

TranceSpell — Übergabeprotokoll (v1.0 • 2025-08-30)

0. Executive Summary

TranceSpell (TS) ist der optionale Quality-Check-Dienst für **Quelltexte** (und auf Wunsch auch Zieltexte). Er liefert Rechtschreib-, Grammatik- und Stil-Hinweise, wendet auf Wunsch **sichere Auto-Korrekturen** an und bewahrt dabei alle ANNI-Invarianten (Platzhalter, HTML, Zahlen, Emojis, URLs, Absätze). TS ist **separat** betreibbar und wird vor ANNI/TC in die Pipeline gehängt.

Komponenten & Ports - TranceSpell API (FastAPI) — Port **8096** - LanguageTool Server (intern/extern) — Port **8097** (nur falls intern gestartet) - Guard (ANNI) — Port **8091** (unverändert)

Kernziele - Finde & melde Fehler in der **Quelle**, ohne gefreezte Elemente zu verändern - Biete „safe fixes“ (konservativ) und lasse riskante Änderungen optional - Integriere sich nahtlos in CLI/GUI; pro Job aktivierbar (per Toggle)

1. Funktionsumfang

- **Spell & Grammar:** Rechtschreibung, einfache Grammatik, häufige Stilpatzer
- **Freeze-Aware:** `{{...}}`, `{token}`, HTML-Tags, URLs, Emojis, Zahlen, Absätze
- **Konfigurierbar** je Request: `mode` (facts/marketing/legal), `level` (0-3), `auto_fix` (none/safe/aggressive)
- **Kundendictionary:** `extra_words`, Ignorierlisten, Regel-Disable
- **Reports:** Findings mit Regel-ID, Offsets, Kontext, Vorschlägen; optional `fixed_text`

Nicht-Ziele v1.0 - Kein stilistisches Umschreiben (das macht **TranceCreate**) - Keine tiefen domänenspezifischen Regeln (kommen als Add-ons)

2. Architektur & Datenfluss

```
Client (GUI/CLI) —(optional TS)—▶ TS /spellcheck —▶ Findings/fixed_text
                                     |
                                     └─▶ ANNI Guard /translate —▶
... —▶ TC (optional)
```

Ablauf in TS 1) **Pre-Freeze** der sensiblen Elemente 2) **Segmentierung** (Sätze/Absätze) 3) **Checks** via LanguageTool HTTP (oder lokaler Adapter) 4) **Filter & Safe-Fixes** abhängig von `mode` / `level` / Policies 5) **Re-Unfreeze** → Findings-Offsets werden auf Originaltext gemappt

3. API-Vertrag

GET /health → {ok:true, role:"TranceSpell", ready:true, version:"1.0.0"}

GET /languages → listet erkannte/unterstützte Sprachen, z. B. [{code:"de", name:"German"}, ...]

POST /spellcheck Request:

```
{
  "lang": "de",
  "text": "Nur heutee: 10,00 €!",
  "mode": "facts",
  "level": 1,
  "auto_fix": "safe",
  "preserve":
  ["placeholders", "single_brace", "html", "numbers", "urls", "emojis"],
  "extra_words": ["TranceLate"],
  "ignore": ["(?i)Ogilvy"],
  "disable_rules": ["EN_QUOTES"]
}
```

Response:

```
{
  "findings": [
    {
      "rule": "GERMAN_SPELLER_RULE",
      "message": "Möglicher Rechtschreibfehler: heutee",
      "replacements": ["heute"],
      "offset": 4,
      "length": 8,
      "context": "Nur heutee: 10,00 €!",
      "severity": "warning"
    }
  ],
  "fixed_text": "Nur heute: 10,00 €!",
  "counters": {"errors": 1, "warnings": 0},
  "trace": {"lt_latency_ms": 42, "freeze_spans": 2}
}
```

Regeln Safe-Fixes (v1.0) - Nur eindeutige Wörterbuch-Typos (1-Zeichen-Diff, keine Namen) → auto-fix - Keine Änderungen in gefreezten Bereichen - Währung/Format: konstante normalisieren wir **nicht** automatisch (nur Hinweis)

4. Konfiguration

Umgebungsvariablen - `TS_PORT` (default 8096) - `TS_LT_URL` (default `http://127.0.0.1:8097/v2/check`) - `TS_START_LT` (`true|false`) - internen LanguageTool-Server starten - `TS_TIMEOUT` (Sek.) - `TS_DEFAULT_MODE` (`facts|marketing|legal`) - `TS_DEFAULT_LEVEL` (0-3)

