

Lattica[‡]

Protocol v1.0

William Flanders^{*‡} Stephen Flanders[‡]

March 2026

Abstract

In this paper, we introduce a decentralized protocol: Lattica. Serving as the liquidity layer of informational finance, Lattica enables fixed-duration epoch loans for prediction markets, allowing position holders to collateralize and borrow liquidity against their ERC1155 outcome tokens. Instead of relying solely on liquidations, Lattica's calibrated risk engine, WARHORSE, protects lenders by calculating and charging a state-conditional premium to the borrower at loan origination.

*william@lattica.fi

†stephen@lattica.fi

‡<https://lattica.fi>

1 Introduction

1.1 Motivation

By allowing users to put their money where their mouth is, prediction markets produce price signals that aggregate dispersed beliefs into a common forecast (Hanson, 2003; Arrow et al., 2008). This informational function is already being incorporated into political science and polling theory (Clinton and Huang, 2025), economics¹, and media². We expect their influence to deepen as these platforms expand into more specialized subjects and broader cultural domains.

Yet prediction markets today operate under a structural constraint that limits their potential. Market participation requires traders to lock capital into contingent claims with capped upside and no interim yield. For markets with resolution horizons spanning weeks, months, or years, this imposes significant opportunity costs. The result is thin books, wide spreads, and shallow depth, particularly in the longer-dated markets where forecasting value is arguably greatest.

The platforms themselves are clearly aware of this problem, and they have attempted a range of solutions: paying position holders and market makers “holding” and “liquidity” rewards from treasuries, contracting external market makers, and running internal market making strategies. These programs are uniformly capital intensive. Holder and liquidity rewards require treasury funds, external market makers require rebates, and internal strategies require the platform to absorb losses. More importantly, they treat the symptom rather than the cause. Locked capital remains inefficient, reward rates are set without rigorous connection to market quality, and prediction markets continue to suffer from illiquidity.

A protocol that lets traders borrow against locked positions would address the cause directly, and the case for building one extends beyond simple capital efficiency.

Forecast accuracy does not scale with raw volume; it scales with the presence and capital depth of informed traders³. When those traders’ capital is trapped in existing positions, they cannot redeploy it to correct mispricings elsewhere. Freeing that capital to circulate across markets increases the effective reach of the informed flow that actually drives price discovery (Wolfers and Zitzewitz, 2004).

The benefits extend to retail participants as well. Della Vedova (2026) show that retail traders on Polymarket who correctly forecast outcomes still lose money because they arrive

¹<https://ir.theice.com/press/news-details/2025/ICE-Announces-Strategic-Investment-in-Polymarket/default.aspx>.

²<https://finance.yahoo.com/news/kalshi-becomes-cnn-prediction-market-200713149.html>.

³<https://fiftycentdollars.substack.com/p/minimum-viable-liquidity>.

late to thin books and pay unfavorable prices. A protocol that recycles locked capital back into the ecosystem deepens those books at every price level, directly improving execution quality for the participants who need it most.

Finally, prediction markets today cannot support the derivative products, structured hedges, and secondary instruments that institutional participants expect from any serious asset class. Borrowable positions are a prerequisite for nearly all of these. Without a credit layer, prediction markets remain a spot-only venue, and the composability required for institutional adoption cannot develop. Building that layer is therefore not an incremental improvement to the existing ecosystem; it is infrastructure that makes an entirely new class of financial products possible.

1.2 Introducing Lattica

Lattica is a protocol designed to serve as that infrastructure. Its purpose is not merely to enable lending in isolation, but to provide the credit primitives on which a composable prediction market ecosystem can be built.

The core technical challenge is underwriting. Prediction market positions are not ordinary collateral. They are anchored by information rather than fundamentals, prices adjust discontinuously, and given a large enough movement, single-sided liquidity can suddenly evaporate as the market searches for a new equilibrium. Standard lending protocols that rely on liquidating underwater collateral in time cannot guarantee solvency under these conditions. Following our work in Flanders and Flanders (2026), where we design and calibrate a risk engine capable of solvent underwriting against this collateral class, this paper presents Lattica, the protocol built around that engine.

Section 2 describes Lattica’s architecture, including the first deployment of the calibrated risk engine, the Weighted Adaptive Risk Engine for Hyperjump-Optimized Rating & Stress Events (WARHORSE). Section 3 presents the protocol’s smart contract design.

