

The background of the slide is a composite image. It features a blue sky with white clouds and several wind turbines. Overlaid on this is a network of white lines connecting various circular icons. The icons include a laptop, a smartphone, a cloud, a Wi-Fi symbol, a robotic arm, and a solar panel. The text "communication simulation with agents" is centered in the lower half of the image, with "communication" and "agents" in blue and "simulation with" in orange.

communication simulation with agents

Frauke Oest, Emilie Frost, Malin Radtke

Who are we?



Frauke Oest

- > Senior Researcher @ OFFIS
- > PhD Student & Researcher @ Computer Science, University of Oldenburg
- > frauke.oest@offis.de



Emilie Frost

- > Researcher @ OFFIS
- > PhD Student & Researcher @ Computer Science, University of Oldenburg
- > emilie.frost@offis.de

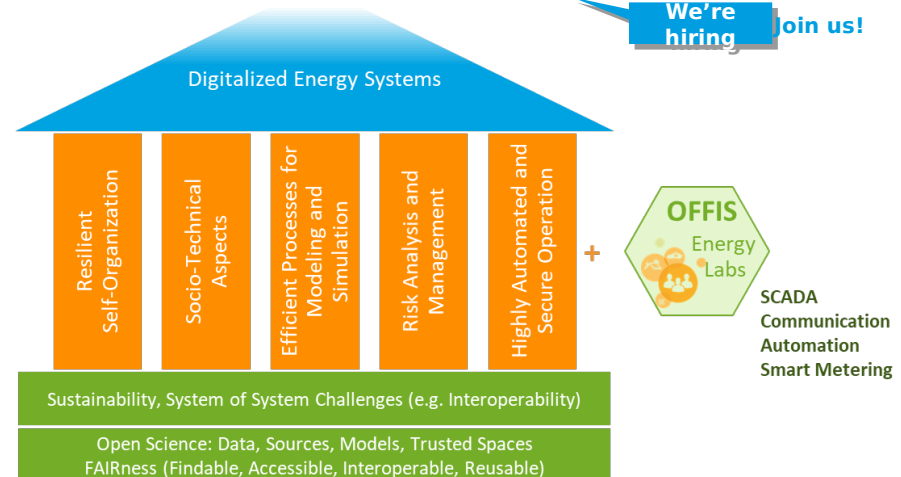


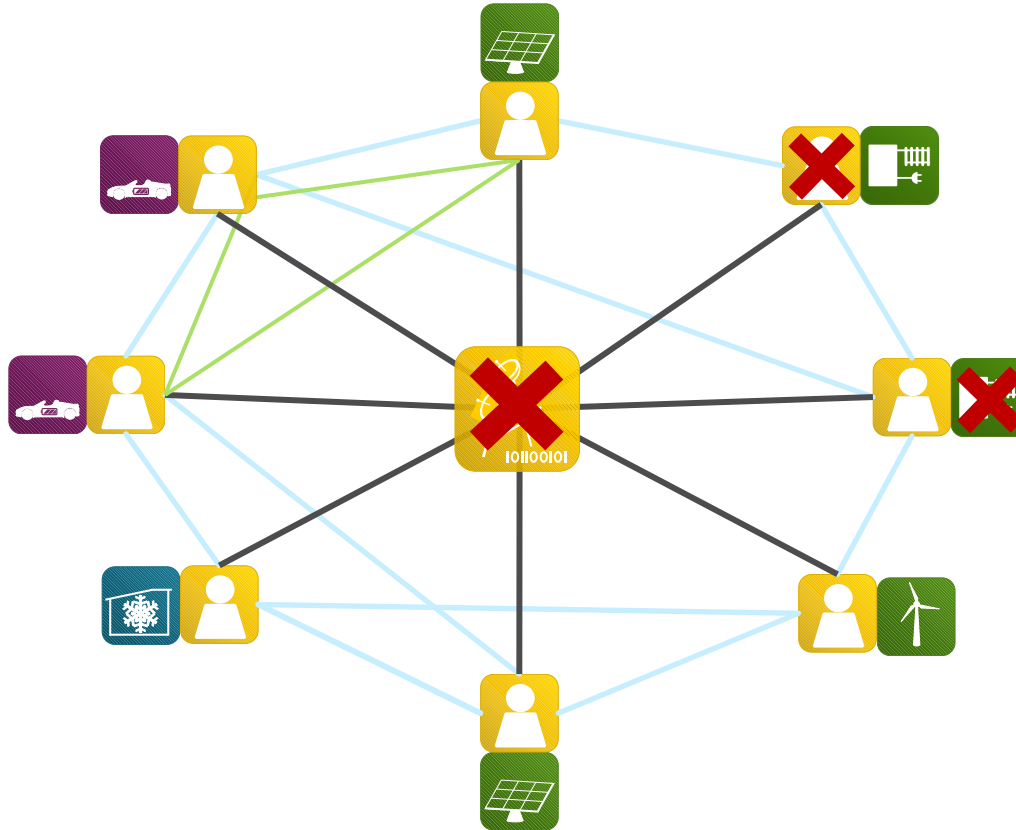
Malin Radtke

- > Researcher @ OFFIS
- > Master Student @ Computer Science, University of Oldenburg
- > malin.radtke@offis.de

