# D4.8 – Open-access tool of linked electricity market models

*Deliverable number:* *D4.8*

*Work Package:* *WP4 – Development of Open-access Market Simulation Models and Tools*

*Lead Beneficiary:* *VTT Technical Research Centre of Finland Ltd*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 864276

| Author(s) information (alphabetical) | | |
|---|---|---|
| **Name** | **Organisation** | **Email** |
| Erkka Rinne | VTT | erkka.rinne@vtt.fi |
| Christoph Schimeczek | DLR | christoph.schimeczek@dlr.de |
| Hugo Algarvio | LNEG | hugo.algarvio@lneg.pt |
| Gabriel Santos | ISEP | gajls@isep.ipp.pt |
| Milos Cvetkovic | TUD | m.cvetkovic@tudelft.nl |
| Ingrid Sanchez Jimenez | TUD | I.J.SanchezJimenez@tudelft.nl |
| Laurens de Vries | TUD | L.J.deVries@tudelft.nl |
| Ricardo Hernandez Serna | TNO | ricardo.hernandezserna@tno.nl |
| German Morales | TNO | german.morales@tno.nl |
| Jos Sijm | TNO | jos.sijm@tno.nl |


| Acknowledgements/Contributions | | |
|---|---|---|
| **Name** | **Organisation** | **Email** |
| Tiago Pinto | ISEP | tcp@isep.ipp.pt |


| Document information | | | |
|---|---|---|---|
| **Version** | **Date** | **Dissemination Level** | **Description** |
| 1.0 | 25/11/2021 | Public | *A software framework of open-access electricity market models development is described in this deliverable.* |

| Review and approval | | |
|---|---|---|
| **Prepared by** | **Reviewed by** | **Approved by** |
| Erkka Rinne (VTT) | Milos Cvetkovic (TUD) | Ana Estanqueiro (LNEG) |

**Disclaimer**

# Executive Summary

For a holistic understanding and simulation of the energy markets, many different aspects need to modelled properly. Often no single modelling tool offers the whole picture, but a combination of methods needs to be used. A model linkage platform has been chosen previously and this deliverable describes the integration of the modelling tools used in TradeRES to the linking application, Spine Toolbox.

Spine Toolbox was used to build data processing and execution workflows around the energy system modelling tools Backbone, AMIRIS, EMLab, COMPETES, RESTrade and MASCEM. The aim is to integrate selected tools together for answering the research questions in the TradeRES project. The integrations of individual tools and some combinations are described in this document. Mostly, the work is still in progress.

Also, a common database to serve the case studies has been created, but populating the database with scenario data is still in progress. A common data model to serve all the modelling tools has been created and the database is implemented using Spine Toolbox.

# Table of Contents

# List of Tables



# List of Figures

# 1. Introduction

## 1.1 Goals of Task 4.3, Subtask 4.3.3 and Deliverable 4.8

**Task 4.3: Integration of models with a model-linkage toolbox**

Linking multiple models that specialise in certain aspects of the electricity markets allows to create more holistic and more robust simulation results. Works of model linkage in this task will be strongly connected to model updates from Tasks 4.1 and 4.2 and feedback from Tasks 6.1 and 6.2. Therefore, the work in this task will be ongoing and proceed even after delivering a first version of the multi-modelling toolbox in milestone M#4.

**Subtask 4.3.3: Integration with a model linkage toolbox**

Herein we create a holistic model-linkage architecture for multi-modelling of energy markets exploiting existing platforms like Mosaik, VirGIL, Spine Toolbox and RCE. Once a platform is selected, we will integrate the partaking models in the platform. This will serve as an open-access market trading tool for expert stakeholders.

**Deliverable 4.8: Open-access tool of linked electricity market models**

A software framework of open-access electricity market models will be developed in this deliverable. Delivery month: M21 – first version (postponed to M22), M41 – final updated version. (Originally D4.3)

## 1.2 Background

Spine Toolbox[1] was chosen as the integration platform for different modelling tools. It is a free software application to manage data, scenarios and workflows for modelling and simulation tasks. The application is a result from H2020 project *Spine* (2017–2021) and is in active development also after the project by several partners. The source code of Spine Toolbox is available under the GNU Lesser General Public License version 3 (LGPLv3).

