

Entwickler-Bewertung – Steve-TradingBot Zwischen-Auswertung

Quelle: MONITOR_REPORT_20260517T221634Z.zip **Stamp:** 2026-05-17 22:16 UTC, Bewertung 2026-05-18 00:30 UTC **Bewerter:** technischer Code- + Daten-Audit **Boundaries:** Bot läuft uninterrupted; kein Eingriff während C1-Monitoring außer kritischem Risk.

1. Executive Summary

In 16 h Session-Laufzeit hat der Bot über 4 SOT-1d-Cutover (A1 / B / C1 / D + SL-TP-Hotfix) 0 Tracebacks produziert. Code-architektonisch ist Phase A1 (Tier-Contract) plus B (Binance-Pair-Gate) sauber gewired. Phase C1 sammelt mit 509 BUY-Samples brauchbare Quality-Shadow-Daten (avg 0.575, sd 0.055) — die im Plan v3.4 vorgeschlagene 0.65-Schwelle wäre mit nur 11 % Pass-Rate zu eng. Phase D BEAR-DCA-Block hat live 2x gefeuert und korrekt verhindert dass HUMA/TAO über DCA refinanziert wurden. Trading-Bilanz heute: **-56.67 USDT realized** (zwei Stop-Loss-Closes), 5 offene Positionen bei -1.90 % bis +0.01 %. Mehrere **Datenqualitäts-Defects** sind aufgedeckt: DCA-State auf File-Ebene reflektiert nicht alle Events, entry_price ist semantisch Average-nach-DCA, GUI zeigt das ohne Hinweis. Event-Counter inkonsistent (6 Stop-Loss-Trigger-Log-Hits aber nur 2 closed trades). Das 4 %-Wochenziel ist **derzeit nicht messbar** weil mehrere KPIs (Trefferquote, MAE/MFE, Capital-Time, DCA-Erfolgsquote) entweder nicht erhoben werden oder durch DCA-State-Inkonsistenz unzuverlässig sind.

Ampelstatus

Domäne	Status	Begründung
Stabilität	🟢 grün	0 Tracebacks über 4 Cutover; healthy main.py; Worker-Recreate ohne Bot-Touch
Risiko	🟡 gelb	Phase D arbeitet, aber EXTERNAL-CHANNEL-CAP-ALIGN noch offen (Cap kann übergangen werden); kein Drawdown-Kill-Switch sichtbar
Performance	🟡 gelb	-0.53 % auf Equity (10 645 → 10 588 nach 2 SL-Closes); zu wenig Sample-Volume für „4 % / Woche“-Aussage
Datenqualität	🔴 rot	DCA-State-Persistence-Gap, entry_price-Semantik unklar, GUI ohne avg/tranches, Event-Counter inkonsistent
GUI-Transparenz	🟠 orange	Polish-1 hat Sections + Back-Button, aber kein avg_price / DCA-Count / Tranches; nur ein Auszug der Wahrheit

2. Bewertung der Zwischen-Auswertung

2.1 Offene Positionen (CSV 01, 5 Rows)

ENA/USDT	entry 0.10767	cur 0.10630	PnL -6.80 USDT (-1.28%)	SL-Buf 0.96 %
SHIB/USDT	entry 5.92e-6	cur 5.81e-6	PnL -7.68 USDT (-1.90%)	SL-Buf 0.69 %
TRUMP/USDT	entry 2.16300	cur 2.15700	PnL -0.59 USDT (-0.28%)	SL-Buf 1.22 %
XAUT/USDT	entry 4545.91	cur 4546.50	PnL +0.03 USDT (+0.01%)	SL-Buf 0.21 %
XLM/USDT	entry 0.15160	cur 0.15110	PnL -0.70 USDT (-0.33%)	SL-Buf 0.79 %
Unrealized total			-15.74 USDT	
Position-Wert			1 555.83 USDT	

