

POS-CAP-1 Investigation

2026-05-14 UTC · master 7f65097

Cap funktioniert

Telemetry-Bug

Kein Code-Fix urgent

Klares Ergebnis: Die "42 echte offene Positionen" sind ein **Telemetry-Reporting-Bug**, kein Risk-Control-Bruch. Der Bot hält tatsächlich genau **5 Positionen** — exakt der BEAR-Cap. Die `position_snapshots`-Tabelle akkumuliert pro Symbol über Tage, weil `emit_position_snapshot` beim Position-Close NIE mit `status='closed'` aufgerufen wird. `DISTINCT ON (symbol) ORDER BY created_at DESC` liefert daher jedes je-gesehene Symbol als "open".

1. Boundary-Snapshot

Item	Wert
master HEAD	7f65097
Bot/Worker/GUI	healthy, restartCount=0
Bot main.py PID	199
regime_latest	BEAR
effective cap (Bot-Log)	Cap: 5 (regime=5 global_max=10 source=regime) ✓
cmd 13 / mp / history	cancelled / 0 / 0
BINANCE_TESTNET	true
0 Tracebacks post 13:31	bestätigt
runtime_config.json / baseline_holdings.json / managed_state.json	nicht im erwarteten Pfad (keine aktive Persistenz dieser Files)

2. Position-Zählung — was sind die 42?

Klasse	Count	Total Nominal	Bedeutung
FRESH (post-13:31, post-TELE-1b-B Restore)	5	\$2,178	echte aktive Bot-Positionen — Cap=5 eingehalten
PRE-RESTORE (07:20-13:31, Telemetry-Outage)	2	\$660	Übergang während Bot-Down-Phase
STALE (älter als 07:20)	36	\$15,512	nie geschlossene snapshot-rows aus alten Sessions
Total DISTINCT (Audit-Zahl "42")	43	\$18,350	irreführend — Mix aus echt + stale

Die 5 echten aktuellen Positionen (post-Restore):

ASTER/USDT	\$226.40	-0.67
FIL/USDT	\$227.89	+0.44
SOL/USDT	\$228.83	+2.06
SXT/USDT	\$1265.44	-45.94
UNI/USDT	\$229.78	+2.77
TOTAL	\$2,178	-41.34

3. Cap-Logik im Code — korrekt

Datei:Zeile	Logik
trading/main.py:698	<code>max_pos = regime.get('max_positions', 5) if regime else 5</code>
trading/main.py:1081	analog (zweiter Lookup-Punkt)
trading/main.py:1424	<code>_regime_cap = regime['max_positions'] + effective = min(_regime_cap, _global_max)</code> Pattern
trading/execution/position_sizer.py:25	<code>MAX_POSITIONS = 5</code> (Hardcoded-Default)

Bot-Log bestätigt das Live-Verhalten: `Cap: 5 (regime=5 global_max=10 source=regime)` bei jedem Scan-Cycle. **effective_cap = min(regime, global_max) = 5.**

Bot hält tatsächlich genau 5 Positionen → Cap wird nicht umgangen.

4. Root Cause des "42"-Befundes

Der wahre Bug: `position_snapshots.status`-Spalte wird NIE auf `'closed'` gesetzt. Verifikation:

```
SELECT status, count(*) FROM position_snapshots GROUP BY status;
status | count
-----+-----
open   | 15173 ← 100% der Rows
(1 row)
```

Plus: `grep "emit_position_snapshot.*closed"` im Code → **0 Treffer**. Trades werden bei Close mit `status='closed'` in `trade_logs` persistiert (`main.py:287`), aber `position_snapshots` bekommt kein closing-event.

Konsequenz

- position_snapshots akkumuliert jeden Symbol-Eintrag den Bot je gesehen hat
- DISTINCT ON (symbol) ORDER BY created_at DESC liefert immer den letzten snapshot — immer status='open'
- → Auditor sieht 43 "open" Positionen wo tatsächlich 5 aktiv sind
- → Unrealized PnL -129 USDT ist ebenfalls verfälscht (38 stale rows mit veralteten Preisen)

5. Entry-Pfade — alle berücksichtigen den Cap

Pfad	Cap-Check
main.py Hauptloop	JA — liest regime.max_positions bei jedem Scan
execution/paper_trade.py	via main.py-Caller; kein eigener Bypass identifiziert
execution/live_trade.py	analog
execution/dca_manager.py	NICHT explizit cap-checked (DCA = Aufstocken bestehender Position, kein neuer Slot) — by-design OK
execution/multi_strategy_position_sizer.py	nutzt MAX_POSITIONS aus position_sizer.py
Tier-3 / Pyramid	Aufstocken bestehender Positionen, kein neuer Slot

Kein Cap-Bypass identifiziert. DCA/Pyramid/Re-Entry sind by-design "Slot-Erweiterung" auf bestehende Positionen.

