

T-SPLIT-2-POSITION-EMIT — Plan-Review

Datum: 2026-05-15 17:18 UTC **Status:** Plan-Review — read-only, 0x Code, 0x DB, 0x Restart, 0x Migration **Master HEAD:** 073bb91 (gui-design-1b-fu1) **Mainnet-Relevanz:** keine (read-only Recon, kein Order-Pfad)

1. Executive Summary

Die "Legacy / Unknown 5 positions / \$3,844"-Anzeige im Dashboard ist **kein GUI-Bug** und **keine Strategie-Drift**. Es ist ein **Call-Site-Bug im Bot**:

- `decision_logs.strategy_group` wird korrekt mit `t1_core` befüllt (12,518 Rows in 24h).
- `execute_buy()` akzeptiert **bereits seit T-SPLIT-2 den Parameter `strategy_group`** — `_new_pos['strategy_group']` wird gesetzt, falls vorhanden.
- **ABER:** an 3 der 6 `execute_buy()` -Call-Sites in `main.py` wird `strategy_group` schlicht **nicht übergeben** — nur an der T3-forwarded Call-Site ist es explizit `STRATEGY_GROUP_T2` gesetzt.
- → `_new_pos['strategy_group']` bleibt unset → `emit_position_snapshot(strategy_group=_pos.get('strategy_group'))` schreibt NULL → COALESCE-Bucket "Legacy / Unknown".

Empfehlung: GO für T-SPLIT-2-POSITION-EMIT als minimalen Code-Fix (~3 Zeilen-Adds in `main.py` plus 1 Operator-Decision für Web-/Bot-Channel-Buckets). **Keine Migration nötig** (alle Spalten existieren). **Keine historische DB-Mutation** (Operator-Pin: "alte Snapshots bleiben Legacy"). Bot-Touch erforderlich → braucht explizites GO in der Implementations-Phase.

2. Root Cause (Phase 3)

2.1 `execute_buy()` ist bereits T-SPLIT-2-fähig

`trading/execution/paper_trade.py:316-321`:

```
def execute_buy(self, symbol: str, price: float, quantity: float,
                stop_loss: float, take_profit: float,
                allow_add: bool = False,
                strategy_group: Optional[str] = None, # ← akzeptiert
                decision_id: Optional[str] = None,
                strategy_id: Optional[str] = None) -> Optional[Dict]:
```

`paper_trade.py:486`:

```
if strategy_group is not None:
    _new_pos['strategy_group'] = strategy_group
```

`live_trade.py:641-642` identisch.

2.2 `emit_position_snapshot()` liest aus `_pos`

`trading/main.py:381` (`status='open'` Snapshot pro Scan-Cycle):

```
emit_position_snapshot(
    ...
    strategy_id=_pos.get('strategy_id'),
    strategy_group=_pos.get('strategy_group'), # ← liest aus _pos
    ...
)
```

→ Wenn `_pos['strategy_group']` nicht gesetzt wurde, wird NULL geschrieben.

2.3 Die 6 `execute_buy()` Call-Sites in `main.py`

Line	Quelle	<code>strategy_group</code>	Status
737	DCA add (<code>allow_add=True</code>)	nicht übergeben	✓ OK — DCA soll Original behalten
749	DCA add 2 (<code>allow_add=True</code>)	nicht übergeben	✓ OK
773	T1 Main-Scanner BUY (tranche_1)	fehlt	🚫 BUG — sollte <code>STRATEGY_GROUP_T1</code>
891	Web-Scraper BUY (tranche_1_web)	fehlt	🚫 BUG — Operator-Decision: T1 oder T2?

1006	Telegram-Bot-Channel BUY (tranche_1_bot)	fehlt	🐛 BUG — Operator-Decision: T1 oder T2?
1196	T3-forwarded BUY (tranche_1_t3fwd)	STRATEGY_GROUP_T2	✓ OK — Pump/Dump-Subsystem

3 Call-Sites müssen `strategy_group=...` **ergänzen**. Eine davon ist sicher T1 (Main-Scanner), zwei brauchen Operator-Decision.

