

STATUS · Handbook · Offener Kataster · work state v3.1 · bundle v3.8 · not in production
FIGURES · all findings synthetic (demo data)
PARTNER · one-person workshop · independent · privately funded
DATE · 2026-09-23 · identifier FORSCH-PROV0445 (PROVISIONAL)
AUTHOR · Thomas Schüller · Syntagma Forschung · Vienna · ORCID 0009-0003-9799-6747

Offener Kataster — Handbook

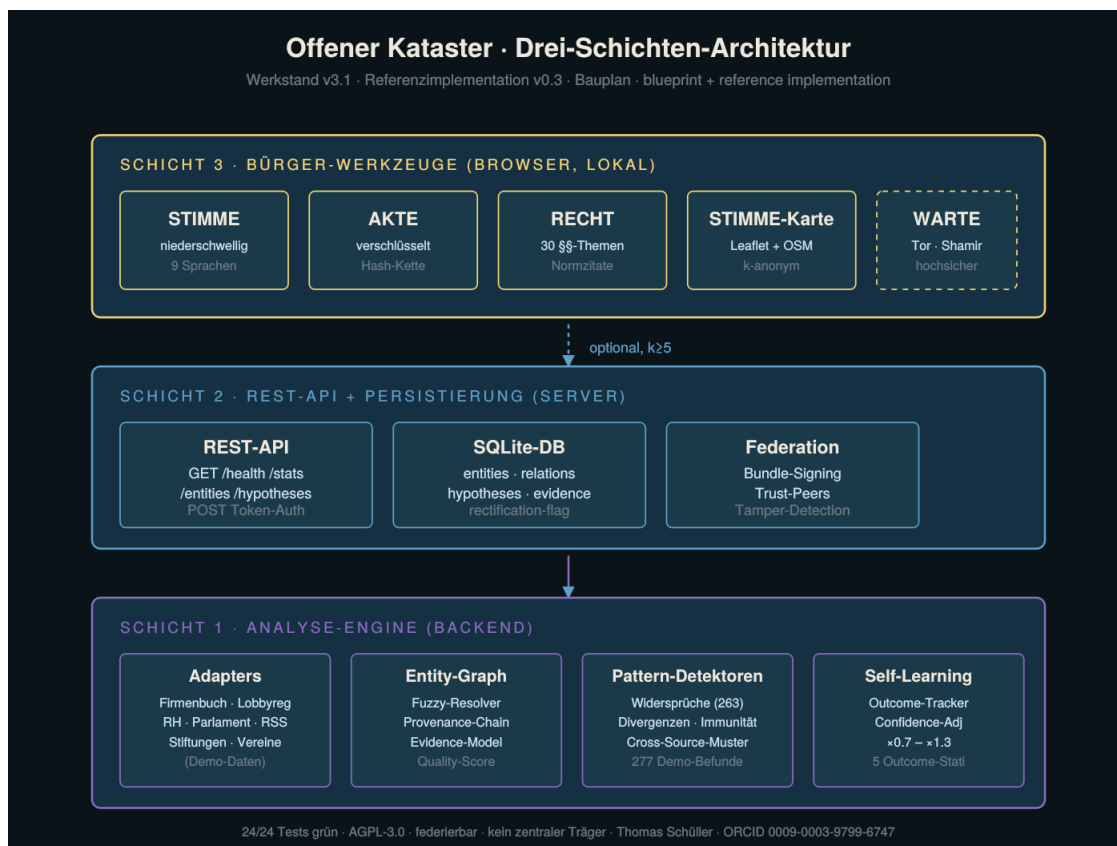


Figure 1: The three layers that the chapters of this handbook follow.

This handbook describes installation, start and operation of all three strands. The work description explains purpose, limits and invariants; this document is about operation.

Chapter	Strand	Archive
1	Backend (analysis engine, REST API, federation)	strand A
2	Hosting your own instance	strand A
3	Citizen tools in the	strand B

Chapter	Strand	Archive
4	browser Manifestos	strand C

1 • Backend

As of: May 2026 · AGPL-3.0 · v0.3

What this is

The analysis engine of Offener Kataster. Python, runs server-side. Crawls public sources, builds an entity graph with quality scoring, produces hypothesis findings, persists to SQLite, offers a REST API, federates with other forks.

