



Strom, Wärme und Mobilität

oemof – a community project to make energy modelling transparent and shareable

Uwe Krien, Birgit Schachler, Berit Müller

4. Mai 2017, Strommarktreffen

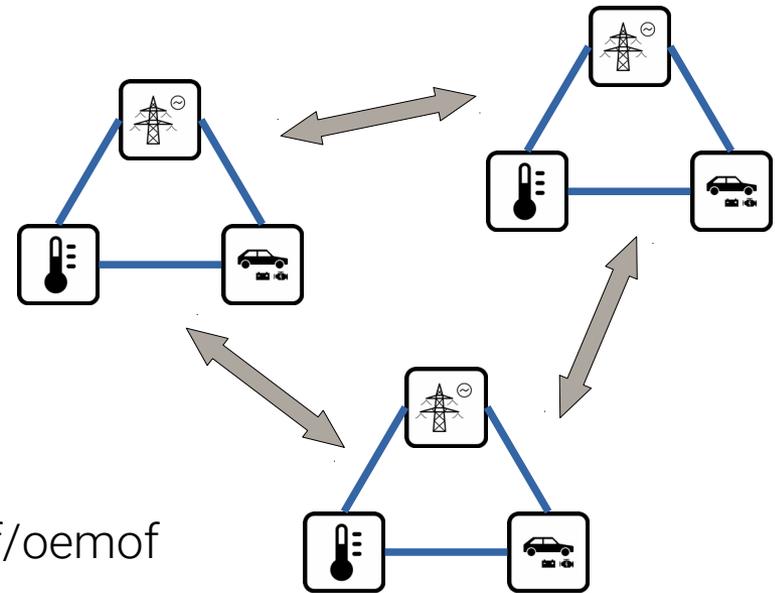
Agenda

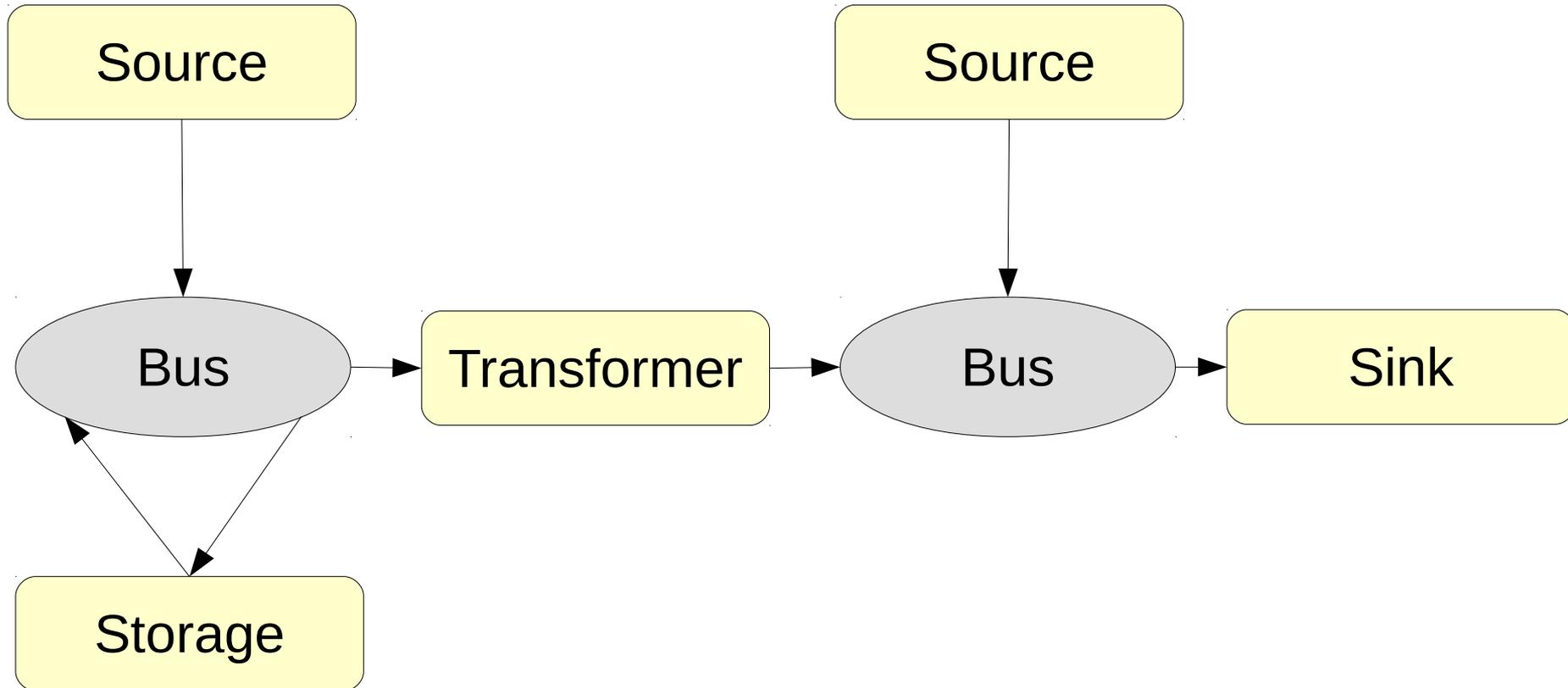
- oemof – Framework und Applikation
- Vorhandene Strom-Wärme-Modelle
- Wärmekomponenten in oemof
- Erweiterungen und Ausblick

