

Steve-TradingBot — 2-Wochen-Roadmap v3 (T1-Quality-Focus)

Erstellt: 2026-05-16 · Branch `master` · HEAD `a6a629d` **Zeitraum:** 2026-05-19 (Mo) bis 2026-05-31 (So) — 2 Kalenderwochen
Status: **Plan**, keine Implementation in diesem Dokument.

0. Leit-Philosophie (Operator-bestätigt 2026-05-16)

T1 soll **langweilig, robust und Binance-only** sein. Der größte Profithebel ist **weniger schlechte Trades**, nicht mehr Indikatoren. Discord-/AI-Signal-Channels kommen NACH dem Quality-Refactor.

Konkret abgeleitet: - Quality > Quantity (Quality-Floor zusätzlich zum Score-Threshold) - Regime-bewusste Entry-Schwellen (BERA-Lesson: Score 5.0 BEAR war zu locker) - ATR-basierte Stops mit Invariant **"SL darf nie schlechter werden"** - DCA nur unter strengen Bedingungen (kein BEAR-Support-Fallback, Max-Loss-Sim, Reclaim-Requirement) - Volatility-Scaled Sizing statt fixer USDT-Beträge - Multi-Timeframe-Confirmation (4h Marktstruktur + 15m Entry-Timing) - Schnellere Gewinnsicherung (Partial @ +1R/+2R, Break-even früh, Time-Stop bei Seitwärts)

Operator-Boundaries (durchgehend für 2 Wochen): - 0x Mainnet (weder Binance noch Solana) - 0x MULTI_STRATEGY_DRY_RUN=false - 0x T2-Solana-Code - 0x T3-Copy-Trading - 0x CommandBus-V6 / MH-1 - 0x Pump.fun Logik in T1 - 0x ML-Modell ohne saubere Datenbasis - 0x Strategie-Parameter-Änderung ohne Operator-GO

1. Phasen-Reihenfolge (operator-priorisiert)

#	Phase	Aufwand	Woche
A	TIER-ARCH-CONTRACT-1 — Foundation: T1=active, T2=plan_only, T3=deactivated	0.5 Tag	W1 Mo
B	T1-BINANCE-SYMBOL-GATE-1 — Symbol-Existence + Binance-Order-Filter-Validation vor jedem BUY	1 Tag	W1 Mo-Di
C	T1-QUALITY-SCORE-1 — Quality-Floor (liquidity/spread/volume/regime/trend/resistance/4h-MTF/no-trade-zones)	2 Tage	W1 Di-Do
D	T1-RISK-GUARD-1 — DCA-Restrictions + SL-Invariants + Volatility-Scaled Sizing + Regime-Adjustment	2 Tage	W1 Do-Fr / W2 Mo
E	T1-EXIT-OPTIMIZER-1 — Schnelleres BE + Partial @ +1R/+2R + Momentum-Trailing + Time-Stop	2 Tage	W2 Mo-Mi
F	T1-POST-TRADE-LEARNING — Auto-Klassifikation pro Closed Trade (entry/exit/DCA/SL/TP/MAE-MFE/Regime-Fit)	1.5 Tage	W2 Mi-Do
G	EXEC-MODE-LABEL-3 Phase 3a (Backward-Compat Renames)	0.5 Tag	W2 Do-Fr
H	REFACTOR-VS-REWRITE-PDF + T2-SOLANA-SHADOW-1-PLAN-PDF	0.5 Tag	W2 Fr-Sa

Gesamt **~10 Arbeitstage netto** mit Puffer für Live-Beobachtungs-Fenster zwischen Cutovers.

Discord-Channel-Integration + AI-Signal-Analyse sind aus diesem Plan **bewusst ausgelagert** auf Woche 3+ als eigene Phase. Begründung: erst T1-Quality festziehen, dann zusätzliche Signal-Quellen draufsetzen. Sonst bedeutet "mehr Signale" = "mehr schlechte Trades".

2. Phase A — TIER-ARCH-CONTRACT-1 (W1 Mo, halbtags)

Ziel

Tier-Architektur als Code-Vertrag (`trading/tier_arch.py`) — Foundation für alle folgenden Phasen. Fail-fast bei unerlaubten Tier-Zugriffen.