With Spine Toolbox, the user can create a workflow containing data items, data processing and validation, data management using a local or shared database, execution of modelling tools and archiving results including simple visualisations. A collection of such directed, acyclic workflows (graphs) is called *a project*. In other words, a project contains the data processing and modelling tasks needed to accomplish certain research or other work.

Spine Toolbox was chosen because of its strong workflow support and database-centred architecture as explained in Deliverable 4.7. (Cvetkovic et al., 2020).

## 1.3 Scope and structure of the document

The scope of this document is to introduce and describe the Spine Toolbox projects demonstrating the integration of various modelling tools (Section 2) and the shared database (Section 3).

[1] https://github.com/Spine-project/Spine-Toolbox

## 2. Integration of models

The integration of all modelling tools into Spine Toolbox is happening piece by piece. First each model is tested on its own by creating a Spine Toolbox project where the particular model is executed. Finally, relevant models will be combined together. Table 1 shows the status and online repository location for each modelling tool. They are discussed in more detail in the following sub-sections.

Table 1: Status of modelling tool integration

| Modelling tool or other aspect | Status | Repository (under https://github.com/TradeRES) |
|---|---|---|
| **Backbone** | In progress | TradeRES-Backbone-demo |
| **AMIRIS** | In progress | toolbox-amiris-demo |
| **EMLab + COMPETES** | Completed | Spine_EMLab_COMPETES |
| **TradeRES database** | In progress | traderes-database-demo |
| **Restrade + MASCEM** | In progress | mascem-restrade-demo |
| **MASCEM** | In progress | mascem-demo |

### 2.1 Backbone

A demo project for executing the Backbone modelling tool using Spine Toolbox has been created. The dataset used is a very simple imaginary single-node system with a gas turbine unit and wind power generation. The model is used to find the minimum operational costs to fulfil the electricity demand during five discrete time steps.

The Toolbox workflow for the project is shown in Figure 1. Original data files are connected through the item **Data Excel** and the data is imported into the **TradeRES** datastore. This data follows the simple TradeRES ontology: geo, conversion and commodity. In order for the data to be compatible with Backbone, some renaming and transformations are done and this results in another datastore, **BB_Spine_DB**.

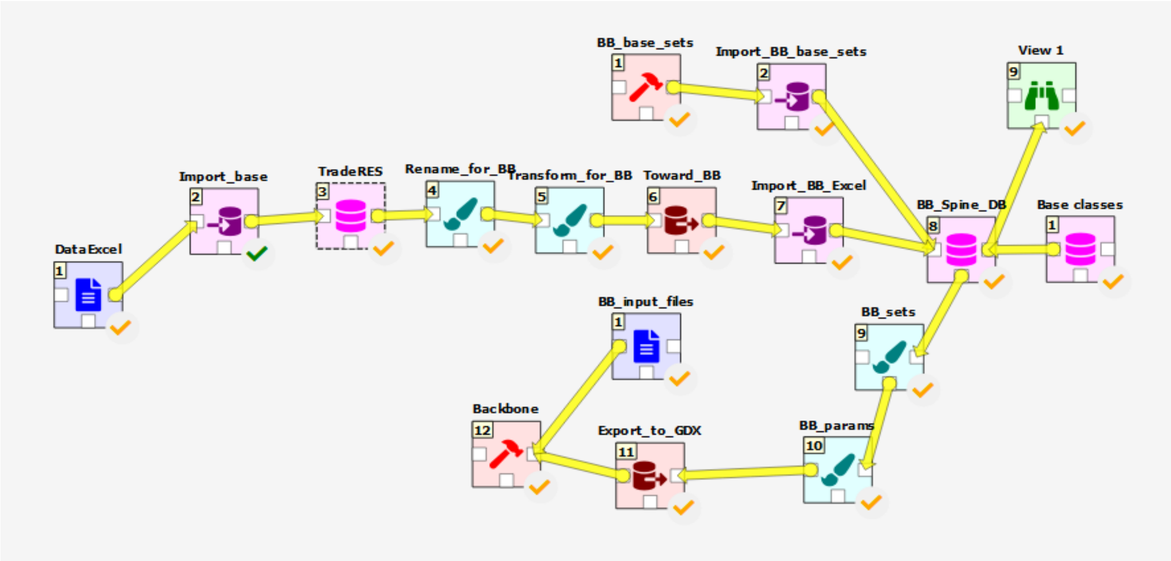


Figure 1: Spine Toolbox Backbone workflow

The workflow items after **BB_Spine_DB** are common to all Backbone projects and they are used to create a binary GDX (GAMS Data eXchange) file which can be consumed by the model code. Backbone modelling tool is executed in item **Backbone**, and some additional configuration files are given by data connection **BB_input_files**.

