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

0. Executive Summary

ANNI ist der robuste Übersetzungs-Baseline-Dienst. Er garantiert Invarianten (Platzhalter, HTML, Zahlen, Emojis, Absätze) und liefert stabile Übersetzungen über 99 Zielsprachen (plus en). Der Stack ist auf Ports fixiert, hat Health-Checks, „Quality Gates“ und CLI/GUI-Flows. Dieses Dokument enthält Architektur, Betrieb, Tests, Fehlerbilder und To-Dos.

Komponenten & Ports - Guard (FastAPI) — Port **8091** - Worker M2M100 (FastAPI) — Port **8093** - GUI (static) — Port **8094** - Metrics — Port **8092** (separat, nicht für MT-Traffic)

Kernmerkmale - Invarianten-Schutz: `{{…}}`, `{token}`, HTML-Tags, Emojis, Zahlen/Zeiten, Absätze - Chunking: ~600 Zeichen pro Teilstück; Mid-Pivot nur wenn weder Quelle noch Ziel Englisch ist - max_new_tokens: default 512 (steuerbar), kein Early Truncation - Sprach-Aliasse: Normalisierung von Locales (`ja-JP`→`ja`, `zh-CN`→`zh`, `pt-BR`→`pt`, `iw`→`he` …) - 99 Zielsprachen verifiziert (langs.json), en ausgenommen - Spacing: **keine** globalen Space-Normalisierungen; Original-Abstände werden respektiert

1. Architektur & Datenfluss

Request → GUI/CLI → Guard (8091) → optional EN-Pivot → Worker (8093) → Guard Post-Processing → Response.

Guard (8091) - Endpunkte: `/meta`, `/translate` - Verantwortungen: Freeze/Unfreeze, Segmentierung, Chunking, Mid-Pivot, Post-Processing (ohne Space-Zwang), Invarianten-Check - Konfig: `MT_BACKEND` (Basis-URL des Workers), `ANNI_MAX_NEW_TOKENS`, `ANNI_CHUNK_CHARS`, `MT_TIMEOUT`

Worker (8093) - Modell: `facebook/m2m100_418M` - Endpunkte: `/health`, `/translate` - Respektiert `max_new_tokens`

GUI (8094) - Lokale Datei `anni_gui.html` im Projektordner - JS-Helfer: `scripts/presets.js`, optional `scripts/api_rewrite.js`, `scripts/copy_cli.js`, `scripts/tc_client.js` (TC-Button)

Dateien - `mt_guard.py` (Guard) - `anni/models/m2m_worker.py` (Worker) - `langs.json` (verfügbare Sprachcodes, inkl. en) - `lang_aliases.json` (Locale→M2M-Mapping) - `start_anni.sh` (Start/Readiness) - `scripts/verify_stack.py`, `scripts/anni_smoke.py` (Tests) - Logs: `/tmp/m2m_worker.log`, GUI/Browser-Console

2. Betrieb (Start/Stop/Health)

Start (launchd) - LaunchAgent: `~/Library/LaunchAgents/com.trancelate.anni.plist` lädt `start_anni.sh` - Env-Datei: `~/.env.anni` (Ports, API-Key, Timeouts)

Manuell prüfen - Guard Meta: `GET http://127.0.0.1:8091/meta` → `backend_alive:true`, `backend_url:http://127.0.0.1:8093` - Worker Health: `GET http://127.0.0.1:8093/health` → `{ok:true, ready:true}`

GUI - `http://127.0.0.1:8094/anni_gui.html` (Targets wählen, Translate auslösen)

3. Qualitätssicherung (Quality Gates)

Gate v3 (Kurzlauf) verifiziert: - Worker: `/health` ok, `max_new_tokens` Wirkung (32 < default ≤ 512) - Guard: Invarianten ok (ph/html/num/paren/len) - Absätze bleiben erhalten (doppelte Newlines) - Long-Text-Chunking: keine Trunkierung; Platzhalter am Ende vorhanden - Optional CLI-Check (`~/bin/anni`)