Lieferziel

```
# trading/tier_arch.py
class TierStatus(Enum):
    ACTIVE = "active"
    PLAN_ONLY = "plan_only"
    DEACTIVATED = "deactivated"

TIER_ARCH = {
    "t1_core": TierStatus.ACTIVE,
    "t2_pump_dump": TierStatus.PLAN_ONLY,
    "t3_copy_trading": TierStatus.DEACTIVATED,
}

def is_active(group: str) -> bool: ...
def assert_tier_active(group: str) -> None: ... # RuntimeError on inactive
```

Wiring

- `main.py` Top-Level: `assert_tier_active('t1_core')` vor Scan-Cycle-Loop
- `multi_strategy_runner.py`: `Top-of-run()` `if not is_active('t1_core'): return []`
- `_process_t3_forwarded_signals`: `if not is_active('t2_pump_dump'): return`
- `wallet_tracker.py` (T3): `assert_tier_active('t3_copy_trading')` als zusätzlicher Schutz neben `COPY_TRADING_ENABLED=false`

Tests

7 Tests: 3 Tier-Status-Asserts + 2 Helper-Behavior + 2 AST-Guards (Wiring vorhanden).

Aufwand: 6h

3. Phase B — T1-BINANCE-SYMBOL-GATE-1 (W1 Mo-Di, 1 Tag)

Ziel

T1 darf nur Symbole handeln, die auf Binance Spot Testnet tatsächlich handelbar sind UND deren Order-Filter erfüllbar sind.

Lieferziel

Neue Methode in `LiveTrader`:

```
def validate_symbol_and_order(symbol: str, qty: float, price: float) -> Tuple[bool, Optional[str]]:
    """
    Returns (ok, reject_reason).
    Reject reasons:
    'symbol_not_on_binance'
    'symbol_not_usdt_quote'
    'symbol_not_spot_tradable'
    'price_below_PRICE_FILTER_min'
    'price_above_PRICE_FILTER_max'
    'price_tick_size_mismatch'
    'qty_below_LOT_SIZE_min'
    'qty_above_LOT_SIZE_max'
    'qty_step_size_mismatch'
    'notional_below_MIN_NOTIONAL'
    """
```

Wiring

- `main.py:run_scan_cycle` vor jedem `execute_buy`: gate-call → bei Reject `emit_decision(action='reject', reject_reason=...)`
- `_process_t3_forwarded_signals` analog (auch wenn T3 heute deaktiviert, später greift's automatisch)
- `multi_strategy_runner.py:_attempt_execution` analog (für künftige MS-Aktivierung)
- External-Channel-Paths (web/telegram) analog

Tests

- 6 Reject-Reason-Tests pro Filter-Violation
- 2 Happy-Path-Tests (echtes Binance-Symbol)
- 1 Cache-Refresh-Test (markets reload nach 1h)

Aufwand: 6h

4. Phase C — T1-QUALITY-SCORE-1 (W1 Di-Do, 2 Tage)

Ziel

Quality-Floor zusätzlich zum bestehenden Score-Threshold. Eine Position wird nur eröffnet, wenn BEIDE Bedingungen erfüllt sind:

```
combined_score >= score_threshold    (existing)
quality_score  >= quality_threshold  (NEW)
```

Quality-Sub-Scores (jeder 0-1)

Sub-Score	Berechnung	Reject-Threshold
liquidity_score	24h Volume / median(24h Vol top-50)	< 0.3
spread_score	1 - (spread / max_spread_allowed)	spread > 0.3%
volume_burst_score	recent_volume / 20-MA-Volume	> 5 oder < 0.5 → reject
regime_fit_score	regime ∈ allowed_for_strategy	reject auf miss-fit
trend_quality_score	4h-trend alignment + ema-stack-coherence	< 0.4
distance_to_resistance	(resistance - price) / ATR	< 1.0 → reject
btc_health	24h BTC > -5% AND BTC nicht in 1h-Crash	falsch → reject
mtf_alignment	4h trend matches 15m direction	misalignment → reject

No-Trade-Zone Filter (Sub-Komponente)

