

# Pyramid Observability Review — DOGE/USDT 2026-05-13 18:06 UTC

Projekt: Steve-TradingBot · Phase: PYRAMID-OBSERVABILITY-REVIEW · Author: claude-opus-4-7[1m]  
Generated: 2026-05-13 18:16 UTC · master HEAD: 2fbc7b5  
Bot PID: 363 (paper-mode flag im Code, aber TESTNET-LIVE-API via ccxt) · Worker PID: 1 · BINANCE\_TESTNET=true  
Status: **NO CODE / NO RUNTIME-CHANGE** Pure Analysis, Backlog, Empfehlungen. STOP nach Output.

## 0 — Executive Summary

Die erste produktive Aufstockungs-Telegram-Nachricht ist **strukturell sauber**: classifier + formatter sind als getrennte Pure-Module realisiert, decision\_log\_id + order\_id sind im Output enthalten, TESTNET-Label korrekt, dca>pyramid-Prioritaet stimmt mit main.py-Control-Flow uberein, MAX\_PYRAMID\_COUNT=1 ist als harter Cap aktiv, Budget-Hard-Cap auf total\_size verhindert >100%-Allokation.

Allerdings: **der Pyramid-Trigger ist regime-blind** — das DOGE-Event wurde im **BEAR-Regime** ausgelost (live verifiziert: decision\_logs.regime='BEAR' fuer alle 2214 BUY-Entscheidungen der letzten Stunden, einschl. decision\_id e8a607af ). Ebenso: **keine Meme-Coin-Restriktion**, keine Volatility-Skalierung, kein Cooldown nach Pyramid, kein Exposure-/Regime-Hinweis in der Telegram-Nachricht. Diese Punkte sind **Mainnet-blocking** und gehoeren ins Backlog vor MH-7 (Bot-Wiring fuer Mainnet).

Aktuelle Risiko-Klassifikation auf **TESTNET**: **akzeptabel** — max. Schaden = Testnet-Verluste, Bot bleibt paper-trade-orchestriert, Mainnet-Block 5-layer aktiv. Auf **MAINNET**: **NO-GO** bis Mindestens P0-Items abgeschlossen sind.

## 1 — Faktenlage (live verifiziert)

### Telegram-Nachricht (operator-zitiert)

```
🔗 Aufstockung (Pyramid) ausgeführt – TESTNET LIVE
Symbol: DOGE/USDT
Subkind: tranche_3
Menge: 1260.0000 (Bestand 2049.0000 → 3309.0000)
Preis: 0.1136 USDT
Orderwert: 143.16 USDT
Avg-Preis: 0.1125 USDT
Score: 6.2/10

Grund:
Pyramid: Kurs +1.6% über Entry
Risk-Checks bestanden (N7.2 + RECON-2.2 + Cash-Reserve + Circuit-Breaker).

decision_log_id: e8a607af
order_id: 385578
👤 Steve Trading Bot
```

### DB-Record (decision\_logs)

decision_id	e8a607af
symbol	DOGE/USDT
action	buy
score	6.0000
combined_score	6.1950
regime	<b>BEAR</b>
environment	testnet
mode	paper
source	run_scan_cycle
decided_at	2026-05-13 18:06:23 UTC

### DCA-Log (state-of-truth fuer Pyramid-Count)

<b>DOGE total_size (Budget)</b>	572.70 USDT
<b>DOGE entry_price (T1)</b>	0.11176
<b>DOGE avg_price post-pyramid</b>	0.112449 (= 0.1125)
<b>DOGE total_invested</b>	372.26 USDT (= 64.9 % von 572.70)
<b>dca_count / pyramid_count</b>	0 / 1 (MAX=1)
<b>tranches</b>	T1 13:36:59 UTC: 229.08 USDT @ 0.11176 / <b>T3-Pyramid 18:06:23 UTC: 143.18 USDT @ 0.11357 (+1.6%)</b>

### Logik-Kette (Code-Pfad, read-only)

1. `main.py scan-cycle` → signal fuer DOGE
2. Position existiert bereits → `dca_mgr.check_dca` liefert None (Preis > Entry, kein DCA-Trigger)
3. `dca_mgr.check_pyramid` prueft `pyramid_count < 1 & Preis > entry*1.015` → True bei +1.6%
4. `Size = get_tranche_size(572.70, 3) = 25% = 143.18 USDT` ≤ verbleibendes Budget
5. `execute_buy(allow_add=True)` → ccxt-Testnet-Order, `order_id 385578`
6. `dca_mgr.confirm_pyramid` erhoeht `pyramid_count 0→1`, Tranche-Eintrag persistiert
7. `buy_classifier.classify_buy` → `kind=rebuy_pyramid, subkind=tranche_3`
8. `rebuy_formatter.format_rebuy_message` → HTML-Text fuer Telegram
9. Reporter sendt `parse_mode='HTML'`