OFFIS— Institute for Information Technology

- > State-funded Research Institute in Oldenburg, Germany (founded in 1991)
 - > 250 Scientist in 4 R&D Divisions (Energy, Health, Manufacturing, Society)
- > Energy Informatics at OFFIS & UOL
 - > 1995: First wind power information system in Germany
 - > 2003: Early “decentralized energy management systems”
 - > 2010: First Energy Informatics Professorship in Germany (we now have 4!)
 - > 2022: ~30 ongoing research projects
 - > Largest EI team in D/EU (>100 researchers)



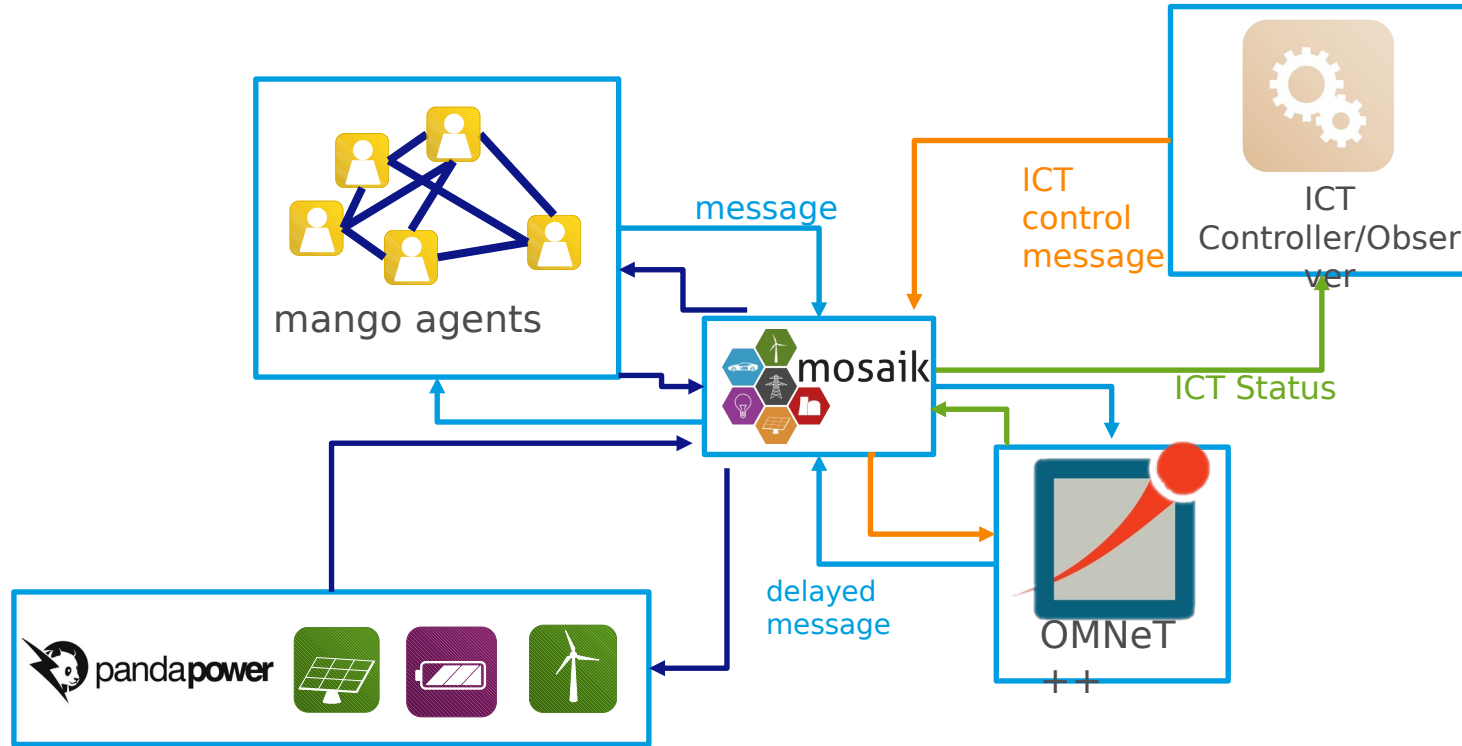


- Create MAS robust to failures
- Consider the interaction between power and communication system
- Optimize the communication flow for multiple smart grid applications

