

Steve-TradingBot — 2-Wochen-Roadmap v4 (FINAL, operator-corrected)

Erstellt: 2026-05-16 · Branch `master` · HEAD `a6a629d` Zeitraum: 2026-05-19 (Mo) ab Phase A Status: **Plan**, keine Implementation in diesem Dokument. Ersetzt v3.

0. Was sich gegenüber v3 geändert hat

Operator-Korrekturen vom 2026-05-16 abend übernommen:

v3	v4
Phase C als ein großer Block (alle Quality-Filter zugleich blockierend)	C1 Shadow + C2 Enforce (24-48h Beobachtung dazwischen)
Phase E als ein Block (BE + TimeStop + Partial-Profit + Momentum-Trailing)	E1 (BE + TimeStop) + E2 (Partial-Profit) separat
Reihenfolge A→B→C→D→E→F	A→B→C1→D, Decision-Gate , dann C2/E1/E2/F
Taxonomie: <code>t1_core</code> + <code>t2_pump_dump</code> + <code>t3_copy_trading</code>	<code>t1_core</code> + <code>t2_solana_pump</code> + <code>t3_copy_trading</code> + <code>legacy_unknown</code>
MS-Runner <code>breakout</code> / <code>volatility_sweep</code> → <code>t2_pump_dump</code> (heute <code>a6a629d</code>)	alle 5 MS-Strategien → <code>t1_core</code> (Rework in Phase A)
external Web/Telegram-Channel BUYS → <code>t2_pump_dump</code>	→ <code>t1_core</code> mit Binance-Gate als Vorbedingung
<code>_process_t3_forwarded_signals</code> als T3→T2 Bridge	disabled, gibt <code>reject_reason='t3_forward_disabled_new_arch'</code>
Phasen-Workflow implizit	6-Schritt-Workflow pro Phase (Plan-Review → Minimal-Impl → Tests → SOT-1d → 24h Beobachtung → next)

1. Neue Taxonomie (Option A: T1=Binance, T2=Solana, T3=Copy)

<code>t1_core</code>	→ ALLE Binance-Spot-Testnet-Trades, unabhängig von Quelle: <ul style="list-style-type: none">- Legacy main-scanner score-threshold pipeline- MS-Runner alle 5 Strategien (<code>trend_follow</code>, <code>breakout</code>, <code>mean_reversion</code>, <code>vwap_mean_reversion</code>, <code>volatility_sweep</code>)- External web-channel signals- External telegram-bot/telegram-channel signals- (T3-Forward-Bridge wird deaktiviert)
<code>t2_solana_pump</code>	→ ALLE Solana-Chain-Trades (zukünftig, NICHT aktiv). Reserviert für T2-Greenfield-Implementation (Pump.fun + Raydium + Jupiter swap).
<code>t3_copy_trading</code>	→ Copy-Trading (zukünftig, deaktiviert). <code>`COPY_TRADING_ENABLED=false` + tier_arch DEACTIVATED.</code>
<code>legacy_unknown</code>	→ Pre-T-SPLIT-2 historische Rows. Nur historisch, wird nie neu geschrieben.

Granularität pro Strategie liegt in `strategy_id`, nicht in `strategy_group`.

2. Per-Phase Workflow (durchgehend)

Operator-vorgegeben für jede Phase:

1. Plan-Review (kurz) 5 min
2. Minimal-Implementation 1-6 h
3. Tests 1-2 h
4. SOT-1d Cutover 30 min
5. 24h Beobachtung 24h (passive, wenn Trading-Logik betroffen)
6. Erst dann nächste Phase

Strikt befolgen — **niemals zwei Trading-Logik-Änderungen im selben Cutover**. Sonst lässt sich nicht trennen, welche Änderung welchen Effekt hat.

3. Phase A — TIER-ARCH-CONTRACT-1 (größer als in v3)

Phase A ist jetzt **1.5 Tage** statt 0.5 Tage, weil sie deutlich mehr enthält. Ist die wichtigste Phase — alles danach baut darauf auf.

