

STATUS · Handbuch · Offener Kataster · Werkstand v3.1 · Bündel v3.8 · kein Produktionsbetrieb
ZAHLEN · alle Befunde synthetisch (Demo-Daten)
PARTNER · Ein-Mensch-Werkstatt · unabhängig · privat finanziert
DATUM · 2026-09-23 · Kennung FORSCH-PROV0445 (PROVISORISCH)
AUTOR · Thomas Schüller · Syntagma Forschung · Wien · ORCID 0009-0003-9799-6747

Offener Kataster — Handbuch

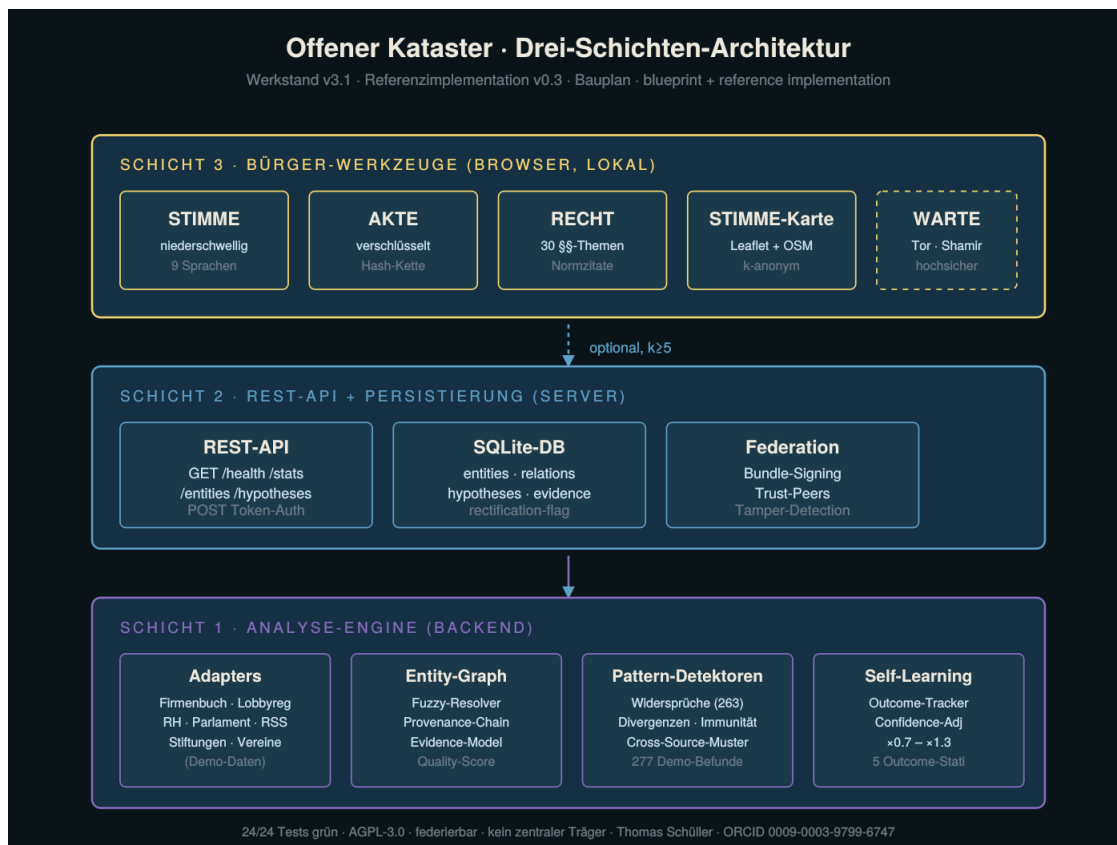


Abbildung 1: Die drei Schichten, denen die Kapitel dieses Handbuchs folgen.

Dieses Handbuch beschreibt Installation, Start und Betrieb aller drei Stränge. Die Werkbeschreibung erklärt Zweck, Grenzen und Invarianten; hier geht es um die Bedienung.

