

Iteration-3 Briefing für Claude Design

Adressat: AI-Designer auf `claude.ai/design` (kein direkter Repo-Zugriff) **Eingabe:** dein eigenes Hand-Off v2 (`Steve_TradingBot.zip` vom 2026-05-10 09:22) **Ausgabe:** Hand-Off v3 (Update v2 → v3, alte Files behalten — parallel) **Stack-Ziel (unverändert):** Laravel + Filament v4 + Tailwind, Dark default **Repo-Stand jetzt:** master `eda3042` (2 Commits weiter als dein v2-Marker `cb555a7`)

0 · Wie du dieses Briefing nutzt

Diese Spec enthält **alle** Daten die du brauchst — du musst keinen Repo-Code lesen. Wenn etwas nicht hier steht, frag zurück (1× zurückmelden), nicht raten.

Für jede Änderung: Datei + Zeilen-Bereich + alter Inhalt + neuer Inhalt. Damit du gezielt patchen kannst.

Sprache: Deutsch für instruktive Teile, Englisch für Code/Tokens/Felder (Repo-Konvention).

Out-of-Scope: alles was nicht explizit unter §5 oder §6 steht. Insbesondere: keine neuen Komponenten, kein Redesign, keine Forms, keine Charts.

1 · Was in v2 stark war (nicht ändern)

✓	Punkt
✓	Token-System mit <code>color-mix(in oklab, ...)</code> — moderner Browser-Tinting-Approach, sauber
✓	Hard-Confirm via Profilname (<code>input.value === active_profile.name</code>) — matcht G10-6-Implementation 1:1
✓	Apply/Clear-Panel split mit roter Spiegelversion
✓	Audit-Timeline-Konzept (collapsed-expandable Detail-Pane)
✓	Worker-Health-Card als eigene KPI
✓	Circuit-Breaker-Chip mit closed/half-open/open
✓	Fonts-lokal-Hinweis (DSGVO) in <code>tokens.css</code>
✓	Section-Trennung: <code>iter2.css</code> + <code>show-iter2.js</code> parallel zu <code>Iter-1-Files</code> (gut für Diff)

→ **Iter-3 ist eine Korrektur-Iteration auf Datenkontrakt-Ebene, kein Redesign.**

2 · Was in v2 NICHT zum Repo passt — must-fix (4)

#	Bereich	Problem	Fix
MF-1	policy-Token-Set	Du hast 8 States: frozen / managed / tradable_quote / quote / dust / blacklisted / quarantined / pending_review	Repo enforced 3 States hart : frozen / managed / tradable_quote . 5 States entfernen.
MF-2	wallet-signature-States	Du hast match / mismatch / stale / unavailable	stale existiert im Repo nicht (Reader liest pro Cycle frisch, kein Cache). Stattdessen 4. State no_baseline einführen für „Datei absent → Bot legacy-mode“
MF-3	audit_events Schema	Du gehst von before_json / after_json / ref_id aus	Reale Tabelle hat 9 Spalten: id, user_id, event_type, subject_type, subject_id, summary, metadata (json), created_at, updated_at . Kein before_json / after_json — alles in metadata . JSON-Diff-Komponente einkürzen auf Single-Pane-metadata-View
MF-4	Worker-Health-Schema	Du hast pid, name, role, cpu_pct, ram_mb, heartbeat_s, queue, state	Reale Felder: pid (file logs/command_worker.pid), claimed_by (string), last_processed_command_id (UUID), queue_depth (count), failed_since_1h (count), status enum . Kein CPU/RAM/Heartbeat/Role im Backend

3 · Should-fix (3)

#	Bereich	Fix
SF-1	Wallet-baseline Row Feldname	qty_baseline → qty_at_baseline (durchgängig)
SF-2	Wallet-baseline Row Felder	signature per row entfernen (signature ist wallet-level, nicht row-level). delta_usdt als „GUI-derived field“ markieren (nicht im Backend)
SF-3	Circuit-Breaker Shape	Real-shape nutzen (siehe §6.5) — service Feld weg ODER hardcoded auf <code>binance.testnet</code>