- > Ability to model and simulate wired and wireless communication systems with regards to different OSI-Layers (e.g. 5G, LTE-A, Ethernet)
- > Capability to flexibly and fast configure the network (e. g. by SDN)
- > Able to ensure QoS (e. g. traffic shaping, prioritization)
- > External control of networks
- > Ability to manipulate infrastructure devices in simulation runtime
- > Co-simulation with power system to integrate information of OT availability into the communication simulation

Envisioned Simulation Environment

with our MAS Framework mango and mosaik DES



Abilities and properties

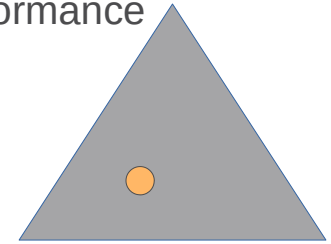
- > Discrete time and discrete event simulation
- > Accelerated and real time simulation
- > Ability to integrate IP-protected components
- > Scaling of simulations on compute clusters
- > Python-based
- > In principle: able to connect to simulators of all languages through TCP- and JSON-based connections

Open source (LGPL) <https://gitlab.com/mosaik>

Utility ecosystem

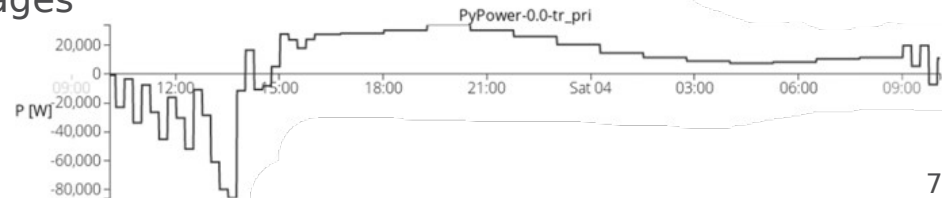
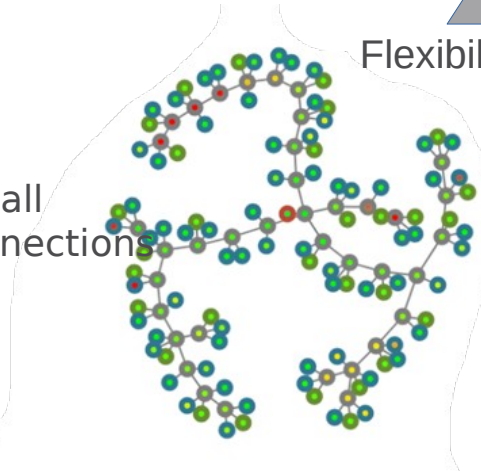
- > Simulation models
- > Wrappers for programming languages
- > Interfaces for simulation tools
- > Visualization and data storage

Performance



Flexibility

Usability





Time-based

- > Simulation components can only schedule steps for themselves (regular times)
- > In order to not miss any potential message from other components, simulators have to step themselves for every simulation step
- > Performance Issues in some scenarios, where very high time resolution is needed (communication simulation)

Event-based (coming with mosaik 3.0.0)

