

PROBIO MARKETS

An execution layer for prediction markets

Leverage, multi leg parlays and routed execution on top of live Polymarket order books, settled in ETH on Robinhood Chain.

Whitepaper, version 1.0

September 2026

probiomarkets.com

Contents

- 1 Summary
- 2 Why we built it
- 3 Notation and position mechanics
- 4 The fee schedule, and why it is charged on margin
- 5 Order routing
- 6 Execution scheduling
- 7 Fill probability and the passive versus aggressive decision
- 8 Parlays
- 9 Settlement, custody and the network
- 10 Market data and caching

1 Summary

Probio is a trading terminal for prediction markets. It reads roughly seven thousand live markets from Polymarket, through the Gamma and CLOB APIs, and puts an execution layer on top of them: leverage of up to ten times, multi leg parlays of up to ten outcomes, portfolio tracking with live profit and loss, a liquidity pool, and a full paper trading engine. Positions settle in ETH on Robinhood Chain, an Ethereum layer two built on the Arbitrum Orbit stack.

Nothing on the terminal is invented. Probio does not create markets, does not quote synthetic prices and does not fabricate volume. Every question, every price and every book on the screen belongs to a real Polymarket market, fetched live. What Probio adds is the part prediction markets never built, which is the machinery a trader actually works in: depth aware order construction, an itemised fee quote before commit, defined maximum loss, position level risk, and a way to express a thesis that spans more than one event without funding it three times over.

The terminal is fully usable without an account. Market data, charts, the parlay builder and paper trading all work while signed out. Signing in derives a persistent wallet address and syncs a portfolio across devices, and that is the whole of it. Nothing is gated behind the login wall.

Sections 1 and 2 are what the product is and why it exists. Everything after that is the mathematics: leverage and liquidation, fee incidence, routing, scheduling, fill probability and parlay pricing. If you only want to know whether the execution is serious, skip to **Section 5**.

2 Why we built it

Prediction markets solved the hard part and then stopped. Price is a probability. A contract at 0.34 is the market saying thirty four percent, and it settles at one or zero against a published rule with a named data source. There is no house line, no vig baked into an opaque number, no argument at the end about what counts. That is a genuinely good financial primitive, better than the one sportsbooks offer, and it has been sitting underused for a decade.

What never got built on top of it is everything a trader takes for granted everywhere else. Four things in particular kept bothering us.

2.1 Every dollar of exposure costs a dollar

On a standard prediction market, exposure is funded one for one. If you want ten thousand dollars of exposure to a contract at 0.34, you post ten thousand dollars. In futures, in FX, in equities, in every other market where people express directional views, a view like that is expressed on margin. Prediction markets are the only place left where conviction is taxed by the size of your bank account rather than the quality of your read.

Probio charges margin. Up to ten times, with maximum loss fixed at the moment the ticket is confirmed. There is no margin call, no partial liquidation, no cascade. You cannot lose more than you posted, and you cannot be shaken out of a position by a wick that never gets close to your liquidation price.

2.2 A thesis is rarely one event

Real views are conditional and they are plural. Inflation prints above consensus, the Fed holds, risk assets rally. That is one idea. On a conventional prediction market it is three tickets, funded separately, closed separately, each with its own slippage and its own reconciliation. The correlation between them, which is the whole point of the trade, is not expressed anywhere.

The parlay builder combines up to ten outcomes into a single position with one maximum loss and one payout. Probabilities multiply, so payouts climb fast and hit rates fall just as fast. Four legs at fifty percent each is a 6.25 percent chance of paying sixteen times. We show both numbers as you add legs, because the second one is the one people forget.

2.3 A Buy button is not execution

Retail prediction market interfaces show one price and one button. They do not show you the book. They do not split your order across levels. They do not tell you what your own size did to your fill. Ten thousand dollars into a thin book at 0.34 does not fill at 0.34, and on most frontends you find that out afterwards.

Probio quotes against live book depth, walks the levels, and itemises every fee before you commit. Sections 5 through 7 are the routing, scheduling and fill probability models behind that quote, written out rather than gestured at, because we think the arithmetic is the product.

