RECON-MH v3.2 — Final Review Data Extraction

Datum: 2026-05-10 **Status:** READ-ONLY · NO IMPLEMENTATION · NO MUTATION **Roadmap-Stand:** v3.2 (Master-Boundaries v3.2 + 4 Session-Audit-Trails) **Bot-Live-Stand:** PID 18185 · master `eda3042` · BINANCE_TESTNET=true · baseline_holdings.json absent · runtime_config.json absent **Boundaries:** keine Code-Änderung, keine Mutation, keine Migration, kein Deploy, kein Restart, kein Worker-Run, kein Mainnet, kein Push.

Executive Summary

Bereich	Bewertung
Architektur-Konsistenz	✅ stark — 21 G-DR + 12 G-SR + 18 MN-DR/SR durchgängig referenziert
Cross-References	✅ vollständig — alle 11 Phase-Files cross-verweisen sich + Master-Boundaries
Erweiterbarkeit	✅ gut — G-DR-16..21 (v3.2) verhindern hardcoded Kopplungen
Deploy-Readiness RECON-2.3-DEPLOY	⚠ **1 File-Drift identifiziert** (`command_worker.py` Host vs Container)
Frozen-only Drill Vorbereitung	✅ vollständig dokumentiert in `06_testnet_drill.md`
MH-1 Readiness	⚠ blockiert durch 12 offene Q-MH (asynchron klärbar) + keine Code-Phase begonnen
Mainnet-Trennung	✅ 5-Layer-Block + organisatorische Disziplin (MN-DR-1..5)

TEIL A · Datei-Inventar

A.1 · Roadmap-Verzeichnis (`docs/roadmap/recon_managed_holdings/`)

Phase-Files (12)

#	File	Lines	Zweck	Restart?	Migration?	Mainnet?	Risk
99	`99_master_boundaries.md`	472	**Zentraler Vertrag** — 21 G-DR + 12 G-SR + 18 Policies. Verbindlich für alle Phasen.	–	–	no	–
00	`00_overview.md`	254	Eintrittsstelle, State-Machine ASCII, Roadmap-Tabelle MH-0..9 + MH-0.5	no	no	no	low
01	`01_foundation.md`	497	Schemas baseline_holdings + managed_state + risk_proposals; Reader/Writer-API; Hashing	no	yes (MH-4)	no	low (Reader), medium (Writer/Migration)
02	`02_risk_proposal_engine.md`	368	proposal_engine V1-min-Spec + 3-Varianten-Logik + SM-4/5/7 + V2-Roadmap	no	no	no	medium
03	`03_state_machine.md`	403	10 States + 18 Transitions + Drift-Kategorien (6 Stufen) + Persisted-vs-Derived + Sub-State-Pattern	yes (MH-7 Wiring)	no	no	high
04	`04_gui_operator_flow.md`	416	Multi-Step-Wizard + 5 UX-Regeln + Audit-Timeline First-Class + Emergency-Action 3 Modi	no	no	no	medium
05	`05_commandbus_worker.md`	425	8 Command-Types + Worker-Handler + Audit-Snapshots + Two-File-Atomic + JSON-SoT	yes (MH-7)	yes (MH-4)	no	high
06	`06_testnet_drill.md`	349	Drill-Plan 4 Sub-Phasen (Frozen-only / Single-Asset Promote / Drift-Sim / Cleanup)	yes (MH-8)	no	no	high
07	`07_mainnet_future.md`	307	BACKLOG/BLOCKIERT · 5-Layer-Block · MN-SR-1..12 + neue MN-DR-1..5 organisatorisch	yes	tbd	**JA (BACKLOG)**	kritisch
08	`08_open_questions.md`	475	Q-MH-1..18 — 6 decided, 12 open, alle mit Defaults	no	no	no	low (Doc), high wenn falsche Antworten
09	`09_test_strategy.md`	387	~280 Tests, 8 Test-Kategorien, Boundaries pro Phase	no	no	no	low
10	`10_backlog_future_extensions.md`	525	14 Backlog-Themen + 4 v3.1-Themen (managed_active Sub-States / Proposal Replay / Quarantine / Risk-Explainability)	tbd	tbd	tbd	–

Total: 12 Files, 4.853 Zeilen.

Session-Audit-Trails (4)

File	Lines	Datum	Inhalt
`sessions/q_mh_session_2026-05-10_architecture_priming.md`	376	2026-05-10	Q-MH-Session-Vorlage mit 6 Architektur-prägenden Fragen + Default-Empfehlungen
`sessions/external_review_2026-05-10.md`	135	2026-05-10	External Review v2 (A1..A9) Mapping + Lessons-Learned
`sessions/external_review_v3_2026-05-10.md`	108	2026-05-10	External Review v3 (R1..R5 + B1..B4) Mapping + Reviewer-Verbote
`sessions/external_review_v4_2026-05-10.md`	117	2026-05-10	External Review v4 (S1..S5 + A..D) Mapping + Strategische-Botschaft

Total: 4 Sessions, 736 Zeilen.

Roadmap-Subdirectory Total: 16 Files, 5.589 Zeilen, ~230 KB

A.2 · Memory-Pins (in `~/.claude/projects/.../memory/`)

Pin-File	Zweck
`recon_mh_sub_roadmap_pin.md` (15.9 KB)	**Master-Pin** — komplette v3.2-Übersicht mit Patch-Historie
`recon_status_pin.md` (18.7 KB)	Globaler RECON-Pin (RECON-2.1..2.3c + RECON-MH-Verweis)
`recon_2_1_status.md`	RECON-2.1 Schema-Scaffolding
`recon_2_2a_status.md`	RECON-2.2a Bot-Awareness wiring
`recon_2_2b_status.md`	RECON-2.2b Restart-Drill
`recon_2_3_status.md`	RECON-2.3 PHP-Side
`recon_2_3c_status.md`	RECON-2.3c Bot-Worker handlers
`designer_iter_3_status.md`	Designer-Iteration v3 closed

