FlexPlan

Advanced methodology and tools taking advantage of storage and FLEXibility in transmission and distribution grid PLANning

Planning tool user documentation

D3.2

Distribution Level	PU
Responsible Partner	N-SIDE
Checked by WP leader	Hanot Maxime (N-SIDE) – WP3 Leader Date: 25/10/2022
Approved by Project Coordinator	Gianluigi Migliavacca (RSE) Date: 26/10/2022

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 863819

Issue Record

Planned delivery date	30/06/2022
Actual date of delivery	26/10/2022
Status and version	FINAL

Version	Date	Author(s)	Notes
0.1	30/09/2022	Maxime Hanot (N-SIDE) Alberte Bouso (N-SIDE)	Initial version of the deliverable, including sections “API Start Guide”, “Data Models”, and “GUI Start Guide”.
0.2	25/10/2022	Maxime Hanot (N-SIDE)	Add “Appendix” and process comments from Project Coordinator.

About FlexPlan

The FlexPlan project aims at establishing a new grid planning methodology considering the opportunity to introduce new storage and flexibility resources in electricity transmission and distribution grids as an alternative to building new grid elements. This is in line with the goals and principles of the new EC package *Clean Energy for all Europeans,* which emphasizes the potential usage of flexibility sources in the phases of grid planning and operation as alternative to grid expansion. In sight of this, FlexPlan creates a new innovative grid planning tool whose ambition is to go beyond the state of the art of planning methodologies, by including the following innovative features: integrated T&D planning, full inclusion of environmental analysis, probabilistic contingency methodologies replacing the N-1 criterion as well as optimal planning decision over several decades. However, FlexPlan is not limited to building a new tool but it also uses it to analyse six regional cases covering nearly the whole European continent, aimed at demonstrating the application of the tool on real scenarios as well as at casting a view on grid planning in Europe till 2050. In this way, the FlexPlan project tries to answer the question of which role flexibility could play and how its usage can contribute to reduce planning investments yet maintaining (at least) the current system security levels. The project ends up formulating guidelines for regulators and for the planning offices of TSOs and DSOs. The consortium includes three European TSOs, one of the most important European DSO group, several R&D companies and universities from 8 European Countries (among which the Italian RSE acting as project coordinator) and N-SIDE, the developer of the European market coupling platform EUPHEMIA.

Partners

Table of Contents

List of Abbreviations and Acronyms

Abbreviation/Acronym	Meaning
API	Application Programming Interface
GUI	Graphical User Interface
T&D	Transmission & Distribution
TSO	Transmission System Operator
DSO	Distribution System Operator
R&D	Research & Development
OPF	Optimal Power Flow
GEP	Grid Expansion Planning
FFP	FlexPlan Full Process
PP	Pre-Processor
SR	Scenario Reduction
URL	Uniform Resource Locator
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
CGMES	Common Grid Model Exchange Standard
PST	Phase Shifting Transformer
LM	Lagrange Multiplier
LMP	Locational Marginal Price
AC	Alternating Current
DC	Direct Current
PTDF	Power Transfer Distribution Factor

Executive Summary

One of the main objectives of the FlexPlan project is the development of an innovative grid expansion planning tool. This planning tool is being used by six different regional cases in order to execute the simulations necessary for the testing of the new planning methodology developed in FlexPlan.

This deliverable consists of a comprehensive manual explaining how the grid planner can use the FlexPlan planning tool in order to be assisted in the definition of its grid reinforcement plan.

This documentation is organized around three chapters:

- First, how to use the Application Programming Interface (API) is explained in chapter 1.
- Then the data models are described in chapter 2.
- Finally, chapter 3 provides a comprehensive description of how to use the Graphical User Interface (GUI).

1 API Start Guide

The FlexPlan planning tool provides an HTTPS[1] Application Programming Interface (API) to interact with. The goal of this section is to introduce how this API has been developed and to provide guidelines on how to use it. A complete description of this API as well as of the data models (chapter 2), following OpenAPI specifications, can be found in the appendix.

Five different types of simulations are available to the user as POST[2] requests through the FlexPlan planning tool API:

- *Optimal Power Flow (OPF),* to run a non-expanded optimal power flow
- *Grid Expansion Planning (GEP),* to run a planning problem
- *Pre-Processor (PP),* to run the pre-processor developed by Tecnalia
- *FlexPlan Full Process (FFP),* to run a non-expanded optimal power flow followed by a pre-processor run and then by a planning problem
- *Scenario Reduction (SR),* to run the scenario reduction procedure

The API is secured with basic authentication which means that username and password must be provided to run a simulation.

HTTPS is a widely used communication protocol and so there exist plenty of different tools to send HTTPS requests. In this document, we will provide examples on how to send requests with two different tools: the command line tool *curl* as well as the Python library *requests*. As a third alternative, one could also use a more user-friendly tool with a graphical user interface like Postman[3].

Here, you find an example of a *curl* command, which can be used by the user to start a simulation:

```
curl --request POST \
     --url "https://flexplan-uat.eu.n-side.com/api/v1/optimal-power-flow/start" \
     --header "content-type: application/json" \
     --data @input_file.json \
     --user username:password
```

where `flexplan-uat.eu.n-side.com` must be replaced by the URL communicated by N-SIDE, `optimal-power-flow` must be replaced by the correct string depending on the type of simulation you want to run:

- OPF: `optimal-power-flow`
- GEP: `grid-expansion-planning`
- PP: `pre-processor`
- FFP: `flexplan-full-process`
- SR: `scenario-reduction`

[1] HTTPS is an extension of the HTTP. It combines HTTP with an encryption layer. It is used for secure communication and it why it has been chosen as communication protocol for the FlexPlan planning tool.
[2] https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/POST
[3] https://www.postman.com/

`input_file.json` must be replaced by the path to a file containing the body of your request (see chapter 2 for more details about the data which must be provided) and finally `username:password` must be replaced by your credentials (username and password).

Alternatively, the following *Python* code can also be used for the same purpose:

```
import requests
import json

url = "https://flexplan-uat.n-side.com/api/v1/optimal-power-flow/start"
input_file = "input_file.json"

with open(input_file, "r") as fp:
  payload = json.load(fp)

headers = {
  "Content-Type": "application/json"
}

res = requests.post(url, headers=headers, data=json.dumps(payload),
                    auth=("username", "password"))
print(res.status_code)
print(res.text)
```

Whatever the solution you decide to use to send HTTPS requests (curl, Python, Postman…), you will receive one of the five following responses to your POST request:

- 202 – Accepted: if the request body is valid and the server is not already running a task. In that case, your simulation will be started and you will receive the `task_id` of your simulation as response. For example:

```
{
  "message": "Optimal Power Flow task is successfully launched!",
  "output_file_full_path": "opf/938a036aa1624a49adb9eae36bd9e7e9.json.gz",
  "task_id": "938a036aa1624a49adb9eae36bd9e7e9"
}
```

- 200 – OK: if the request body is valid but the server is already running a task. In that case, your simulation will not be started. Indeed, the FlexPlan planning tool is only able to manage one simulation at a time.
- 400 – Bad Request: if the request body is invalid. In that case, your simulation will not be started and you will receive additional details about the error in the `detail` field of the response. For example:

```
'nbScenarios' is a required property - 'genericParameters'
```

- 401 – Unauthorized: if the provided authorization (credentials) is not valid. In that case, your simulation will not be started.
- 500 – Internal Server Error: if an internal server error occurred. In that case, your simulation will not be started.

In addition to the five POST requests available in the API to launch simulations, the following GET[4] requests are also available:

- *List*, to collect the list of all the `task_id` of the simulation results available on the cloud storage. This allows the user to know if its simulation is finished and that the results are available.
- *Download*, to download the result file of a specific simulation.

Here, you find two examples of a *curl* commands which can be used by the user to either list all the simulation results available on the cloud storage or to download a specific output file:

```
curl --request GET \
     --url "https://flexplan-uat.n-side.com/api/v1/list/request_id" \
     --user username:password
```

where `request_id` must be replaced by one of the following:

- `all`: get list of all available results
- `opf`: get list of available OPF results
- `gep`: get list of available GEP results
- `pp`: get list of available PP results
- `ffp`: get list of available FFP results
- `sr`: get list of available SR results

and

```
curl --request GET \
     --url "https://flexplan-uat.n-side.com/api/v1/download/task_id" \
     --user username:password > task_id.json
```

where `task_id` must be replaced by the one corresponding to the desired simulation results.

Finally, a DELETE[5] request is also available in order to delete specific results. Here is an example of *curl* command having this purpose:

```
curl --request DELETE \
     --url "https://flexplan-uat.n-side.com/api/v1/delete/task_id" \
     --user N-SIDE:password
```

[4] https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/GET
[5] https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/DELETE

2 Data Models

This section has the purpose to give an overview of the data models used in the FlexPlan planning tool. If you want to have a full view on the data models, please check the full API and data model specifications in the appendix. As those specifications follow the OpenAPI standard, there exist multiple free tools to read them with a user-friendly interface. Among them, we recommend either Swagger Editor[6] (web app) or Stoplight Studio[7] (desktop app).

Depending on the type of simulation you want to perform, different data will be required in the input file. All data has been divided in five different categories; each being explained in one of the subparts of this chapter. As mentioned in the chapter 1, the different simulations available are the following:

- Optimal Power Flow (OPF)
- Grid Expansion Planning (GEP)
- Pre-Processor (PP)
- FlexPlan Full Process (FFP)
- Scenario Reduction (SR)

For all of them, generic parameters (part 2.1), grid model input file (part 2.2) and scenario data input file (part 2.3) are required. For the OPF and FFP simulations, contingency states (part 2.5) **can** also **be** added and for the GEP simulation, candidate input (part 2.4) **needs** to be added.

For any type of simulation, the complete data must be provided in JSON format. This format was chosen as it is a customizable, light and easy to understand format. It is also easy for machines to parse and generate and it is the standard format for HTTPS API communication.

On top of the input models, output models are also described in this chapter. This is the case for OPF output (part 2.6) and GEP output (part 2.7).

Figure 2.1 depicts how the different models are being exchanged and processed in the FlexPlan full process.

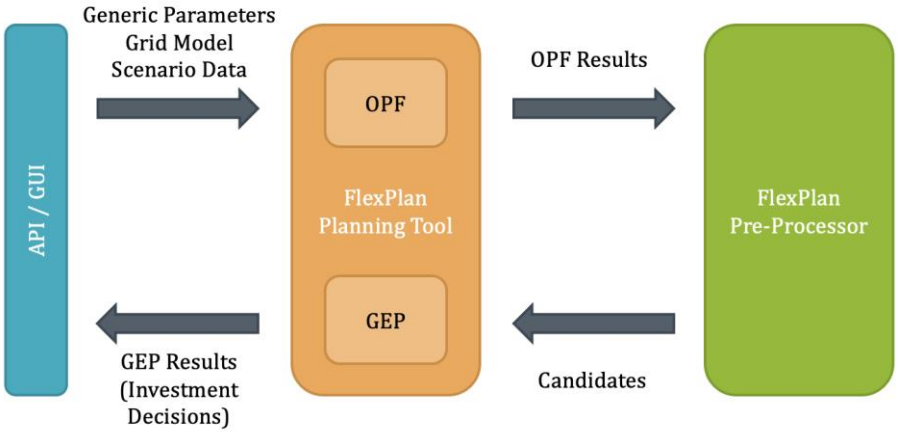

Figure 2.1 – Flow of the FlexPlan Full Process (FFP)

[6] https://editor.swagger.io/
[7] https://stoplight.io/studio

2.1 Generic Parameters

This data model contains all the information to define a simulation. This includes the years considered, the reference year, the number of variants per year, the number of hours per variant and the probabilities of the variants. Additional parameters like the base power, the maximum number of selected candidates and the discount rate can also be added. Moreover, the user can also define some of the CPLEX parameters which will be used to solve the optimization problem. Finally, parameters linked to the T&D decomposition must also be added here. This includes the estimated cost of exchanged energy as well as information about the connection nodes between transmission and distribution.

2.2 Grid Model Input File

This data model contains all the information linked to topology of the power system (grid). This includes:

- AC and DC buses with geographical coordinates, voltage information and characteristics useful for the pre-processor
- AC and DC branches, converters, transformers and PSTs with connected buses and static technical characteristics such as rated power, susceptance, losses, tap ratio, max shift, max angle difference…
- Loads, generators and storage units with connected bus and static characteristics such as capacities, efficiencies, costs, bounds on flexibility…

This data model is similar to the equipment profile (EQ) data files, used in the Common Grid Model Exchange Standard[8] (CGMES) by ENTSO-E.

2.3 Scenario Data Input File

This data model contains all the time-dependent information linked to loads, generators and storage units. This includes:

- Loads: reference demand and bounds on flexibility (if time-dependent)
- Generators: capacity factors for RES generators
- Storage units: initial and final energy levels, power coming from external process, maximum injected and absorbed power (if time-dependent)

This data model is similar to the steady-state hypothesis (SSH) data files, used in the Common Grid Model Exchange Standard (CGMES) by ENTSO-E.

2.4 Candidates Input File

This data model must be provided only in case of a GEP simulation. In the case of a FFP simulation, this data model is generated by the pre-processor. It contains all the potential candidates to reinforce the grid.

[8] https://www.entsoe.eu/data/cim/cim-for-grid-models-exchange/

Those can be classical reinforcement assets such as AC branches (new lines for transmission and replacement of lines for distribution), DC branches with converters, PSTs and transformers or it can also be flexibilization of some existing loads or installation of new storage units.

Compared to the data provided in grid model input file and scenario data input file, this model also expect that every candidate asset is associated with a set of years in which they can be installed, a lifetime and investments costs (year dependent).

2.5 Contingency States

This data model is optional. If provided, it must contain all the contingencies to be considered when performing an optimal power flow. A contingency is a list of assets which could be in outage (out of service) at the same time and for which we want to assess the impact on the operational costs of the system.

2.6 Optimal Power Flow Output File

This data model contains the results of an optimal power flow. First, it includes operational costs such as generation costs, curtailment costs (for both generation and load), costs linked to the use of flexibility (load shifting and shedding) and reliability costs (linked to the contingency states). Then, it provides Locational Marginal Prices (LMPs) for AC and DC buses, Lagrange Multipliers (LMs) for AC and DC branches and operational results for all type of assets such as power flows through branches, generation dispatch, storage usage and use of flexibility. Finally, it also provides the PTDF matrix of the network contained in the grid model input file.

2.7 Grid Expansion Planning Output File

This data model contains the results of a grid expansion planning or of a FlexPlan full process. It contains operational results and operational costs as for the OPF output file. On top of those results, it also provides additional output linked to the planning process: investment decisions for every candidate and associated investment costs

3 GUI Start Guide

As an alternative to the API, the FlexPlan user has the possibility to use the FlexPlan planning tool through an intuitive Graphical User Interface. This interface gives him not only the opportunity to use the same features of the FlexPlan engine than with the API but also to visualize in a nice way the inputs and results of the simulations. This start guide will follow the flow of the application in order to explain its main features.

3.1 Login

As a first step, the user must log into the application. As it is protected with basic authentication, the user has to provide its username and its password. Without authentication, it is not possible to see other views, nor to interact further with the application.

