

# TranceCreate — Übergabeprotokoll (v1.1 • 2025-08-30)

## 0. Executive Summary

**TranceCreate (TC)** ist der eigenständige Veredelungs-Service über der ANNI-Baseline. TC verbessert Stil, Tonalität und Überzeugungskraft („Copywriting“) **ohne** die Invarianten zu brechen. TC ist **fail-closed**: Bei Unsicherheit oder Policy-Verstoß liefert TC die Baseline zurück.

**Komponenten & Ports** - TranceCreate API (FastAPI) — Port **8095** - Guard (ANNI) — Port **8091** (Baseline-Quelle) - Mistral (optional) — Port **8092** ( /generate )

**Kernmerkmale v1.1** - Baseline aus Guard; Freezing von Platzhaltern/HTML/URLs/Zahlen/Emojis - Mistral-Integration **oder** heuristischer Fallback (deterministisch) - Policies: `preserve`, `max_change_ratio`, `forbidden_terms`, `domains_off` - Deterministische **Seeds** (explizit oder stabiler Hash) - Transparente **degrade\_reasons** bei Rückfall auf Baseline - Tracing: Guard-/TC-Latenzen, Modell, Seed; einfache Diffs/Char-Ratio

## 1. Architektur & Datenfluss

```
Client → TC /transcreate → (wenn keine baseline_text übergeben) Guard /  
translate → Baseline  
↓  
Text (tc_core: Mistral ODER Fallback) → Vorschlag  
Policy-/Invariant-Check → Baseline ODER Transcreated
```

**Trennung der Zuständigkeiten** - **Guard** garantiert Korrektheit (Invarianten, Chunking, Ports stabil) und liefert die Ausgangsbasis. - **TC** optimiert Stil/Copy unter Einhaltung von Policies.

## 2. API-Vertrag (stabilisiert)

**Endpunkte** - `GET /health` → `{ok:true, role:"TranceCreate", ready:true, version:"1.1.0"}` - `GET /profiles` → verfügbare `profiles`, `personas`, optionale Locale-Hinweise - `POST /transcreate` → Request:

```
{  
  "source":"en", "target":"ja", "text":"...",  
  "baseline_text": null,  
  "profile":"marketing", "persona":"ogilvy", "level":2,  
  "seed": null,  
  "policies":{  
    "preserve":
```

```
[
  "placeholders", "single_brace", "html", "numbers", "urls", "emojis",
  "max_change_ratio": 0.25,
  "forbidden_terms": [],
  "domains_off": ["legal", "privacy", "tos", "gdpr"]
}
```

Response:

```
{
  "baseline_text": "...",
  "transcreated_text": "...",
  "degraded": false,
  "degrade_reasons": [],
  "diffs": {"char_ratio": 0.18, "ops": []},
  "applied": {"profile": "marketing", "persona": "ogilvy", "level": 2, "policies":
    { }},
  "trace": {"guard_latency_ms": 123, "tc_latency_ms": 45, "tc_model": "mistral|
    fallback", "seed": 12345}
}
```

**Fail-Closed-Regel** - Wenn Invarianten/Policies verletzt → `degraded=true`, `degrade_reasons=[...]`,  
`transcreated_text = baseline_text`.

### 3. Konfiguration

**Umgebungsvariablen** - `TC_GUARD_URL` (default `http://127.0.0.1:8091/translate`) - `TC_API_KEY` (Header `X-API-Key` zum Guard) - `TC_MISTRAL_URL` (default `http://127.0.0.1:8092/generate`) - `TC_USE_MISTRAL` (`true|false`, default `true`) - `TC_TIMEOUT` (Sek., default 90)

**Konfigdateien** - `config/trance_profiles.json` (Profile + CTA/Emoji/Hints je Sprache) - `config/tc_personas.json` (Persönlichkeiten, Stilregler) - `config/tc_locales.json` (Markt-Nuancen pro Ziel)

### 4. Pipeline & Stages

1) **Baseline**: falls keine `baseline_text` übergeben → Guard call 2) **Freeze**: `{{...}}`, `{token}`, HTML-Tags, URLs, Emojis, Zahlen (wie Guard) 3) **tc\_core** - **Mistral-Weg**: System+User Prompt (Persona/Profile/Level/Locale), `seed` weiterreichen - **Fallback-Weg**: konservative, satzweise Glättung; CTA/Emoji nur bei `level>0`; max Änderung ≤ `max_change_ratio/2` 4) **Policy-Check**: `preserve` / `max_change_ratio` / `forbidden_terms`; Domains-Off (kein aggressives Marketing in legal/privacy) 5) **Post**: leichte Markenkonformität (Profile/Locales), keine Änderung gefreezter Elemente 6) **Diff/Trace**: Edit-Distanz, `char_ratio`, Latenzen, Modell, Seed

