A participant who understands one family can read the specification of any other.

2.3 Position accounting

For each participant, LEVYX maintains a signed position q in each instrument. A positive q is long; a negative q is short. The contract size is S ; the current mark price is P ; the entry price of the position is P_0 . The notional value of the position is

$$N = |q| \times S \times P. \quad (1)$$

Unrealized profit and loss is

$$\text{PnL}_{\text{unrealized}} = q \times S \times (P - P_0). \quad (2)$$

The specification of each instrument states how these formulas are adapted for inverse quotes, options, or non-linear payoffs.

Initial margin is the collateral required to open a position. Maintenance margin is the minimum collateral required to keep it open. Both are set per instrument by the risk engine and published before order entry. When account equity falls below maintenance margin, the engine enters the liquidation sequence described in Section 4.3. Liquidation is not a guarantee of a good price; in a fast market, the closing price can be worse than the mark price that triggered the sequence. Any shortfall is assigned by a published waterfall described in Section 4.3.

Funding keeps the perpetual price aligned with the underlying spot market. At each funding interval, the venue computes a funding rate from the difference between the perpetual's mark and the reference spot index, and transfers the resulting payment from one side of the book to the other. The rate formula, the interval, the cap, and the settlement of each payment are published in the instrument specification. Funding is not a fee paid to the venue; it is a transfer between participants.

The mark price is used for risk, not for trade execution. It is computed from a robust combination of the venue's own order book and an external index, with explicit rules for rejecting stale or outlying observations. The arithmetic-mean time-weighted average price (TWAP) that many on-chain oracles use is vulnerable to manipulation and requires a long observation window to resist it, which in turn introduces delay and error [9]. LEVYX uses a manipulation-resistant construction in the same family as Ormer's piecewise-parabolic median estimation [9]: the estimate is a median rather than a mean, the storage requirement is constant regardless of the number of observations, and the delay is suppressed by fusing window sizes. The method, the update frequency, the stale-data timeout, and the fallback behavior are part of the instrument specification and are visible to every participant.

3 The Life of a Position

A position passes through a sequence of moments, and each moment has a corresponding event in the ledger. Figure 1 shows the sequence. This section then describes the two operational processes that govern how positions are created and how the resulting state is recorded: order matching and ledger reconstruction.

3.1 Order, matching, and reconstruction

LEVYX may use a central limit order book, a request-for-quote workflow, or both. The matching rule is published. The default is price-time priority: the best price matches first, and orders at the same price match in arrival order. The matching engine records the maker, the taker, the timestamp, the fee, and the resulting position change as part of the trade event.

Order entry is separated from matching so that identity checks, rate limits, and balance verification can run without adding latency to the matching path. The gateway validates the order against the participant's current position and collateral, assigns a client order identifier, and forwards the order to the matching engine. Rejections carry a machine-

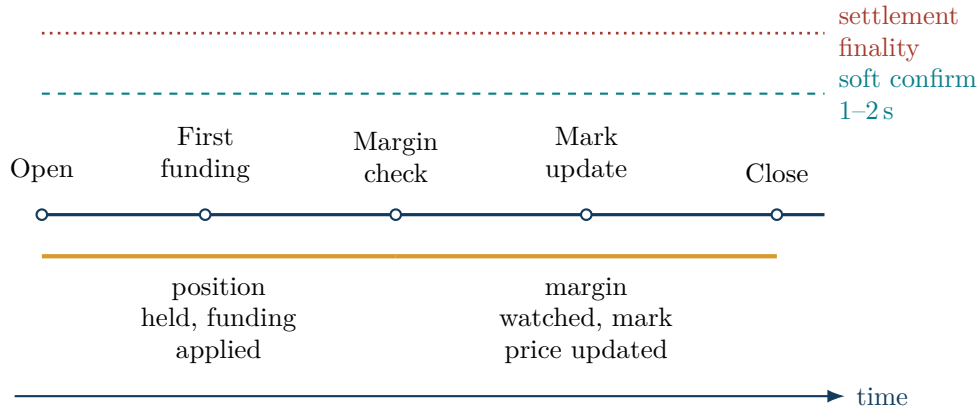


Figure 1: The life of a perpetual position: open, funding, margin checks, mark updates, and close. Two finality horizons govern when the result is treated as confirmed by different parts of the system.

readable reason code, and every rejection is recorded as an event even though no trade occurred. This matters for dispute resolution: a participant who claims an order should have been accepted can point to the rejection event and the state that produced it.

Every state change is appended to the ledger with a unique identifier and a timestamp. A participant can reconstruct their balance, position, funding payments, fees, and settlement amounts from the event history alone. Silent edits are not permitted; corrections are new events that reference the records they amend. Reconstruction is designed to be verifiable without operator cooperation. A participant who stores their own copy of the event stream can replay it against the published specification of each instrument and recover the same balance and position that the operator reports. If the two disagree, the disagreement is itself a dispute that can be opened under the bounded protocol. The ledger is therefore not merely an internal record; it is the shared source of truth that makes every other claim about the account checkable.