Sprachsweeps - en→ *(alle 99 Ziele) PASS* - →en: Len-Ratio-Schranken gelockert (kurze Texte toleranter); bei Bedarf erneut laufen lassen

4. Invarianten (was darf nie brechen)

- **Platzhalter**: `{{NAME}}` bleibt exakt erhalten
- **Single-Brace Tokens**: `{app}` unverändert
- **HTML**: gleiche Tag-Signatur; Attribute werden nicht verändert
- **Emojis**: werden nicht entfernt/ersetzt; Original-Abstände beibehalten
- **Zahlen/Zeit**: Ziffern bleiben vorhanden; 12h→24h wird als lenient gewertet
- **Absätze**: `\n\n` bleibt erhalten
- **Längenverhältnis**: dynamisch; kurze Quellen großzügiger toleriert

5. Sprachen & Aliasse

- `langs.json`: 100 Codes inkl. `en`; 99 Zielsprachen im Sweep verifiziert
- `lang_aliases.json`: Normalisiert häufige Locale-Varianten auf M2M-Codes
- **Limitationen**: `zh` ohne Script-Zwang (Hans/Hant); `sr` ohne Script-Zwang (Cyrl/Latn). Optionaler Post-Step denkbar (Konverter).

6. CLI & GUI

CLI - Syntax: `anni <src> <tgt> 'Text…'` - Mehrfachziele: `anni <src> -m <t1,t2,…> 'Text…'` - Lange Texte: via Datei/STDIN nutzen; Guard chunked intern

GUI - Presets: CJK/Indic/RTL, Africa, Asia, Europe, Europe+ (nur Test-Hilfen) - TC-Button vorhanden; ruft TC-Server (8095) separat an

7. Fehlerbilder & Troubleshooting

HTTP 404 vom Worker - Ursache: falscher Pfad (`/` statt `/translate`) oder falscher Port - Fix: `MT_BACKEND` muss Basis oder `/translate` sein; Guard hängt fehlendes `/translate` selbst an

HTTP 502 im Guard - Ursache: Worker nicht erreichbar oder Fehler-Body - Fix: Worker neu starten; `m2m_worker.log` prüfen; `max_new_tokens` ggf. erhöhen

„ready:false“ in `/health` - Modell noch nicht geladen; kurz warten; danach erneut testen

Emoji/Spacing - Keine Zwangs-Spaces; Spiegelung der Quelle

Port-Drift - Guard: 8091, Worker: 8093, GUI: 8094. In GUI keine 8092-MT-Aufrufe verwenden

8. Sicherheit & Policies

- API-Key Header: `X-API-Key` (GUI/CLI setzen)
- Keine externen Calls im Guard/Worker
- Logs enthalten keine Texte, wenn nicht explizit aktiviert

9. Backlog (Priorität absteigend)

1) Chinesisch Script-Konvertierung (Hans/Trad) als optionaler Post-Step 2) Serbisch Script-Konvertierung (Cyrl/Latn) optional 3) Terminologie/TM Anbindung (Soft-Match ist skizziert; produktiv machen) 4) Caching von Worker-Antworten (Hash-Key aus src/tgt/text) 5) GUI: Job-History, Retry, minimaler Progress 6) Metrics: konsolidieren, Guard-Metriken ergänzen 7) *→en-Sweep weiter justieren (Len-Ratio)

10. Anhänge

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

```
ANNI_GUARD_PORT=8091
ANNI_WORKER_PORT=8093
ANNI_GUI_PORT=8094
MT_BACKEND=http://127.0.0.1:8093
ANNI_MAX_NEW_TOKENS=512
ANNI_CHUNK_CHARS=600
ANNI_API_KEY=topsecret
```

Schnelltest - Guard Meta: `GET 127.0.0.1:8091/meta` - Worker Health: `GET 127.0.0.1:8093/health` - Smoke (GUI/CLI): kurzer Text mit Emoji/HTML/Zahlen, Check-Objekt `ok:true`

Hinweis: TranceCreate (TC, Port 8095) und TranceSpell (QC/Spell, geplant 8096) sind **separate** Dokumente. Dieses Protokoll deckt **nur ANNI** ab.