Lokaltreu AGENTS.md — Gold-Standard

Einleitende Analogie: „Die Verfassung der KI“

Stell dir das Repo als Staat vor. Code sind Gesetze, CI ist die Justiz, und AGENTS.md ist dieVerfassung.

Sie legt verbindlich fest, wer was tun darf, wie entschieden wird und welche Prüfungen jedenMerge regeln. Ohne Verfassung herrscht Auslegungssache; mit Verfassung entstehtVorhersagbarkeit, Wiederholbarkeit, Auditierbarkeit.

**0) Metadaten**

• **Geltung**: Verbindlich für alle Codex-Aktionen (CLI/IDE) und menschliche Beiträge.• **Zielgruppe**: Einsteiger bis Senior. Kurze Sätze. MUSS/SOLL/KANN.• **Version**: AGENTS.md v2.2-gold

• **Owner**: Tech Lead

• **Letzte Prüfung**: 29. Oktober 2025

**1) Referenzen**

• **SPEC**: Technische Spezifikation – Lokaltreu v2.0 • **ARCH**: Architektur-Lokaltreu.pdf

• **ROADMAP**: Lokaltreu Gesamt-Roadmap 2.2• **OpenAPI**: apps/api/openapi/lokaltreu-openapi-v2.0.yaml• **Konvention**: Bei Widerspruch gilt Reihenfolge: SPEC → OpenAPI → ARCH → AGENTS.md → Code.

**2) Projektfakten**

• **Architektur**: Modularer Monolith (TypeScript), Next.js PWA, Node.js API • **Standards**: OpenAPI 3.1, RFC 7807, Idempotency-Key

• **Sicherheit**: Ed25519 Device-Proof, WORM-Audit, R2-Export, Retention 180 Tage • **Environments**: dev, stage, prod (EU-Region), isolierter Remote-State

**3) Build | Run | Test**

• **Install**: npm ci

• **Build**: npm run build • **Lint**: npm run lint

• **Tests**: npm test --workspaces -- --coverage• **Web Dev**: npm run -w apps/web dev• **API Dev**: npm run -w apps/api dev

• **Mock-Server**: npx prism mock apps/api/openapi/lokaltreu-openapi-

v2.0.yaml -p 4010

**4) Contracts & Standards**

• **SSOT**: OpenAPI 3.1. schema\_drift = 0 ist MUSS. • **Codegen**:

• npx openapi-typescript apps/api/openapi/lokaltreu-openapi-v2.0.yaml \ • -o packages/types/src/index.d.ts

**5) Rollen, Guardrails, KPIs, Mehrwert**

• **Prinzip**: Eine Aufgabe ruft stets eine primäre und ggf. eine sekundäre Rolle auf. • **Contract-Sheriff**

o **Aufgabe**: OpenAPI-Codegen, Type-Sync, Schema-Drift-Erkennung o **No-Gos**: Handgeschriebene API-Typen

o **KPI**: schema\_drift = 0

o **Mehrwert**: Keine Überraschungen zwischen Frontend und Backend • **ProblemJSON-Arbiter**

o **Aufgabe**: RFC 7807-Factory, einheitliche Fehlercodes o **No-Gos**: Ad-hoc-Fehler

o **KPI**: error\_conformity = 100%

o **Mehrwert**: Debugging beschleunigt, Support reproduzierbar • **Idempotency-Guardian**

o **Aufgabe**: Anti-Replay via Redis SETNX + TTL, Header-Pflicht o **No-Gos**: Idempotenz im Business-Code

o **KPI**: replay\_blocks = 100%

o **Mehrwert**: Kein Doppelbuchen, saubere Audit-Spuren • **Device-Proof-Engineer**

o **Aufgabe**: Ed25519-Validierung für X-Device-Proof o **No-Gos**: Private Keys serverseitig speichern o **KPI**: proof\_failures\_caught = 100% o **Mehrwert**: Missbrauchsschutz bei sensiblen Aktionen• **Audit-Officer**