A.3 · Cross-References

Externe Roadmap-Bezüge

Bezug	Pfad
Original RECON-Architektur	`docs/roadmap/mainnet_preflight_recon.md` (572 Zeilen)
Monolithisches MH-Quell-Doc	`docs/roadmap/recon_managed_holdings_architecture.md` (574 Zeilen) — bleibt als Synthesis-Referenz
T-SPLIT-Roadmap	`docs/roadmap/t_split_roadmap.md`
GUI-Design-Briefings	`docs/gui/iteration_2_briefing.md` + `docs/gui/iteration_3_briefing.md`
GUI-Iter-3-Archive	`docs/gui/iter3/` (564 KB, 21 Files)

Memory-Cross-References (zentral)

Memory-File	Inhalt
`feedback_backup_before_live_actions.md`	DR-9 durable rule (Backup-Pflicht)
`feedback_testnet_permanent_paper_rename.md`	TESTNET permanent durable rule
`t_split_status.md`	t3_copy_trading-Block durable für proposal_engine
`cap_1_preflight_backlog.md`	CAP-1 Integration BACKLOG
`g10_runtime_config_operator_runbook.md`	G10-Apply-Pattern als Vorbild
`b_fee_fix_3_status.md`	B-FEE-FIX Mainnet-Blocker durable rule

TEIL B · G-DR / G-SR Audit

B.1 · 21 Globale Durable Rules

ID	Inhalt (gekürzt)	Quelle	Bewertung
G-DR-1	Frozen-by-default für unbekannte Assets	RECON-2.1 + DR-2	✅ kritisch, sauber verankert
G-DR-2	Operator final authority bei state-transitions	DR-11	✅ kritisch
G-DR-3	CommandBus-only für File-Mutationen	DR-1 erweitert	✅ kritisch, source-grep gepinnt
G-DR-4	TESTNET-first dauerhaft	DR-3	✅ kritisch
G-DR-5	Audit-heavy (audit_event + Snapshot pro State-Transition)	DR-6 + DR-13	✅ stark
G-DR-6	Backup-before-mutate (sha256 + Backup-File)	DR-5 + DR-9	✅ stark
G-DR-7	Multi-Step-Wizard für Promote (≥5 Steps)	UX-2	✅ verbindlich
G-DR-8	`'managed'` NIE als default_policy_for_unlisted	DR-2	✅ kritisch
G-DR-9	Bot autonom managed→frozen verboten	DR-11	✅ kritisch
G-DR-10	Risk-Proposals versioniert (proposal_version + risk_model_version)	DR-12	✅ stark
G-DR-11	Audit-Prefix-Separation (managed. / *baseline.* / runtime_config.*)	DR-13	✅ source-grep gepinnt
G-DR-12	wallet-signature excludes tradable_quote	DR-7	✅ stark, RECON-2.2a-implementiert
G-DR-13	Watchdog-clawbot-Konvention	DR-8	✅ kritisch (RECON-2.2b Befund)
G-DR-14	Two-File-Atomic-Pattern für Policy-Wechsel	Q-MH-Session 2026-05-10 DR-7	✅ stark, neu in v2
G-DR-15	synthetic_entry NIE als echte Cost-Basis · 4 Pflicht-Metadaten-Felder	External Review v2 A1 + v3 R3	✅ stark, Mainnet-relevant
G-DR-16	Policy-Resolver-Pattern statt `if asset == "BTC"`	External Review v4 S1	✅ neu in v3.2, future-safe
G-DR-17	Bot↔GUI-Boundary (Bot kennt nur Commands/Policies/State-Files/Events)	External Review v4 S2	✅ neu in v3.2, kritisch
G-DR-18	Event-Versionierung (`event_version` Pflicht)	External Review v4 A	✅ neu in v3.2, ML-Vorbereitung
G-DR-19	Feature-Flag-Service zentralisiert	External Review v4 B	✅ neu in v3.2
G-DR-20	Canonical Asset-Identity (`BTC ≠ BTCUSDT`)	External Review v4 C	✅ neu in v3.2, Mainnet-kritisch
G-DR-21	Exposure-Provider-Pattern statt ad-hoc-Berechnungen	External Review v4 D	✅ neu in v3.2, CAP-1-/T-SPLIT-Vorbereitung

Bewertung: - alle 21 G-DR sind **eindeutig formuliert + cross-referenziert** - keine sichtbar redundanten Regeln - G-DR-16..21 (v3.2) sind Constraints für **zukünftige Komponenten** — gut, dass früh verankert - mögliche **Lücke**: G-DR-22 für Bot-side Operator-Notification-Channel-Disziplin (Telegram-only? mehr?) — aktuell nicht erfasst, könnte später nötig werden

B.2 · 12 Globale Stop-Regeln

ID	Trigger	Status
G-SR-1	außerhalb-CommandBus-Mutation → STOP	✅
G-SR-2	Proposal ohne `risk_model_version` → STOP	✅
G-SR-3	Promote ohne user-actor-Audit → STOP	✅ DR-11 enforcement
G-SR-4	Bot autonom managed→frozen → STOP	✅ DR-11
G-SR-5	command_type_registry-Version-Drift → STOP	✅ G6.5 invariant
G-SR-6	Bot-PID-Wechsel ohne dokumentierten Phasen-Trigger → STOP	✅
G-SR-7	Drift-Detection autonomous-Sell → STOP	✅ Bot ist Berater bei Drift
G-SR-8	Watchdog steve-tradingbot statt clawbot → STOP	✅ RECON-2.2b Befund
G-SR-9	Confidence < threshold ohne Override → STOP	✅ SM-4
G-SR-10	Volatility-Kill → autonomous pause (nicht sell)	✅ SM-5
G-SR-11	Mainnet-Apply-Versuch → STOP (5-Layer halten)	✅ kritisch
G-SR-12	confidence_score Feld fehlt in Proposal → STOP	✅

B.3 · 18 MN-DR / MN-SR (Mainnet-spezifisch)

In `07_mainnet_future.md` : - **MN-SR-1..12:** technische Mainnet-Sicherheitsregeln (aggressive disabled, DCA disabled, t2-fallback, Cooldown 7d, Max-Alloc 5%, Confidence 0.65, Volatility 15%, etc.) - **MN-DR-1..5:** organisatorische Disziplin (separate Branch / Pipeline / .env / Secrets / Audit-Logs / PDF-Preflight + Hotfix-Verbote + Bypass-Verbote)

Mainnet-Block ist 5-Layer-redundant (Settings + Writer + Worker + Validator + GUI).

