

¹ District Energy Model (DEM): An open-source model ² for local energy system simulation and optimisation.

³ **Ueli Schilt**  ^{1,2}, **Pascal M. Vecsei**  ¹, **Somesh Vijayananda**  ¹, and **Philipp Schuetz**  ¹

⁵ **1** Competence Centre for Thermal Energy Storage, School of Engineering and Architecture, Lucerne University of Applied Sciences, Horw, Switzerland **2** School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).²³

⁸ Summary

⁹ The transition to local, decentralised, renewable energy technologies is a promising pathway toward net-zero emission targets in a decarbonised energy system. This transition requires systematic evaluation of potential future scenarios across multiple spatial scales [ref_1]. Computational energy-system models are used for this purpose. The *District Energy Model (DEM)* is a Python-based multi-energy system model designed to simulate scenarios from neighbourhood to regional scale, with a focus on the integration of decentralised renewable energy technologies such as solar, wind, and biomass. DEM executes simulation and optimisation studies at hourly resolution using a snapshot-year methodology [ref_2]. It is released as an open-source Python library on PyPi. The model can be launched via a command-line interface, not requiring any Python programming knowledge. Alternatively, it can be accessed programmatically through a Python API. DEM requires two classes of input: *configuration files* and *data files*. *Configuration files* specify the simulation settings (e.g., included buildings, temporal scope, output variables) and define the energy system configuration (e.g., scenarios, technologies, selected year). They are provided in YAML format ([Ben-Kiki et al., 2009](#)). Configuration parameters may also be passed directly through the API. *Data files* contain model data such as energy demand profiles, generation potential, and ambient conditions. They are provided in Apache Feather format ([Apache Software Foundation, 2025](#)). For selected regions, these data files have been pre-compiled from public datasets and made available in a public repository, providing a fully parameterised model without requiring users to source or preprocess data. For example, data for a full parametrisation of DEM is available for all municipalities in Switzerland.

²⁹ DEM comes with a detailed [documentation](#) with step-by-step instructions, descriptions of modelling approaches and methods, and links to related publications and research.

³¹ Statement of need

³² Several countries have defined national net-zero emission targets. Switzerland, for example, aims to reach net-zero by 2050. Achieving such targets generally requires a structural shift from large, centralised generation to decentralised renewable resources, including solar, wind, and biomass. To support energy-system planning and policy design, scenario evaluation must be carried out at local scales such as districts, municipalities, cities, or similarly sized regions. This includes analysing system-integrated deployment of renewable-energy generation, conversion, and storage technologies; assessing alternative demand trajectories; and identifying solutions optimised for specific objectives such as cost, emissions, or security of supply.

⁴⁰ DEM provides these functions with a specific focus on the use of local renewable-energy resources and the integration of decentralised technologies within local system boundaries.

42 Multiple energy-system and demand scenarios can be defined, simulated, and compared.
43 Increased penetration of variable, distributed resources raises the relevance of supply- and
44 demand-side flexibility [ref]. DEM models several flexibility options, including flexible electric-
45 vehicle charging, thermal and electrical storage, hydrogen storage, and photovoltaic curtailment.
46 Existing multi-energy system models have been applied extensively in case studies of local
47 energy scenarios [refs], but they typically target a single location. Each new application demands
48 re-parametrisation and new data collection, including demand profiles, cost estimates, and
49 technology characteristics. Data acquisition and preparation dominate the modelling workload
50 in such studies. DEM removes this burden for selected regions by providing pre-compiled and
51 pre-processed datasets assembled from public sources. Simulation and optimisation studies can
52 therefore be executed with minimal configuration (e.g., selecting the buildings to include) while
53 maintaining full flexibility to replace any pre-configured dataset with user-defined data when
54 required. For regions not included in the provided dataset, users can construct the necessary
55 data using the specifications provided in the documentation. DEM's input data architecture
56 allows datasets to be provided at a large regional scale (e.g., an entire country) and then
57 used to run simulations on any spatial subset of that data, such as individual municipalities or
58 districts.
59 Optimisation is optional in DEM. Many scenario questions, such as assessing the impact of a
60 specific technology, do not require optimisation. In such cases, DEM runs simulations without
61 invoking the optimisation module. This yields short computation times and rapid generation
62 of results.

63 Modelling approach

64 DEM simulates energy flows within a defined district using a hybrid bottom-up and top-down
65 modelling approach. A “district” can represent any spatial aggregation from a small group of
66 buildings to an entire municipality or city. Building-level attributes are modelled individually
67 (e.g., type, location, size, age, heat and electricity demand, heating system, and on-site
68 solar potential). Other parameters are defined at district scale, including wind and biomass
69 resources, ambient conditions, and mobility demand. Each simulation is constructed from three
70 elements: a set of available resources (e.g., wind, solar, biomass, hydro), a set of technologies
71 for generation, conversion, and storage, and a set of demand profiles for heat, electricity, and
72 mobility. These elements interact through defined flows of resources and energy carriers such
73 as electricity and heat. An example system layout is shown in Fig. [Figure 1](#). DEM imposes
74 no fixed limit on the number of buildings included, allowing customised definitions of district
75 boundaries and building selections.
76 The workflow consists of: (1) input-data collection; (2) model parametrisation and configuration;
77 (3) scenario generation; (4a) simulation; (4b) optimisation (optional); (6) output generation.
78 The optimisation module in DEM is implemented using the Calliope framework ([Pfenninger &](#)
79 [Pickering, 2018](#)), which is based on the Pyomo optimisation programming environment ([Hart
et al., 2011](#)).

