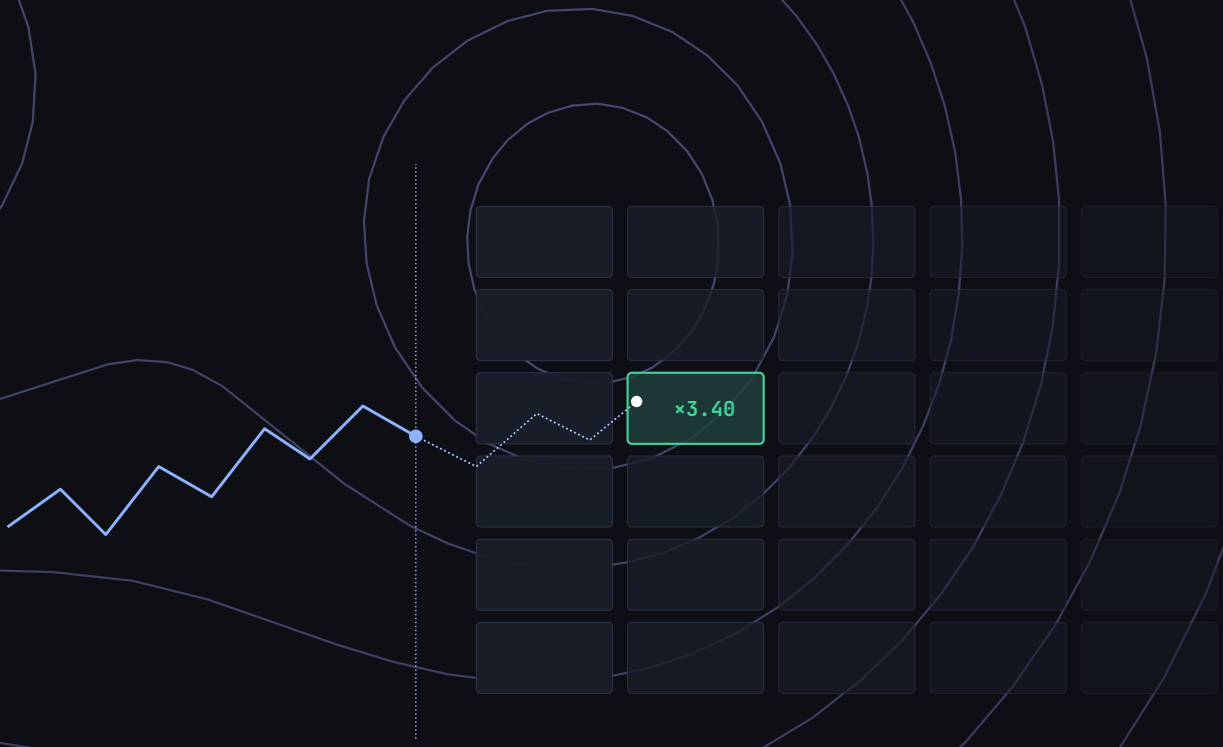
White paper  v1.0 (beta)  November 2026

# One Tap Trade

## Bespoke and exotic one-touch options: a volatility trading platform




**Satoshi of High Frequency Volatility Trading**
One Tap Trade

Pricing, verification and risk for short-dated volatility, set out in full.

## Abstract

One Tap Trade is a venue for trading short-dated volatility through *tiles*: forward-starting range one-touch options on a live price, each defined by a price band and a time window lasting seconds. A tile pays a fixed multiple of its stake if the price passes through its band while its window is open. This paper sets out how the platform (i) forms a manipulation-resistant reference price from several exchanges with a Kalman filter; (ii) forecasts integrated variance from tick data contaminated by microstructure noise, anchored to an arbitrage-free eSSVI implied-volatility surface; (iii) prices a tile's touch probability with the reflection principle on a variance clock, with corrections for discrete monitoring, jumps and volatility uncertainty; (iv) converts probabilities into multipliers under a published house margin of $e = 6\%$, giving a return to player of exactly 94% on an unloaded tile, never more than 98% on any tile, under the pricing measure; (v) extends the same pricing to user-chosen window lengths and band heights (*Pro*); and (vi) makes every trade verifiable by anyone through canonical records sealed with SHA-256, a hash-chained price recording, and a fill-after-commit rule. We describe the vault that takes the other side of every tile, its Monte Carlo risk budget, and *Spider*, a Markov-chain and Monte Carlo trading agent whose every decision is sealed into a hash-linked, verifiable decision trace. A zero-knowledge proof layer for quote correctness is described as work in development. The paper publishes the rules of the game in full; it does not publish the parameters of the risk charges that protect the vault.

**Status.** One Tap Trade is in beta. Balances are held in USDO, a practice unit with no cash value. Nothing in this paper is an offer of a financial product. Figures marked *illustrative* use a simulated market with the stated parameters.

### Notation at a glance

| | | | |
|---|---|---|---|
| $S_t$ | reference price (mark) | $\Delta$ | settlement tick interval |
| $[a, b)$ | tile band, height $h = b - a$ | $[t_1, t_2]$ | tile window, length $w = t_2 - t_1$ |
| $V(t)$ | integrated variance (variance clock) | $s_1$, $s_2$ | price s.d. to window start, across window |
| $\eta$ | print-noise s.d.; vol-uncertainty width | $P$, $\bar{P}$ | touch probability; pricing probability |
| $e$ | house margin (6%) | $A$ | risk-charge factor |
| $M$ | multiplier (payout per unit stake) | $f_i$, $f_o$ | entry fee, exit fee (each 10 bp) |
| NAV | vault net asset value | SHA-256 | the hash function of FIPS 180-4 |

### Main results

**Pricing.** A tile's touch probability has a closed form on the variance clock, corrected for discrete ticks, print noise, jumps and uncertain volatility (Proposition 2). Checked against Monte Carlo simulation of the settlement rule, the corrected price is accurate or slightly conservative (Fig. 4).

**Return to player.** Every quote returns at most $(1 - e)/A \leq 98\%$ under the pricing probability; an unloaded tile returns 94%, or 93.8–93.9% after fees (Theorem 3). On any book the house margin is at least 2% of turnover (Theorem 5).

**Verifiability.** Each trade carries a SHA-256 seal over its canonical record and points into a hash-chained price recording; six public checks recompute fill, multiplier, settlement and payout (Algorithm 1).

**Pro.** Custom windows and band heights are priced by the same function, so quotes are monotone: a tile that contains another never pays more than it, up to rounding (Proposition 6).

**Spider.** Every decision of the trading agent is sealed into a hash-linked trace that cannot be altered or reattributed after the fact (Section 12).

# Contents


1 Introduction 4

2 The instrument 4

3 The consensus mark 5

4 Volatility 5
- 4.1 Two-scale realised variance 5
- 4.2 The variance clock 6
- 4.3 Jumps 6
- 4.4 An arbitrage-free implied anchor: eSSVI 6

5 The touch probability 7
- 5.1 Continuous monitoring 7
- 5.2 Discrete monitoring 7
- 5.3 Print noise (live markets) 8
- 5.4 Jumps and volatility uncertainty 8
- 5.5 Self-calibration 8

6 From probability to multiplier 9
- 6.1 The multiplier 9
- 6.2 Return to player 9
- 6.3 A verifiable margin 10

7 Bespoke tiles: Pro 11

8 Fill after commit, and settlement 11

9 Cryptographic integrity and verification 12
- 9.1 Sealed trade records 12
- 9.2 Hash-chained price recording 12
- 9.3 Verification 13

10 Zero-knowledge pricing proofs (in development) 13

11 The vault and its risk budget 14
- 11.1 Accounting 14
- 11.2 Limits 14
- 11.3 Monte Carlo value at risk 15

12 Spider: a Markov-chain agent with a verifiable decision trace 15
- 12.1 Model 15
- 12.2 A verifiable decision trace 16