B.4 · Bewertung G-DR-16..21 spezifisch

ID	Sauber genug?	Zu allgemein?	Zu eng?	Zukünftige Probleme?	Erweiterungspunkte?
G-DR-16 (Policy-Resolver)	✅	nein, klar abgegrenzt durch Source-Grep-Test	nein	Test muss alle Asset-Symbol-String-Vergleiche erfassen — vorsichtig formulieren	`policy_resolver` Service-Layer in MH-1 implementieren
G-DR-17 (Bot↔GUI-Boundary)	✅	nein, klar Listet 4 erlaubte Concepts	nein	Telegram-Reporter ist Bot-Side aber kommuniziert mit external — könnte als „GUI" fehlinterpretiert werden	**Empfehlung:** in G-DR-17 Klarstellen: „GUI" = Filament/Browser-Layer, nicht alle externen Kanäle
G-DR-18 (Event-Versionierung)	✅	nein	nein	Migration `_v1` → `_v2` Reader-Branches müssen pro Event-Type definiert sein	dual-compute-Pattern aus G-DR-16-Section §16 reused
G-DR-19 (Feature-Flags)	✅	nein	nein	env-spezifische Flags (testnet vs mainnet) müssen klar definiert sein	`FeatureFlags.is_enabled(name, environment=...)`
G-DR-20 (Canonical Asset-Identity)	✅	nein	nein	wrapped Assets vs original (WBTC vs BTC) — semantische Frage was canonical heißt	`asset_identity.py` Mapping-Tabelle in MH-1 oder eigener kleiner Phase
G-DR-21 (Exposure-Provider)	✅	nein	nein	Performance bei vielen Positionen — Provider muss caching haben	CAP-1-Integration verbindlich über Provider

Insgesamt: alle 6 neuen G-DR sind **sauber, future-safe, kein Redundanz-Risiko**.

TEIL C · Schnittstellen / Extensibility Review

C.1 · Komponenten-Audit

Komponente	Abstrahiert?	Hardcoded?	Zukünftige Probleme	Empfehlung
exposure_provider	⚠ noch nicht implementiert	nein (G-DR-21 verbietet Hardcode)	Performance-Cache nötig bei vielen Positionen	in MH-1 oder als 1-File-Phase vor MH-3
policy_resolver	⚠ noch nicht implementiert	nein (G-DR-16)	Source-Grep-Test muss alle String-Vergleiche erfassen	in MH-1 zwingend
strategy_group	✅ T-SPLIT-3 etabliert (`t1_core/t2_pump_dump/t3_copy_trading/legacy_unknown`)	nein	t3 ist „planned · disabled", proposal_engine darf t3 NICHT ausgeben	G-SR-14 + Validator
managed_state	✅ schema-Skizze in `01_foundation §2` , Reader-API geplant	nein	Sub-State-Erweiterung als `{state, substate}` (G-DR-… S3-Pattern)	bereits dokumentiert in `03_state §9.4`
baseline_holdings	✅ implementiert (RECON-2.1/2.3)	nein	wallet_signature-Drift bei Policy-Wechsel	gelöst durch G-DR-14 (Two-File-Atomic)
commandbus	✅ G6.5 etabliert + erweitert für MH	nein (CommandTypeRegistry zentral)	Worker-Daemon-Aktivierung in MH-0.5	bereits geplant
audit_snapshots	⚠ noch nicht implementiert	nein (immutable per G-DR-5)	File-System-Bloat über Zeit (90-Tage-Cleanup-Job)	in MH-6 zwingend, Cleanup-Phase Backlog
proposal_engine	⚠ noch nicht implementiert	nein (G-DR-10 + G-DR-21)	V1-min-Scope einhalten (External Review v2 A3)	Scope-Lock in `02_engine §0`
wallet_signature	✅ implementiert (RECON-2.2a + DR-7-Resolution)	nein	Algorithmus-Migration über `_v1` / `_v2` (G-DR-18)	bereits geplant

C.2 · Multi-Exchange-Probleme (zukünftig)

Problem	Risiko	Mitigation in v3.2
BTC vs BTCUSDT in symbol-Mapping	hoch	G-DR-20 Canonical Asset-Identity + `asset_identity.py`
ccxt-API-Differenzen Coinbase vs Binance	hoch	Backlog `Exchange-Abstraction` (10_backlog §5)
signature-Format unterschiedlich pro Exchange	mittel	wallet_signature.exchange-Feld bereits im Schema (`baseline_holdings.json§2`)
Order-API-Differenzen	mittel	LiveTrader.execute_buy bereits exchange-getrennt (binance vs binance_testnet)

C.3 · Event-Versionierung (G-DR-18) — wo fehlt sie aktuell?

Stelle	Status
`audit_events.metadata` (DB)	⚠ kein `event_version` -Feld in v3.2 dokumentiert; **Empfehlung:** in MH-1 oder MH-4 ergänzen
`commands.payload_json`	⚠ kein `event_version` ; **Empfehlung:** Phase MH-1
`risk_proposals/<id>.json`	✅ `proposal_version` + `risk_model_version` (G-DR-10)
`managed_state.json`	✅ `_meta.schema_version`
`audit_snapshots/<id>.json`	⚠ noch nicht spezifiziert; **Empfehlung:** in MH-6 ergänzen mit `snapshot_version`
`baseline_holdings.json`	✅ `_meta.schema_version` (RECON-2.1)

→ G-DR-18 ist **konsequent durchzuziehen** in MH-1/4/6.

C.4 · Policies hardcoded?