## 2 — Phase 1: Live-Review der Telegram-Nachricht

#	Frage	Befund
1	Sofort verstaendlich?	<b>JA</b> — klare Header, Spaltenausrichtung, sinnvolle Labels
2	Kritische Infos fehlen?	<b>JA</b> — Regime, Cash-Reserve, Exposure-%, P&L unrealized fehlen
3	Zahlen konsistent?	<b>JA</b> — $1260 * 0.1136 = 143.14$ (msg sagt 143.16, Floating-Roundoff toleriert); Bestand $2049+1260=3309$ stimmt; avg-Preis 0.1125 matcht DCA-Log 0.112449
4	„Kurs +1.6% ueber Entry“ erklarend genug?	<b>TEILWEISE</b> — Operator versteht +1.6%, aber unklar WO der Trigger sitzt (+1.5% pinned). Ergaenzen: „Trigger ab +1.5%“
5	<code>tranche_3</code> verstaendlich?	<b>NEIN</b> — intern; Operator sollte sehen: „Pyramid-Tranche 3/3 (25%)“ oder „Aufstock-Tranche (25% des Budgets)“
6	Avg-Preis korrekt?	<b>JA</b> — mathematisch korrekt: $229.08 \text{ USDT} / 0.11176 = 2049.7 \text{ Qty}$ ; $+1260 @ 0.1136 = \text{ges. } 3309.7 \text{ Qty}$ ; $\text{weighted-avg} = 372.26 / 3309.7 = 0.11248$
7	Positionsgroessenaenderung intuitiv?	<b>JA</b> — absolutes Format (2049 → 3309), +61.5% in Qty
8	Gesamt-Exposure fehlt?	<b>JA</b> — weder gesamt-positionen noch USDT-Exposure noch %-vom-Capital sichtbar
9	Regime-State fehlt?	<b>JA</b> — <b>BEAR</b> ist in DB, aber NICHT in der Nachricht. Kritisch — Operator sieht nicht, dass aktuell BEAR-Regime herrscht
10	Risk-Level fehlt?	<b>JA</b> — statisches „Risk-Checks bestanden“ ohne konkrete Werte (Cash-Reserve-%, Circuit-Breaker-Status, Open-Pos-Count)
11	Verbleibende Cash-Reserve fehlt?	<b>JA</b>
12	<code>decision_log_id</code> sichtbar genug?	<b>JA</b> — eigene Zeile, monospace
13	<code>order_id</code> ausreichend auditierbar?	<b>JA</b> — eigene Zeile, monospace; verlinkbar mit DB / Exchange
14	TESTNET prominent genug?	<b>JA</b> — im Header neben dem Action-Label
15	Zu lang bei vielen Signalen?	<b>TENDENZ JA</b> — aktuell 15 Zeilen; mit Regime/Exposure/Cash dehnt sich auf 18-20 Zeilen. Telegram-Mobile-View wird scrollig