13 Limitations 16

A Proof of the forward-start touch formula 17

B Parameters used in the illustrative figures 17

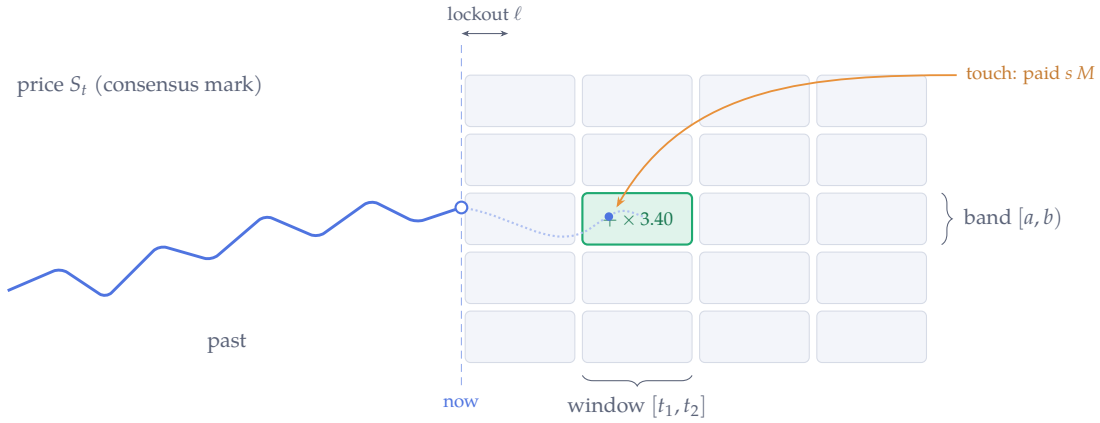



**Figure 1.** Anatomy of the trading grid. Columns are time windows and rows are price bands; each tile is an option that pays $s \cdot M$ if the price passes through its band while its window is open. Tiles inside the lockout $\ell$ after *now* cannot be bought.

# 1 Introduction

Options on volatility are among the most informative instruments in markets, yet the products that expose short-dated volatility directly are either institutional (variance swaps, over-the-counter barrier options) or opaque (binary-option apps that set their own payouts and settle against their own prices). One Tap Trade takes the barrier option, the simplest instrument whose value is *pure* path volatility, and makes it the unit of trading: a live price chart becomes a grid, every cell of the grid is an option, and one tap buys it.

Three properties guide the design.

**P1. Fair pricing.** Each multiplier is derived from a stated probability model and a stated margin. The margin is applied identically to every participant.

**P2. Verifiability.** Every trade, including those of automated participants, is sealed into a canonical record. Any third party can recompute the fill, the multiplier, the settlement and the payout from the record and a public price recording.

**P3. Solvency.** A vault takes the other side of every tile. Its exposure is bounded by limits that scale with its net asset value and by a Monte Carlo value-at-risk budget.

**Contributions.** Section 2 defines the tile as a forward-starting range one-touch option with discrete monitoring. Section 3 describes the consensus mark. Section 4 derives the variance forecast and its eSSVI anchor. Section 5 gives the touch probability with its corrections and validates it against simulation. Section 6 converts probabilities into multipliers and proves the return-to-player bounds. Section 7 treats bespoke tiles, Section 8 the fill and settlement rules, Section 9 the cryptographic sealing and verification, and Section 10 the zero-knowledge layer in development. Section 11 covers the vault and its risk budget, and Section 12 the Spider agent and its verifiable decision trace. Section 13 states the limitations.

# 2 The instrument

Let $(S_t)_{t \geq 0}$ be the reference price and let settlement ticks occur at $t_i = i\Delta$, with $\Delta = 0.5\,\text{s}$ in the simulated market and $0.2\,\text{s}$ for BTCUSD.

**Definition 1** (Tile)**.** A tile $\mathcal{T} = [a, b) \times [t_1, t_2]$ with $a < b$, $t_1 < t_2$ on the tick grid is the claim that pays $s\,M$ at the first tick of $[t_1, t_2]$ at which the event

$$E_{\mathcal{T}} = \left\{ \max_{t_i \in [t_1, t_2]} S_{t_i} \geq a \text{ and } \min_{t_i \in [t_1, t_2]} S_{t_i} < b \right\} \tag{1}$$

becomes true, and pays nothing if the window closes without $E_{\mathcal{T}}$. Here $s$ is the stake and $M$ the multiplier fixed at the fill.

Equivalently, the range of the ticks printed inside the window overlaps the band: a price that jumps over the band between two ticks has still passed through it. For $b \to \infty$ the tile is a forward-starting one-touch digital with barrier $a$; for a band containing the current price and a window that is already open, it has already been touched. The grid uses a common band height $h = b - a$ and window length $w = t_2 - t_1$ (3 s by default); *Pro* lets the trader choose both (Section 7).

A tile is thus an exotic option whose value is driven almost entirely by volatility over seconds: at these horizons drift is negligible, and the probability of touching a band is a function of the distance to the band measured in units of expected price variation.

# 3 The consensus mark

No single exchange settles a bet. For BTCUSD the mark is a filtered consensus of Binance, Bybit, OKX and Deribit, computed by the platform's hub every 200 ms.

**Measurement model.** Each venue $v$ publishes a fair price (mid of its book); its log is modelled as the latent log-price $x_t$ plus a venue bias $b_v$ and noise with venue-specific variance $R_v$:

$$z_{v,t} = \ln \text{fair}_{v,t} - b_v = x_t + \varepsilon_{v,t}, \qquad \varepsilon_{v,t} \sim \mathcal{N}(0, R_v). \tag{2}$$

**Filter.** The latent state follows a random walk with diffusion $q$ per second. Updates are fused in arrival order by a scalar Kalman filter [8],

$$\hat{x}_t^- = \hat{x}_{t'}, \quad P_t^- = P_{t'} + q\,(t - t'), \qquad K_t = \frac{P_t^-}{P_t^- + R_v}, \qquad \hat{x}_t = \hat{x}_t^- + K_t\,(z_{v,t} - \hat{x}_t^-), \qquad P_t = (1 - K_t)P_t^-. \tag{3}$$

Each venue's weight in the consensus is proportional to its precision $1/R_v$, which is learned online from the venue's own innovations, so a jittery or thin venue counts less. Biases are re-centred so that the median venue has zero bias; the mark therefore cannot drift away from all venues at once.

**Outliers and genuine moves.** An update is rejected when

$$|z_{v,t} - \hat{x}_t^-| > \max\big(5\sqrt{P_t^- + R_v},\ 10\,\text{bp}\big), \tag{4}$$

unless at least two venues, and at least half of the live venues, move together within one second, in which case the move is genuine and the filter follows at once. A single venue's glitch therefore cannot settle a bet, while a real market move is never filtered away.

**Recording.** Every venue input and every mark is appended to recording segments chained by hash (Section 9). When no venue has quoted for two seconds the mark is flagged stale; every open window that overlaps a stale stretch is voided and refunded in full, so no bet is ever settled on a price nobody quoted.

# 4 Volatility

A tile's value depends on how far the price can travel before and during its window. Over seconds this is governed by the integrated variance, which must be forecast from noisy prints and kept consistent with the options market.