2.4 Fees should be legible

Two components, both charged on the margin you post rather than the notional you control. A ten times position controls notional it has rented for the duration of the trade; charging the full fee on that notional means paying ten times for the same risk, and it quietly punishes exactly the users the leverage exists for. The full incidence argument, with the breakeven arithmetic, is in Section 4. The short version is that the fee schedule fits in three lines and you can compute your own cost without a calculator.

3 Notation and position mechanics

A prediction market contract is the simplest instrument in finance. It settles at one if the event happens and zero if it does not, so its price is its probability. Everything below follows from that single fact, and most of the apparatus that makes derivatives pricing hard, discounting, dividends, unbounded upside, disappears.

Symbol	Meaning
p	Contract price, equal to the implied probability, in the open interval from 0 to 1
M	Margin posted, in ETH
L	Leverage, an integer from 1 to 10
$N = LM$	Notional controlled
$q = N / p$	Number of contracts held
F	Total fee charged at entry
σ	Instantaneous volatility of the contract price
Q	Order size in contracts
dk, pk	Depth and price at book level k

Table 1. Notation used throughout.

3.1 Position value and return

Posting margin M at leverage L buys q contracts at entry price p , where the notional is $N = LM$ and $q = N / p$. Equity at any later price is the margin plus the mark to market:

$$E(p_t) = M + q(p_t - p_0)$$

Return on margin is the price return geared by L , which is the usual result and the reason leverage exists:

$$R = [E(p_t) - M] / M = L \cdot (p_t - p_0) / p_0$$

Unlike a perpetual future, the upside here is bounded, because the price cannot exceed one. The maximum return on margin for a long position is fixed at entry:

$$R_{max} = L \cdot (1 - p_0) / p_0$$

This is worth pausing on. At ten times leverage a contract bought at 0.34 can return at most 19.4 times the margin, and a contract bought at 0.90 can return at most 1.1 times. The cheap tail is where geared prediction market positions pay, and the ceiling is knowable before you click.

3.2 Liquidation

Equity reaches zero when the mark to market loss equals the margin, so setting the equity expression to zero and substituting $q = LM / p$ gives a liquidation price that depends on nothing but the entry price and the leverage:

$$p_{liq} = p_0 (1 - 1/L)$$

The distance to liquidation in probability points is p divided by L , and in relative terms it is exactly $1 / L$ regardless of where the contract trades. There are no margin calls and no partial liquidations. Once the ticket is confirmed the maximum loss is locked, and the liquidity pool takes the other side of the liquidation.

Leverage	Liquidation at $p = 0.10$	Liquidation at $p = 0.34$	Liquidation at $p = 0.90$	Relative buffer
1x	0.000	0.000	0.000	100%
2x	0.050	0.170	0.450	50%
5x	0.080	0.272	0.720	20%

Leverage	Liquidation at p = 0.10	Liquidation at p = 0.34	Liquidation at p = 0.90	Relative buffer
7x	0.086	0.291	0.771	14.3%
10x	0.090	0.306	0.810	10%

Table 2. Liquidation price by leverage and entry. At one times leverage the position only dies at settlement.

4 The fee schedule, and why it is charged on margin

Two proportional components and one fixed one:

entry fee	1.0% of margin, flat
leverage fee	0.2% of margin per 1x of leverage, linear
network fee	0.0005 ETH, fixed

$$F(M, L) = 0.010 M + 0.002 L M + \varphi, \quad \varphi = 0.0005 \text{ ETH}$$

The leverage component is linear, not exponential. Each additional turn of leverage adds a constant twenty basis points of margin, so the rate on margin is an affine function of L:

$$r_M = 0.010 + 0.002 L$$

Expressed against the notional the position actually controls, the same schedule is decreasing in leverage:

$$r_N = (0.010 + 0.002 L) / L = 0.010 / L + 0.002$$

4.1 Why not charge the notional

Because the notional on a geared position is capital the trader has rented for the life of the trade, not capital they have committed. A trader posting one hundred ETH at ten times and a trader posting one thousand ETH unlevered carry very different risk and very different claims on the pool, and charging both the same percentage of notional would mean the first pays ten times as much fee per unit of capital at risk. A flat notional fee of 1.2 percent, which is what one times leverage costs today, would cost a ten times position twelve percent of its margin at entry. Nobody would use it, which would be a strange outcome for a feature the product is built around.