4 Risk Engine

The risk engine is the part of the platform that decides whether an account is safe, whether it needs more collateral, and whether a position must be reduced. It runs continuously while the market is open.

4.1 Computation

The engine repeatedly evaluates a large number of open positions against a moving mark price, applying non-linear margin schedules and liquidation checks. In EVM bytecode this is expensive per position, and the cost grows with the number of accounts. LEVYX runs the margin engine under WASM execution, where compute- and memory-intensive operations cost roughly an order of magnitude less than equivalent Solidity logic. The engine can therefore run more frequently, with richer margin schedules, than an EVM-only design would economically allow.

The engine is deterministic: given the same account state and the same mark price, two

runs produce the same result. This property is required for dispute resolution, because a challenger must be able to reproduce the operator's computation independently. The engine is unit-tested against a published fixture set, and the fixtures are part of the instrument specification.

4.2 Mark price construction

The mark price is not the last traded price. It is a robust estimate built from the venue's own order book and an external reference index, with the following properties:

- the estimate rejects observations outside a published bound;
- the estimate falls back to the reference index when the book is stale;
- the estimate has a defined maximum age; observations older than the bound are not used;
- the update frequency and the fallback rules are published per instrument.

The construction follows the manipulation-resistant design pattern established by Ormer [9], which uses a piecewise-parabolic median estimate rather than an arithmetic mean, with constant storage and lower gas than TWAP. Because the estimate is a median, an attacker who controls a minority of observations cannot move it outside the range of honest observations; because the storage is constant, the cost of the estimate does not grow with the observation window; and because window sizes are fused, the delay between a price move and its reflection in the mark is bounded.

4.3 Liquidation and the loss waterfall

When account equity falls below the maintenance margin, the engine enters a deterministic liquidation sequence:

1. warn the participant;
2. cancel orders that would increase risk;
3. request additional collateral, if the instrument allows a grace window;
4. partially close the position under the published reduction rule;
5. close the remaining position under the liquidation rule.

The sequence is deterministic given the account state and the mark price; two runs of the engine on the same inputs produce the same outcome. The sequence and its parameters are part of the instrument specification. The stepwise, bounded reduction approach stands in contrast to abrupt hard-liquidation models and to the overly complex soft-liquidation designs that couple price dynamics to virtual reserves [6].

When a position closes at a loss larger than the account's remaining collateral, the shortfall is assigned in a fixed order. Figure 2 shows the waterfall.

Every stage of the waterfall is published and auditable. The first stage absorbs the loss against whatever collateral remains in the account. The second absorbs it against profit

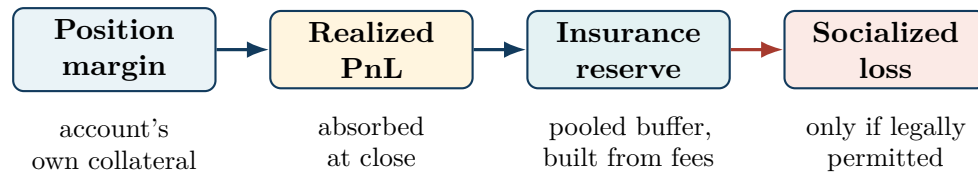


Figure 2: Default loss waterfall. Each stage is published and auditable. The socialized-loss stage is enabled only where local law permits.

and loss that has already been realized and credited. The third draws on the insurance reserve — a pooled buffer built from trading fees and held in a segregated account. The fourth, socialized loss, is enabled only where local law permits; it is spread across the accounts that hold positions in the affected instrument, in proportion to the size of each position. The order of the waterfall is fixed in the specification and cannot be changed retroactively for a position that is already open.

5 System Architecture

LEVYX separates fast trading from slower, verification-heavy controls. The separation is deliberate: the trading path must be responsive, while the settlement path must be reconstructable and independently verifiable.

