TradeRES

New Markets Design & Models for
100% Renewable Power Systems

AMIRIS

Installation, Execution and Market Design Parametrisation

Market Designs in Germany and the EU – Research &Tools Workshop

12th of October 2023, online

Christoph Schimeczek (DLR)

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

Motivation
Market Modelling with AMIRIS

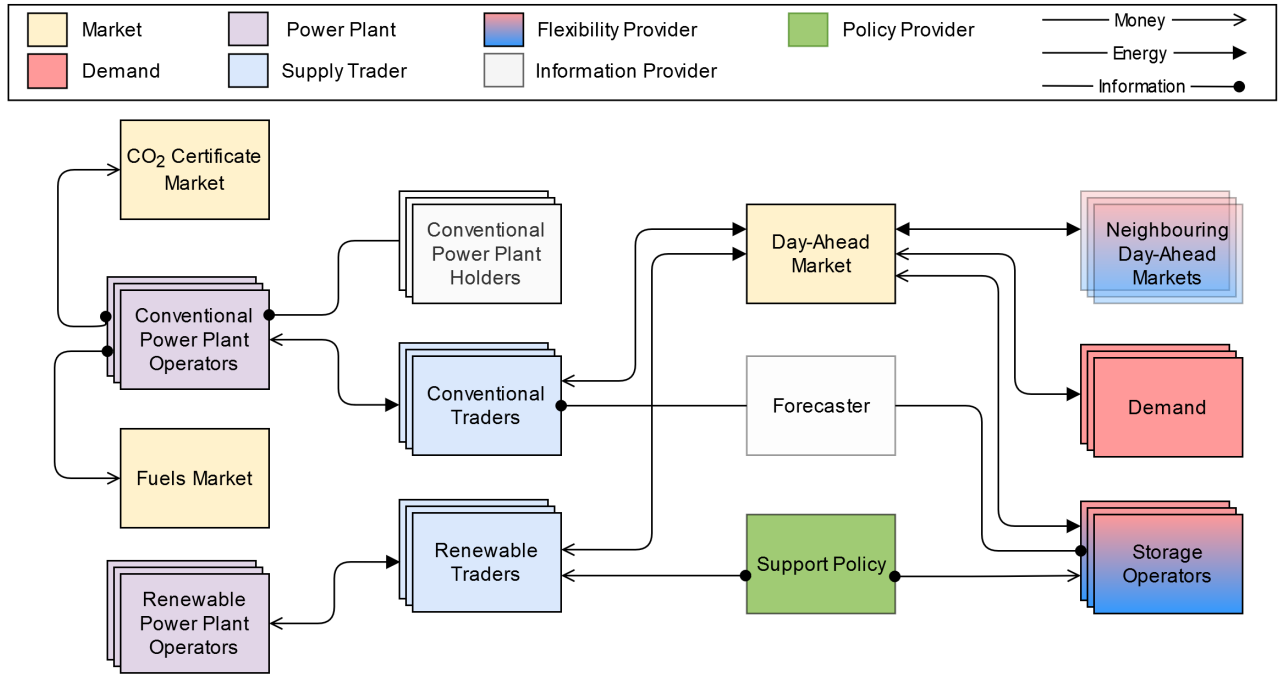
- Simulate trading and operation of power generation plants and flexibility options
- Model business-oriented behaviour under uncertainty
- Temporal resolution: ≤ hourly
- Spatial resolution: market zone(s)

Input

- Power plant park
- RES-E feed in potential
- Demand
- Efficiencies
- Availabilities
- Fuel prices
- CO_2 prices