Policy-Bereich	Aktuell	Soll (G-DR-16)
Frozen-Check	bereits in `BaselineHoldingsReader.frozen_assets()`	✅ über Reader
Strategy-Group-Allowlist	proposal_engine Phase MH-3	✅ über Validator + Allowlist
Drift-Schwellen	aktuell hardcoded in §3.1 (`5%/2-ATR/BEAR↔BULL`)	⚠ später aus Settings/ `runtime_config.json` ziehen, nicht hardcoded im Bot-Code
Cooldown-Dauer	Q-MH-12 noch open (default 24h/7d)	wird aus `feature_flags` oder `runtime_config` gelesen werden

TEIL D · Deploy-Readiness Review

D.1 · Live-Stand 2026-05-10

Indikator	Wert
Bot-PID	**18185** (seit RECON-2.2b 2026-05-09 22:30 UTC)
Container	`clawbot` (Up)
master HEAD	`eda3042`
Bot-Cmdline	`python3 main.py --paper`
.env mtime	`1778324885`
BINANCE_TESTNET	`true` ✓
baseline_holdings.json	**ABSENT** ✓
runtime_config.json	**ABSENT** ✓

D.2 · Container vs Host File-Hash-Audit (9 Files)

File	Host-Hash	Container-Hash	Status
`baseline_bootstrap.py`	`6dded3ae…`	`6dded3ae…`	✅ MATCH
`baseline_holdings_writer.py`	`baca5cf3…`	`baca5cf3…`	✅ MATCH
`baseline_holdings_reader.py`	`3861d43a…`	`3861d43a…`	✅ MATCH
`execution/balance_provider.py`	`473a58c9…`	`473a58c9…`	✅ MATCH
`execution/live_trade.py`	`3855596e…`	`3855596e…`	✅ MATCH
`execution/paper_trade.py`	`223cfe71…`	`223cfe71…`	✅ MATCH
`main.py`	`d92a57dc…`	`d92a57dc…`	✅ MATCH
`scanner/universe.py`	`09020495…`	`09020495…`	✅ MATCH
`command_worker.py`	**`ecc8a41d…`**	**`17b8ca32…`**	⚠ **DIFF** (RECON-2.3-DEPLOY pending)

→ **8 von 9** Files synchron; **1 File divergent** — exakt der Drift, den RECON-2.3-DEPLOY auflöst.

D.3 · Deploy-Risiken

Risiko	Severity	Mitigation
`command_worker.py` divergent	mittel — Worker läuft heute manuell `--once`, alter Code-Stand wird beim nächsten Aufruf geladen	RECON-2.3-DEPLOY: docker cp + sha256-verify
Watchdog-Konvention drift	low — wurde in RECON-2.2b auf `clawbot` korrigiert	Pre-Deploy Watchdog-Dry-Run-Check (G-DR-13)
Backup-Verlust beim Deploy	low (durable rule G-DR-9)	pg_dump + state/ + .env vor Deploy verbindlich
Bot-Restart unbeabsichtigt	low — Worker-Code-Drop berührt Bot-Prozess nicht	main.py importiert command_worker NICHT — kein Restart-Trigger

D.4 · Pre-Deploy Pflicht-Checks (verbindlich für RECON-2.3-DEPLOY)

```
[ ] Backup pg_dump GUI-DB → /srv/shares/backups/recon_2_3_deploy_pre/
[ ] Backup live_portfolio.json
[ ] Backup .env (read-only kopiert)
[ ] Backup state/ snapshot
[ ] Health-Snap (Bot-PID + container-status + key-Files)
[ ] Memory tar.gz Snapshot
[ ] Pre-cp Hash-Vergleich command_worker.py Host vs Container dokumentiert
[ ] Watchdog-Skript clawbot-Konvention verifiziert (G-DR-13)
[ ] Settings.BINANCE_TESTNET=true verifiziert
[ ] runtime_config.json absent verifiziert
[ ] baseline_holdings.json absent verifiziert
```

D.5 · Post-Deploy Pflicht-Checks

```
[ ] Post-cp sha256 command_worker.py == Host
[ ] Bot-PID 18185 unchanged (kein implizit Restart)
[ ] Bot-Stdout: keine neuen Tracebacks
[ ] managed.* Audit-Events: 0 (managed-Lifecycle-Phase noch nicht aktiv)
[ ] python -c "import command_worker" im Container OK (Import-Sanity)
[ ] command_worker.py --help (oder ähnliches) zeigt erweiterte Command-Types

OPTIONAL (nicht zu RECON-2.3-DEPLOY gehörig):
[ ] worker --once (= RECON-2.4 Frozen-only Drill, NICHT in 2.3-DEPLOY)
```

D.6 · Hot-Reload vs Restart-Bound

Komponente	Hot-Reload?	Restart-Bound?	Begründung
`command_worker.py` Code	✅ — beim nächsten `--once` neu geladen	nein	Worker ist short-lived per design
`baseline_holdings.json` Reader	✅ — Reader.snapshot() liest pro Cycle frisch	nein	RECON-2.2a Pattern
`managed_state.json` Reader	✅ — analog (sobald implementiert)	nein	gleiche Architektur
`main.py` (Bot-Hauptloop)	❌	**JA**	nur in MH-7 erforderlich
`baseline_bootstrap.py`	❌	**JA**	läuft nur bei Bot-Start (managed-auto-import + signature-check)
`paper_trade.py` / `live_trade.py`	❌	**JA**	Klassen werden bei Bot-Start instanziert (`paper_trader = LiveTrader()`)
`scanner/universe.py`	❌	**JA**	Funktion wird in main.py importiert
Settings (`.env`)	⚠ teils — `runtime_config.json` ist hot-reload via ActiveConfigProvider; `BINANCE_TESTNET` etc. sind restart-bound	teils	G10-4.1 Pattern

D.7 · Dormant-Pfade (solange baseline_holdings.json absent)

Pfad	Verhalten
`baseline_bootstrap.bootstrap_baseline()`	wallet_signature wird ALWAYS geloggt, baseline_present=False, kein auto_import
`universe.get_tradeable_universe(baseline_reader=)`	filter ist no-op (frozen_set leer)
`paper_trade._is_base_baseline_frozen`	returns False für jeden Symbol
`live_trade.execute_buy` Guard	greift nicht
`balance_provider._baseline_frozen_subtract`	no-op

→ **100% Legacy-Verhalten** solange JSON absent. RECON-2.3-DEPLOY ändert das **nicht** — File bleibt absent.