## 4.1 Two-scale realised variance

Printed prices carry microstructure noise (venue bounce, filter wobble) on top of the diffusion: $y_t = x_t + \epsilon_t$ with $\epsilon_t$ i.i.d. of variance $\eta^2$. A return over an interval $\delta$ then satisfies

$$\mathbb{E}\big[r_\delta^2\big] = \sigma^2\delta + 2\eta^2. \tag{5}$$

Measuring exponentially weighted second moments $m_\Delta$ and $m_L$ at the tick interval $\Delta$ and at a longer lag $L \approx 2$ s separates the two:

$$\hat{\sigma}^2 = \frac{m_L - m_\Delta}{L - \Delta}, \qquad \hat{\eta}^2 = \tfrac{1}{2}\big(m_\Delta - \hat{\sigma}^2\Delta\big). \tag{6}$$

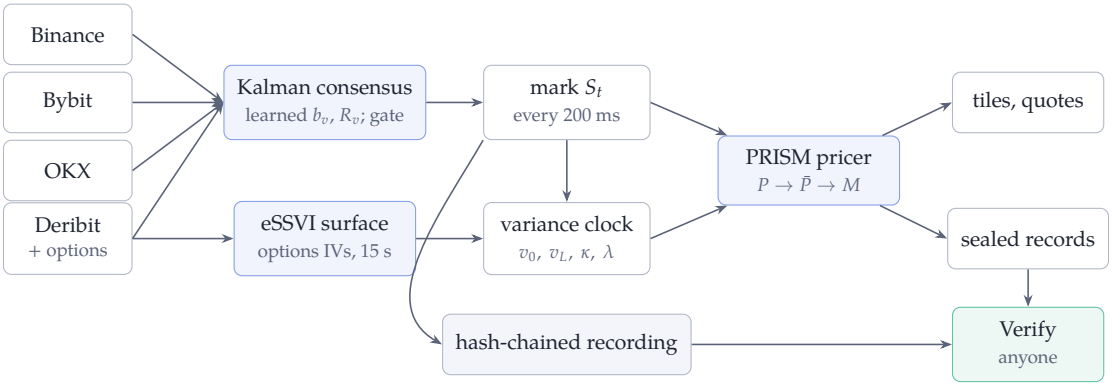



**Figure 2.** Data flow. Exchange prints are fused into a mark; the options market anchors the volatility forecast; the pricer turns both into multipliers. Every mark is written to a hash-chained recording and every settled trade to a sealed record, which together let anyone verify any trade.

This is the two-scale idea of [11]; noise is thereby kept out of the diffusion forecast and priced separately at settlement (Section 5.3). Fast and slow exponentially weighted averages of the de-noised returns give the current level $v_0$ and the long-run level $v_L$.

## 4.2 The variance clock

Let $v_u$ be the instantaneous variance with mean-reverting expectation $\mathbb{E}[v_u] = v_L + (v_0 - v_L)e^{-\kappa u}$, as in the Heston model [6]. The forecast integrated variance from now to $t$ is

$$V(t) = \int_0^t \mathbb{E}[v_u]\,\mathrm{d}u = v_L\,t + (v_0 - v_L)\,\frac{1 - e^{-\kappa t}}{\kappa}. \tag{7}$$

With $V$ expressed in relative (return) variance per second, the price standard deviations used for a tile are

$$s_1 = S\sqrt{V(t_1)}, \qquad s_2 = S\sqrt{V(t_2) - V(t_1)}, \qquad s_\Delta = S\sqrt{\frac{V(t_2) - V(t_1)}{t_2 - t_1}}\,\Delta. \tag{8}$$

By the Dambis–Dubins–Schwarz theorem a continuous martingale with deterministic quadratic variation is a time-changed Brownian motion, so touch probabilities depend on the integrated variance only and not on clock time; (7) is the clock. A calm minute before known volatility is therefore not priced as calm: the forecast reverts towards $v_L$ over the window. A super-fast estimator lifts $v_0$ immediately during a volatility shock.

## 4.3 Jumps

Fat tails are represented by compound Poisson jumps with intensity $\lambda$ and relative size $J \sim \mathcal{N}(\mu_J, \sigma_J^2)$, estimated from realised history in the live market. The probability of at least one jump before the window closes is $q = 1 - e^{-\lambda t_2}$.

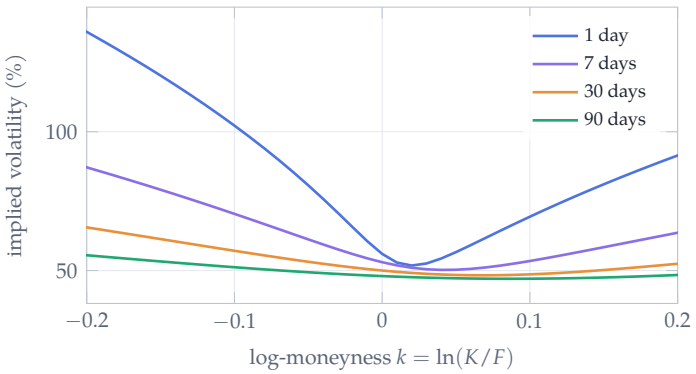
## 4.4 An arbitrage-free implied anchor: eSSVI

The long-run level and the term structure are anchored to the options market. Every 15 seconds the platform fits an extended SSVI surface [5, 2] to option mark implied volatilities across venues (median per strike). With $T$ in years, log-moneyness $k = \ln(K/F)$ and total implied variance $w = \sigma_{\text{imp}}^2 T$, each expiry slice is

$$w(k;\theta,\rho,\psi) = \frac{\theta}{2}\left(1 + \rho\varphi k + \sqrt{(\varphi k + \rho)^2 + 1 - \rho^2}\right), \qquad \varphi = \psi/\theta, \tag{9}$$

where $\theta$ is the at-the-money total variance, $\rho \in (-1, 1)$ the skew and $\psi = \theta\varphi$ (the at-the-money slope of $w$ is $\rho\psi$ and its curvature $\psi^2(1-\rho^2)/(2\theta)$). In eSSVI, $\rho$ and $\psi$ vary by expiry. The fit (Nelder–Mead with penalties) enforces the sufficient no-butterfly-arbitrage conditions of Gatheral and Jacquier,

$$\psi\,(1 + |\rho|) < 4, \qquad \psi^2(1 + |\rho|) \le 4\theta, \tag{10}$$




| $T$ (d) | $\theta$ | $\rho$ | $\psi$ |
|---|---|---|---|
| 1 | $8.592 \times 10^{-4}$ | −0.38 | 0.0322 |
| 7 | $5.387 \times 10^{-3}$ | −0.32 | 0.0807 |
| 30 | $2.055 \times 10^{-2}$ | −0.26 | 0.1577 |
| 90 | $5.681 \times 10^{-2}$ | −0.20 | 0.2622 |

All four slices satisfy the butterfly bounds; consecutive slices satisfy the calendar bounds and $w_{i+1}(k) \geq w_i(k)$ on the plotted grid.

**Figure 3.** An eSSVI surface of the kind fitted every 15 seconds (illustrative parameters, verified arbitrage-free). Short expiries carry steeper skew; the surface anchors the long-run variance $v_L$ and the term structure used by the pricer.

and, between consecutive slices $1 \to 2$, the calendar conditions

$$\theta_2 \geq \theta_1, \qquad \psi_2 \geq \psi_1, \qquad |\rho_2\psi_2 - \rho_1\psi_1| \leq \psi_2 - \psi_1. \tag{11}$$

After fitting, $w_2(k) \geq w_1(k)$ is also checked numerically on a grid of $k$, which is the actual no-calendar-arbitrage condition. The term structure of the fitted surface also flags scheduled events, around which quotes widen.

# 5 The touch probability

## 5.1 Continuous monitoring

Over horizons of seconds the relative price change is of order $10^{-4}$, so the price is modelled as Gaussian on the variance clock (the difference from a log-normal model is of the same small order). The building block is the reflection principle [9].

**Lemma 1** (Reflection). *For a Brownian motion $W$ started at $x < a$ and run for variance $\Sigma^2$, $\mathbb{P}\big(\max_{u \leq \Sigma^2} W_u \geq a\big) = 2\,\bar{\Phi}\big(\frac{a-x}{\Sigma}\big)$, where $\bar{\Phi} = 1 - \Phi$.*

Conditioning on the price at the start of the window and applying the Markov property gives the probability of touching a band during a forward-starting window.