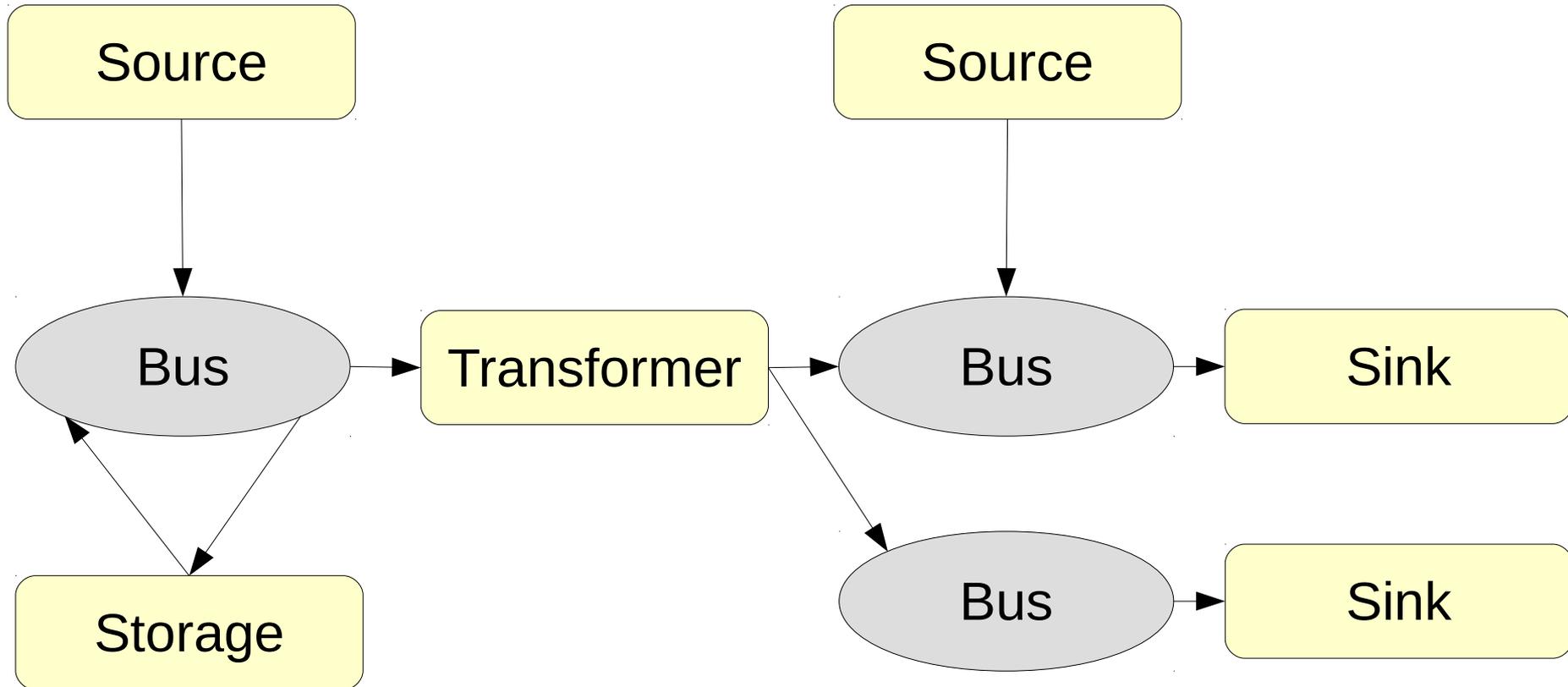


- oemof steht für **o**pen **e**nergy **m**odelling **f**ramework
- Open Source Tool zur Modellierung und Optimierung von Energiesystemen
- Hauptsächliche Entwicklung durch
 - Zentrum für Nachhaltige Energiesysteme (ZNES), Flensburg
 - Reiner Lemoine Institut, Berlin
 - Otto-von-Guericke-Universität, Magdeburg

- Grundlegendes Konzept – Graphentheorie
 - Knoten stellen Busse (Bilanzräume) oder Komponenten dar
 - Kanten sind gerichtete Flüsse
