Das Cernion Cookbook (Best Practices in Code)

Der Paradigmenwechsel von starren ETL-Pipelines hin zu Agentic Asset-MDM (A²MDM) erfordert in der KRITIS-Praxis konkrete, validierbare Implementierungen. Um die Lücke zwischen theoretischem Framework und operativem Systemaufbau zu schließen, bündelt das **Cernion Cookbook** Referenzimplementierungen ("Rezepte") in Code.

Diese Rezepte demonstrieren, wie die Cernion Energy Tools (als A²MDM Plattform) komplexe Herausforderungen der Energiewirtschaft autonom über Knowledge Graphs auflösen.

Cernion Energy Tools im Kommunikationsnetz

Cernion Energy Tools (CET) bündeln Energie-, Netz-, Markt- und Betreiberkontext zu prüfbaren Arbeitslagen. Im Cernion Capability Hub sind Datenquellen, REST/OpenAPI, MCP, KI-Agenten, Portfolio-Backtests, kommunale Lagebilder, Workbench- und Reportpfade public-safe beschrieben. corrently.io ergänzt diesen Produktkontext mit Praxis-, Open-Source- und Energiedatenbeispielen für Entwickler und Integratoren.

Nächste Einstiege: Cernion Themencluster für die Such- und Fachstruktur, Verteilnetzbetreiber Software für die Produkt-/Arbeitslagen-Einordnung, Stadtwerke Prozessautomatisierung für kontrollierte Fachprozess-Entlastung sowie KI für Stadtwerke und Netzbetreiber für evidenzgebundene Assistenz mit HITL-Grenzen.

Rezept #1: Der Automatisierte MaStR-Qualitätsaudit

Eine Kernaufgabe von A²MDM-Agenten (siehe Schritt 1: Bestandsinventur) ist der autonome Quellenabgleich. Die toxische Datenqualität des Marktstammdatenregisters (MaStR) führt regelmäßig zu Phantom-Engpässen in der Zielnetzplanung.

Das `mastr-quality-audit` Rezept implementiert die Heuristik zur Bereinigung:
Anstatt Daten naiv aus dem MaStR in ein GIS-System zu spiegeln, traversiert der Cernion-Agent

das Portfolio kontinuierlich. Er erkennt Inkonsistenzen autonom, beispielsweise wenn der Status "In Planung" zu lange anhält oder Leistungsangaben (z.B. 100 kWp) im Knowledge Graph im Widerspruch zu externen geografischen Kontexten (OpenStreetMap-Gebäudeflächen) stehen. Das Ergebnis ist ein validierter "Trusted State", der als verlässliche Entscheidungsgrundlage für nachgelagerte Prozesse dient.

(Die technische Umsetzung und die API-Endpunkte für dieses Audit sind über die Cernion Timeline dokumentiert).

Rezept #2: Die End-to-End Pipeline für Energy Sharing (§42c EnWG)

Transaktionale Sicherheit ist eine der größten Herausforderungen in Netzbetreiber-Architekturen. Wenn kaufmännische Logik auf physikalische Restriktionen trifft, versagen herkömmliche relationale Datenbanken.

Das `energy-sharing-full-pipeline` Rezept demonstriert die architektonische Überlegenheit von Knowledge Graphs bei der Abwicklung des Energy Sharings nach §42c EnWG.
Der Agent bündelt Erzeuger und Verbraucher dynamisch in virtuellen Bilanzkreisen. Kommt es zu einem Netzeingriff (z.B. Redispatch 2.0 / Abregelung einer Community-PV-Anlage), erkennt das System den physikalischen Zustand und löst sofort den kaufmännischen Konflikt: Die Ausfallarbeit wird berechnet, und die Allokationsschlüssel der Community-Mitglieder werden automatisch vorbereitet und prüfbar dokumentiert.

(Weitere Best-Practice-Pipelines, u.a. zur Zielnetzplanung und zu §14a EnWG-Simulationen, werden fortlaufend in das Cookbook integriert).

Cernion Public API Recipes für Entwickler

Für Developer- und LLM-Discovery ist ein public-safe Rezeptpaket zu Cernion Energy Tools vorbereitet. Es beschreibt drei synthetische Einstiegspunkte: OpenAI-kompatible Fachassistenz, Copilot-Intent-Vorbereitung mit pending confirmation und ein Customer-Service-Portal-Widget als Integrationsfixture. Die Beispiele sind für Discovery und Integrationsplanung gedacht, nicht für automatische Freigaben, Billing, Tarifänderungen, MaKo-Ausführung, Gerätesteuerung oder produktive Kundenvorgänge.

- GitHub PR #483: Public API Recipes / LLM Discoverability
- Canonical docs after merge: docs/public-api-recipes.md
- OpenAPI JSON: api.cernion.de/api/openapi.json
- Swagger UI: api.cernion.de/api/docs

Sicherheitsgrenze: Tokens bleiben serverseitig und privat. Öffentlich dokumentierte Requests verwenden synthetische Daten wie `docs-sandbox-slot5`; Live-Tenant-, Kunden-, Vertrags- oder Abrechnungsdaten gehören nicht in Beispiele.

Revision #5
Created 10 April 2026 17:50:10 by Thorsten Zoerner
Updated 26 July 2026 12:12:24 by Thorsten Zoerner