Contents

backend/	
├── core/	
│ ├── graph.py	Entity graph with quality scoring
│ ├── entity_resolver.py	Fuzzy match
│ (Levenshtein/Jaccard/substring)	
│ ├── db_layer.py	SQLite persistence
├── ingestion/	
│ ├── adapters.py	9 adapters (company register,
lobby reg, ...)	
│ ├── stimme_bridge.py	STIMME→graph with k-anonymity
│ ├── foundations_adapters.py	Foundations, NGOs, party-
affiliated associations	
│ ├── realtime_adapters.py	Parliament, ministries, Court of
Audit	
├── graph/	
│ ├── cross_source_synthesizer.py	5 pattern detectors
├── modules/	
│ ├── contradictions.py	contradiction engine (demo: 263
pairs from 73 catalogue rules)	
│ ├── regulation_catalog.py	Rule library
│ ├── precedent_divergence.py	OGH/VfGH ↔ CJEU/ECtHR
│ ├── mortality_anomaly.py	Mortality anomalies
│ ├── immunity_tracker.py	Extradition proceedings
│ ├── vulnerable.py	k≥5 for protection-worthy groups
│ ├── institution_atlas.py	41 Austrian institutions
│ ├── sozialsystem.py	AMS / BFA / MA40 tracker
│ ├── outcome_tracker.py	Self-learning loop
├── api/	
│ ├── rest_api.py	HTTP API with token auth (read
open)	
├── federation/	
│ ├── federation.py	Bundle signing, cross-fork sharing
├── crawlers/	
│ ├── gemeinde_protokoll.py	Municipal-council resolutions

```
└─ tests/
  └─ test_core.py          24 unittest tests
  └─ run_pipeline.py      End-to-end pipeline
```

Quick start

```
tar xzf kataster-strang-A-backend.tar.gz
cd backend
python3 run_pipeline.py          # produces 277 demo findings
python3 tests/test_core.py      # 24 tests green
```

Start the REST API

```
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()
"
```

Then reachable: - GET /api/v1/health — status - GET /api/v1/stats — global statistics - GET /api/v1/entities?type=ORGANIZATION&limit=100 — lists - GET /api/v1/entities/{id} — single entry with relations - GET /api/v1/hypotheses — findings - POST /api/v1/dispute — mark an entry as disputed (token auth)

Federation

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

Sender:

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

Recipient:

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

Run modules individually

All modules have `__main__` demos:

```
python3 ingestion/stimme_bridge.py      # k-anonymity
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      # institutions
python3 federation/federation.py          # federation with demo
```

Architecture invariants

1. **Hypothesis language**, never assertions
2. **Provenance chain** for every statement
3. **k≥5** for protection-worthy groups (technically enforced)
4. **Persons** only in a public role (PERSON_OFFICIAL)
5. **Rectification** instead of deletion (is_marked_disputed)
6. **AGPL-3.0**, federatable, no central operator
7. **No commercialization**
8. **Minors** never individually

Forking

Anyone forking this: - Add adapters for other countries (the data model is generic) - Add your own pattern detectors - Swap the DB backend (e.g. PostgreSQL instead of SQLite) - If you change the invariants: choose a different name

License

AGPL-3.0. No warranty. Whoever deploys bears responsibility for GDPR, personality rights, correct handling of sources.

2 • Hosting

Self-hosting guide for a Kataster instance.

Prerequisites

- Linux server (Debian/Ubuntu is enough)
- Docker + Docker Compose
- Your own domain (for federation; without a domain it runs locally)
- Optional: reverse proxy with TLS (Caddy / nginx-acme)

Quick start

1. Clone the repo

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

2. Place frontend files in ./frontend

```
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
```

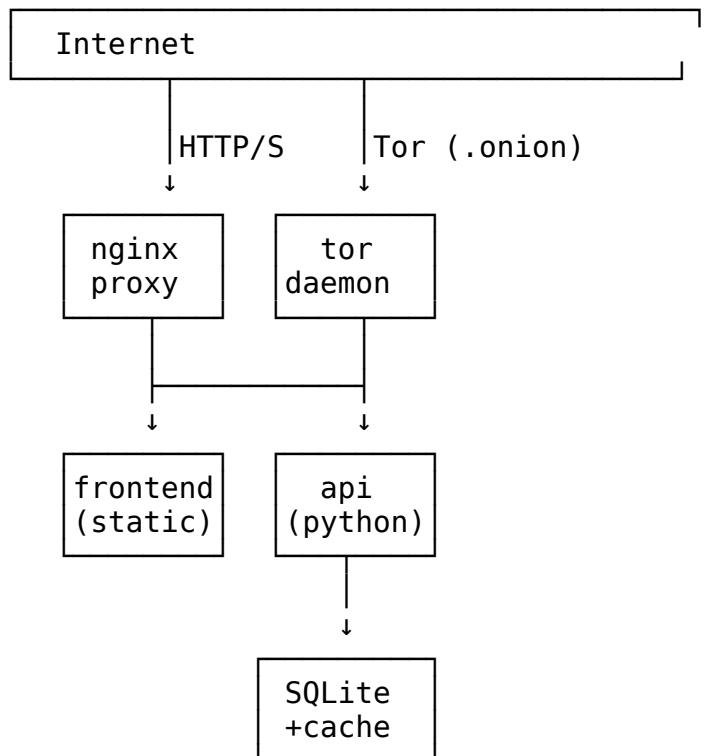
6. Initial-Crawl ausführen

```
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
```

