



EASY-RES

# SUMMER SCHOOL “ENABLING DRES TO OFFER ANCILLARY SERVICES” 20TH – 24TH SEPTEMBER 2021

## Software Tool for Evaluation of AS

Aleksandra Lekić and Miloš Cvetković



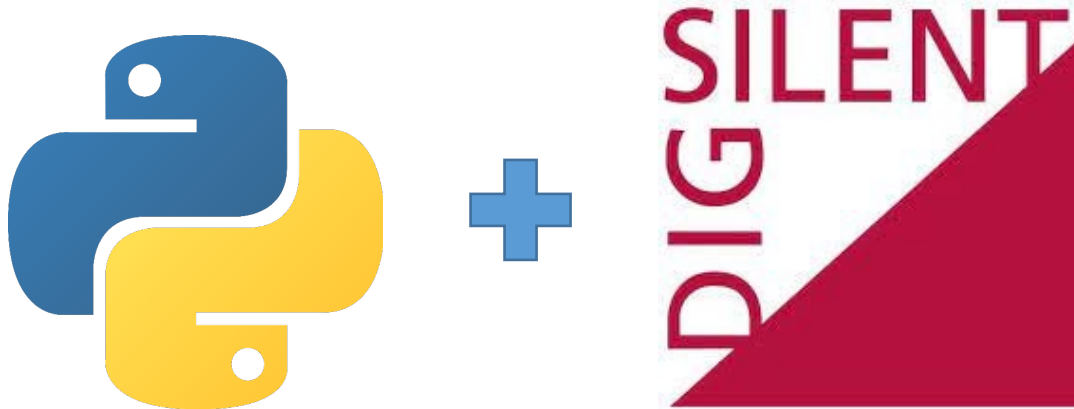
This project has received funding from the European Union’s Horizon 2020 Programme for research and innovation under Grant Agreement no 764090.

# Objectives and KPIs

	Objective	KPI
1	Increase the robustness of the power system to abrupt frequency changes using virtual inertia and damping in DRES	Increase the average overall system inertia by at least 1.5% for every 10% increase in DRES penetration
2	Contribute to the stability of the grid by providing frequency-dependent active power	For every 3 MW of DRES entering the system more than 2.5 MW of conventional reserves will be decommissioned. The rest of 0.5 MW or less will appear as reduction of the power of conventional base load units.
3	Increase the renewable energy penetration levels at both LV and MV level, while avoiding investments for grid reinforcement	The relative increase in DRES with EASY-RES is higher than with conventional approach
4	Make the RES more grid-friendly by i) reducing the short-term electric power fluctuations at both DRES and HV/MV substation level and ii) introducing active harmonics filtering to each DRES converter	i) Power smoothing functionality reduces power fluctuations $\max\Delta P/\text{minute} < 30\%$ of DRES rated power and $\max\Delta P/\text{minute} < 10\%$ of HV/MV (MV/LV) transformer rating ii) THD is kept within the limit of EN50160 standard ( $< 8\%$ ) even with 100% penetration of DRES
5	Preserve the long-term grid security	The additional DRES penetration due to developed functionality that does not violate existing fault-protection means in MV and LV (symmetrical and non-symmetrical)

## Tool idea

- To speed up simulation.
- Enable fast simulation of thousands of scenarios.
- Give an access for easy parameter setting.
- Solution: setting Python with DigSILENT PowerFactory (PF)





# Setting up the stage – Python installation

- Anaconda chosen – good GUI named Spyder
- Package installation?
  - Inside Anaconda: open Anaconda prompt  
`conda install package-name`
  - Python – command line  
`python -m pip install package_name`
  - Python – command line + git  
`python -m pip install git+git_location`



# Setting up the stage – Spyder interface

The screenshot displays the Spyder Python IDE interface. The left pane shows a script named `rms_sc.py` with the following content:

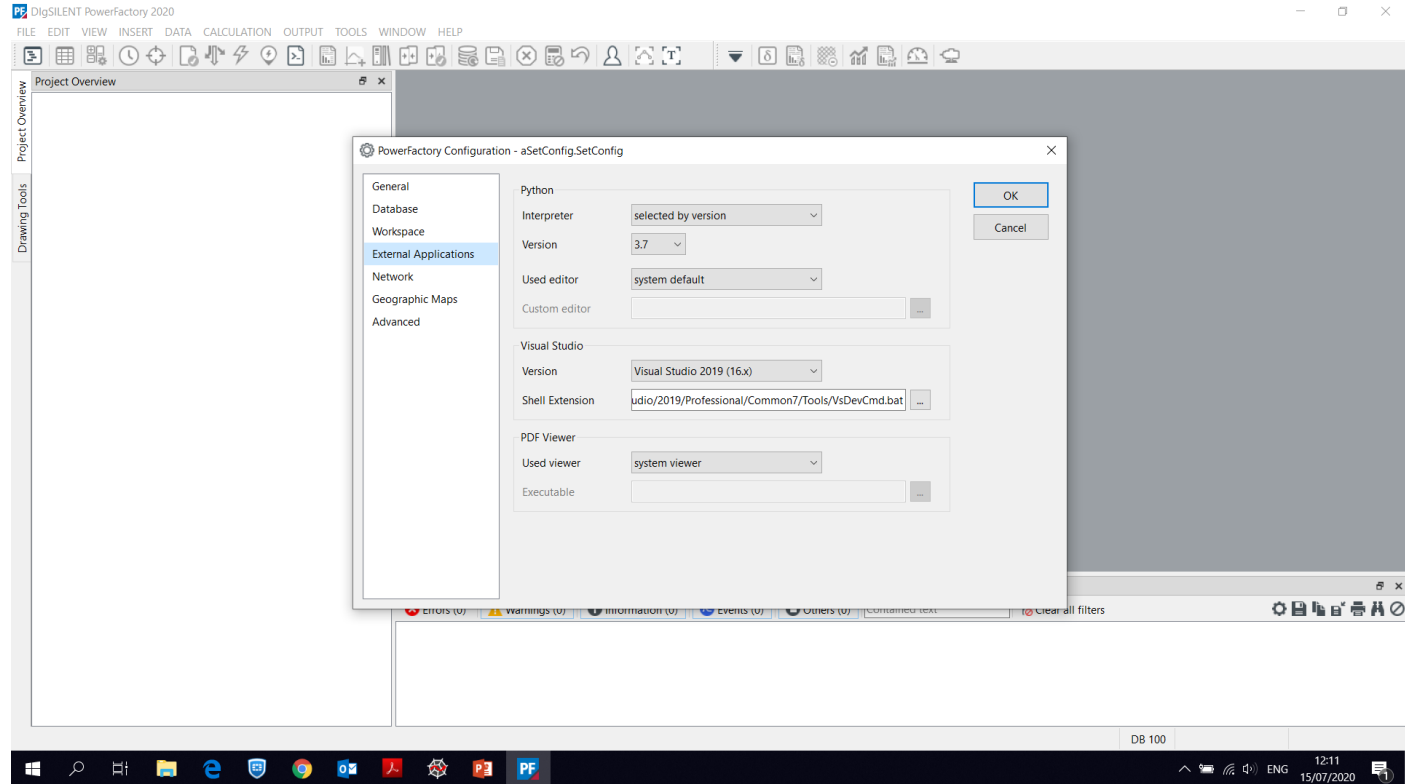
```
1  # RMS
2  Shows how to create and delete short circuits, and retrieve results for an
3  RMS simulation.
4
5  For an EMT, only change sim_type to 'emt' and the adjust the monitored variables
6  accordingly (the names differ between RMS and EMT).
7  ***
8
9  import matplotlib.pyplot as plt
10 from pfsim import PowerFactorySim
11
12 FOLDER_NAME = ''
13 PROJECT_NAME = 'CIGRE LV grid'
14 STUDY_CASE_NAME = 'Study Case'
15
16 MONITORED_VARIABLES = {
17     'PV System.ElmPvsys': ['m:P:bus1', 'm:Q:bus1'], # PV system power injection
18     '*.ElmTerm': ['m:u'] # Voltage magnitude at all ElmTerm elements (buses)
19 }
20
21 # activate project and study case
22 sim = PowerFactorySim(
23     folder_name=FOLDER_NAME,
24     project_name=PROJECT_NAME,
25     study_case_name=STUDY_CASE_NAME)
26
27 # get all buses in network
28 buses = sim.app.GetCalcRelevantObjects('*.ElmTerm')
29
30 # create result dictionaries
31 t = {}
32 P = {}
33 Q = {}
34
35 # simulate a short circuit in each bus and see how the PV responds
36 # (not very interesting, there is no controller)
37 for bus in buses:
38     # create short circuit on every bus
39     sim.create_short_circuit(
40         target_name=bus.loc_name+'.ElmTerm',
41         time=2.0.
```

