

STATUS · Method protocol · API and strand separation · v0.4 · ready for publication
FIGURES · no empirical figures; templates without real keys
PARTNER · one-person workshop · independent · privately funded
DATE · 2026-09-23 · identifier FORSCH-PROV0446 (PROVISIONAL)
AUTHOR · Thomas Schüller · Syntagma Forschung · Vienna · ORCID 0009-0003-9799-6747

API and strand separation — .env templates and checking tool

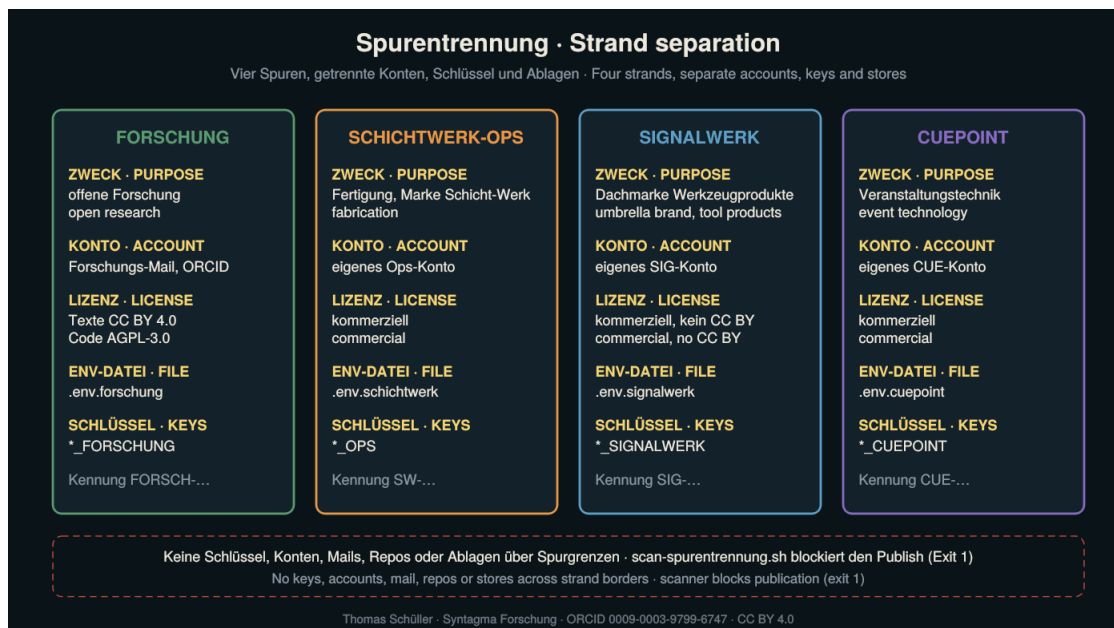


Figure 1: Four strands with separate accounts, keys, licences and stores. The checking tool blocks publication as soon as traces of a foreign strand appear in the bundle.

Basic rule: per strand one .env file, one account, its own keys, its own repositories. Never a key of one strand in the file of another. This edition deliberately contains no operational data of the commercial strands: addresses, mail accounts and providers are placeholders.

The four strands

Strand	Purpose	Licence of results	Identifier series	Key suffix
FORSCHUNG	open research (Syntagma Forschung)	texts CC BY 4.0, code AGPL-3.0	FORSCH-	_FORSCHUNG
SCHICHTWERK-OPS	commercial fabrication, brand Schicht-Werk	commercial	SW-	_OPS
SIGNALWERK	umbrella brand	commercial, no	SIG-	_SIGNALWERK

Strand	Purpose	Licence of results	Identifier series	Key suffix
CUEPOINT	of the tool products event technology	CC BY on product code commercial	CUE-	_CUEPOINT

Directory

~/secrets/

```
├── .env.forschung
├── .env.schichtwerk
├── .env.signalwerk
├── .env.cuepoint
└── README.md
```

All four files: `chmod 600`, outside every repository, never committed.