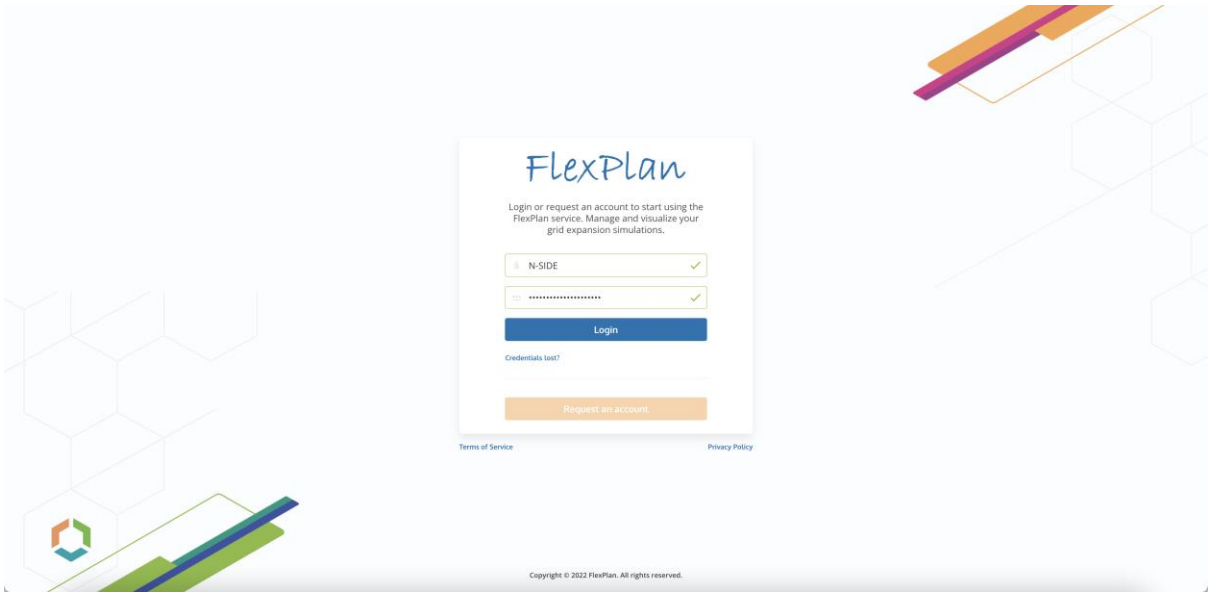

Figure 3.1 – Login screen

3.2 Simulations

After login, the user directly reaches the simulations view. This is also known as the cockpit of the application. From there, the user can first see a list of all its past simulations. It includes several details such as project name (to be used if you want to group simulations together), simulation name, simulation type, status, launch date, total costs and link to the input file. In order to ease the navigation through the past simulations, the user can filter the list of simulations thanks to the search bar. Of course, if the user is accessing the application for the first time the list will be empty.

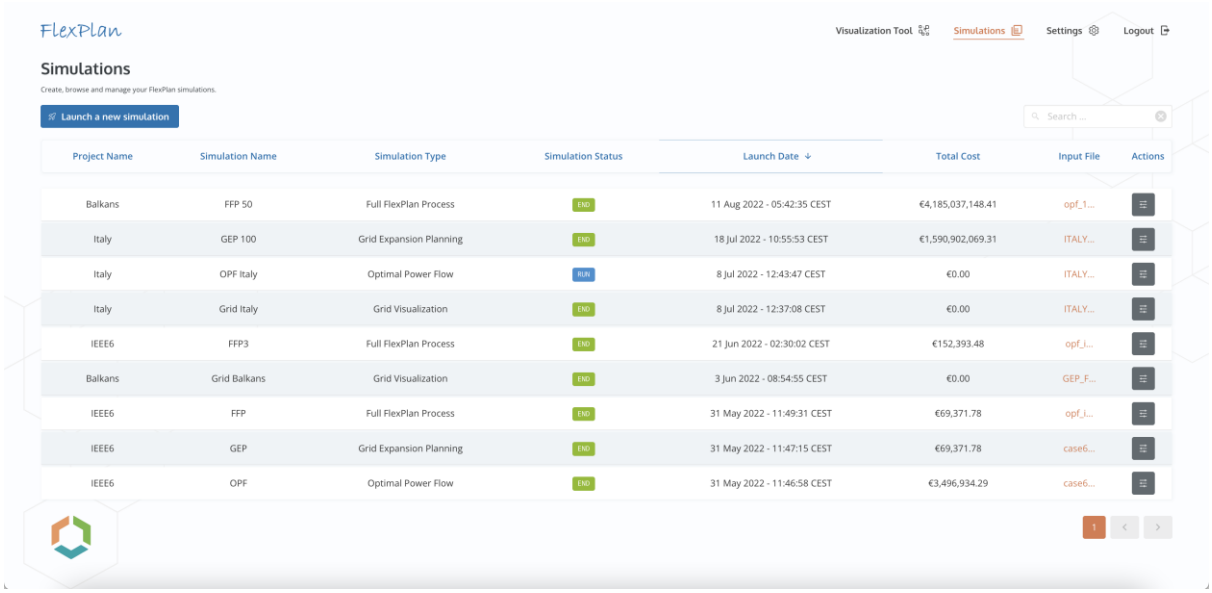

Figure 3.2 – Simulations

By clicking on the button “Launch a new simulation”, the user is able to start a new run (OPF, GEP, FFP or simply grid visualization in order to visually validate the topology of the network). In that case, a modal opens and the user has then two different possibilities:

- either actually starting a new simulation with the FlexPlan planning tool engine by uploading an input file, illustrated on Figure 3.3,
- or opening a past simulation by uploading the input and output files generated by the engine for this simulation, illustrated on Figure 3.4. This option allows the user to visualize results of simulations which were started directly through the API.

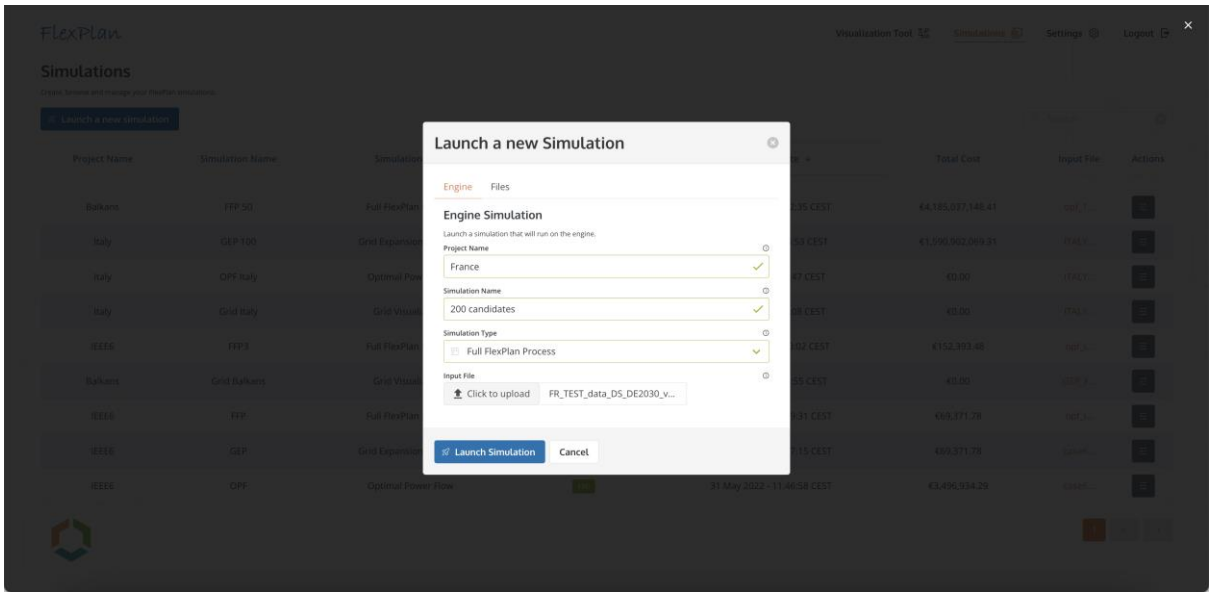

Figure 3.3 – New simulation with engine

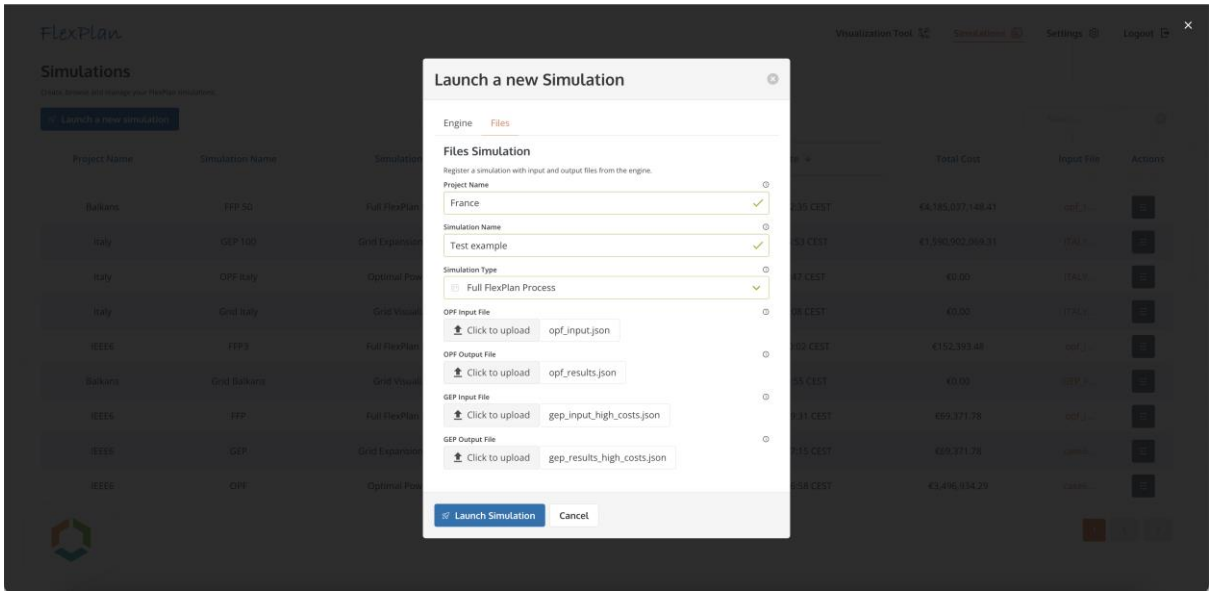

Figure 3.4 – New simulation with files

Back to the simulation list, the user has the possibility to browse the details of a simulation by clicking on the "Actions" button on the row of the simulation of interest. The same button gives the possibility to the user to delete a simulation.

The details panel (illustrated on Figure 3.5) gives very useful high-level information about the results of a simulation. It includes:

- Total costs (including operational costs and investments costs) before and after optimization (before/after investment in new assets)
- Selected candidates (decisions) with investment costs
- General information about the run (metadata and execution times)

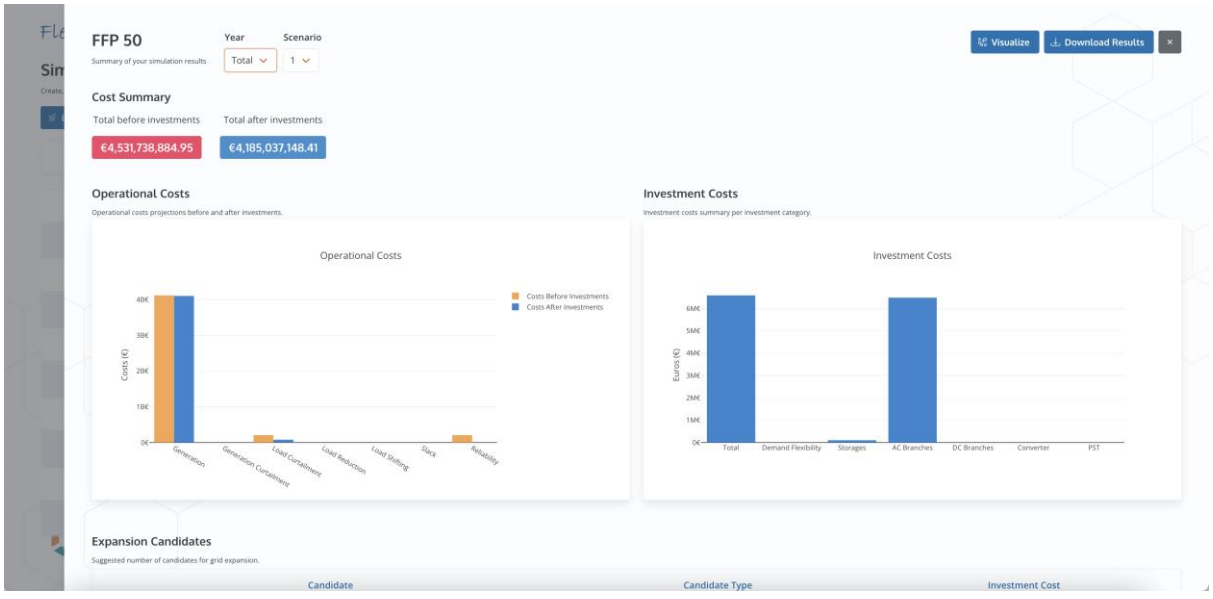

Figure 3.5 – Details panel

3.3 Visualization Tool

Next to the simulations screen, the second main module of the FlexPlan Graphical User Interface (GUI) is the Visualization Tool. This component can be accessed either directly from the details panel of a simulation or by clicking on the tab "Visualization Tool" and selecting the simulation to be visualized, as illustrated on Figure 3.6.

