

POS-CAP-1 Investigation — v2

2026-05-14 UTC · master 7f65097 ·

Cap eingehalten

SNAPSHOT-LIFECYCLE-Bug

Klares Ergebnis: Bot hält tatsächlich **5 echte Positionen = BEAR-Cap exakt**. Die im Strategy-Loss-Audit gemeldeten "42 offenen Positionen" sind ein **SNAPSHOT-LIFECYCLE-Bug**: `position_snapshots.status` wird beim Position-Close nie auf `'closed'` gesetzt, sodass DISTINCT-ON-Queries 38 historische Symbol-Rows als "offen" zurückgeben. **Risk-Control ist intakt.**

1. FRESH vs STALE — Klassifikation

Cutoff-Zeitpunkt: **2026-05-14 13:31 UTC** = TELE-1b-B Bot-Container-Recreate mit `psycpg2-fix` + `OpenClaw-Removal`. Ab diesem Zeitpunkt schreibt der Bot wieder kontinuierlich `position_snapshots`.

Klasse	Definition	Count	Total Nominal	Interpretation
FRESH	<code>created_at > '2026-05-14 13:31'</code>	5	\$2,178	echte aktive Bot-Positionen, vom Bot in jeder Scan-Iteration upgedatet
PRE-RESTORE	<code>'2026-05-14 07:20' < created_at ≤ '2026-05-14 13:31'</code>	2	\$660	Übergang zwischen SEC-1c-3b-Recreate und TELE-1b-B-Restore (Telemetry-Outage-Phase)
STALE	<code>created_at ≤ '2026-05-14 07:20'</code>	36	\$15,512	nie geschlossene Snapshot-Rows aus alten Sessions (mehrere Tage alt)
Total DISTINCT	einfaches DISTINCT ON (symbol)	43	\$18,350	irreführende Audit-Zahl — Mix aus echt + stale

Die 5 frischen (echten) Positionen

symbol	pos_value	upnl	created_at (latest)
ASTER/USDT	\$226.40	-\$0.67	2026-05-14 13:49:29
FIL/USDT	\$227.89	+\$0.44	2026-05-14 13:49:29
SOL/USDT	\$228.83	+\$2.06	2026-05-14 13:49:29
SXT/USDT	\$1265.44	-\$45.94	2026-05-14 13:49:29
UNI/USDT	\$229.78	+\$2.77	2026-05-14 13:49:29
TOTAL	\$2,178	-\$41.34	

Nachweis BEAR-Cap eingehalten: 5 frische Positionen = exakt der BEAR-Cap=5 aus `P0S-CAP-10-CONSERVATIVE`. Bot-Live-Log bestätigt: `Cap: 5 (regime=5 global_max=10 source=regime)` bei jedem Scan-Cycle.

2. SNAPSHOT-LIFECYCLE-Bug — Nachweis

Beweis A: Alle `position_snapshots` haben `status='open'`

```
SELECT status, count(*) FROM position_snapshots GROUP BY status;
```

```
status | count
-----+-----
open   | 15173 ← 100% der Rows, kein einziger 'closed'
(1 row)
```

Beweis B: `emit_position_snapshot` mit `status='closed'` = 0 Treffer im Code

```
$ grep -rnE "emit_position_snapshot.*closed" /projekte/Steve-TradingBot/trading/
(keine Treffer)
```

Vergleich: `emit_trade(status='closed', ...)` ist im Code vorhanden (`trading/main.py:287`) — TradeLogs werden bei Close korrekt persistiert. Position-Snapshots hingegen NICHT.

Beweis C: DISTINCT-ON-Falle erklärt

Audit-Query war:

```
SELECT DISTINCT ON (symbol) symbol, position_value, status, created_at
FROM position_snapshots
ORDER BY symbol, created_at DESC;
```

Erwartete Semantik: „für jedes Symbol nimm den neuesten Snapshot“. Wenn dieser Snapshot dann `status='closed'` hätte, müsste man via `WHERE status='open'` filtern. Da aber **kein Snapshot je auf 'closed' gesetzt wird**, ist der letzte Snapshot pro Symbol immer `'open'` — egal ob die Position längst geschlossen wurde. Die Query liefert damit:

- Jedes Symbol, das jemals geöffnet wurde, mit dessen letzten geknippten Werten
- Plus aktuelle Bot-Positionen (mit frischen Updates)
- = Vermischung „echt aktiv“ und „historisch geöffnet, längst geschlossen“

Die korrekte Frage „welche Positionen hat der Bot ECHT offen?“ kann **diese Query gar nicht beantworten**. Sie ist eine reine Lifetime-Distinct-Liste.

3. Cap-Logik-Verifikation (korrekt im Code)

Pfad	Logik
trading/main.py:698, 1081, 1424	max_pos = regime.get('max_positions', 5) if regime else 5
trading/execution/position_sizer.py:25	MAX_POSITIONS = 5 hardcoded fallback
Bot-Live-Log bei jedem Scan-Cycle	[Regime] Markt: BEAR Min Score: 7.5 Cap: 5 (regime=5 global_max=10 source=regime)

Bot hält real **5 Positionen**. **Cap wird nicht umgangen** — der Eindruck "42 offene Positionen" entstand allein durch den SNAPSHOT-LIFECYCLE-Bug.