**Proposition 2** (Forward-start range touch). *Under continuous monitoring, with $X = S_{t_1} \sim \mathcal{N}(S, s_1^2)$,*

$$P(\mathcal{T}) = \mathbb{E}\big[g(X)\big], \qquad g(x) = \begin{cases} 2\,\bar{\Phi}\big((a-x)/s_2\big), & x < a, \\ 1, & a \leq x \leq b, \\ 2\,\bar{\Phi}\big((x-b)/s_2\big), & x > b. \end{cases} \tag{12}$$

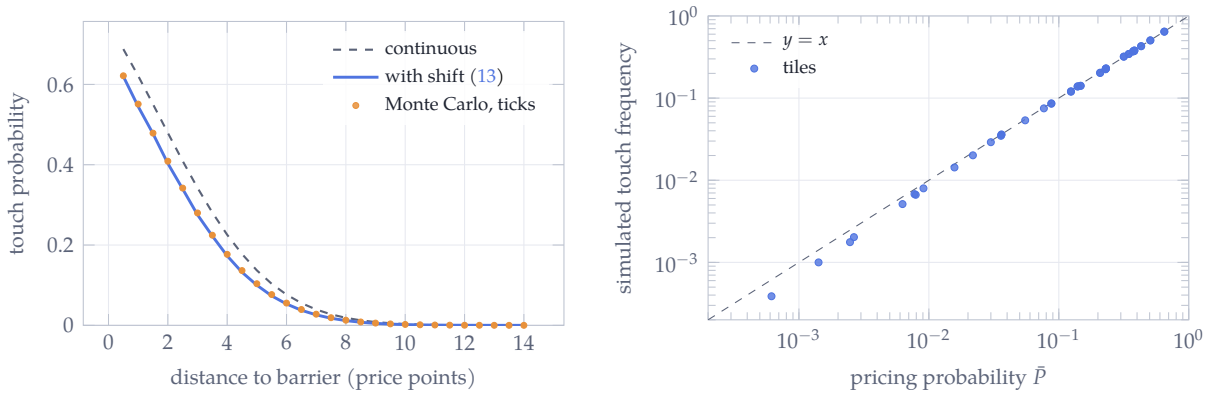
The proof is in Appendix A. The expectation is evaluated with a 121-node rule on $[-8, 8]$ standard deviations; when the window is already open ($s_1 = 0$) it reduces to $g(S)$.

## 5.2 Discrete monitoring

Settlement observes ticks, not a continuous path, so the continuous formula overstates the touch probability. Following Broadie, Glasserman and Kou [1], the barrier is shifted outward by $\beta s_\Delta$ with $\beta = -\zeta(\frac{1}{2})/\sqrt{2\pi} \approx 0.5826$:

$$g(x) \;\to\; 2\,\bar{\Phi}\big((a + \beta s_\Delta - x)/s_2\big) \quad (x < a), \qquad \text{and symmetrically for } x > b. \tag{13}$$

Figure 4(a) compares the continuous formula, the corrected formula and a 400,000-path Monte Carlo of the tick rule: the correction removes a bias of up to about ten percentage points and leaves the formula within about one percentage point of the simulated frequency.




**(a)** Discrete-monitoring correction: window $[3, 6]$ s, ticks every 0.5 s, $\sigma = 1.39$ points per $\sqrt{\text{s}}$.

**(b)** Reliability: 47 tiles over four windows and twelve bands, 300,000 paths each, with the engine's settlement rule and uncertain volatility.

**Figure 4.** Validation of the touch probability against simulation (illustrative market). Points on or just below the diagonal mean the quoted probability matches, or slightly exceeds, the frequency of touches: the model is accurate and, where it errs, conservative.

## 5.3 Print noise (live markets)

In a live market a window is hit when the range of its *printed* prices overlaps the band, and print noise stretches that range. The band is widened on both sides by $w_N = 0.7\,\eta S\sqrt{\min(\eta S/s_\Delta, 3)}$. Monte Carlo calibration against the exact settlement rule (diffusion plus i.i.d. print noise, 200 ms ticks, windows of 1–8 s) leaves every cell within 0 to +3% of the simulated hit rate (log-RMSE 0.026), on the conservative side. Synthetic markets have no print noise and $w_N = 0$.

## 5.4 Jumps and volatility uncertainty

Jumps enter through a three-node Gauss–Hermite average over the jump size,

$$P_J = P_0 + q\Big(\textstyle\sum_{i=1}^{3} \omega_i\, P_0\big(S(1+\mu_J+\sigma_J\xi_i)\big) - P_0\Big). \tag{14}$$

Volatility itself is uncertain, and the touch probability is convex in volatility for distant bands. The pricer therefore averages over a log-normal spread of volatilities using seven-node Gauss–Hermite quadrature with nodes $x_j$ and weights $\omega_j$ for the weight $e^{-x^2}$, scaling every standard deviation by $\exp(\eta_\sigma x_j - \eta_\sigma^2)$:

$$P_{\text{mix}} = \max\Big(\textstyle\sum_{j=1}^{7} \omega_j\, P\big(\sigma e^{\eta_\sigma x_j - \eta_\sigma^2}\big) + (P_J - P_0),\ P_J\Big). \tag{15}$$

The maximum guarantees that uncertainty about volatility is never used to quote a probability below the model's own point estimate.

## 5.5 Self-calibration

Settled bets are grouped by their quoted probability into six buckets with edges 0, 0.3%, 1%, 3%, 10%, 30%. Once enough bets have settled, the ratio of realised to expected hits in a bucket, shrunk towards one by prior pseudo-counts and bounded to $[0.9, 3]$, scales the probability. The pricing probability is

$$\bar{P} = \min\big(0.999,\ \max(P_{\text{mix}},\ c_j\, P_J)\big), \tag{16}$$

so calibration can only make quotes more conservative when the model has under-predicted, and never pushes $\bar{P}$ below the mixture estimate.

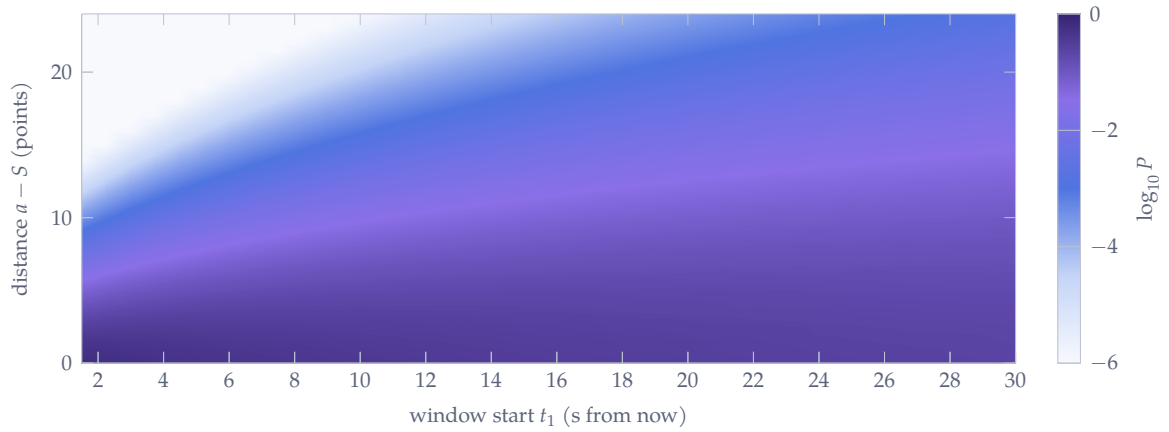



Figure 5. Touch probability of a 2-point band over a 3-second window as a function of its distance and of when its window opens (illustrative market, $\sigma = 1.39$ points per $\sqrt{\text{s}}$). Probability falls off as a Gaussian tail in distance and rises with time: a far band becomes reachable only for later windows.

# 6 From probability to multiplier

## 6.1 The multiplier

The house margin is $e = 6\%$. For a marginal (infinitesimal) bet the multiplier is

$$M_0 = \frac{K}{A}, \qquad K = \frac{1-e}{\bar{P}}, \qquad A = \text{clamp}\Big(1 + \textstyle\sum_c \pi_c,\ f_{\min},\ 6\Big), \qquad f_{\min} = \tfrac{1-e}{1-e/3}. \tag{17}$$

The factor $A$ collects risk charges $\pi_c$ that depend on the state of the vault: the tile's correlation with the open book, the use of the risk budget, the open payout on the same side of the price, scheduled events, price velocity, a participant's record of beating the model, and crowding on the tile. A charge can be negative when a tile hedges the open book, which is why $A$ can fall below one, but never below $f_{\min}$. For a bet of size $s$ on a tile with capacity $C$ and crowding coefficient $\kappa_c$, the multiplier solves

$$M\Big(A + \frac{\kappa_c\, s\, M}{2C}\Big) = K, \tag{18}$$

so a bet pays the average of the marginal prices along its own size, and the next bettor on the same tile receives a lower multiplier. Finally the multiplier is rounded *down* to two decimals and capped, and a tile whose multiplier would fall below 1.01 is not offered:

$$M = \min\left(\lfloor 100\, M_{\text{raw}} \rfloor / 100,\ M_{\text{cap}}\right). \tag{19}$$

Figure 6 shows the resulting quotes for an unloaded vault.

