Emobility in Python & Vehicle Energy Consumption in Python
Demonstration of two Open Source tools describing electric vehicle energy demand

DLR Knowledge exchange workshop

2020/05/05

Niklas Wulff, niklas.wulff@dlr.de

EVer Energie und Verkehr

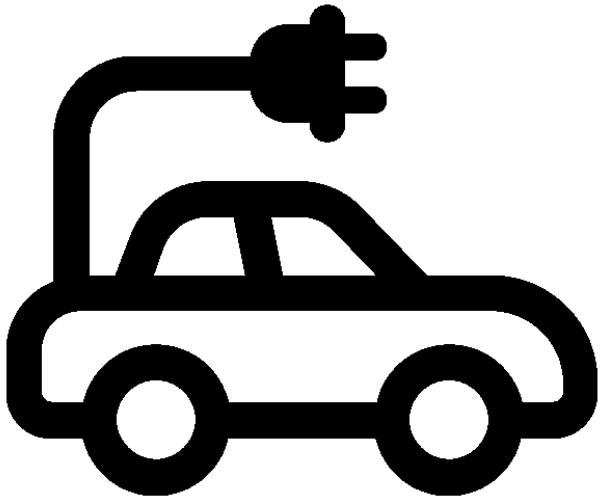
DLR

BEniVer
Begleitforschung Energiewende im Verkehr

When does electric charging occur?

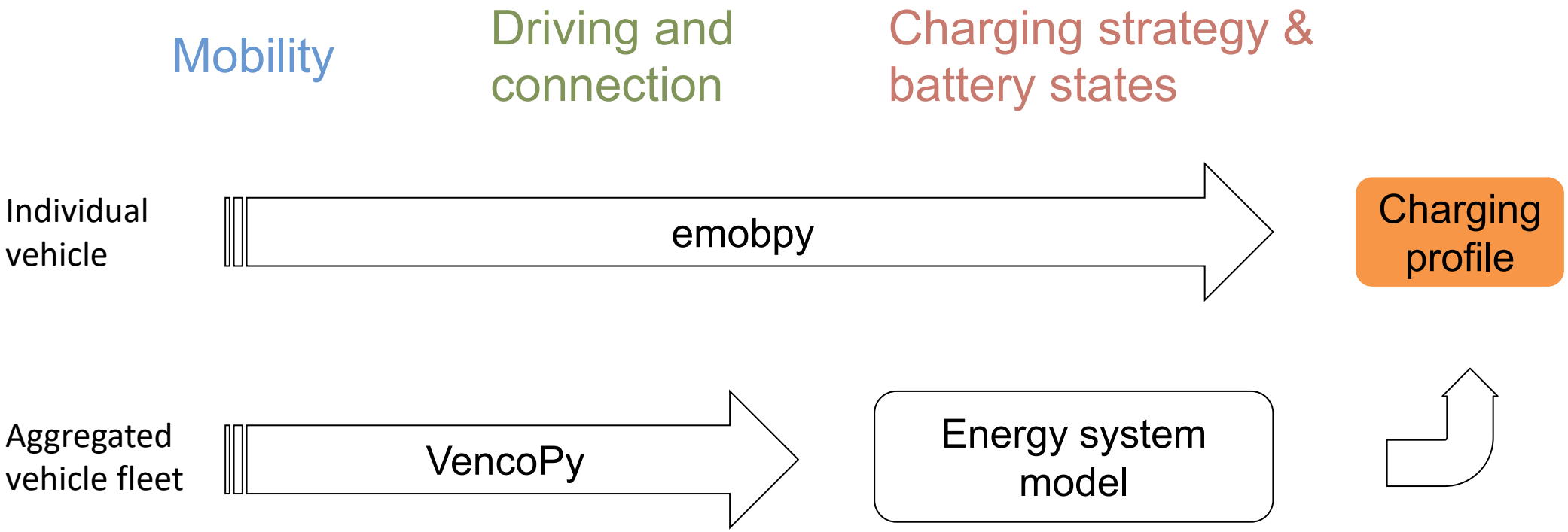
Electric demand is mainly driven by

- Number of electric vehicles
- Technical characteristics of vehicles
- Individual mobility behavior
- Charging infrastructure availability
- Charging strategy



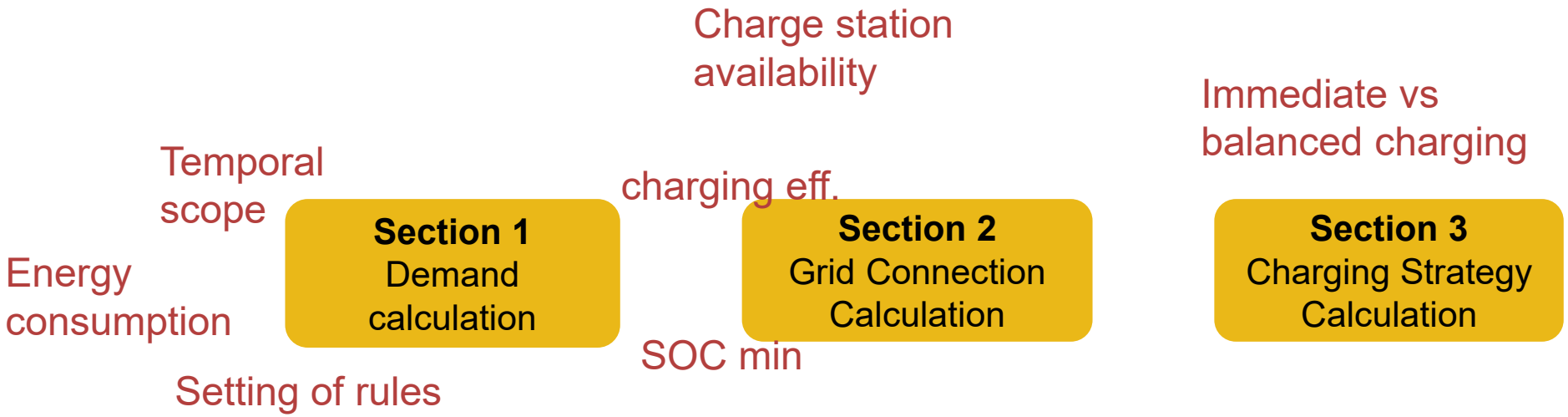
Created by Adrien Coquet
from Noun Project