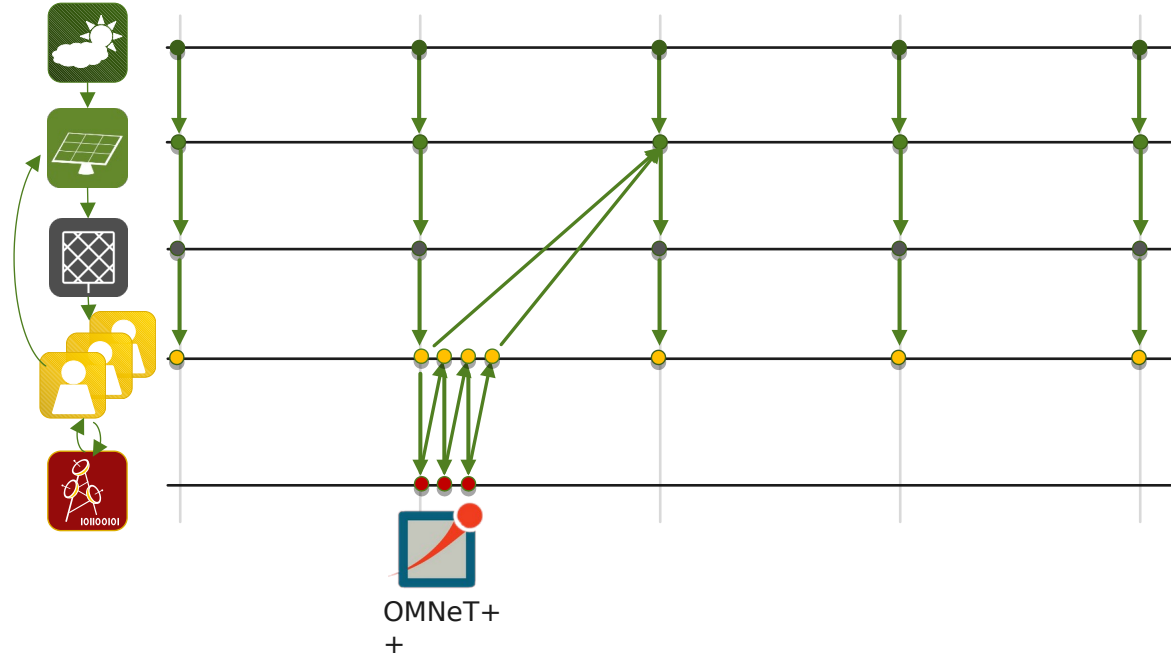
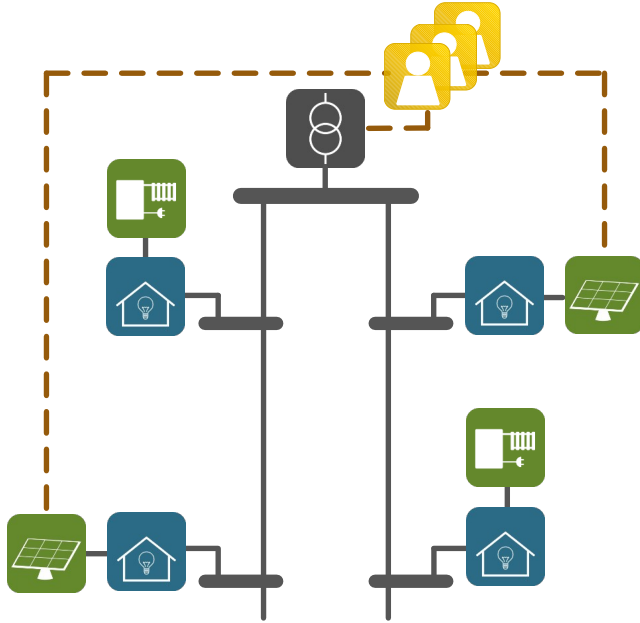
- > state of the system can change at a specific time: simulator is stepped as soon as there is new data available
- > Can optionally still trigger themselves

