G10-3b — Scope-Bestätigung

Steve Trading Bot — apply_profile_testnet Bot-side Dry-Run Handler

Datum: 2026-05-09

Master-Tip: `4519fc8`

Bot-PID: 14381 (B-FEE-FIX-4-1 + B-OUTAGE-RES-1 aktiv)

Status: vor Implementierung — wartet auf GO mit Antworten auf 3 offene Fragen.

Geplante Dateien

Datei	Status	Inhalt
`trading/command_worker.py`	M	+1 dispatch elif, +1 PERMITTED_ENVIRONMENTS entry, +1 Handler `_handle_apply_profile_testnet` (~150 LoC), +Allowlist-Konstanten (G7_ALLOWED_SUMMARY_KEYS, G9_ALLOWED_RISK_KEYS, RISK_BOUNDS), +Cross-Field-Helper
`trading/tests/test_g10_3b_apply_profile_testnet.py`	NEU	~600 LoC, 33 Tests in 7 Klassen
`trading/tests/run_g10_3b_tests.sh`	NEU	Runner analog `run_g65_tests.sh` : throwaway python:3.11-slim + isolated test-DB schema-clone + drop

Handler-Name

`_handle_apply_profile_testnet(self, cmd: dict) -> dict` in `command_worker.py` . Fügt sich zur bestehenden if/elif-Dispatcher-Chain (line ~227) und zum `PERMITTED_ENVIRONMENTS` -Dict (line ~114) hinzu — gleicher Pattern wie die 5 existierenden Handler.

Validation-Kaskade (Reihenfolge fest, fail-fast)

```
0. _mark_running(cmd)  + audit "apply_profile_testnet.dry_run_started"

1. PAYLOAD-VALIDATION (12 Checks, fail-fast):
   - payload.dry_run === true                          else: validation/dry_run_must_be_true
   - payload.environment === "testnet"                 else: validation/environment_must_be_testnet
   - payload.exchange_connection_id === null           else: validation/exchange_connection_must_be_nul
   - payload.profile_id present + UUID format          else: validation/missing_or_bad_profile_id
   - payload.version present + int >= 1                else: validation/missing_or_bad_version
   - payload.checksum present + sha256-hex (64 chars)  else: validation/checksum_format
   - payload.expires_at (if present) > NOW()           else: validation/expired

2. DB-LOOKUP (6 Checks):
   - SELECT profile FROM config_profiles WHERE profile_id=$1
   - profile found                                     else: validation/profile_not_found
   - profile.version === payload.version               else: validation/version_mismatch
   - profile.status === 'ready'                        else: validation/profile_not_ready
   - profile.source === 'gui_edit'                     else: validation/profile_source_not_gui_edit
   - profile.environment === 'paper' (Phase-1 hard)    else: validation/profile_env_not_paper
   - SELECT risk_settings FROM risk_settings WHERE config_profile_id=$1

3. SCHEMA-VALIDATION (Allowlist + Bounds + Cross-Field):
   - alle keys in profile.summary_json in G7_ALLOWED_SUMMARY_KEYS  else: validation/unknown_summary_key
   - alle keys in risk_settings in G9_ALLOWED_RISK_KEYS            else: validation/unknown_risk_key
   - pro risk_setting: value in [min_value, max_value] (DB) ∪ [hardcoded conservative bounds]
                                                                    else: validation/risk_bounds_violat
   - Cross-Field a) tier_low <= tier_mid <= tier_high             else: validation/cross_field_tier_ord
   - Cross-Field b) daily_loss_limit <= weekly_loss_limit          else: validation/cross_field_daily_w
   - Cross-Field c) cash_reserve + fee_buffer < 1.0                else: validation/cross_field_cash_fe
   - Cross-Field d) tier_high_allocation <= max_total_exposure     else: validation/cross_field_tier_ma

4. AUDIT + STATUS-TRANSITION:
   - on success:  _mark_succeeded(cmd, result_json) + audit "apply_profile_testnet.dry_run_succeeded"
   - on failure:  _mark_failed(cmd, error_message)   + audit "apply_profile_testnet.dry_run_failed"
   - NEVER mutate runtime_config.json / live_portfolio.json / .env / commands beyond status+result/erro
```

Audit-Event-Shape

3 mögliche `command_audit_log.event_type`:

```
apply_profile_testnet.dry_run_started      (immer beim Handler-Eintritt)
apply_profile_testnet.dry_run_succeeded    (bei Validation-Pass)
apply_profile_testnet.dry_run_failed       (bei Validation-Fail oder Exception)
```