## 2.2 AMIRIS

The **A**gent-based **M**arket model for the **I**nvestigation of **R**enewable and **I**ntegrated energy **S**ystems (see, e.g. Deissenroth et al., 2017) is one of the four agent-based models within the TradeRES project. It is based on FAME[2], the open Framework for distributed Agent-based Modelling of Energy system. FAME simulations need to provide their input in a Protobuf[3] format and provide their output in a Protobuf file as well. Thus, to run an AMIRIS simulation, the Python-base tool FAME-Io is required to translate human-readable data to FAME-compatible files and vice versa.

---

[2] https://gitlab.com/fame-framework

[3] https://developers.google.com/protocol-buffers

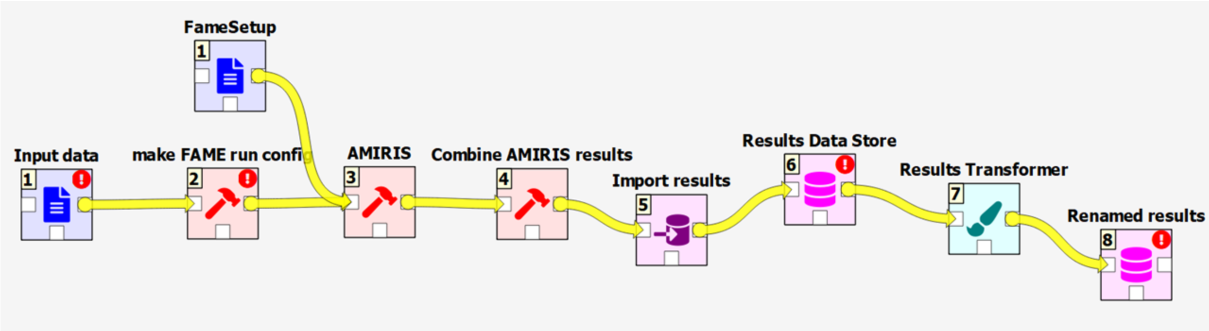


Figure 2: Workflow for AMIRIS execution within Spine Toolbox

Figure 2 shows an exemplary workflow for the execution of an AMIRIS simulation run including the following steps: First, the input data in YAML[4] and CSV format are translated into a single file in Protobuf format utilising FAME-Io[5] (step 'make FAME run config'). The protobuf file is directed to "AMIRIS", which is given as JAR[6]-container and must be executed on any Java Virtual Machine supporting Java 1.8 or higher versions. The optional file "Fame-Setup" controls basic simulation parameters like path / naming parameters for the output file. After a few seconds, the simulation has finished, and its output file can be converted to files in CSV format in step ('Combine AMIRIS results'). To simplify further content assessment, the output files' content is read and joined to a data frame utilising the wide-spread library Pandas[7] in that same step. The resulting data frame can then be imported into the Spine database 'Results Data Store', from which the symbols can be extracted, renamed and further transformed to any required format.

Please check out the FAME-Core and FAME-Io ReadMe files and FAME Wiki[8] for detailed instructions and troubleshooting advice. Note that AMIRIS and FAME are not included in the 'toolbox-amiris-demo'-project but need to be installed separately.

## 2.3 EMLab and COMPETES

The first prototype of the coupled EMLab and COMPETES model is created in Spine Toolbox. This coupling is made in support of the Dutch national case study of WP5. The coupled model is used to explore the investment trajectory in new generation assets in the presence of additional revenues from the capacity market. The $CO_2$ market model is included in the analysis. The investment considerations are made yearly, and hence, this coupling operates with a time resolution of one year. The time horizon can be specified by user. EMLab modules provide capacity mechanism and $CO_2$ market functionalities, while COMPETES provides accurate information about operational costs in the current year and

[4] https://yaml.org/

[5] https://gitlab.com/fame-framework/fame-io

[6] https://docs.oracle.com/javase/tutorial/deployment/jar/

[7] https://pandas.pydata.org/

[8] https://gitlab.com/fame-framework/wiki/-/wikis/home

is additionally used to support investment decisions by providing valuation of different technologies in a future year.