Architecture



Federation-Setup

Trust other instances — per 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,
))
"

```

Other instances pull your bundle every few hours 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

Data persistence across 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)

Recommendation: daily `restic` or `borg` backup on separate storage.

Securitys-Hinweise

- API-Tokens niemals in Git committen
- Do not share the federation identity (that would be like identity theft)
- Do not share Tor hidden-service keys
- Reverse-Proxy mit Rate-Limiting (in `nginx.conf` konfiguriert)
- Regelmäßige Container-Updates
- GDPR: do not store personal logs outside the EU

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 not responding → `docker compose logs api`
- Tor address missing → wait 60s, then `docker compose logs tor`
- Crawler scheitert → siehe Crawl-Manifest in `/data/cache/manifests/`
- DB-Migration → automatisch beim API-Start

3 • Citizen tools

As of: May 2026 · AGPL-3.0

What this is

Browser tools that people use directly — no account, no tracking, no central server. Three levels depending on protection needs.

The three levels

Level 1 • low-threshold (for the majority, for people who otherwise go unheard): - `stimme.html` — Something is wrong. Say it. - `stimme-karte.html` — Observations with a location reference - `recht.html` — AI legal help with statute citations - `akte.html` — Case-file keeping with hash chain + encryption

Level 2 • structured (for formal legal casework): - `evidentia.html` — 78 case categories, sensitivity tiers

Level 3 • high-security (for insiders at risk): - `warte-client.html` — Tor-based, Shamir 3-of-5 secret sharing

Overview / dispatcher: - `buerger-stack.html` — Decision aid: which tool - `dashboard.html` — Public dashboard for browsing findings

These are voice tools for self-advocacy by those directly affected — not informant tools for third-party surveillance of others.

Quick start

All tools are static HTML files — just open in the browser. No server, no build step.

```
tar xzf kataster-strang-B-buergertools.tar.gz
cd buergertools
# open in the browser:
firefox buerger-stack.html
```

Or: put all files on a web server (nginx, Apache, even GitHub Pages works). Paths are relative — they link to each other.

STIMME multilingualism

`stimme-i18n.js` contains strings in 9 languages: - `de` · `en` · `tr` · `bks` · `ar` · `fa` · `ro` · `uk` · `leichte` (Easy Language)

Included in `stimme.html`; the language buttons call `applyTranslations(lang)`. So far only some elements carry `data-il8n` markers. For full multilingualism: 1. `stimme-il8n.js` is included (done) 2. Wire language buttons to `applyTranslations(lang)` 3. Mark all texts with `data-il8n="key"`

AKTE: encryption

AKTE stores in IndexedDB unencrypted by default (browser protection suffices for non-sensitive matters).

The lock button sets a password: - PBKDF2 200,000 iterations → 256-bit key - AES-GCM for case-file encryption - The password is **never** stored. Forget it → files gone. - A test blob in the vault store detects a wrong password.

Data flow

Everything stays local. **Except** when you explicitly choose to “pass on” or “have counted”: - STIMME → optionally to the STIMME bridge (strand A) → k-anonymous aggregation - AKTE → export JSON, pass on manually (no auto-upload) - RECHT → no data leaves the browser - Evidentia → manual handover to NGOs / lawyers - WARTE → Tor + Shamir → recipient ceremony

Forking

Paths in the HTML files are relative (`href="stimme.html"`). Anyone integrating individual tools into their own setup can adjust or remove these paths.

The design system of the tools is inline CSS, no external stylesheets. The Atkinson Hyperlegible font comes from Google Fonts. For offline operation, host the font locally.

Accessibility (WCAG 2.1 AA)

- Atkinson Hyperlegible (for visual impairment)
- Minimum font size 18px (STIMME: 22px)
- Tap targets $\geq 56 \times 56$ px
- Contrast $\geq 4.5:1$ (plus a maximum-contrast mode)
- Skip links, semantic HTML
- Read-aloud via Web Speech API
- Voice recording via MediaRecorder

Limitations

- STIMME: voice recording is currently purely local (speech-to-text missing)
- STIMME map: Leaflet + OpenStreetMap; tiles and geocoding run on public servers with fair-use limits (own servers for production)
- RECHT: 30 knowledge-base entries with statute citations; links lead to the RIS home page, legal review of the citations pending
- AKTE: no automatic backup (clearing the browser → data gone, unless exported beforehand)

License & liability

AGPL-3.0. Not legal advice in the juristic sense. In an acute emergency (Austria): - Police 133 · Ambulance 144 · Crisis line 142 - Women's helpline 0800 222 555 · Rat auf Draht 147

4 • Manifestos

The four manifestos are in strand C, each as DE and EN version (HTML, readable directly in the browser): `civic-stack`, `kataster-architecture`, `kataster-manifest`, `warte-manifest`. STATUS header and author line are identical in both languages; project names (Offener Kataster, WARTE, Evidentia, Offenes Werkzeug) stay untranslated.

Licence: code AGPL-3.0-or-later · text CC BY 4.0. Not legal advice.