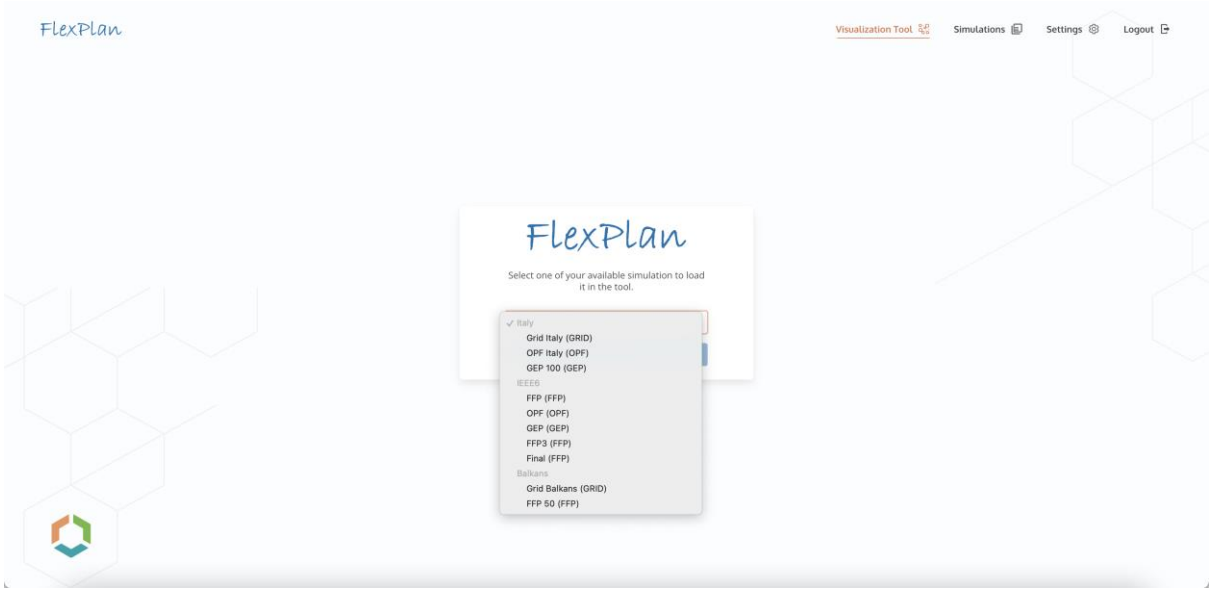

Figure 3.6 – Loading a simulation in the Visualization Tool

The Visualization Tool gives the opportunity to visualize geographically (on a map, with zooming capability) the four steps of the FlexPlan methodology:

- Grid visualization, to validate the topology of the power system, both for Transmission and Distribution
- Optimal power flow, to identify where are the bottlenecks in the grid (congested lines assessed through Lagrange Multipliers and areas with high Locational Marginal Prices)
- Candidates' visualization, to discover and locate the reinforcement candidates (AC & HVDC lines, PSTs, storage units and flexibility programs) proposed by the pre-processor or manually defined
- Grid expansion planning, to visualize the optimal investment plan (minimizing total costs) computed by the FlexPlan grid planning tool

Those four steps are illustrated with a dummy example (based on the IEEE6 grid mapped on six market bidding zones of Italy) on Figures 3.7 to 3.10.

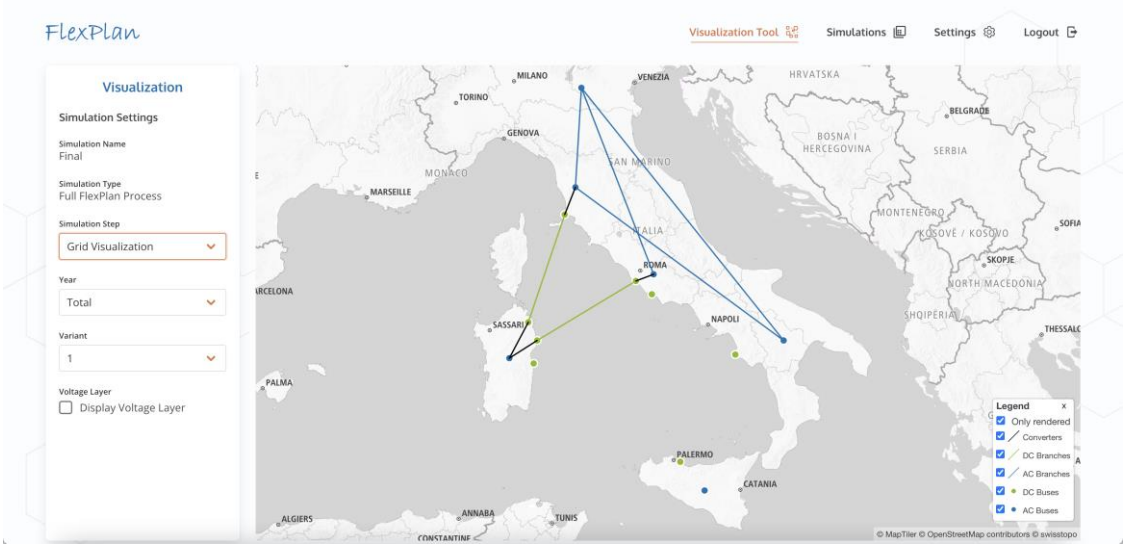

Figure 3.7 – Grid visualization

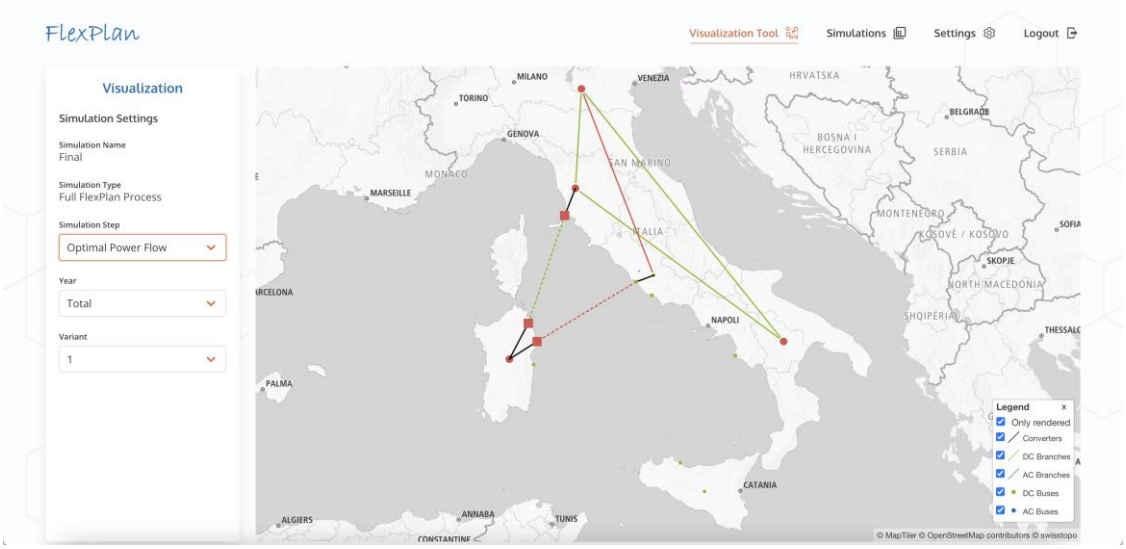

Figure 3.8 – Optimal power flow visualization

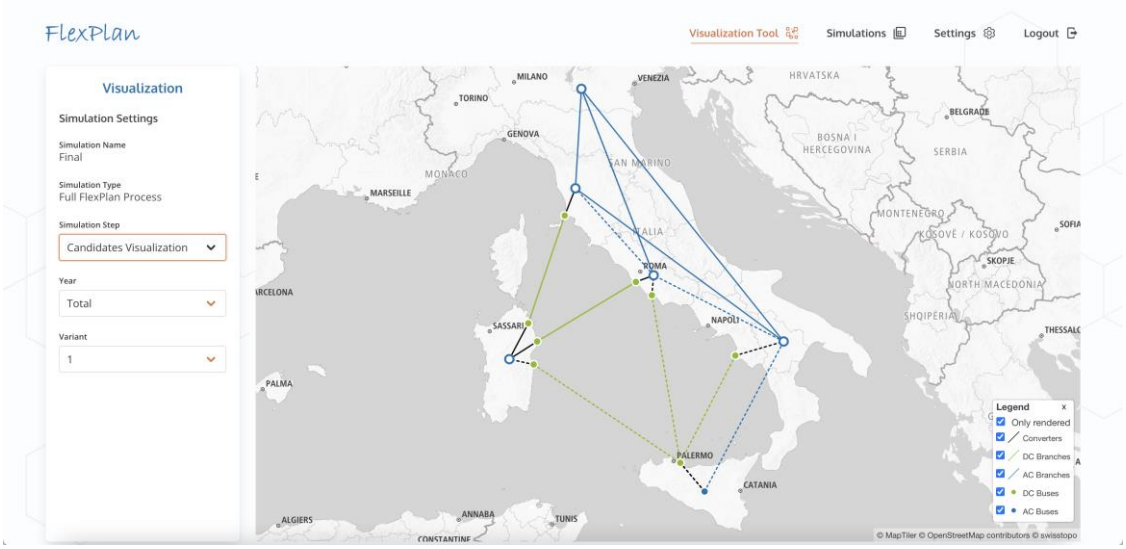

Figure 3.9 – Candidates' visualization

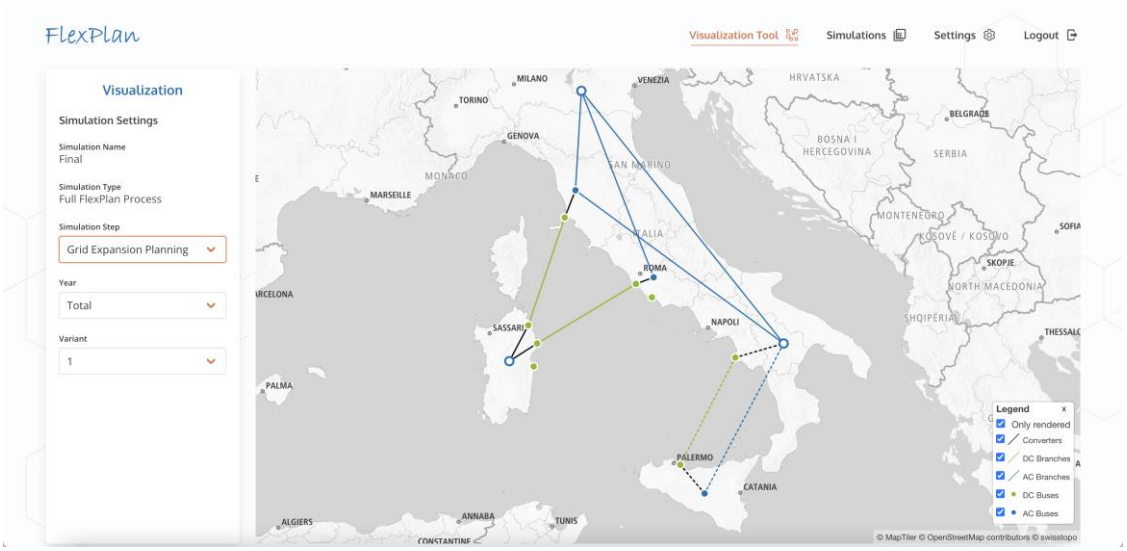

Figure 3.10 – Grid expansion planning visualization

Of course, not only dummy examples can be displayed with the Visualization Tool. Thanks to its advanced technologies and robust performances, it also supports full cases spanning fully one or several countries. This is illustrated on Figure 3.11 with the representation of the Italian transmission grid. Moreover, a voltage layer can be added on the map for the grid visualization in order to distinguish the different voltage levels.

Figure 3.11 – Grid visualization with voltage layer

Finally, by clicking on any asset (branch or bus) on the map at any of the four steps of the FlexPlan methodology, the user can access a details panel giving extra information on the selected asset (existing or candidate) and on any of the components (generators, loads or storage units connected to buses) connected to this asset.

Figure 3.12 shows an example of details panel for an AC bus with in particular a focus on the connected flexible load (invested candidate). Information available includes comparison between the reference demand and the delivered demand for any hour and any year as well as the used flexibility (demand shift, demand reduction and demand curtailment).

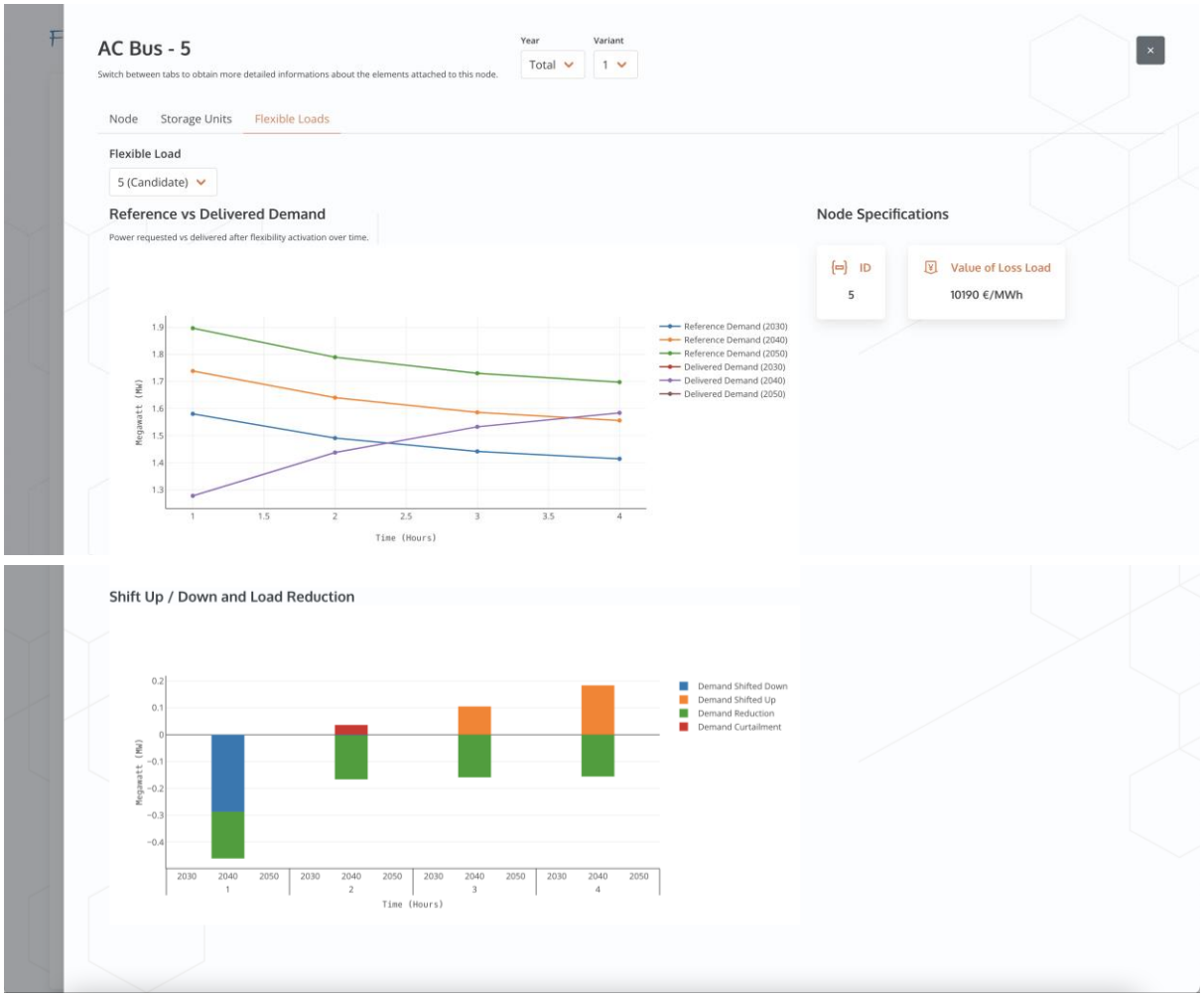

Figure 3.12 – Details panel for AC bus

3.4 Settings

To end the explanations of the FlexPlan planning tool Graphical User Interface, it is also important to mention that the user has access to a last screen by clicking on the button “Settings”. From this view, the user can set some generic application and notification settings as well as updating his password. This is shown on Figures 3.13 and 3.14.