Leverage	Entry	Leverage fee	Total on margin	Total on notional
1x	1.0%	0.2%	1.2%	1.20%
2x	1.0%	0.4%	1.4%	0.70%
5x	1.0%	1.0%	2.0%	0.40%
7x	1.0%	1.4%	2.4%	0.34%
10x	1.0%	2.0%	3.0%	0.30%

Table 3. The schedule seen two ways. Fees are charged once at entry and are not refunded on close.

4.2 Breakeven, and the trade you are actually making

Set the geared return equal to the fee rate on margin and solve for the price move required to break even. The L cancels in a satisfying way:

$$L \cdot (\Delta p / p) = 0.010 + 0.002 L \implies \Delta p / p = 0.010 / L + 0.002 = r_N$$

The percentage move needed to break even is exactly the fee rate on notional. At one times leverage the contract has to move 1.2 percent in your favour; at ten times, 0.3 percent. Leverage makes you cheaper to be right.

It also makes you easier to be wrong. The relative buffer to liquidation is $1 / L$, so the ratio of room to be wrong against cost to be right is:

$$(1 / L) / r_N = 1 / (0.010 + 0.002 L)$$

That ratio runs from about 83 at one times leverage to about 33 at ten times. The buffer shrinks by a factor of two and a half across the whole range, not by a factor of ten, and that is the direct consequence of charging margin rather than notional. Under a flat 1.2 percent notional fee the same ratio at ten times would be roughly 8. The fee schedule is not cosmetic; it changes the shape of the risk.

5 Order routing

This is the section we care most about. A prediction market book is thin, discrete and event driven, which means the difference between a naive fill and a routed one is large and shows up on every ticket rather than occasionally.

5.1 The book

Polymarket runs a central limit order book per outcome. For a buy, the ask side is a finite ladder of levels indexed $k = 1$ to K , with prices in ascending order and a resting quantity at each:

$$p_1 \leq p_2 \leq \dots \leq p_K \quad d_k > 0$$

Define cumulative depth as the running sum of the ladder, which is the quantity available at or better than level j :

$$D_j = \sum_{k=1}^j d_k$$

5.2 The static routing problem

Filling Q contracts at the lowest achievable cost is a linear program in the per level allocations:

$$\text{minimise } \sum_{k=1}^K p_k x_k \quad \text{subject to } \sum_{k=1}^K x_k = Q, \quad 0 \leq x_k \leq d_k$$

The feasible region is a box intersected with a single equality, and the objective is linear, so the optimum is the greedy solution: consume the cheapest level in full, move to the next, and stop when Q is filled. The argument is a one line exchange. If any allocation puts weight on level j while a cheaper level i has room left, shifting one contract from j to i lowers the objective by the price difference and preserves feasibility, so no such allocation can be optimal. This is the book walk, and it is not a heuristic. It is the exact solution.

Let m be the first level at which cumulative depth covers the order, that is the smallest j with cumulative depth at least Q . The average fill price is then closed form:

$$\bar{p} = (1/Q) [\sum_{k=1}^{m-1} p_k d_k + p_m (Q - D_{m-1})]$$

Slippage is the difference between that average and the touch, which the ticket shows before you commit rather than after:

$$s(Q) = \bar{p}(Q) - p_1$$

Two properties matter operationally. Slippage is non decreasing and convex in Q , so the marginal contract always costs at least as much as the one before it, and the whole curve can be precomputed from a single book snapshot. And because the fee F depends only on M , L and the fixed network component, it does not enter the routing objective at all. Routing minimises book cost alone, which is a small thing that keeps the optimisation honest.

5.3 Two venues, and where the pool comes in

Probio has a second source of liquidity. The pool quotes when the external book is thin. That turns a single ladder into an allocation across two venues with different cost curves, and the problem stops being a linear program.