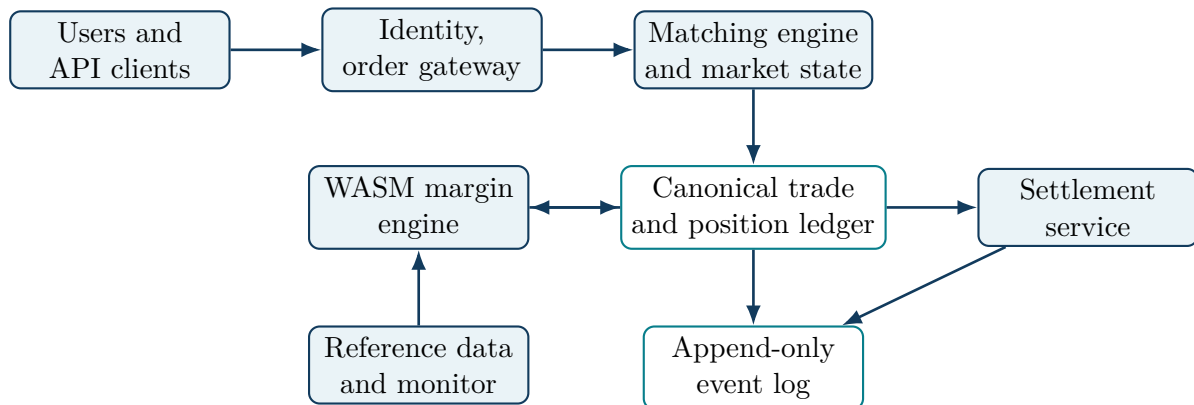


Figure 3: LEVYX logical architecture. Latency-sensitive work is on the left and centre; finality-sensitive work is on the right. The ledger is the authoritative record.

5.1 Latency-sensitive path

Order matching, mark-price updates, and margin recomputation run off-chain in high-throughput services. These components operate on the hot path and are optimised for throughput and predictable latency. They do not, by themselves, constitute finality; they propose state changes to the ledger.

The matching engine maintains the order book in memory and publishes trade events as they occur. The risk engine consumes trade events and mark-price updates, recomputes account equity, and emits margin-call and liquidation events when thresholds are crossed. Both services are horizontally scalable and can be restarted from the ledger without loss,

because the ledger is the authoritative record and the services are reconstructable from it.

5.2 Finality-sensitive path and reconciliation

Settlement assertions, collateral custody, and zero-knowledge verification are anchored on-chain. Once a settlement is posted, its correctness can be challenged and defended under a bounded protocol. The finality horizon for settlement is longer than for matching, and the instrument specification states both horizons.

Settlement assertions are posted as batched state commitments rather than as individual trade-by-trade transactions. Batching reduces the on-chain cost per settlement and lets the settlement service amortise proof verification across many positions. The batch interval is published, and the soft confirmation horizon for trades inside a batch is independent of the batch interval.

The two paths exchange only the minimal state needed to keep them consistent. The matching engine tells the ledger what trades occurred; the risk engine tells the ledger what positions must be reduced; the settlement service tells the ledger what payments have been made. No component can silently modify the ledger; every change is an event. Minimum controls include encrypted transport, protected keys, multi-factor authentication, separated operator duties, rate limits, withdrawal delays for high-risk changes, monitoring for abnormal behavior, backups, and tested recovery procedures. Smart-contract security follows the established pattern of specification-driven audits: reentrancy is addressed through checks-effects-interactions and mutex locks, integer overflow through checked arithmetic, and batch-transfer operations through the constraint-based analysis techniques that identify missing validations before deployment [10]. Controls are tested through regular drills, and the results are reported alongside the capability metrics described in the roadmap.

6 Arbitrum as Execution and Settlement Layer

LEVYX is deployed on Arbitrum. Three properties of the stack are used directly and are worth stating explicitly. Each property maps to a specific operational requirement, and each is described in its own subsection below.

6.1 Stylus and bounded dispute resolution

Stylus executes WASM directly and reduces the cost of compute- and memory-intensive operations by roughly an order of magnitude relative to equivalent Solidity logic. The margin engine is written in a memory-safe systems language, unit-tested independently, and exposed through a Solidity interface that settlement contracts call. The result is a risk engine that runs more frequently and with richer margin schedules than an EVM-only design would economically allow. Because the margin engine runs under Stylus, the settlement contracts remain simple. They do not contain the margin computation itself; they contain the interface to it and the final settlement logic. This separation keeps the on-chain attack surface small while allowing the compute-heavy work to run efficiently.

Settlement assertions posted to Arbitrum resolve under a permissionless, time-bounded dispute protocol. If a settlement state is challenged, an honest party can defend the

correct result, and the dispute concludes within a fixed window rather than an open-ended interactive game. This matters when settlement amounts are large enough that an unbounded dispute period would tie up capital or create uncertainty for counterparties. Contract specifications therefore define two finality states: a soft state visible to users within seconds, and a settlement-final state after the dispute window has closed. The dispute protocol is what makes the soft state safe to rely on for trading while preserving the option to challenge a bad settlement. A participant who acts on a soft confirmation accepts a small risk of reversion; the instrument specification states that risk explicitly. A participant who requires settlement finality waits for the dispute window to close. Both horizons are published, and neither is hidden behind operational discretion.