Kapitel	Strang	Archiv
1	Backend (Analyse-Engine, REST-API, Förderierung)	Strang A
2	Hosting einer eigenen Instanz	Strang A
3	Bürger-Werkzeuge im	Strang B

Kapitel	Strang	Archiv
	Browser	
4	Manifeste	Strang C

1 • Backend

Stand: Mai 2026 · AGPL-3.0 · v0.3

Was ist das

Die Analyse-Engine des Offenen Katasters. Python, läuft serverseitig. Crawlts öffentliche Quellen, baut einen Entity-Graph mit Quality-Scoring, produziert Hypothesen-Befunde, persistiert in SQLite, bietet REST-API, fördert mit anderen Forks.

Inhalt

backend/	
├── core/	
│ ├── graph.py	Entity-Graph mit Quality-Scoring
│ ├── entity_resolver.py	Fuzzy-Match
│ (Levenshtein/Jaccard/Substring)	
│ └── db_layer.py	SQLite-Persistierung
├── ingestion/	
│ ├── adapters.py	9 Adapter (Firmenbuch,
│ Lobbyreg, ...)	
│ ├── stimme_bridge.py	STIMME→Graph mit k-Anonymität
│ └── foundations_adapters.py	Stiftungen, NGOs, parteinahe
Vereine	
├── realtime_adapters.py	Parlament, Ministerien, RH
├── graph/	
│ └── cross_source_synthesizer.py	5 Pattern-Detektoren
├── modules/	
│ ├── contradictions.py	Widerspruchs-Engine (Demo: 263
│ Paare aus 73 Katalogregeln)	
│ ├── regulation_catalog.py	Regelbibliothek
│ ├── precedent_divergence.py	OGH/VfGH ↔ EuGH/EGMR
│ ├── mortality_anomaly.py	Sterblichkeits-Anomalien
│ ├── immunity_tracker.py	Auslieferungsverfahren
│ ├── vulnerable.py	k≥5 für Schutzbedürftige
│ ├── institution_atlas.py	41 österreichische Institutionen
│ ├── sozialsystem.py	AMS / BFA / MA40 Tracker
│ └── outcome_tracker.py	Self-Learning-Loop
├── api/	
│ └── rest_api.py	HTTP-API mit Token-Auth (lesend
offen)	
├── federation/	
│ └── federation.py	Bundle-Signing, Cross-Fork-Sharing
├── crawlers/	
│ └── gemeinde_protokoll.py	Gemeinderats-Beschlüsse
└── tests/	

```
| └─ test_core.py
| └─ run_pipeline.py
```

24 unittest Tests
End-to-End-Pipeline

Schnellstart

```
tar xzf kataster-strang-A-backend.tar.gz
cd backend
python3 run_pipeline.py           # 277 Demo-Befunde produziert
python3 tests/test_core.py       # 24 Tests grün
```

REST-API starten

```
python3 -c "
from api.rest_api import KatasterAPI, TokenStore, make_handler
from core.db_layer import KatasterDB
from http.server import HTTPServer
db = KatasterDB('kataster.db')
tokens = TokenStore()
api = KatasterAPI(db, tokens)
HTTPServer(('127.0.0.1', 8080), make_handler(api)).serve_forever()
"
```

Dann erreichbar: - GET /api/v1/health — Status - GET /api/v1/stats — globale Statistik - GET /api/v1/entities?type=ORGANIZATION&limit=100 — Listen - GET /api/v1/entities/{id} — Einzelner Eintrag mit Relationen - GET /api/v1/hypotheses — Befunde - POST /api/v1/dispute — Eintrag als bestritten markieren (Token-Auth)

Föderierung

```
from federation.federation import FederationBundle, sign_bundle,
verify_and_apply
```

Sender:

```
bundle = FederationBundle(issuer="kataster.example.at",
                           hypotheses=[])
signed = sign_bundle(bundle, private_key=...)
```

Empfänger:

```
report = verify_and_apply(signed, db_b,
                           trusted_peers=["kataster.example.at"],
                           public_keys={...})
print(report) # {'valid_signature': True, 'hypotheses_imported':
N, ...}
```

Module einzeln laufen lassen

Alle Module haben `__main__`-Demos:

```
python3 ingestion/stimme_bridge.py    # k-Anonymität
python3 core/entity_resolver.py        # Fuzzy-Match
python3 core/db_layer.py               # SQLite-Demo
python3 modules/sozialsystem.py        # AMS/BFA/MA40
python3 modules/outcome_tracker.py     # Self-Learning
```

```
python3 modules/institution_atlas.py      # Institutionen
python3 federation/federation.py          # Föderierung mit Demo
```