Write the total cost of taking x contracts from the book as a convex increasing function, and the same for y contracts from the pool. Any bonding curve with a monotonically increasing marginal price satisfies this, as does the discrete ladder above, whose cost function is piecewise linear and convex. The problem is:

$$\text{minimise } C_B(x) + C_P(y) \quad \text{subject to } x + y = Q, \quad x \geq 0, \quad y \geq 0$$

Form the Lagrangian with multiplier λ on the quantity constraint:

$$\mathcal{L}(x, y, \lambda) = C_B(x) + C_P(y) + \lambda (Q - x - y)$$

The first order conditions give the entire answer. At an interior optimum the marginal cost of the next contract is equal at both venues, and that common value is the multiplier:

$$C_B'(x^*) = C_P'(y^*) = \lambda^*$$

With the corner cases handled by complementary slackness: if the pool is uncompetitive even for the first contract, the whole order goes to the book, and vice versa.

$$x^* = 0 \iff C_B'(0) \geq C_P'(Q) \quad y^* = 0 \iff C_P'(0) \geq C_B'(Q)$$

The economic reading is water filling. You pour size into whichever venue is currently cheapest at the margin, and the two marginal prices rise together until they meet. The multiplier is the clearing marginal price of the order as a

whole, and it is the correct number to quote a trader as the true cost of the next contract, ahead of the average fill price, which is backward looking by construction.

5.4 A closed form worth having

Linearising both marginal cost curves around the touch, with a and c the respective starting marginal prices and b and d the respective depth slopes, makes the split explicit:

$$C'_B(x) = a + b x, \quad C'_F(y) = c + d y$$

$$x^* = \text{clip}((c - a + d Q) / (b + d), 0, Q), \quad y^* = Q - x^*$$

Read the numerator. The pool takes flow when it starts cheap relative to the book, when c minus a is small or negative, and it takes more of it as the order grows, through the dQ term. Read the denominator. A steep book, meaning a large b , pushes flow to the pool; a steep pool pushes flow back. The phrase in the product documentation, that the pool quotes when the external book is thin, is exactly the statement that b is large.

5.5 Staleness

Every routed quote is computed against a snapshot with some age. Over an interval of length τ the contract price drifts by a random amount with standard deviation σ times the square root of τ , so a quote is only meaningful while that drift is small relative to the slippage being quoted:

$$\sigma \sqrt{\tau} \ll s(Q)$$

When it is not, the honest move is to re-quote rather than to commit against a book that no longer exists. The same principle runs through the infrastructure. Balance reads are bounded at three seconds and degrade to an unknown balance rather than hanging the interface, order books and price history are fetched on demand for the selected market only rather than being served from the bulk cache, and explicit refresh always bypasses the cache. A stale number that looks live is worse than an obviously missing one.

6 Execution scheduling

Routing answers where a given order goes. Scheduling answers a harder question: given that trading fast costs you impact and trading slow costs you risk, how should an order be spread over time. We use the Almgren and Chriss framework, which is the standard institutional answer and which specialises unusually cleanly to binary contracts.

6.1 Setup

Acquire or unwind X contracts over a horizon T , divided into N intervals of length τ equal to T over N . Let the holdings after interval k be x , with x at time zero equal to X and x at the end equal to zero, and let the traded quantity in interval k be the difference between successive holdings. Price evolves with a random walk plus permanent impact, and each fill is executed at a worse price because of temporary impact:

$$S_k = S_{k-1} + \sigma \sqrt{\tau} \cdot \xi_k - \tau g(n_k / \tau)$$

$$\tilde{S}_k = S_{k-1} - h(n_k / \tau)$$

Take the standard linear forms, with γ the permanent impact coefficient, ϵ a fixed cost per trade covering the half spread, and η the temporary impact coefficient:

$$g(v) = \gamma v, \quad h(v) = \epsilon \text{sgn}(v) + \eta v$$

6.2 Cost and risk

The expected implementation shortfall decomposes into a permanent term that is independent of the schedule, a fixed cost proportional to the number of trades, and a quadratic temporary impact term:

$$E(x) = \gamma X^2 / 2 + \epsilon \sum_{k=1}^N |n_k| + (\tilde{\eta} / \tau) \sum_{k=1}^N n_k^2, \quad \tilde{\eta} = \eta - \gamma \tau / 2$$