TEIL E · State-Machine Review

E.1 · 10 States (5 persisted + 5 derived, External Review v3 R2)

Persisted States (5)

State	In `managed_state.json`	Persistierungs-Grund
`frozen`	implizit (kein Eintrag)	Default
`proposal_pending`	ja	UI zeigt Spinner; TTL-Job liest
`risk_proposed`	ja	TTL-Job liest
`managed_active`	ja	Bot-Decision-Cycle liest
`managed_paused`	ja	Bot stoppt Trades
`release_pending`	ja	Bot wartet auf Position-Close

Derived States (5)

State	Berechnet aus	Anzeige
`proposal_aborted`	letztes audit_event = `managed.asset_proposal_aborted`	UI-Badge
`proposal_rejected`	persisted= `frozen` + audit_event	UI-Badge
`managed_drift_alert`	persisted= `managed_active` + Drift-Detection	UI-Banner red
`exit_executed`	persisted= `frozen` + audit_event	UI-Badge
`cooldown_active`	now - last_release_ts < cooldown_threshold	UI-Badge

E.2 · Übergangs-Matrix (18 Transitions)

aus `03_state_machine §2` — vollständig dokumentiert. Auszug der kritischsten:

From	To	Actor	Guard
frozen → proposal_pending	user	wallet qty > 0 + Cooldown abgelaufen	
risk_proposed → managed_active	user	Hard-Confirm `<asset>:<variant>` + Confidence-Override-Flag falls < threshold	
risk_proposed → proposal_rejected (TTL)	system	proposal.expires_at < now	
managed_active → managed_drift_alert	**bot**	qty/price/regime drift threshold	
managed_*/managed_active → frozen	**NIE bot** (G-DR-9)	nur user oder system (TTL/release)	

E.3 · Drift-Kategorien (6 Stufen, External Review v3 R1)

#	Drift-Typ	Schwelle	Reaktion
D1	Rundungsdrift	<0.5%	Warning-Audit, kein State-Change
D2	Externer Transfer	5%+	Freeze (managed_drift_alert)
D3	Massive Qty-Abweichung	>50%	Hard-Stop + Telegram
D4	Asset fehlt	qty_real==0	Pause
D5	Neues unbekanntes Asset	nicht in baseline	Quarantine-Hint
D6	Wallet-Hash-Mismatch	beim Bot-Restart	Refuse to start (sys.exit(1))

E.4 · Identifizierte Risiken (von vorigen Reviews + Re-Audit)

Risiko	Mitigation in v3.2	Restrisiko
R1 (Operator approve + Bot drift Race)	row-level lock + idempotency_key	✅ low
R2 (Wallet-Balance ändert sich während Engine-Run)	Engine refetcht balance pre-write	✅ low
R3 (Zwei Admins approven gleichzeitig)	idempotency_key on proposal_id	✅ low
R4 (Bot-Cycle vs Pause-Command-Race)	**Worker-Daemon required** (Q-MH-14 decided MH-0.5)	✅ gelöst sobald MH-0.5 läuft
R5 (Worker --once Latency vs Bot-Cycle)	identisch zu R4	✅ gelöst durch MH-0.5
R6 (TTL-Expiry vs Operator approve)	row-lock	✅ low
Deadlock-Risiko	keiner identifiziert (alle State-Transitions sind asynchron)	✅ none
Drift-Risiko	in 6 Kategorien aufgeteilt	✅ verbessert
Operator-vs-Bot-Konflikt C1 (Override SL)	managed_state speichert HIST-Wert; state['positions'] LIVE	✅
C2 (Bot SELL vs Operator Pause)	Worker-Daemon (MH-0.5)	✅ ab MH-0.5
C3 (Operator setzt managed→frozen via Apply)	G-DR-14 Two-File-Atomic + DR-11 enforcement	✅

E.5 · Zukünftige State-Probleme

Problem	Mitigation
managed_active wird zu generisch	G-DR-…(S3-Pattern): `{state, substate}` statt `state_v2` (BACKLOG R2)
Quarantine-Lifecycle nicht definiert	BACKLOG B2 in `10_backlog §17`
Sub-State-Übergänge zwischen tracking_only/trading/dca/exit_only/reduce_only	später in eigener Phase, nicht jetzt

TEIL F · GUI / UX Review

F.1 · Wizard (5 Steps, UX-2)

Step	Inhalt	Pflicht?
1	Asset & Intent	ja, Begründung ≥20 chars
2	Bot-Proposal Review	ja, Bot fertig
3	Variant Selection + Override	env-abhängig (Testnet 3, Mainnet 2) + Synthetic-Entry-Marker
4	Risk-Summary + Exposure-Impact	UX-3 + UX-4
5	Hard-Confirm `<asset>:<variant>`	ja, Override-Pfad: `<asset>:override:<variant>`

Bewertung: Wizard ist **ausreichend differenziert** für Operator-Mental-Model.

F.2 · Hard-Confirm-Patterns (4 verschiedene)

Action	Pattern
Apply Profile (G10-6)	`$record->name` (case-sensitive)
Clear Runtime Config (G10-6)	`$record->name`
Apply Baseline (RECON-2.3)	`<count>:<sha8>` z.B. `47:b25e09fb`
Approve Managed Proposal (RECON-MH)	`<asset>:<variant>` z.B. `SOL:recommended`
Override-Pfad	`<asset>:override:<variant>`
Clear Baseline	statisch `clear-baseline`
Emergency Action 3 Modi	`FREEZE-ALL-MANAGED` / `EXIT-ONLY-MANAGED` / `REDUCE-ONLY-MANAGED`

Bewertung: alle Patterns dokumentiert, **keine Kollision identifiziert**.