Architektur-Invarianten

1. **Hypothesen-Sprache**, niemals Behauptungen
2. **Provenance-Chain** für jedes Statement
3. **k≥5** für Schutzbedürftige (technisch erzwungen)
4. **Personen** nur in öffentlicher Rolle (PERSON_OFFICIAL)
5. **Rectification** statt Löschen (is_marked_disputed)
6. **AGPL-3.0**, federierbar, kein zentraler Träger
7. **Keine Kommerzialisierung**
8. **Minderjährige** nie individuell

Forken

Wer das forkt: - Adapter für andere Länder einbauen (Datenmodell ist generisch) - Eigene Pattern-Detektoren ergänzen - DB-Backend austauschen (z.B. PostgreSQL statt SQLite) - Bei Veränderung der Invarianten: anderen Namen wählen

Lizenz

AGPL-3.0. Keine Garantie. Wer einsetzt trägt Verantwortung für DSGVO, Persönlichkeitsrecht, korrekten Umgang mit Quellen.

2 • Hosting

Self-Host-Anleitung für eine Kataster-Instanz.

Voraussetzungen

- Linux-Server (Debian/Ubuntu reicht)
- Docker + Docker Compose
- Eigene Domain (für Federation; ohne Domain läuft es lokal)
- Optional: Reverse-Proxy mit TLS (Caddy / nginx-acme)

Quickstart

1. Repo klonen

```
git clone <your-fork-url> kataster
cd kataster/hosting
```

2. Frontend-Files in ./frontend ablegen

```
mkdir -p frontend
cp ../buerger-bundle/*.html frontend/
cp ../buerger-bundle/stimme-il8n.js frontend/
```

3. Backend-Code in ./backend ablegen

```
ln -s ../backend ./backend
```

4. Federation-Identity erzeugen (einmalig!)

```
docker compose run --rm api python3 -c "
from backend.federation.federation import InstanceIdentity
from pathlib import Path
i = InstanceIdentity.generate('kataster.MEINE-DOMAIN.example')
Path('/federation').mkdir(exist_ok=True)
i.to_secret_file(Path('/federation/identity.json'))
print('Public Key:', i.public_key.hex())
print('Fingerprint:', i.fingerprint)
"
```

5. Stack starten

```
docker compose up -d
```

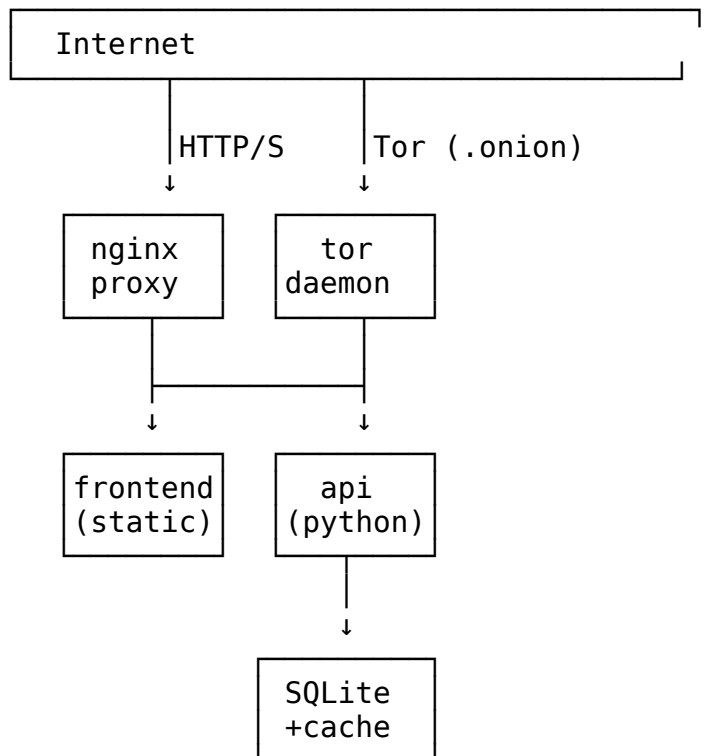
6. Initial-Crawl ausführen

```
docker compose --profile cron run --rm crawler
```