6.2 Orbit as a scaling path and zero-knowledge attestation

If aggregate LEVYX traffic ever dominates the cost of posting to Arbitrum One, an Orbit chain settling to Arbitrum One preserves the same execution model and dispute security while isolating LEVYX traffic from general congestion. This is a scaling path, not a v1 requirement, and it does not change the contract specification format or user-facing rules. The decision to move to an Orbit chain is governed by published thresholds: sustained cost per settlement above a stated level, or sustained congestion on Arbitrum One that affects soft-confirmation latency. The thresholds and the migration process are part of the governance record. Until a threshold is crossed, LEVYX remains on Arbitrum One, and the simpler deployment is preferred.

LEVYX also publishes a zero-knowledge solvency attestation. The operator periodically proves that the aggregate collateral held across all accounts is at least the aggregate margin requirement plus the insurance reserve. The circuit takes private inputs (individual account balances and positions) and public inputs (a Merkle root commitment to account state and the aggregate margin requirement). The proof verifies in constant time on Arbitrum regardless of the number of accounts.

The construction draws on two lines of work. First, the Groth16 proof system [11] provides the succinct, constant-size proofs used for the aggregate solvency statement. Second, the approach follows the compound-statement technique of Kamara et al. [8], which proves both asset ownership and liability coverage in a single proof, and the Xiezhi construction [7], which addresses the cryptographic mismatch between account-based asset keys and pairing-friendly SNARK curves by transferring ownership claims into the SNARK-friendly setting rather than translating keys directly.

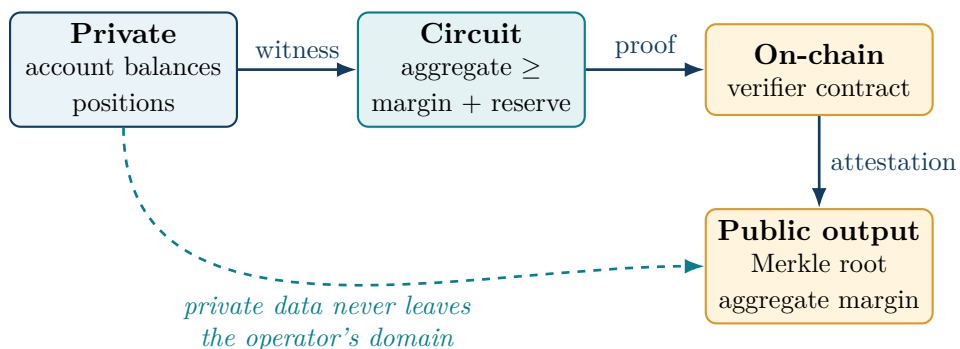


Figure 4: zk-solvency flow. Only the Merkle root and aggregate margin are public.

The attestation is published on a fixed schedule, and the schedule is part of the instrument specification. A participant can verify the proof independently using the public inputs and the verifier contract. If the operator fails to publish an attestation on schedule, the failure is visible and is itself a governance event. The attestation does not replace audited financial statements, but it provides a frequent, cryptographically verifiable check between audits.

Note

This proof does not hide trade sizes, prices, or counterparties. Those remain visible in the ledger because auditability is required for both regulatory and dispute-resolution purposes. The proof asserts one specific property — aggregate solvency — and nothing more.

7 Trust Boundaries

Every trading system rests on a set of trust assumptions. Some components must be trusted because they hold keys or execute logic; others can be verified independently. LEVYX makes those assumptions explicit. Figure 5 shows the three domains: on-chain contracts, off-chain operator services, and reference-data providers. The boundaries between them are narrow and well-defined.

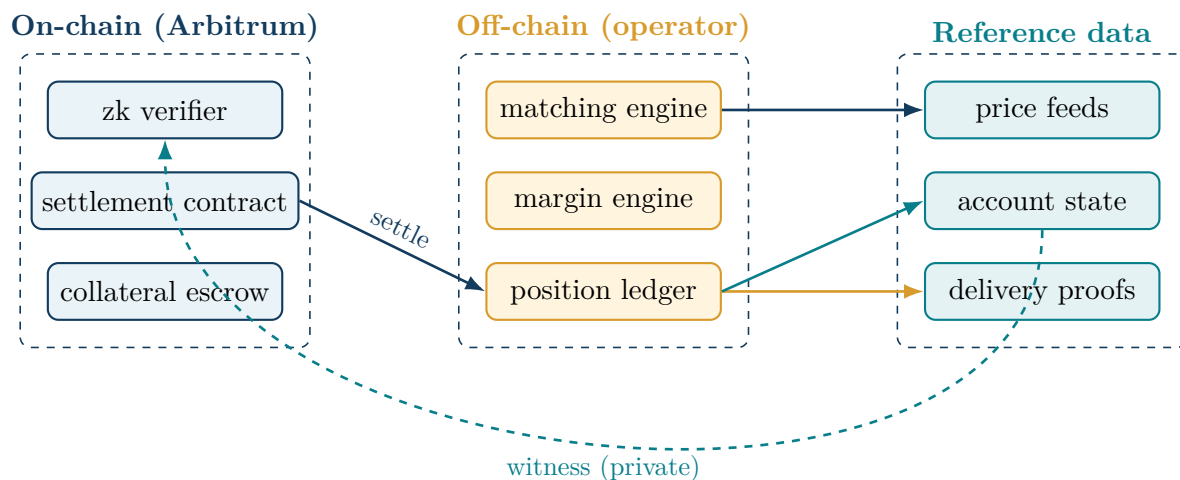


Figure 5: Trust boundaries. Solid arrows carry operational data; the dashed teal arrow carries the private witness used only inside the zk circuit.