`context_json` Schema (durable):

```
{
  "profile_id": "<uuid>",
  "profile_pk": <int>,
  "version": <int>,
  "dry_run": true,
  "environment": "testnet",
  "exchange_connection_id": null,
  "checksum_status": "valid_hex|missing|bad_format",
  "checksum_match": "skipped",
  "expires_at_status": "valid|expired|missing",
  "validated_summary_keys": ["log_level", "decision_log_verbosity"],
  "validated_risk_keys": ["max_risk_per_trade_pct", "..."],
  "risk_settings_count": <int>,
  "validation_errors": [{"code": "...", "field": "...", "got": ..., "expected": ...}],
  "warnings": [],
  "would_apply": false,
  "runtime_mutation": false,
  "handler_phase": "G10-3b",
  "elapsed_ms": <int>
}
```

Status-Transition

Resultat	commands.status	commands.result_json	commands.error_message	command_audit_log.event_type
Validation-Pass + dry-run ok	succeeded	`{ok:true, dry_run:true, validated:true, ...}`	NULL	dry_run_succeeded
Validation-Fail	failed	NULL	`validation/<reason>` (<=200 chars)	dry_run_failed
Unexpected Exception	failed	NULL	`str(exc)[:200]`	dry_run_failed

→ Bestehende State-Machine respektiert. Kein `rejected`-Status (existiert nicht in G6.5).

Testplan (33 Tests in 7 Klassen)

Klasse 1 — HandlerWiringTests (3)

1. `test_handler_function_exists` — `_handle_apply_profile_testnet` ist Methode auf CommandWorker
2. `test_dispatch_chain_routes_command_type` — Worker.run_one mit ctype="apply_profile_testnet" → handler
3. `test_permitted_environments_only_testnet` — `PERMITTED_ENVIRONMENTS["apply_profile_testnet"] == {"testnet"}`

Klasse 2 — PayloadValidationTests (10)

4. `test_dry_run_happy_path` — vollständig korrekter Payload + DB-Profile → succeeded
5. `test_missing_profile_id_fails`
6. `test_missing_version_fails`
7. `test_dry_run_false_rejected`
8. `test_environment_paper_rejected`
9. `test_environment_mainnet_rejected`
10. `test_exchange_connection_id_non_null_rejected`

11. `test_checksum_missing_rejected`
12. `test_checksum_bad_format_rejected`
13. `test_expired_command_rejected`

Klasse 3 — DBValidationTests (5)

14. `test_profile_not_found_fails`
15. `test_profile_version_mismatch_fails`
16. `test_profile_archived_rejected`
17. `test_profile_source_bot_capture_rejected`
18. `test_profile_env_testnet_rejected`

Klasse 4 — SchemaAllowlistTests (6)

19. `test_unknown_summary_key_rejected`
20. `test_unknown_risk_key_rejected`
21. `test_risk_bounds_violation_fails`
22. `test_cross_field_tier_order_fails`
23. `test_cross_field_daily_weekly_fails`
24. `test_cross_field_cash_fee_fails`
25. `test_cross_field_tier_max_exposure_fails`

Klasse 5 — AuditAndStatusTests (3)

26. `test_audit_log_started_then_succeeded`
27. `test_audit_log_started_then_failed`
28. `test_commands_status_updated_correctly`

Klasse 6 — NoMutationTests (5)

29. `test_no_config_profiles_mutation` — pre/post row-hash compare
30. `test_no_risk_settings_mutation`
31. `test_no_live_portfolio_mutation` — sha256 unchanged
32. `test_no_runtime_config_json_written` — file does NOT exist before AND after
33. `test_no_state_directory_writes` — `trading/state/` mtime unchanged

Klasse 7 — IntegrationTests (3)

34. `test_full_dry_run_with_g10_3a_test_profile` — end-to-end success
35. `test_full_dry_run_failure_with_archived_profile`
36. `test_concurrent_command_lock_works` — FOR UPDATE SKIP LOCKED

→ **36 Tests** (User wollte 20–30, ich gehe leicht drüber für Klasse 7 Integration; falls strikt: 33 ohne Klasse 7).

Boundary-Tests (AST/Source-grep)

Eigene Test-Klasse `BoundaryASTTests` mit 5 Tests:

#	Check	Methode
B 1	Keine `create_market_buy_order` / `create_market_sell_order` / `create_order` Calls innerhalb `_handle_apply_profile_testnet`	AST-walk: für jeden ast.Call innerhalb der Funktion-AST prüfen, dass callee-name nicht in FORBIDDEN_ORDER_NAMES
B 2	Keine `live_portfolio.json` open(..., 'w') / write() Calls	AST-walk: für jeden ast.Call mit "open" als func name prüfen, dass weder zweiter Arg "w"/"a"/"r+" enthält noch das filename "live_portfolio" enthält. Plus: keine `.write_text()` oder `.write_bytes()` mit live_portfolio im Arg
B 3	Keine `runtime_config.json` writes	AST-walk: kein open() oder write_text() mit "runtime_config.json" im Arg-String
B 4	Keine `.env` writes	AST-walk: kein open()/write_text() mit ".env" im Arg-String
B 5	Kein `ccxt.binance(...)` / `_get_exchange(...)` Constructor in Handler-Scope	AST-walk: kein ast.Call mit func.attr in {"binance","binance_testnet","_get_exchange","Exchange"}

Strategie: parse `command_worker.py`, finde die FunctionDef für `_handle_apply_profile_testnet`, walke AST nur in dieser Funktion. Selbsttest: ein synthetischer Code-Snippet mit echtem forbidden-call wird gefangen (gleiches Pattern wie STATE-RECON-1 AST-Test).