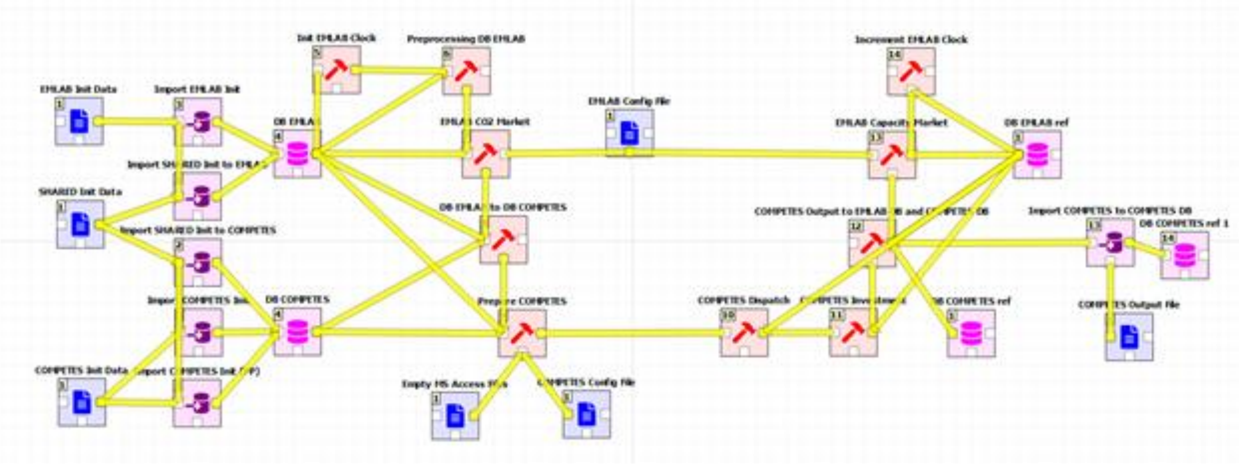


Figure 3: Workflow for coupling of EMLab and COMPETES within Spine Toolbox

The Toolbox workflow is shown in Figure 3. This workflow represents one iteration of a larger simulation, in which all iterations repeat the same steps. Spine Toolbox provides looping capabilities which allow us to consider the workflow in Figure 3 as one iteration. One iteration represents one year, i.e., the investment decision for the current year is taken within the workflow in Figure 3.

The workflow in Figure 3 will be explained from left to right. In this implementation, three input files with initialization data are envisioned, one of which is a common dataset shared by both models. The datasets are imported to databases of EMLab and Competes. The so-called **EMLab clock** is used to initialize and increment the iteration mechanism, leading to the looping functionality. The **EMLab CO2 Market** module calculates the costs and revenues from the $CO_2$ market for all generation companies. The **COMPETES Dispatch** item runs the electricity system model of the current year, calculating operational costs and revenues. The **COMPETES Investment** item decides which new assets to install, and these are made available (and become implemented in the **COMPETES Dispatch** item) considering their respective building times. Next, the **EMLab Capacity Market** calculates the capacity market clearing price. The capacity market remuneration is considered for the new investments and decisions on decommissioning. The final step in the workflow is to save the results into databases of COMPETES and EMLab so that these become the initial state for the next iteration.

## 2.4 RESTrade and MASCEM

This project integrates some model capabilities from MASCEM and RESTrade, namely by integrating MASCEM's wholesale day-ahead market models with RESTrade's secondary and tertiary energy markets.

In the current stage of development, the RESTrade system has models for secondary capacity procurement and imbalance settlement. It also has models for secondary capacity and energy markets, such as for the tertiary energy market. In the current stage of development, the Toolbox workflow of the coupled MASCEM and RESTrade models is as presented in Figure 4.