2.4 Warum sind alle 5 aktuellen Fresh-Open NULL?

Die 5 Live-Positionen (BERA, FIL, LAYER, SAND, SXT) wurden alle **VOR DATA-LINK-1-Cutover (2026-05-15 11:33 UTC)** geöffnet — DB-Beweis: alle 3 Felder (`strategy_id`, `strategy_group`, `decision_id`) sind NULL, `metadata_json` leer. Diese 5 sind **pre-Fix-Bestand** und werden auch nach T-SPLIT-2-POSITION-EMIT-Implementation NULL bleiben (Operator-Pin: keine historische Mutation).

Erst **NEUE Buys nach T-SPLIT-2-POSITION-EMIT-Cutover** werden korrekt klassifiziert auf der GUI erscheinen.

3. DB-Kontrakt heute (Phase 2)

3.1 Spalten-Existenz (alle vorhanden — keine Migration)

Tabelle	strategy_group	strategy_id	decision_id
decision_logs	✓ VARCHAR	✓ VARCHAR	✓ VARCHAR
position_snapshots	✓ VARCHAR	✓ VARCHAR	✓ VARCHAR
trade_logs	✓ VARCHAR	✓ VARCHAR	✓ VARCHAR

3.2 Befüllung heute

Tabelle	strategy_group=NULL	t1_core	t2_pump_dump
decision_logs (24h)	0	12,518	0
position_snapshots (alle)	18,157	0	407
position_snapshots fresh-open	5	0	0
trade_logs (closed)	54	0	2

→ Bot kann `t1_core` (in `decision_logs`) und `t2_pump_dump` (in `trade_logs/position_snapshots` historisch) schreiben. Lücke ist nur **position_snapshots für aktuelle Live-Positionen**.

3.3 Fresh-Open Sample (alle pre-cutover)

```

created_at      symbol      sid      sg      did      metadata
2026-05-15 17:09:52 BERA/USDT  NULL    NULL    NULL    {}
2026-05-15 17:09:52 FIL/USDT  NULL    NULL    NULL    {}
2026-05-15 17:09:52 LAYER/USDT NULL    NULL    NULL    {}
2026-05-15 17:09:52 SAND/USDT NULL    NULL    NULL    {}
2026-05-15 17:09:52 SXT/USDT  NULL    NULL    NULL    {}

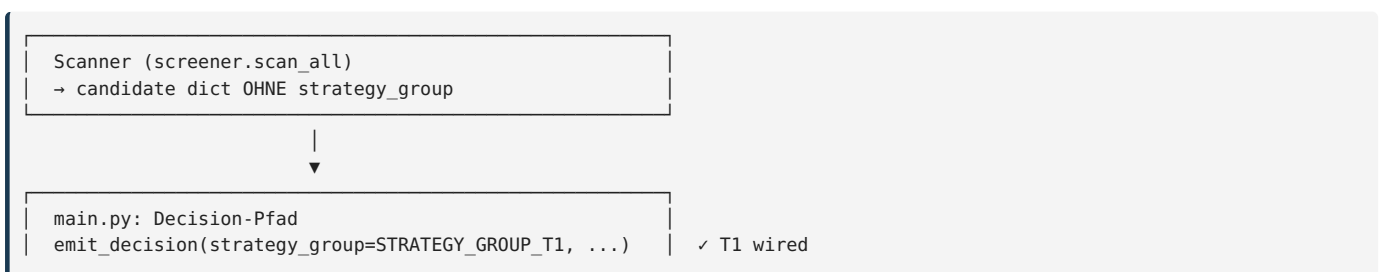
```