2 Protocol Architecture

Lattica’s architecture is inspired by catastrophe reinsurance rather than traditional DeFi lending. In reinsurance, investors do not select which peril zones to underwrite; they buy equity in the book and internal underwriting manages the portfolio. Lattica follows the same principle. Lenders deposit USDC into a single pool. Borrowers post prediction market positions, ERC-1155 tokens, as collateral and receive an all-in quote consisting of an upfront risk premium and a fixed interest rate for the epoch. Between the two sits an internal risk

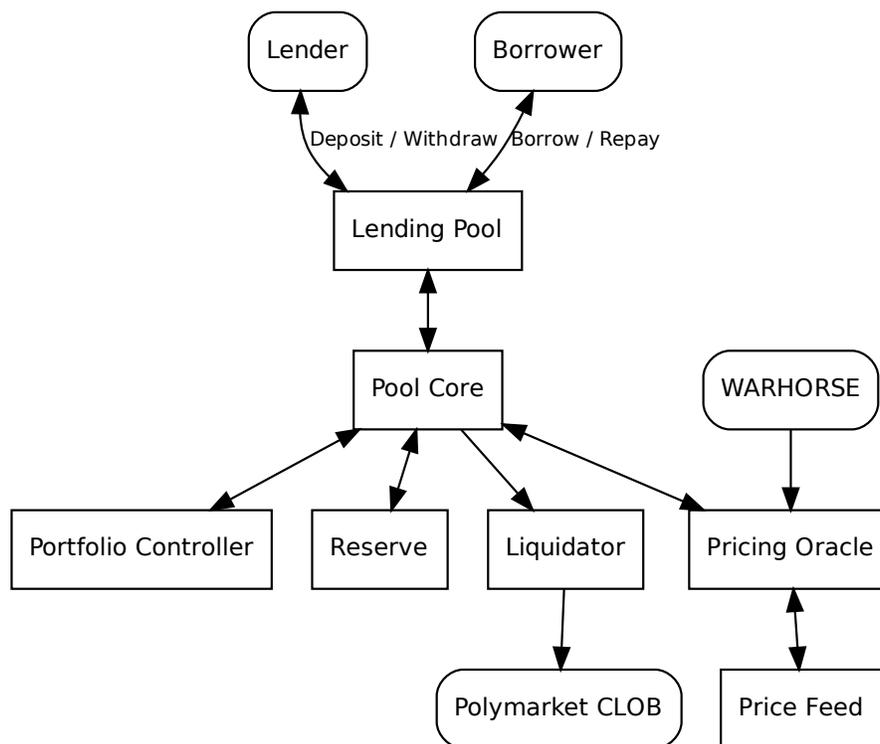


Figure 1: Lending pool architecture overview.

layer that manages portfolio exposure. Neither side sees the machinery.

2.1 Lending Lifecycle

Lattica operates on fixed-duration epochs. Borrowers request loans by selecting a collateral position, a loan amount, and an epoch length. The protocol returns a quote: a risk premium priced by WARHORSE and a fixed interest rate for the epoch. If accepted, collateral transfers into the pool, the premium and interest are deducted, and net USDC disburses to the borrower. At epoch end, the borrower repays principal and recovers their collateral, or walks away.

All loans are overcollateralized. If the value of the posted collateral falls below the maintenance threshold during the epoch, the protocol liquidates the position to protect lender capital. This is distinct from the walk-away case, which is a borrower’s choice at epoch end when repayment is uneconomical. In both cases, the collateral is sold on the underlying platform’s orderbook. Premiums are priced assuming zero recovery, so any USDC recovered

from either path flows to the pool as surplus.

Healthy loans may roll into the next epoch at a freshly quoted premium and rate reflecting updated market conditions.

All loans terminate before the underlying market resolves. Volatility spikes near resolution as information arrives in bursts, making the final hours of a market’s life structurally different from the rest. Handling resolution is a separate problem from ensuring loans remain healthy over their term, and conflating the two would force the pricing engine to account for dynamics it is not designed to capture. Fixed-duration epochs let the protocol cleanly separate these concerns: WARHORSE prices drawdown risk over a bounded horizon, and loan duration never overlaps with the resolution window.

2.2 Pool Design

The pool is denominated in USDC. Lenders deposit and receive share tokens representing pro-rata claims on pool capital. At the end of each epoch, net income (interest collected plus premiums minus realized losses plus any liquidation recovery) accrues to the pool, and share values update accordingly.

A single-pool design maximizes available liquidity and avoids the fragmentation that arises from category-specific or market-specific vaults. Risk segmentation is handled entirely through internal pricing and the Portfolio Controller rather than through pool structure. This means lenders face one decision: how much USDC to deposit. All portfolio construction complexity is absorbed by the protocol.

2.3 WARHORSE

WARHORSE (Weighted Adaptive Risk Engine for Hyperjump-Optimized Rating & Stress Events) is Lattica’s pricing engine and the first production deployment of the risk engine developed in Flanders and Flanders (2026). Given an origination state vector containing the current price, epoch length, time to resolution, market category, and microstructure features, WARHORSE returns a risk premium quote for the loan. This premium covers the expected shortfall over the epoch and is separate from the interest rate, which compensates lenders for the time value of their capital.