6. Historie der 5 echten Positionen

Alle 5 echten Positionen wurden NACH POS-CAP-10-CONSERVATIVE eröffnet (BEAR-Cap=5 schon damals aktiv) und überlebten den Telemetry-Outage. Sie wurden vom Bot nach TELE-1b-B-Restore weitergeschnaphottet — Cap wurde nie verletzt.

Die 36 stale Symbols sind Spuren aus mehreren Monats-Sessions des Bots seit ca. Anfang Mai, in denen Positionen geöffnet+geschlossen wurden, ohne dass emit_position_snapshot(status='closed') jemals aufgerufen wurde.

7. Risk-Control-Antworten

Operator-Frage	Antwort
Warum wurden bei BEAR nicht nur 5 gehalten?	Wurden — Bot hat exakt 5 Positionen aktuell.
Blockiert Cap nur neue Symbole, nicht bestehende?	Korrekt — Cap blockiert nur neue Slots, hält bestehende.
Gibt es keinen Abbau-/Liquidation-Mechanismus?	Nicht beim Cap-Niveau — Positionen werden nur über TP/SL/manual close geschlossen. Bei Regime-Wechsel (BULL→BEAR) wird Cap-Senkung nicht durch Liquidation, sondern durch "keine neuen Entries" umgesetzt.
Soll Cap harte Positionsanzahl oder neue Entry-Grenze sein?	Aktuell: neue Entry-Grenze . Operator-Decision für Future: Liquidation-Mechanismus separat (P1 Backlog).
Müssen Altpositionen in runoff/cleanup?	Nein, Cap wird nicht verletzt. Nur die Telemetrie braucht Cleanup (siehe Fix-Empfehlung).

8. Root-Cause-Hypothese (verifiziert)

Hypothese: Bot hält 5 Positionen (Cap eingehalten). Die "42"-Zahl aus dem Strategy-Loss-Audit war ein Reporting-Artefakt — DISTINCT ON (symbol) FROM position_snapshots liefert 43 Symbols, weil position_snapshots.status nie auf 'closed' gesetzt wird. → 38 davon sind nie-geschlossen-gesnapshottete Symbols aus alten Sessions.

9. Fix-Empfehlung

ID	Fix	Prio
POS-SNAP-1	emit_position_snapshot(status='closed', ...) -Call beim Position-Close in paper_trade.py + live_trade.py einfügen. Single-line-Edit pro File.	P1
POS-SNAP-2	Historische 38 stale-rows als status='stale_closed' markieren (einmaliges UPDATE-Statement nach POS-SNAP-1 deployment). Wartungs-Migration.	P2
POS-SNAP-3	Audit-Query-Pattern dokumentieren: NICHT DISTINCT ON (symbol), sondern WHERE status='open' -Filter. ggf. SQL-View v_current_open_positions definieren.	P2
RUNOFF-1	Operator-Decision: bei Regime-Wechsel BULL→BEAR aktive Positions-Abbau ("runoff mode")? — separate Strategy-Phase, NICHT POS-CAP-1.	P3 Operator-Decision

10. GO/NO-GO Code-Fix

NO-GO sofortiger Code-Fix. Cap funktioniert korrekt — kein urgentes Risk-Control-Problem. POS-SNAP-1 (status='closed' beim Close) ist eine wichtige Telemetrie-Korrektur, kann aber als ruhige P1-Phase laufen. Empfohlene Sequenz:

1. **STOP** — diese Investigation closure
2. Strategy Interim Report mit korrigierter "5 echte Positionen, nicht 42"-Zahl
3. POS-SNAP-1 Plan-Review (vor Implementation)
4. POS-SNAP-1 Implementation in eigener Phase
5. POS-SNAP-2 Wartungs-Migration für stale Rows

11. Boundaries — alle eingehalten

0x Code · 0x DB-Write · 0x Restart · 0x docker cp · 0x Mainnet · 0x Orders · 0x Positionsänderung · 0x runtime_config-touch · 0x Secret-Output · 0x env-Dumps · 0x `compose config` .

2026-05-14 UTC · POS-CAP-1 Investigation · master `7f65097`