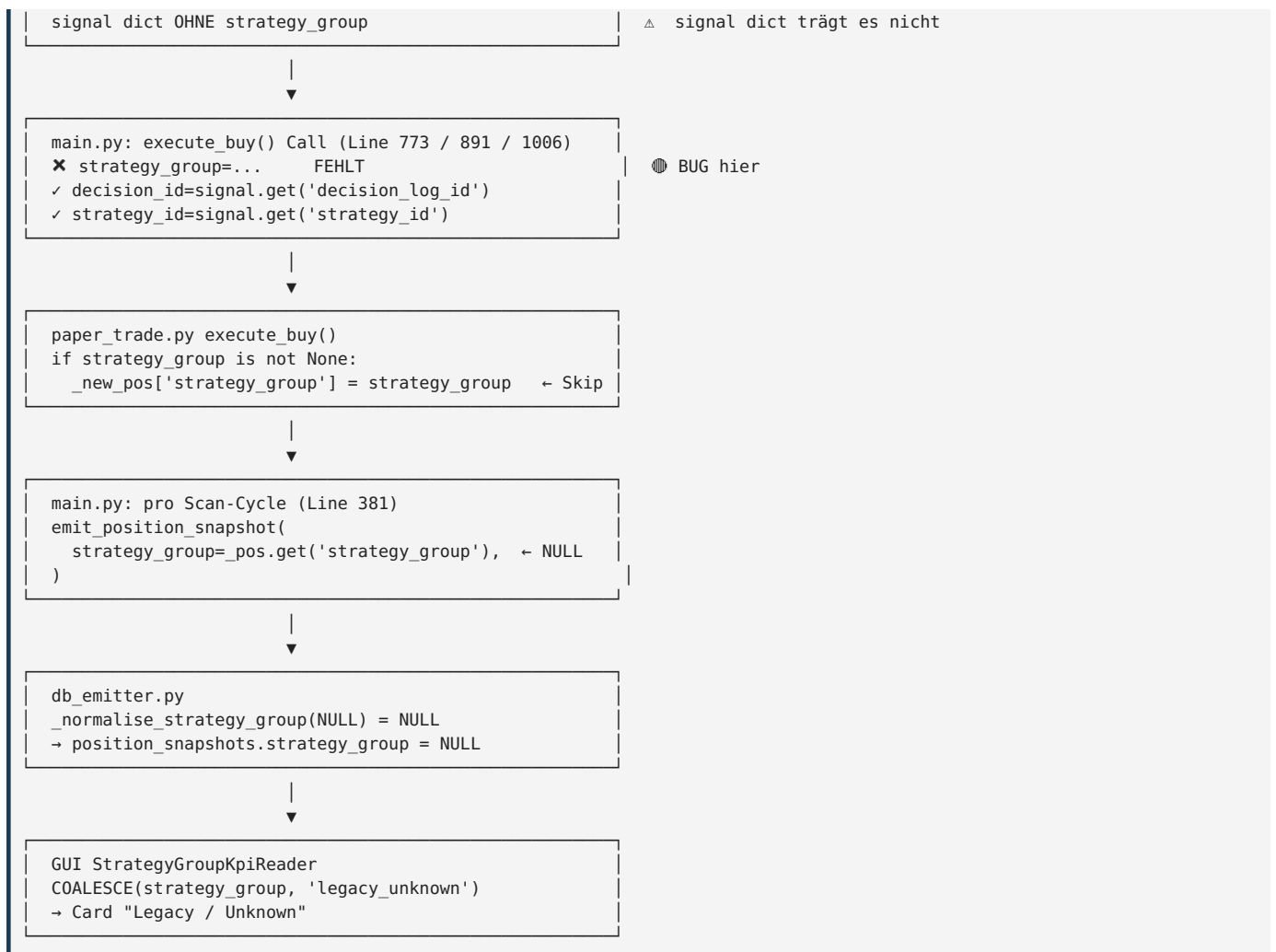
Beobachtung: `metadata_json` ist leer (`{}`), nicht nur `strategy_group` fehlt. Der `metadata=` Parameter im `emit_position_snapshot(status='open')` an Line 381 wird in `main.py` **nicht gesetzt** — das ist außerhalb des T-SPLIT-2-POSITION-EMIT-Scopes, aber als Beobachtung relevant.

3.4 trade_logs since DATA-LINK-1 cutover

0 Rows seit 11:33 UTC — kein Close in 5h45min. Wenn DATA-LINK-1's `strategy_id`-Propagation funktioniert, sehen wir das beim ersten echten Close. Aktuell noch unverifiziert.

4. Code-Flow heute (Phase 3)





5. Ziel-Datenvertrag (Phase 4)

5.1 Bei neuer Entry-Decision

decision_logs (bereits OK): - decision_id ✓ wird gesetzt - strategy_group ✓ T-SPLIT-2 markers überall im main.py - strategy_id ● derzeit immer NULL (Backlog separates Thema — kein Mapping verfügbar im Bot heute)

5.2 Bei neuem BUY

execute_buy() muss bekommen: - decision_id ✓ bereits propagiert (DATA-LINK-1) - strategy_id ✓ bereits propagiert (DATA-LINK-1) - strategy_group ← **NEU für die 3 Call-Sites (T-SPLIT-2-POSITION-EMIT Scope)**

Position-State _new_pos speichert (paper/live identisch): - decision_id ✓ - strategy_id ✓ - strategy_group ✓ wird übernommen wenn übergeben - entry_time ✓

5.3 Bei DCA / Pyramid / Rescue (allow_add=True)

Aktuelle Logik bewahrt Original — Zeile 737/749 übergeben **kein** strategy_group und allow_add=True überspringt den _new_pos -Pfad in execute_buy . **Keine Änderung nötig**. Pin: nicht überschreiben.