Premiums are priced assuming zero liquidation recovery. The full expected shortfall is baked into the upfront charge, and any proceeds from selling liquidated or walk-away collateral flow back to the pool as surplus. Risk engine parameters can be adjusted at the protocol level, allowing the platform to dial risk appetite up or down without retraining

the underlying model. Quotes are generated off-chain with an expiry window to prevent execution against stale pricing.

2.4 Portfolio Controller

WARHORSE prices idiosyncratic loan risk. The Portfolio Controller controls portfolio risk. These are separate systems by design: a loan can be correctly priced in isolation and still be dangerous to the book if it concentrates exposure along a correlated dimension.

The Portfolio Controller tracks correlation structure across the book using data-driven clustering of markets by co-movement, updated each epoch. It enforces concentration caps across multiple overlapping views: condition, event, theme, resolution window, and liquidity regime. Every loan must pass checks against all relevant views, not just its assigned cluster. It manages capital budgets per cluster, sized so that a complete cluster blowup is absorbable by reserves and premium income from other clusters. Diversification credit is partial, state-dependent, and revocable. It also monitors calibration by comparing realized losses to predicted losses at both pool and cluster levels after each epoch settles, feeding that information back into the protocol's risk controls.

The precise mechanism by which these signals translate into lending decisions, whether through pricing adjustments, origination gates, or some combination, is a design parameter that will be refined as the protocol matures. What is fixed is the principle: the Portfolio Controller exists to ensure that portfolio-level risk remains bounded despite the performance of individual loans.

2.5 Reserve System

The reserve is a retained premium buffer that absorbs losses exceeding premium income before any lender capital is impaired. At the end of every epoch, a percentage of net premium income is retained in the reserve. The target reserve level is sized as a multiple of the estimated portfolio tail loss ratio, explicitly accounting for correlation and aggregation misspecification.

The retention rate is dynamic. When the reserve is below target, the protocol enters rebuilding mode: retention rates increase and cluster budgets tighten. When the reserve is at target, normal retention applies.

The loss absorption waterfall is therefore: premium income absorbs expected losses by design, the reserve absorbs excess losses, and lender capital takes residual hits only when the reserve is exhausted. Critically, the reserve does not rely on liquidation recovery. In stress states, the protocol expects more walk-aways, worse marks, thinner books, and weaker

recovery simultaneously. Liquidation impairment is itself correlated with portfolio stress, so the conservative design prices assuming zero recovery and treats all disposal proceeds as surplus.

2.6 Liquidations

When a borrower's collateral falls below the maintenance threshold during an epoch, the protocol liquidates the position to protect lender capital. When a borrower walks away at epoch end, the protocol retains the collateral. In both cases, premiums have already been priced assuming zero recovery. The position is, from an accounting perspective, fully written off at the moment liquidation triggers or the borrower fails to repay.

That said, the protocol still attempts to recover value. Collateral is immediately placed as sell orders on the underlying platform at market prices. There is no staged unwinding, no reserve price, and no attempt to time the sale. The position is liquidated in full as aggressively as the orderbook can absorb. Whatever USDC comes back flows to the pool pro-rata as bonus return. Because the premium already covered the expected loss, every dollar recovered is pure surplus that improves lender yield above the underwritten baseline.

The reason recovery is not relied upon in the pricing model is that the states in which liquidations and walk-aways occur are precisely the states in which recovery is worst. A position that has lost enough value to breach maintenance or make walking away rational is, almost by definition, on the wrong side of the market. The orderbook for that side will be thin, spreads will be wide, and depth will be shallow. Pricing as though recovery is zero and treating any proceeds as a windfall is the only honest accounting for this correlation between default and illiquidity.

3 Smart Contracts

The contract stack follows a separation between an entry point that users interact with, a logic layer that executes operations, and peripheral contracts that provide data and enforce constraints. This mirrors the architecture of established lending protocols, where the user-facing surface is thin and the complexity lives beneath it.

3.1 Lending Pool

The Lending Pool is the single entry point for both lenders and borrowers. Lenders call deposit and withdraw. Borrowers call borrow and repay. Every user action routes through

this contract. It does not hold business logic itself; it validates inputs, checks that the caller is authorized, and delegates execution to the Pool Core.

3.2 Pool Core

Pool Core contains the lending logic. When a borrow request arrives from the Lending Pool, Core orchestrates the full sequence: it queries the Pricing Oracle to verify the WARHORSE quote and confirm it has not expired, checks the Portfolio Controller for capacity approval, transfers collateral into escrow, deducts the premium and interest, updates loan state, and disburses USDC to the borrower. Repayment, liquidation, and epoch settlement logic also live here. Core reads and writes all protocol state: loan records, collateral positions, share balances, reserve levels, and aggregate accounting.