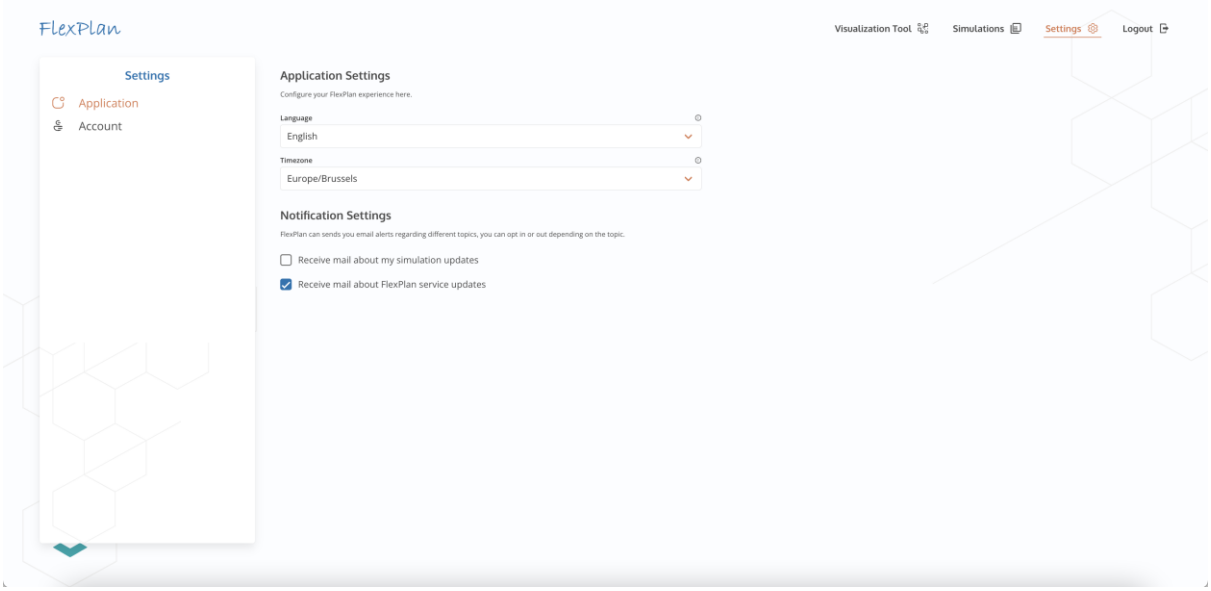

Figure 3.13 – Settings – Application

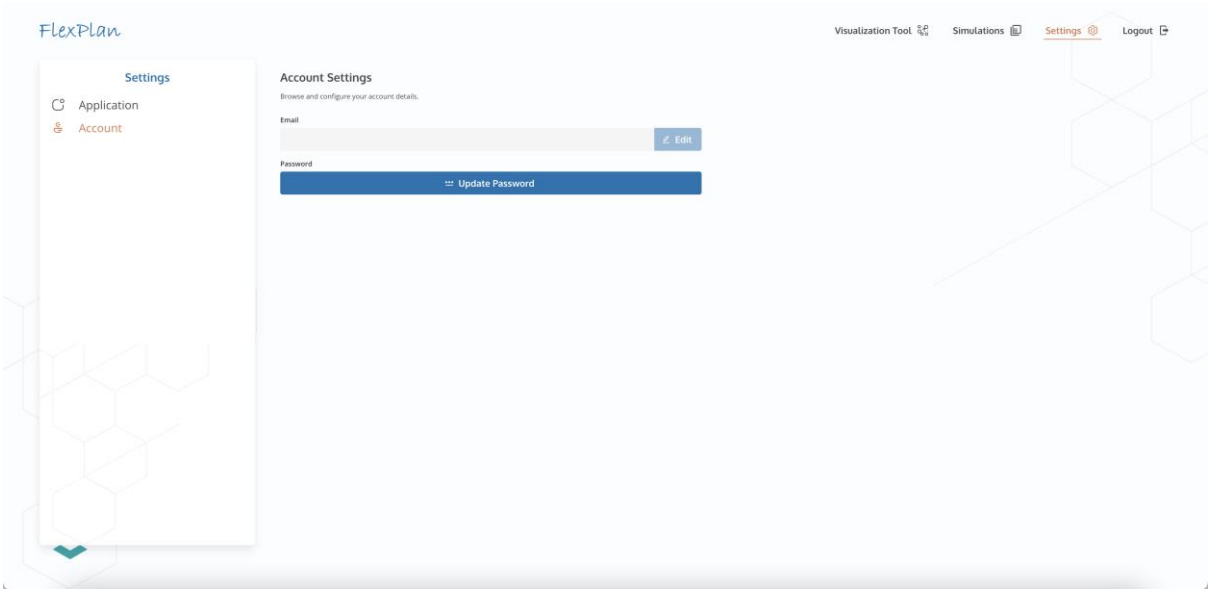

Figure 3.14 – Settings – Account

Appendix

API and Data Model Specifications

```
openapi: 3.0.4
tags:
  - name: optimal-power-flow
    description: Everything about the optimal power flow
  - name: grid-expansion-planning
    description: Everything about the grid expansion planning
  - name: flexplan-full-process
    description: Everything about the FlexPlan full process
  - name: scenario-reduction
    description: Everything about the Scenario Reduction methodology
  - name: pre-processor
    description: Everything about the pre-processor
  - name: list
    description: Everything about listing the results stored in S3
  - name: download
    description: Everything about downloading the results stored in S3
  - name: delete
    description: Everything about deleting the results stored in S3
info:
  title: Customer
  version: '1.0'
  contact:
    name: Maxime Hanot
    email: mha@n-side.com
  description: The API that is made available by the customer to access the FlexPlan optimization
solver.
servers:
  - url: 'https://flexplan.customer.domain/api/v1'
paths:
  /grid-expansion-planning/start:
    post:
      summary: Start a Grid Expansion Planning
      tags:
        - grid-expansion-planning
      operationId: post-grid-expansion-planning-start
      responses:
        '200':
          description: OK - The server is already running a task.
        '202':
          description: Accepted - The server has launched a new Grid Expansion Planning.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/StartResponse'
        '400':
          description: Bad Request - The request body is invalid.
        '500':
          description: Internal Server Error
      security:
        - basic: []
      description: |-
        Request to start a Grid Expansion Planning.

        If the request provides a valid request body and the server is not already running a task,
it will answer with a 202 - Accepted status code.

        If the server is already running a task, it will answer with a 200 - OK status code.
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/GridExpansionPlanningInputFile'
        description: 'The request body should contain the following input data:
GridExpansionPlanningInputFile.'
  /flexplan-full-process/start:
```

```
    post:
      summary: Start a FlexPlan Full Process
      tags:
        - flexplan-full-process
      operationId: post-flexplan-full-process-start
      responses:
        '200':
          description: OK - The server is already running a task.
        '202':
          description: Accepted - The server has launched a new FlexPlan Full Process.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/StartResponse'
        '400':
          description: Bad Request - The request body is invalid.
        '500':
          description: Internal Server Error
      security:
        - basic: []
      description: |-
        Request to start a FlexPlan Full Process.

        If the request provides a valid request body and the server is not already running a task,
it will answer with a 202 - Accepted status code.

        If the server is already running a task, it will answer with a 200 - OK status code.
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/FlexplanFullProcessInputFile'
        description: 'The request body should contain the following input data:
FlexPlanFullProcessInputFile.'
  /optimal-power-flow/start:
    post:
      summary: Start an Optimal Power Flow
      tags:
        - optimal-power-flow
      operationId: post-optimal-power-flow-start
      responses:
        '200':
          description: OK - The server is already running a task.
        '202':
          description: Accepted - The server has launched a new Optimal Power Flow.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/StartResponse'
        '400':
          description: Bad Request - The request body is invalid.
        '500':
          description: Internal Server Error
      security:
        - basic: []
      description: |-
        Request to start an Optimal Power Flow.

        If the request provides a valid request body and the server is not already running a task,
it will answer with a 202 - Accepted status code.

        If the server is already running a task, it will answer with a 200 - OK status code.
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/OptimalPowerFlowInputFile'
        description: 'The request body should contain the following input data:
OptimalPowerFlowInputFile.'
  /scenario-reduction/start:
    post:
      summary: Start a Scenario Reduction
      tags:
```

```
        - scenario-reduction
      operationId: post-scenario-reduction-start
      responses:
        '200':
          description: OK - The server is already running a task.
        '202':
          description: Accepted - The server has launched a new Scenario Reduction.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/StartResponse'
        '400':
          description: Bad Request - The request body is invalid.
        '500':
          description: Internal Server Error
      security:
        - basic: []
      description: |-
        Request to start an Scenario Reduction.

        If the request provides a valid request body and the server is not already running a task,
it will answer with a 202 - Accepted status code.

        If the server is already running a task, it will answer with a 200 - OK status code.
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ScenarioReductionInputFile'
        description: 'The request body should contain the following input data:
ScenarioReductionInputFile.'
  /pre-processor/start:
    post:
      summary: Start a Pre-Processor run
      tags:
        - pre-processor
      operationId: post-pre-processor-start
      responses:
        '200':
          description: OK - The server is already running a task.
        '202':
          description: Accepted - The server has launched a new Pre-Processor run.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/StartResponse'
        '400':
          description: Bad Request - The request body is invalid.
        '500':
          description: Internal Server Error
      security:
        - basic: []
      description: |-
        Request to start a Pre-Processor run.

        If the request provides a valid request body and the server is not already running a task,
it will answer with a 202 - Accepted status code.

        If the server is already running a task, it will answer with a 200 - OK status code.
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/PreProcessorInputFile'
        description: 'The request body should contain the following input data:
PreProcessorInputFile.'
  '/list/{requestId}':
    get:
      summary: List Results from S3 storage for a given Request ID or for all Requests
      tags:
        - list
      responses:
        '202':
```

```
          description: Accepted
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ListResults'
        '500':
          description: Internal Server Error
      operationId: list_res
      security:
        - basic: []
      description: This request will return the list of results for the given request id from the S3
storage.
    parameters:
      - schema:
          type: string
        name: requestId
        in: path
        description: This is the request id used to identify the right API request among many.
        required: true
  '/download/{taskId}':
    parameters:
      - schema:
          type: string
        name: taskId
        in: path
        required: true
        description: This is the task id of the results to be downloaded.
    get:
      summary: Download Results from S3 storage for the given task id
      operationId: download
      responses:
        '202':
          description: Accepted
          content:
            application/gzip:
              schema:
                type: string
                format: binary
        '404':
          description: The result file for the given task id does not exist.
          content:
            application/gzip:
              schema:
                type: string
                format: binary
        '500':
          description: Internal Server Error
      description: This request will return the result file as a JSON given the file name located in
the S3 storage.
      security:
        - basic: []
      tags:
        - download
  '/delete/{taskId}':
    parameters:
      - schema:
          type: string
        name: taskId
        in: path
        required: true
        description: This is the task id of the results to be deleted.
    delete:
      summary: Delete result file from S3 storage for the given task id
      operationId: delete-taskId
      responses:
        '202':
          description: The results file corresponding to the given task id is deleted.
        '404':
          description: The result file for the given task id does not exist.
        '500':
          description: Internal Server Error
      description: This request will delete the result file from the S3 storage given the task id.
      security:
```

```
        - basic: []
      tags:
        - delete
components:
  securitySchemes:
    basic:
      type: http
      scheme: basic
      x-basicInfoFunc: app.basic_auth
  schemas:
    GridModelInputFile:
      type: object
      description: 'Structure describing grid model, such as it can be used by the Planning Tool to
run an Optimal Power Flow'
      title: Grid Model Input File
      properties:
        acBuses:
          type: array
          description: List of AC buses in the power system
          items:
            $ref: '#/components/schemas/AcBus'
        dcBuses:
          type: array
          description: List of DC buses in the power system
          items:
            $ref: '#/components/schemas/DcBus'
        acBranches:
          type: array
          description: List of AC branches in the power system
          items:
            $ref: '#/components/schemas/AcBranch'
        dcBranches:
          type: array
          description: List of DC branches in the power system
          items:
            $ref: '#/components/schemas/DcBranch'
        converters:
          type: array
          description: List of AC/DC converters in the power system
          items:
            $ref: '#/components/schemas/Converter'
        transformers:
          type: array
          description: List of standard transformers in the power system
          items:
            $ref: '#/components/schemas/AcBranch'
        storage:
          type: array
          description: List of storage devices in the power system
          items:
            $ref: '#/components/schemas/Storage'
        generators:
          type: array
          description: List of thermal generators in the power system
          items:
            $ref: '#/components/schemas/Generator'
        loads:
          description: List of loads in the power system
          type: array
          items:
            $ref: '#/components/schemas/FlexibleLoad'
        psts:
          type: array
          description: List of PST present in the power system
          items:
            $ref: '#/components/schemas/Pst'
        acBranchesPreCandidates:
          type: array
          description: List of potential AC branches candidates to share with pre-processor
          items:
            $ref: '#/components/schemas/AcBranchPreCandidate'
        dcBranchesPreCandidates:
          type: array
```

```
          description: List of potential DC branches candidates to share with pre-processor
          items:
            $ref: '#/components/schemas/DcBranchPreCandidate'
      required:
        - acBuses
        - dcBuses
        - acBranches
        - dcBranches
        - converters
        - transformers
        - storage
        - generators
        - loads
        - psts
    ScenarioDataInputFile:
      title: Scenario Data Input File
      type: object
      description: 'Input file containing the scenario data needed to complete grid model. '
      properties:
        storage:
          type: array
          description: List of scenario data for storage systems
          items:
            $ref: '#/components/schemas/StorageScenario'
        loads:
          type: array
          description: List of scenario data for loads
          items:
            $ref: '#/components/schemas/FlexibleLoadScenario'
        generators:
          type: array
          description: List of scenario data for thermal generators
          items:
            $ref: '#/components/schemas/GeneratorScenario'
      required:
        - storage
        - loads
        - generators
    CandidatesInputFile:
      title: Candidates Input File
      type: object
      description: 'Structure describing the input file expected from the pre-processor tool, which
includes the potential investment candidates, alongside with its technical details and costs'
      properties:
        acBranches:
          type: array
          description: List of AC branches candidates
          items:
            $ref: '#/components/schemas/AcBranchCandidate'
        dcBranches:
          type: array
          description: List of DC branches candidates
          items:
            $ref: '#/components/schemas/DcBranchCandidate'
        converters:
          type: array
          description: List of AC/DC converters candidates
          items:
            $ref: '#/components/schemas/ConverterCandidate'
        psts:
          type: array
          description: List of PSTs candidates
          items:
            $ref: '#/components/schemas/PstCandidate'
        storage:
          type: array
          description: List of storage candidates
          items:
            $ref: '#/components/schemas/StorageCandidate'
        flexibleLoads:
          type: array
          description: List of Flexible loads candidates
          items:
```

```
            $ref: '#/components/schemas/FlexibleLoadCandidate'
        transformers:
          type: array
          description: List of generic transformers candidates
          items:
            $ref: '#/components/schemas/AcBranchCandidate'
      required:
        - acBranches
        - dcBranches
        - converters
        - psts
        - storage
        - flexibleLoads
        - transformers
    AcBus:
      title: AC Bus
      type: object
      description: Required inputs to model a generic AC bus/node
      properties:
        id:
          type: string
          description: AC bus ID
        location:
          type: array
          minItems: 2
          maxItems: 2
          description: 'Geographic coordinates to locate the bus: [Latitude [deg], Longitude [deg]]'
          items:
            type: number
        nominalVoltageMagnitude:
          type: number
          description: 'Nominal AC bus voltage magntiude [p.u.]'
          minimum: 0
        slackStatus:
          type: boolean
          description: 'Shows if slack costs are used at the bus for its nodal balance constraint.
False = No use of slack, True = Use of slack'
        maxVoltageMagnitude:
          type: number
          description: 'Maximum voltage magnitude [p.u.]. Only needed if isTransmission == False'
          minimum: 0
        minVoltageMagnitude:
          type: number
          description: 'Minimum Voltage magnitude [p.u.]. Only needed if isTransmission == False'
          minimum: 0
        isTransmission:
          description: 'True=Transmission, False=Distribution'
          type: boolean
        busType:
          type: integer
          description: 'Bus type code, where 1=PQ bus (load bus), 2=PV bus (generator bus),
3=Reference bus (slack bus), 4=Disconnected (isolated bus). Only needed when isTransmission ==
False'
          minimum: 1
          maximum: 4
        charateristics:
          $ref: '#/components/schemas/BusCharacteristics'
        baseVoltage:
          type: number
          description: 'Base voltage used for the per-unit notation [kV]'
        distributionNetworkId:
          type: string
          description: 'The ID of the distribution network, used for the T&D decomposition.'
      required:
        - id
        - location
        - nominalVoltageMagnitude
        - slackStatus
        - isTransmission
        - baseVoltage
    DcBus:
      title: DC Bus
      type: object
```

```
      description: Required inputs to model a generic DC bus/node
      properties:
        id:
          type: string
          description: DC bus ID
        location:
          type: array
          minItems: 2
          maxItems: 2
          description: 'Geographic coordinates to locate the bus: [Latitude [deg], Longitude [deg]]'
          items:
            type: number
        nominalVoltageMagnitude:
          type: number
          description: 'Nominal DC bus voltage magntiude [p.u.]'
          minimum: 0
        slackStatus:
          type: boolean
          description: 'Shows if slack costs are used at the bus for its nodal balance constraint
False = No use of slack, True = Use of slack'
        characteristics:
          $ref: '#/components/schemas/BusCharacteristics'
        baseVoltage:
          type: number
          description: 'Base voltage used for the per-unit notation [kV]'
      required:
        - id
        - location
        - nominalVoltageMagnitude
        - slackStatus
        - baseVoltage
    OptimalPowerFlowOutputFile:
      title: Optimal Power Flow Output File
      type: object
      description: 'Structure describing the output of the Optimal Power Flow run by the Planning
Tool, given a Grid model and Scenario data which satisfies the requirements '
      properties:
        message:
          type: string
        executionTime:
          type: number
        costs:
          $ref: '#/components/schemas/CostsOPF'
        acBuses:
          type: array
          description: 'Array of the different AC buses of the system, including its operational
results'
          items:
            $ref: '#/components/schemas/AcBusOpf'
        dcBuses:
          type: array
          description: 'Array of the different DC buses of the system, including its operational
results'
          items:
            $ref: '#/components/schemas/DcBusOpf'
        acBranches:
          type: array
          description: 'Array of the different AC branches of the system, including its operational
results'
          items:
            $ref: '#/components/schemas/AcBranchOpf'
        dcBranches:
          type: array
          description: 'Array of the different DC branches of the system, including its operational
results'
          items:
            $ref: '#/components/schemas/DcBranchOpf'
        converters:
          type: array
          description: 'Array of the different AC/DC converters of the system, including its
operational results'
          items:
            $ref: '#/components/schemas/ConverterOpf'
```

```
        psts:
          type: array
          description: 'Array of the different PSTs of the system, including its operational
results'
          items:
            $ref: '#/components/schemas/PstOpf'
        transformers:
          type: array
          description: 'Array of the different generic transformers of the system, including its
operational results'
          items:
            $ref: '#/components/schemas/AcBranchOpf'
        generators:
          type: array
          description: 'Array of the different generators of the system, including its operational
results'
          items:
            $ref: '#/components/schemas/GeneratorOpf'
        loads:
          type: array
          description: 'Array of the different flexible loads of the system, including its
operational results'
          items:
            $ref: '#/components/schemas/FlexibleLoadOpf'
        storage:
          type: array
          description: 'Array of the different storages of the system, including its operational
results'
          items:
            $ref: '#/components/schemas/StorageOpf'
        ptdf:
          $ref: '#/components/schemas/PtdfMatrix'
      required:
        - message
        - costs
        - acBuses
        - dcBuses
        - acBranches
        - dcBranches
        - converters
        - psts
        - transformers
        - ptdf
    GridExpansionPlanningOutputFile:
      title: Grid Expansion Planning Output File
      type: object
      description: 'Structure describing the output of the Transmission Network Expansion Problem,
i.e. the investment decisions after running the optimization algorithm.'
      properties:
        message:
          type: string
        executionTime:
          type: number
        costs:
          $ref: '#/components/schemas/CostsGEP'
        acBranchesDecisions:
          type: array
          description: List of investment decisions on AC Branches
          items:
            $ref: '#/components/schemas/AcBranchDecision'
        dcBranchesDecisions:
          type: array
          description: List of investment decisions on DC Branches
          items:
            $ref: '#/components/schemas/DcBranchDecision'
        convertersDecisions:
          type: array
          description: List of investment decisions on AC/DC Converters
          items:
            $ref: '#/components/schemas/ConverterDecision'
        pstsDecisions:
          type: array
          description: List of investment decisions on PST
```

```
          items:
            $ref: '#/components/schemas/PstDecision'
        storageDecisions:
          type: array
          description: List of investment decisions on storage devices
          items:
            $ref: '#/components/schemas/StorageDecision'
        flexibleLoadsDecisions:
          type: array
          description: List of investment decisions on flexible loads
          items:
            $ref: '#/components/schemas/FlexibleLoadDecision'
        transformersDecisions:
          type: array
          description: List of investment decisions on generic transformers
          items:
            $ref: '#/components/schemas/AcBranchDecision'
        acBuses:
          type: array
          description: 'Array of the different AC buses of the system, including its operational
results'
          items:
            $ref: '#/components/schemas/AcBusOpf'
        dcBuses:
          type: array
          description: 'Array of the different DC buses of the system, including its operational
results'
          items:
            $ref: '#/components/schemas/DcBusOpf'
        acBranches:
          type: array
          description: 'Array of the different AC branches of the system (built and existing),
including its operational results'
          items:
            $ref: '#/components/schemas/AcBranchOpf'
        dcBranches:
          type: array
          description: 'Array of the different DC branches of the system (built and existing),
including its operational results'
          items:
            $ref: '#/components/schemas/DcBranchOpf'
        converters:
          type: array
          description: 'Array of the different converters of the system (built and existing),
including its operational results'
          items:
            $ref: '#/components/schemas/ConverterOpf'
        psts:
          type: array
          description: 'Array of the different PSTs of the system (built and existing), including
its operational results'
          items:
            $ref: '#/components/schemas/PstOpf'
        transformers:
          type: array
          description: 'Array of the different transformers of the system (built and existing),
including its operational results'
          items:
            $ref: '#/components/schemas/AcBranchOpf'
        loads:
          type: array
          description: 'Array of the different flexible loads of the system (built and existing),
including its operational results'
          items:
            $ref: '#/components/schemas/FlexibleLoadOpf'
        storage:
          type: array
          description: 'Array of the different storage units of the system (built and existing),
including its operational results'
          items:
            $ref: '#/components/schemas/StorageOpf'
        generators:
          type: array
```

```
          description: 'Array of the different generators of the system, including its operational
results'
          items:
            $ref: '#/components/schemas/GeneratorOpf'
        mipGap:
          type: number
          description: RELATIVE gap between the best integer objective and the objective of the best
node remaining. Only for Transmission if the T&D decomposition is used.
        status:
          type: string
          description: Status of the solution produced by CPLEX. Only for Transmission if the T&D
decomposition is used.
      required:
        - message
        - costs
        - acBranchesDecisions
        - dcBranchesDecisions
        - convertersDecisions
        - pstsDecisions
        - storageDecisions
        - flexibleLoadsDecisions
        - transformersDecisions
        - acBuses
        - dcBuses
        - acBranches
        - dcBranches
        - converters
        - psts
        - transformers
    Generator:
      title: Generator
      type: object
      description: Required inputs to model a generic thermal generator or RES
      properties:
        id:
          type: string
          description: Generator ID
        acBusConnected:
          description: AC bus connected
          type: string
        maxActivePower:
          description: 'Maximum rated active power [p.u.] for each year of the planning horizon'
          type: array
          items:
            type: number
        minActivePower:
          description: 'Minimum rated active power [p.u.] for each year of the planning horizon'
          type: array
          items:
            type: number
        maxReactivePower:
          description: 'Maximum rated reactive power [p.u.] for each year of the planning horizon'
          type: array
          items:
            type: number
        minReactivePower:
          type: array
          description: 'Minimum rated reactive power [p.u.] for each year of the planning horizon'
          items:
            type: number
        meanTimeToRepair:
          type: number
          description: 'Mean time to repair [h]'
          minimum: 0
        failureRate:
          type: number
          minimum: 0
          description: 'Failure rate [1/year]'
        generationCosts:
          type: array
          description: 'Generation costs [€/p.u.]'
          items:
            type: number
```

```
        curtailmentCosts:
          type: array
          description: 'Curtailment costs [€/p.u.]'
          items:
            type: number
        isSurrogate:
          type: boolean
          default: false
          description: Used only internally for T&D decomposition.
      required:
        - id
        - acBusConnected
        - maxActivePower
        - minActivePower
        - maxReactivePower
        - minReactivePower
        - generationCosts
        - curtailmentCosts
    FlexibleLoadCandidate:
      title: Flexible Load Candidate
      type: object
      description: 'Generic model for load flexibility candidate, can be applied to Electric
Vehicles, Demand Response or Thermal Loads'
      properties:
        load:
          $ref: '#/components/schemas/FlexibleLoad'
        invCost:
          description: 'Flexibilization investment costs [€] for each year of the planning horizon
at which the candidate can be invested'
          type: array
          items:
            type: number
        lifetime:
          type: integer
          description: Lifetime of the candidate from its investment
          multipleOf: 10
        horizons:
          type: array
          description: All years of the planning horizon at which the candidate can be invested
          uniqueItems: true
          items:
            type: integer
        isUnique:
          type: boolean
          default: false
          description: Whether the candidate can be invested at several years of the planning
horizon (false) or not (true)
      required:
        - load
        - invCost
        - lifetime
        - horizons
    Storage:
      title: Storage
      type: object
      description: 'Required inputs to model a generic storage model, can be used for all kind of
storage (Reservoir Hydro, Battery Energy Storage Systems, Thermo-electric storage...)'
      properties:
        id:
          type: string
          description: Storage ID
        acBusConnected:
          type: string
          description: AC Bus Connected
        maxEnergy:
          description: 'Maximum Energy Content [p.u.] for each year of the planning horizon'
          type: array
          items:
            type: number
        selfDischargeRate:
          description: 'Self-discharge rate [0, 1] of the storage asset for each year of the
planning horizon. When not relevant, it should be set to zero'
          type: array
```

```
          items:
            type: number
        minEnergy:
          type: array
          description: 'Minimum Energy Content [p.u.] for each year of the planning horizon'
          items:
            type: number
        maxEnergyYear:
          description: 'Maximum abosrbed energy over a year [p.u.] for each year of the planning
horizon'
          type: array
          items:
            type: number
        absEfficiency:
          description: 'Absorption efficiency [0, 1] for each year of the planning horizon'
          type: array
          items:
            type: number
        injEfficiency:
          description: 'Injection efficiency between 0 and 1 [-] for each year of the planning
horizon'
          type: array
          items:
            type: number
        maxAbsRamp:
          description: 'Maximum absorption ramp rate [p.u.] for each year of the planning horizon'
          type: array
          items:
            type: number
        maxInjRamp:
          description: 'Maximum injection ramp rate [p.u.] for each year of the planning horizon'
          type: array
          items:
            type: number
        maxReactivePowerExchange:
          type: array
          description: 'Maximum reactive power exchange [p.u.] for each year of the planning
horizon'
          items:
            type: number
        minReactivePowerExchange:
          type: array
          description: 'Minimum reactive power exchange [p.u.] for each year of the planning
horizon'
          items:
            type: number
        maxInjActivePower:
          type: array
          description: 'Maximum injected active power [p.u.]'
          items:
            type: number
        maxAbsActivePower:
          type: array
          description: 'Maximum absorbed active power [p.u.]'
          items:
            type: number
        isSurrogate:
          type: boolean
          default: false
          description: Used only internally for T&D decomposition.
      required:
        - id
        - acBusConnected
        - maxEnergy
        - selfDischargeRate
        - minEnergy
        - absEfficiency
        - injEfficiency
        - maxReactivePowerExchange
        - minReactivePowerExchange
        - maxInjActivePower
        - maxAbsActivePower
    AcBranch:
```

```
      title: AC Branch
      type: object
      description: Required inputs to model a generic AC transmission/distribution line as well as a
standard transformer
      properties:
        id:
          type: string
          description: AC branch ID
        acBusOrigin:
          type: string
          description: AC bus connected origin
        acBusExtremity:
          type: string
          description: AC bus connected extremity
        isTransmission:
          type: boolean
          description: 'True=Transmission, False=Distribution'
        susceptance:
          type: number
          description: 'Inductive series admittance [p.u.], i.e. not the shunt susceptance, but the
real-valued susceptance, with siemens as absolute unit'
        voltageTapRatio:
          type: number
          description: 'Voltage tap ratio [-]. Only used when modeling a standard transformer'
          minimum: 0
          exclusiveMinimum: true
        maxAngleDifference:
          type: number
          description: 'Maximum voltage angle difference [rad]'
        minAngleDifference:
          type: number
          description: 'Minimum voltage angle difference [rad]'
        resistance:
          type: number
          description: 'Resistance [p.u.], only needed if isTransmission == False'
          minimum: 0
        reactance:
          type: number
          description: 'Reactance [p.u.]'
          minimum: 0
        meanTimeToRepair:
          type: number
          description: 'Mean time to repair [h]'
          minimum: 0
        failureRate:
          type: number
          description: 'Failure rate [1/year]'
          minimum: 0
        emergencyRating:
          type: number
          description: 'Emergency rating [p.u.] to be used for contingency analysis'
          minimum: 0
        ratedApparentPower:
          type: array
          description: 'Rated apparent power of the line [p.u.] for each year of the planning
horizon'
          items:
            type: number
            minimum: 0
        isInterconnection:
          type: boolean
          description: 'True=the branch is an interconnection between two countries, False=normal
branch. Default = False'
          default: false
        length:
          type: number
          description: 'Length of the line [km]'
          minimum: 0
        distributionNetworkId:
          type: string
          description: 'The ID of the distribution network, used for the T&D decomposition.'
      required:
        - id
```

```
        - acBusOrigin
        - acBusExtremity
        - isTransmission
        - susceptance
        - voltageTapRatio
        - maxAngleDifference
        - minAngleDifference
        - reactance
        - meanTimeToRepair
        - failureRate
        - emergencyRating
        - ratedApparentPower
    DcBranch:
      title: DC Branch
      type: object
      description: Required inputs to model generic DC transmission line (HVDC)
      properties:
        id:
          type: string
          description: DC branch ID
        dcBusOrigin:
          type: string
          description: DC bus connected origin
        dcBusExtremity:
          type: string
          description: DC bus connected extremity
        ratedActivePower:
          description: 'Rated active power DC [p.u.] for each year of the planning horizon'
          type: array
          items:
            type: number
            minimum: 0
        meanTimeToRepair:
          type: number
          description: 'Mean time to repair [h]'
          minimum: 0
        failureRate:
          type: number
          description: 'Failure rate [1/year]'
          minimum: 0
        emergencyRating:
          type: number
          description: 'Emergency rating [p.u.] to be used for contingency analysis'
          minimum: 0
        isInterconnection:
          type: boolean
          description: 'True=the branch is an interconnection between two countries, False=normal
branch'
          default: false
        length:
          type: number
          description: 'Length of the line [km]'
          minimum: 0
      required:
        - id
        - dcBusOrigin
        - dcBusExtremity
        - ratedActivePower
        - meanTimeToRepair
        - failureRate
        - emergencyRating
    Converter:
      title: Converter
      type: object
      description: Required inputs to model a generic AC/DC converter
      properties:
        id:
          type: string
          description: AC/DC converter ID
        acBusConnected:
          type: string
          description: AC bus connected
        dcBusConnected:
```

```
          type: string
          description: DC bus connected
        auxiliaryLosses:
          description: 'Auxiliary converter losses [p.u.] for each year of the planning horizon'
          type: array
          items:
            type: number
        linearLosses:
          description: 'Linear converter losses [p.u./p.u.], equivalent to [MW/MW], for each year of
the planning horizon'
          type: array
          items:
            type: number
        ratedActivePowerAC:
          type: array
          description: 'Maximum rated active power AC [p.u.] for each year of the planning horizon'
          items:
            type: number
            minimum: 0
        ratedActivePowerDC:
          description: 'Maximum rated active power DC [p.u.] for each year of the planning horizon'
          type: array
          items:
            type: number
            minimum: 0
        meanTimeToRepair:
          type: number
          description: 'Mean time to repair [h]'
          minimum: 0
        failureRate:
          type: number