7.1 On-chain components

The on-chain domain contains three contracts: the settlement contract, the collateral escrow, and the zero-knowledge verifier. These contracts hold custody of collateral and define the final settlement state. Their code is public, and their execution is enforced by Arbitrum’s virtual machine. Because they are on-chain, these components are subject to the same security discipline as any smart contract. They are written in Solidity,

audited against a published specification, and tested against adversarial inputs. The escrow contract holds user collateral; the settlement contract releases it according to the published rules; the verifier contract checks the solvency proof. No operator key can unilaterally move collateral.

7.2 Off-chain components and reference data

The off-chain domain contains the matching engine, the margin engine, and the position ledger. These components run on operator-controlled hardware. They are responsible for speed, throughput, and the continuous evaluation of risk. They are not, by themselves, final. They produce state changes that the on-chain settlement contract must confirm. Because they are off-chain, they are subject to operational security controls: encrypted transport, protected keys, multi-factor authentication, separated duties, and rate limits. The operator cannot silently alter the ledger; every change is an event with a unique identifier and a timestamp. The ledger is append-only, and corrections are new events that reference the records they amend.

The reference-data domain contains price feeds, account-state inputs for the zk circuit, and delivery proofs where applicable. Price feeds enter the mark price construction. Account state is the private witness for the solvency proof. Delivery proofs are used only for contracts that require them. Reference data is external to both the operator and the chain. It must be authenticated, timestamped, and validated against published bounds. A stale or outlying observation is rejected. The fallback behavior is part of the instrument specification. The operator does not control the reference price; it consumes it under published rules.

The dashed teal arrow in Figure 5 carries the private witness from the account-state store to the zk verifier. The witness never leaves the operator's domain in plaintext. The verifier receives only the proof and the public inputs: the Merkle root and the aggregate margin requirement. This is the only path in the system where private data crosses a trust boundary, and it crosses in encrypted, proof-wrapped form. The witness flow is deliberately narrow. It does not carry trade sizes, prices, or counterparties. Those remain visible in the ledger because auditability is required for both regulatory and dispute-resolution purposes. The proof asserts one specific property — aggregate solvency — and nothing more.

8 Market Integrity and User Protection

A market is only useful if participants can trust that prices reflect real supply and demand, that rules are applied consistently, and that abuse is detected and addressed. LEVYX makes manipulation costly and detectable through a combination of prevention, detection, and emergency controls. Figure 6 summarizes the control stack.

8.1 Prevention and detection

Prevention controls operate at order entry. They stop abusive patterns before they reach the book. Self-trade prevention blocks an account from matching against itself. Price-band checks reject orders that would execute at a price too far from the current mark. Maximum

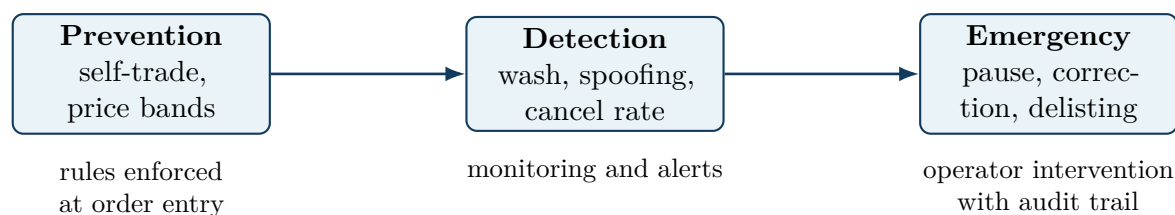


Figure 6: The market-integrity control stack. Each layer is published and auditable.

order size and maximum position size limits cap the risk any single account can introduce. These controls are deterministic and published. A participant can read the exact bounds before placing an order. The bounds are versioned alongside the instrument specification, and changes are announced in advance. Prevention is preferred over punishment because it avoids the need to unwind a trade after the fact.

Detection controls operate after orders are placed. They identify patterns that suggest manipulation or abuse. Wash-trade detection looks for offsetting trades between accounts under common control. Spoofing and layering alerts identify rapidly placed and cancelled orders that create a false impression of liquidity. Cancel-rate monitoring flags accounts whose cancellation rate is far above the market norm. Detection uses interpretable models. Hidden Markov models, for example, can identify sequences of rapid placement and cancellation while remaining suitable for compliance review [12]. The output is an alert, not an automatic penalty. A human reviewer examines the evidence and decides whether to open a case. This preserves due process and avoids false positives.