4 · Nice-to-have (3)

#	Punkt
NTH-1	Build-Marker auf eda3042 aktualisieren
NTH-2	Neue Section: Hard-Confirm-String für Baseline-Apply ist NICHT Profilname, sondern dynamisch <count>:<sha8> (z.B. 47:b25e09fb). Profilname-Pattern bleibt nur für Apply-Profile-Action
NTH-3	Hinweis-Section: RECON-Managed-Holdings ist BACKLOG , Q-MH-1..15 noch offen. v3 darf KEINE managed-lifecycle States (proposal_pending / risk_proposed / managed_active / managed_paused / drift_alert / release_pending / exit_executed) als UI mocken

5 · Konkrete File-Edit-Anweisungen

5.1 iter2.css

Edit MF-1 — policy-Token-Set reduzieren:

ENTFERNEN (5 Klassen):

```
.pol-quote      { ... }
.pol-dust        { ... }
.pol-blacklisted { ... }
.pol-quarantined { ... }
.pol-pending     { ... }
```

BEHALTEN (3 Klassen):

```
.pol-frozen      { ... } /* Bot ignoriert */
.pol-managed     { ... } /* Bot trackt + tradet */
.pol-tradable    { ... } /* Quote-Asset (USDT/USDC) – Bot darf ausgeben */
```

Edit MF-2 — wallet-signature-States anpassen:

ENTFERNEN:

```
.wsig-stale     { ... }
```

HINZUFÜGEN:

```
.wsig-no-baseline { color: var(--text-tertiary); background: var(--bg-surface-2);
border-color: var(--border-subtle); border-style: dashed; }
```

Begründung: 4 States bleiben (match / mismatch / unavailable / no-baseline), stale weg.

Edit SF-3 — Circuit-Breaker Shape (optional CSS-Anpassung):

Falls du `cb-state-`-spezifische Klassen hast (z.B. `.cb-retry`): kein direkter CSS-Edit nötig, aber stelle sicher dass JS keine `service`-Variable mehr verwendet (siehe §5.2).

5.2 `show-iter2.js`

Section „policy-Badge“ (~Z.48-79): - Reduziere die Showcase-Cells von 8 auf 3 (`frozen`, `managed`, `tradable`) - Ergänze Tooltip-Text pro Cell: - `frozen` → "Bot ignoriert dieses Asset" - `managed` → "Bot trackt + tradet (nach Operator-Approval)" - `tradable` → "Quote-Asset (USDT/USDC) — frei ausgebbar"

Section „wallet-signature-Chip“ (~Z.81-103): - Ersetze `stale` durch `no-baseline` Cell - Cell-Texte: - `match` → "Wallet-Signature OK" - `mismatch` → "MISMATCH — Bot refused start" - `unavailable` → "Live-Balance Fetch failed" - `no-baseline` → "No baseline applied yet"

Section „Worker-Health-Card“ (~Z.105+): - Ersetze KPI-Cells: - **WEG:** `cpu_pct`, `ram_mb`, `heartbeat_s`, `role` - **NEU:** `pid` (Mono), `claimed_by` (string Mono), `last_processed_command_id` (Mono, truncated 13 chars), `queue_depth` (number), `failed_since_1h` (number, rot wenn >0) - Status-States: nur `idle` / `running` / `degraded` / `down` (kein `up`)

Section „Circuit-Breaker-Chip“ (~Z.105+): - Entferne `service`-Anzeige (oder fest auf `binance.testnet`) - Felder pro State: - `closed` → grün, Label „Exchange OK“ - `half-open` → gelb, Label „Probing recovery N/M“ (N = `half_open_successes`, M = `half_open_required_successes`) - `open` → rot, Label „Buy paused“ + `remaining-cooldown` (derived: `opened_at + cooldown_seconds - now`)