- **Plausibilität:** Pricing konsistent, PnL-Math stimmt im Display (nach gui-positions-pnl-pct-display-fix).
- **Critical:** SHIB SL-Buffer **0.69 %** und XAUT **0.21 %** — extrem knapp. Beide tradierbar bei nächstem Bear-Schub.
- **Anomalie ENA:** entry 0.10767 ist laut Code (paper_trade.py:507) der post-DCA-Average. Aus dem bot_log erwartet wäre 0.1077 nach 2 DCAs — passt zu Floating-Point-Rundung. **Aber** keine Aufzeichnung der ursprünglichen Initial-Entry-Quote → Operator weiß nicht, dass diese Position 2x nachgekauft wurde.
- **Anomalie XLM:** entry 0.15160 ist identisch zum allerersten Buy. Im bot_log gab es DCA bei 0.1510 mit neuem Avg 0.1524. **Im State steht das nicht** — XLM-DCA wurde **nicht persistiert** (oder zurückgerollt durch State-Reload).

2.2 Closed Trades (CSV 02, 2 Rows)

HUMA/USDT	exit_reason=legacy_stop_loss	pnl -52.48 USDT (-6.11%)	opened 10:52	closed 19:45
TAO/USDT	exit_reason=legacy_stop_loss	pnl -4.19 USDT (-1.97%)	opened 05:42	closed 22:08
Realized total session		-56.67 USDT		

- Beide Closes mit legacy_stop_loss — das ist der **LABEL-1-Fallback** für Positionen ohne _label1_started=True -Marker. HUMA/TAO waren beide neu eröffnet nach LABEL-1-Cutover → der Marker hätte gesetzt sein sollen. **Defect:** irgendwo wird der Marker nicht zuverlässig gesetzt (vermutlich External-Channel-Web-Buy oder Initial-Buy-Pfad).
- TAO-Loss -1.97 % ist atypisch klein für einen ATR-SL → vermutlich war SL bereits dicht am Entry (mglw. nach Trailing). Da trailing_events=0 in CSV 07, wahrscheinlich nicht.
- **duration_seconds NULL** → SNAPSHOT-EMIT-COMPLETENESS-Bug; das Feld wird im close-emit nicht berechnet.

2.3 Decision-Distribution (CSV 03)

evaluating	t1_core	run_scan_cycle	BEAR	4 499
reject	t1_core	run_scan_cycle	BEAR	3 955
buy	t1_core	run_scan_cycle	BEAR	544
total				8 998
reject-Rate				87.9 % (gesund konservativ in BEAR)
buy/eval-Rate				12.1 %

- 100 % t1_core — TIER-ARCH-CONTRACT-1 hält.
- 100 % BEAR-Regime — keine Regime-Variation in 16h. Sample limitiert auf BEAR-Bias.
- 544 buy-decisions vs. 7 tatsächliche Position-Opens (5 jetzt offen + HUMA closed + TAO closed = 7) → **emit_decision feuert pre-cap und pre-sizer**. Discrepancy 537 ≈ 99 % der BUYs landen NICHT als Trade. Das ist im Wesentlichen die **EXTERNAL-CHANNEL-CAP-ALIGN**-Problematik plus reguläre Cap-/Sizer-Rejects.

2.4 Quality-Shadow (CSV 04, 509 BUYS)

n=509	avg=0.575	stddev=0.055	min=0.459	max=0.723
Pass-Rate bei				
0.50:	465/509	=	91.4 %	
0.55:	327/509	=	64.2 %	
0.60:	161/509	=	31.6 %	
0.65:	55/509	=	10.8 %	

Befund: Threshold 0.65 (Plan v3.4 default) wäre mit 10.8 % Pass-Rate zu eng — der Bot würde ~90 % seiner Setups blocken. Operator-relevanter Korridor: - **0.55** als balanced (64 %) - **0.60** als selektiv (32 %) - **0.65** als „Top-Decile-only“

Empfehlung für Phase C2: **0.575 ± 1σ** = Threshold-Band 0.52 ↔ 0.63 mit regime-spezifischer Abstufung. Konkreter Vorschlag (auf BEAR-Daten): C2 startet bei **0.55** für Test, kann tighter ziehen wenn Pass-Rate > 70 % bleibt.