F.3 · Mögliche UX-Fallen

UX-Falle	Mitigation in v3.2	Restrisiko
Operator klickt durch Wizard ohne zu lesen	Multi-Step-Wizard ≥5 Steps + Hard-Confirm dynamisch	✅ low
Confidence-Score wird ignoriert	Hard-Gate bei Score < threshold (SM-4) → Override-Confirm zwingend	✅
Synthetic-Entry wird als echte Cost-Basis interpretiert	UI-Marker `synthetic/estimated/imported/unresolved` (G-DR-15)	✅
Mainnet-Verwechselung	sticky Mainnet-Banner permanent + permanent rote Color-Coding	✅
Drift-Warning übersehen	Banner-red + sticky in Page-Header	✅
Quarantine-UX	nicht implementiert (BACKLOG)	⚠
3 Emergency-Modi werden verwechselt	jeder Modus eigener Hard-Confirm-String	✅

F.4 · Audit-Timeline First-Class (External Review v2 A8)

- in v3 explizit zur Hauptkomponente befördert
- Filter / Pagination / Export / JSON-Diff (Detail-Route) dokumentiert
- 30s Polling
- color-coded prefix (managed. */ baseline.* / runtime_config. */ risk_settings.*)

Bewertung: ✅ ausreichend dokumentiert.

F.5 · Mainnet-Warnungen + Drift-Warnungen

Element	Mainnet	Testnet
Mainnet-Banner sticky top	red + permanent	red statisch
Variant-Selector	aggressive ausgeblendet (Q-MH-13)	alle 3 sichtbar
Confidence-Threshold	0.65	0.50
Volatility-Kill-Threshold	15%	25%

Bewertung: ✅ konsequent.

F.6 · Default-Buttons-Risiko

Default-Button	Gefährlich?	Mitigation
Wizard „Next" Step 1→2	nein	nur Trigger der Bot-Analyse
Wizard Submit Step 5	wäre gefährlich	Hard-Confirm-String dynamisch verhindert Auto-Submit
Pause-Button	ja, einfach	Hard-Confirm `pause-<asset>`
Release-Button	ja, weil destruktiv	Hard-Confirm `release-<asset>` + Q-MH-4 Behavior-Choice
Emergency-Action-Button	sehr gefährlich	3 verschiedene Match-Strings + sticky-rot

TEIL G · Test-Strategie Review

G.1 · Geschätzte Tests pro Phase (aus `09_test_strategy.md`)

Phase	Bot-Tests	PHP-Tests	Boundary	Gesamt
MH-1 (Registry)	0	~25	5	~30
MH-2 (Reader)	~25	0	5	~30
MH-3 (proposal_engine)	~30	0	5	~35
MH-4 (Services + Migration)	0	~30	5	~35
MH-5 (Filament UI)	0	~25	5	~30
MH-6 (Worker-Handler)	~50	0	5	~55
MH-7 (Bot-Wiring)	~25	0	5	~30
MH-8 (Drill)	0	0	0	manual
MH-9 (Daemon)	~10	~10	5	~25
Total	**~165**	**~115**	**~40**	**~280**

G.2 · Boundary-AST-Test-Coverage (Soll vs Ist)

Test-Bereich	Soll	Ist (in v3.2 spezifiziert)
`proposal_engine` ohne commands-INSERT / order-API	✅	dokumentiert
`managed_state_writer` ohne ccxt	✅	dokumentiert
`managed_state_reader` ohne file-mutation	✅	dokumentiert
Worker-Handler audit-Emit Anfang+Ende	✅	dokumentiert
G-DR-14 Boundary (tradable_quote-Crossings reject)	✅	dokumentiert
Two-File-Atomic Tests	✅	dokumentiert
JSON-SoT (kein DB-Read aus Bot)	✅	dokumentiert
G-DR-16 Policy-Resolver source-grep	✅ neu in v3.2	
G-DR-17 Bot-GUI-Boundary source-grep	✅ neu in v3.2	
G-DR-18 Event-Versionierung Test	⚠ erwähnt aber nicht detailliert	
G-DR-19 Feature-Flags zentralisiert source-grep	⚠ erwähnt aber nicht detailliert	
G-DR-20 Asset-Identity Test	⚠ noch nicht spezifiziert	
G-DR-21 Exposure-Provider source-grep	⚠ noch nicht spezifiziert	

→ **Empfehlung:** Test-Plan-Erweiterung für G-DR-18..21 in einem v3.3-Patch (nach RECON-2.3-DEPLOY).

G.3 · Restart-Tests

- post-MH-7 Restart-Test: bootstrap importiert nur managed_active, NICHT pause/drift/exit
- wallet_signature mismatch → sys.exit(1)
- post-Restart kein Behaviour-Drift (Cycle weiter wie vorher)

G.4 · Drift-Tests

- 6 Drift-Kategorien à mindestens 2 Tests (positive + negative-Schwelle) = 12 Tests in MH-7

G.5 · Replay-Tests

⚠ noch nicht spezifiziert — gehört zur Backlog-Phase **Proposal Replay System** (B1).

G.6 · Rollback-Tests

- Two-File-Atomic-Rollback: einer der Files schlägt fehl → Backup-Restore beider + audit
- managed_state restore from snapshot (BACKLOG)
- baseline_holdings clear → Bot fällt auf Legacy zurück (RECON-2.3 implementiert)

G.7 · GUI-Tests

Bereich	Tests in MH-5
ManagedHoldings-Page admin-only	5
Wizard skippable=false	5
Hard-Confirm dynamic validation	5
Polling/Stale-Threshold	5
Audit-Trail-Read	5
Boundary-AST	5

G.8 · Worker-Tests

- pro Handler ~6-7 Tests in MH-6 (8 × 6 = 48 Tests)
- Two-File-Atomic-Test pro Handler der beide Files schreibt
- DR-11 enforcement: Bot-actor wird abgelehnt für managed→frozen

G.9 · Bewertung

- **realistisch?** ja — ~280 Tests sind angemessen für Risk-Profile
- **ausreichend?** medium — G-DR-18..21 brauchen Test-Plan-Erweiterung
- **unterdimensioniert?** Replay-Tests + Quarantine-Tests fehlen — beides BACKLOG
- **gefährliche Lücken?** keine kritischen — die Lücken sind alle in BACKLOG-Themen, die explicit Backlog sind