## 3 — Phase 2: Strategie-Review

#	Frage	Befund
1	+1.5% Trigger sinnvoll?	<b>SELTSAM</b> — PYRAMID_TRIGGER=0.015 ist sehr niedrig (Noise-Bereich bei volatilen Coins). Industrie-Standard fuer Pyramiding: 3-5%. <b>Backlog</b>
2	tranche_3 zu aggressiv?	<b>NEIN per se</b> — 25% des Budgets ist konservativ; das Risiko liegt in der niedrigen Trigger-Schwelle
3	DOGE fuer Pyramid geeignet?	<b>PROBLEMATISCH</b> — DOGE ist Meme-Coin mit hoher Volatilitaet; +1.5%-Trigger ist im Intra-Hour-Noise; Pyramid auf Meme-Coin in BEAR-Regime ist kein gutes Setup
4	Score 6.2 ausreichend?	<b>GRENZWERTIG</b> — threshold ist 6.0 (combined_score 6.195 nur knapp drueber); fuer Pyramid sollte ein HOEHERER Confirmation-Score gelten als fuer Erstkauf (z. B. 7.0)
5	Positionssprung 2049→3309 zu gross?	<b>NEIN absolut</b> — in USDT-Termini: 229 → 372 = 64.9% Budget-Auslastung (von 572 max). Tranche-3 ist designwise die kleinste
6	Passt zur BEAR-Regime-Cap?	<b>NEIN</b> — <b>kritischster Findings</b> : check_pyramid in dca_manager.py:251 liest <b>weder Regime noch Cap</b> ; die regime_cap=5 fuer BEAR (POS-CAP-10-CONSERVATIVE) gilt nur fuer Open-Position-Count, nicht fuer Pyramid-Adds
7	Overexposure-Risiko?	<b>JA</b> — Hard-Cap pro Position ( total_invested + size <= total_size ) schuetzt EINZELPOSITION, aber GESAMT-Exposure (Summe aller Positionen) wird im Pyramid-Pfad nicht geprueft
8	Pyramid regime-abhaengig?	<b>SOLLTE</b> — in BEAR/CRASH-Regimes deaktivieren oder Trigger erhoehen (z. B. +3%)
9	Meme-Coin restriktiver?	<b>SOLLTE</b> — Meme-Coin-Allowlist + hoeherer Pyramid-Trigger (5%+) ODER komplettes Pyramid-Verbot
10	tranche_3 zusaetzliche Confirmation?	<b>SINNVOLL</b> — RSI > 50, EMA50>EMA200, Volume-Confirmation als Add-On (analog DCA-Support-Check in check_dca , aber spiegelverkehrt fuer Up-Move)
11	Max-Pyramid-Level?	<b>EXISTIERT</b> — MAX_PYRAMID_COUNT=1 ; korrekt minimal
12	Vol-skaliert?	<b>NEIN</b> — DCA-Pfad nutzt ATR-Multiplikator ( DCA_ATR_MULT_1/2 ); Pyramid-Pfad ist starr +1.5%. Ungleichgewicht im Design

## 4 — Phase 3: Risk & Failure-Mode Review

#	Frage	Befund
1	Rapid-Rebuy-Loops?	<b>GESCHUETZT</b> — MAX_PYRAMID_COUNT=1 verhindert Loops; pyramid_count persistiert in dca_log.json
2	Starke Pumps?	<b>TEILWEISE</b> — +1.5% Trigger feuert frueh; bei +5% Pump wuerde Pyramid bereits am +1.5% gefeuert haben und kann nicht erneut. Slow-Pump-Friendly aber Fast-Pump-FOMO-Risiko bleibt
3	Slippage?	<b>UNGEPRUEFT</b> — execute_buy -Pfad geht via ccxt market-order auf Testnet; Slippage-Simulation nicht in Code; bei Mainnet kritisch
4	Illiquide Assets?	<b>UNGEPRUEFT</b> — kein Liquidity-Check im Pyramid-Pfad; min_volume -Filter sitzt auf Signal-Generation-Seite, nicht Pyramid
5	Gleichzeitige Pyramid-Events?	<b>SINGLE-THREADED</b> — run_scan_cycle verarbeitet Symbol fuer Symbol sequenziell; keine Race
6	Telegram-Delay?	<b>NICHT KRITISCH</b> — Notify ist async/best-effort; bot-execution wartet nicht; im Worst-Case verliert man eine Notification, kein Trade
7	order_id fehlt?	<b>GRACEFUL</b> — formatter rendert n/a (siehe rebuy_formatter.py:164 )
8	decision_log_id fehlt?	<b>GRACEFUL</b> — ebenfalls n/a
9	Avg-Preis stale?	<b>MOEGLICH</b> — _resolve_avg_price liest portfolio['positions'][symbol]['entry_price']; falls PaperTrader und LiveTrader divergieren (position_snapshots zeigt 3184 @ 0.10907 vs. DCA-Log 3309 @ 0.112449), kann msg/state-Drift entstehen. <b>Backlog: dual-source-of-truth reconcile</b>
10	Double-Notification?	<b>UNWAHRSCHEINLICH</b> — formatter wird nach confirm_pyramid aufgerufen; pyramid_count==1 nach erstem Mal → check_pyramid liefert None bei Wiederholung
11	Race DCA vs Pyramid?	<b>GELOEST</b> — main.py:642+ ist if dca: ... else: pyr = check_pyramid(); sequential, dca hat Vorrang (operator-pinned in classifier-Docstring)
12	Cooldowns?	<b>NICHT VORHANDEN</b> — Pyramid kann theoretisch direkt nach DCA passieren (oder umgekehrt), wenn Preis-Bewegungen entsprechend sind. Cooldown-Pin (z. B. 30 min minimum zwischen Tranchen) fehlt