Section „Audit-Timeline“ (~Z.254+): - JSON-Diff-Komponente reduzieren: kein side-by-side before/after - Einfache Single-Pane-View: `event_type` (Mono, color-coded prefix), `summary` (text), `created_at` (Mono), Expand-Trigger zeigt `metadata` als formatted-JSON-Block (collapsed by default) - Color-coding `event_type` prefix: - `profile.apply.*` → sky/info - `profile.runtime_config.*` → amber/warning - `risk_settings.*` → emerald/success - `baseline.*` → indigo - `managed.*` → **NICHT mocken** (Backlog)

Section „baseline-diff Modal“ (~Z.349+): - Feldname `qty_baseline` → `qty_at_baseline` durchgängig - Entferne `signature`-Spalte (nur wallet-level Hinweis im Modal-Header zeigen) - `delta_usdt` als kursiv markieren mit Tooltip „GUI-derived field“ - Hard-Confirm-Input erwartet `<count>:<sha8>` Pattern (nicht Profilname) - Beispiel-Confirm-String: `47:b25e09fb`

Section „RECON-2.3 Screen“ (~Z.444+): - gleicher Feld-Rename `qty_at_baseline` - Mainnet-Banner-Text exakt: "TESTNET · Mainnet durably blocked · BINANCE_TESTNET=true" - Build-Marker im Hero auf `eda3042` aktualisieren

5.3 `HANDOFF-v3.md` (parallel zu `HANDOFF-v2.md`, alte nicht überschreiben)

Verbindliche Sections: 1. **Header:** Repo-Stand `eda3042`, Phasen abgedeckt: G10-5/6, T-SPLIT-2-RESTART, RECON-2.1, **RECON-2.2a/b**, **RECON-2.3 (PHP)**, **RECON-2.3c (Bot-Worker)**, B-OUTAGE 2. **Was neu in v3:** Tabelle der MF/SF/NTH-Korrekturen (Felder, States,

Datenkontrakte) 3. **Datenkontrakte (verbindlich):** kompletter Block aus §6 unten — alle 6 Schemas 4. **Hard-Confirm-Patterns (zwei!):** Profile-Apply (Profilname) + Baseline-Apply (<count>:<sha8>) + Clear (statisch clear-baseline oder Profilname) 5. **Bekannt offen:** RECON-Managed-Holdings ist BACKLOG; Q-MH-1..15 nicht entschieden; v3 mockt KEINE managed-lifecycle States 6. **Out-of-Scope:** Forms, Charts, Mobile, Managed-UI

6 · Datenkontrakte (verbindlich, kopier 1:1 in HANDOFF-v3.md)

6.1 baseline_holdings.json (Bot-Side, geschrieben durch BaselineHoldingsWriter)

```
{
  "_meta": {
    "schema_version": "recon-2.1",
    "captured_via": "operator_apply | recon_dry_run | ...",
    "canonical_hash": "<64-hex sha256>",
    "command_id": "<uuid>"
  },
  "captured_at": "2026-05-10T...",
  "environment": "testnet",
  "default_policy_for_unlisted": "frozen | tradable_quote",
  "wallet_signature": {
    "exchange": "binance",
    "account_hash": "sha256:<64-hex>",
    "total_assets": 47
  },
  "holdings": {
    "BTC": { "qty_at_baseline": 1.0, "policy": "frozen" },
    "SOL": { "qty_at_baseline": 5.0, "policy": "managed" },
    "USDT": { "qty_at_baseline": null, "policy": "tradable_quote" }
  }
}
```