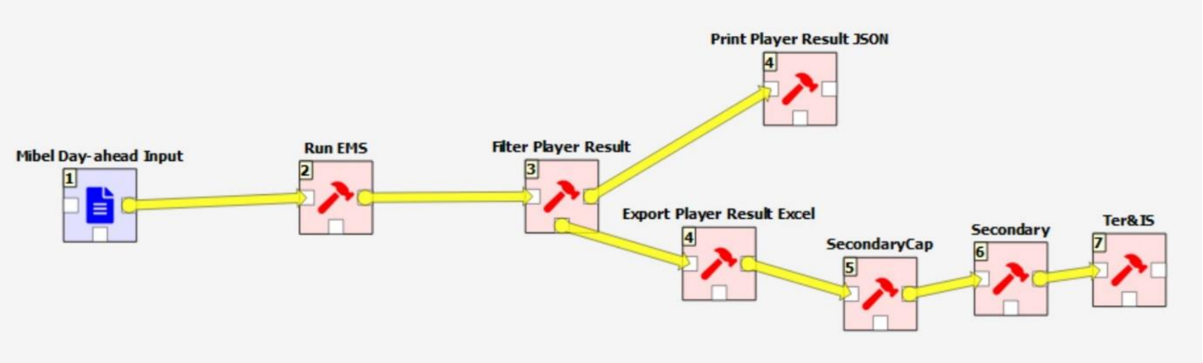


Figure 4: Workflow of Spine Toolbox project comprising RESTrade

The **Mibel Day-ahead Input** item consists of a JSON file that will act as input for the Electricity Markets Service, containing information regarding the players and their price/amount bids for each of the periods.

The **Run EMS** item consists in the execution of the Electricity Markets Service, to simulate a market session, in this case, a MIBEL Day-ahead session, using the JSON file described above as input and returning as results another JSON file containing the prices and the quantities traded for each period globally and by player. It also verifies the input locally, by using a schema created to verify the structure of the JSON input.

The **Filter Player Result** item will pick the results coming from the previous tool and will filter them by one of the players present in the market, showing the quantities he traded and the market prices for each one of the periods as well as globally. The player is passed as an argument to the tool.

The **Print Player Result JSON** item simply prints the filtered player result in the previous tool.

The **Export Player Result Excel** item transforms the filtered player result in JSON to an excel format.

The **SecondaryCap** item contains both the models of the procurement of secondary capacity and of the secondary capacity market. To compute the secondary procurement is necessary to receive the maximum expected consumption per period under simulation (currently, it is in an excel file). Furthermore, the TSO submits to the market the up and down needs of the secondary capacity and collects the agents' bids to the market (excel file), computing the market-clearing price(s) and up and down capacities, and exporting them to an excel file.

The **Secondary** item contains the market model of the secondary energy. The TSO reads the secondary dispatch needs from an excel file, collects the agents' bids (excel file), and computes the up and down energy prices, and exports the results to an excel file.

The **Ter&IS** item contains both the models of the tertiary energy market and of the imbalance settlement. The TSO reads the tertiary dispatch needs from an excel file, collects the agents' bids (excel file), computes the up and down energy prices, and exports the results to an excel file. Furthermore, the TSO computes all costs with the reserve markets and the respective penalties of the imbalanced agents. Also, the TSO reads the day-ahead prices computed in MASCEM from an excel file and computes the up and down imbalance prices. The penalties and imbalance prices are exported to an Excel workbook.

## 2.5 MASCEM

This project provides the implementation of MASCEM's electricity market models and power flow models in Spine Toolbox. This specific project is prepared to execute the Iberian electricity market – MIBEL[9] and validate the power network constraints, according to the market results. The Toolbox workflow is shown in Figure 5.