Template .env.forschung

```
TRACK=forschung
AUTHOR_NAME="Thomas Schüller"
AUTHOR_ORCID="0009-0003-9799-6747"
AUTHOR_MAIL="forschung@tschueller.com"
AUTHOR_BRAND="Syntagma Forschung · Wien"
ZENODO_TOKEN_FORSCHUNG=
ORCID_CLIENT_ID_FORSCHUNG=
ORCID_CLIENT_SECRET_FORSCHUNG=
ANTHROPIC_API_KEY_FORSCHUNG=
HOSTING_KEY_FORSCHUNG=
GIT_TOKEN_FORSCHUNG=
```

Template .env.schichtwerk

```
TRACK=schichtwerk
COMPANY_MAIL=<ops account>
COMPANY_DOMAIN=<ops domain>
HOSTING_HOST_OPS=
HOSTING_USER_OPS=
HOSTING_PASS_OPS=
SHOP_API_TOKEN_OPS=
ANTHROPIC_API_KEY_OPS=
GIT_TOKEN_OPS=
```

Template .env.signalwerk

```
TRACK=signalwerk
BRAND_MAIL=<SIG account>
BRAND_DOMAIN=<SIG domain>
LICENSE_SERVER_KEY_SIGNALWERK=
STORE_API_TOKEN_SIGNALWERK=
```

```
ANTHROPIC_API_KEY_SIGNALWERK=  
GIT_TOKEN_SIGNALWERK=
```

Template .env.cuepoint

```
TRACK=cuepoint  
BRAND_MAIL=<CUE account>  
BRAND_DOMAIN=<CUE domain>  
HOSTING_KEY_CUEPOINT=  
ANTHROPIC_API_KEY_CUEPOINT=  
GIT_TOKEN_CUEPOINT=
```

Loading mechanism (Python)

```
from pathlib import Path  
  
SPUREN = {"forschung": "_FORSCHUNG", "schichtwerk": "_OPS",  
          "signalwerk": "_SIGNALWERK", "cuepoint": "_CUEPOINT"}  
  
def load_track_env(track: str) -> dict:  
    assert track in SPUREN, f"unknown strand: {track}"  
    env_path = Path.home() / "secrets" / f".env.{track}"  
    mode = env_path.stat().st_mode & 0o777  
    assert mode == 0o600, f"{env_path} must be chmod 600"  
    out = {}  
    for line in env_path.read_text().splitlines():  
        line = line.strip()  
        if not line or line.startswith("#") or "=" not in line:  
            continue  
        key, _, val = line.partition("=")  
        out[key.strip()] = val.strip().strip('\"')  
    suffix = SPUREN[track]  
    foreign = [k for k in out if k.endswith(tuple(s for s in  
SPUREN.values() if s != suffix))]  
    assert not foreign, f"foreign keys in .env.{track}: {foreign}"  
    return out
```

The loader checks file permissions and rejects keys carrying the suffix of another strand.

Browser tools (BYOK)

Browser tools (STIMME, AKTE, RECHT) use no .env. Users enter their own key locally:

```
const loadBYOK = (tool, provider) => localStorage.getItem(`$  
{tool}_byok_${provider}`);  
const saveBYOK = (tool, provider, key) => localStorage.setItem(`$  
{tool}_byok_${provider}`, key);  
const clearBYOK = (tool, provider) => localStorage.removeItem(`$  
{tool}_byok_${provider}`);
```

The key goes to no server of ours, only directly to the chosen provider.

Checking tool `scan-spurentrennung.sh` (neutral edition)

The forbidden patterns are not in the script but in text files in the folder `muster/`: one file per strand (`forschung.txt`, `ops.txt`, `signalwerk.txt`, `cuepoint.txt`) and `alle.txt` for patterns forbidden in every strand (common key formats). The neutral edition with example values (`*.example.invalid`) is published. Each organisation enters the contacts, domains and accounts of its own strands there; these real patterns stay internal.

```
./scan-spurentrennung.sh <directory>
# strand forschung, patterns ./muster
./scan-spurentrennung.sh <directory> --track=cuepoint
./scan-spurentrennung.sh <directory>
--muster=/path/to/internal/patterns
```

Exit	Meaning
0	clean
1	hit, publication blocked
2	usage error (directory, pattern folder or pattern file missing)

Intentional, documented mentions carry the marker `SPUREN-DOC`. Format conversion (for instance MD – HTML with `pandoc`) strips HTML comments; the scan is therefore re-run over all output formats after every conversion. The test suite `test-scan-spurentrennung.sh` checks seven cases with its own test patterns: clean, leak of another strand, marker, leak of a fourth strand, key, missing directory, missing pattern file.

Checklist before publication

1. scan clean over all formats
2. no `.env` file in the bundle
3. no keys in code
4. no operational data of commercial strands in research material
5. strand named in the STATUS header and in the handover

Key rotation

Strand	Interval
FORSCHUNG	90 days
SCHICHTWERK-OPS	60 days
SIGNALWERK	60 days
CUEPOINT	60 days

When switching strands in the browser, use a different browser profile, not the same tab.

Licence: text CC BY 4.0 · scripts AGPL-3.0-or-later. Closure type `methoden-protokoll`.