Rules (hart): - policy ∈ {frozen, managed, tradable_quote} — exact 3, nichts anderes - default_policy_for_unlisted ∈ {frozen, tradable_quote} — 'managed' ist NIE als Default erlaubt - tradable_quote Slots haben IMMER qty_at_baseline=null - frozen/managed Slots haben IMMER eine numerische qty_at_baseline ≥ 0 - environment ist immer "testnet" (Mainnet hard-blocked)

6.2 audit_events Tabelle (Postgres, GUI-DB)

id	bigint	PK	
user_id	bigint	nullable	- Operator/Admin der die Action ausgelöst hat
event_type	varchar		- z.B. profile.apply.requested
subject_type	varchar	nullable	- polymorphic Eloquent-Model class
subject_id	varchar	nullable	- UUID der referenzierten Resource
summary	varchar		- human-readable (1 Zeile)
metadata	json	nullable	- alle Action-Daten (command_id, checksum, idempotency_key, ...)
created_at	timestamp		
updated_at	timestamp		

Reale event_types die im Repo emittiert werden: - profile.apply.dry_run_requested - profile.apply.requested - profile.apply.precondition_rejected - profile.runtime_config.clear_requested - config_profile.created - risk_settings.applied - baseline.apply.requested - baseline.apply.precondition_rejected - baseline.clear.requested

Beispiel metadata -JSON für profile.apply.requested :

```
{
  "command_id": "acc8a14c-aae6-43f6-b7cd-e50b53a549a5",
  "profile_id": "b4033597-0f90-4c6a-9b55-2a31648f10d5",
  "profile_version": 1,
  "dry_run": false,
  "environment": "testnet",
  "checksum": "f86a171bab47...",
  "idempotency_key": "apply:b4033597-...:1:apply:f86a171b",
  "expires_at": "2026-05-09T12:57:10Z",
  "risk_settings_keys": ["cash_reserve_pct", "max_open_positions", ...],
  "config_summary_keys": ["log_level", "decision_log_verbosity"]
}
```

→ JSON-Diff-Komponente macht NICHT before/after side-by-side. Stattdessen: summary als Text-Zeile + collapsed metadata -Block (<details> -Expander mit pretty-printed JSON).

6.3 commands Tabelle (Worker-State)

```

id                bigint    PK
command_id        uuid      UNIQUE
command_type      varchar    - apply_profile_testnet, clear_runtime_config,
apply_baseline_holdings, clear_baseline_holdings, noop, close_position, ...
status            varchar    - pending / claimed / running / succeeded / failed /
cancelled
payload_json      json
result_json       json
error_message     text      nullable
idempotency_key   varchar    UNIQUE
requested_by      bigint    nullable
claimed_by        varchar    nullable - Worker-ID wie "worker-a8f2c4e1"
claimed_at        timestamptz nullable
started_at        timestamptz nullable
finished_at       timestamptz nullable
timeout_at        timestamptz nullable
cancelled_at      timestamptz nullable
created_at, updated_at timestamp

```

→ Audit-Timeline kann `audit_events` ↔ `commands` via `metadata->>'command_id' = commands.command_id` joinen für full-Verlauf.

6.4 Worker-Health (derived aus DB + PID-File)

Quellen: - `commands.claimed_by` (string, Worker-ID) - `commands.status`, `commands.finished_at`
 - File `logs/command_worker.pid` (PID oder absent)

Aggregations für UI:

```

{
  "pid": 4419 | null,                // aus PID-File
  "claimed_by": "worker-a8f2c4e1" | null, // letzter Worker der Command claimed hat
  "last_processed_command_id": "uuid-..." | null, // commands wo status='succeeded' AND
  finished_at MAX
  "last_processed_at": "2026-05-10T..." | null,
  "queue_depth": 3,                // count(commands) where status IN
  ('pending', 'claimed', 'running')
  "failed_since_1h": 0              // count(commands) where status='failed'
  AND finished_at > NOW() - 1h
}

```

State-Mapping für Chip:

state	Bedingung	Visual
idle	PID absent + queue_depth=0	grau
running	PID vorhanden + ≥1 command status='running'	blau pulsiert
degraded	queue_depth > 5 OR failed_since_1h > 0	warning gelb
down	PID absent + queue_depth > 0	danger rot

→ **Kein** `cpu_pct`, `ram_mb`, `heartbeat_s`, `role` — diese Felder sind im Backend nicht verfügbar (Worker läuft heute manuell `--once`, keine Daemon-Heartbeats; CPU/RAM bräuchten `docker stats` Endpoint, separate Phase).