7. Tor-Onion-Adresse abfragen (nach ~30s warten)

```
docker compose exec tor cat /var/lib/tor/warte/hostname
```

Architektur



Federation-Setup

Andere Instanzen vertrauen — pro Peer:

```
docker compose run --rm api python3 -c "
from backend.federation.federation import FederationSubscriber,
TrustedPeer
```

```

from pathlib import Path
sub = FederationSubscriber(Path('/federation/peers.json'))
sub.trust_peer(TrustedPeer(
    instance_id='kataster.OTHER-INSTANCE.example',
    public_key='HEX_PUBLIC_KEY',
    fingerprint='SHORT_FINGERPRINT',
    trust_level=2,
))
"

```

Andere Instanzen pull-en dein Bundle alle paar Stunden via Cron:

```

# Crontab auf Peer-Instanz:
0 */4 * * * curl
https://kataster.MEINE-DOMAIN.example/federation/bundle.json | \
    docker compose run --rm api python3 -m
backend.federation.import_bundle

```

Kontroll-Endpoints

- <https://DEINE-DOMAIN/api/v1/health> — Server-Status
- <https://DEINE-DOMAIN/api/v1/stats> — globale Statistik
- <https://DEINE-DOMAIN/api/v1/entities> — Liste der Entities
- <https://DEINE-DOMAIN/api/v1/hypotheses> — Liste der Befunde
- <https://DEINE-DOMAIN/federation/bundle.json> — föderiertes Bundle
- <https://DEINE-DOMAIN/> — Bürger-Stack-Übersicht
- <https://DEINE-DOMAIN/dashboard.html> — Public Dashboard

Backup-Strategie

Datenpersistierung in 4 Volumes: - `api-data`: SQLite-DB + Cache + Manifests → täglich sichern - `tokens`: API-Tokens → einmalig sichern, dann kaum Änderungen - `federation-data`: Identity + Peers → einmalig sichern (Identity ist KRITISCH) - `tor-data`: onion-Adresse → einmalig sichern (sonst neue Adresse)

Empfehlung: tägliches `restic` oder `borg`-Backup auf separatem Storage.

Sicherheits-Hinweise

- API-Tokens niemals in Git committen
- Federation-Identity nicht teilen (das wäre wie Identitäts-Diebstahl)
- Tor-Hidden-Service-Keys nicht teilen
- Reverse-Proxy mit Rate-Limiting (in `nginx.conf` konfiguriert)
- Regelmäßige Container-Updates
- DSGVO: keine personenbezogenen Logs außerhalb der EU speichern

Ressourcen

- 1 GB RAM minimum, 2 GB empfohlen
- 5 GB Disk minimum (Crawl-Cache wächst!)
- 1 vCPU reicht für kleine bis mittlere Instanzen

Troubleshooting

- API antwortet nicht → `docker compose logs api`
- Tor-Adresse nicht da → 60s warten, dann `docker compose logs tor`
- Crawler scheitert → siehe Crawl-Manifest in `/data/cache/manifests/`
- DB-Migration → automatisch beim API-Start

3 • Bürger-Werkzeuge

Stand: Mai 2026 • AGPL-3.0

Was ist das

Browser-Werkzeuge, die Menschen direkt benutzen — ohne Account, ohne Tracking, ohne zentralen Server. Drei Ebenen je nach Schutzbedarf.

Die drei Ebenen

Ebene 1 • niedrigschwellig (für die Mehrheit, für Menschen die sonst nirgends gehört werden): - `stimme.html` — Etwas stimmt nicht. Sag es. - `stimme-karte.html` — Beobachtungen mit Ortsbezug - `recht.html` — KI-Rechtshilfe mit §-Zitaten - `akte.html` — Aktenführung mit Hash-Kette + Verschlüsselung

Ebene 2 • strukturiert (für rechtsförmliche Fallarbeit): - `evidentia.html` — 78 Fallkategorien, Sensitivity-Tiers

Ebene 3 • hochsicher (für Insider:innen mit Risiko): - `warte-client.html` — Tor-basiert, Shamir 3-of-5 Secret Sharing

Übersicht / Dispatcher: - `buerger-stack.html` — Entscheidungs-Hilfe welches Werkzeug
- `dashboard.html` — Public Dashboard zum Stöbern in Befunden

Schnellstart

Alle Werkzeuge sind statische HTML-Dateien — einfach im Browser öffnen. Kein Server, kein Build-Step.

```
tar xzf kataster-strang-B-buergertools.tar.gz
cd buergertools
# Im Browser öffnen:
firefox buerger-stack.html
```

Oder: alle Files auf einen Webserver legen (nginx, Apache, sogar GitHub Pages funktioniert). Pfade sind relativ — verlinken sich gegenseitig.