Zone	Block-Bedingung
Extreme Spread	spread > 0.5%
Niedriges Volumen	< 1M USD 24h
Direkt unter Widerstand	distance_to_resistance < 1.0 × ATR
Nach grüner Mega-Candle	letzte 4h-Candle > +8%
BTC-Crash	1h BTC < -2%
HIGH_VOL + BEAR-Combo	beides gleichzeitig

Regime-Adjusted Thresholds (statt fixer 7.5)

Regime	combined_score Threshold	quality_threshold
BULL	6.5	0.50
RANGE	7.0	0.55
WEAK_TREND	7.0	0.55
STRONG_TREND	6.0	0.50
BEAR	8.0 (statt 7.5)	0.65
HIGH_VOL	8.5	0.70
HIGH_VOL+BEAR-Combo	blocked	n/a

Tests

- 8 Sub-Score-Berechnungs-Tests (pro Quality-Component)
- 6 No-Trade-Zone-Block-Tests
- 7 Regime-Threshold-Tests
- 1 BERA-Replay-Test: synthetischer BEAR + Score 5.0 → reject (war früher BUY)

Aufwand: 12h

5. Phase D — T1-RISK-GUARD-1 (W1 Do-Fr → W2 Mo, 2 Tage)

Ziel

- **Hartes SL-Invariant:** "SL darf nach Entry nie schlechter werden" (außer kontrolliertes DCA-Rebuild mit Max-Loss-Check)
- **DCA-Restrictions:** kein blindes Nachkaufen
- **Volatility-Scaled Position-Sizing:** kleinere Positions bei höherer Volatilität

SL-Never-Worse-Invariant

```
def adjust_sl(pos: dict, new_sl: float, reason: str) -> bool:
    """
    Returns True if SL was adjusted, False if blocked by invariant.
    Long-Trades: new_sl MUST be >= pos['stop_loss'] (only tighten upward).
    Exception: explicit DCA-Rebuild path with assert_max_loss_under_cap().
    """
```

Wird in `position_manager.py` an jeder SL-Adjustment-Stelle eingehängt (Trailing, News-Adjustment, Breakeven, DCA-Rebuild). Falls Code je versucht, SL nach unten zu setzen → Block + Warning-Log + Telegram.

DCA-Guards

```
def dca_allowed(pos, current_price, regime, dca_history) -> Tuple[bool, str]:
    # 1. Regime-Check: BEAR → DCA disabled (BERA-Lesson)
    if regime == 'BEAR':
        return False, 'dca_blocked_in_bear'
    # 2. Reclaim-Requirement: nur DCA wenn Preis schon mal über altem SL+ATR war
    if not has_reclaimed_above(pos):
        return False, 'dca_blocked_no_reclaim'
    # 3. Max-Loss-Simulation: würde Nachkauf den Max-Loss-Cap durchbrechen?
    sim_max_loss = simulate_max_loss_after_dca(pos, current_price, dca_size)
    if sim_max_loss > MAX_LOSS_PER_TRADE_CAP:
        return False, 'dca_blocked_max_loss_breach'
    # 4. DCA-Anzahl-Cap (existing)
    if dca_history.count >= 2:
        return False, 'dca_blocked_count'
    return True, ''
```

Volatility-Scaled Position-Sizing

```
# Statt fixer 5% des Cash:
def position_size_volatility_scaled(cash, atr_pct, base_pct=0.05):
    """
    base_pct = 5% bei normaler Volatilität (atr_pct ≈ 2%)
    high vol (atr_pct = 5%): reduziert auf 0.05 * (2/5) = 2%
    low vol (atr_pct = 1%): erhöht auf 0.05 * (2/1) = 10%, capped auf 8%
    """
    target_atr = 0.02
    scale = target_atr / max(atr_pct, 0.005)
    return cash * min(base_pct * scale, 0.08)
```

Tests

- 5 SL-Invariant-Tests (Trailing/News/Breakeven/DCA/RebuildException)
- 6 DCA-Guard-Tests (BEAR-Block, Reclaim, MaxLoss, Count, OK, Edge)
- 4 Vol-Scaled-Sizing-Tests (high-vol / low-vol / normal / cap)
- 1 BERA-Replay-Test: BEAR + DCA-attempt → blocked mit reason

Aufwand: 12h

6. Phase E — T1-EXIT-OPTIMIZER-1 (W2 Mo-Mi, 2 Tage)

Ziel

Schnellere und bessere Gewinnsicherung. Heute fallen Altcoins oft nach kurzem Anlauf — T1 sichert zu spät.