TEIL H · Offene Entscheidungen

H.1 · 12 offene Q-MH-Fragen

Q-MH	Frage	Default	Priorität
Q-MH-1	Proposal-TTL	7 Tage	mittel
Q-MH-3	Operator-Override-Tiefe	medium (SL/TP/max_alloc/trailing/dca)	mittel
Q-MH-4	release_pending Behavior	choice (default graceful)	mittel
Q-MH-5	Multi-Asset-Bulk-Proposal	C (Bulk-Markierung mit Einzel-Workflows)	niedrig
Q-MH-6	Bot self-initiated Proposals	B (Bot darf alerten, Operator-Confirm)	niedrig
Q-MH-7	Drift-Schwellen	qty>5%, price>2-ATR, regime BEAR↔BULL	mittel
Q-MH-8	Pause-Verhalten	A (SL/TP bleibt aktiv)	mittel
Q-MH-9	Re-Engage nach exit_executed	A (frozen, manuell propose)	niedrig
Q-MH-10	synthetic_entry-Method	D (Operator-choice, default current_price)	mittel
Q-MH-12	Cool-down nach reject/release	D (24h Testnet, 7d Mainnet)	mittel
Q-MH-16	Cooldown-Override durch Operator	B (mit extra Hard-Confirm)	niedrig
Q-MH-17	Confidence-Threshold-Wert	0.50 Testnet, 0.65 Mainnet	mittel
Q-MH-18	Volatility-Kill-Schwelle	25% Testnet, 15% Mainnet	mittel

H.2 · Offene technische Entscheidungen

Bereich	Status
`policy_resolver.py` Implementation	offen (in MH-1)
`asset_identity.py` Mapping-Tabelle	offen (in MH-1)
`feature_flags.py` Service	offen (in MH-1)
`exposure_provider.py` Service	offen (vor MH-3)
Event-Versionierung pro Event-Typ	offen (in MH-1)
audit_snapshots Cleanup-Job	offen (BACKLOG, post-MH-9)

H.3 · Offene organisatorische Entscheidungen

Bereich	Status	Mainnet-relevant?
2-Personen-Sign-Off Process	offen	ja
Mainnet-Branch-Setup	offen	ja
Mainnet-CI/CD-Pipeline	offen	ja
Mainnet-Secrets-Storage (Vault?)	offen	ja
Telegram-Push-Channel-Config für Mainnet	offen	ja
Mainnet-Drill-Frequency (alle 30 Tage)	offen	ja

H.4 · Mainnet-Blocker

Alle 5 Layer sind aktiv. Mainnet bleibt blockiert bis: - ✓ alle MH-1..9 abgeschlossen - ✓ Q-MH-1..18 alle decided - ✓ B-FEE-FIX 1-4 verifiziert auf Mainnet-Sample - ✓ B-OUTAGE-RES-1 Operator-Drill auf Mainnet-Connection grün - ✓ Frozen-only Drill (RECON-2.4) erfolgreich - ✓ MH-8 Testnet-Drill 4 Sub-Phasen grün - ✓ Operator-Drill 502 Mainnet-Wiederholung grün - ✓ 2-Personen-Sign-Off etabliert - ✓ 5-Min-Rollback-Plan auf TESTNET geübt

H.5 · Offene Deploy-Themen

Thema	Status
RECON-2.3-DEPLOY	**bereit** — 1 File-Drift, alle anderen synchron
Frozen-only Drill	dokumentiert in `06_testnet_drill §3`
MH-0.5 Worker-Daemon-Aktivierung	dokumentiert in `00_overview §4` + `05_commandbus_worker §6`
MH-7 Bot-Side-Wiring Restart	dokumentiert in `05_commandbus_worker §7`

TEIL I · Review-Fazit

I.1 · Gesamtbewertung der Architektur

✅ **Reife: hoch.** Nach 4 externen Reviews + 6 Q-MH-Decisions + v3.1+v3.2 Patches ist die Architektur **konsistent**, **erweiterbar**, **auditierbar** und **rollbackfähig**.

Strategische Einordnung (vom 4. Reviewer): *„Das Projekt entwickelt sich inzwischen eher wie ein echtes Operator- und Risk-System mit kontrollierter Trading-Engine, nicht mehr wie ein einfacher Trading-Bot."*

I.2 · Größte Risiken

#	Risiko	Mitigation
1	**State-Drift während Mainnet** — externe Wallet-Mutationen, Delistings, Symbol-Renames	6 Drift-Kategorien + G-DR-14 + G-DR-15 + G-DR-20
2	**Operator-Fatigue** auf Mainnet → unbedachte Approves	Multi-Step-Wizard + Hard-Confirm dynamisch + Confidence-Hard-Gate
3	**synthetic_entry wird stillschweigend zur „Wahrheit"**	G-DR-15 mit 4 Pflicht-Marker + UI-Banner
4	**MH-1 wird zu groß** (Big-Bang)	A9 MH-1-Scope-Lock (External Review v2)
5	**Worker-Daemon-Crash unbemerkt**	MH-0.5 Healthcheck + Telegram-Notification
6	**organisatorischer Mainnet-Bypass**	MN-DR-1..5 Verbote + 2-Personen-Sign-Off

I.3 · Größte Stärken

#	Stärke
1	**frozen-by-default + Operator-Authority** als unumstößliches Fundament (G-DR-1+2+9)
2	**CommandBus-only** als einziger Mutation-Pfad (G-DR-3) — verhindert State-Korruption
3	**5-Layer-Mainnet-Block** + organisatorische Disziplin
4	**modulare 12-File-Sub-Roadmap** + 4 Audit-Trail-Files für Reviewer-Trace
5	**21 G-DR durable rules** als zukunftssicherer Vertrag — Architekturdisziplin festgehalten
6	**Multi-Step-Wizard** mit Hard-Confirm dynamisch — kein One-Click-Promote möglich
7	**immutable Audit-Snapshots** + JSON-SoT + DB-Cache — Forensik + Compliance bereit
8	**Two-File-Atomic-Pattern** (G-DR-14) — verhindert wallet-signature-Drift bei Policy-Wechsel

I.4 · Was noch fehlt

Bereich	Status	Wann fertig?
RECON-2.3-DEPLOY	bereit, noch nicht ausgeführt	Operator-GO erforderlich
Frozen-only Drill	bereit dokumentiert	nach RECON-2.3-DEPLOY
MH-1 Code	nicht begonnen	nach Drill
12 verbleibende Q-MH	offen mit Defaults	asynchron entscheidbar
Test-Plan G-DR-18..21 Detail	erwähnt aber nicht ausgearbeitet	v3.3-Patch nach Drill
Mainnet-Branch-Setup	offen	weit in Zukunft

I.5 · Was überengineered wirkt

Nach Re-Audit: **nichts kritisches**. Jede der 21 G-DR + 12 G-SR + 18 MN-DR/SR ist auf konkrete Risiken zurückführbar.