## 6.2 Return to player

**Definition 2.** The return to player of a quote under a probability measure $Q$ is its expected payout per unit stake, $\text{RTP}_Q = Q(E_{\mathcal{T}})\, M$. Net of fees, the player pays $s(1 + f_i)$ and receives $s\,(M - f_o(M-1))$ on a win.

**Theorem 3** (Return to player under the pricing probability)**.** *For every quote, with $\bar{P}$ the pricing probability,*

$$\text{RTP}_{\bar{P}} = \bar{P}\, M \ \le\ \frac{1-e}{A} \ \le\ 1 - \frac{e}{3} = 98\%. \tag{20}$$

*For an unloaded tile ($A = 1$, marginal stake, uncapped) the bound is attained up to rounding:*

$$1 - e - 0.01\,\bar{P} \ <\ \text{RTP}_{\bar{P}} \ \le\ 1 - e = 94\%. \tag{21}$$

*Proof.* From (18) with $s \ge 0$, $M \le K/A$; rounding down and the cap only lower $M$. Hence $\bar{P}M \le \bar{P}K/A = (1-e)/A$, and $A \ge f_{\min}$ gives $(1-e)/A \le (1-e)/f_{\min} = 1 - e/3$. For $A = 1$ and $s \to 0$, $M_{\text{raw}} = (1-e)/\bar{P}$ and $\lfloor 100 M_{\text{raw}} \rfloor / 100 > M_{\text{raw}} - 0.01$, so $\bar{P}M > (1-e) - 0.01\,\bar{P}$. □

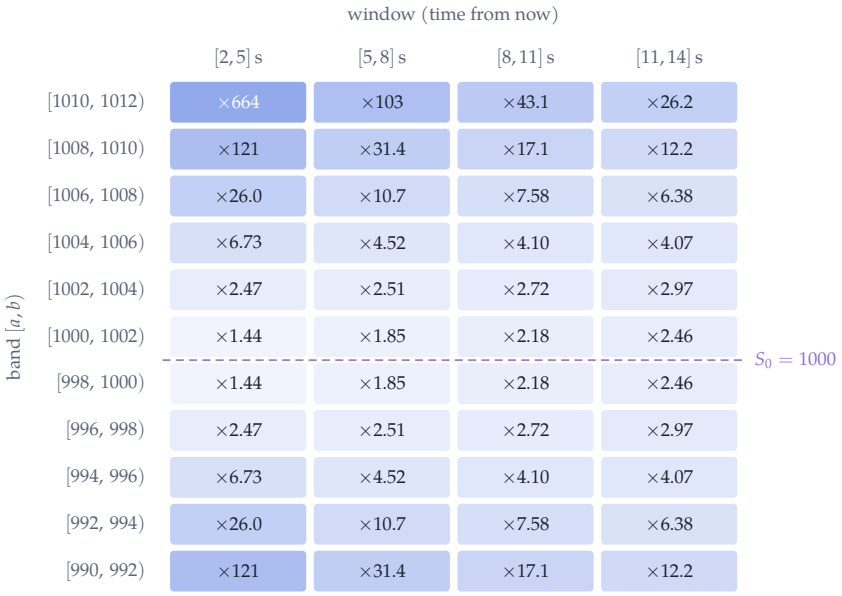



**Figure 6.** Multipliers quoted for an unloaded vault ($A = 1$): four 3-second windows (columns) and eleven 2-point bands (rows) around $S_0 = 1000$ (illustrative market). Bands next to the price in the nearest window pay about $\times 1.4$; long shots pay hundreds.

**Corollary 4** (Net of fees)**.** *With $f_i = f_o = 10\,bp$, an unloaded tile returns*

$$\mathrm{RTP}_{\bar{P}}^{\mathrm{net}} = \frac{\bar{P}\big(M - f_o(M-1)\big)}{1+f_i} = \frac{(1-f_o)\,\bar{P}M + f_o\bar{P}}{1+f_i} \in [93.8\%,\ 93.9\%]. \tag{22}$$

**Theorem 5** (House margin on the book)**.** *For any set of filled bets with stakes $s_i$, the vault's expected gross margin under the pricing probabilities, $\sum_i s_i(1-\bar{P}_i M_i)$, is at least $\frac{e}{3}\sum_i s_i = 2\%$ of turnover, equals $e\sum_i s_i = 6\%$ when all charges vanish, and grows with every charge.*

*Proof.* Sum Theorem 3 over bets: $1-\bar{P}_i M_i \geq 1-(1-e)/A_i \geq e/3$, with equality to $e$ when $A_i = 1$ (up to rounding, which only adds to the margin). □

**From the pricing measure to realised returns.** $\bar{P}$ is the probability under which the margin is exact. When volatility is uncertain in the way the pricer integrates over, $\bar{P}$ equals the true touch probability and the realised return converges to $\mathrm{RTP}_{\bar{P}}$. Figure 4(b) shows that, under the engine's settlement rule, $\bar{P}$ matches or slightly exceeds simulated frequencies, so realised returns sit at or a little below the bound. On the illustrative 60-bet book of Section 11 the model expects 93.9% and simulation returns 92.1%.

**Why a 24-hour figure moves.** The platform publishes the realised 24-hour return to player on the trading screen. Its sampling error over $n$ settled bets is

$$\mathrm{sd}\big(\widehat{\mathrm{RTP}}\big) = \frac{\sqrt{\sum_i s_i^2 M_i^2\,\bar{P}_i(1-\bar{P}_i)}}{\sum_i s_i}. \tag{23}$$

For 1,000 equal bets at $\times 3$ this is 4.4 percentage points, and at $\times 20$ it is 13.4; a single day can therefore print well above or below 94% without any change in pricing. Over many bets the realised figure converges to the bound.

## 6.3 A verifiable margin

The pricing core is a pure function: it reads no engine state, only the inputs stored with a bet (band, window, mark, variance-clock standard deviations, jump and noise terms, the uncertainty width, the calibration factor, the charge components, capacity and stake). These inputs are sealed into the trade record (Section 9). Any verifier can therefore recompute $P$, $\bar{P}$, $A$ and $M$ for any trade and check that

$$\bar{P}\,M\,A \;\leq\; 1-e, \qquad \text{with equality up to rounding and crowding,} \tag{24}$$

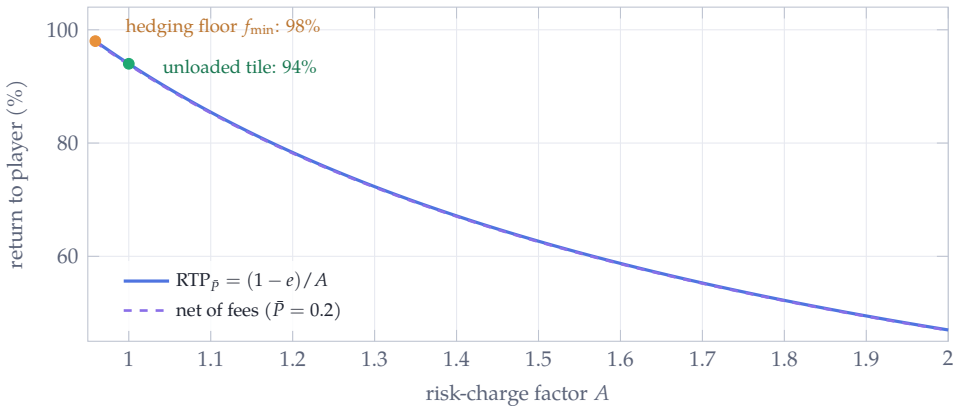



**Figure 7.** Return to player against the risk-charge factor. Every quote lies on or below the curve; the unloaded tile sits at 94%, a tile that hedges the vault can reach 98%, and charges lower the return only when the vault is concentrated.

which is a per-trade certificate that the published margin, and nothing else, separated the payout from the fair odds. The values of the charges are sealed; how they are derived from the state of the vault is not published.