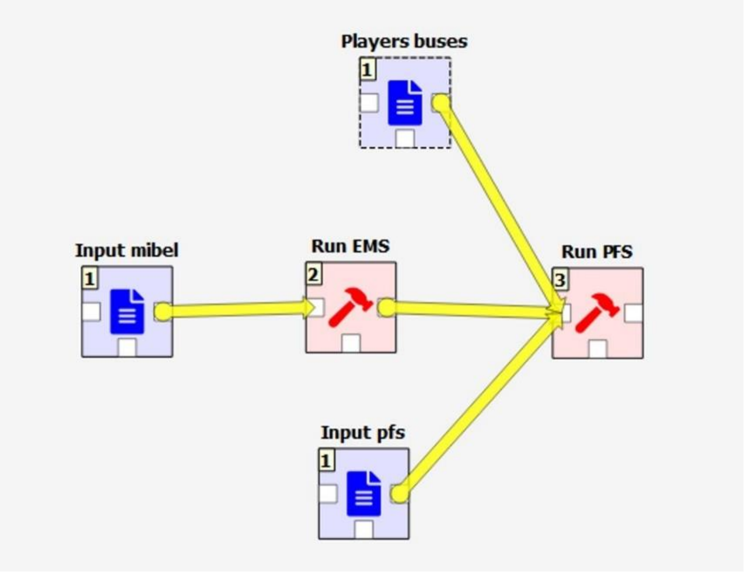


Figure 5: Workflow of MASCEM's Spine Toolbox project

The **Input mibel** item is the input for the **Run EMS** component and contains information regarding the players and their price/amount bids for each period.

The **Player buses** item is one of the inputs of the **Run PFS** component. This file maps the players present in the MIBEL market to the buses of the network defined in the **Input pfs**.

The **Input pfs** item is one of the inputs of the **Run PFS** component and contains information regarding the elements of the network, as well as the power flow algorithm to be used and its parameters.

The **Run EMS** item executes the MIBEL day-ahead market (defined in the input of the previous component), validating it with the JSON schema created to verify the structure of the JSON input. The results of simulating the MIBEL market are then sent to the **Run PFS** tool.

The **Run PFS** item executes the Power Flow service receiving the network, the power flow algorithm to be used and the loads of each bus by mapping the players' results to the buses. It also validates the **Input pfs** with the JSON schema created to verify the structure of the JSON input. The results are then saved in a JSON file.

---

[9] https://www.mibel.com/en/home_en/

# 3. TradeRES shared database

A common data model to serve most features of all the modelling tools has been designed. The main entity classes of the model are node, commodity, storage, flow, unit, group and reserve type. Table 2 and Figure 6 list the entity classes and their relationships, respectively.

Table 2: Entity classes of the common data model

| Class | Description |
|---|---|
| Node | Energy balance is calculated at nodes or they act as a source or a sink of energy |
| Commodity | Energy commodities or carriers |
| Storage | Storages can store energy over time |
| Flow | Flows describe energy inputs like wind power |
| Unit | Units transform energy from different energy carriers to another |
| Group | Group for sharing reserves between nodes |
| ReserveType | Power reserve product type |