5.4 Bei position_snapshot

- strategy_group ✓ wird aus _pos.get('strategy_group') gelesen (existing)
- strategy_id ✓ wird aus _pos.get('strategy_id') gelesen (existing)
- decision_id ● wird in metadata=t mitgeführt beim CLOSE-Path (Line 341), aber **NICHT** als eigene Spalte beim status='open' - Path (Line 381). Außerhalb T-SPLIT-2 Scope — Backlog-Pin.

5.5 Bei Close

- trade_logs.strategy_group ● wird in live_trade.py:769 aus pos['strategy_group'] propagiert (existing), **falls Position das Feld trägt** — was nach T-SPLIT-2-POSITION-EMIT der Fall sein wird für neue Trades.

6. Mapping-Empfehlung (Phase 5)

strategy_group kommt **nicht** aus dem Scanner-Signal als Feld — sondern wird **per-Call-Site als Konstante** gesetzt (analog wie für emit_decision schon gemacht).

Call-Site	line	Empfehlung	Begründung
Main-Scanner T1	773	STRATEGY_GROUP_T1	Pfad ist explizit "T1 [main scan]" — alle emit_decision-Vorschritte sind T1
Web-Scraper	891	Operator-Decision	Sub-System für externe Signal-Quellen; Empfehlung: STRATEGY_GROUP_T2 (pump/dump-typische Signale) — aber Bestätigung erforderlich
Telegram-Bot-Channel	1006	Operator-Decision	Externe Bot-Channel-Signale; Empfehlung: STRATEGY_GROUP_T2 analog Web — Bestätigung erforderlich
T3-forwarded	1196	STRATEGY_GROUP_T2 (existing)	bereits korrekt ✓

Operator-Decisions D1+D2: Web-Channel + Telegram-Bot-Channel = T1 oder T2?

Argument für **T2 (Pump & Dump)**: - Operator-Description T2: "Pump & Dump / Volatility / Listing / DEX momentum" - Web-Scraper holt typischerweise hype-getriebene news-Signale → pump-Charakter - Telegram-Channels sind häufig pump/dump-Signal-Quellen

Argument für **T1 (Standard)**: - Manche Web-Channels sind seriöse News-Aggregatoren - Telegram kann auch standard-trading-Signale liefern

Kein hartes Fantasie-Mapping ohne Bestätigung. **STOP-Trigger:** wenn unklar, separates Plan-Sub-Review.

7. Migration: NEIN

Alle benötigten Spalten existieren bereits in: - position_snapshots: strategy_group VARCHAR, strategy_id VARCHAR, decision_id VARCHAR, opened_at TIMESTAMPTZ - decision_logs: strategy_group VARCHAR, strategy_id VARCHAR, decision_id VARCHAR - trade_logs: strategy_group VARCHAR, strategy_id VARCHAR, decision_id VARCHAR, opened_at TIMESTAMPTZ

→ Reine Code-Änderung. **0 Migrationen.**

8. Minimaler Implementierungsplan (Phase 7)

Geänderte Files: 1 (trading/main.py), geschätzte LoC: +3 (modulo 2 Operator-Decisions = +6).