When does electric charging occur? emobpy and VencoPy model pipelines



Emobpy – individual profile calculation

emobpy



Emobpy – individual profile calculation

emobpy

Demonstration Time

https://jupyter.org/

RISE 5.6.1

Jupyter slideshow extension

https://rise.readthedocs.io/en/stable/

Emobpy – individual profile calculation

emobpy

Function

- individual profile
- given charging strategy

Modeling

- Single electric vehicle
- One profile set per vehicle
- Stochastic trip synthesization procedure

Code

- Object-oriented programming
- Interfaces: jupyter notebook,
- License: MIT

Data

- Probabilities of distance, trip-purpose and departure times in .csv,
- technical assumptions in dictionary

VencoPy – vehicle fleet profiles as constraints

VencoPy

Trip profiles

Connection profiles

Technical assumptions

Config

Section 1
Input & Preprocessing

Section 2
Profile calculation

Section 3
Filtering, correction & aggregation

VencoPy – vehicle fleet profiles as constraints

VencoPy

Demonstration Time

https://jupyter.org/

RISE 5.6.1

Jupyter slideshow extension

https://rise.readthedocs.io/en/stable/

Comparing emobpy & VencoPy

emobpy

VencoPy

Function

- individual profile
- given charging strategy

- Normalized scalable profiles
- Input to energy system model

Modeling

- Single electric vehicle
- One profile set per vehicle
- Stochastic trip synthesization procedure

- Vehicle fleets
- One output per technology
- Aggregated, normalized profiles
- deterministic

Code

- Object-oriented programming
- Interfaces: jupyter notebook,
- License: MIT

- Functional programming
- Four library files holding functions
- Repo on gitlab
- Interfaces: .csv, .xlsx, .yaml
- Licensed via BSD-3-Clause

Data

- Distance, trip-purpose and departure times in .csv,
- technical assumptions in dictionary,

- Hourly driving and connection profiles
- Technical assumptions via .xlsx

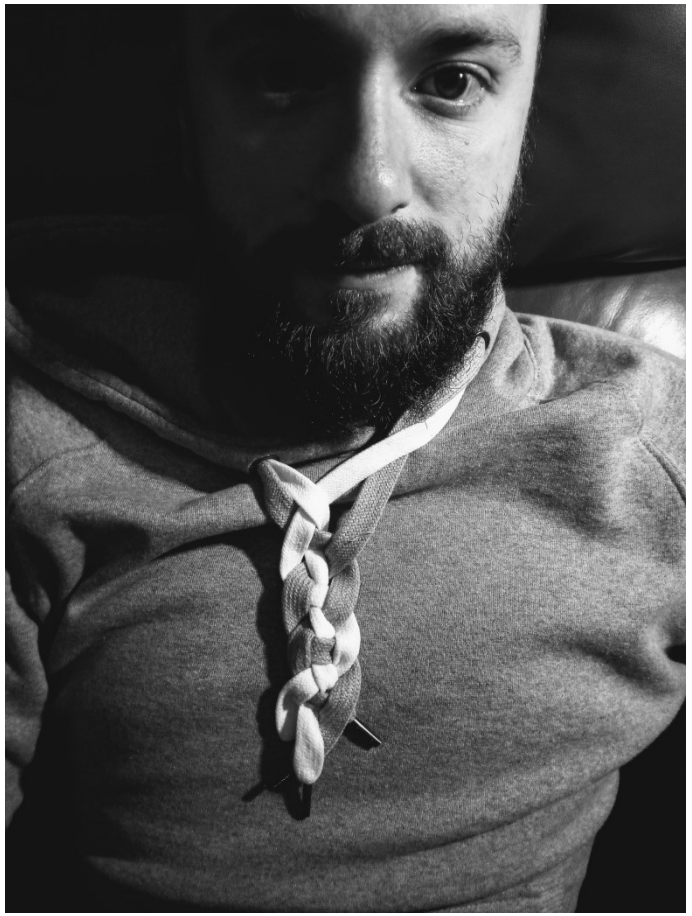
Wrap-up and conclusion

- Both tools are still **in development** and you‘re welcome to join into development
  - Publication applying VencoPy: https://www.mdpi.com/1996-1073/13/5/1093
- If you are interested in describing single vehicles and their charging, use emobpy
- If you want to calculate a technical flexibility potential of a future electric vehicle fleet use VencoPy
- Outlook: Potential merge of multiple emobpy runs with VencoPy fleet aggregation

https://about.gitlab.com/press/press-kit/

Links & Contact

Emobpy: https://gitlab.com/diw-evu/emobpy
VencoPy: https://gitlab.dlr.de/wulf_ni/vencopy



German Aerospace Center e.V. (DLR)

Institute of Engineering Thermodynamics | Pfaffenwaldring 38-40 | 70569 Stuttgart

Niklas Wulff | Ph.D. Candidate

+49 711 6862-348 | niklas.wulff@dlr.de