Mögliche Vereinfachung später: - 14 Backlog-Themen sind viel — könnten in „Phase-Sets" gruppiert werden bei Aktivierung (CAP-Set, T-SPLIT-Set, Mobile-Set, etc.) - 4 Hard-Confirm-Patterns könnten in einer Helper-Funktion zentralisiert werden

I.6 · Was gefährlich unterengineered wäre

Bereich	Risiko falls nicht früh gemacht
Worker-Daemon (MH-0.5)	R4-Race in Mainnet wäre fatal
Asset-Identity (G-DR-20)	wrapped-Assets-Verwechselung kann Mainnet-Verluste verursachen
Event-Versionierung (G-DR-18)	Replay-System / ML-Auswertung später unmöglich nachzuziehen
Two-File-Atomic (G-DR-14)	wallet-signature-Drift würde Mainnet-Bot crashen
G-DR-15 synthetic_entry-Marker	Operator-Verwirrung + Tax-Layer-Probleme

I.7 · Was vor MH-1 zwingend geklärt sein muss

✅ **Q-MH-2/11/13/14/15 + DR-7 sind decided.** Damit ist MH-1 startbar.

Empfohlene zusätzliche Klärung VOR MH-1 (nicht zwingend, aber sinnvoll): - Q-MH-17 (Confidence-Threshold) — beeinflusst proposal_engine V1 - Q-MH-18 (Volatility-Kill-Threshold) — beeinflusst proposal_engine V1

Restliche 10 Q-MH können asynchron während Code-Phase entschieden werden.

I.8 · Ist RECON-2.3-DEPLOY bereit?

✅ **JA.**

Begründung: - 1 File-Drift identifiziert (`command_worker.py`) - alle anderen 8 RECON-2.3-relevanten Files synchron - Watchdog-Konvention OK (clawbot) - Backup-Strategie etabliert (G-DR-9) - Pre/Post-Check-Liste in Teil D.4/D.5 verbindlich - kein Bot-Restart erforderlich (main.py importiert command_worker NICHT) - vollständig reversibel via docker cp + alter Hash

Zeit-Aufwand: ~30-45 min mit Backup + Hash-Verify.

I.9 · Ist Frozen-only Drill (RECON-2.4) sinnvoll vorbereitet?

✅ **JA, voll dokumentiert.**

In `06_testnet_drill §3 Sub-Phase A` : - 10 Schritte - Acceptance-Criteria (5 Punkte) - Restore-Plan - Failure-Szenarien

Voraussetzung: RECON-2.3-DEPLOY abgeschlossen.

I.10 · Empfohlene Reihenfolge ab jetzt

#	Phase	Aufwand	Risk
1	**RECON-2.3-DEPLOY**	30-45 min	low
2	**Frozen-only Drill (RECON-2.4 Sub-Phase A)**	1-2h	medium
3	**MH-0.5 Worker-Daemon-Aktivierung**	2-3h + 24h Beobachtung	medium
4	**MH-1 minimal** (Schemas + Reader/Writer + Canonicalization + Validation + Tests)	2-3 Tage	low (Scope-Lock)
5	**MH-2 + MH-4 parallelisierbar**	je 2-3 Tage	medium
6	**MH-3 (proposal_engine V1-min)**	3-4 Tage	medium
7	**MH-5 (Filament UI)**	3-4 Tage	medium
8	**MH-6 (Worker-Handler)**	4-5 Tage	high
9	**MH-7 (Bot-Side-Wiring + Restart)**	1-2 Tage + Drill	high
10	**MH-8 (Testnet-Drill 4 Sub-Phasen)**	1 Tag manueller Drill	high
11	**MH-9 (Hardening)**	2-3 Tage	medium

Gesamt geschätzt: ~3-4 Wochen MH-Code-Phase nach RECON-2.3-DEPLOY + Frozen-only Drill.

I.11 · Technische-Schulden-Bewertung

Schuldenkategorie	Bewertung
Architektur-Design	✅ keine Schulden — Reviews 4× durch
Datenmodell	✅ keine Schulden — Schemas in `01_foundation` vollständig
State-Machine	✅ keine Schulden — 10 States + 18 Transitions + 6 Drift-Kategorien
Test-Strategie	⚠ minor — G-DR-18..21 Detail-Tests fehlen (v3.3-Patch nach Drill)
Backlog-Management	✅ saubere Trennung BACKLOG vs Active
Versionierung	✅ Master-Boundaries v3.2 + Changelog

Final-Bewertung

Architektur-Freigabe: ✅ erteilt **Deploy-Freigabe RECON-2.3-DEPLOY:** ✅ erteilt (mit Pre/Post-Checks Teil D.4/D.5) **Frozen-only-Drill-Entscheidung:** ✅ bereit, kann nach RECON-2.3-DEPLOY starten **MH-1 Readiness:** ⚠ bereit, aber empfohlene Q-MH-17/18 Klärung kann parallel laufen **Langfristige Erweiterbarkeit:** ✅ sehr gut (G-DR-16..21 future-safe) **Risiko-Level (Gesamtprojekt):** mittel — alle kritischen Risiken haben Mitigations **Technische Schulden:** ✅ keine kritischen

— Ende RECON-MH v3.2 Final Review Data Extraction (2026-05-10) —