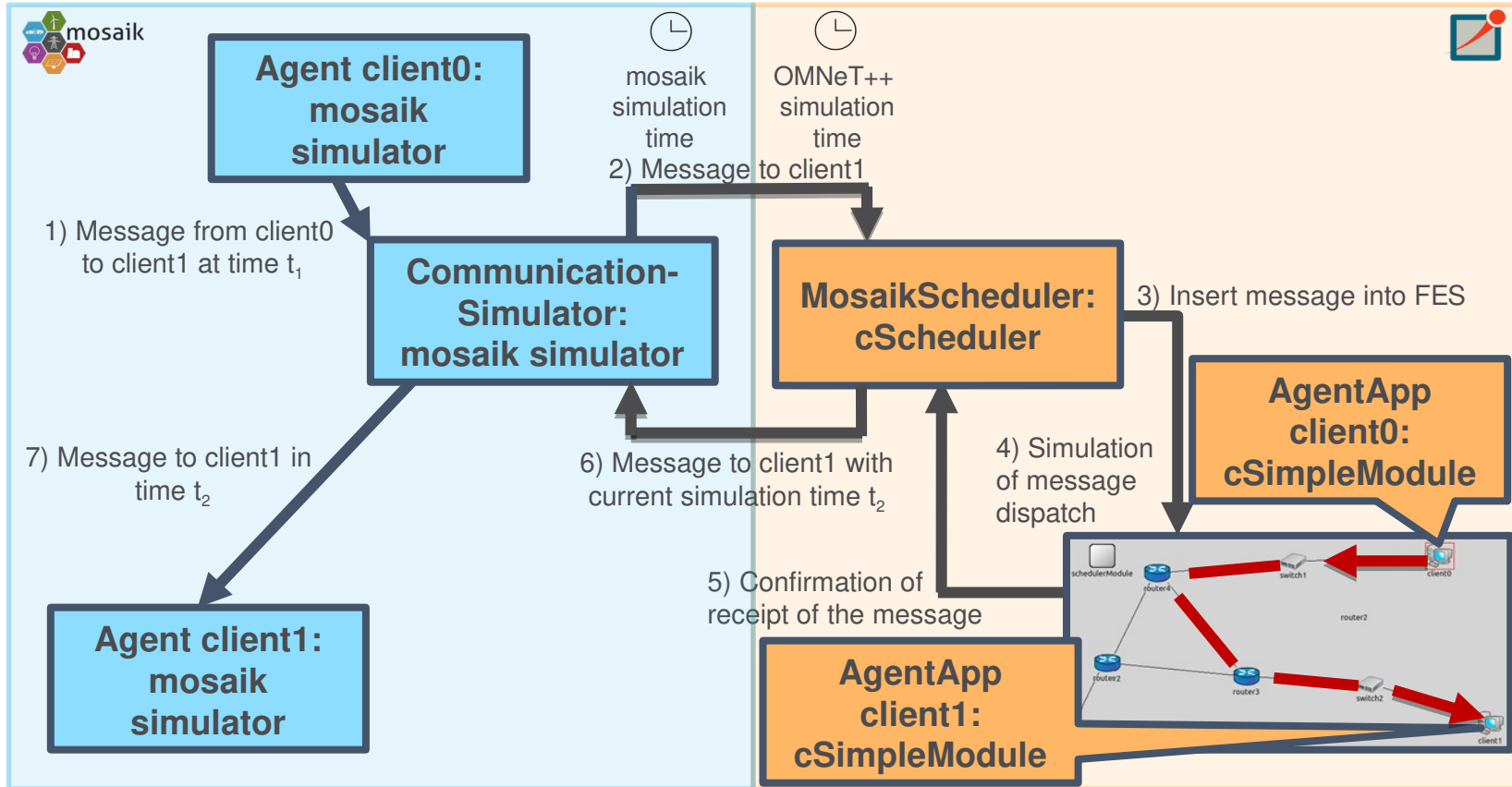


Event-based (coming with mosaik 3.0.0)