8.1 main.py:773 — T1 Main-Scanner BUY

```
trade = live_trader.execute_buy(
    symbol=symbol,
    price=current_price,
    quantity=qty,
    stop_loss=position['stop_loss'],
    take_profit=position['take_profit'],
+ # T-SPLIT-2-POSITION-EMIT: T1 main-scanner BUY — pin origin-tag
+ # on the position-state so subsequent emit_position_snapshot calls
+ # surface this as T1 Standard instead of Legacy / Unknown.
+ strategy_group=STRATEGY_GROUP_T1,
  decision_id=signal.get('decision_log_id'),
  strategy_id=signal.get('strategy_id'),
)
```

8.2 main.py:891 — Web-Scraper BUY (D1-pending)

```
+ # T-SPLIT-2-POSITION-EMIT: web-channel — Operator-Decision D1
+ strategy_group=STRATEGY_GROUP_T<X>,
  decision_id=signal_entry.get('decision_log_id'),
```

8.3 main.py:1006 — Telegram-Bot-Channel BUY (D2-pending)

```
+ # T-SPLIT-2-POSITION-EMIT: telegram-bot-channel — Operator-Decision D2
+ strategy_group=STRATEGY_GROUP_T<X>,
  decision_id=bot_signal_entry.get('decision_log_id'),
```

8.4 Out of Scope

- `paper_trade.py / live_trade.py` — unverändert (schon T-SPLIT-2-fähig)
- `db_emitter.py` — unverändert (akzeptiert schon `strategy_group`)
- `main.py:381 emit_position_snapshot(status='open')` — unverändert (liest schon `_pos.get('strategy_group')`)
- `main.py:341 emit_position_snapshot(status='closed')` — unverändert (liest schon `t.get('strategy_group')`)
- `proposal_engine.py` — unverändert (separat: managed proposals)

8.5 Cutover-Pfad

1. Repo-Edit auf master
2. `docker compose build clawbot && docker compose up -d --no-deps clawbot` (SOT-1d, ~20s)
3. 3-Way-MD5-Verify Repo==Image==Container für main.py
4. Bot-Restart (Container) ist nötig — fällt unter "Bot-Restart" Boundary → braucht **explizites Operator-GO** in der Implementation-Phase
5. Watchdog-Freeze während Recreate (analog vorherige Cutovers)
6. Live-Verify nach ersten 1-2 BUY-Cycles: neue `position_snapshots` haben `strategy_group` gesetzt

Commit-Vorschlag: `t-split-2-position-emit: propagate strategy_group at BUY call-sites`

9. Test-Plan (Phase 8)

9.1 Unit-Tests (~80 LoC) in `trading/tests/test_t_split_2_position_emit.py`

1. `test_main_scan_buy_persists_t1_core` — Mock `execute_buy` assert `strategy_group='t1_core'` im `_new_pos`
2. `test_dca_allow_add_preserves_strategy_group` — bestehende Position hat `_pos['strategy_group']='t1_core'`, `allow_add=True` → unverändert
3. `test_t3_forwarded_keeps_t2_pump_dump` — Regression: Line 1196 weiterhin T2
4. `test_web_channel_buy_persists<decision>` — abhängig von D1
5. `test_telegram_bot_channel_buy_persists<decision>` — abhängig von D2
6. `test_execute_sell_forwards_strategy_group_to_trade_record` — Regression: `live_trade.py:769`

9.2 Live-Cutover-Verify (manuell post-deploy)

```
-- Nach dem nächsten BUY-Cycle (~2-5min nach Recreate):
SELECT created_at, symbol,
       COALESCE(strategy_group, '(NULL)') AS sg,
       COALESCE(strategy_id, '(NULL)') AS sid,
       COALESCE(decision_id, '(NULL)') AS did
FROM position_snapshots
WHERE status='open' AND created_at > '<cutover_ts>'
ORDER BY created_at DESC LIMIT 10;
```

Erwartung: alle neuen Rows haben `strategy_group='t1_core'` (oder T2 wenn T3-forwarded). NULL nur noch in den 5 pre-cutover Rows.

9.3 GUI-Verify

- `/admin` Monitoring-Dashboard:
- "T1 Standard" Card erhält erstmals **Live-Positions-Count > 0** + Exposure > 0
- "Legacy / Unknown" Card stagniert/sinkt mit jedem Position-Close (alte 5 schließen sich nach und nach)
- `StrategyGroupKpiReader` hat sich seit FU1 als fresh-filter+DB-agnostisch korrekt erwiesen → keine Reader-Änderung nötig

9.4 Boundaries-Test

- `BINANCE_TESTNET=true` unverändert
- `LIVE_TRADING_ENABLED=true` unverändert
- `MULTI_STRATEGY_ENABLED=false` unverändert (DATA-LINK-1-FU2 Pin)
- Mainnet-Guards in `_get_exchange` unverändert
- 0 Strategie-Param-Änderung
- 0 künstliche Orders

