StayLiquid v2 — What Changed, and Why It's Safer

Last updated: 30 Jul 2026

1. The short version

v1 was a harvesting bot that **trusted its own database** and let the AI act on almost anything. It made small gains and then gave them all back, because the numbers it was reasoning about were sometimes wrong and there was no hard floor under the AI's decisions.

v2 is built on three rules:

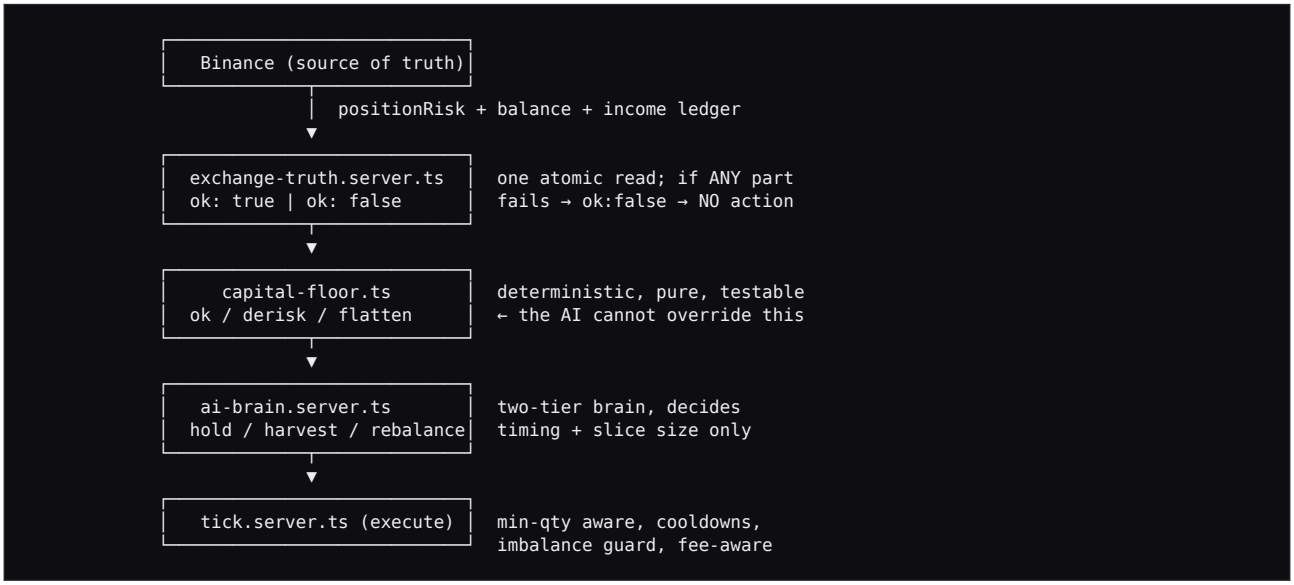
1. **Binance is the truth.** Our database is only a cache.
2. **A deterministic safety floor sits below the AI.** The AI may always be safer than the floor. It can never be riskier.
3. **The AI is a tactician, not a god.** It decides *when* and *how much*, inside limits it cannot argue its way out of.

2. What actually broke in v1

Problem	What the trader saw
DB drift	Trade "completed" with the wrong PnL; real profit sat on Binance, unreported
Wrong liquidation price	Dashboard showed a liq level Binance didn't agree with
One leg auto-closed	Hedge silently became a naked directional position
Over-trimming loop	The winning leg was trimmed to nothing, leaving only the loser
Fee blindness	Harvests below the fee line — structurally negative yield
Pending gains "lost"	8–9 USDT of unbooked profit not written on stop

Every one of these is a *data integrity* failure, not a strategy failure. That's what v2 attacks first.

3. The v2 architecture



3.1 Exchange Truth (`src/lib/exchange-truth.server.ts`)

One call returns everything the engine is allowed to reason about: mark price, real leg quantities, real entry prices, **real Binance liquidation prices and distances**, unrealized PnL per leg, wallet/margin/available balance, and the **income ledger** (realized PnL, commission, funding) for that symbol since the bot started.

If any part of that read fails, it returns `{ ok: false }` and the engine takes **no action at all**. v1's worst bugs came from acting on a half-read snapshot.