The right pane shows the execution results in the console and plots. The console output includes:

```
In [1]: runfile('C:/Users/alekic/Desktop/KPIs/PythonPowerFactory4Milos/scripts/rms_ld.py', wdir='C:/Users/alekic/Desktop/KPIs/PythonPowerFactory4Milos/scripts')
Figures now render in the Plots pane by default. To make them also appear inline in the Console,
uncheck "Mute Inline Plotting" under the Plots pane options menu.
In [2]: runfile('C:/Users/alekic/Desktop/KPIs/PythonPowerFactory4Milos/scripts/rms_pv.py', wdir='C:/Users/alekic/Desktop/KPIs/PythonPowerFactory4Milos/scripts')
Reloaded modules: pfsim, powerfactory
In [3]: runfile('C:/Users/alekic/Desktop/KPIs/PythonPowerFactory4Milos/scripts/rms_sc.py', wdir='C:/Users/alekic/Desktop/KPIs/PythonPowerFactory4Milos/scripts')
Reloaded modules: pfsim, powerfactory
In [4]:
```

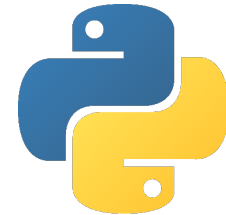
The plots pane shows a line plot of voltage magnitude (V) over time (s). The plot title is 'V (V)' and the x-axis is 'Time (s)'. The y-axis ranges from 0 to 1.0, and the x-axis ranges from 0 to 10. The plot shows a step function that starts at 1.0 and drops to 0.0 at approximately 2.0 seconds.

# Setting up the stage – PF interface for Python



# Setting up the stage – PF interface Python

- PF supports Python functions for
  - Change of element parameters
  - Setting up simulation properties
  - Performing simulation and saving results
- Disadvantage of PF+Python
  - Insufficient capabilities for creating power system components in PF from Python  
*It is possible, but difficult.*
- There is PF support in books and tutorials
  - Manual, PowerFactory User. "DigSILENT Power Factory." (2017).
  - López, Claudio David, and José Luis Rueda Torres. "Python scripting for DigSILENT PowerFactory: Leveraging the Python API for scenario manipulation and analysis of large datasets." *Advanced Smart Grid Functionalities Based on PowerFactory*. Springer, Cham, 2018. 19-48.





# Python class – Methods for reading/setting values

## Nodes voltages

- `get_bus_voltages(self)`

## Loads

- `get_all_loads_pq(self)`
- `set_all_loads_pq(self, p_load, q_load)`

## •Toggle

- `toggle_out_of_service(self, elm_name)`
- `toggle_switches(self, elm_name)`





## Python class – Methods for creating short circuits and events

- `create_short_circuit(self, target_name, time, duration=None, name='sc')`
- `delete_short_circuit(self, name='sc')`
  
- `create_load_event(self, target_name, time, p_change_percent, q_change_percent, name='lodc', type='step', ramp_duration=0.0)`
- `delete_load_event(self, name='lodc')`



# Python class – Methods for power flow simulation

- `prepare_loadflow(self, ldf_mode='balanced')`
  - Sets up power flow mode (balanced or unbalanced) and connects with PowerFactory simulator.
- `run_loadflow(self)`
  - Runs power flow.

Example: powerflow.py



# Python class – Methods for dynamic simulation

- `prepare_dynamic_sim(self, monitored_variables, sim_type='rms', start_time=0.0, step_size=0.01, end_time=10.0)`
  - Sets monitored variables and simulation type to rms or emt.
- `run_dynamic_sim(self)`
  - Runs dynamical simulation (rms or emt).
- `get_dynamic_results(self, elm_name, var_name)`
  - Getting simulation results.