8.2 Emergency controls and the market-rule handbook

Emergency controls allow the operator to intervene when the market is disrupted. An emergency pause stops new risk while preserving the event record. Existing positions remain open, but no new orders are accepted. A correction event can amend a ledger entry if a technical error is identified; the correction references the original event and states the reason. A delisting event closes an instrument under published rules. Emergency actions are logged and reviewed. The operator cannot use them to silently change a position or erase a trade. Every action is an event with a timestamp and a rationale.

The market-rule handbook states which events trigger a pause, a correction, or a delisting, and how participants will be notified. It covers who may list an instrument, how fees and rebates are calculated, how disputes are opened and resolved, what conduct is prohibited, and how rule changes are announced and versioned. The handbook is part of the contract specification, not a separate policy document. It is written in plain language and in machine-readable form. The plain-language version is for participants; the machine-readable version is for integration partners and auditors. Both versions are versioned together. A participant who reads the handbook knows exactly what to expect when the market is under stress.

9 User Journey and Governance

A participant's experience of LEVYX is shaped by two things: the sequence of steps they follow from account opening to close-out, and the process by which the rules they rely on can change. This section describes both.

9.1 Account opening, trading, and close-out

A participant begins by creating an account and completing required identity or organization checks. The checks are jurisdiction-specific and are described in the compliance section. Once the account is approved, the participant deposits an accepted collateral asset. The deposit is credited to the account and appears in the ledger as a distinct event.

The participant then opens the instrument page and reads its one-page specification. The specification states the underlying, the size, the quote, the schedule of events, the settlement rule, the margin requirements, the funding interval, and the finality horizons. A preview tool shows quantity, price, dates, fees, margin, and worst-case liquidation behavior before the order is placed. The preview is not a guarantee, but it is a faithful calculation under the published rules.

When the participant places an order, the matching engine applies the published priority rule and returns a confirmation with a unique trade ID. The trade event records the maker, the taker, the timestamp, the fee, and the resulting position change. The participant can monitor mark price, margin level, funding schedule, and settlement status through the interface or the API. The interface uses direct warnings. For example: "Your position can be liquidated if equity falls below 30% of required initial margin." Risks are stated plainly, in the same register as the trade confirmation itself. If the account enters the liquidation sequence, the participant receives a warning and a request for additional collateral, if the instrument allows a grace window. Each step in the sequence is visible in advance.

A position can be closed by the participant, or it can be liquidated under the published sequence. When it closes, the difference between entry and exit is settled in the quote currency. The participant can download a statement that explains every balance change: deposits, withdrawals, fees, funding payments, realized profit and loss, and settlement amounts. The statement is generated from the event log, so it is reconstructable and auditable. If a dispute arises, the participant can open a case under the published dispute process. The case references specific events in the ledger. The operator responds within a defined window. If the dispute cannot be resolved, it proceeds to the bounded dispute protocol described in the settlement section. The participant does not need specialist knowledge to follow the process; the handbook explains each step.

9.2 Governance and change management

Rules change because markets, laws, and technology change. LEVYX therefore versions every instrument specification and every risk parameter. A change record states what changed, why, when it takes effect, and which open positions it affects. The old version remains available for historical statements. For material changes, the minimum process is: publish a plain-language proposal; publish the exact machine-readable change; allow a defined review period; announce the activation time in UTC and the relevant local time;

and keep the old version available. Governance does not remove operator responsibility. A vote can choose a bad rule. The system needs accountable people, documented authority, and an incident process.

10 Composability, Compliance, and Operating Model

LEVYX is designed to be legible to other systems. Its positions settle on-chain, its specifications are machine-readable, and its risk rules are published. This section describes how LEVYX composes with on-chain systems, how it meets compliance obligations, and how it is operated.

10.1 Composability with on-chain systems

Because LEVYX positions settle on Arbitrum and follow a published specification, they can serve as inputs to other on-chain systems. Three integration shapes are worth naming explicitly: collateral, hedging, and structured products.

A LEVYX position with a known mark price, a known maintenance requirement, and a defined liquidation sequence is legible to a lending market. A lending protocol that accepts the position as collateral must account for LEVYX's finality horizon and its liquidation rules; if it does, it can price the collateral more precisely than it could price a generic token. A LEVYX perpetual is a hedge against a spot exposure in the same underlying. A stablecoin issuer holding BTC reserves, or a treasury holding ETH, can offset price risk by opening the opposite position on LEVYX, with the mark price and settlement rules published in advance. Because the specification format is machine-readable, a LEVYX position can serve as a leg in a structured product — a capped note, a range accrual, a principal-protected structure. The receiving protocol reads the specification, prices the leg, and includes it in the product's payoff. Composability is not automatic. Each integration requires the receiving protocol to accept LEVYX settlement finality, margin accounting, and event semantics.