## 5 — Phase 4: Audit & Telemetrie

---

#	Frage	Befund
1	Eindeutige Rekonstruktion?	<b>JA</b> — decision_logs + trade_logs + dca_log.json zusammen geben den vollen Trail
2	decision_log_id + order_id ausreichend?	<b>JA</b> — in Telegram + DB; verlinkbar
3	proposal_id ergaenzen?	<b>N/A</b> — RECON-MH/managed_proposals nicht aktiv (0 rows); irrelevant solange MH-Capability nicht greift
4	current_regime ergaenzen?	<b>SOLLTE</b> — in DB vorhanden, fehlt in Telegram-Nachricht
5	current_cap ergaenzen?	<b>SOLLTE</b> — effective_cap=5 in BEAR-Regime; Operator weiss nicht, ob Position-Cap erreicht ist
6	exposure_pct ergaenzen?	<b>SOLLTE</b> — total_invested/total_size pro Position UND sum / available_capital global
7	realized/unrealized P&L ergaenzen?	<b>KOMFORT</b> — unrealized-P&L pro Position waere fuer Operator-Sicht nuetzlich (z. B. „DOGE +1.6% / +5.97 USDT“)
8	Structured Telemetry?	<b>TEILWEISE</b> — tranches[] -Array in dca_log.json ist semi-strukturiert; kein expliziter Event-Stream
9	Eigenes pyramid_event_log?	<b>SINNVOLL</b> — aktuell mit DCA-Tranches vermischt; eigener Stream fuer Pyramid-spezifische Auswertung (z. B. Win-Rate-Vergleich first_buy vs. pyramid_add)
10	Hash/sha8 in Telegram-Meldung?	<b>KOSMETISCH</b> — nicht zwingend, aber als kompakter Idempotency-Marker (z. B. sha8 von (decision_id+order_id) ) nuetzlich fuer Duplicate-Detection durch Operator-Auge

---

## 6 — Phase 5: Priorisiertes Backlog

---

ID	Prio	Kategorie	Item	Mainnet-Blocking?
BL-P01	P0	Regime-awareness	check_pyramid muss aktuelles Regime einlesen; in BEAR/CRASH per default deaktivieren ODER PYRAMID_TRIGGER auf +3% erhoehen	JA
BL-P02	P0	Position-sizing	Pyramid-Confirmation-Score > Erstkauf-Threshold (z. B. combined_score >= 7.0)	JA
BL-P03	P0	Exposure-limits	Globale Exposure-Cap im Pyramid-Pfad pruefen (sum of total_invested <= max_capital_exposure_pct)	JA
BL-P04	P0	Strategy	Meme-Coin-Allowlist/Denylist fuer Pyramid; DOGE/SHIB/PEPE/BOME/PUMP/FUN/NOT/etc. blocken ODER Trigger 5%+	JA
BL-P05	P0	Cooldowns	Mindest-Abstand zwischen Tranchen (z. B. 30 min); verhindert Same-Cycle-Pyramid-nach-DCA	JA
BL-P06	P0	Risk	Slippage-Schutz: max_slippage_pct vor execute_buy validieren (orderbook-spread-check)	JA
BL-P07	P1	Notifications	Telegram-Msg um Regime: BEAR, Cap: 5/5, Exposure: 64% (372/572 USDT), Cash-Reserve: ___ USDT erweitern	NEIN (Komfort, aber wichtig)
BL-P08	P1	UX	Subkind-Label: tranche_3 → Pyramid-Tranche 3/3 (25% Budget) oder Aufstockung 25%	NEIN
BL-P09	P1	UX	Reason-Zeile erweitern: Pyramid: Kurs +1.6% über Entry (Trigger: +1.5%)	NEIN
BL-P10	P1	Strategy	ATR-/Vol-Skalierung fuer Pyramid-Trigger analog DCA-ATR-Multiplikator	NEIN (Komfort, aber Konsistenz mit DCA-Pfad)
BL-P11	P1	DCA/Pyramid interaction	Pyramid-Confirmation-Check analog zu DCA-Support-Check (RSI > 50, EMA50 > EMA200, MACD-positive)	NEIN
BL-P12	P1	Telemetry	Eigener structured-event-stream fuer Pyramid (separat von tranches[] in dca_log.json) — ermöglicht Win-Rate-Auswertung	NEIN
BL-P13	P1	Audit-trail	position_snapshots vs. dca_log.json reconcile-Tool (Drift-Detection)	JA (Mainnet)
BL-P14	P2	UX	Realtime unrealized-P&L pro Position in Telegram-Msg ( P&L: +1.6% / +5.97 USDT )	NEIN
BL-P15	P2	Logging	Pyramid-Decision-Log mit Reject-Reasons (warum NICHT pyramid'd) — ermöglicht missed-opportunity-Analyse	NEIN
BL-P16	P2	UX	Mobile-friendly Compact-Mode: Multi-Signal-Burst zusammenfassen statt 15 Einzel-Messages	NEIN
BL-P17	P2	Operator-Controls	Per-Symbol Pyramid-Disable via runtime_config (Operator kann DOGE/Meme einzeln deaktivieren ohne Code-Touch)	NEIN
BL-P18	P3	Strategy	Pyramid-Trigger trailing (z. B. ueberhoehe entry+1.5% UND keine RSI>70-Falle)	NEIN
BL-P19	P3	Mainnet-hardening	Pre-Mainnet-Drill: Pyramid auf TESTNET unter mind. 3 Regime-Wechseln testen (BEAR/RECOVERY/BULL); Backtest-Vergleich	JA
BL-P20	P3	Telemetry	Pyramid-Idempotency-Hash (sha8 von decision_id+order_id) im Telegram-Footer fuer manuelles Operator-Duplicate-Check	NEIN