o **Aufgabe**: WORM-Audit, signierte Exporte → R2, Retention 180 Tageo **No-Gos**: PII in Logs, fehlende Artefakteo **KPI**: audit\_gaps = 0

o **Mehrwert**: Prüf- und Nachweisfähigkeit • **Test-Pilot**

o **Aufgabe**: Teststrategie, Parallelisierung, Contract-Tests, Coverage o **No-Gos**: Snapshots für JSON-APIs

o **KPI**: coverage ≥ 80%, contract\_pass = 100% o **Mehrwert**: Qualität messbar, Regressionen sichtbar• **Docs-Keeper**

o **Aufgabe**: JSDoc, README, „Was ist neu?“-Abschnitte, Entwickler-

Onboarding

o **No-Gos**: Marketing-Sprache, veraltete Beispiele

o **KPI**: docs\_freshness ≤ 7 Tage, broken\_links = 0

o **Mehrwert**: Einstiegspunkte klar • **KPI-Zielwerte (Bestätigung)**

o schema\_drift = 0 o coverage ≥ 80%

o replay\_blocks = 100%o proof\_failures\_caught = 100%o audit\_gaps = 0

**6) CI-Gates (MUSS) + KPIs**

• **Coverage**: lines ≥ 80%, functions ≥ 80%, branches ≥ 80%, statements ≥ 80%• **Contract**: OpenAPI-Validierung grün; schema\_drift = 0• **Fehlerformat**: error\_conformity = 100% (RFC 7807)• **Anti-Replay**: Paralleltest grün; replay\_blocks = 100%• **Device-Proof**: Positiv/Negativ-Fälle grün

• **Plan-Gates**: Referral Starter → 403 PLAN\_NOT\_ALLOWED getestet • **OpenAPI-Lint**: pass

• **Terraform**: fmt + validate grün (EU-Region erzwungen)• **GDPR**: Checks grün (Art. 11, Retention 180 Tage)• **Merge-Block**: Automatisierte Gates verhindern Merge bei Fehlschlag (required checks aktiv)

**7) Workflows und Begründung der Trennung**

• **ci.yml**: Kernqualität (Lint, Codegen, Build, Tests, Contract, Coverage).

Vorteil: Schnell, fokussiert, läuft bei jedem PR.

• **gdpr-compliance.yml**: Retention, Art. 11, Log-PII-Checks, Artefakt-Aufbewahrung. Vorteil: Rechtliche Anforderungen separat versionier- und eskalierbar.• **security-gates.yml**: Device-Proof, Anti-Replay, Rate-Limits. Vorteil: Sicherheitsregeln mit eigenen Lastprofilen und Reviewern.• **Strategischer Gewinn**: Entkoppelte Verantwortlichkeiten, schnellere Feedback- Schleifen, klare Ownership und Audits pro Domäne.

**8) Aufgabenrezepte (CLI/IDE, kurz und reproduzierbar)**

• 1) OpenAPI → Types

• codex exec "Generiere @lokaltreu/types aus OpenAPI und fixe Imports"

\

• --role Contract-Sheriff \

• --context @apps/api/openapi/lokaltreu-openapi-v2.0.yaml

@packages/types/\*\*

• **2) Handler (idempotent, RFC 7807)**

• codex exec "Implementiere POST /stamps/claim idempotent, Fehler

RFC7807" \

• --role Idempotency-Guardian,ProblemJSON-Arbiter,Audit-Officer \

• --context @apps/api/openapi/\*\* @apps/api/src/mw/\*\*

@apps/api/src/handlers/\*\*

• **3) Contract-Tests**

• codex exec "Schreibe Vitest Contract-Tests mit chai-openapi-response-

validator" \

• --role Contract-Sheriff,Test-Pilot \ • --context @apps/api/openapi/\*\* @tests/\*\* • **4) Parallel-Anti-Replay**