**Seeds/Determinismus** - `seed` vom Request oder stabiler Hash aus  
(baseline,target,profile,persona,level)

---

## 5. Betrieb

**Start** - Script: `start_trance_creation.sh` (lädt Env, prüft Guard, prüft optional Mistral, startet TC 8095)

**Health/Smoke** - `GET 127.0.0.1:8095/health` → ok - `GET 127.0.0.1:8095/profiles` → Profile/Personas sichtbar - Beispiel-Call:

```
curl -sS -H 'Content-Type: application/json'
--data-binary '{"source":"en","target":"ja","text":"Only today {{COUNT}}
...","profile":"marketing","persona":"ogilvy","level":2}'
http://127.0.0.1:8095/transcreate
```

**Logs** - `/tmp/tc_server.log` (Uvicorn/stdout)

---

## 6. Qualität & Tests

**scripts/test\_tc\_fallback.py** - Case A: `TC_USE_MISTRAL=false` → `tc_model=fallback`, `degraded=false` - Case B: `forbidden_terms=["guarantee"]` → `degraded=true`, reason `forbidden_term:...` - Case C: `max_change_ratio=0.05` → `degraded=true`, reason `max_change_ratio_exceeded` - Seeds: gleicher Input ⇒ gleicher Output bei aktivem Mistral

**Akzeptanzkriterien** - Invarianten bleiben erhalten (wie Guard) - Keine globalen Space-Normalisierungen; Spacing der Baseline bleibt erhalten - Fail-Closed greift immer nachvollziehbar

---

## 7. Sicherheit & Compliance

- **Preserve** schützt Preise, Zahlen, Platzhalter, Links
  - **Domains-Off** verhindert „Veredelung“ in sensiblen Bereichen (legal/privacy)
  - **Audit:** `trace.seed`, `tc_model`, Latenzen, `diffs.char_ratio`
  - **Mandantenfähigkeit** (optional): Request-Felder `org_id`, `project_id` vorsehen
- 

## 8. Bekannte Limitationen & Risiken

- Mistral-Prompting kann Aussagen zuspitzen; Policies begrenzen Risiko
  - Script-/Schriftvarianten (z. B. `zh` Hans/Hant) nicht erzwungen → optionaler Post-Konverter
  - Kosten/Latenz steigen mit Textlänge und Level; Cache empfehlenswert
-

## 9. Backlog (Priorität absteigend)

1) **Plugin-Staging**: sauberes Stage-Interface (`pre_tc`, `tc_core`, `post_tc`, `policy_check`, `ranker`) inkl. Hot-Reload 2) **Ranking**: Mehrere Varianten generieren, heuristisch/LLM-basiert ranken 3) **Terminologie**: Kunden-Glossare einbinden (harte/sanfte Regeln) 4) **A/B-Support**: deterministische Seeds je Variante, Multi-arm Rückgabe 5) **Caching**: Key = Hash(baseline,target,profile,persona,level,policies) 6) **Observability**: Prometheus-Metriken, per-Stage Latenzen 7) **GUI**: TC-Tab mit Profil/Persona/Level, `degrade_reasons` sichtbar, Diff-Viewer 8) **Policy-Ausbau**: Claim-Whitelists, „Do-not-change“ Regeln je Kunde 9) **Batch/Jobs**: `/jobs` für große Mengen, Queue + Status API

---

## 10. Anhänge

**.env Beispiel** (`~/ .env.tc`)

```
TC_GUARD_URL=http://127.0.0.1:8091/translate
TC_API_KEY=topsecret
TC_MISTRAL_URL=http://127.0.0.1:8092/generate
TC_USE_MISTRAL=true
TC_TIMEOUT=90
```

### Schnellstart

```
nohup uvicorn tc_server:app --host 0.0.0.0 --port 8095 >/tmp/tc_server.log
2>&1 &
```

### Beispiel-Profiles (Auszug)

```
{
  "marketing": {"cta": {"ja": "今すぐチェック", "de": "Jetzt entdecken", "en":
"Shop now"}, "emoji": "#"},
  "social": {"emoji": "✅"}
}
```

**Hinweis:** TranceSpell (QC/Spell, geplant Port 8096) ist eigenes Dokument.