The variance of the shortfall is driven by how much inventory is still exposed to the random walk, interval by interval:

$$V(x) = \sigma^2 \tau \sum_{k=1}^N x_k^2$$

A trader with risk aversion λ minimises the mean variance combination of the two. λ is the only preference parameter in the model, and it has units of inverse currency, so it can be calibrated against a stated tolerance rather than guessed:

$$\text{minimise } U(x) = E(x) + \lambda V(x)$$

6.3 Solution

Differentiating with respect to each interior holding and rearranging gives a second order linear difference equation, the discrete analogue of a diffusion with a restoring force:

$$(x_{k-1} - 2x_k + x_{k+1}) / \tau^2 = \tilde{K}^2 x_k, \quad \tilde{K}^2 = \lambda \sigma^2 / \tilde{\eta}$$

With the boundary conditions at both ends, the solution is a hyperbolic sine trajectory:

$$x_j = [\sinh(\kappa (T - t_j)) / \sinh(\kappa T)] \cdot X$$

The exact discrete decay rate solves the cosh relation below, and converges to kappa tilde as the interval length goes to zero, so for practical interval sizes the two are interchangeable:

$$\cosh(\kappa \tau) = 1 + \tilde{K}^2 \tau^2 / 2$$

The single number that summarises the schedule is the half life of the position, which is one over kappa. Substituting back gives the form worth memorising:

$$\theta = 1 / \kappa \approx \sqrt{\tilde{\eta} / (\lambda \sigma^2)}$$

More risk aversion or more volatility shortens the half life and pushes the schedule to the front. More liquidity, meaning a larger eta, lengthens it. In the two limits the model reproduces the strategies people actually use: as lambda goes to zero the sinh becomes linear and the schedule is a uniform time slice, and as lambda grows large the schedule collapses to an immediate fill.

6.4 The binary specialisation

Here is where prediction markets differ from equities in a way that matters. A contract that settles at zero or one cannot diffuse like a stock, because the price is confined to the unit interval. The natural bounded process is the Jacobi diffusion, whose instantaneous volatility vanishes at both boundaries:

$$dp = \sigma \sqrt{p(1-p)} dW$$

Substituting the state dependent volatility into the decay rate makes kappa a function of where the contract trades:

$$\kappa(p) = \sqrt{\lambda \sigma^2 p(1-p) / \tilde{\eta}}$$

Kappa is maximised at a price of one half and falls away toward both boundaries, so the optimal schedule is fastest for genuinely uncertain markets and slowest for markets that have nearly resolved. Concretely, the volatility factor at 0.50 is 0.500 and at 0.05 it is 0.218, so the half life of an order in a long shot is roughly 2.3 times the half life of the same order in a coin flip. That is the correct behaviour and it falls out of the model rather than being bolted on. It also has a second consequence: the horizon T is capped by the resolution date, which no equity execution model has to worry about, and as resolution approaches the schedule is forced to compress regardless of lambda.

7 Fill probability and the passive versus aggressive decision

Sections 5 and 6 assume you cross the spread. You do not always have to. Posting inside the book saves the spread but risks not filling at all, and on a thin prediction market book that risk is real. The decision needs a number, not an instinct.

7.1 First passage

Model the mid as a driftless random walk over the working horizon and place a passive buy at a distance delta below it. The order fills if the running minimum ever touches the limit, and the reflection principle gives that probability in closed form:

$$\pi(\delta, T) = P(\min_{t \leq T} W_t \leq -\delta) = 2 \Phi(-\delta / (\sigma \sqrt{T}))$$

The factor of two is the whole content of the reflection principle: every path that ends below the limit is matched by a path that touched it and came back. It doubles the naive estimate, and ignoring it systematically underprices patience.

7.2 Arrival intensity

The first passage model prices the price path but ignores the queue. The complementary approach treats fills as a counting process whose intensity decays exponentially in the distance from the touch, which is the Avellaneda and Stoikov parameterisation and which fits observed market order flow well:

$$\lambda(\delta) = A e^{-k\delta}, \quad \pi(\delta, T) = 1 - \exp(-A e^{-k\delta} T)$$

A is the arrival rate at the touch and k measures how quickly flow thins out as you move away from it. Both are estimable per market from observed queue depletion, and k is the more informative of the two: a small k means a market where posting one tick back barely costs you fill probability, and a large k means a market where anything other than the touch is decoration.

