

Verwertbarkeit der eGoⁿ Projektergebnisse

Clara Büttner, Ilka Cußmann, Stephan Günther

Hochschule Flensburg, Otto-von-Guericke-Universität Magdeburg

01. Oktober 2020



Teilnehmende der Session D2



Gliederung



- 1 Zentrale Fragestellungen
- 2 Ergebnisse Vorab-Umfrage
- 3 Veröffentlichungen und Dokumentationen der Projektergebnisse
- 4 Die open_eGo Toolchain
- 5 Datenerstellung
- 6 Vorstellung OEP
- 7 Vorstellung eTraGo

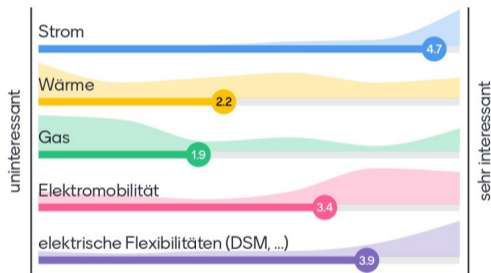


- Projekte eGoⁿ und open_BEA verfolgen Open Data und Open Source Prinzipien
- Unabhängig davon bestehen Einstiegshürden zur Nutzung von Daten und Tools für Externe
- Identifikation und Senkung dieser Einstiegshürden ist gewünscht, um die Projektergebnisse für eine breitere Nutzercommunity verfügbar zu machen
- Interessenschwerpunkte der Nutzercommunity identifizieren

Ergebnisse aus der Vorab-Umfrage



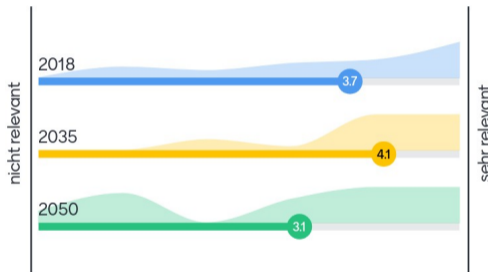
Für Daten aus welchen Bereichen interessieren Sie sich?



Ergebnisse aus der Vorab-Umfrage



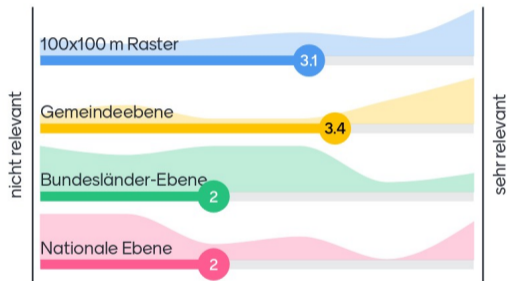
Welche Datenjahre sind für Sie am relevantesten?



Ergebnisse aus der Vorab-Umfrage



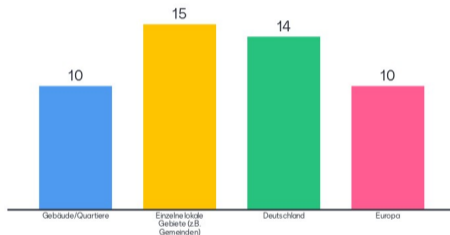
In welcher geographischen Auflösung sind Daten für Sie von Relevanz?



Ergebnisse aus der Vorab-Umfrage



Welchen geographischen Fokus haben Sie in Ihrer Arbeit?





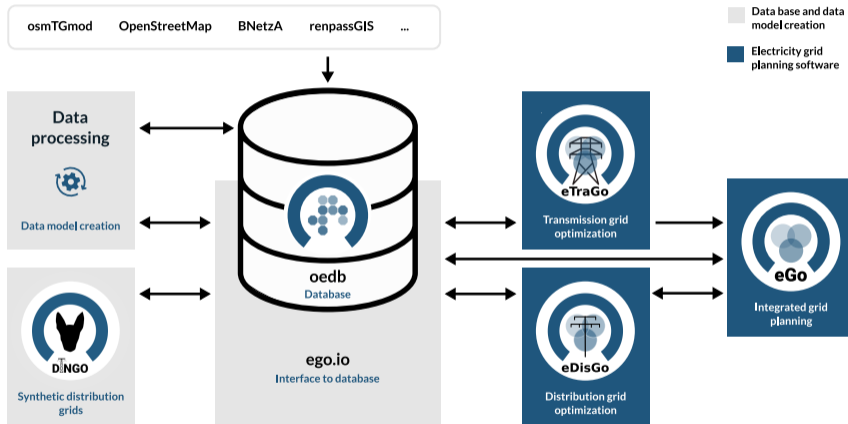
eGoⁿ

- Projekthomepage:
<https://ego-n.org/>
 - allgemeine Projektinformationen
 - Veröffentlichungen
- Github
<https://github.com/openego/>
- Read the Docs
- Open Energy Platform <https://openenergy-platform.org/>
 - Datensätze
 - Metadaten