## 7 — Phase 6: Empfehlungen & GO/NO-GO

### 7.1 Was ist bereits sehr gut

- **Saubere Modul-Trennung:** buy\_classifier.py + rebuy\_formatter.py sind Pure-Module ohne I/O, ohne Bot-Runtime-Imports → trivial testbar
- **Operator-pinned dca>pyramid Priority** (classifier:52-54) ist im Docstring dokumentiert und matcht main.py-Control-Flow
- **Graceful Fallbacks:** order\_id=n/a + decision\_log\_id=n/a + honest-fallback-reason wenn signal incomplete
- **Hard-Cap pro Position:** total\_invested + size <= total\_size verhindert >100%-Allokation pro Symbol
- **MAX\_PYRAMID\_COUNT=1:** konservativ, blockiert Rebuy-Loops
- **TESTNET-Label** im Header prominent
- **decision\_log\_id + order\_id** beide in Telegram-Output
- **DB-Audit-Trail vollstaendig:** decision\_logs + trade\_logs + dca\_log.json zusammen rekonstruierbar
- **Reasons-String** ist mensch-lesbar („Pyramid: Kurs +1.6% ueber Entry“)

### 7.2 Was ist kritisch

- **Pyramid ist regime-blind** — im BEAR-Regime gefeuert (live verifiziert)
- **Pyramid auf Meme-Coin (DOGE)** ohne Allowlist-Check
- **Score-Threshold gleich Erstkauf** (6.0); Pyramid sollte hoehere Confirmation brauchen
- **Kein Slippage-Check** vor execute\_buy
- **Kein Cooldown** zwischen Tranchen
- **Kein globaler Exposure-Cap** im Pyramid-Pfad
- **Regime / Cap / Exposure / Cash** fehlen in Telegram-Nachricht
- **position\_snapshots vs dca\_log.json Drift** (3184 @ 0.10907 vs 3309 @ 0.112449) — Recon-Tool-Bedarf

### 7.3 Was sollte NICHT geaendert werden

- dca>pyramid Prioritaet (operator-pinned, korrekt)
- MAX\_PYRAMID\_COUNT=1 (konservativ richtig)
- 40/35/25 Tranchen-Split (T1/T2/T3)
- Pure-Modul-Architektur fuer classifier + formatter (Leaf-Only, no-I/O)
- Honest-Fallback-Reason im formatter (statt Liesüber Default)
- Hard-Cap auf total\_size pro Position
- Mainnet-5-Layer-Block bleibt unangetastet