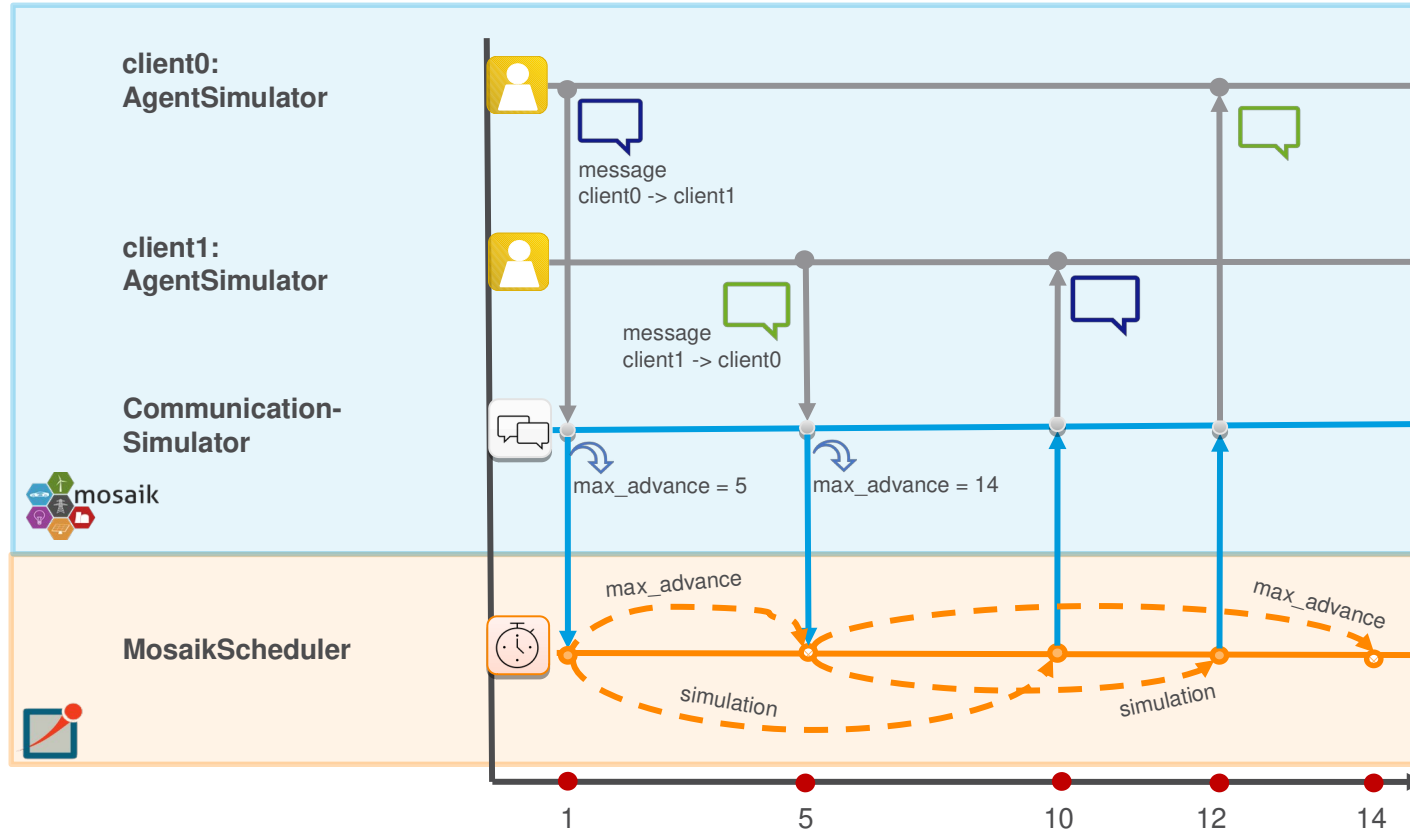
>Max_advance value

- >Information how far a simulator can advance in time without expecting new inputs
- >Simulator can progress until max_advance without being interrupted by mosaik and risking causality errors