# Python class

It is made as a package and it can be installed using flit!

```
flit install --symlink
```

# Task

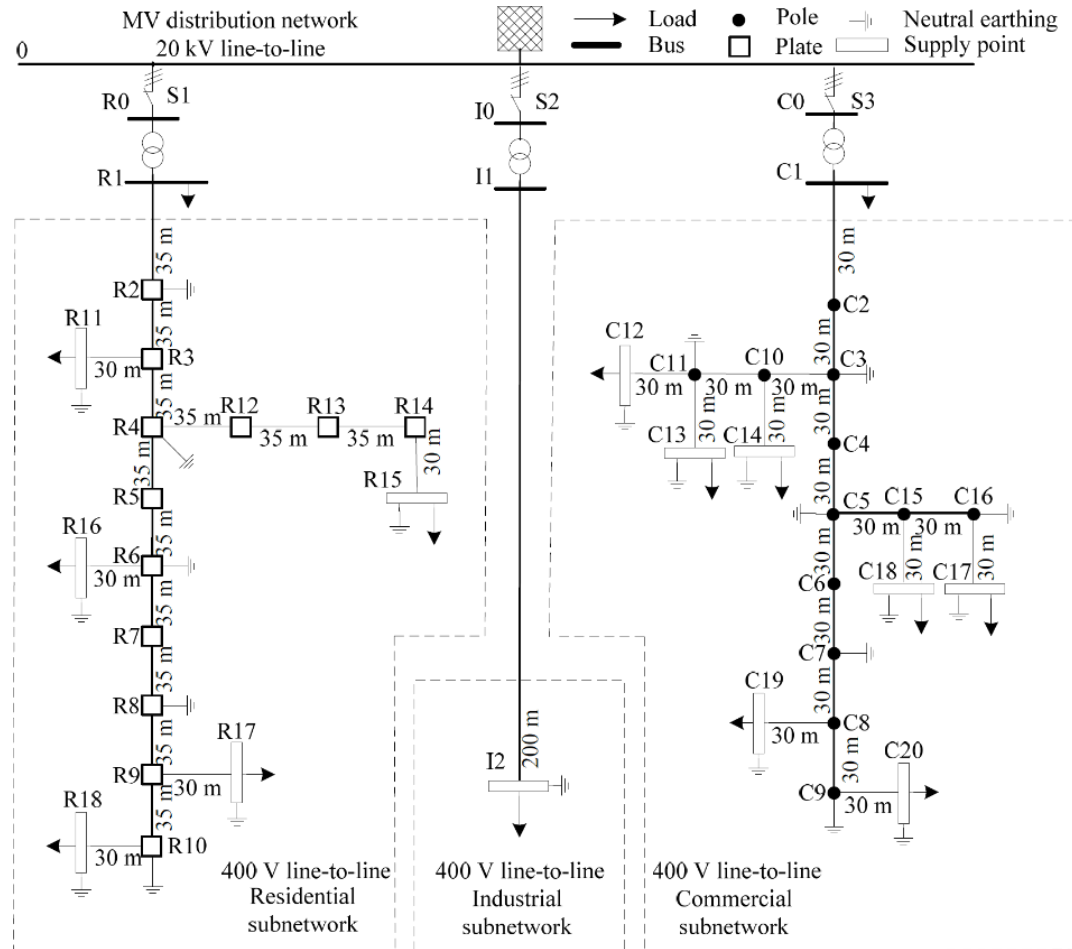
Design a tool for a fast estimation of possible PV penetration

- Enable PV placements on different nodes in distribution grids
- Change active/reactive power of PVs
- Change control of PVs
- Estimate maximum penetration and save results

# PV placements in the tool – PF

- Create distribution power grid (LV CIGRE on the right) in PF
- Name nodes with numbers
- Create PVs for all nodes in the distribution system and name them *PV\_nodenumber*.

*Now we can switch to Python!*



# PV placements in the tool – PF

The screenshot displays the DigSILENT PowerFactory 2020 software interface. The title bar indicates the project is "CIGRE LV grid". The menu bar includes FILE, EDIT, VIEW, INSERT, DATA, CALCULATION, OUTPUT, TOOLS, WINDOW, and HELP. The toolbar contains various icons for file operations, editing, and simulation.

The left sidebar shows the "Project Overview" panel with the following structure:

- Study Cases
  - 07/02/2106 07:28:15
    - Study Case
- Grids (1 active)
  - Grid

The main workspace displays a "Grid" view of the "MV distribution network 200V Line-to-line". The diagram shows a complex network of electrical components, including transformers, breakers, and lines, connected in a hierarchical structure. A dashed box highlights a specific section of the network.

The bottom panel shows the "Output Window" with the following content:

- Errors (0)
- Warnings (0)
- Information (3)
  - Activating Project CIGRE LV grid...
  - Study Case Study Case activated.
  - Project CIGRE LV grid activated.
- Events (0)
- Others (0)

The status bar at the bottom indicates the current view is "Grid", with coordinates X= 273.359, Y= 127.977, and a zoom level of D8 1800. The date and time are 07/02/2106 07:28:15.