- Generische Implementierung ermöglicht
 - Sektorkopplung
 - Kopplung multipler Regionen
 - Flexible Zeitschrittweite
 - Dispatch / Investment
- Weitere Informationen
 - Homepage: oemof.org
 - Source Code: <https://github.com/oemof/oemof>
 - Dokumentation: <http://oemof.readthedocs.org>

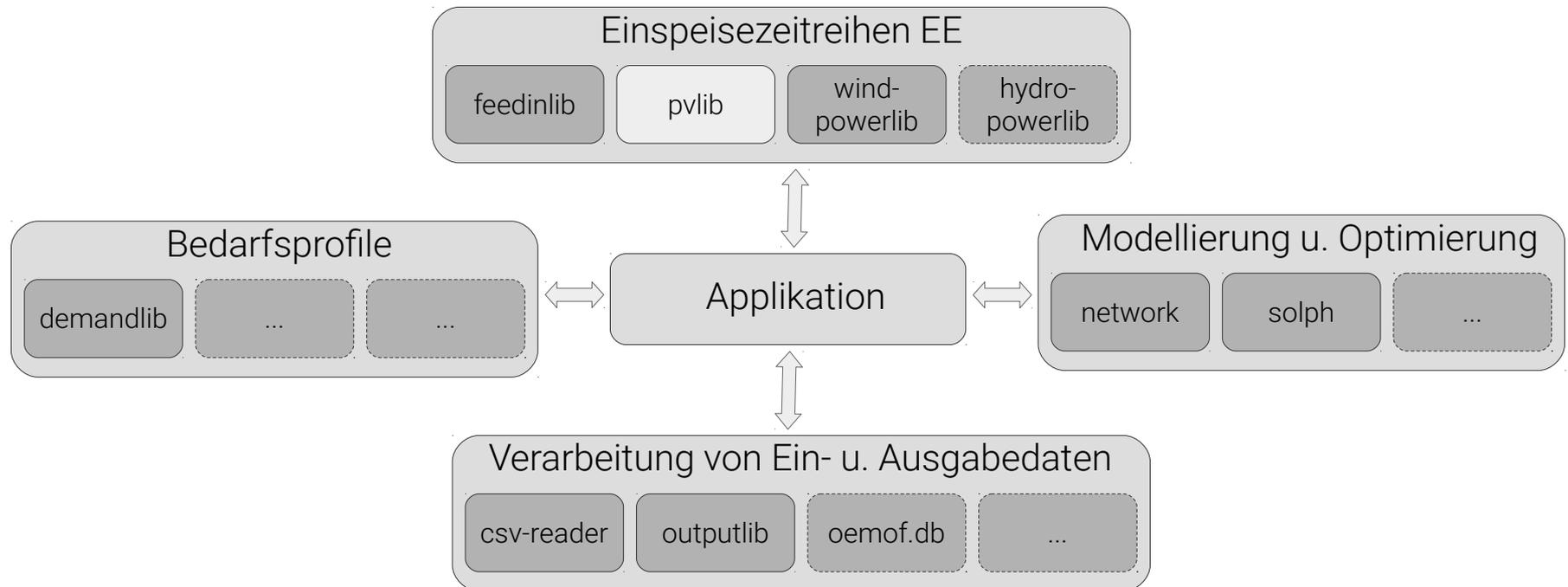






oemof - Applikation

- Nutzung und Kombination vorhandener Bibliotheken
- Alternative Bibliotheken können genutzt werden
- Neue Bibliotheken hinzufügen oder API für vorhandene integrieren