3.1 Sub-Komponenten

A.1 — Tier-Status-Code-Vertrag: `trading/tier_arch.py`

```
class TierStatus(Enum):
    ACTIVE      = "active"
    PLAN_ONLY  = "plan_only"
    DEACTIVATED = "deactivated"

    TIER_ARCH = {
        "t1_core":      TierStatus.ACTIVE,
        "t2_solana_pump": TierStatus.PLAN_ONLY,
        "t3_copy_trading": TierStatus.DEACTIVATED,
    }
```

A.2 — DB CHECK-Constraint-Erweiterung: neue Postgres-Migration, die `t2_solana_pump` zur CHECK-Allowlist hinzufügt UND `t2_pump_dump` als historisch erlaubt belässt (3 trade_logs + 510 position_snapshots existieren mit altem Wert, nicht mutieren).

```
ALTER TABLE trade_logs DROP CONSTRAINT trade_logs_strategy_group_check;
ALTER TABLE trade_logs ADD CONSTRAINT trade_logs_strategy_group_check
CHECK (strategy_group IS NULL OR strategy_group IN (
    't1_core', 't2_solana_pump', 't3_copy_trading',
    'legacy_unknown',
    't2_pump_dump' -- historisch, neue Writes verboten
));
-- gleiche Logik für position_snapshots + decision_logs
```

Operator-Decision-Punkt: war bisher Boundary "0x DB-Migration". Phase A braucht **eine kleine Migration** für die Taxonomie-Umstellung. Begründet weil Codeseite das neue Wertset emittiert; ohne Migration würde der Bot 100 % rejecten.

A.3 — `db_emitter.py` Konstanten umbenennen:

```
STRATEGY_GROUP_T1      = "t1_core"
STRATEGY_GROUP_T2      = "t2_solana_pump" # vorher: "t2_pump_dump"
STRATEGY_GROUP_T3      = "t3_copy_trading"
STRATEGY_GROUP_LEGACY  = "legacy_unknown"
_STRATEGY_GROUP_ALLOWED = frozenset({...alle 4 + 't2_pump_dump' als legacy-readonly...})
```

A.4 — `strategy_group_map.py` Rework (das ist die Rückgängig-Korrektur von heutigem `a6a629d`):

```
STRATEGY_ID_TO_GROUP = {
    "trend_follow":      STRATEGY_GROUP_T1,
    "mean_reversion":    STRATEGY_GROUP_T1,
    "vwap_mean_reversion": STRATEGY_GROUP_T1,
    "breakout":          STRATEGY_GROUP_T1, # vorher T2 – NEU: t1_core
    "volatility_sweep":  STRATEGY_GROUP_T1, # vorher T2 – NEU: t1_core
}
```

A.5 — `main.py` Re-Tagging: - Lines 997 (web-channel) + 1114 (telegram-bot-channel): `STRATEGY_GROUP_T2` → `STRATEGY_GROUP_T1` - Vorausgesetzt das Binance-Symbol-Gate (Phase B) ist installiert; in Phase A schon `if not is_active('t1_core'): return` einbauen.

A.6 — `_process_t3_forwarded_signals` deaktivieren:

```
def _process_t3_forwarded_signals(tier3, live_trader, reporter, ...):
    logger.info("T3-Forward-Bridge deaktiviert (Plan v4 / neue Tier-Architektur).")
    forwarded = getattr(tier3, 'forwarded_signals', [])
    for sig in forwarded:
        emit_decision(
            symbol=sig.get('symbol'),
            action='reject',
            reject_reason='t3_forward_disabled_new_arch',
            strategy_group=STRATEGY_GROUP_T1, # tag as "would have been T1"
            source='_process_t3_forwarded_signals',
            metadata={'tier3_signal_summary': str(sig)[:200]},
        )
    tier3.forwarded_signals.clear()
```

→ Signale werden nicht mehr executed, aber sauber dokumentiert. Operator sieht in DecisionLog welche Forwards früher passiert wären.