# Developed tool in Python

## Class in Python properties

- Functions for intancing – used to start simulation in PF and get an access to PF settings/functions
- Get pointers to buses and the solutions of power flow/RMS/EMT
- Prepare power flow/RMS/EMT simulation settings
- Create and clear faults
- Change properties of PVs
  - Active/inactive
  - Power
  - Control principle

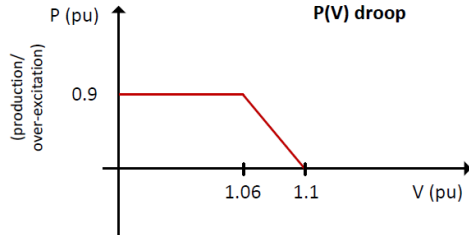


# Developed tool in Python

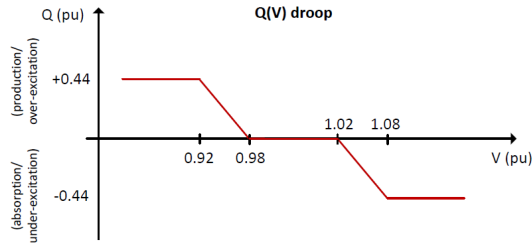
## PV penetretation simulation procedure

- Add number of newly installed DRESs equipped with one of the examined voltage control scheme:

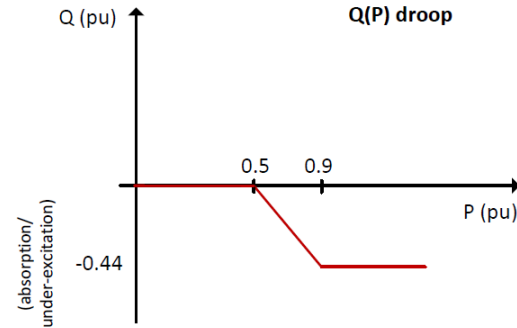
- $P(V)$  denoted as **constq** in PF



- $Q(V)$  is denoted as **qvchar** control



- $\cos(\phi)(P)$  is denoted as **cpchar**
- $\cos(\phi) = 1$  is denoted as **constc**
- $Q(P)$  is defined using QP curve and desired active power (**pgini**)



# Developed tool in Python

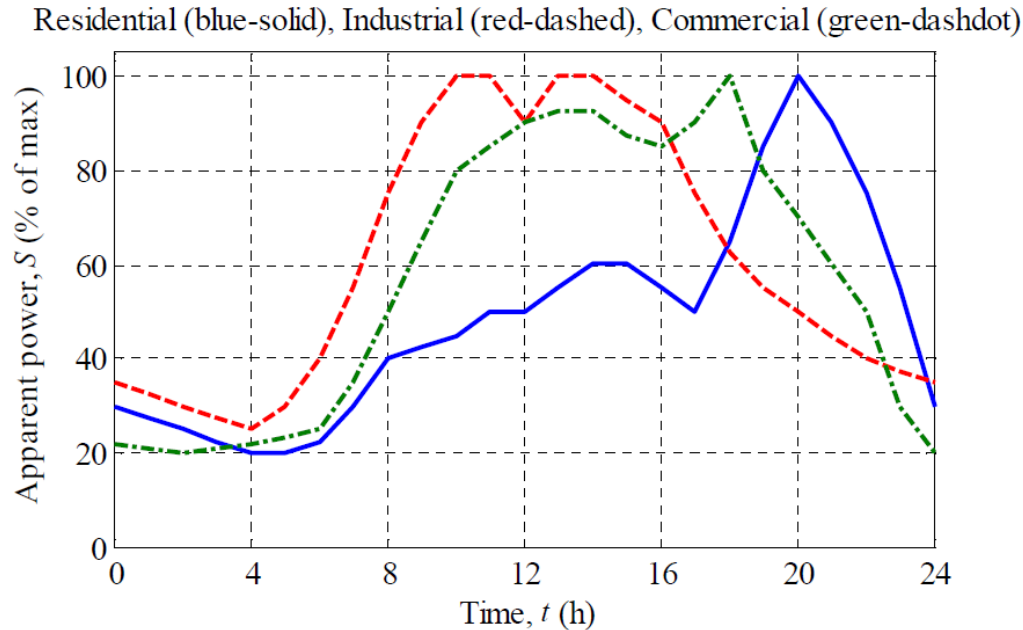
## PV penetretation simulation procedure

- Add number of newly installed DRESs (equipped with one of the examined voltage control scheme).
- Randomly select PV location and make it active or inactive.
- Change DRES installed capacity.
- Set voltage type control. It is set using local controller property in PF.
- Statistical analysis of the obtained results:
  - Voltage deviations from the nominal value. It reports voltage violation if for any bus voltage appears to be outside the set  $[V_{min}, V_{max}]$ .
  - Load violation in any circuit component.
  - Produce summary of the results.