Sub-Module

Sub-Modul	Heute	Soll
Break-Even	nach 60h Hold	nach +1R Profit (PnL = 1× Risk)
Partial Profit	nicht implementiert	50% sell @ +1R, 25% sell @ +2R , Rest läuft mit Trailing
Trailing-Start	sofort nach Entry	erst nach +0.8R Momentum bestätigt (sonst Whipsaw)
Time-Stop	96h Hard-Exit	bei Seitwärts (< +0.5R nach 4h) + 96h Hard-Exit als Backup

"Echtes Momentum" Definition

Für Trailing-Start:

```
def momentum_confirmed(pos, current_price, ohlcv_15m):
    # 1. PnL > 0.8R
    # 2. Letzte 3 von 4 Candles grün
    # 3. Volume in letzter Candle > 1.2x MA
    # 4. Preis über Entry+ATR
```

Tests

- 4 Break-Even-Tests (1R-trigger, no-trigger, edge)
- 6 Partial-Profit-Tests (50%@+1R, 25%@+2R, Reststellung, ordering)
- 5 Momentum-Trailing-Tests (kein false start, echtes Momentum, edge)
- 4 Time-Stop-Tests (Seitwärts-Trigger, Hard-Exit-Backup, time-window)

Aufwand: 12h

7. Phase F — T1-POST-TRADE-LEARNING (W2 Mi-Do, 1.5 Tage)

Ziel

Jeder geschlossene Trade wird automatisch klassifiziert. Daten landen in `trade_logs.metadata.json` und werden im Chart + GUI sichtbar.

Klassifikations-Schema

```
{
  "post_trade_analysis": {
    "entry_timing":    "early" | "good" | "late",
    "dca_assessment":  "helpful" | "harmful" | "neutral" | "no_dca",
    "sl_quality":      "appropriate" | "too_tight" | "too_loose",
    "tp_quality":      "appropriate" | "too_far" | "too_close",
    "mae_pct":         0.052,    // Max Adverse Excursion
    "mfe_pct":         0.103,    // Max Favorable Excursion
    "mae_mfe_ratio":   0.504,
    "regime_at_entry": "BULL",
    "regime_fit":       "good" | "mismatch",
    "exit_reason_optimal": true | false,
    "overall_grade":    "A" | "B" | "C" | "D" | "F"
  }
}
```

Berechnungslogik

- Liest OHLCV-Daten für Hold-Window aus existing OHLCV-Cache
- Berechnet MAE/MFE
- Vergleicht mit Entry/Exit-Levels
- Klassifiziert nach Heuristik-Regeln

Wiring

Hook im `main.py` post-close-Flow:

```
for closed_trade in closed_by_sl_tp:
    analysis = analyze_trade(closed_trade)
    closed_trade['metadata']['post_trade_analysis'] = analysis
    emit_trade(..., metadata=closed_trade['metadata'])
```

GUI

Bestehende TradeLogResource bekommt: - TextEntry für `overall_grade` Badge mit Color - ViewEntry für full `post_trade_analysis` JSON - Filter `overall_grade` (A-F) im Index

Tests

- 5 Klassifikations-Tests (Entry-Timing-Edge-Cases)
- 4 MAE/MFE-Berechnungs-Tests
- 3 Grade-Assignment-Tests
- 1 End-to-End: closed-trade → metadata enthält analysis

Aufwand: 10h

8. Phase G — EXEC-MODE-LABEL-3 Phase 3a (W2 Do-Fr, 0.5 Tag)

Ziel

Backward-Compatible Cleanup: - `--paper` Flag aus `bot_watchdog.sh` entfernen (heute schon dort entfernt) — ABER aus `main.py` CLI-Parser sauber raus - `PaperTrader` Class-Alias zu `BaseTrader` (neuer Name) - `paper_portfolio.json` STATE_FILE-Konstante via symlink zu `state_portfolio.json` - `PaperBalanceProvider` → `LocalBalanceProvider`

Alte Namen bleiben **parallel** als Alias, damit Tests grün bleiben. Phase 3b (Hard-Delete) kommt in einer späteren Woche nach 24h-Beobachtung.