A.7 — Tier3-Subsystem-Forwarding-Code retieren: - `Tier3Loop.should_forward_to_t1` → log-only (kein Append zu `forwarded_signals` mehr) - ODER: komplett aus Phase A herauslassen und in Phase H als Cleanup → empfohlen, weil sonst Phase A zu groß

A.8 — Tests: - 4 Tier-Status-Tests (T1=active, T2=plan_only, T3=deactivated, fallback) - 2 db_emitter CHECK-Constraint-Tests (neue Werte akzeptiert, alte erlaubt-readonly) - 5 strategy_group_map Tests (alle 5 Strategien → t1_core) - 3 AST-Guards (main.py importiert tier_arch, MS-Runner check) - 4 main.py Re-Tag Tests (external channels jetzt t1_core) - 2 T3-Forward-Disable Tests (emit

reject, kein execute_buy)

3.2 Aufwand

Block	Zeit
tier_arch.py + Code-Wiring	2 h
DB-Migration + Postgres-Test	2 h
db_emitter + strategy_group_map Rework	1 h
main.py Re-Tagging	1 h
<code>_process_t3_forwarded_signals</code> disable	1 h
~20 Tests	3 h
Build + Cutover + 24h Verify	2 h Operator-Beobachtung

Gesamt ~12 h = 1.5 Arbeitstage netto + 24h Beobachtungs-Fenster.

3.3 Stop-Gates Phase A

- alle ~20 Tests grün
- DB-Migration in Test-DB durchgelaufen
- Bot-Recreate clean
- erste 5 decision_logs zeigen `strategy_group='t1_core'` (vorher hätten manche `t2_pump_dump` bekommen)
- `_process_t3_forwarded_signals` -Reject-Log mit `reject_reason='t3_forward_disabled_new_arch'` erscheint min. 1x (wenn Tier3 was forwardet)
- 24h: 0 Tracebacks

4. Phase B — T1-BINANCE-SYMBOL-GATE-1

Unverändert gegenüber v3. **1 Tag.**

Validation **VOR** jedem execute_buy: - Symbol exists in Binance Spot Testnet - Quote-Asset = USDT - Order-Filter erfüllbar: PRICE_FILTER, LOT_SIZE, MIN_NOTIONAL, Tick-Size, Step-Size - Bei Reject: `emit_decision(action='reject', reject_reason='symbol_not_on_binance' | ...)`.

Aufwand: 6 h. **24h Beobachtung danach Pflicht.**

5. Phase C1 — T1-QUALITY-SCORE-SHADOW-1

Shadow-Mode: Quality-Score wird berechnet und in `decision_logs.metadata_json` geloggt, **aber NICHT als Block-Bedingung verwendet.**

Operator sieht über 24-48h: - Welche Quality-Sub-Scores wie häufig unter Threshold landen - Welche No-Trade-Zone wie oft greifen würde - Welche Regime-Threshold-Änderung wie viele Trades blockieren würde

Damit kann Phase C2 später mit **realen Daten** statt mit synthetischen Annahmen aktiviert werden.

Sub-Scores (alle 0-1, Shadow-only)

Sub-Score	Berechnung
<code>liquidity_score</code>	24h Volume / median(24h Vol top-50)
<code>spread_score</code>	1 - (spread / max_spread_allowed)
<code>volume_burst_score</code>	recent_volume / 20-MA-Volume
<code>regime_fit_score</code>	regime ∈ allowed_for_strategy
<code>trend_quality_score</code>	4h-trend + ema-stack-coherence
<code>distance_to_resistance</code>	(resistance - price) / ATR
<code>btc_health</code>	24h BTC % + 1h BTC-Crash-Check
<code>mtf_alignment</code>	4h trend matches 15m direction

No-Trade-Zone-Score (Composite, Shadow-only)

```

ntz_blocked = any([
    spread > 0.5%,
    volume_24h < 1M_USD,
    distance_to_resistance < 1.0 * ATR,
    last_4h_candle > +8%,
    btc_1h < -2%,
    (regime == 'HIGH_VOL' and base_regime == 'BEAR'),
])

```

→ In Shadow: nur Logged. In Enforce (C2): tatsächliches Block-Verhalten.

Regime-Adjusted-Thresholds (Shadow-only)

Regime	combined_score Threshold (würde gelten)	quality_threshold (würde gelten)
BULL	6.5	0.50
STRONG_TREND	6.0	0.50
RANGE / WEAK_TREND	7.0	0.55
BEAR	8.0	0.65
HIGH_VOL	8.5	0.70
HIGH_VOL+BEAR-Combo	blocked	n/a

Aufwand

~8 h. **24-48h Beobachtung Pflicht** vor Phase C2.

6. Phase D — T1-RISK-GUARD-1

Identisch zu v3. **2 Tage**.