### 7.4 Mainnet-Blocking Items (vor MH-7 Bot-Wiring)

1. BL-P01 Regime-aware Pyramid-Gate
2. BL-P02 hoehere Pyramid-Confirmation-Score
3. BL-P03 globaler Exposure-Cap
4. BL-P04 Meme-Coin-Restriktion
5. BL-P05 Tranchen-Cooldown
6. BL-P06 Slippage-Schutz
7. BL-P13 position\_snapshots vs dca\_log Reconcile
8. BL-P19 Pre-Mainnet-Drill mit Regime-Wechseln

### 7.5 Nur kosmetisch

- BL-P08 tranche\_3 → „Pyramid-Tranche 3/3 (25%)“
- BL-P09 Trigger-Zahl im Reason
- BL-P14 unrealized-P&L in Telegram
- BL-P16 Compact-Mode
- BL-P20 sha8-Hash im Footer

### 7.6 Welche Punkte vor MH-6 (Worker-Handler)?

Keine direkt — MH-6 ist GUI-side Worker fuer managed\_proposals (RECON-MH). Pyramid-Logik ist Bot-side und beruehrt MH-6 nicht. **Pyramid-Punkte koennen parallel zu MH-6 angegangen werden**, sofern Bot-Code-Touch und Worker-Code-Touch atomar getrennt bleiben.

### 7.7 Welche Punkte gehoeren in SEC-2?

- BL-P17 (Operator-Controls per runtime\_config) ist eher POS-CAP-2/3 Operator-Control-Layer, nicht SEC-2 (SEC-2 = Docker/Container-Hardening)
- Pyramid-Items haben keine direkte SEC-2-Kreuzung

### 7.8 Welche Punkte gehoeren in POS-CAP-2/3?

- BL-P03 globaler Exposure-Cap — passt zur POS-CAP-Familie (POS-CAP-10 hat Open-Pos-Count-Cap, POS-CAP-2/3 koennten USDT-Exposure-Cap definieren)
- BL-P17 Operator-Controls per runtime\_config — passt zur POS-CAP-Familie
- BL-P01 Regime-aware Pyramid — eng verwandt mit POS-CAP-10-CONSERVATIVE (regime → cap)

### 7.9 Empfohlene Reihenfolge

1. **P0-Block-A (Risk-Hardening)**: BL-P01 + BL-P04 + BL-P05 + BL-P06 (Regime-Gate / Meme-Restriktion / Cooldown / Slippage)
2. **P0-Block-B (Exposure-Hardening)**: BL-P02 + BL-P03 + BL-P13 (Confirmation-Score / Globale-Exposure-Cap / Reconcile)

3. **P1-Block-C (UX + Observability):** BL-P07 + BL-P08 + BL-P09 + BL-P10 + BL-P11 + BL-P12 (Telegram-Felder / Subkind-Label / ATR-Skalierung / Confirmation-Signals / Event-Stream)
4. **P2-Block-D (Komfort):** BL-P14 + BL-P15 + BL-P16 + BL-P17
5. **P3-Block-E (langfristig):** BL-P18 + BL-P19 + BL-P20

### 7.10 GO/NO-GO Einschätzung fuer unveraenderten Weiterbetrieb

Umgebung	Empfehlung
<b>TESTNET (aktuell)</b>	<b>GO — weiter beobachten</b> Maximaler Schaden = Testnet-Verluste; Bot bleibt paper-orchestriert; Mainnet-5-Layer-Block aktiv; Data-Collection fuer Backlog-Analyse weiterhin wertvoll.
<b>MAINNET (zukuenftig)</b>	<b>NO-GO</b> bis Mindestens <b>P0-Block-A + P0-Block-B</b> abgeschlossen sind. BL-P19 (Pre-Mainnet-Drill mit Regime-Wechseln) ist Pflicht-Gate vor MH-7 Bot-Wiring.

## 8 — Boundaries (Review-End-State)

master HEAD	2fbc7b5 unveraendert
git status	unveraendert
Bot in-container PID	363 unveraendert
Worker in-container PID	1 unveraendert
BINANCE_TESTNET	true
managed_proposals / history	0 / 0
docker cp / restart / stop	0
Bot-Code-Touch	0 — pure read
Threshold-Tuning / Runtime-Change	0
Mainnet	0
Push	0
Output-Status	STOP nach Bericht (Operator-Pin)

**Status:** **REVIEW FERTIG** Backlog gepflegt. **STOP** per Operator-Pin.

20 Backlog-Items in 5 Prio-Blocken; 8 davon Mainnet-Blocking. Empfohlene Reihenfolge: P0-Block-A → P0-Block-B → P1-Block-C → P2-Block-D → P3-Block-E.