Aufwand: 4h

9. Phase H — REFACTOR-VS-REWRITE + T2-PLAN PDFs (W2 Fr-Sa, 0.5 Tag)

Ziel

Zwei Plan-Dokumente: 1. **REFACTOR-VS-REWRITE-PLAN-PDF**: finale Architektur-Entscheidung mit Hybrid-Empfehlung (siehe heutige Diskussion) 2. **T2-SOLANA-SHADOW-1-PLAN-PDF**: T2-Greenfield-Plan für Woche 3+ (kein Code in dieser Phase)

Lieferung an files.gewerbespeicher-rechner.de.

Aufwand: 4h

10. Tagesplan-Tabelle (kalendarisch)

Tag	Phase	Hauptaktivität
W1 Mo (19.05.)	A + B start	TIER-ARCH morgens, SYMBOL-GATE Recon nachmittags
W1 Di (20.05.)	B → C	SYMBOL-GATE Cutover, QUALITY-SCORE Sub-Score-Module
W1 Mi (21.05.)	C	QUALITY-SCORE No-Trade-Zones + Regime-Thresholds
W1 Do (22.05.)	C → D	QUALITY-SCORE Cutover, RISK-GUARD SL-Invariant
W1 Fr (23.05.)	D	RISK-GUARD DCA-Guards + Vol-Scaled-Sizing
W1 Sa-So	Review	Operator-Beobachtung (3 Phasen live)
W2 Mo (26.05.)	D ende → E	RISK-GUARD Cutover, EXIT-OPTIMIZER Break-Even-Logic
W2 Di (27.05.)	E	EXIT-OPTIMIZER Partial Profit + Momentum-Trailing
W2 Mi (28.05.)	E → F	EXIT-OPTIMIZER Cutover, POST-TRADE-LEARNING Klassifikation
W2 Do (29.05.)	F → G	POST-TRADE-LEARNING Cutover + GUI, EXEC-MODE-LABEL-3 Phase 3a
W2 Fr (30.05.)	G → H	EXEC-MODE-3a Cutover, Refactor-vs-Rewrite-PDF
W2 Sa (31.05.)	H	T2-SHADOW-Plan-PDF
W2 So	Review	Endabnahme + Operator-Decision für Woche 3

11. Was NACH den 2 Wochen kommt (Woche 3+ Vorschau)

Nicht Teil dieser Roadmap, aber konzeptuell vorbereitet:

Phase	Aufwand	Inhalt
W3	3-4 Tage	T1-EXTERNAL-SIGNAL-CHANNELS-1 — Discord + Telegram-Channels in unified Tabelle
W3	3-4 Tage	T1-AI-SIGNAL-ANALYSIS-1 — LLM-basierte Auswertung jeder Channel-Nachricht (Claude/GPT/Gemini), source_type-agnostisch
W4	4-5 Wochen	T2-SOLANA-SHADOW-1 Implementation (Greenfield-Modul, dedicated wallet, Jupiter SDK, eigene Risk-Engine)
W4+	tbd	EXEC-MODE-LABEL-3 Phase 3b (Hard-Delete alter Paper-Namen)
W4+	tbd	STRATEGY-PROFILE-1 (Cash-Quote-Split T1/T2, wenn T2 läuft)

12. Risiko-Matrix

Risiko	Schwere	Mitigation
Quality-Filter zu strikt → 0 BUYS für Tage	mittel	BERA-Replay-Test + Live-Beobachtung 48h nach Cutover
SL-Invariant blockt legitime SL-Updates	niedrig	DCA-Rebuild-Path mit Max-Loss-Check als explizite Exception
Vol-Scaled-Sizing zu klein bei normaler Vol	niedrig	Floor bei 1% + Cap bei 8%
Partial-Profit verkauft zu früh	mittel	Backtest vor Cutover (nutzt OHLCV-Cache)
Time-Stop bei trending market vorzeitig	niedrig	4h-Trigger nur bei Seitwärts (< 0.5R PnL)
Post-Trade-Learning OHLCV-Fehler	niedrig	Fail-soft: kein analysis-Field statt Bot-Crash
EXEC-MODE-3a-Aliase brechen Test-Suite	mittel	Phase 3a hält alle Aliase parallel; full delete erst Phase 3b