- **SL-Never-Worse-Invariant:** SL darf nach Entry nie schlechter werden (außer DCA-Rebuild mit Max-Loss-Check)
- **DCA-Guards:**
 - BEAR-Regime → DCA komplett geblockt (BERA-Lesson)
 - Reclaim-Requirement: nur wenn Preis über altem SL + ATR war
 - Max-Loss-Simulation: simulierter Worst-Case darf Cap nicht übersteigen
 - DCA-Count-Cap (existing)
- **Volatility-Scaled Position-Sizing:** kleinere Positionen bei hoher ATR

15 Tests (Invariant + DCA-Guards + Vol-Sizing + BERA-Replay).

24h Beobachtung Pflicht — wenn DCA in BEAR vor diesem Cutover noch passierte, MUSS er danach durchgängig blockiert sein.

7. DECISION-GATE nach Phase D

Operator-Pflichtgate. Nach Phase A + B + C1 + D ist Bot in einem stabilen, sicheren Zustand. Operator entscheidet:

Option	Konsequenz
GO C2 + E1 + E2 + F + G + H	Restliche Phasen wie geplant
GO nur C2 (Quality Enforce)	Erst Quality scharf schalten, andere Phasen später
GO nur E1 (BE + TimeStop)	Erst Exit-Optimierung, dann andere
STOP + Daten sammeln	Bot läuft 1+ Woche, sammelt mehr Daten für Decision
Rollback einzelne Phase	wenn unerwartete Probleme

Operator sieht zu dem Zeitpunkt: - C1-Shadow-Daten (welche Quality-Filter würden wie oft blocken) - D-Risk-Guard Effekt (DCA-Verhalten, SL-Invariant-Verletzungen, Vol-Sizing-Impact) - Live-PnL der ersten Phasen - Mainstream Trade-Beispiele
Erst dann Phasen-Sequenz für Woche 2.

8. Phase C2 — T1-QUALITY-SCORE-ENFORCE-1 (nach Decision-Gate)

Aktiviert die Block-Funktionalität aus C1 mit den C1-Daten als Begründung für Threshold-Werte.

Insbesondere: wenn C1 zeigt "BEAR + score-Threshold 8.0 würde 95 % aller Trades blocken", entscheidet Operator ob das so gewollt ist oder ob Threshold relaxed werden muss.

Aufwand: 4 h (kleines Code-Diff: Sub-Scores → Block-Decisions).

9. Phase E1 — T1-EXIT-OPTIMIZER-BE-TIMESTOP-1

Kleinerer Block aus v3-E.

- **Break-Even nach +1R** (statt nach 60h Hold)
- **Time-Stop bei Seitwärtsverhalten** (< +0.5R PnL nach 4h) — zusätzlich zum 96h-Hard-Exit-Backup
- **Momentum-Trailing-Vorbereitung** (Definition + tests, aber noch nicht aktiv — kommt in E2 oder später)

Aufwand: 6 h.

24h Beobachtung vor E2.

10. Phase E2 — T1-PARTIAL-PROFIT-1

Eigene Mini-Architektur. **2 Tage.**

- 50 % der Position sell @ +1R
- 25 % sell @ +2R
- Rest läuft mit Trailing
- **Korrekte Buchführung:** qty-Reduktion, avg/entry Preserve, anteiliger PnL, trade_logs partial-Row (status='partial_closed?'), position_snapshots Update, Chart-Marker, keine Doppel-Closes
- Tests: ~12

Aufwand: 12 h.

11. Phase F — T1-POST-TRADE-LEARNING

Wie v3. Auto-Klassifikation Grade A-F + MAE/MFE in `trade_logs.metadata.json`. **1.5 Tage.**