2.5 Limiting Subscores (CSV 05)

mtf_alignment_score	184	38 %
volume_burst_score	141	29 %
regime_fit_score	60	12 %
no_trade_zone_score	42	9 %
trend_quality_score	27	6 %

- **mtf_alignment_score** ist Single-TF-Proxy (RSI + MACD bullish) → in einer BEAR-Phase praktisch immer schwach → korrekt limitierend, aber nicht echte „Multi-Timeframe“-Information. Sub-Score-Funktion ist **bewusst grob** lt. Code-Kommentar (quality_score.py: score_mtf_alignment). **Verbesserung:** echtes 4h/1d-Confirmation einbauen würde Vorhersagekraft erhöhen.
- **volume_burst_score** zweitstärkster Limiter — sinnvoll, weil schwaches Volume in BEAR oft False-Breakouts produziert.

2.6 Phase Tracebacks (CSV 06)

5 Cutover, **0 Tracebacks** seit jeweiligem Cutover. Stabilität top.

2.7 Bot Events (CSV 07)

T3_bridge_archive_log	5	(= 5 Bot-Prozess-Restarts heute; korrekt, log-once)
risk_guard_blocks	2	(HUMA + TAO BEAR-DCA-Blocks live verifiziert)
dca_rescue_attempted	0	(= 0 Logs ' DCA-Rettung' = 0 tatsächliche Rescues; korrekt da geblockt)
dca_signal_emitted	2	(Manager hat DCA proposed, RISK-GUARD hat geblockt)
stop_loss_triggered	6	△ aber nur 2 closed_trades
trailing_events	0	(kein Trade hatte 2 % Gain → kein Trigger; konsistent)

Inkonsistenz: 6 ● Stop-Loss getriggert vs. 2 closed trades. Mögliche Ursachen: - SL-Check trifft, aber DCA-Rescue blockt nicht das log-statement, sondern fällt drüber → log fires aber Close nicht (4 davon) - Mehrfacher Stop-Loss-Log-Print für gleichen Trade - Log-Grep matched auch Test-Strings

→ **EVENT-COUNTER-CONSISTENCY-CHECK P1:** forensisch klären.

3. Bewertung im Verhältnis zum Ziel „ca. 4 % Profit pro Woche“

3.1 Aktueller Stand vs. Ziel

Session-Start Equity	~10 700 USDT
Aktuell Equity	~10 588 USDT (Cash 9 032 + Pos-Wert 1 556)
Realized this session	-56.67 USDT
Unrealized	-15.74 USDT
Session-PnL	-72.41 USDT ≈ -0.68 %

Bei 4 %/Woche entspricht das einem Ziel von **+428 USDT/Woche** auf 10 700 Basis. Aktuelle 16h-Bilanz: **-72.41 USDT** = -0.68 %. **Linear projiziert** wäre das -7.1 %/Woche — aber Linearprojektion ist hier statistisch unzulässig.

3.2 Ist das Ziel mit aktueller Strategie realistisch messbar?

Nein, derzeit nicht. Probleme: 1. **Sample-Volumen zu klein**: 2 closed trades in 16h → keine Statistik möglich 2. **Ein Regime**: 100 % BEAR im Sample → keine Bull-/Range-Phase zur Vergleichbarkeit 3. **DCA-State unzuverlässig**: avg-Entry-Anzeige inkorrekt → PnL-Berechnung kann auf Tranchen-Ebene falsch sein 4. **Cap-Bug**: EXTERNAL-CHANNEL erlaubt 6/5 Positionen → Risk-Profile nicht reproduzierbar

3.3 Welche KPIs fehlen?

Aus Operator-KPI-Liste:

KPI	aktuell verfügbar	wie messen?
Wochen-PnL	✓ via trade_logs realized + unrealized	aggregieren, jetzt machbar
Max Drawdown	✗ nicht persistiert	neuen Equity-Snapshot pro Cycle in DB
Trefferquote	✗ (nur 2 Trades, beide Loser)	≥30 Trades nötig für Aussage
Avg Win / Avg Loss	✗	derselbe Daten-Mangel
Risk/Reward	✓ pro Trade via SL/TP berechenbar	aus trade_logs derivierbar
Exposure-Zeit	teilweise	duration_seconds NULL → SNAPSHOT-EMIT-COMPLETENESS-Fix nötig
Stop-Loss-Häufigkeit	✓ via exit_reason	gegeben, aber sample n=2
DCA-Erfolgsquote	✗ State-Inkonsistenz	erst nach DCA-STATE-RECONCILE
Risk-Guard-Blockrate	✓ via log-grep	aktuell 2 / 2 Block-Events = 100 % BEAR-Block
Buy-Decisions vs. Orders	✓ via decision_logs vs. trade_logs	544 buys / 7 orders = 1.3 %
Kapitalbindung pro Position	✓ via position_snapshots.position_value	gegeben
Netto-Performance nach Fees	✓ trade_logs.fees + realized_pnl	gegeben
Stabilität über Marktphasen	✗ einseitig BEAR	Daten aus 2+ Wochen mit Phasenwechsel nötig

3.4 Datenbasis vor Aktivierungs-Optimierung

Minimum für 4 %-Ziel-Bewertung: 30 closed trades über min. 2 unterschiedliche Regimes (BEAR + RANGE/BULL). Bei aktuellem Rate 7 trades / 16h → ~7 Tage. Mit BEAR-Bias und Risk-Guard-Blocks tendiert das eher zu 10-14 Tagen.

4. Technische Befunde

4.1 DCA-State-Inkonsistenzen

Code-Fakt (paper_trade.py:507):

```
pos['entry_price'] = round_price(new_cost / new_qty, price) # Neuer Durchschnittspreis
```

→ live_portfolio.json.entry_price ist **nach DCA der gewichtete Durchschnitt**, nicht der Initial-Entry.

Consequence: Operator sieht in /admin/positions einen entry_price, der semantisch wackelt: - bei Position ohne DCA: == Initial-Entry - bei Position mit DCA: == aktueller Average - bei Position mit DCA aber State-Verlust (XLM): == Initial-Entry (DCA-Wirkung verschwunden)

Es gibt keine GUI/State-Anzeige für **Initial-Entry + Tranchen-Liste** als Zusatz.

4.2 State-Persistence-Risiken

Beobachtet: XLM hatte bestätigte DCA-Rettung 2026-05-16 07:14, aber dca_log.json zeigt dca_count=0 und live_portfolio.json entry_price=0.1516 (Initial), nicht 0.1524 (post-DCA-Avg).

Mögliche Ursachen: 1. _save_state() Race nach DCA und vor Crash/Restart 2. mtime-cookie -Reload (HISTORY-1 Fix B) hat externe Mutation als „neuer“ Zustand interpretiert 3. KITE-Repair-File-Edit vom 2026-05-16 hatte unbeabsichtigte Cross-Position-Effekte

Forensik vor Code-Fix nötig → **DCA-STATE-RECONCILE P1.5** (Plan existiert).

4.3 trade_logs-Limitierungen

trade_logs enthält nur closed_trades (status='closed'). Für offene Positionen → keine Quelle. Die DCA-Tranchen sind nicht abrufbar. Für GUI / Analytics ist trade_logs nicht alleine ausreichend.

4.4 position_snapshots-Nutzung

Time-Series-Log mit per-scan-tick (~2 min). Pro offener Position wachsen die Rows linear (23 218 rows aktuell). Per-scan-emit (main.py:448) trägt nur **volatile** Felder (current_price, unrealized_pnl, position_value) — die statischen Identifier (decision_id,

opened_at, strategy_id, profile_id, risk_reward, metadata) bleiben leer auf den Snapshot-Rows. → SNAPSHOT-EMIT-COMPLETENESS-Backlog.