10. Risikoanalyse (Phase 9)

#	Risiko	Wahrscheinlichkeit	Impact	Mitigation
R1	Falsche T1/T2-Klassifizierung an Web-/Bot-Channel-Call-Sites	mittel	niedrig	Operator-Decisions D1+D2 vor Implementation
R2	DCA überschreibt Entry-Attribution	sehr niedrig	niedrig	<code>allow_add=True</code> skip-pfad existiert seit T-SPLIT-2

				+ DATA-LINK-1
R3	T3-forwarded fällt versehentlich auf T1 zurück	sehr niedrig	niedrig	Line 1196 wird NICHT angefasst in diesem Scope
R4	Neue NULLs in position_snapshots durch übersehene Call-Site	sehr niedrig	niedrig	grep+audit zeigt nur 6 execute_buy() in main.py — komplett abgedeckt
R5	GUI versteckt echte Unknowns	nicht-Risiko	n/a	Legacy/Unknown Card bleibt sichtbar (Operator-Pin)
R6	Historische 18,157 NULL-Rows werden falsch interpretiert	gering	gering	nicht-mutation. Hinweis: GUI rendert sie weiter unter Legacy/Unknown
R7	MultiStrategyRunner-Pfad bricht	nicht-relevant	n/a	MULTI_STRATEGY_ENABLED=false + DATA-LINK-1-FU2-Pin bleibt aktiv
R8	DATA-LINK-1-FU2 (MS-Runner) wird komplizierter	niedrig	niedrig	MS-Runner muss ohnehin neu wired werden — FU2 dann gleich strategy_group mitnehmen
R9	Mainnet-Verwechslung	sehr niedrig	hoch	Read-only Plan-Phase. Implementation: explizit BINANCE_TESTNET-Verify vor Recreate
R10	Bot-Restart-Risiko (stale bot.pid PID-Lock)	niedrig	niedrig	WATCHDOG-PID-CLEANUP (commit 54a554f) live — cmdline-aware acquire_pid_lock

Höchstes Risiko: R1 — wird durch zwei Operator-Decisions vor Implementation aufgelöst.

11. GO/NO-GO

Empfehlung: GO mit 2 Operator-Decisions

Decision	Frage	Empfehlung	Konsequenz
D1	Web-Scraper BUY → Strategy-Group?	STRATEGY_GROUP_T2 (Pump-typisch)	external signal sources werden in T2 gebucketed
D2	Telegram-Bot-Channel BUY → Strategy-Group?	STRATEGY_GROUP_T2 (Pump-typisch)	dito
(D3)	T-SPLIT-2-POSITION-EMIT-IMPL JETZT oder erst nach Operator-Aktivierung anderer Set-C-Widgets?	Sofort, blockt nichts	T1-Card wird mit Live-Daten gefüllt

Implementierungs-Aufwand

Block	Aufwand
Code-Änderung main.py (3 Call-Sites × ~2 Zeilen)	10 min
Unit-Tests (6 Cases ~80 LoC)	1 h
Docker build + Bot-Recreate (SOT-1d-Pfad)	5 min
Watchdog-Freeze + Verify + Live-Smoke	30 min
Gesamt T-SPLIT-2-POSITION-EMIT-IMPL	~2 h

Boundaries

0× DB-Migration · 0× DB-Writes außer Tests · 0× historische Mutations · 0× Strategieparameter · 0× Orders künstlich · 0× Mainnet · 0× neue Routes · 0× Push · 0× Secrets · BINANCE_TESTNET=true.

Bot-Restart (Container-Recreate) ist erforderlich für Cutover → **separates explizites GO für Implementation-Phase erforderlich**, weil "Kein Bot-/Worker-Restart" in den Plan-Review-Boundaries steht.

12. STOP

Plan-Review abgeschlossen. Operator-Entscheidungen ausstehend: - **D1**: Web-Scraper-Channel → T1 oder T2? - **D2**: Telegram-Bot-Channel → T1 oder T2? - **D3 (Impl-GO)**: Nach D1+D2 explizites GO für T-SPLIT-2-POSITION-EMIT-IMPL (inkl. Bot-Restart-Freigabe)

Bot weiterhin healthy: PID 1229553 alive, decision_logs 17:07:42 UTC, 0 Tracebacks, BEAR cap=5 voll.