# Developed tool in Python

## PV penetretation simulation procedure

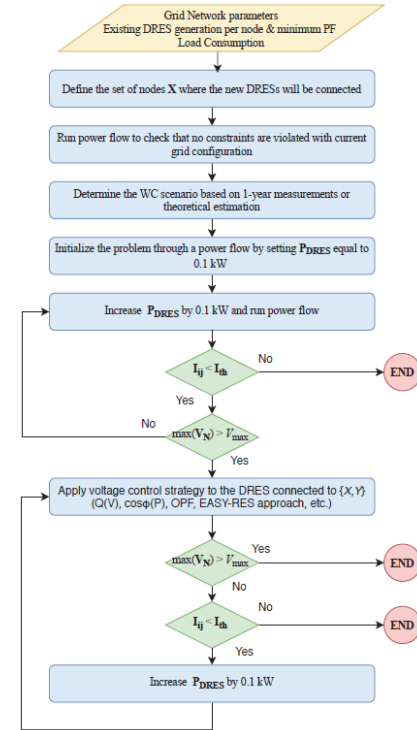
- Change load profiles to enable daily simulation



# Developed tool in Python

## PV penetretation simulation procedure

- Enable comparison with Easy-RES static control
  - Selection of P(V) scenario
  - Simulate for one time instance
  - Run simulation for one day



# Developed tool in Python

## Result interpretation

Results in csv file:

1. Marked bus voltage in p.u.
2. Given DRES power
3. Resulting losses
4. Given external grid power

File named

*"gridtype\_PV\_controltype\_X\_PVlocations.csv"*

0	1	10	11	12	13	14	15	16	17	18	2	3	4	5	6	7	8	9	DRES power [kW]	Ploss [MW]	External grid power [MW]
1.05	1.020643	0.691813	1.000382	0.72108	0.71829	0.715255	0.712655	0.705576	0.692848	0.689385	1.010841	1.000901	0.723219	0.715784	0.708354	0.704168	0.699583	0.694647	63	0.065986	0.386498776
1.05	1.02266	0.738081	1.004976	0.765205	0.762903	0.760044	0.757594	0.750608	0.739051	0.735802	1.01425	1.005562	0.766901	0.76006	0.753223	0.74944	0.745374	0.740741	72	0.049733	0.361447428
1.05	1.020213	0.677441	1.000213	0.709228	0.706844	0.703761	0.70112	0.692054	0.678497	0.674963	1.010522	1.000342	0.710972	0.702924	0.694883	0.69015	0.685368	0.680332	68	0.07118	0.386577916
1.05	1.02287	0.736123	1.005954	0.763815	0.761508	0.758644	0.75619	0.749068	0.737096	0.733838	1.014843	1.006328	0.765594	0.758639	0.751688	0.747798	0.743435	0.73879	82	0.050237	0.351945096
1.05	1.021347	0.704376	1.001842	0.733451	0.730854	0.72787	0.725314	0.718101	0.705392	0.70199	1.012098	1.00243	0.735347	0.728086	0.720831	0.716766	0.712011	0.707161	70	0.061244	0.374832807



Item 4: The EASY-RES Consortium

# The Consortium



ARISTOTLE  
UNIVERSITY OF  
THESSALONIKI



ΑΔΜΗΕ  
ΑΝΕΞΑΡΤΗΤΟΣ ΔΙΑΧΕΙΡΙΣΤΗΣ  
ΜΕΤΑΦΟΡΑΣ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ



This project has received funding from the European Union's Horizon 2020 Programme for research and innovation under Grant Agreement no 764090.



# Thank you!

Aleksandra Lekić

Affiliation: EASY-RES

E-Mail: [A.Lekic@tudelft.nl](mailto:A.Lekic@tudelft.nl)

EASY-RES website: <http://www.easyres-project.eu/>

This presentation reflects only the author's view. The Innovation and Networks Executive Agency (INEA) and the European Commission are not responsible for any use that may be made of the information it contains.