**What this paper does not publish**

The coefficients and functional forms of the individual risk charges, the calibration state, the vault's internal limit settings beyond those stated in Section 11, and the internals of Spider's scoring beyond Section 12. These protect the vault against informed flow; none of them can raise a return above the bounds of Theorem 3, and every one of them is reflected in the sealed inputs of each trade.

# 7 Bespoke tiles: Pro

Pro lets a trader choose the window length $w$ and the band height $h$ (down to a single price tick) and quotes every tile of that shape. Bespoke tiles are priced by exactly the same function, (12)–(18), so they carry the same margin and the same verification.

**Proposition 6** (Monotone quotes)**.** *Let $\mathcal{T}_1 \subseteq \mathcal{T}_2$ be two tiles on the same tick grid (band and window inclusion). Then $P(\mathcal{T}_1) \le P(\mathcal{T}_2)$ for the tick rule (1), and the unloaded multipliers satisfy $M(\mathcal{T}_1) \ge M(\mathcal{T}_2)$ up to rounding.*

*Proof.* A larger window contains every tick of a smaller one, so the maximum over it is at least as large and the minimum at least as small; a larger band is reached whenever a smaller band inside it is. Hence $E_{\mathcal{T}_1} \subseteq E_{\mathcal{T}_2}$. The continuous-monitoring formula (12) is exact for its event and therefore inherits the ordering; the discrete correction (13) shifts both tiles' barriers by the per-tick standard deviation, which is equal for both when the variance rate is flat over the windows and differs only marginally otherwise. With $A = 1$, $M_0 = (1-e)/\bar{P}$ is decreasing in $\bar{P}$. □

A trader can therefore never buy the same exposure more cheaply by choosing a dominated shape in an unloaded vault. Charges that depend on the open book can break the ordering locally; they do so only by lowering the multiplier of the tile that loads the vault.

# 8 Fill after commit, and settlement

**Lockout.** A tile can be bought only if its window opens at least $\ell$ after the current time ($\ell = 1.5\,\mathrm{s}$ in the simulated market, 2 s for BTCUSD).

**Fill after commit.** A tap is first recorded as a commitment. The fill is the first mark strictly after the commitment is recorded, and the multiplier is the quote at that state. Neither side can see the fill price when the order is sent, so neither can time it: the house cannot choose a favourable tick, and a player cannot exploit a stale quote.

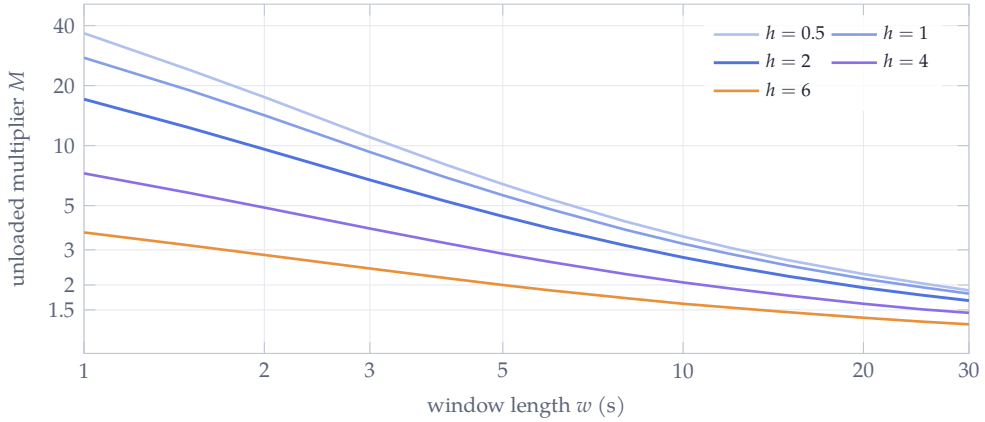



**Figure 8.** Pro pricing (illustrative market): a band centred 5 points above the price, with its window opening 2 s from now, for several band heights $h$ (points). A sniper who wants a precise level picks a short window and a thin band and is paid for the precision; longer windows and taller bands are touched more often and pay less, as Proposition 6 requires.

**Capacity.** Stakes beyond a tile's capacity are refunded, never filled at a worse price.

**Settlement.** On every tick inside the window the running minimum and maximum of the window are updated; the bet is won at the first tick at which (1) holds, paying $sM$ less the exit fee on the profit. A window that closes untouched transfers the stake to the vault.

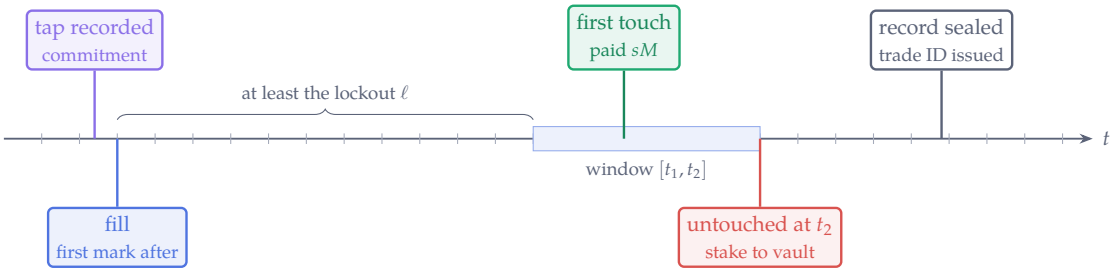



**Figure 9.** Life of a tap. The fill price is the first mark strictly after the commitment; settlement reads only the ticks inside the window; the record is sealed once the bet settles.

# 9 Cryptographic integrity and verification

## 9.1 Sealed trade records

When a bet settles, the platform assembles a record $r$ containing everything needed to reproduce it: participant and device identifiers, the commitment and fill times, the fill tick and price, the tile, the stake and fees, the sealed pricing inputs of Section 6.3, the multiplier, the price path of the window and the outcome. For automated participants it also contains the hash of the decision record that produced the bet (Section 12). The record is serialised canonically (keys sorted, fixed number formats) and hashed with SHA-256 [10]:

$$\mathrm{seal}(r) = \mathrm{SHA\text{-}256}\big(\mathrm{canon}(r)\big), \qquad \mathrm{id}(r) = \texttt{GT-}\,\mathrm{hex}\big(\mathrm{seal}(r)\big)_{[0:16]}. \tag{25}$$

The trade ID both names the trade and commits to it: changing any field of the record changes the seal and breaks the match with the ID that participants and observers already hold.

## 9.2 Hash-chained price recording

Every venue input and every mark is written to recording segments, each linked to the previous one:

$$h_k = \mathrm{SHA\text{-}256}\big(h_{k-1} \,\|\, \mathrm{seq}_k \,\|\, t_k \,\|\, \mathrm{mark}_k \,\|\, q_k \,\|\, \mathrm{SHA\text{-}256}(\mathrm{quotes}_k)\big). \tag{26}$$

Rewriting any past mark changes every subsequent link, so history cannot be edited without detection by anyone holding a later chain head or a trade ID whose record includes the affected path.

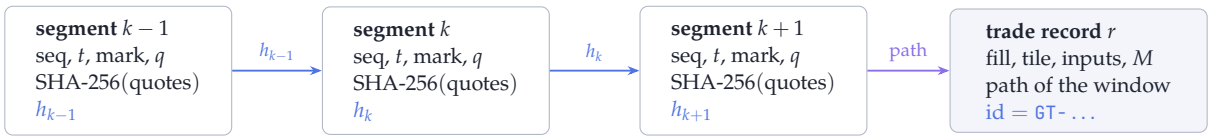



Figure 10. Tamper evidence. Each recording segment commits to its predecessor; each trade record commits to the window's path and to its own pricing inputs.

### 9.3 Verification

Verification needs only the record and the public recording. The six checks of Algorithm 1 are performed by the Verify tool for any trade ID, by anyone.