6.5 Circuit-Breaker Snapshot (`LiveTrader._circuit_breaker.snapshot()`)

```
{
  "state": "closed | half_open | open",
  "consecutive_failures": 0,
  "failure_threshold": 3,
  "cooldown_seconds": 300,
  "opened_at": null | <unix_ts>,
  "half_open_attempts": 0,
  "half_open_successes": 0,
  "half_open_probes": 2,
  "half_open_required_successes": 2
}
```

UI-derived fields: - `retry_at_s` = `opened_at` + `cooldown_seconds` - `now` (nur sinnvoll wenn `state='open'`) - `recovery_progress` = `half_open_successes` / `half_open_required_successes` (für „Probing N/M“ Label)

Boundary: - Es gibt **nur einen** CircuitBreaker pro LiveTrader, dedicated zur Binance-Connection. Kein `service`-Feld nötig (UI kann `binance.testnet` hardcoden). - Settings kommen aus `.env` als `OUTAGE_*`-Keys: `OUTAGE_CIRCUIT_OPEN_THRESHOLD`, `OUTAGE_CIRCUIT_OPEN_DURATION_MIN`, `OUTAGE_CIRCUIT_HALF_OPEN_PROBES`, `OUTAGE_CIRCUIT_HALF_OPEN_REQUIRED_SUCCESSES`.

6.6 Hard-Confirm-Patterns (zwei!)

Action	Pattern	Beispiel	wo definiert
Apply Profile to Bot (G10-6)	exakt \$record->name (Case-sensitive)	g10-3-test-profile	ProfileActionFactory::applyToTestnetAction()
Clear Runtime Config (G10-6)	exakt \$record->name	g10-3-test-profile	ProfileActionFactory::clearRuntimeConfigAction()
Apply Baseline Holdings (RECON-2.3)	<count>: <sha8> dynamisch	47:b25e09fb	BaselineHoldings::deriveHardConfirm()
Clear Baseline Holdings (RECON-2.3)	statisch clear-baseline	clear-baseline	BaselineHoldings::clearAction()

→ Designer muss in v3 dokumentieren: **welcher Pattern für welchen Modal**. Profilname-Pattern ist NICHT universal.

7 · UI-Beispiel-Strings (verbindlich, 1:1 nutzen)

Element	exakter String
Mainnet-Banner-Text	TESTNET · Mainnet durably blocked · BINANCE_TESTNET=true
Build-Marker im Hero	commit eda3042 · G10-6 · T-SPLIT-2 · RECON-2.3c · B-FEE-FIX 1-4 · B-OUTAGE-RES-1
Bot-PID Beispiel (Worker-Health Mock)	18185
Container-Name	clawbot (NICHT steve-tradingbot — alter falscher Name)
Apply-Profile Hard-Confirm Beispiel	g10-3-test-profile
Apply-Baseline Hard-Confirm Beispiel	47:b25e09fb
Clear-Baseline Hard-Confirm Beispiel	clear-baseline
Strategy-Group Tags (T-SPLIT-3)	t1_core (sky), t2_pump_dump (amber), t3_copy_trading (slate, planned-disabled , NIE emittiert), legacy_unknown (slate)
Reject-Reason Beispiele in Decision-Logs	baseline_frozen, stable_or_peg_asset, circuit_breaker_open, balance_fetch_failed