3.3 Pricing Oracle

The Pricing Oracle is the on-chain interface to WARHORSE. It accepts signed premium quotes generated off-chain by the risk engine, verifies the signer and checks the deadline, and returns the verified premium to Pool Core. Quotes that have expired or carry an invalid signature revert. This keeps the computationally expensive model inference off-chain while ensuring that every premium used in origination is cryptographically authenticated and time-bound.

3.4 Portfolio Controller

The Portfolio Controller enforces portfolio-level risk constraints on-chain. When Pool Core checks capacity for a new loan, the Portfolio Controller evaluates the request against concentration caps (per condition, per event, per resolution window, per liquidity regime), cluster capital budgets, and any active circuit breakers. If any check fails, origination reverts. The controller also stores calibration state: realized-versus-predicted loss ratios at pool and cluster levels, updated at epoch settlement and readable by the circuit breaker logic.

3.5 Market Registry

The Market Registry stores parameters for each onboarded prediction market: collateral factor, maximum exposure cap, resolution time, origination cutoff, and active/paused status. Pool Core reads from the registry at origination to confirm the market is eligible, the cutoff has not passed, and the exposure cap will not be breached. Markets are onboarded and configured by the protocol admin.

3.6 Price Feed

Each supported market has a Price Feed contract that receives price updates from an authorized off-chain service. The feed stores the current price, the previous price, and timestamps for both. It enforces a minimum deviation threshold to filter noise and a circuit breaker that trips when a single update exceeds a configured threshold, pausing the feed for a cooldown period. Pool Core and the Liquidator read from the feed to value collateral.

3.7 Liquidator

The Liquidator executes forced sales of collateral. It is triggered in two cases: mid-epoch when a borrower's health factor falls below the maintenance threshold, and at epoch end when a borrower has not repaid. In both cases, the Liquidator seizes the ERC-1155 collateral tokens from the pool, places aggressive sell orders on the prediction market's orderbook, and routes any USDC proceeds back to the pool. Since premiums are priced assuming zero recovery, the Liquidator has no solvency responsibility. It is a best-effort recovery mechanism.

3.8 Reserve

The Reserve contract holds retained premium income and enforces the loss absorption waterfall. At epoch settlement, Pool Core routes a configurable percentage of net premium income into the Reserve. When epoch losses exceed premium income, the Reserve covers the shortfall before lender capital is touched. Reserve health is readable on-chain and feeds into the Portfolio Controller's circuit breaker logic: when the reserve falls below threshold levels, origination constraints tighten automatically.

3.9 Gasless Interaction

Polymarket users hold positions inside Gnosis Safe proxy wallets on Polygon. These wallets contain USDC and conditional tokens but no native gas token. Lattica's relay service solves this by exploiting a property of the Safe architecture: the `execTransaction` function verifies that valid owner signatures are present in the calldata but does not require the caller to be the owner. The user signs a Safe transaction off-chain, and the relay submits it on-chain from a funded hot wallet. The user experience reduces to two signature popups across the entire session: one for authentication and one for the deposit transaction. Neither costs gas.

The relay validates all inner calldata server-side before submission. Only whitelisted function selectors on whitelisted contract addresses are relayed, preventing the service from being used to execute arbitrary transactions through users' wallets.

References

- Arrow, Kenneth J., Robert Forsythe, Michael Gorham, Robert Hahn, Robin Hanson, John O. Ledyard, Saul Levmore, Robert Litan, Paul Milgrom, Forrest D. Nelson, George R. Neumann, Marco Ottaviani, Thomas C. Schelling, Robert J. Shiller, Vernon L. Smith, Erik Snowberg, Cass R. Sunstein, Paul C. Tetlock, Philip E. Tetlock, Hal R. Varian, Justin Wolfers, and Eric Zitzewitz, “The Promise of Prediction Markets,” *Science*, May 2008, *320* (5878), 877–878.
- Clinton, Joshua D and TzuFeng Huang, “Prediction Markets? The Accuracy and Efficiency of \$2.4 Billion in the 2024 Presidential Election,” December 2025.
- Flanders, William and Stephen Flanders, “Unlocking Liquidity on Prediction Markets,” February 2026.
- Hanson, Robin, “Combinatorial Information Market Design,” *Information Systems Frontiers*, January 2003, *5* (1), 107–119.
- Vedova, Joshua Della, “Who Profits from Prediction Markets? Execution, Not Information,” February 2026.
- Wolfers, Justin and Eric Zitzewitz, “Prediction Markets,” *Journal of Economic Perspectives*, June 2004, *18* (2), 107–126.