**Algorithm 1** Verify a trade

**Require:** trade ID $\iota$, record $r$, price recording $\{(t_k, \text{mark}_k, h_k)\}$

1: **Seal:** check GT-hex(SHA-256(canon($r \setminus \{\text{id}, \text{seal}\}$)))$_{[0:16]} = \iota$
2: **Path:** check every price in $r$'s window equals the recorded mark at the same tick, and the chain links verify
3: **Fill after commit:** check the fill tick is strictly after the commitment time
4: **Multiplier:** recompute $P, \bar{P}, A, M$ from the sealed inputs with the public pricing core; check $M$ matches and $\bar{P}MA \leq 1 - e$
5: **Settlement:** replay (1) over the recorded window; check the outcome and its tick
6: **Payout:** check payout $= sM$ less the exit fee if won, 0 otherwise
7: **return** all checks passed

**Security argument.** If SHA-256 is second-preimage resistant, no party can produce a different record with the same seal, nor a different recording consistent with a chain head already held by a third party. The fill rule removes the ability to choose the fill after seeing the price. The checks are deterministic and use only public code and data, so two independent verifiers always agree.

**Trust model.** The operator runs the server that produces records and recordings. Tamper evidence is relative to what third parties have observed: a trade ID, a record or a chain head saved by anyone fixes the history up to that point. Publishing chain heads to an external timestamping service strengthens this further and is planned.

## 10 Zero-knowledge pricing proofs (in development)

**Status**

This layer is in development and is *not* live in the beta. Today, fairness is established by the sealed records, the hash-chained recording and public recomputation described in Section 9.

Recomputation proves a quote correct by revealing its inputs, including the sealed charge values. The zero-knowledge layer aims to prove the same statement without revealing the private inputs, so that the house can keep its risk state private while every participant still receives a proof that the published rule was applied.

**Statement.** Let $x$ be the public inputs of a quote (tile, mark, variance-clock terms, margin $e$, multiplier $M$) and $y$ the private inputs (charge components, calibration factor, capacity). The platform publishes a commitment $c = H(y \,\|\, \rho)$ with randomness $\rho$ in the sealed record and a succinct proof $\pi$ of knowledge

of $(y, \rho)$ such that

$$c = H(y \,\|\, \rho) \quad \wedge \quad M = \mathrm{Price}(x, y) \quad \wedge \quad \bar{P}(x, y)\, M\, A(y) \le 1 - e, \tag{27}$$

where Price is the fixed-point arithmetisation of (12)–(18). A succinct non-interactive argument such as Groth16 [3] or PLONK [4] keeps verification fast for any participant. The Gaussian tail $\bar{\Phi}$ and the exponential are evaluated in the circuit by range-checked lookup tables with stated error bounds, and the bound on the return to player is proved directly, so a verifier learns that the quote respects Theorem 3 and nothing about the vault's book.

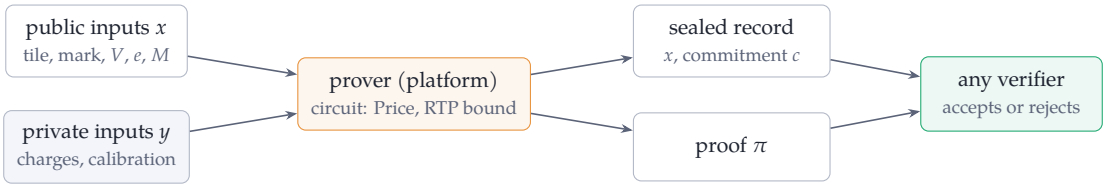



**Figure 11.** Design of the zero-knowledge layer (in development): a verifier checks that a quote follows the published rule and the return bound, without learning the private risk inputs.

# 11 The vault and its risk budget

## 11.1 Accounting

Each market has a vault that takes the other side of every tile. Its net asset value is cash less the expected payout of the open book, marked at current probabilities:

$$\mathrm{NAV}_t = \mathrm{cash}_t - \sum_{i \in \mathrm{open}} s_i\, M_i\, \bar{P}_i(t). \tag{28}$$

Vault tokens are minted and burned at the token price $p_t = \mathrm{NAV}_t / N_t$: a deposit $D$ mints $D / p_t$ tokens, and a withdrawal of $n$ tokens pays $n\, p_{t'}$ at its release time $t'$, 48 hours after the request, so that open bets remain covered. A token holder's share of every result is their fraction of tokens outstanding.




stake 454 mean payout 418.1 $\mathrm{VaR}_{99}$ 2170.4 $\mathrm{ES}_{99}$ 2850.5 return 92.1% (model 93.9%)

**Figure 12.** Distribution of the payout of an illustrative open book of 60 bets across four windows (200,000 simulated paths, engine settlement rule, uncertain volatility). The distribution is concentrated near zero with a long right tail; the vault's budget constrains its 99% quantile, not its mean.

## 11.2 Limits

Every limit scales with NAV: the largest stake and payout of a single bet, the open payout of one participant, the open payout on one tile, the number of open tiles per participant (scaling with $\sqrt{\mathrm{NAV}}$), and the open payout on each side of the price, capped at 20% of NAV. Stakes above a limit are refunded, never filled at a worse price.

### 11.3 Monte Carlo value at risk

Hits on the open book are strongly dependent: one price path decides all of them. The vault therefore simulates price paths on the variance clock, with volatility drawn from the same uncertainty law the pricer integrates over and with jumps, settles the whole book on each path with the tick rule, and estimates the 99% value at risk and expected shortfall of the book's payout. The 99% value at risk is kept within a budget of 8% of NAV; tail scenarios also set the payout capacity of each tile, and the use of the budget enters the capacity charge.

## 12 Spider: a Markov-chain agent with a verifiable decision trace

Spider is an automated strategy that a participant can run with its own bankroll, stop-loss, take-profit, trade-rate and open-tile caps. It sees only what a player sees, namely the tape, one-minute bars and the multipliers on screen; it has no access to the vault's pricing, inventory or limits. It must therefore form its own view of each tile's touch probability and decide whether the multiplier pays for it.

### 12.1 Model

**Regimes.** One-minute bar returns are classified into three states (down, flat, up), split at $\pm 0.43$ standard deviations so that each state is a tercile under normality. The transition matrix is estimated with add-one smoothing,

$$\hat{p}_{ij} = \frac{n_{ij} + 1}{n_i + 3}, \qquad i, j \in \{\downarrow, \rightarrow, \uparrow\}, \tag{29}$$

and retrained every 60 seconds on the last 30 bars. The drift is the transition-weighted mean of the state means from the current state, shrunk towards zero in the manner of James and Stein [7], because a 30-bar mean is mostly noise. The variance is the transition-weighted mix of per-state variances, blended with a short exponentially weighted variance of tick returns.

**Variance-aware Monte Carlo.** Each scan simulates $N$ paths. Every path draws its own volatility from a log-normal law around the blended estimate, with dispersion equal to the measured volatility of volatility of the bars, and then walks Student-$t$ ($\nu = 5$) steps. Touches are checked against each band widened by the discrete-monitoring shift $0.5826\,\sigma(\sqrt{h_{\mathrm{MC}}} - \sqrt{\Delta})$, where $h_{\mathrm{MC}}$ is the simulation step and $\Delta$ the game's tick.

**Decision rule.** For each visible tile, $\hat{P} = (\text{hits} + \tfrac{1}{2})/(N + 1)$, the expected value per unit stake is $\mathrm{EV} = \hat{P}\, M_{\mathrm{net}}/(1 + f_i) - 1$ with standard error from $\hat{P}$, and tiles are ranked by $\mathrm{EV} - k\,\mathrm{se}$ to counter the winner's curse. Spider bets on tiles that clear a threshold set by the participant's risk level, within a multiplier band and the participant's caps.