STIMME-Mehrsprachigkeit

`stimme-i18n.js` enthält Strings in 9 Sprachen: - `de` · `en` · `tr` · `bks` · `ar` · `fa` · `ro` · `uk` · `leichte` (Leichte Sprache)

In `stimme.html` eingebunden, die Sprachknöpfe rufen `applyTranslations(lang)` auf. Bisher tragen erst einzelne Elemente `data-i18n`-Marker. Für volle Mehrsprachigkeit: 1.

stimme-i18n.js ist eingebunden (erledigt) 2. Sprach-Buttons mit applyTranslations(lang) callbacken 3. Alle Texte mit data-i18n="key" markieren

AKTE: Verschlüsselung

AKTE speichert standardmäßig in IndexedDB unverschlüsselt (Browser-Schutz reicht für nicht-sensible Vorgänge).

Lock-Button setzt ein Passwort: - PBKDF2 200.000 Iterationen → 256-bit Key - AES-GCM für die Akten-Verschlüsselung - Passwort wird **nie** gespeichert. Vergisst du es → Akten weg. - Test-Blob im vault-Store erkennt falsches Passwort.

Datenfluss

Alles bleibt lokal. **Außer** wenn man explizit „weitergeben“ oder „zählen lassen“ wählt: - STIMME → optional an STIMME-Bridge (Strang A) → k-anonyme Aggregation - AKTE → Export-JSON, manuell weitergeben (kein Auto-Upload) - RECHT → keine Daten verlassen den Browser - Evidentia → manuelle Übergabe an NGOs / Anwält:innen - WARTE → Tor + Shamir → Empfänger-Zeremonie

Forken

Pfade in den HTML-Dateien sind relativ (href="stimme.html"). Wer einzelne Werkzeuge in eigenes Setup integriert, kann diese Pfade anpassen oder entfernen.

Das Gestaltungssystem der Werkzeuge ist eingebettetes CSS, keine externen Stylesheets. Atkinson-Hyperlegible-Schrift kommt von Google Fonts. Wer Offline-Betrieb will, muss die Schrift lokal hosten.

Barrierefreiheit (WCAG 2.1 AA)

- Atkinson Hyperlegible (für Sehbeeinträchtigung)
- Mindest-Fontgröße 18px (STIMME: 22px)
- Tap-Ziele $\geq 56 \times 56$ px
- Kontrast $\geq 4.5:1$ (Höchstkontrast-Modus zusätzlich)
- Skip-Links, semantisches HTML
- Vorlesen via Web Speech API
- Sprach-Aufnahme via MediaRecorder

Limitierungen

- STIMME: Sprach-Aufnahme bleibt aktuell rein lokal (Speech-to-Text fehlt)
- STIMME-Karte: Leaflet + OpenStreetMap; Kacheln und Geocoding laufen über öffentliche Server mit Fair-Use-Grenzen (für Produktivbetrieb eigene Server)
- RECHT: 30 Wissensbasis-Einträge mit Normzitaten; Links führen auf die RIS-Startseite, juristische Prüfung der Zitate offen
- AKTE: kein automatisches Backup (Browser löschen → Daten weg, außer vorher Export)

Lizenz & Haftung

AGPL-3.0. Keine Rechtsberatung im juristischen Sinn. Bei akuter Notlage: - Polizei 133 · Rettung 144 · Telefonseelsorge 142 - Frauen-Helpline 0800 222 555 · Rat auf Draht 147

4 • Manifeste

Die vier Manifeste liegen in Strang C je als DE- und EN-Fassung (HTML, direkt im Browser lesbar): `civic-stack`, `kataster-architecture`, `kataster-manifest`, `warte-manifest`. STATUS-Kopf und Autorenzeile sind in beiden Sprachen gleich; Projektnamen (Offener Kataster, WARTE, Evidentia, Offenes Werkzeug) bleiben unübersetzt.

Lizenz: Code AGPL-3.0-or-later · Text CC BY 4.0. Nicht Rechtsberatung.