Live-Verifikationsplan für G10-3c (separater Schritt, nicht in G10-3b)

Phase	Aktion	Wer	Erwartet
1. Code-Sync	`docker cp /projekte/.../command_worker.py clawbot:/home/.../command_worker.py`	Operator	Hash-Match host vs container
2. Optional Worker-Start	`docker compose --profile workers up -d clawbot-worker` ODER one-shot `docker exec clawbot python3 -m trading.command_worker --once`	Operator	Container running, `command_worker.pid` written
3. Filament-UI: Test-Command erzeugen	im Filament `CommandResource`-Form: `command_type=apply_profile_testnet`, payload mit profile_id der G10-3a Test-Profile (`b4033597-…`), version=1, dry_run=true, environment=testnet, exchange_connection_id=null, checksum=valid sha256-hex	Operator	command-row in `commands` table mit status=pending
4. Worker pickt Command	warte bis poll_interval (oder one-shot)	passiv	command claimed → running → succeeded
5. `command_audit_log` zeigt Dry-Run-Report	`SELECT event_type, context_json FROM command_audit_log WHERE command_id=...`	Operator/Audit	2 rows: started + succeeded mit context_json shape wie spezifiziert
6. Bot-Side-Health	live_portfolio.json hash unverändert, no orders, .env mtime unverändert, kein runtime_config.json	Operator	alle ok
7. Bot main.py unberührt	PID-Check + log-grep auf "apply_profile"	Operator	main.py-Prozess hat 0 Hits, Worker-Prozess hat >=1 Hit

Out-of-Scope für G10-3c: dry_run=false (das ist G10-4), Mainnet, Risk-/Apply-Mutation.

Offene Fragen (3, brauchen Bestätigung vor Code)

#	Frage	Mein Vorschlag	Bestätigung nötig?
Q1	`commands.status` für Validation-Failure	`failed` (kein `rejected`, das existiert nicht in der State-Machine)	ja — User-Spec erwähnte rejected, aber wir brauchen Migration falls erwünscht
Q2	Checksum-Match-Verification	nur Format-Check in G10-3b. Hash-Match braucht shared canonicalisation mit PHP-side, eigene Mini-Phase	ja
Q3	Cross-Field-Regel-Liste	4 Regeln (siehe Validation-Kaskade Schritt 3); falls PHP-side mehr hat, brauche Pointer	ja

Stop-Regeln

Bedingung	Action
Test-DB-Schema-Clone schlägt fehl	abort, kein Worker-Start
Bot-PID 14381 ändert sich während Implementation	sofort STOP, diagnose
`.env` mtime ändert sich	STOP
live_portfolio.json hash ändert sich durch G10-3b-Aktion	STOP
Tests nicht stabil grün (>= 1 Failure)	STOP, bericht
AST-Boundary-Test fängt verbotenen Call	abort vor commit
Push-Versuch	nicht möglich (kein Remote), aber STOP wenn detected

Hard-Boundaries für G10-3b (durable)

```
KEIN Worker-Start
KEIN Runtime-Apply
KEIN dry_run=false (Handler weist es als ersten Validation-Check zurück)
KEIN live_portfolio.json Write
KEIN runtime_config.json Write
KEINE Orders (AST-validated)
KEIN ccxt-Konstruktor (AST-validated)
KEINE .env-Mutation
KEIN Mainnet
KEIN GUI-Apply-Enable (ConfigProfilePolicy::apply()=false bleibt)
KEIN G10-4 (mutation)
KEIN Rollback
KEIN neuer commands.status
KEIN Push (Repo hat kein Remote)
```

Lieferbericht-Plan (was nach Implementation kommt)

Nach Code+Tests werden geliefert:

- Geänderte Dateien (Diff-Stat)
- Vollständige Liste der 33–36 Tests mit PASS/FAIL pro Test
- AST-Boundary-Test-Resultat (5/5)
- Volltest-Regression-Sweep (erwartet 380+ green)
- DB-State-Verifikation (config_profiles + risk_settings unverändert nach Test-Run)
- Bot-Health-Sweep (PID, .env mtime, hash)
- Live-Verifikationsplan für G10-3c (durable im Memory)

Stop

Scope-Bestätigung übergeben. Wartet auf:

- **GO G10-3b** + Antworten auf die 3 offenen Fragen (oder: alle 3 OK → mein Vorschlag fix)
- ODER **Anpassung X** (z.B. „Klasse 7 weglassen", „Q3 hat 6 Regeln, hier sind sie", etc.)

Stand: 2026-05-09 06:42 UTC. Vor Implementierung. Erfordert explizites GO.