7.3 Queue position

Even at the touch, resting size ahead of you has to clear first. If the volume arriving in the horizon is Poisson with rate mu, the fill probability is the tail of that distribution above the queue ahead:

$$\pi = P(N(\mu T) > Q_ahead) = 1 - \sum_{j=0}^{Q_ahead} e^{-\mu T} (\mu T)^j / j!$$

This is the term that punishes joining a queue late, and it is the one most retail interfaces do not model at all because they never show the queue.

7.4 The decision rule

Crossing now costs delta with certainty. Posting saves delta with probability pi and, with the complement, leaves you with an unfilled order and some penalty c, which is the cost of either missing the trade or crossing later at a worse price. Expected cost of posting, measured against crossing now:

$$\Delta C = -\pi \delta + (1 - \pi) c$$

Posting is preferred when that quantity is negative, which reduces to a clean threshold on fill probability:

$$\pi > \pi^* = c / (\delta + c)$$

7.5 Adverse selection

One correction is not optional. Conditional on a passive buy filling, the market came to you, and the reason it came to you is usually information. The expected value of the contract conditional on a fill is below its unconditional value by some amount a, which is the winner curse applied to limit orders. The effective saving is the spread you captured minus the information you paid for, and the threshold moves accordingly:

$$\pi > c / (\delta - a + c)$$

If a exceeds delta the passive order is never worth posting at that distance no matter how likely it is to fill, which is the correct and slightly deflating conclusion. In prediction markets a spikes hard around scheduled resolution inputs, which is to say around exactly the moments when a trader most wants to be filled. The practical rule that follows is to trade aggressively into news and passively into quiet, and the model says by how much rather than merely which.

7.6 Putting the three sections together

A full ticket is a joint decision across all three. The scheduler of Section 6 sets how much of the order should be worked in each interval. Within an interval, the rule above splits that quantity into an aggressive slice that crosses immediately and passive slices resting at distances where the fill probability clears the threshold. The router of Section 5 then allocates the aggressive slice across the book ladder and the pool at equal marginal cost. Three models, one quote, and every term in it visible on the ticket before the trader commits.

8 Parlays

Up to ten independent outcomes on a single ticket. Probabilities multiply, so the payout multiple is the reciprocal of the product of the leg prices:

$$P = p_1 \cdot p_2 \cdots p_n, \quad \text{payout multiple } m = 1 / P$$

Four legs at fifty percent each is a 6.25 percent chance of paying sixteen times. Both numbers update live in the builder as legs are added, because the second one is the one people stop looking at around leg five. Parlays carry no leverage, so only the fixed network fee applies.

8.1 Parlays are fair, which is the point

If the legs are correctly priced and genuinely independent, the expected value of a parlay is the stake back minus the network fee. Nothing more and nothing less:

$$E[\text{payoff}] = P \cdot (M / P) - M - \varphi = -\varphi$$

This is worth stating plainly, because it is the opposite of how a sportsbook parlay works. A sportsbook parlay compounds the vig on every leg, so a five leg parlay is a considerably worse bet than any of its legs. A Probio parlay compounds nothing except the probabilities themselves. The house is not on the other side.

8.2 Where the edge actually is

Let q_i be your own probability for leg i against the market price p_i . The expected value multiple of the parlay is the product of the per leg ratios, which is more usefully written as a sum in logs:

$$\text{EV multiple} = (q_1 \cdots q_n) / (p_1 \cdots p_n), \quad \log \text{EV} = \sum_{i=1}^n \log(q_i / p_i)$$

Edge is additive in log space, which is the real argument for parlays and the only honest one. If you hold a uniform proportional edge e on every leg, the expected value multiple is exponential in the number of legs:

$$q_i = p_i (1 + e) \implies \text{EV multiple} = (1 + e)^n$$

Ten legs at a five percent edge each is a 63 percent expected return. That is the seductive part. Here is the sobering part.