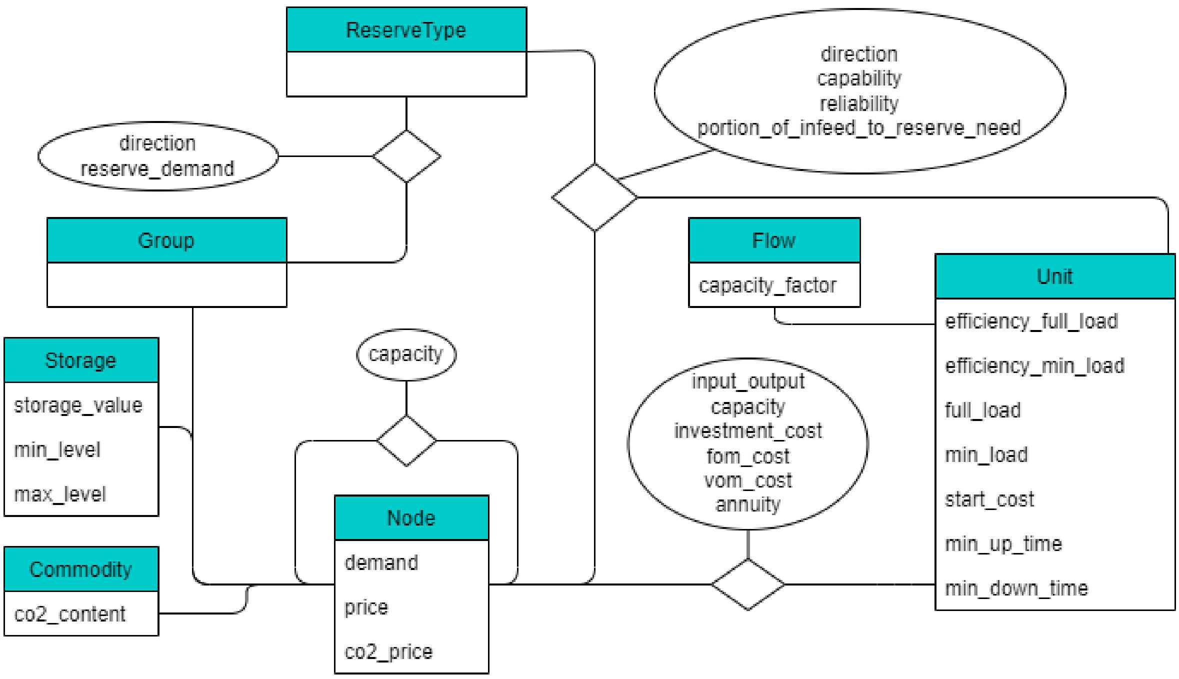


Figure 6: Entity–relationship diagram for the TradeRES common data model

Table 3: Attributes of the common data model

| Class or relationship | Attribute | Data type | Description |
|---|---|---|---|
| Node | demand | time series (real) | Exogenous demand for energy (MWh) |
| | price | time series (real) | Price of commodity at the node (EUR/MWh) |
| | co2_price | time series (real) | Price of CO2 emissions (EUR/tonne) |
| Commodity | co2_content | decimal | CO2 content of fuel (tonne/MWh) |
| Storage | storage_value | time series (real) | Specific value of stored energy (EUR/MWh) |
| | min_level | real | Minimum level of storage (MWh) |
| | max_level | real | Maximum level of storage (MWh) |
| Flow | capacity_factor | time series (real) | Capacity factor |
| Unit | efficiency_full_load | real | Efficiency at full load |
| | efficiency_min_load | real | Efficiency at minimum load |
| | full_load | real | Full load operation point |
| | min_load | real | Minimum operation point |
| | start_cost | real | Start-up cost (EUR/MW) |
| | min_up_time | real | Minimum up-time |
| | min_down_time | real | Minimum down-time |
| Node, Unit, ReserveType | direction | text | Direction of reserve {up, down} |
| | capability | real | Proportion of the unit capacity that can contribute to the reserve |
| | reliability | real | Reliability for providing this reserve (%) |
| | portion_of_infeed_to_reserve_need | real | Proportion of the generation of a tripping unit that needs to be covered by reserves from other units |
| Node, Unit | input_output | text | Is the node providing input or getting the output of the unit {input, output} |
| | capacity | real | Installed capacity for this unit (MW) |
| | investment_cost | real | Investment cost for this unit (EUR/MW) |
| | fom_cost | real | Fixed operation and maintenance cost (EUR) |

| Class or relationship | Attribute | Data type | Description |
|---|---|---|---|
| | vom_cost | real | Variable operation and maintenace cost (EUR/MWh) |
| | annuity | real | Annuity factor (%) |
| Node, Node | capacity | real | Transfer capacity form node to node (MW) |
| Group, ReserveType | direction | text | Direction of reserve {up, down} |
| | reserve_demand | time series (real) | Demand for reserve (MWh) |

A shared MySQL database server has been set up to serve the modelling activities. The server is hosted and managed by LNEG. The database is being populated using the TradeRES scenario data as described in Deliverable 2.1 (Helistö et al., 2021). At this point, only a fraction of all the data is stored but this is a work in progress. Figure 7 shows a preview of the database using Spine Database Editor.

The four TradeRES scenarios (S1, S2, S3 and S4) were divided into a number of alternatives which are used to store actual data values. For example, scenario *S1-low* ('conservative' with low fuel prices) consists of the following alternatives: low fuel prices, some thermal power capacity, low sector coupling, few new electricity loads and little flexible demand. Table 4 shows all the alternatives and how they are linked with the TradeRES scenarios.