          description: 'Failure rate [1/year]'
          minimum: 0
        emergencyRating:
          type: number
          description: 'Emergency rating [p.u.] to be used for contingency analysis'
          minimum: 0
      required:
        - id
        - acBusConnected
        - dcBusConnected
        - auxiliaryLosses
        - linearLosses
        - ratedActivePowerAC
        - ratedActivePowerDC
        - meanTimeToRepair
        - failureRate
        - emergencyRating
    Pst:
      title: PST
      type: object
      description: Required inputs to model a generic Phase Shift Transformer model
      properties:
        id:
          type: string
          description: PST ID
        acBusOrigin:
          type: string
          description: 'AC bus connected origin '
        acBusExtremity:
          type: string
          description: AC bus connected extremity
        susceptance:
          type: number
          description: 'Inductive series admittance [p.u.], i.e. not the shunt susceptance, but the
real-valued susceptance, with siemens as absolute unit'
        ratedActivePowerAC:
          type: array
          description: 'Maximum rated active power AC [p.u.] for each year of the planning horizon'
          items:
            type: number
        maxAngleDifference:
          type: number
```

```
        description: 'Maximum voltage angle difference [rad]'
      minAngleDifference:
        type: number
        description: 'Minimum voltage angle difference [rad]'
      maxPhaseShift:
        type: number
        description: 'Maximum phase shift [rad]'
      minPhaseShift:
        type: number
        description: 'Minimum phase shift [rad]'
      meanTimeToRepair:
        type: number
        description: 'Mean time to repair [h]'
        minimum: 0
      failureRate:
        type: number
        description: 'Failure rate [1/year]'
        minimum: 0
      emergencyRating:
        type: number
        description: 'Emergency rating [p.u.] to be used for contingency analysis'
        minimum: 0
    required:
      - id
      - acBusOrigin
      - acBusExtremity
      - susceptance
      - ratedActivePowerAC
      - maxAngleDifference
      - minAngleDifference
      - maxPhaseShift
      - minPhaseShift
      - meanTimeToRepair
      - failureRate
      - emergencyRating
  AcBusOpf:
    title: AC Bus OPF
    type: object
    description: OPF results for generic AC bus
    properties:
      id:
        type: string
        description: AC bus ID
      lmp:
        type: array
        description: 'Locational Marginal Price [€/p.u.]'
        items:
          type: array
          items:
            type: array
            items:
              type: number
      slack:
        type: array
        description: 'Nodal Slack [p.u.]'
        items:
          type: array
          items:
            type: array
            items:
              type: number
    required:
      - id
      - lmp
  DcBusOpf:
    title: DC Bus OPF
    type: object
    description: OPF results for generic DC bus
    properties:
      id:
        type: string
        description: DC bus ID
      lmp:
```

```
        type: array
        description: 'Locational Marginal Price [€/p.u.]'
        items:
          type: array
          items:
            type: array
            items:
              type: array
              items:
                type: number
      slack:
        type: array
        description: 'Nodal Slack [p.u.]'
        items:
          type: array
          items:
            type: array
            items:
              type: array
              items:
                type: number
    required:
      - id
      - lmp
  DcBranchOpf:
    title: DC Branch OPF
    type: object
    description: OPF results for generic DC branch
    properties:
      id:
        type: string
        description: DC branch ID
      dcPowerFlow:
        description: 'DC Power Flow [p.u.]'
        type: array
        items:
          type: array
          items:
            type: array
            items:
              type: number
      lm:
        type: array
        description: 'Lagrange Mutiplier associated with the DC branch [€/p.u.]'
        items:
          type: array
          items:
            type: array
            items:
              type: number
    required:
      - id
      - dcPowerFlow
      - lm
  ConverterOpf:
    title: Converter OPF
    type: object
    description: 'OPF results for generic AC/DC converter '
    properties:
      id:
        type: string
        description: AC/DC converter ID
      acPowerFlow:
        description: 'AC power flow [p.u.]'
        type: array
        items:
          type: array
          items:
            type: array
            items:
              type: number
      dcPowerFlow:
        description: 'DC power flow [p.u.]'
        type: array
        items:
          type: array
          items:
```

```
            type: array
            items:
              type: number
    required:
      - id
      - acPowerFlow
      - dcPowerFlow
  PstOpf:
    title: PST OPF
    type: object
    description: OPF results for generic PST transformer
    properties:
      id:
        type: string
        description: PST ID
      acPowerFlow:
        type: array
        description: 'AC Power Flow [p.u.]'
        items:
          type: array
          items:
            type: array
            items:
              type: number
    required:
      - id
      - acPowerFlow
  AcBranchOpf:
    title: AC Branch OPF
    type: object
    description: OPF results for generic AC branch
    properties:
      id:
        type: string
        description: AC branch ID
      acPowerFlow:
        description: 'AC Power Flow [p.u.]'
        type: array
        items:
          type: array
          items:
            type: array
            items:
              type: number
      lm:
        type: array
        description: 'Lagrange Mutiplier associated with the AC branch [€/p.u.]'
        items:
          type: array
          items:
            type: array
            items:
              type: number
    required:
      - id
      - acPowerFlow
      - lm
  GeneratorOpf:
    title: Generator OPF
    type: object
    description: OPF results for a generic Generator
    properties:
      id:
        type: string
        description: Generator ID
      activePower:
        type: array
        description: 'Active power [p.u.]'
        items:
          type: array
          items:
            type: array
            items:
```

```
                type: number
                minimum: 0
        activePowerCurtailment:
          type: array
          description: 'Active Power curtailment [p.u.]'
          items:
            type: array
            items:
              type: array
              items:
                type: number
        reactivePower:
          type: array
          description: 'Reactive power [p.u.] only for distribution generators'
          items:
            type: array
            items:
              type: array
              items:
                type: number
      required:
        - id
        - activePower
        - activePowerCurtailment
    FlexibleLoadOpf:
      title: Flexible Load OPF
      type: object
      description: OPF results for a generic Flexible load
      properties:
        id:
          type: string
          description: Flexible load ID
        demandCurtailment:
          type: array
          description: 'Demand curtailment [p.u.]'
          items:
            type: array
            items:
              type: array
              items:
                type: number
        demandReduction:
          type: array
          description: 'Demand reduction [p.u.] if flexbile load'
          items:
            type: array
            items:
              type: array
              items:
                type: number
                minimum: 0
        demandShiftedUp:
          type: array
          description: 'Demand shifted up [p.u.] if flexbile load'
          items:
            type: array
            items:
              type: array
              items:
                type: number
        demandShiftedDown:
          type: array
          description: 'Demand shifted down [p.u.] if flexbile load'
          items:
            type: array
            items:
              type: array
              items:
                type: number
        totalFlexibleDemand:
          type: array
          description: 'Total flexible demand [p.u.] if flexbile load. Computed as reference demand
- curtailment - reduction - shiftedUp - shiftedDown'
```

```
          items:
            type: array
            items:
              type: array
              items:
                type: number
      required:
        - id
        - demandCurtailment
    StorageOpf:
      title: Storage OPF
      type: object
      description: OPF result for a generic Storage unit
      properties:
        id:
          type: string
          description: Storage ID
        stateOfCharge:
          type: array
          description: 'State of charge [0, 1]'
          items:
            type: array
            items:
              type: array
              items:
                type: number
                maximum: 1
                minimum: 0
        activePowerAbsorbed:
          type: array
          description: 'Active power absorbed [p.u.]'
          items:
            type: array
            items:
              type: array
              items:
                type: number
                minimum: 0
        activePowerInjected:
          type: array
          description: 'Active power injected [p.u.]'
          items:
            type: array
            items:
              type: array
              items:
                type: number
                minimum: 0
        reactivePowerExchanged:
          type: array
          description: 'Reactive power exchange [p.u.] if distribution'
          items:
            type: array
            items:
              type: array
              items:
                type: number
      required:
        - id
        - stateOfCharge
        - activePowerAbsorbed
        - activePowerInjected
    AcBranchCandidate:
      title: AC Branch Candidate
      type: object
      description: Required inputs to model a candidate AC branch
      properties:
        acBranch:
          $ref: '#/components/schemas/AcBranch'
        invCost:
          description: 'Investment costs [€] for each year of the planning horizon at which the
candidate can be invested'
          type: array
```

```
          items:
            type: number
        lifetime:
          type: integer
          description: Lifetime of the candidate from its investment
          multipleOf: 10
        horizons:
          type: array
          description: All years of the planning horizon at which the candidate can be invested
          uniqueItems: true
          items:
            type: integer
        isUnique:
          type: boolean
          default: false
          description: Whether the candidate can be invested at several years of the planning
horizon (false) or not (true)
      required:
        - acBranch
        - invCost
        - lifetime
        - horizons
    DcBranchCandidate:
      title: DC Branch Candidate
      type: object
      description: Required inputs to model a candidate DC branch
      properties:
        dcBranch:
          $ref: '#/components/schemas/DcBranch'
        invCost:
          description: 'Investment costs [€] for each year of the planning horizon at which the
candidate can be invested'
          type: array
          items:
            type: number
        lifetime:
          type: integer
          description: Lifetime of the candidate from its investment
          multipleOf: 10
        horizons:
          type: array
          description: All years of the planning horizon at which the candidate can be invested
          uniqueItems: true
          items:
            type: integer
        isUnique:
          type: boolean
          default: false
          description: Whether the candidate can be invested at several years of the planning
horizon (false) or not (true)
      required:
        - dcBranch
        - invCost
        - lifetime
        - horizons
    ConverterCandidate:
      title: Converter Candidate
      type: object
      description: Required inputs to model a candidate AC/DC converter
      properties:
        converter:
          $ref: '#/components/schemas/Converter'
        invCost:
          description: 'Investment costs [€] for each year of the planning horizon at which the
candidate can be invested'
          type: array
          items:
            type: number
        lifetime:
          type: integer
          description: Lifetime of the candidate from its investment
          multipleOf: 10
        horizons:
```

```
          type: array
          description: All years of the planning horizon at which the candidate can be invested
          uniqueItems: true
          items:
            type: integer
        isUnique:
          type: boolean
          default: false
          description: Whether the candidate can be invested at several years of the planning
horizon (false) or not (true)
      required:
        - converter
        - invCost
        - lifetime
        - horizons
    PstCandidate:
      title: PST Candidate
      type: object
      description: Required inputs to model a candidate PST
      properties:
        pst:
          $ref: '#/components/schemas/Pst'
        invCost:
          description: 'Investment costs [€] for each year of the planning horizon at which the
candidate can be invested'
          type: array
          items:
            type: number
        lifetime:
          type: integer
          description: Lifetime of the candidate from its investment
          multipleOf: 10
        horizons:
          type: array
          description: All years of the planning horizon at which the candidate can be invested
          uniqueItems: true
          items:
            type: integer
        isUnique:
          type: boolean
          default: false
          description: Whether the candidate can be invested at several years of the planning
horizon (false) or not (true)
      required:
        - pst
        - invCost
        - lifetime
        - horizons
    StorageCandidate:
      title: Storage Candidate
      type: object
      description: 'Required inputs to model a storage candidate '
      properties:
        storage:
          $ref: '#/components/schemas/Storage'
        invCost:
          description: 'Investment costs [€] for each year of the planning horizon at which the
candidate can be invested'
          type: array
          items:
            type: number
        lifetime:
          type: integer
          description: Lifetime of the candidate from its investment
          multipleOf: 10
        horizons:
          type: array
          description: All years of the planning horizon at which the candidate can be invested
          uniqueItems: true
          items:
            type: integer
        isUnique:
          type: boolean
```

```
          default: false
          description: Whether the candidate can be invested at several years of the planning
horizon (false) or not (true)
        storageData:
          $ref: '#/components/schemas/StorageScenario'
      required:
        - storage
        - invCost
        - storageData
        - lifetime
        - horizons
    AcBranchDecision:
      title: AC Branch Decision
      type: object
      description: Investment decision for AC branch candidate
      properties:
        id:
          type: string
          description: AC branch ID
        invDecision:
          type: array
          description: Investment decision for each year of the planning horizon at which the
candidate can be invested
          items:
            type: boolean
      required:
        - id
        - invDecision
    DcBranchDecision:
      title: DC Branch Decision
      type: object
      description: Investment decision for DC branch candidate
      properties:
        id:
          type: string
          description: DC branch ID
        invDecision:
          type: array
          description: Investment decision for each year of the planning horizon at which the
candidate can be invested
          items:
            type: boolean
      required:
        - id
        - invDecision
    ConverterDecision:
      title: Converter Decision
      type: object
      description: Investment decision for AC/DC candidate
      properties:
        id:
          type: string
          description: AC/DC converter ID
        invDecision:
          type: array
          description: Investment decision for each year of the planning horizon at which the
candidate can be invested
          items:
            type: boolean
      required:
        - id
        - invDecision
    PstDecision:
      title: PST Decision
      type: object
      properties:
        id:
          type: string
          description: PST ID
        invDecision:
          type: array
          description: Investment decision for each year of the planning horizon at which the
candidate can be invested
```

```
          items:
            type: boolean
      required:
        - id
        - invDecision
      description: Investment decision for PST candidate
    StorageDecision:
      title: Storage Decision
      type: object
      description: Investment decision for storage candidate
      properties:
        id:
          type: string
          description: Storage ID
        invDecision:
          type: array
          description: Investment decision for each year of the planning horizon at which the
candidate can be invested
          items:
            type: boolean
      required:
        - id
        - invDecision
    FlexibleLoadDecision:
      title: Flexible Load Decision
      type: object
      description: Investment decision for Flexible Load candidate
      properties:
        id:
          type: string
          description: Storage ID
        invDecision:
          type: array
          description: Investment decision for each year of the planning horizon at which the
candidate can be invested
          items:
            type: boolean
      required:
        - id
        - invDecision
    StorageScenario:
      title: Storage Scenario
      type: object
      description: Required scenario data to model a generic storage system