4. Konsequenz für Strategy-Loss-Audit (Korrektur)

Vorheriger Befund (Strategy-Audit)	Korrektur post-POS-CAP-1
"42 echte offene Positionen trotz BEAR-Cap=5"	5 echte Positionen, Cap eingehalten
"Nominal Exposure \$19,145.97 USDT"	\$2,178 USDT (echt) — 88% Reduktion
"Unrealized PnL -\$129.66 USDT"	-\$41.34 USDT (echt) — 68% Reduktion
"Exposure = 2.08x Cash → Massive Cap-Verletzung"	Exposure \$2,178 vs \$9,206 Testnet-Balance = 0.24x Cash , konservativ
"POS-CAP-1 Untersuchung dringend"	Cap-Logik intakt. Echte P1 = SNAPSHOT-LIFECYCLE-1 (Telemetrie)

Strategie-Audit-Empfehlung (Update): „Offene Positionen“ sollen NICHT mehr aus historischer `DISTINCT ON (symbol)` -Query abgeleitet werden. Verlässliche Quellen für aktuelle aktive Positionen:

- FRESH-Snapshot-Filter:** `WHERE created_at > <letzter Bot-Restart-Zeitpunkt>` (im aktuellen Stand: > 2026-05-14 13:31)
- Live Wallet / Paper State:** Bot-internes `live_portfolio.json` oder analoge Datei (aktuell nicht im erwarteten Pfad — siehe IMAGE-2-Backlog für Code-Baking)
- Nach SNAPSHOT-LIFECYCLE-1-Fix:** `WHERE status='open' + DISTINCT ON (symbol)` wird dann verlässlich

5. NEUES Backlog-Item — SNAPSHOT-LIFECYCLE-1

ID	Beschreibung	Prio
SNAPSHOT-LIFECYCLE-1	position_snapshots beim Position-Close korrekt auf status='closed' setzen ODER ein echtes <code>position_state</code> -Modell einführen (z.B. dedizierte Tabelle <code>open_positions</code> oder Materialized View). Konkrete Pfade: (a) <code>trading/execution/paper_trade.py</code> beim <code>close_position(): emit_position_snapshot(status='closed', ...)</code> (b) <code>trading/execution/live_trade.py</code> analog (c) ggf. <code>db_emitter.py</code> erweitern um ein <code>emit_position_close(position_id)</code> -Convenience-Wrapper Alternativ-Architektur: <code>position_state</code> -Tabelle mit echter Lifecycle-Tracking (open/closing/closed) und <code>UNIQUE(position_id)</code> , Snapshots bleiben rein historisch.	P1
SNAPSHOT-LIFECYCLE-2	Einmalige Wartungs-Migration: 36 historische stale-Rows (älter 13:31) markieren — z.B. <code>UPDATE position_snapshots SET status='stale_historic' WHERE created_at < '2026-05-14 13:31' AND symbol NOT IN (<5 fresh>)</code> . Nur nach SNAPSHOT-LIFECYCLE-1 ausführen.	P2
SNAPSHOT-LIFECYCLE-3	SQL-View bzw. dokumentiertes Query-Pattern: <code>v_current_open_positions = letzter Snapshot pro Symbol WHERE status='open' AND created_at > (NOW() - threshold)</code> . Für GUI- und Audit-Konsumenten.	P2
RUNOFF-1	Operator-Decision: aktiver Positions-Abbau bei BULL→BEAR-Regime-Wechsel ("runoff mode")? Separate Strategie-Phase, nicht Telemetrie.	P3

6. Boundary-Snapshot

Item	Wert
master HEAD	7f65097 (unverändert)
Bot main.py PID	199 (in-container)
Bot Container Host-PID	3082815 — running healthy, restartCount=0
Worker Container Host-PID	2658058 — unverändert
GUI Container Host-PID	2656633 — unverändert
regime_latest	BEAR
effective_cap (Live-Log)	5 (regime=5 global_max=10 source=regime)
BINANCE_TESTNET	true
cmd 13 / mp / history	cancelled / 0 / 0
0 Tracebacks post 13:31	bestätigt
Worker heartbeat	frisch (age < 30s)

7. GO/NO-GO

NO-GO sofortiger Code-Fix. Risk-Control ist intakt — der Bot hält den Cap ein. SNAPSHOT-LIFECYCLE-1 ist eine wichtige Telemetrie-Korrektur, aber kein Mainnet-Blocker. Empfohlene Sequenz:

1. STOP (diese Investigation closure)
2. **Strategy Interim Report** mit korrigierten Zahlen (\$2,178 / -\$41, nicht \$19,145 / -\$129)
3. SNAPSHOT-LIFECYCLE-1 Plan-Review (wenn Operator soweit)
4. SNAPSHOT-LIFECYCLE-1 Implementation
5. SNAPSHOT-LIFECYCLE-2 einmalige Wartungs-Migration
6. SNAPSHOT-LIFECYCLE-3 SQL-View / Query-Standard

8. Boundaries — alle eingehalten

0x Code · 0x DB-Write · 0x Restart · 0x docker cp · 0x Mainnet · 0x Orders · 0x Positionsänderung · 0x runtime_config-touch · 0x managed_state-touch · 0x baseline-touch · 0x Secret-Output · 0x env-Dumps · 0x docker compose config mit Secrets · 0x /proc/*/environ bulk.

2026-05-14 UTC v2 · POS-CAP-1 Investigation · master `7f65097` · Cutoff TELE-1b-B-Restore: 2026-05-14 13:31 UTC · NEU: SNAPSHOT-LIFECYCLE-1 P1 Backlog