4.5 Adminpanel-/GUI-Schwächen

- PositionSnapshotResource zeigt **post-DCA-Average** als „entry_price“ ohne Hinweis
- Kein avg_price Feld
- Kein dca_count / tranches -Liste
- Kein Initial-Entry-Differential-Anzeige
- unrealized_pnl_pct jetzt korrekt nach gui-positions-pnl-pct-display-fix (×100)
- View-Position-Detail hat 5 Sections + Back-Button nach gui-view-position-polish-1
- **Aber:** viele Felder NULL weil per-scan-emit dünn (SNAPSHOT-EMIT-COMPLETENESS)

4.6 Event-Counter-Inkonsistenzen

```
stop_loss_triggered 6 vs. closed_trades 2
dca_signal_emitted 2 vs. risk_guard_blocks 2 (passt)
```

6 → 2 ist plausibel wenn: - DCA-Rescue versucht, blocked, log printed, dann fall-through zu SL-Trigger → 2× Log pro Close (DCA-Signal + SL-Triggered) - Multiple Update-Cycles vor finaler Close-Execution

→ **EVENT-COUNTER-CONSISTENCY-CHECK P1:** forensisch klären, Counter sauber benennen.

5. Risikoanalyse

Risiko	Severity	Wahrscheinlichkeit	Mitigation
Trading: SL nicht eingehalten, weil entry_price nach DCA falsch interpretiert	hoch	mittel	Code-Fakt: SL ist im State korrekt, nur Display zeigt avg als entry. Trading-Logik nicht betroffen, PR-Anzeige + Operator-Decision-Basis aber schon
Datenqualität: GUI zeigt avg, Operator denkt Initial-Entry	hoch	hoch	Quick: GUI-Hinweis-Label „avg after DCA“; Full: separates Feld initial_entry + tranches
Operator-Risk: Operator setzt SL/TP auf Basis falschen entry_price-Verständnisses	mittel	mittel	Tooltip / Section-Header in View-Position; Backlog GUI-VIEW-POSITION-POLISH-2
State-Persistence: XLM-ähnliche DCA-Drops in Zukunft	mittel	mittel	DCA-STATE-SAVE-RACE-FIX nach Forensik
PnL-Berechnung falsch: avg-Entry-Inkonsistenz beeinflusst unrealized_pnl-Display	niedrig	hoch	unrealized_pnl wird live aus current+entry+qty berechnet — wenn entry korrupt (XLM), ist Pnl falsch. Operator-relevant für Risk-Decision
EXTERNAL-CHANNEL-CAP-ALIGN: 6/5 Positionen möglich	mittel	nicht-deterministisch	Fix steht im Backlog P1
Quality-Threshold-Drift: Ohne sample > 100 BUys per Regime keine seriöse C2-Schwelle	mittel	hoch	C2 erst nach 24-48h sample MIT Phasen-Variation
DCA-Erfolgsquote unbekannt: aktuell 0 erfolgreiche Rescues (alle blockt BEAR), keine Datenbasis für DCA-Sinn-Validierung	niedrig	hoch	BEAR-Block ist Operator-Decision; sample aus BULL nötig
CronCreate session-only Wake-up geht verloren bei Session-End	mittel	mittel	Host-cron als Backup setzen