12. Phase G — EXEC-MODE-LABEL-3 Phase 3a

Wie v3. Paper-Cleanup mit Backward-Compat-Aliasen. **0.5 Tag.**

13. Phase H — PDFs

- REFACTOR-VS-REWRITE-PLAN-PDF
- T2-SOLANA-SHADOW-1-PLAN-PDF

Wie v3. **0.5 Tag.**

14. Tagesplan (v4, mit Decision-Gate)

Tag	Phase	Notizen
W1 Mo (19.05.)	A start	Tier-Arch + DB-Migration + Tests
W1 Di (20.05.)	A → B	A Cutover; Symbol-Gate start
W1 Mi (21.05.)	B → C1	Symbol-Gate Cutover + 24h Beob; C1 start
W1 Do (22.05.)	C1 → D	C1 Cutover (shadow-only); D start
W1 Fr (23.05.)	D	D Cutover + 24h Beob
W1 Sa-So	Review + Decision-Gate	Operator-Decision für Woche 2
W2 Mo (26.05.)	C2 (wenn approved)	Quality Enforce
W2 Di (27.05.)	E1	BE + Time-Stop
W2 Mi (28.05.)	E2 start	Partial-Profit
W2 Do (29.05.)	E2 → F	E2 Cutover; F start
W2 Fr (30.05.)	F → G	F Cutover; G start
W2 Sa (31.05.)	G → H	G Cutover; H PDFs
W2 So (01.06.)	Review	Endabnahme

Total: ~10 Arbeitstage netto + 6 explizite 24h-Beobachtungs-Fenster.

15. Was NICHT in Plan v4

- T2 als Binance-Pump-Profil weiter ausbauen
- MS-Runner-DRY_RUN=false Aktivierung
- Quality-Score sofort hart blockierend (ohne Shadow-Phase)

- Partial-Profit im selben Cutover wie BE + Time-Stop
- Discord + AI (Woche 3+)
- T2 Solana Implementation (Woche 4+)
- T3 Copy Trading (deactivated)
- Mainnet, Leverage, Futures

16. Risiken & Mitigationen

Risiko	Mitigation
DB-Migration in Phase A bricht	Test-DB-Migration als Vorlauf; Rollback-SQL in <code>00_README</code>
<code>_process_t3_forwarded_signals</code> -Disable bricht Pump-Detection	Tier3-Monitor läuft weiter, Signale werden nur nicht ausgeführt; in DB sichtbar als <code>reject_reason='t3_forward_disabled_new_arch'</code>
C1-Shadow zeigt 100 % Block über alle Thresholds	Decision-Gate erlaubt Threshold-Recalibration vor C2-Cutover
D-DCA-Block produziert keine Rescues mehr → erhöhte SL-Hits	erwartetes Verhalten; PnL-Vergleich pre/post 24h
E2-Partial-Profit-Buchhaltung-Bug	extensive Tests + Shadow-Phase als E2.1 möglich (auf Operator-Wunsch)
Inkrementeller Mode dauert > 2 Wochen	Operator entscheidet pro Phase; kein Hard-Deadline

17. Boundaries (durchgehend)

- 0x Mainnet (Binance + Solana)
- 0x MULTI_STRATEGY_DRY_RUN=false
- 0x T2-Solana-Code (Code-Pfad bleibt 0 LoC)
- 0x T3-Copy-Trading
- 0x CommandBus-V6 / MH-1
- 0x Leverage / Futures
- 0x ML ohne saubere Datenbasis
- **AUSNAHME:** Phase A erfordert kleine DB-Migration (CHECK-Constraint-Erweiterung), Operator-explicit-erlaubt
- 0x DB-Mass-Mutation an historischen Rows
- 0x Push auf Remote ohne separates GO
- 0x Secrets im Chat
- 0x env dump
- Niemals 2 Trading-Logik-Änderungen im selben Cutover

18. Per-Phase Stop-Gates

Wie in Sektion 2 dargestellt. Wichtig pro Phase: - Tests müssen grün sein **vor** Cutover - 24h post-Cutover-Beobachtung Pflicht wenn Trading-Logik berührt - Bei unerwartetem Effekt: Rollback erlaubt, Operator-Entscheidung - Niemals nächste Phase ohne explizites Operator-GO

19. Operator-Startsignal

Wenn dieser Plan akzeptiert:

```
GO PHASE A
```

Antwort enthält implizit: - Operator akzeptiert kleine DB-Migration in Phase A - Operator akzeptiert Taxonomie-Korrektur (t2_pump_dump → t2_solana_pump) - Operator akzeptiert das Disable von `_process_t3_forwarded_signals` - Operator akzeptiert Rework des heutigen `a6a629d` -Commits (Mapping)

20. STOP

Plan v4, **operator-corrected**, ersetzt v3. Keine Implementation in diesem Dokument.

Boundaries: 0x Code-Change, 0x Container-Aktion, 0x Push, 0x Secrets.