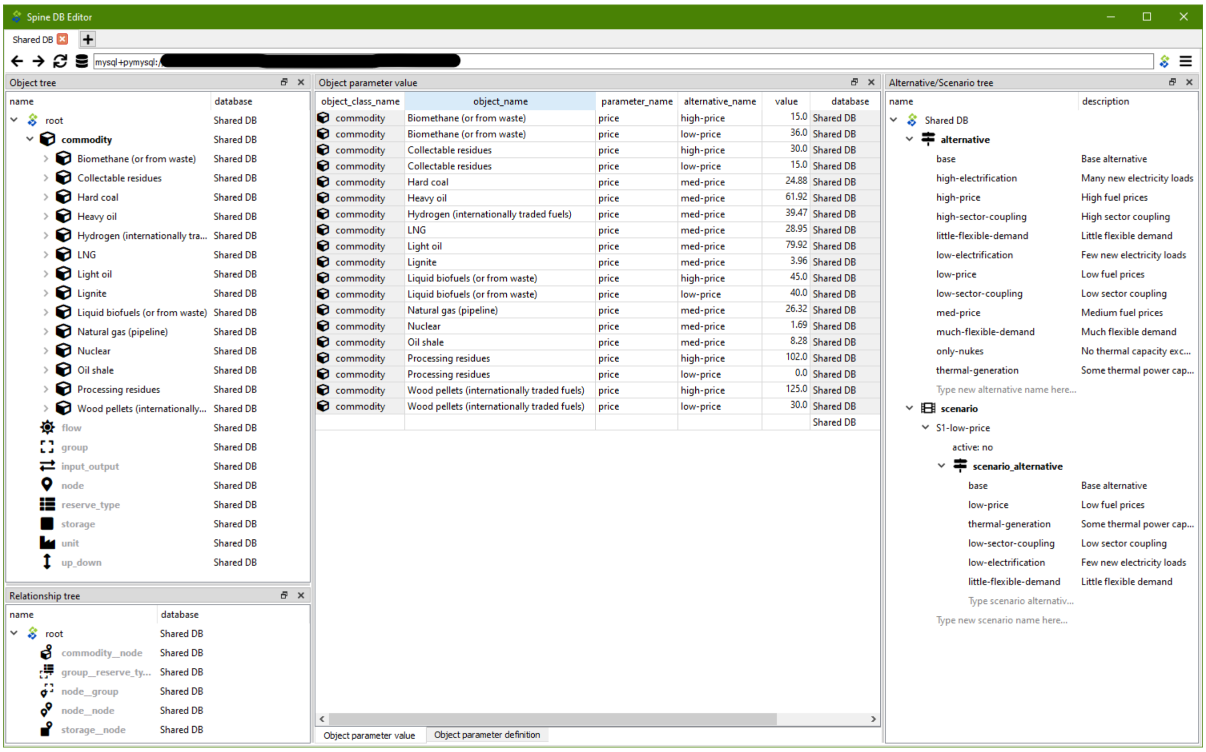


Figure 7: Preview of the common database with fuel price data

Table 4: Spine database alternatives and TradeRES scenarios

| Alternative name | Description | Scenarios |
|---|---|---|
| high-electrification | Many new electricity loads | S2, S3 |
| high-price | High fuel prices | (-high variants) |
| high-sector-coupling | High sector coupling | S2, S3 |
| little-flexible-demand | Little flexible demand | S1, S4 |
| low-electrification | Few new electricity loads | S1, S4 |
| low-price | Low fuel prices | (-low variants) |
| low-sector-coupling | Low sector coupling | S1, S4 |
| med-price | Medium fuel prices | (-med variants) |
| much-flexible-demand | Much flexible demand | S2, S3 |
| only-nukes | No thermal capacity except nuclear | S3, S4 |
| thermal-generation | Some thermal power capacity | S1, S2 |

# 4. Conclusion

Integration of the various modelling tools is still in progress and the project team is also learning as they work with Spine Toolbox. Next steps are finalising the individual tool integrations and building combined workflows for the case studies. To this end, also the common database is required, and entering all the data provided by WP2 is underway. These developments will be described in the final version of this deliverable (due M41).

# References

Cvetkovic, Milos, Juha Kiviluoma, Erkka Rinne, Ingrid Sanchez Jimenez and Christoph Schimeczek. (2020). *D4.3.2 – Principles and usage of a multi-simulation electricity market tool (D4.7)*. TradeRES project deliverable.

Deissenroth, M., M. Klein, K. Nienhaus, and M. Reeg, ‘Assessing the Plurality of Actors and Policy Interactions: Agent-Based Modelling of Renewable Energy Market Integration,’ Complexity, vol. 2017, pp. 1–24, 2017, doi:10.1155/2017/7494313.

Helistö, Niina, Juha Kiviluoma, Lassi Similä, Kristina Nienhaus and Ricardo Hernandez Serna. (2021). ‘Data sheets for the database of TradeRES scenarios’ (D2.1)