      properties:
        id:
          type: string
          description: Storage ID
        powerExternalProcess:
          description: 'Power provided or demand by external process [p.u.]'
          type: array
          items:
            type: array
            items:
              type: array
              items:
                type: number
        maxInjActivePower:
          type: array
          description: 'Fraction of the maximum injected active power available in [0, 1]'
          items:
            type: array
            items:
              type: array
              items:
                type: number
                minimum: 0
                maximum: 1
        maxAbsActivePower:
          type: array
          description: 'Fraction of the maximum absorbed active power [p.u.] available in [0, 1]'
          items:
            type: array
```

```
            items:
              type: array
              items:
                type: number
                minimum: 0
                maximum: 1
        initEnergy:
          type: array
          description: 'Initial energy content [p.u.] for each scenario and each year of the
planning horizon'
          items:
            type: array
            items:
              type: number
        finalEnergy:
          type: array
          description: 'Final energy content [p.u.] for each scenario and each year of the planning
horizon'
          items:
            type: array
            items:
              type: number
      required:
        - id
        - initEnergy
        - finalEnergy
    GeneratorScenario:
      title: Generator Scenario
      type: object
      description: 'Required scenario data to model a generic generator, both thermal and RES.
generationCosts are associated to thermal generators, whereas curtailmentCosts are related to RES.'
      properties:
        id:
          type: string
          description: Generator ID
        capacityFactor:
          type: array
          description: 'Ratio between 0 and 1 indicating the share of the maxActivePower that is
available for a RES unit at each hour []'
          items:
            type: array
            items:
              type: array
              items:
                type: number
      required:
        - id
    FlexplanFullProcessInputFile:
      title: FlexPlan Full Process Input File
      type: object
      description: 'The FlexplanFullProcessInputFile composed of genericParameters,
gridModelInputFile and scenarioDataInputFile.'
      properties:
        gridModelInputFile:
          $ref: '#/components/schemas/GridModelInputFile'
        scenarioDataInputFile:
          $ref: '#/components/schemas/ScenarioDataInputFile'
        genericParameters:
          $ref: '#/components/schemas/GenericParameters'
        contingencyStates:
          type: array
          description: Contingency states to take into account to compute the costs related to
reliability of supply of the OPF.
          items:
            $ref: '#/components/schemas/ContingencyState'
      required:
        - gridModelInputFile
        - scenarioDataInputFile
        - genericParameters
    OptimalPowerFlowInputFile:
      title: Optimal Power Flow Input File
      type: object
```

```
      description: 'The OptimalPowerFlowInputFile composed of genericParameters, gridModelInputFile
and scenarioDataInputFile.'
      properties:
        gridModelInputFile:
          $ref: '#/components/schemas/GridModelInputFile'
        scenarioDataInputFile:
          $ref: '#/components/schemas/ScenarioDataInputFile'
        genericParameters:
          $ref: '#/components/schemas/GenericParameters'
        contingencyStates:
          type: array
          description: Contingency states to take into account to compute the costs related to
reliability of supply of the OPF.
          items:
            $ref: '#/components/schemas/ContingencyState'
      required:
        - gridModelInputFile
        - scenarioDataInputFile
        - genericParameters
    GridExpansionPlanningInputFile:
      title: Grid Expansion Planning Input File
      type: object
      description: 'The GridExpansionPlanningInputFile composed of genericParameters,
candidatesInputFile, gridModelInputFile and scenarioDataInputFile.'
      properties:
        candidatesInputFile:
          $ref: '#/components/schemas/CandidatesInputFile'
        gridModelInputFile:
          $ref: '#/components/schemas/GridModelInputFile'
        scenarioDataInputFile:
          $ref: '#/components/schemas/ScenarioDataInputFile'
        genericParameters:
          $ref: '#/components/schemas/GenericParameters'
      required:
        - candidatesInputFile
        - gridModelInputFile
        - scenarioDataInputFile
        - genericParameters
    PreProcessorInputFile:
      title: Pre-Processor Input File
      type: object
      description: The PreProcessorInputFile composed of OptimalPowerFlowInputFile and
OptimalPowerFlowOutputFile.
      properties:
        optimalPowerFlowInputFile:
          $ref: '#/components/schemas/OptimalPowerFlowInputFile'
        optimalPowerFlowOutputFile:
          $ref: '#/components/schemas/OptimalPowerFlowOutputFile'
      required:
        - optimalPowerFlowInputFile
        - optimalPowerFlowOutputFile
    BusCharacteristics:
      title: Bus Characteristics
      type: object
      description: 'Information related to the buses, used by the pre-processor in order to assess
the constraint checking process. More information available at Methodology T2.3'
      properties:
        classBus:
          type: string
          description: |-
            Type of bus. Only one choice accepted, eg.  Substation (SBSTAIRR: air, SBSTCPCT: air-
compact, SBSTUNDG: underground, SWITSTAT: Switching Station); Industrial load: INDLLOAD -if generic-
(
            (INDLSWMW: Sawmills and Wood Preservation, INDLNMMM: Non-metallic Mineral Mining and
Quarrying, INDLPPMN: Converted Paper Product Manufacturing, INDLCMNT: Cement, INDLFRVG: Fruit and
Vegetable Preserving and Specialty Food Manufacturing, INDLACMN: Agriculture, Construction and
Mining Machinery Manufacturing, INDLCHMN Basic Chemical Manufacturing, INDLEEMN: Other Electrical
Equipment and Component Manufacturing, INDLFFMN: Resin, Synthetic Rubber and Artificial Synthetic
Fibres and Filaments Manufacturing, INDLDPMN: Dairy Product Manufacturing, INDLCRPR: Support
Activities for Crop Production, INDLAPMN: Aerospace Product and Parts Manufacturing, INDLMPMN: Other
Fabricated Metal Product Manufacturing, INDLASPR: Animal Slaughtering and Processing, INDLTXTM:
Other Textile Product Mills, INDLSTMN: Steel Product Manufacturing from Purchased Steel, INDLWTSW:
Water, Sewage and other systems, INDLBKMN: Bakeries and Tortilla Manufacturing, INDLCTFR: Cattle
```

```
Ranching and Farming, INDLBVMN: Beverage Manufacturing, INDLPPPM: Pulp, Paper and Paperboard Mills,
INDLPRMN: Clay Product and Refractory Manufacturing, INDLMCMN: Other General Purpose Machinery
Manufacturing, INDLWHST: Warehousing and Storage, INDLPLMN: Plastic manufacture industry, INDLPRGR:
Printing/Graphic industry); power plant (PWPLWIND: wind, PWPLPVPV:  solar,  PWPLTHRM: thermal coal,
PWPLCCYC:  Combined Cycle, PWPLBMSS: biomass, PWPLHYDR: hydro, PWPLNCLR: nuclear); commercial load:
CMCLLOAD (COMMHOTL: hotel, COMMHPTL: hospital, COMMSPMT: supermarket, COMMMRST: other general
merchandise stores)
          default: SBSTAIRR
          enum:
            - SBSTCPCT
            - SBSTUNDG
            - INDLSWMW
            - INDLNMMM
            - INDLPPMN
            - INDLCMNT
            - INDLFRVG
            - INDLACMN
            - INDLCHMN
            - INDLEEMN
            - INDLFFMN
            - INDLDPMN
            - INDLCRPR
            - INDLAPMN
            - INDLMPMN
            - INDLASPR
            - INDLTXTM
            - INDLSTMN
            - INDLWTSW
            - INDLBKMN
            - INDLCTFR
            - INDLBVMN
            - INDLPPPM
            - INDLPRMN
            - INDLMCMN
            - INDLWHST
            - INDLPLMN
            - INDLPRGR
            - COMMMRST
            - COMMHOTL
            - COMMHPTL
            - COMMSPMT
            - SBSTAIRR
            - SWITSTAT
            - PWPLWIND
            - PWPLPVPV
            - PWPLBMSS
            - PWPLHYDR
            - PWPLTHRM
            - PWPLCCYC
            - PWPLNCLR
            - INDLLOAD
            - CMCLLOAD
        naturalResourcesAvailability:
          type: array
          description: 'Availability of natural resources (for substation type buses). Multiple
choice accpeted, eg.: RSRCWATR: water (river, reservoir); RSRCWIND: wind (area with wind parks
near); RSRCSUNN: sun (solar power plants near); RSRCCVRN: cavern; RSRCBMSS: biomass'
          uniqueItems: true
          items:
            type: string
            enum:
              - RSRCWATR
              - RSRCWIND
              - RSRCSUNN
              - RSRCBMSS
              - RSRCCVRN
        loadSupplied:
          type: array
          description: 'Loads supplied (for substation type buses). Multiple choice accepted, eg.:
RSDTLOAD: residential (mainly); CMCLLOAD: commercial (mainly); INDLLOAD: industrial (mainly);
MIXDLOAD mixed (lower voltage level networks, sub-transmission/distribution, big industrial); '
          uniqueItems: true
          items:
```

```
            type: string
            enum:
              - RSDTLOAD
              - CMCLLOAD
              - INDLLOAD
              - MIXDLOAD
            default: MIXDLOAD
        busLocation:
          type: string
          description: 'Location of bus. Only one choice accepted, eg. LCTNURBN: urban (populated
city); LCTNINAR: industrial area; LCTNSMRR: semi-rural (outskirts of populated city, small city);
LCTNRURL: rural. Default value: no restriction (if no value is provided, no restriction is
considered)'
          enum:
            - LCTNURBN
            - LCTNSMRR
            - LCTNRURL
            - LCTNINAR
        geographicalCharacteristics:
          type: string
          description: 'Geographic characteristics (for rural buses). Only one choice accepted, eg.
LCTNMNTN: mountainous; LCTNHILL: hilly; LCTNPLAI: plain. Default value: no restriction (if no value
is provided, no restriction is considered)'
          enum:
            - LCTNMNTN
            - LCTNHILL
            - LCTNPLAI
        restrictedArea:
          type: array
          description: 'Restricted area (not allowed to build new installations). Multiple choice
accepted, eg.: RSTRLINE: for lines; RSTRPPHY: for hydro plants; RSTRHDRG: for hydrogen; RSTRBTTR:
for batteries; RSTRCAES: for CAES/LAES; RSTRTOTL: total restriction. If no value is provided, no
restriction considered'
          uniqueItems: true
          items:
            type: string
            enum:
              - RSTRLINE
              - RSTRPPHY
              - RSTRHDRG
              - RSTRBTTR
              - RSTRCAES
              - RSTRTOTL
    FlexibleLoadScenario:
      title: Flexible Load Scenario
      type: object
      description: Required scenario data to model a non-flexible load and flexible load.
      properties:
        id:
          type: string
          description: Load ID
        demandReference:
          type: array
          description: 'Demand reference [p.u.]'
          items:
            type: array
            items:
              type: array
              items:
                type: number
        superiorBoundNCP:
          type: array
          description: 'Superior bound on not consumed power as a fraction of the demandReference in
[0, 1]. Only needed if modeling a flexible load.'
          items:
            type: array
            items:
              type: array
              items:
                type: number
                minimum: 0
                maximum: 1
        superiorBoundUDS:
```

```
          type: array
          description: 'Superior bound on upward demand shifted as a fraction of the demandReference
in [0, 1]. Only needed if modeling a flexible load.'
          items:
            type: array
            items:
              type: array
              items:
                type: number
                minimum: 0
                maximum: 1
        superiorBoundDDS:
          type: array
          description: 'Superior bound on downward demand shifted as a fraction of the
demandReference in [0, 1]. Only needed if modeling a flexible load.'
          items:
            type: array
            items:
              type: array
              items:
                type: number
                minimum: 0
                maximum: 1
      required:
        - id
        - demandReference
    FlexibleLoad:
      title: Flexible Load
      type: object
      description: Required inputs to model both flexible and non-flexible loads
      properties:
        id:
          type: string
          description: Load ID
        acBusConnected:
          type: string
          description: AC Bus connected
        powerFactor:
          type: number
          description: 'Ratio (between 0 and 1) between the active power and apparent power consumed
by the load [-]. Only needed if acBusConnected has isTransmission==False, i.e. only for loads
connected to distribution systems'
          minimum: 0
          maximum: 1
        gracePeriodUDS:
          type: array
          description: 'Grace period for upward demand shifted [h] for each year of the planning
horizon. Only needed if modeling a flexible load'
          items:
            type: integer
        gracePeriodDDS:
          type: array
          description: 'Grace period for downward demand shifting [h] for each year of the planning
horizon. Only needed if modeling a flexible load'
          items:
            type: integer
        maxEnergyNotConsumed:
          type: array
          description: 'Maximum energy not consumed as a fraction of the accumulated reference
demand in [0, 1] for each year of the planning horizon. Only needed if modeling a flexible load'
          items:
            type: number
            minimum: 0
            maximum: 1
        compensationDemandShift:
          type: array
          description: 'Compensation for demand shifting [€/p.u.]. Only needed if modeling a
flexible load'
          items:
            type: number
        compensationConsumeLess:
          type: array
```

```
          description: 'Compensation for consuming less [€/p.u.]. Only needed if modeling a flexible
load'
          items:
            type: number
        superiorBoundNCP:
          type: array
          description: 'Superior bound on not consumed power as a fraction of the demandReference in
[0, 1]. Only needed if modeling a flexible load'
          items:
            type: number
            minimum: 0
            maximum: 1
        superiorBoundUDS:
          type: array
          description: 'Superior bound on upward demand shifted as a fraction of the demandReference
in [0, 1]. Only needed if modeling a flexible load'
          items:
            type: number
            minimum: 0
            maximum: 1
        superiorBoundDDS:
          type: array
          description: 'Superior bound on downward demand shifted as a fraction of the
demandReference in [0, 1]. Only needed if modeling a flexible load'
          items:
            type: number
            minimum: 0
            maximum: 1
        valueOfLossLoad:
          type: array
          description: 'Cost of involuntary demand reduction (value of lost load) [€/p.u.]'
          items:
            type: number
        maxEnergyShifted:
          type: array
          description: 'Maximum energy shifted as a fraction of the accumulated reference demand in
[0, 1] for each year of the planning horizon. Only needed if modeling a flexible load'
          items:
            type: number
            minimum: 0
            maximum: 1
        isFlexible:
          type: boolean
          description: 'True=flexible load, False=unflexible load. Default = False'
          default: false
        isSurrogate:
          type: boolean
          default: false
          description: Used only internally for T&D decomposition.
      required:
        - id
        - acBusConnected
        - valueOfLossLoad
        - isFlexible
    GenericParameters:
      title: Generic Parameters
      type: object
      description: Object describing generic information of the process
      x-examples: {}
      properties:
        years:
          type: array
          description: 'Array containing the years covered in the process, eg. [2030, 2040, 2050]'
          items:
            type: integer
        nbHours:
          type: array
          description: |-
            List of number of hours covered in each scenario and each year of the process.

            Example: for a two year planning horizon with three different scenarios, the input
should look like: [[744, 744], [672, 696], [720, 720]], i.e. it should be composed of as many sub-
arrays as 'nbScenarios', each one with the same length as the array 'years'.
```

```
          items:
            type: array
            items:
              type: integer
        nbScenarios:
          type: integer
          description: 'Number of scenarios covered in the process, covering the number of variants
times the number of periods. Eg. for two variants of one year divided in 12 months, this values
should be equal to 2 x 12 = 24.'
        scenarioProbabilities:
          type: array
          description: |-
            Probabilities associated to the years covered in the process per scenario.

            Example: for a two year planning horizon with three different scenarios, the input
should look like: [[0.3, 0.2], [0.4, 0.5], [0.3, 0.3]], i.e. it should be composed of as many sub-
arrays as 'nbScenarios', each one with the same length as the array 'years'.

            If not specified, uniform distribution is assumed.
          items:
            type: array
            items:
              type: number
        maxNumberOfCandidates:
          description: Maximum number of candidates that the pre-processor can propose for each year
of the planning horizon
          type: array
          items:
            type: integer
            minimum: 0
        basePower:
          type: number
          description: 'Base power used for the per-unit notation [MW]'
        discountRate:
          type: number
          default: 0
          description: 'Discount rate [0, 1]. Default value: 0'
        cplexParameters:
          $ref: '#/components/schemas/CplexParameters'