open_BEA

- Projekthomepage:
<https://openbeaproject.wordpress.com/>
 - allgemeine Projektinformationen
 - Veröffentlichungen
- Github/Gitlab
 - eDisGo <https://github.com/openego/eDisGo>
 - simSES <https://gitlab.lrz.de/open-ees-ses/simses>
- Read the Docs

Die open_eGo Toolchain

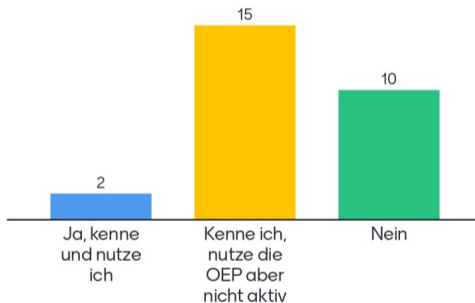


- Inputdaten für Tools wurden in open_eGo im Rahmen des data processings erstellt
- Sammlung verschiedener Skripte (SQL, Python) und einem ausführenden Skript
- Dokumentation des Data processing und angewandter Methoden über readthedocs
- Erweiterung, Umgestaltung und Umbenennung des data processings im Rahmen des aktuellen Projekts
 - Erweiterung des Datenmodelle um zusätzliche Sektoren
 - Nutzung Apache Airflow zum Workflow-Management
 - Nur (Zwischen-)Ergebnisse werden auf OEP gespeichert
 - Ziel: Bessere Nutz- und Nachvollziehbarkeit (einzelner Teile) der Datenerstellung

Ergebnisse aus der Vorab-Umfrage



Kennen und nutzen Sie die Open Energy Platform?
(<https://openenergy-platform.org>)



Vorstellung der Open Energy Platform

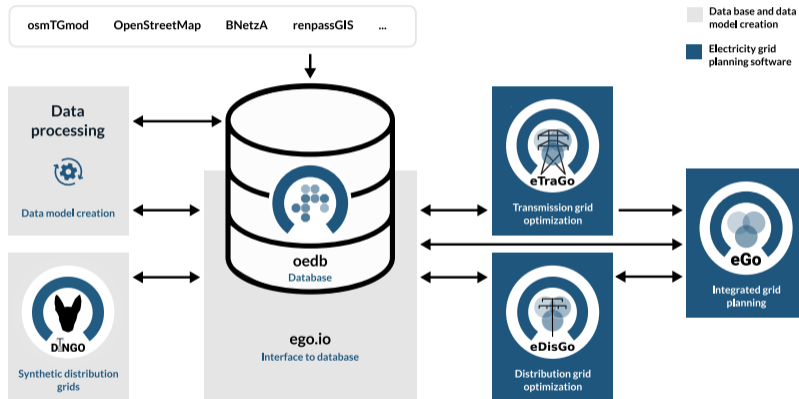
<https://openenergy-platform.org/>



Was ist die OEP?



1. Das was hier fehlt: ein Frontend für die oedb



Was ist die OEP?



2. Erreichbar unter <https://openenergy-platform.org/>
3. Quelloffen: <https://github.com/OpenEnergyPlatform/oeplatform>
4. Ein HTTP API Endpunkt
5. Der Versuch die Daten in der oedb durch Zusatzinformationen miteinander zu Vernetzen und leichter zugänglich zu machen

- Factsheets: Model, Framework, Scenario
- Die Open Energy Ontology (OEO)
 - hat ihre eigene Sektion auf der OEP.
 - wird quelloffen unter <https://github.com/OpenEnergyPlatform/ontology> entwickelt.
- Metadaten: Open Energy Metadata (OEM)
 - “standardisiertes”, beschreibendes JSON Objekt für jede Tabelle
 - ebenfalls quelloffen unter <https://github.com/OpenEnergyPlatform/oemetadata> entwickelt,
 - unter `metadata/latest/metadata_key_description.md` dokumentiert sowie mit einer Schablone und Beispielen versehen
 - programmatische Anwendung in Python unterstützt durch `omi` und `oem2orm`