3.2 Capital Floor (`src/lib/capital-floor.ts`)

Pure functions, no I/O, unit-testable. Four hard rules:

Rule	Trigger	Action
`liq_floor`	Either leg within **3%** of liquidation	de-risk immediately
`neutrality_cap`	Net exposure exceeds **35%** of total size	force back to neutral
`equity_stop`	Net PnL (realized + unrealized − fees) breaches the bot's max drawdown	flatten
`broken_hedge`	One leg is gone / hedge structurally invalid	flatten

Net PnL is always **realized + unrealized − fees**. Fees are never ignored again.

3.3 Two-tier AI brain (`src/lib/ai-brain.server.ts`)

- **FLAGSHIP —** `openai/gpt-5.6-sol`**.** Best reasoning available. Runs when risk is elevated, volatility spikes, the hedge is skewed, or a harvest is on the table.
- **SCOUT —** `google/gemini-3.1-flash-lite`**.** Cheap and fast, used only for quiet ticks where nothing is happening.

Tier is chosen per tick from the risk state, so you pay flagship prices only for decisions that matter. A quiet tick costs roughly **$0.0005**; a whole trade typically lands in the **$0.01–$0.03** range of raw

model spend.

3.4 Execution guards (`src/lib/tick.server.ts`)

- Price sanity guard (rejects >40% deviation ticks)
- Two-tick confirmation before target-hit close
- Harvest cooldown, so no runaway trim loop
- Imbalance guard — refuses a trim that would break leg symmetry
- Min-qty awareness — waits for the slice to clear Binance's `minQty` instead of forcing an oversized harvest
- Reconciliation from the Binance income ledger, plus a manual ↻ **sync** button

3.5 Proof before money (`/dashboard/proof`)

`backtest-real.server.ts` replays **real Binance candles** through the exact same floor + brain code path. You can see expectancy after fees *before* risking capital. v1 had no honest way to answer "does this actually make money?"

3.6 Fees, simplified

One line item: **AI fee = 1% of realized profit**, floored at the bot's raw model spend so a profitable trade can never cost the platform money. **No profit → fee is $0.00.** The old 5.5% performance fee is gone — it double-charged subscribers and scaled absurdly on large wins.

4. v1 vs v2 at a glance

	v1	v2
Source of truth	Local database	Binance (`positionRisk` + income ledger)
Liquidation price	Estimated from price	Real value from the exchange
Safety limits	Scattered in the AI prompt	Deterministic floor below the AI
AI model	One cheap model	Flagship + Scout, risk-routed
Partial reads	Acted anyway	Refuses to act
Fee accounting	Partial	Commission + funding in every decision
Backtesting	None	Real-candle replay on the same code path
Fee model	5.5% + compute	Single 1% AI fee on profit
Broken hedge	Silent	Circuit breaker + flatten

5. "The AI decides everything and that's the best" — the honest answer

Half right, and the half that's wrong is the important half.

Where you're right: for *timing and sizing*, the AI genuinely is better than a human. It doesn't get bored at 3am, doesn't revenge-trade, doesn't move a stop because it "feels" the coin will come back, and it applies the same discipline on tick 4,000 as on tick 1.

Where "AI decides everything" is the wrong design:

1. **LLMs are non-deterministic.** The same snapshot can produce a slightly different answer twice. That's tolerable for "harvest 8% or 11%". It is not tolerable for "are we 2% from liquidation?" — that must be arithmetic.
2. **v1 lost money precisely because judgement had no floor.** The trims were individually defensible; nothing stopped the sequence.
3. **An LLM can be wrong about numbers.** So v2 never asks it to compute risk — it gives the AI numbers already verified against the exchange.
4. **Latency and cost.** A hard 3%-to-liquidation exit cannot wait on a model round-trip.

So the correct framing, and what v2 implements:

> **Arithmetic decides what is forbidden. AI decides what is optimal inside what is allowed.**

That is stronger than "AI decides everything", not weaker. The AI still drives the trade — it just can't drive it off a cliff.

6. Current state

- `kill_switch`: **off**
- `live_trading_enabled`: **on**
- Recommended next step: run `/dashboard/proof` on the pair you intend to trade and read expectancy **after fees** before sizing up.