8.3 Optimal size collapses faster than edge grows

With net odds b equal to one over P minus one, and true win probability W equal to the product of your own probabilities, the Kelly fraction of bankroll is:

$$f^* = [bW - (1 - W)] / b = W - (1 - W)P / (1 - P)$$

Legs	Win chance	Payout	EV multiple	Kelly fraction
1	50.0%	2x	1.05	5.00%
2	25.0%	4x	1.10	3.42%
3	12.5%	8x	1.16	2.25%
4	6.25%	16x	1.22	1.44%
5	3.13%	32x	1.28	0.89%
6	1.56%	64x	1.34	0.54%

Table 4. Legs priced at 0.50 with a uniform five percent edge. Expected return climbs; the amount you should risk falls by roughly a third per leg.

The edge grows at five percent per leg and the optimal stake falls at roughly thirty five percent per leg. A six leg parlay with genuine edge on every leg is still a position you should size at half a percent of bankroll. We show the combined odds and the implied chance side by side for exactly this reason.

8.4 The independence assumption

Product pricing assumes the legs are independent. They frequently are not. The Frechet bounds say what is possible without that assumption, for any dependence structure whatever:

$$\max(0, \sum_{i=1}^n p_i - (n - 1)) \leq P(\cap) \leq \min_i p_i$$

The direction of the error is what matters at the ticket. If the legs are positively dependent, the true joint probability exceeds the product, so a parlay priced on the product is generous and the trader is being paid too much. If they are negatively dependent, the reverse, and the trader is being paid too little. The pathological case is easy to construct and easy to walk into: two mutually exclusive outcomes of the same election priced at 0.60 and 0.50 multiply to 0.30, while the true joint probability is zero. No payout multiple compensates for that.

Correlation is therefore the trader edge or the trader trap, depending on its sign, and it is not something the pricing engine can infer for you. It is the one place in the product where we ask the user to think.

9 Settlement, custody and the network

Probio settles in ETH on Robinhood Chain, an Ethereum layer two on the Arbitrum Orbit stack, chain ID 4663 with testnet 46630, explored through Blockscout. The bundled public RPC is rate limited with no uptime guarantee, which is acceptable for browsing and wrong for anything else, so Alchemy is the recommended provider and the endpoint is configurable under Settings, Advanced.

Wallet addresses are derived, never stored. The account identifier is combined with a server secret through HMAC-SHA256 to produce a secp256k1 key, so the same account resolves to the same address in every environment with no key material sitting in any database:

```
seed    = HMAC-SHA256(server_secret, "probio-wallet:v2:evm:" + userId)
address = secp256k1(seed).address // EIP-55 checksummed
```

Only the public address is ever returned to the client. The derived private key is used server side and is never serialised into a response.

Derivation depends on a single server secret, so the secret is the rotation boundary: changing it changes every derived address. The scheme is versioned in the seed string, which is what makes a future migration to per user key material held in a KMS a version bump rather than a rewrite.

Positions still open when a market resolves are settled automatically against the published resolution, at one or zero. There is no manual close required and no expiry decision to get wrong.

Network	Robinhood Chain, Arbitrum Orbit stack
Chain ID	4663 mainnet, 46630 testnet
Native asset	ETH, 18 decimals
Explorer	Blockscout
Recommended RPC	Alchemy
Balance read bound	3 seconds, then degrade to unknown

Table 5. Network parameters. `lib/chain.ts` is the single source of truth for these values.

10 Market data and caching

The catalogue is crawled from the Polymarket events endpoint and cached server side, so browsing several thousand markets does not become several thousand upstream calls. Order books and price history are fetched on demand for the selected market only, because they are the two things that must never be stale.

Every read path falls back to the live Polymarket feed if the local store is unavailable, so a cold instance serves the full catalogue rather than a truncated sample. Requests are never cached by a shared CDN. Signed out visitors get a five minute private browser cache matching the server side feed TTL; signed in visitors always get fresh data. Explicit refresh and the auto refresh interval bypass the cache in both cases.

Two deployment facts follow from the architecture and are worth stating. Serverless filesystems are read only outside the temp directory and each instance gets its own copy, so the SQLite catalogue does not persist or share between instances, which is handled by the fallback above; setting a real Postgres connection string gives a shared store and noticeably faster cold requests. And the public environment variables are inlined at build time, so changing them requires a redeploy rather than an environment update.