Zeit die Demogötter zu Erzürenen



Die API der OEP



Kann mit der Programmiersprache Ihrer Wahl, hier Python, per HTTP angesprochen werden:

```
import requests
response = requests.get(
    "https://openenergy-platform.org/api/v0/" +
    "{schema}/{table}/rows/?{format}{order}".format(
        schema="schema/demand",
        table="tables/ego_demand_federalstate",
        format="form=csv",
        order=(
            "&orderby=elec_consumption_households"
            "&orderby=federal_states")))
```

Komplexere Abfragen, z.B. mit JOINS oder Datenänderungen, via POST möglich:

- Abfrage wird über dokumentiertes JSON Payload spezifiziert
- Authorisierungstoken wird über HTTP Header gesendet

Alternative: SQLAlchemy mit oedialect

- zwei Python Pakete
- SQLAlchemy: Objektrelationaler Mapper (ORM), der SQL Komfortabel in Python einbettet
- oedialect: Zusatzpaket, welches SQLAlchemy Konstrukte transparent in OEP HTTP API Aufrufe übersetzt



SQLAlchemy plus oedialect Beispiel

```
import oedialect
import sqlalchemy as sql
import sqlalchemy.orm as orm
engine = sql.create_engine(
    "postgresql+oedialect://openenergy-platform.org"
)
database = sql.MetaData(bind=engine)
database.reflect(schema="demand")
session = orm.sessionmaker(bind=engine)()
results = session.query(
    database.tables["demand.ego_demand_federalstate"]
).order_by(
    "elec_consumption_households", "federal_states"
).all()
```

SQLAlchemy plus oedialect Beispiel



Nach Ausführung des Codes auf der letzten Folie enthält `results` folgende Liste:

```
[('DE500', 'Bremen', 993.0, 2370.0, 1213.0, ...),  
 ('DEC00', 'Saarland', 1667.0, 4339.0, 2037.0, ...),  
 ...  
]
```

Zukünftige Features



- Verknüpfen der Daten mit den Szenariofactsheets
- Verknüpfen der Szenariofactsheets mit den Modelfactsheets
- Vernetzen der Daten und deren Metadaten über die Ontologie
- Wizard zur Erstellung von Tabellen über die Plattform

Diskussion zu Daten und Open Energy Platform



- Gibt es Erfahrungen mit der OEP in der Gruppe?
 - Was läuft gut, was schlecht?
 - Gibt es Verbesserungsvorschläge?
 - Gibt es Erfahrungen mit der API in der Gruppe?
 - Wird die Existenz der API als sinnvoll erachtet? Kritik?
- Wie kann die Datenverwertbarkeit verbessert werden?
 - Wie hilfreich sind die auf der OEP bereitgestellten Metadaten?
 - Wäre eine über Metadaten und Dokumentation der Datenerstellung hinausgehende Beschreibung von Datensätzen hilfreich?
 - Ansätze und Beispiele für gute Dokumentationen von Datensätzen

Verbesserungsvorschläge von Nutzern zu Daten/OEP

- Map-Ansicht war oft leer, wäre aber wichtig zur schnellen Einschätzung der Relevanz der Daten, alternativ Screenshot
- Informationen erschlagen: mehr/einfacher auffindbare Möglichkeiten zur Suche
- Leere Resultate bei Filtern durch Tags vermeiden
- Zwischenergebnisse von Endergebnissen besser unterscheidbar machen
- Datensatzbeschreibung präsender machen

Vorstellung von eTraGo





- Gibt es Erfahrungen mit den eGo-Tools in der Gruppe?
- Sehen Sie in Ihrer Arbeit Anwendungsbereich und können Sie sich vorstellen die Tools zu verwenden?
- Wie können die Einstiegshürden zur Nutzung unserer Tools am besten gesenkt werden? (z.B. Minimalbeispiele, Support, Grafische Oberfläche...)
- Bietet die Veröffentlichung von Ergebnisdatensätzen zusätzlich zu Inputdaten und Tools einen signifikanten Mehrwert?

Tools

- iplot bzw. Visualisierung gut
- Empfehlungen für Computerausstattungen (CPU, RAM)
- Beispiele/Input- u. Outputdaten für definierte Fälle herunterladbar machen
- Beispiel für gut dokumentierte/präsentierte Daten: SciGRID_gas

Kontakt Daten



Ilka Cußmann
Hochschule Flensburg
+49 (0)461 805 3017
ilka.cussmann@hs-flensburg.de

Clara Büttner
Hochschule Flensburg
+49 (0)461 805 3008
clara.buettner@hs-flensburg.de

Stephan Günther
Otto-von-Guericke-Universität Magdeburg
+49 (0)391 67 51358
stephan.guenther@ovgu.de