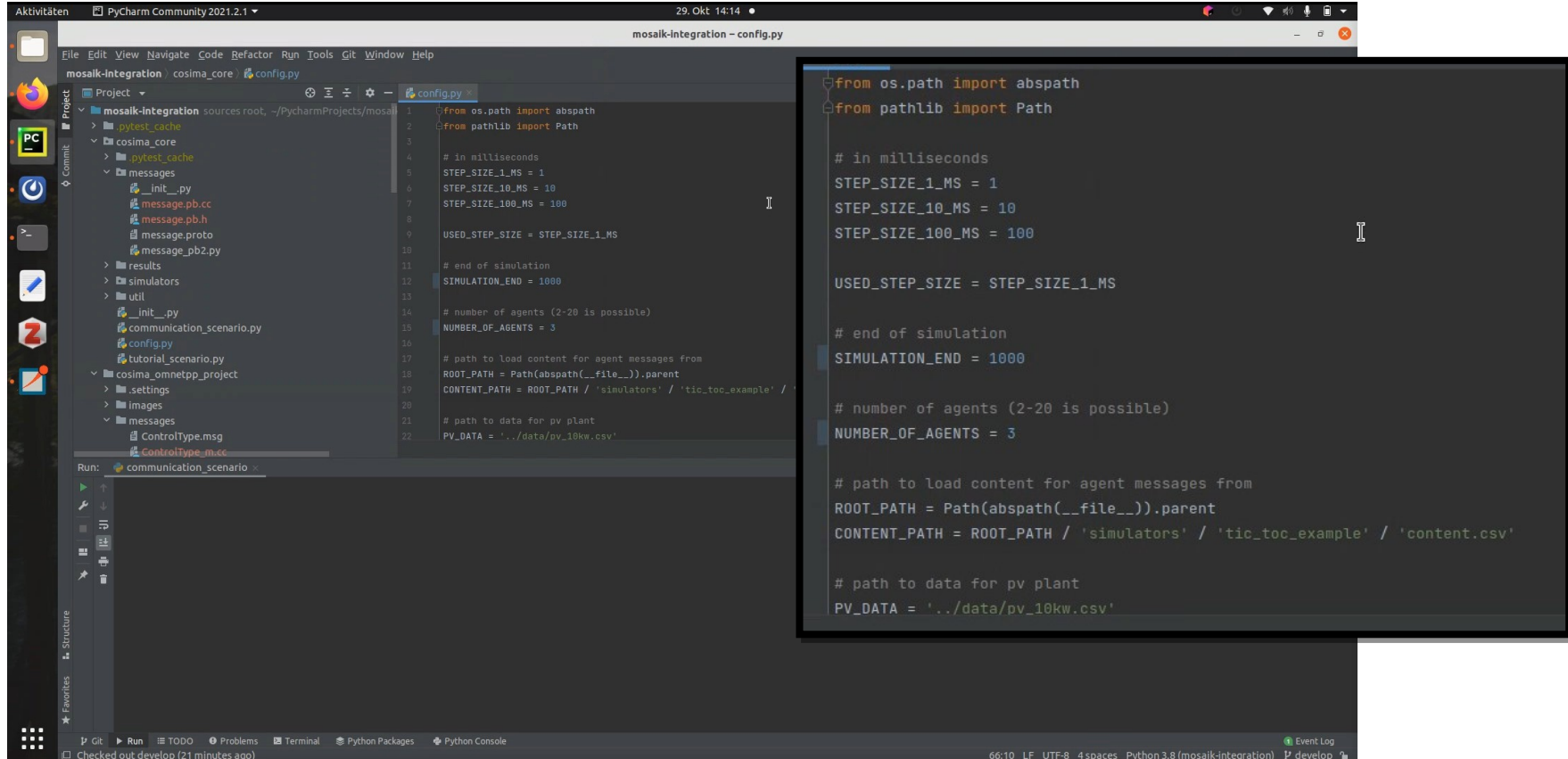




Synchronisation between mosaik and OMNeT++



- > Start modes using IDE or compiled executables (Qtenv or cmd)
- > Adjust the number of agents
- > Apply infrastructure changes (dis- and reconnects of clients, routers and switches) during simulation
- > Connect PV plant simulators to agent(s)
- > Multiple exemplary OMNeT++ networks
- > Overview of the simulation based on collected information
 - > e.g., number of simulation steps and evaluation graphs



The screenshot shows the PyCharm IDE interface. The main editor displays the `config.py` file for the `mosaik-integration` project. The file contains configuration parameters for a simulation, including step sizes, simulation end time, number of agents, and file paths. A zoomed-in view of the code is shown on the right side of the image.

```
1 from os.path import abspath
2 from pathlib import Path
3
4 # in milliseconds
5 STEP_SIZE_1_MS = 1
6 STEP_SIZE_10_MS = 10
7 STEP_SIZE_100_MS = 100
8
9 USED_STEP_SIZE = STEP_SIZE_1_MS
10
11 # end of simulation
12 SIMULATION_END = 1000
13
14 # number of agents (2-20 is possible)
15 NUMBER_OF_AGENTS = 3
16
17 # path to load content for agent messages from
18 ROOT_PATH = Path(abspath(__file__)).parent
19 CONTENT_PATH = ROOT_PATH / 'simulators' / 'tic-toc-example' /
20
21 # path to data for pv plant
22 PV_DATA = '../data/pv_10kw.csv'
```

The zoomed-in view on the right shows the following code:

```
from os.path import abspath
from pathlib import Path

# in milliseconds
STEP_SIZE_1_MS = 1
STEP_SIZE_10_MS = 10
STEP_SIZE_100_MS = 100

USED_STEP_SIZE = STEP_SIZE_1_MS

# end of simulation
SIMULATION_END = 1000

# number of agents (2-20 is possible)
NUMBER_OF_AGENTS = 3

# path to load content for agent messages from
ROOT_PATH = Path(abspath(__file__)).parent
CONTENT_PATH = ROOT_PATH / 'simulators' / 'tic-toc-example' / 'content.csv'

# path to data for pv plant
PV_DATA = '../data/pv_10kw.csv'
```



Scenarios

Large Networks, 50 agents with PV plants



```
3
4 # in milliseconds
5 STEP_SIZE_1_MS = 1
6 STEP_SIZE_10_MS = 10
7 STEP_SIZE_100_MS = 100
8
9 USED_STEP_SIZE = STEP_SIZE_1_MS
10
11 # end of simulation
12 SIMULATION_END = 1000
13
14 # number of agents (2-20 is possible)
15 NUMBER_OF_AGENTS = 3
16
17 # path to load content for agent messages from
18 ROOT_PATH = Path(abspath(__file__)).parent
19 CONTENT_PATH = ROOT_PATH / 'simulators' / 'tic_toc_example'
20
21 # path to data for pv plant
22 PV_DATA = '../data/pv_10kw.csv'
23 START = '2014-01-01 00:00:00'
24
```

```
# in milliseconds
STEP_SIZE_1_MS = 1
STEP_SIZE_10_MS = 10
STEP_SIZE_100_MS = 100

USED_STEP_SIZE = STEP_SIZE_1_MS

# end of simulation
SIMULATION_END = 1000

# number of agents (2-20 is possible)
NUMBER_OF_AGENTS = 3

# path to load content for agent messages from
ROOT_PATH = Path(abspath(__file__)).parent
CONTENT_PATH = ROOT_PATH / 'simulators' / 'tic_toc_example' / 'content.csv'

# path to data for pv plant
PV_DATA = '../data/pv_10kw.csv'
START = '2014-01-01 00:00:00'
```