For a protocol considering integration with LEVYX, the minimum checklist is: confirm the finality horizon acceptable for the use case; confirm the margin accounting and liquidation sequence are understood; confirm the event semantics are compatible with the receiving system; and confirm the reference-data sources meet the receiving system's requirements. LEVYX publishes an integration guide with sample code, a testnet deployment, and a reference implementation of a lending market that accepts a LEVYX position as collateral. The guide is versioned alongside the contract specification. Integration partners are encouraged to test against the testnet before deploying to mainnet.

10.2 Compliance and operating model

The required legal model depends on the asset, user type, country, custody model, and settlement method. Before live operation, LEVYX obtains jurisdiction-specific advice and implements, at minimum, identity and sanctions screening, transaction monitoring for abnormal patterns, record retention consistent with the trading ledger, privacy controls for personal data, complaints handling, and a process for lawful requests. Personal data is minimized. The trading ledger retains what is needed to prove a trade and meet legal duties;

sensitive identity documents are stored in a separate protected service with restricted access. The zero-knowledge solvency attestation is designed so that aggregate solvency can be published without publishing individual account data. Periodic independent review of the risk and compliance controls is required.

LEVYX operates with separated duties: the trading path, the risk path, the settlement path, and the compliance path are run by distinct teams with distinct access. No single operator can both change a position and publish the resulting settlement. Rule changes are versioned and announced. Incidents are logged and reviewed. The operating model is documented in a runbook that states who has authority to take each action, what approval is required, and how the action is recorded. The runbook is tested through regular drills. A key principle is that no emergency action can erase the event record. Even a pause or a correction leaves a permanent trace.

11 Capability Roadmap

LEVYX acquires capability in stages. Each stage is complete when the capability is live and measurable, not when it is announced. This section describes the stages, the metrics that define completion, and the reporting that accompanies each stage.

11.1 Staged acquisition

The first stage is the contract registry: instrument specifications, versioning, and the machine-readable schema that every other component consumes. The second stage is order and match: order entry, matching under published priority rules, and the trade events that follow. The third stage is the ledger: the append-only event log and the reconstruction tools that make it useful. The fourth stage is the risk engine: WASM margin computation, manipulation-resistant mark-price construction, funding, and the liquidation sequence. The fifth stage is on-chain settlement: settlement assertions, collateral custody, and the bounded dispute protocol. The sixth stage is zk attestation: the solvency circuit, the on-chain verifier, and the published attestation schedule. The seventh stage is market integrity: detection, prevention, and emergency controls, in operation. The eighth stage is compliance: jurisdiction-specific controls, monitoring, and reporting, in operation. New contract families are added only when the base capabilities above are live and measurable.

Capability	What is being acquired
Contract registry	Instrument specifications, versioning, and the machine-readable schema that every other component consumes.
Order and match	Order entry, matching under published priority rules, and the trade events that follow.
Ledger	The append-only event log and the reconstruction tools that make it useful.
Risk engine	WASM margin computation, manipulation-resistant mark-price construction, funding, and the liquidation sequence.
On-chain settlement	Settlement assertions, collateral custody, and the bounded dispute protocol.
Zk attestation	The solvency circuit, the on-chain verifier, and the published attestation schedule.
Market integrity	Detection, prevention, and emergency controls, in operation.
Compliance	Jurisdiction-specific controls, monitoring, and reporting, in operation.
Instruments	New contract families are added only when the base capabilities above are live and measurable.

11.2 Metrics and reporting

Metrics for each capability are published alongside the capability itself. Order-to-trade latency, uptime, rejected-order rate, margin-call response time, settlement completion time, data freshness, and incident recovery time are measured continuously and reported on a schedule. The reporting schedule is part of the governance record. A capability that cannot be measured cannot be declared complete. When a metric falls below its published threshold, the capability is treated as degraded, and the degradation is reported alongside the metric. This prevents a capability from being declared complete once and then silently decaying.

12 Conclusion

LEVYX is a proposal for making perpetual and time-based crypto markets understandable and enforceable. Its main product is not a chart or a token. Its main product is a shared contract record that says what was agreed, when each event occurs, how risk is measured, and how the final result is calculated.

12.1 What Levyx is

LEVYX is a venue and settlement layer for time-based contracts, built on Arbitrum. It provides a transparent order and position system, explicit collateral rules, a published funding mechanism, a manipulation-resistant mark price, a deterministic liquidation sequence, and on-chain settlement with a bounded dispute protocol. It also publishes a zero-knowledge solvency attestation, so that aggregate solvency can be verified without exposing individual accounts. The platform is designed around five principles: continuous exposure without expiry confusion; collateral before leverage; funding that keeps the

market anchored; liquidation as a controlled sequence rather than a single event; and fast trading with auditable settlement. Each principle maps to a concrete mechanism described in this paper, and each mechanism is reconstructable from the event record.