6. Verbesserungsvorschläge (priorisiert)

ID	Titel	P	Nutzen	Risiko	Aufwand	Files/Module	Abhängig	Empfehlung
DCA-STATE-RECONCILE	DCA-State forensisch rekonstruieren	P1.5	hoch (Trust-Wiederherstellung)	0 (read-only)	1-2 h	logs, dca_log.json, snapshots	Monitor-Ende	nach Monitor
DCA-STATE-SAVE-RACE-FIX	atomare Persistenz nach DCA-Rettung	P1	hoch (Daten-Integrität)	mittel (Bot-Touch)	4-6 h	paper_trade._save_state, mtime	DCA-RECONCILE	später (cutover)
EXTERNAL-CHANNEL-CAP-ALIGN	regime_cap auch in web/telegram-Pfaden	P1	hoch (Cap-Disziplin)	klein	1 h	main.py:970+1109	—	nach Monitor
SNAPSHOT-EMIT-COMPLETENESS	per-scan-emit füllt opened_at/decision_id/etc	P2	mittel (GUI-Vollständigkeit)	klein	2 h	main.py:448	bundle mit Cap-Align	nach Monitor
EVENT-COUNTER-CONSISTENCY-CHECK	6 SL-trigger vs 2 closes klären	P1	hoch (Trust in Metrics)	0	2 h	bot_stdout-Parser + Doku	—	nach Monitor
GUI-VIEW-POSITION-POLISH-2	initial_entry + avg + dca_count + tranches	P2	hoch (Operator-Transparenz)	klein	4 h	PositionSnapshotResource	SNAPSHOT-EMIT-COMP	nach SNAPSHOT-EMIT
WEEKLY-PROFIT-KPI-DASHBOARD	KPI-Widget: WeeklyPnL/Drawdown/Hitrate/etc	P2	hoch (Ziel-Messbarkeit 4%)	klein	6 h	neue Filament-Page + KPIBuilder	trade_logs sample	später
LABEL-1-MARKER-AUDIT	warum HUMA/TAO als legacy_stop_loss?	P2	mittel (Exit-Subtype-Korrektheit)	klein	1 h	execute_buy paths	—	bundle mit P2-GUI
C2-THRESHOLD-CALIBRATION	Threshold 0.55 für C2 ableiten + AB-Test-Plan	P2	hoch (Quality-Logic enforce)	mittel (Bot-Touch)	4 h	quality_score.py + main.py	DECISION-GATE	nach 24h sample
HOST-CRON-WAKE-UP-BACKUP	Host-side cron als Backup für CronCreate	P3	klein (Resilienz)	0	10 min	/etc/cron.d	—	nice-to-have
RISK-GUARD-TELEGRAM-RATE-LIMIT	Operator-Notify nur 1x / 60 min	P3	klein (Spam-Schutz)	klein	3 h	reporter.py + aggregator	—	nice-to-have

7. Konkrete empfohlene nächste Schritte

Was als Nächstes umgesetzt werden soll

1. **Monitor weiterlaufen lassen** bis Wake-up 14:00 UTC am 2026-05-18 (bereits eingestellt).
2. **Nach Wake-up:** DCA-STATE-RECONCILE als 1.5h read-only Forensik → liefert Klarheit über XLM-DCA und entry_price-Semantik.
3. **Direkt danach** Bundle-Cutover (Bot-Touch, 1 Recreate):
4. EXTERNAL-CHANNEL-CAP-ALIGN (Cap-Disziplin)
5. SNAPSHOT-EMIT-COMPLETENESS (GUI-Felder)
6. Optional: LABEL-1-MARKER-AUDIT wenn quick win
7. **Dann** GUI-VIEW-POSITION-POLISH-2 mit initial_entry + tranches + dca_count.
8. **Erst dann** C2 Threshold Calibration auf Basis der frischen + reichhaltigen Daten.

Was aktuell nicht angefasst werden darf

- Bot main.py (PID=300 healthy) → 0 Recreate während Monitor
- Worker neu in Ruhe (post SL-TP-Hotfix) → kein zweiter Cutover bevor nötig
- C1-Quality-Threshold-Tuning → keine Code-Änderung, nur Daten sammeln
- State-Files (live_portfolio.json, dca_log.json) → 0 Edit ohne separates Operator-GO

Was zuerst forensisch geklärt werden muss

1. **DCA-STATE-RECONCILE:** warum XLM-DCA verloren? → Plan existiert
2. **EVENT-COUNTER-INCONSISTENCY:** 6 vs 2 → trivial via log-walking
3. **LABEL-1-MARKER:** warum legacy_stop_loss bei post-LABEL-1-Trades? → execute_buy-Pfad-Audit