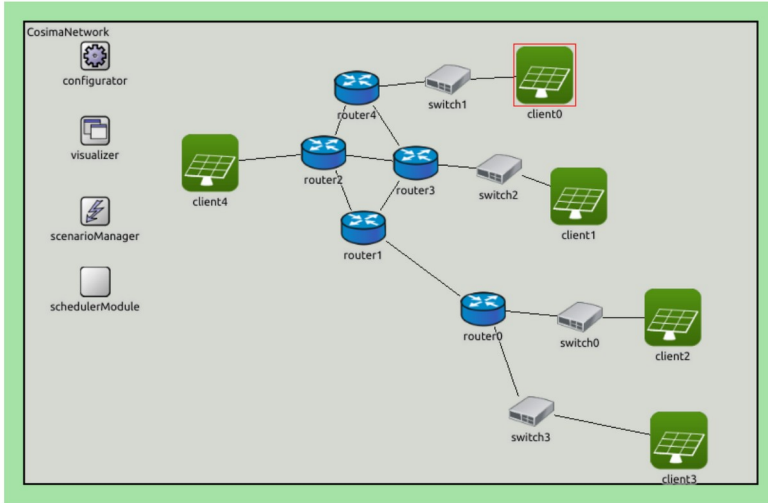


Fig. 3. Small network set-up in OMNeT++

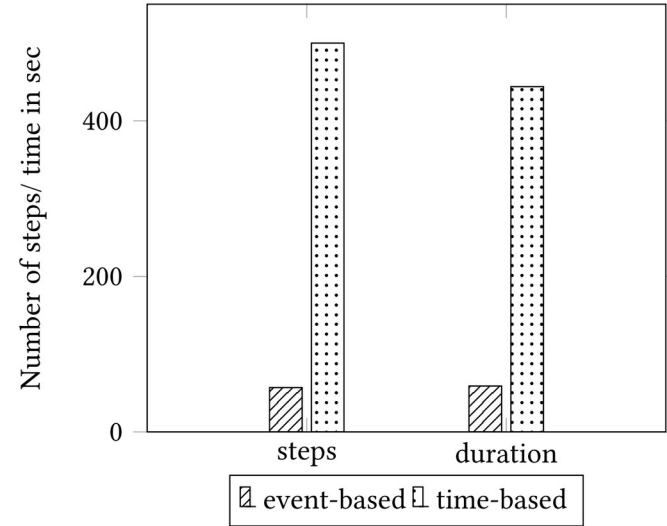
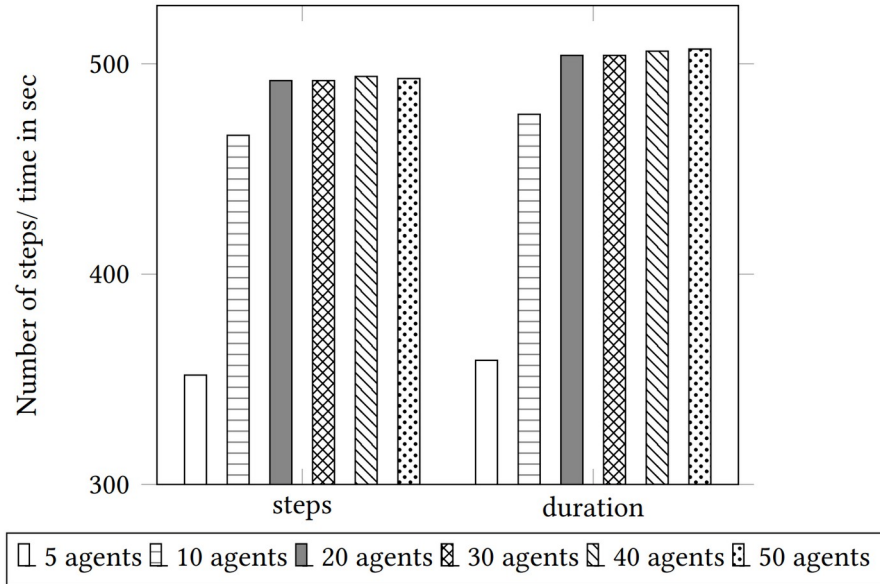


Fig. 4. Evaluation of steps and simulation duration

The number of simulation steps and the simulation duration are significantly higher for the time-based simulation.



The number of simulation steps and the simulation duration do not significantly increase with the number of agents.

Fig. 6. Evaluation of steps and simulation duration

- > Integrate the multiagent system library *mango* (<https://gitlab.com/mango-agents/mango>)
- > Enhance usability
- > Add more functionality to the ICT-controller

Further Information:

- > cosima.offis.de
 - > <https://cosima.readthedocs.io>
 - > <https://gitlab.com/mosaik/examples/cosima>
 - > Oest, F., Frost, E., Radtke, M., & Lehnhoff, S. (2022). *Coupling OMNeT++ and mosaik for integrated Co-Simulation of ICT-reliant Smart Grids*. In *arXiv*. <https://doi.org/10.48550/arXiv.2209.12550>
- > <https://www.offis.de/en/applications/energy/distributed-artificial-intelligence.html>