Interner LanguageTool-Start (optional) - Java Prozess als Sub-Prozess auf **8097**; bei Stop säubern - Alternativ externen LT-Server nutzen (Docker/JAR), TS zeigt nur Health

5. Spracherkennung & Aliasse

- `lang` wird wie bei ANNI normalisiert (`de-DE` → `de`, `pt-BR` → `pt`, `iw` → `he`)
 - Unterstützte LT-Sprachen: de, en, fr, es, it, pt, nl, pl, ru, uk, ... (CJK/Indic eingeschränkt)
 - Fallback bei nicht unterstützten Codes: **nur Rechtschreibung** mit Hunspell/Wordlist (Backlog)
-

6. Policies & Modi

mode=facts (konservativ) - Grammatik & Rechtschreibung, keine Stil-Risiken; Auto-Fix nur sichere Typos

mode=marketing (etwas mutiger) - Zusätzlich gängige Satzzeichen/Leerzeichen-Korrekturen (niemals HTML/Platzhalter anrühren)

mode=legal (sehr konservativ) - Nur Hinweise, keine Auto-Fixes; stärkeres Whitelisting (kein Eingriff in Zitate, Paragraphen, Datumsangaben)

`level` 0-3 skaliert Empfindlichkeit und erlaubte Fix-Arten (innerhalb des Modus)

7. Integration (GUI/CLI/Pipeline)

GUI - Toggle „Source prüfen (TranceSpell)“ - Pane „Findings“ mit Apply-Buttons je Vorschlag; „Apply Safe Fixes“ für alles sichere

CLI - `anni --spell de --mode facts --level 1 < input.txt` → gibt `findings.json` und optional `fixed.txt` aus

Pipeline 1) Optional TS → `fixed_text` (oder original) 2) Guard /translate → Baseline 3) Optional TC /transcreate → Veredelung

8. Betrieb

Start - `start_trancespell.sh` prüft optional LT, startet TS (8096)

Health - `GET 127.0.0.1:8096/health` → ok; TS prüft bei Bedarf LT-Reachability

Logs - `/tmp/trancespell.log` (Uvicorn)

9. Qualitätssicherung

Smoke - Deutsch-Typos im Satz, Platzhalter/HTML/URLs enthalten → Fixes nur außerhalb der Freezes

Quality Gate v1 (Scripts) - `T1` Platzhalter/HTML/URLs/Emojis bleiben unangetastet - `T2` Safe-Fix ersetzt einfache Typos - `T3` mode=legal: keine Auto-Fixes - `T4` Große Texte: Performance < 1.5× baseline (Batch-Segmentierung)

10. Sicherheit & Datenschutz

- API-Key Header optional (wie ANNI) – empfehlen aktivieren
 - Keine Texte persistieren; Logs ohne Inhalte (nur Metadaten)
 - Kundendictionaries als Dateien je Mandant (`/config/dicts/<org>.txt`)
-

11. Bekannte Limitationen

- CJK/Thai/Laoisch: Tokenisierung bedingt → eingeschränkte Qualität
 - Sprachen ohne LT-Support: v1.0 liefert nur Hinweise (oder überspringt)
 - Währungs-/Datumsformate werden v1.0 nicht automatisch normiert
-

12. Backlog

1) Hunspell-Adapter + Wordlists für zusätzliche Sprachen 2) Benutzerdefinierte Regeln (Regex-Checks je Kunde) 3) Termdatenbank-Abgleich („verbotene/gewünschte Begriffe“) 4) PDF/HTML-Input mit Positionsmapping 5) Batch-Jobs + Queue 6) Prometheus-Metriken

13. Anhänge

.env Beispiel (`~/ .env.ts`)

```
TS_PORT=8096
TS_LT_URL=http://127.0.0.1:8097/v2/check
TS_START_LT=true
TS_TIMEOUT=60
TS_DEFAULT_MODE=facts
TS_DEFAULT_LEVEL=1
```

Beispiel-Requests

```
curl -sS -H 'Content-Type: application/json'  
  --data-binary '{"lang":"de","text":"Nur heutee: 10,00  
€!","auto_fix":"safe"}'  
  http://127.0.0.1:8096/spellcheck
```

Akzeptanzkriterien v1.0 - Freezes bleiben exakt erhalten - Safe-Fixes korrigieren eindeutige Typos, sonst nur Hinweise - GUI/CLI können Findings anzeigen und Safe-Fixes anwenden - Performance akzeptabel für ~5k-10k Zeichen je Request