13. Boundaries (durchgehend 2 Wochen)

- 0x Mainnet (Binance + Solana)
- 0x MULTI_STRATEGY_DRY_RUN=false
- 0x T2-Solana-Code
- 0x T3-Copy-Trading-Code
- 0x CommandBus-V6 / MH-1
- 0x Pump.fun-Logik in T1
- 0x ML-Modell ohne saubere Datenbasis

- 0x Strategie-Parameter-Änderung außerhalb der Plan-Modules
- 0x DB-Mass-Mutation
- 0x Push auf Remote
- 0x Secrets im Chat (Lesson Learned)
- 0x env dump

14. Stop-Gates pro Phase

Phase	Pre-Cutover	Post-Cutover (24h)
A TIER-ARCH	alle 7 Tests grün	0 Tracebacks
B SYMBOL-GATE	alle 9 Tests grün; reject-Logs erscheinen in decision_logs	min. 1 echter Reject mit <code>symbol_not_on_binance</code>
C QUALITY-SCORE	alle 22 Tests grün; BERA-Replay-Test pass	min. 5 Reject-Reasons mit Quality-Sub-Score < threshold
D RISK-GUARD	alle 15 Tests grün; SL-Invariant pass	kein SL-Downgrade in trade_logs.metadata
E EXIT-OPTIMIZER	alle 19 Tests grün; OHLCV-Backtest Sanity	min. 1 Partial-Profit-Event in trade_logs
F POST-TRADE	alle 13 Tests grün	analysis-Field in jedem neuen closed trade_logs
G EXEC-MODE-3a	alle bestehenden Tests grün (mit alten Aliasen)	0 Tracebacks
H PDFs	Operator-Review	—

15. Erwartetes Resultat nach 2 Wochen

Komponente	Stand
T1 Binance Testnet	Quality-First-Pipeline: Symbol-Gate + Quality-Floor + No-Trade-Zones + Regime-Thresholds + SL-Invariant + DCA-Guards + Vol-Scaled-Sizing + Exit-Optimizer + Post-Trade-Learning
Tier-Architektur	im Code-Vertrag, T1=active, T2=plan_only, T3=deactivated
PaperTrader-Legacy	Phase 3a parallel-Aliase aktiv; Phase 3b später
GUI	+ post_trade_analysis-Badge in TradeLog + Quality-Sub-Score in DecisionLog-metadata
Postgres	Schema unverändert; metadata_json wächst um post_trade_analysis-Feld
Operator-Workflows	klarer Quality-Audit-Trail pro Trade
Refactor-Plan-PDF	erstellt, Entscheidungsgrundlage für W3+
T2-Plan-PDF	erstellt, Implementation-Ready für W4+
T2-Code	bleibt 0 LoC (Plan-only)
Discord + AI	bleibt 0 LoC, geplant für W3

16. Operator-Decisions vor Start (kompakt)

#	Decision	Default
1	T2-Status <code>PLAN_ONLY</code> (Empfehlung) oder <code>DEACTIVATED</code> ?	PLAN_ONLY
2	Quality-Threshold global oder regime-individuell?	regime-individuell (Tabelle in Phase C)
3	DCA in BEAR komplett blockiert oder mit höheren Restriktionen?	komplett blockiert (BERA-Lesson)
4	Partial-Profit-Stufen 50%@+1R + 25%@+2R ok?	ja, ggf. anpassen wenn Operator andere Vorgabe
5	Post-Trade-Analysis-Grade (A-F) ok oder lieber numerisch?	A-F (operator-friendly, GUI-renderbar)

17. STOP

Plan v3. Keine Implementation. Operator-Entscheidung erforderlich für:

G0 TIER-ARCH-CONTRACT-1

als Startsignal für Woche 1.

Boundaries hart, Sicherheitsgates konservativ. Discord + AI bleiben ausdrücklich für Woche 3+ vorgesehen, NICHT in diesen 2 Wochen.