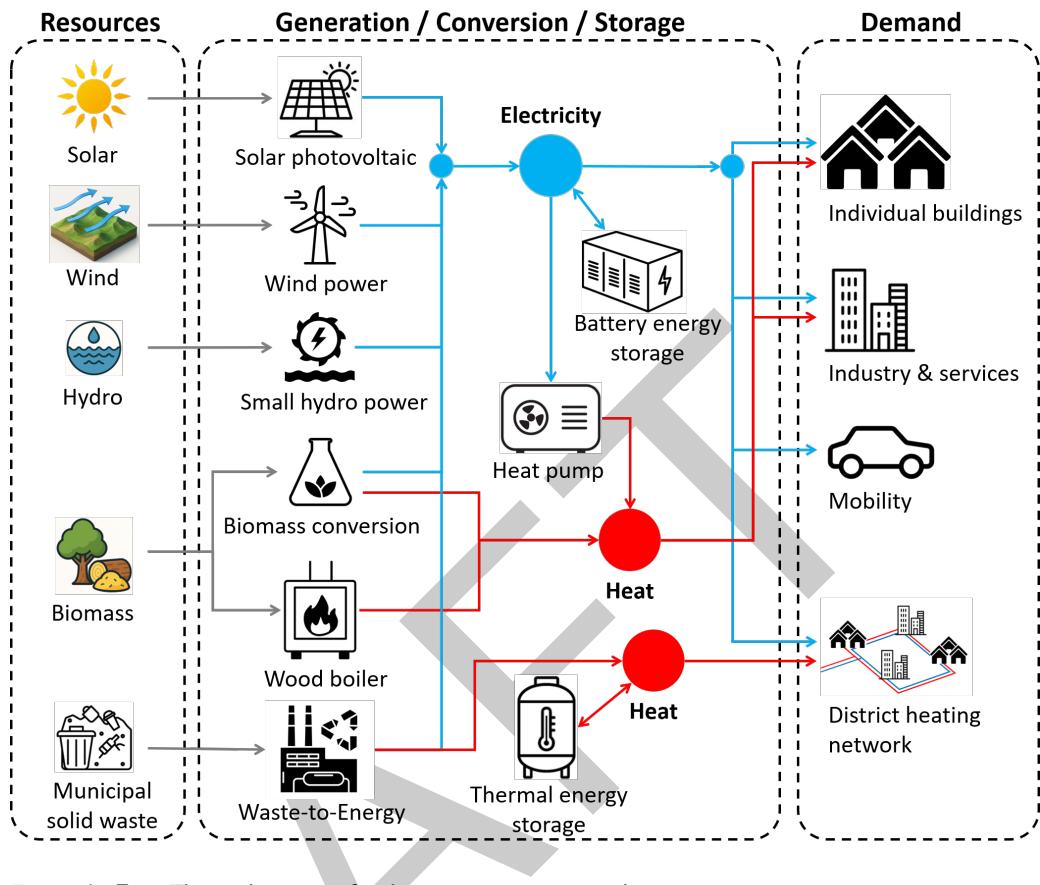


Figure 1: Exemplary schematic of a district energy system showing resources, generation, conversion, and storage technologies, and associated heat, electricity, and mobility demands. DEM supports many more technologies and scenarios than those illustrated here, as detailed in the [documentation](#).

81 Acknowledgements

82 The research published in this publication was carried out with the support of the Swiss Federal
 83 Office of Energy as part of the SWEET consortium EDGE. The authors bear sole responsibility
 84 for the conclusions and the results presented in this publication.

85 References

- 86 Apache Software Foundation. (2025). *Feather file format (apache arrow)*. <https://arrow.apache.org/docs/python/feather.html>
- 87 Ben-Kiki, O., Evans, C., & Net, I. d"ot. (2009). *YAML ain't markup language (YAML™)*
 88 version 1.2, 3rd edition. <https://yaml.org/spec/1.2/spec.html>.
- 90 Hart, W. E., Watson, J.-P., & Woodruff, D. L. (2011). Pyomo: Modeling and solving mathematical
 91 programs in python. *Mathematical Programming Computation*, 3(3), 219–260.
 92 <https://doi.org/10.1007/s12532-011-0026-8>
- 93 Pfenninger, S., & Pickering, B. (2018). Calliope: A multi-scale energy systems modelling
 94 framework. *Journal of Open Source Software*, 3(29), 825. <https://doi.org/10.21105/joss.00825>