8 · Workflow für v3-Lieferung

1. Behalte alle v2-Files unverändert (für Diff-Reviewer)
2. Edit `iter2.css` mit den 3 Edits aus §5.1
3. Edit `show-iter2.js` mit den 6 Section-Edits aus §5.2
4. Edit `index.html` : Build-Marker im Hero auf `eda3042`
5. NEUE Datei `HANDOFF-v3.md` schreiben (HANDOFF-v2.md NICHT löschen)
6. Pack alles als zip, lade hoch

Optional aber empfohlen: v3-changelog-Section in `HANDOFF-v3.md` am Anfang mit **Referenz auf MF-1..MF-4 / SF-1..SF-3 / NTH-1..NTH-3 IDs** aus diesem Briefing — macht Reviewer-Trace einfach.

9 · Was du NICHT ändern darfst

File	Begründung
shell.css , components.css , components-2.css , page.css	Iter-1 Foundation, stabil
show-foundations.js , show-components.js , show-dashboard.js	Iter-1 Showcases — bleiben Read-Only-Referenz
tokens.css	Foundation; einzige erlaubte Änderung wäre Geist-Font-Lokal-Pfade falls deployed (unkritisch)
HANDOFF.md (v1)	historisches Dokument, nicht löschen
HANDOFF-v2.md	dein eigenes v2 — bleibt für Diff

10 · Out-of-Scope für v3 (BACKLOG)

Diese Themen erst in späteren Iterationen:

Thema	Warum jetzt nicht
Forms (Strategy-Editor, Profile-Editor)	wartet auf Backend-Form-Schema; nicht aktuell blockiert
Charts (PnL, Equity, Drawdown)	wartet auf G3-Aggregator + uPlot-Entscheidung
Mobile-Pass	Operator-Default = Desktop, eigener Pass später
Managed-Holdings-Lifecycle UI	RECON-Managed-Holdings ist BACKLOG; Q-MH-1..15 unbeantwortet; State-Machine proposal_pending → risk_proposed → managed_active → managed_paused → managed_drift_alert → release_pending → exit_executed darf NICHT gemockt werden, weil Backend-Schemas erst gefroren werden müssen
Trade History Table	wartet auf Audit-Timeline-Pattern aus v3 als Vorbild

11 · Zusammenfassung — was v3 muss

Hard-Liste (10 Edits insgesamt):

- ✓ MF-1: 5 policy-Klassen aus iter2.css löschen + 5 Showcase-Cells aus show-iter2.js
- ✓ MF-2: wsig-stale durch wsig-no-baseline ersetzen
- ✓ MF-3: Audit-Timeline JSON-Diff durch Single-Pane-metadata-View
- ✓ MF-4: Worker-Health KPI-Felder austauschen (CPU/RAM/heartbeat/role → pid/claimed_by/last_processed/queue/failed_1h)
- ✓ SF-1: qty_baseline → qty_at_baseline rename

6. ✓ SF-2: `signature` per row entfernen, `delta_usdt` als derived markieren
7. ✓ SF-3: Circuit-Breaker Shape an `snapshot()` anpassen (`service` weg/hardcoded)
8. ✓ NTH-1: Build-Marker `eda3042`
9. ✓ NTH-2: Hard-Confirm-Section mit beiden Patterns (Profilname + `count:sha8`)
10. ✓ NTH-3: Hinweis Managed-Holdings BACKLOG

Liefer-Format: zip mit allen Files (alte v2 + neue v3) → upload nach `.ccgram-uploads/` .

Boundary-Reminder: - Nur die 5 Files (`iter2.css` , `show-iter2.js` , `index.html` , neue `HANDOFF-v3.md`) anfassen - Iter-1-Files bleiben unverändert - Keine neuen Komponenten, keine neuen States außer denen aus §6 - Keine Managed-Lifecycle-UI (BACKLOG) - Bei Unklarheit: 1× zurückmelden, nicht raten

— Ende Iteration-3-Briefing —