• codex exec "Erstelle Anti-Replay-Test für Idempotency-Key (10

parallel)" \

• --role Idempotency-Guardian,Test-Pilot \ • --context @apps/api/src/mw/idempotency.ts @tests/security/\*\* • **5) Plan-Gate**

• codex exec "Füge Plan-Gate (403 PLAN\_NOT\_ALLOWED) inkl. Tests hinzu"

\

• --role ProblemJSON-Arbiter,Test-Pilot \ • --context @apps/api/src/handlers/referrals.ts @apps/api/openapi/\*\* • **6) Device-Proof**

• codex exec "Implementiere Ed25519-Validierung für X-Device-Proof" \ • --role Device-Proof-Engineer \

• --context @apps/api/src/middleware/device-auth.ts • **7) Docs-Sync**

• codex exec "Aktualisiere JSDoc+README mit Beispielen" \ • --role Docs-Keeper \

• --context @apps/api/src/\*\* @README.md

**9) Security & DSGVO**

• **Keine PII in Logs**. Nur erlaubt: tenant\_id, device\_id, card\_id. • **Secrets nur via SOPS** (\*.enc.tfvars).

• **EU-Region erzwingen**.

• **Art. 11 DSR** ohne Zusatzidentifizierung.

• **Retention 180 Tage**. **WORM-Audit**. **Signierte Exporte** (R2).

**10) Audit-Trail**

• Jeder Codex-Run per tee spiegeln:

• … | tee -a artifacts/codex-$(date +%Y%m%d-%H%M%S).log

**11) ZENTRALE PR-CHECKLISTE (Referenz für Mensch & KI)**

• **Nutzen**: Ein Schalter pro Gate. Einsteiger sehen sofort, was fehlt. KI kann gezielt

nachbessern.

• **Lint grün**

• **Build grün**

• Tests grün + Coverage ≥ 80 **%**

• **Contract-Tests grün, schema\_drift = 0**

• **Fehler 100 % RFC 7807 konform**

• **Parallel-Anti-Replay grün (1×201, 9×409)**

• **Device-Proof-Fälle grün**

• **Plan-Gate-**Cases grün (Starter → 403 PLAN\_NOT\_ALLOWED)

• **OpenAPI-Lint pass**

• **Terraform fmt + validate grün (EU-Only)**

• **GDPR-Checks grün (Art. 11, Retention 180 Tage)**

• **Audit-Artefakte erzeugt (Logs, Diffs, Reports)**

• **Commit nach Conventional Commits, PR-Beschreibung verlinkt CI-Run +**

**benannte Rolle(n)**

**12) Pflege & Versionierung**

• Änderungen an AGENTS.md nur via PR. Reviewer: Owner + betroffene Rolle. • Quartalsreview: Rollen, KPIs, Gates.

• Changelog-Eintrag in docs/CHANGELOG.md unter „Governance“.

**13) CI-Sentinels (für automatisches Linting dieser Datei)**

• Diese Zeilen dienen der CI-Validierung. Nicht löschen. • [SENTINEL] SECTION: Projektfakten

• [SENTINEL] SECTION: CI-Gates (MUSS)• [SENTINEL] SECTION: Aufgabenrezepte• [SENTINEL] SECTION: Audit-Trail

• [SENTINEL] SECTION: ZENTRALE PR-CHECKLISTE

**14) Anhang: Implementierungs-Snippets**

• **RFC 7807-Factory (vereinfacht)**

• import type { components } from '@lokaltreu/types'; •

• type ErrorCode = components['schemas']['Problem']['error\_code']; •

• export interface ProblemDocument { • type: string; title: string; status: number; • error\_code: ErrorCode; correlation\_id: string; • detail?: string; instance?: string; • } •

• export const problem = (p: ProblemDocument) => • new Response(JSON.stringify(p), { • status: p.status,

• headers: { 'content-type': 'application/problem+json' } • });