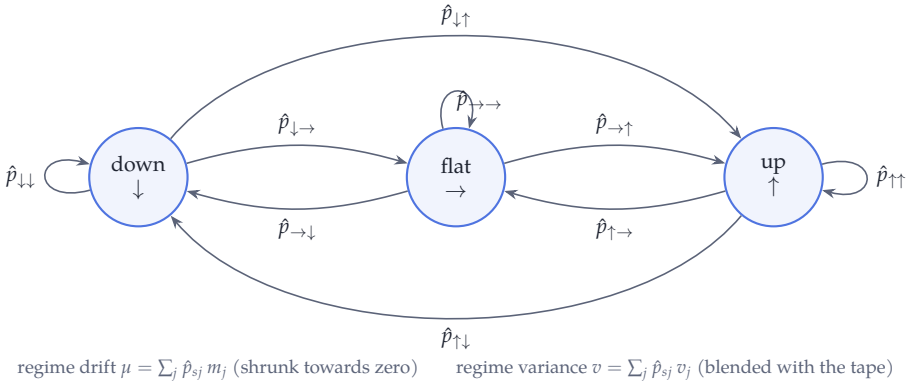



Figure 13. Spider's three-state Markov chain on one-minute bar returns. The row of the current state $s$ weights the state means and variances that drive the Monte Carlo scan.

### 12.2 A verifiable decision trace

Every decision that places bets produces a decision record, Spider's "chain of thought" made auditable:

$$\theta_n = \big(\text{bot},\ n,\ h_{n-1},\ t,\ \text{params},\ \text{params version and hash},\ \text{model view},\ \text{market terms},\ \text{Monte Carlo size},\ \text{rules},\ \text{ranked candidates},\ \text{picks},\ \text{skips}\big), \tag{30}$$

with $h_n = \text{SHA-256}(\text{canon}(\theta_n))$. Each record names the hash of the previous one, so the trace is a chain; each bet placed by the decision seals $h_n$ into its trade record; and every change to Spider's parameters is itself appended to a hash-linked parameter log that the decision record references. The model view records the regime, the transition row, the variances used and the drift; the ranked candidates record each tile's estimated probability, expected value and standard error, and why it was taken or skipped.

**Proposition 7** (Integrity of the trace)**.** *Assuming* SHA-256 *is second-preimage resistant, no decision can be inserted into, removed from or reordered within a Spider's trace, and no trade can be attributed to a decision other than the one that produced it, without detection by Verify.*

*Proof.* Changing, inserting or removing $\theta_m$ changes $h_m$ and therefore the $h_m$ stored in $\theta_{m+1}$, which changes every later hash. A trade record seals $h_n$, so attributing it to another decision breaks the record's own seal. Parameter changes are covered in the same way by the parameter log. □

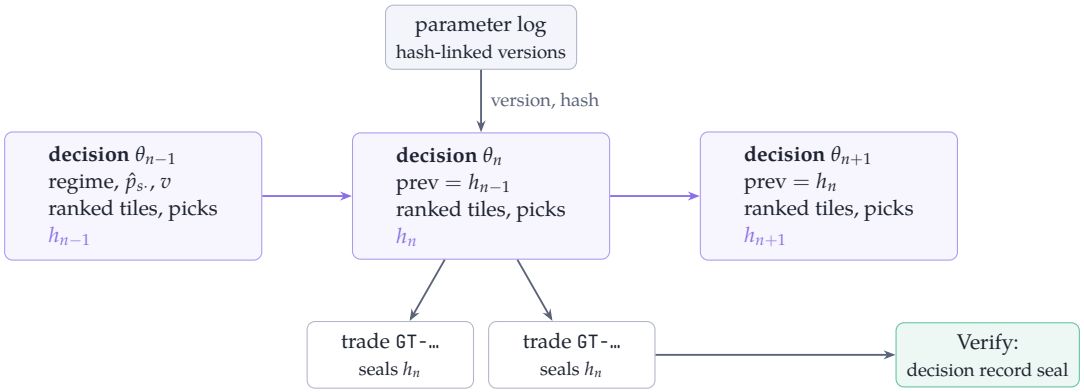



Figure 14. Spider's verifiable decision trace. Decisions form a hash chain; each trade seals the hash of the decision that produced it; parameter changes are chained separately and referenced by version and hash.

The trace makes Spider's reasoning reviewable and its integrity provable. The Monte Carlo draws themselves are not published, so a decision can be checked for integrity and internal consistency but not replayed path by path; this keeps strategies from being front-run.

## 13 Limitations

- **Model risk.** The touch probability assumes Gaussian increments on a variance clock over seconds, corrected for discrete monitoring, jumps and volatility uncertainty. The discrete-monitoring correction is asymptotic and is least accurate for very short windows on very distant bands (an overestimate of up to about 8% in relative terms in our tests), which is conservative for the vault. Realised returns depend on how closely real volatility follows the forecast.
- **Operator trust.** Records and recordings are produced by the operator. Tamper evidence holds relative to anything a third party has observed; external anchoring of chain heads is planned, and the zero-knowledge layer is in development.
- **Data dependence.** The live mark depends on four exchanges and their feeds. Feed gaps void and refund every affected window rather than settling on a held price.
- **Beta.** Parameters, limits and features may change between releases. Balances are practice USDO with no cash value.

# A Proof of the forward-start touch formula

Let $W$ be the price process on the variance clock, a Brownian motion in integrated variance. By the Markov property at $t_1$, conditionally on $X = S_{t_1} = x$ the window path is a Brownian motion started at $x$ and run for variance $s_2^2$. If $a \le x \le b$, the band contains the starting point and the event holds at once. If $x < a$, a continuous path touches $[a, b]$ if and only if its maximum reaches $a$, which by the reflection principle has probability $2\bar{\Phi}((a - x)/s_2)$; the case $x > b$ is symmetric with the minimum and the barrier $b$. Taking expectations over $X \sim \mathcal{N}(S, s_1^2)$ gives (12). Replacing the barrier $a$ by $a + \beta s_\Delta$ (and $b$ by $b - \beta s_\Delta$) yields the discrete-monitoring approximation (13) of [1], whose error is $o(\sqrt{\Delta})$. □

# B Parameters used in the illustrative figures

| | | | |
|---|---|---|---|
| price $S_0$ | 1000 | volatility $\sigma$ | 1.39 points per $\sqrt{\text{s}}$ |
| tick $\Delta$ | 0.5 s | lockout $\ell$ | 1.5 s |
| window $w$ | 3 s | band $h$ | 2 points |
| margin $e$ | 6% | vol-uncertainty width $\eta_\sigma$ | 0.15 |
| fees $f_i, f_o$ | 10 bp each | multiplier cap | 5000 |

These match the simulated market (SYNTH). Data for every figure is produced by the same formulas as the pricing engine; Monte Carlo figures use the engine's settlement rule.

# References


[1] M. Broadie, P. Glasserman and S. Kou. A continuity correction for discrete barrier options. *Mathematical Finance*, 7(4):325–349, 1997.

[2] J. Gatheral and A. Jacquier. Arbitrage-free SVI volatility surfaces. *Quantitative Finance*, 14(1):59–71, 2014.

[3] J. Groth. On the size of pairing-based non-interactive arguments. In *EUROCRYPT 2016*, LNCS 9666, pp. 305–326.

[4] A. Gabizon, Z. J. Williamson and O. Ciobotaru. PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. *IACR ePrint* 2019/953.

[5] S. Hendriks and C. Martini. The extended SSVI volatility surface. *Journal of Computational Finance*, 22(5), 2019.

[6] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6(2):327–343, 1993.

[7] W. James and C. Stein. Estimation with quadratic loss. In *Proc. Fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 361–379, 1961.

[8] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[9] I. Karatzas and S. E. Shreve. *Brownian Motion and Stochastic Calculus*, 2nd ed. Springer, 1991.

[10] National Institute of Standards and Technology. *Secure Hash Standard (SHS)*, FIPS PUB 180-4, 2015.

[11] L. Zhang, P. A. Mykland and Y. Aït-Sahalia. A tale of two time scales: determining integrated volatility with noisy high-frequency data. *Journal of the American Statistical Association*, 100(472):1394–1411, 2005.

# One Tap Trade

Tap a tile. Trade the volatility.

onetaptrade.xyz

One Tap Trade is in beta. Balances are held in USDO, a practice unit with no cash value.
Nothing in this paper is an offer of a financial product.