        referenceYear:
          type: integer
          default: 2030
          description: 'Year taken as reference for the discounting of investment and operational
costs. Default value: 2030'
        nbRepresentedYears:
          type: integer
          default: 10
          description: |-
            Number of years between each considered year in the process. Default value: 10

            Example: if years = [2030,2040,2050], then nbRepresentedYears = 10.
        applyDiscount:
          type: boolean
          default: true
          description: 'Boolean variable representing the use of discounting for investment and
operational costs. Default value: True.'
        scalingFactor:
          type: number
          default: 1
          description: 'Scaling factor to be used in the objective function, i.e. the factor by
which all costs will be divided in the objective function. Default value: 1'
        useTdDecomposition:
          type: boolean
          default: false
          description: Whether to use the T&D decomposition (true) or not (false).
        estimateCostTdExchange:
          type: number
          description: 'Estimate of the cost of energy exchanged between transmission and
distribution. Required to use the T&D decomposition. Default value: 1'
          default: 1
        allDistributionNetworks:
          type: object
```

```
          description: Dictionary of distribution network IDs with the ID of the transmission AC bus
to which the distribution network is connected. Required to use the T&D decomposition.
          additionalProperties:
            type: string
        maxNumberOfCandidatesPerNode:
          type: array
          description: Maximum number of investments that can be connected to any node in the system
for each year of the planning horizon
          items:
            type: integer
            minimum: 0
      required:
        - years
        - nbHours
        - nbScenarios
    PtdfMatrix:
      title: PTDF matrix
      type: object
      description: 'Power Transfer Distribution Factors of a given network. The matrix has size
{n_branches x n_buses}, expressed as an array of arrays  '
      properties:
        buses:
          type: array
          description: List of buses IDs
          items:
            type: string
        branches:
          type: array
          description: List of branches IDs
          items:
            type: string
        values:
          type: array
          description: 'List of lists, being each nested list a row of the PTDF matrix. There are as
many nested lists as branches and transformers in the network'
          items:
            type: array
            description: 'Nested list, representing a row of the PTDF matrix, composed its values
(numbers). All nested lists have the same length, being equal to the number of buses'
            items:
              type: number
      required:
        - buses
        - branches
        - values
    AcBranchPreCandidate:
      title: AC Branch Pre-Candidate
      type: object
      description: A potential AC branch candidate to be considered by the pre-processor.
      properties:
        id:
          type: string
          description: AC Branch ID
        acBusOrigin:
          type: string
          description: AC bus connected origin
        acBusExtremity:
          type: string
          description: AC bus connected extremity
        ratedApparentPower:
          type: number
          description: 'Rated apparent power of the line [p.u.]'
          minimum: 0
        length:
          type: number
          description: 'Length of the line [km]'
          minimum: 0
    DcBranchPreCandidate:
      title: DcBranchPreCandidate
      type: object
      description: A potential DC branch candidate to be considered by the pre-processor.
      properties:
        id:
```

```
          type: string
          description: DC Branch ID
        dcBusOrigin:
          type: string
          description: DC bus connected origin
        dcBusExtremity:
          type: string
          description: DC bus connected extremity
        ratedActivePower:
          type: number
          description: 'Rated active power of the line [p.u.]'
          minimum: 0
        length:
          type: number
          description: 'Length of the line [km]'
          minimum: 0
    ScenarioReductionInputFile:
      title: Scenario Reduction Input File
      type: object
      properties:
        generators:
          type: array
          items:
            $ref: '#/components/schemas/ScenarioReductionGeneratorScenario'
        loads:
          type: array
          items:
            $ref: '#/components/schemas/ScenarioReductionLoadScenario'
        granularity:
          type: string
          enum:
            - yearly
            - monthly
            - weekly
            - daily
          default: yearly
        normalization:
          type: string
          enum:
            - minmax
            - nominal
          default: minmax
        numberOfClusters:
          type: integer
      required:
        - generators
        - loads
        - numberOfClusters
    ScenarioReductionOutputFile:
      title: Scenario Reduction Output File
      type: object
      properties:
        message:
          type: string
        probabilities:
          type: array
          items:
            $ref: '#/components/schemas/ScenarioReductionProbabilities'
      required:
        - message
        - probabilities
    ScenarioReductionGeneratorScenario:
      title: ScenarioReductionGeneratorScenario
      type: object
      description: 'This contains the scenario data for each generator: As an array of yearly time-
series!'
      properties:
        acBusConnected:
          type: string
        type:
          type: string
          enum:
            - pv
```

```
            - wind
        scenario:
          type: array
          items:
            type: array
            items:
              type: number
      required:
        - acBusConnected
        - type
        - scenario
    ScenarioReductionLoadScenario:
      title: ScenarioReductionLoadScenario
      type: object
      description: 'This contains the scenario data for each load: As an array of yearly time-
series!'
      properties:
        acBusConnected:
          type: string
        scenario:
          type: array
          items:
            type: array
            items:
              type: number
      required:
        - acBusConnected
        - scenario
    ScenarioReductionProbabilities:
      title: ScenarioReductionProbabilities
      type: object
      description: This contains the cluster or class or year against its probability of
importance/occurance value!
      properties:
        selectedVariant:
          type: integer
        probability:
          type: number
        nbVariantsInCluster:
          type: integer
        allVariantsInCuster:
          type: array
          items:
            type: integer
      required:
        - selectedVariant
        - probability
        - nbVariantsInCluster
        - allVariantsInCuster
    StartResponse:
      title: StartResponse
      type: object
      description: 'This contains the details provided to the user once the user launches a request!
'
      properties:
        message:
          type: string
        taskId:
          type: string
        outputFileFullPath:
          type: string
      required:
        - message
        - taskId
        - outputFileFullPath
    ListResults:
      title: ListResults
      type: object
      description: This contains the taskid and the corresponding file name of the results
accumulated from the S3 storage.
      properties:
        message:
          type: string
```

```
        results:
          type: array
          items:
            $ref: '#/components/schemas/TaskIdResultFileName'
    TaskIdResultFileName:
      title: TaskIdResultFileName
      type: object
      description: This contains the taskid and the corresponding file name for a single result
present in the S3 storage.
      properties:
        taskId:
          type: string
        outputFileFullPath:
          type: string
    DownloadResults:
      title: downloadResults
      type: object
      properties:
        message:
          type: string
        results:
          $ref: '#/components/schemas/AllResults'
      required:
        - message
        - results
      description: 'This model contains the results provided by the download API request. '
    AllResults:
      title: AllResults
      type: object
      description: This model provides the opportunity to group all the results. However it is
essentially used to provide one of the "OPF/SR/GEP/FFP/PP' results at a given time.
      properties:
        opf:
          $ref: '#/components/schemas/OptimalPowerFlowOutputFile'
        sr:
          $ref: '#/components/schemas/ScenarioReductionOutputFile'
        gep:
          $ref: '#/components/schemas/GridExpansionPlanningOutputFile'
        ffp:
          $ref: '#/components/schemas/GridExpansionPlanningOutputFile'
        pp:
          $ref: '#/components/schemas/GridExpansionPlanningInputFile'
      required:
        - opf
        - sr
        - gep
        - ffp
        - pp
    ContingencyState:
      title: ContingencyState
      type: object
      properties:
        acBranches:
          type: array
          description: ids of AC branches in outage for this contingency state
          items:
            type: string
        dcBranches:
          type: array
          description: ids of DC branches in outage for this contingency state
          items:
            type: string
        converters:
          type: array
          description: ids of converters in outage for this contingency state
          items:
            type: string
        transformers:
          type: array
          description: ids of transformers in outage for this contingency state
          items:
            type: string
        psts:
```

```
          type: array
          description: ids of PSTs in outage for this contingency state
          items:
            type: string
        storage:
          type: array
          description: ids of storage assets in outage for this contingency state
          items:
            type: string
        generators:
          type: array
          description: ids of generators in outage for this contingency state
          items:
            type: string
      description: Structure representing the set of grid assets in outage for a contingency state.
    CostsOPF:
      title: CostsOPF
      type: object
      description: 'Costs of an optimal power flow solution, breaked down by type'
      properties:
        totalCosts:
          type: number
          description: 'Value of the objective function = sum of all costs for the requested
planning horizons, weighted by scenario probabilites.'
        generationCosts:
          type: array
          description: Generation costs per scenario and per planning horizon.
          items:
            type: array
            items:
              type: array
              items:
                type: number
        genCurtailmentCosts:
          type: array
          description: Generation curtailment costs per scenario and per planning horizon.
          items:
            type: array
            items:
              type: array
              items:
                type: number
        loadCurtailmentCosts:
          type: array
          description: Load curtailment costs per scenario and per planning horizon.
          items:
            type: array
            items:
              type: array
              items:
                type: number
        loadReductionCosts:
          type: array
          description: Load reduction costs per scenario and per planning horizon.
          items:
            type: array
            items:
              type: array
              items:
                type: number
        loadShiftingCosts:
          type: array
          description: Load shifting costs per scenario and per planning horizon.
          items:
            type: array
            items:
              type: array
              items:
                type: number
        slackCosts:
          type: array
          description: Nodal slack costs per scenario and per planning horizon (only if slackStatus
is set to True for at least one bus).
```

```
          items:
            type: array
            items:
              type: array
              items:
                type: number
        reliabilityCosts:
          type: array
          description: 'Costs related to reliability of supply per contingency set, scenario and
planning horizon (only if at least one reliability set is specified in the input).'
          items:
            type: array
            items:
              type: array
              items:
                type: array
                items:
                  type: number
    CplexParameters:
      title: CPLEX Parameters
      type: object
      description: Object describing CPLEX parameters
      properties:
        CPXPARAM_Read_Scale:
          type: integer
          minimum: -1
          maximum: 1
          description: |-
            Decides how to scale the problem matrix:
            -1 = no scaling; FlexPlan default,
            0 = equilibration scaling,
            1 = more aggressive scaling.
        CPXPARAM_Barrier_StartAlg:
          type: integer
          description: |-
            Sets the algorithm to be used to compute the initial starting point for the barrier
optimizer:
            1 = dual is 0 (zero); FlexPlan default,
            2 = estimate dual,
            3 = average of primal estimate, dual 0 (zero),
            4 = average of primal estimate, estimate dual.
          minimum: 1
          maximum: 4
        CPXPARAM_Barrier_Crossover:
          type: integer
          description: |-
            Decides which crossover is performed at the end of a barrier optimization:
            0 = automatic: let CPLEX choose,
            1 = primal crossover; FlexPlan default,
            2 = dual crossover.
          minimum: 0
          maximum: 2
        CPXPARAM_Benders_Strategy:
          type: integer
          minimum: -1
          maximum: 3
          description: |-
            Specifies whether CPLEX should apply Benders algorithm as a strategy to solve a model:
            -1 = OFF (not used); OPF default,
            0 = AUTO (let CPLEX decide),
            1 = USER (custom decomposition); GEP default,
            2 = WORKERS (custom master, CPLEX workers),
            3 = FULL (CPLEX decomposition).
        CPXPARAM_LPMethod:
          type: integer
          minimum: 0
          maximum: 6
          description: |-
            Controls which algorithm is used to solve continuous linear models or to solve the root
relaxation of a MIP:
            0 = automatic: let CPLEX choose,
            1 = primal simplex,
            2 = dual simplex,
```

```
            3 = network simplex,
            4 = barrier; FlexPlan default,
            5 = sifting,
            6 = concurrent.
        CPXPARAM_MIP_Tolerances_MIPGap:
          type: number
          description: Sets a RELATIVE tolerance on the gap between the best integer objective and
the objective of the best node remaining.
          minimum: 0
          maximum: 1
        CPXPARAM_TimeLimit:
          type: number
          description: 'Sets the maximum time, in seconds, for a call to an optimizer (for an OPF,
this the time per variant and per year).'
          minimum: 0
        CPXPARAM_MIP_Tolerances_AbsMIPGap:
          type: number
          minimum: 0
          description: Sets an ABSOLUTE tolerance on the gap between the best integer objective and
the objective of the best node remaining.
    CostsGEP:
      title: CostsGEP
      type: object
      description: 'Costs of a grid expansion planning optimization program, breaked down by type'
      properties:
        totalCosts:
          type: number
          description: 'Value of the objective function = sum of all costs for the requested
planning horizons, weighted by scenario probabilites.'
        investmentCosts:
          type: object
          description: Total incurred investment costs over the entire planning horizon per type of
candidates (for which the residual value have been discounted).
          properties:
            acBranches:
              type: number
            dcBranches:
              type: number
            converters:
              type: number
            psts:
              type: number
            transformers:
              type: number
            storage:
              type: number
            flexibleLoads:
              type: number
            total:
              type: number
        generationCosts:
          type: array
          description: Generation costs per scenario and per planning horizon.
          items:
            type: array
            items:
              type: array
              items:
                type: number
        genCurtailmentCosts:
          type: array
          description: Generation curtailment costs per scenario and per planning horizon.
          items:
            type: array
            items:
              type: array
              items:
                type: number
        loadCurtailmentCosts:
          type: array
          description: Load curtailment costs per scenario and per planning horizon.
          items:
            type: array
```

```
            items:
              type: array
              items:
                type: number
        loadReductionCosts:
          type: array
          description: Load reduction costs per scenario and per planning horizon.
          items:
            type: array
            items:
              type: array
              items:
                type: number
        loadShiftingCosts:
          type: array
          description: Load shifting costs per scenario and per planning horizon.
          items:
            type: array
            items:
              type: array
              items:
                type: number
        slackCosts:
          type: array
          description: Nodal slack costs per scenario and per planning horizon (only if slackStatus
is set to True for at least one bus).
          items:
            type: array
            items:
              type: array
              items:
                type: number
```