12.2 What Levyx is not, and the path forward

LEVYX is not a delivery venue. It does not handle physical settlement, delivery locations, or inspection. Its obligations are entirely financial. It is not a guarantee of liquidity; liquidity depends on market makers and on the broader market environment. It is not a substitute for audited financial statements; the zero-knowledge attestation is a frequent cryptographic check between audits, not a replacement for them. LEVYX is also not a promise that liquidation will occur at a good price. In a fast market, the closing price can be worse than the mark price that triggered the liquidation. The platform publishes the sequence and the waterfall, but it cannot guarantee the outcome. Participants who understand this are better prepared to manage their risk.

The strongest first version is deliberately narrow: a small set of well-defined instruments, conservative leverage, reliable reference data, clear settlement, and complete audit records. From those foundations, LEVYX expands to more assets and more timing models. The roadmap in Section 11 states the sequence of capabilities and the metrics that define completion. New instruments are added only when the base capabilities are live and measurable. The platform's growth is therefore constrained by its own published standards, not by the speed at which new products can be announced.

Glossary

This appendix defines the terms used throughout the paper. Where a term has a specific meaning in the LEVYX specification that differs from common usage, the LEVYX meaning is given.

BoLD	Bounded Liquidity Delay, Arbitrum's permissionless dispute protocol for state assertions.
CLOB	Central limit order book.
Funding	The periodic transfer between longs and shorts that anchors a perpetual to spot.
Groth16	A pairing-based succinct non-interactive argument of knowledge with constant-size proofs.
Initial margin	Collateral required to open a position.
Maintenance margin	Minimum collateral required to keep a position open.
Mark price	The price used for risk calculations, distinct from the last traded price.
Merkle root	A single hash committing to a set of data, used here to commit to account state.
Orbit	Arbitrum's framework for deploying application-specific chains that settle to Arbitrum One.
Soft finality	A fast, sequencer-level confirmation that may still be reverted in principle.
Settlement finality	A state assertion that has passed the dispute window and is anchored on L1.
Stylus	Arbitrum's WASM execution environment that runs alongside the EVM.
TWAP	Time-weighted average price.
zk-SNARK	A succinct zero-knowledge proof; used here only for aggregate solvency attestations.

References

- [1] BitMEX. *Perpetual Contracts Guide*. 2016. URL: <https://www.bitmex.com/app/perpetualContractsGuide>.
- [2] dYdX Foundation. *dYdX: A Decentralized Derivatives Exchange*. 2024. URL: <https://dydx.exchange/>.
- [3] GMX. *GMX: Decentralized Perpetual Exchange*. 2024. URL: <https://gmx.io/>.
- [4] Aave Companies. *GHO: A Decentralized Stablecoin*. 2023. URL: <https://docs.gho.xyz/>.
- [5] Bank for International Settlements. *Liquidity Provision in Decentralized Exchanges*. Tech. rep. BIS Working Papers, 2025. URL: <https://www.bis.org/publ/work1227.pdf>.
- [6] Various. “Tick-Based Soft Liquidation for Lending Protocols with LP Tokens as Collateral”. In: *ScienceDirect* (2026). URL: <https://www.sciencedirect.com/science/article/pii/S2096720926000989>.
- [7] Youwei Deng, Majid Khabbazzian, and Jeremy Clark. *Xiezhi: Toward Succinct Proofs of Solvency*. 2024. Cryptology ePrint Archive: 2024/2001. URL: <https://eprint.iacr.org/2024/2001>.
- [8] Seny Kamara, Tarik Moataz, et al. “Privacy-Preserving Audits of Bitcoin Exchanges”. In: *Cryptology ePrint Archive*. 2018. URL: <https://www.iacr.org/archive/crypto2018/10993251/10993251.pdf>.
- [9] Dongbin Bai et al. “Ormer: A Manipulation-resistant and Gas-efficient Blockchain Pricing Oracle for DeFi”. In: *arXiv preprint arXiv:2410.07893* (2024). URL: <https://arxiv.org/abs/2410.07893>.
- [10] IEEE. “Large-Scale Security Analysis of Multi-Token Smart Contracts”. In: *2026 56th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2026. URL: <https://www.computer.org/csdl/proceedings-article/dsn/2026/514900a220/2hXbLH6WvgQ>.
- [11] Jens Groth. “On the Size of Pairing-Based Non-interactive Arguments”. In: *Advances in Cryptology – EUROCRYPT 2016*. 2016.
- [12] IEEE. “Manipulation Detection of the Bank Sector”. In: *2025 IEEE International Conference*. 2025. URL: <https://ieeexplore.ieee.org/document/11208483>.