8. Backlog-Vorschlag

Bereits in `gui/docs/roadmap/ROADMAP.php` gepinnt:

- COMMAND-BUS-SL-TP-TESTNET P1 ✓ done
- DCA-STATE-RECONCILE P1.5 — plan docs/PLAN_DCA_STATE_RECONCILE.md

- DCA-STATE-SAVE-RACE-FIX P1 — folgt nach Forensik
- EXTERNAL-CHANNEL-CAP-ALIGN P1 — plan docs/PLAN_EXTERNAL_CHANNEL_CAP_ALIGN.md
- SNAPSHOT-EMIT-COMPLETENESS P2 — plan docs/PLAN_SNAPSHOT_EMIT_COMPLETENESS.md
- GUI-VIEW-POSITION-POLISH-1 P2 ✓ done (Sections + Back-Button)
- GUI-POSITIONS-LIST-FIX P2 ✓ done
- RISK-GUARD-TELEGRAM-RATE-LIMIT P3 — plan docs/PLAN_RISK_GUARD_TELEGRAM_RATE_LIMIT_P3.md

Neu vorgeschlagen aus dieser Bewertung:

- EVENT-COUNTER-CONSISTENCY-CHECK **P1** (read-only, forensic + naming-cleanup)
- LABEL-1-MARKER-AUDIT **P2** (warum legacy_stop_loss bei post-LABEL-1-Trades)
- GUI-VIEW-POSITION-POLISH-2 **P2** (initial_entry + tranches + dca_count)
- WEEKLY-PROFIT-KPI-DASHBOARD **P2** (Filament-KPI-Page mit den 13 Operator-Metriken)
- C2-THRESHOLD-CALIBRATION **P2** (Threshold 0.55 + AB-Test)
- HOST-CRON-WAKE-UP-BACKUP **P3** (Resilienz für [session-only] -Wake-ups)

9. Fazit

Ist der Bot auf Kurs?

Code-stabilitätsseitig: ja — 0 Tracebacks, 5 Cutover ohne Drama, Phase A1/B/C1/D wired, Risk-Guard live bei 2 Real-Events korrekt gefeuert.

Datenqualitätsseitig: nein — DCA-State unzuverlässig, GUI zeigt avg ohne Hinweis, Event-Counter inkonsistent, LABEL-1-Marker nicht überall gesetzt. Vor einer Performance-Bewertung muss das geheilt werden.

Performance-seitig: zu wenig Daten — 7 Trades in 16h, 1 Regime, knapp 60 USDT realized loss. Für ein 4 %-Wochen-Statement braucht es 2-3 Wochen, mehrere Marktphasen, und konsistente State-Files.

Ist das 4 %-Wochenziel aktuell seriös bewertbar?

Nein. Daten-Basis fehlt. KPIs teils nicht erhoben. State-Inkonsistenz verzerrt Stichproben.

Was muss vor aggressiver Optimierung zwingend erledigt werden?

1. DCA-STATE-RECONCILE + Folge-Code-Fix (Daten-Integrität)
2. EXTERNAL-CHANNEL-CAP-ALIGN (Risk-Disziplin)
3. SNAPSHOT-EMIT-COMPLETENESS (GUI/Analytics)
4. WEEKLY-PROFIT-KPI-DASHBOARD (Messbarkeit)
5. Sample-Sammeln über min. 30 Trades und 2 Regimes

Erst dann Phase C2 / regime-aware-thresholds / T1-profile-configs. Vorher würde der Bot Risiken multiplizieren auf einer unsicheren Datenbasis.

Stilanmerkung: Diese Bewertung ist optimiert auf Operator-Entscheidung — keine Floskeln, jede Aussage codegestützt. Quellen: 7 CSVs des Reports + Code-Lektüre in `paper_trade.py`, `main.py`, `quality_score.py`, `risk_guard.py`.

STOP.