- HesysOpt
 - Heating System Optimization Tool
 - Mixed-integer-linear programming eines Fernwärmesystems
 - Weitere Informationen: <https://github.com/znes/HESYSOPT>
- reegis_hp
 - Ecological and economic evaluation of district heating and combined heat and power in an energy system based on renewable sources
 - Erstellung von Modellen auf Basis frei verfügbarer Daten
 - Weitere Informationen: https://github.com/rl-institut/reegis_hp

Wärmepumpe / Heizstab

- Wärmepumpen und Heizstäbe können als *Transformer* modelliert werden: $E_{out}(t) = E_{in}(t) \cdot cf(t)$ cf = conversion factor
- Für Heizstäbe ist cf häufig konstant
- Für Wärmepumpen kann der cf im Vorfeld berechnet werden (Kennfeld)
 - Wärmequellentemperatur ist vorher bekannt und unendlich groß
 - Die Vorlauftemperatur ist bekannt

```
COP = [3.1, 2.4, 2.3, 3.5]
```

```
LinearTransformer(  
    Label='heat_pump',  
    inputs={b_el: Flow()},  
    outputs={b_th: Flow()},  
    conversion_factors={b_th: COP})
```



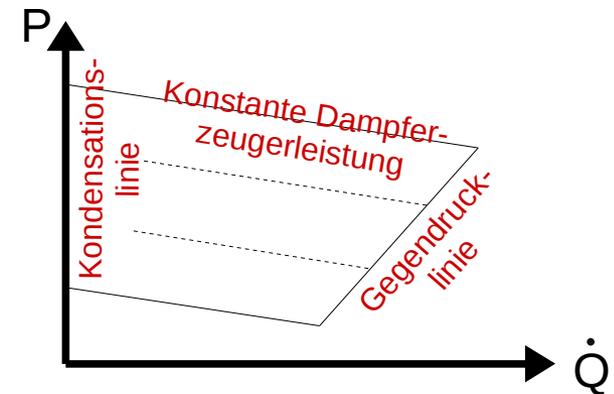
- Eine Wärmepumpe kann auch zwei Inputs haben

Entnahmekondensationsturbine

- Erweiterte Klassen mit zusätzlichen Attributen
- Entnahmeturbine hat zusätzlich einen Wirkungsgrad für den reinen Kondensationsbetrieb

$$E_{in}(t) = \frac{E_{el}(t) + E_{th}(t) \cdot li(t)}{cf_{cond}(t)} \quad li(t) = \frac{cf_{cond}(t) - cf_{el}(t)}{cf_{th}(t)}$$

$$\frac{E_{el}(t)}{cf_{el}(t)} \geq \frac{E_{th}(t)}{cf_{th}(t)}$$



```
VariableFractionTransformer(  
    label='variable_chp_gas',  
    inputs={b_gas: solph.Flow(nominal_value=10e10)},  
    outputs={b_el: solph.Flow(), b_th: solph.Flow()},  
    conversion_factors={b_el: 0.3, b_th: 0.5},  
    conversion_factor_single_flow={b_el: 0.5}  
)
```

Erweiterungen in oemof

- MILP Erweiterungen
 - Binary Flows
 - Discrete Flows
- Kombination von Basisklassen
 - Elektroauto mit Laderegulung
 - Druckluftspeicher
- Zusätzliche Nebenbedingungen (Constraints)
 - Intuitivere Nutzung geplant
- Eigene Klassen
 - Beispiel in solph vorhanden

Vielen Dank



uwe.krien@reiner-lemoine-institut.de

+49 (0)30 1208 434 76

Other developers via website:

www.oemof.org/contact