Output

- Electricity prices
- Plant dispatch, FLH
- Market values
- System costs
- Costs for support instruments
- CO_2 emissions



https://dlr-ve.gitlab.io/esy/amiris/home/

AMIRIS

Agents

AMIRIS
Agent Types

Markets
- Determine prices

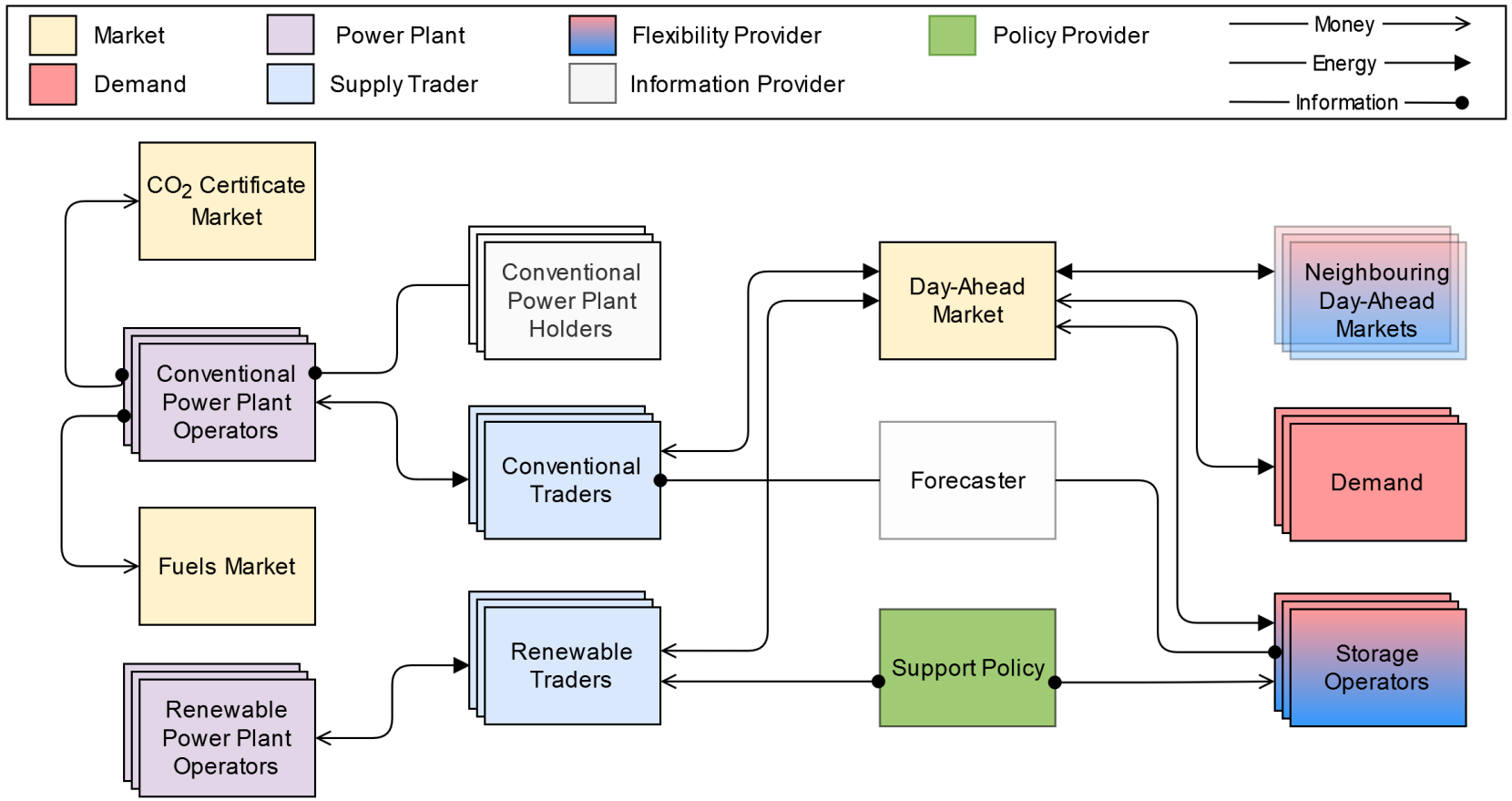
Plant operators
- Control power plants

Traders
- Fulfil marketing strategies

Flexibility providers
- Optimise dispatch

Information provider
- Create forecasts

Policy
- Provide support



AMIRIS

Interactions

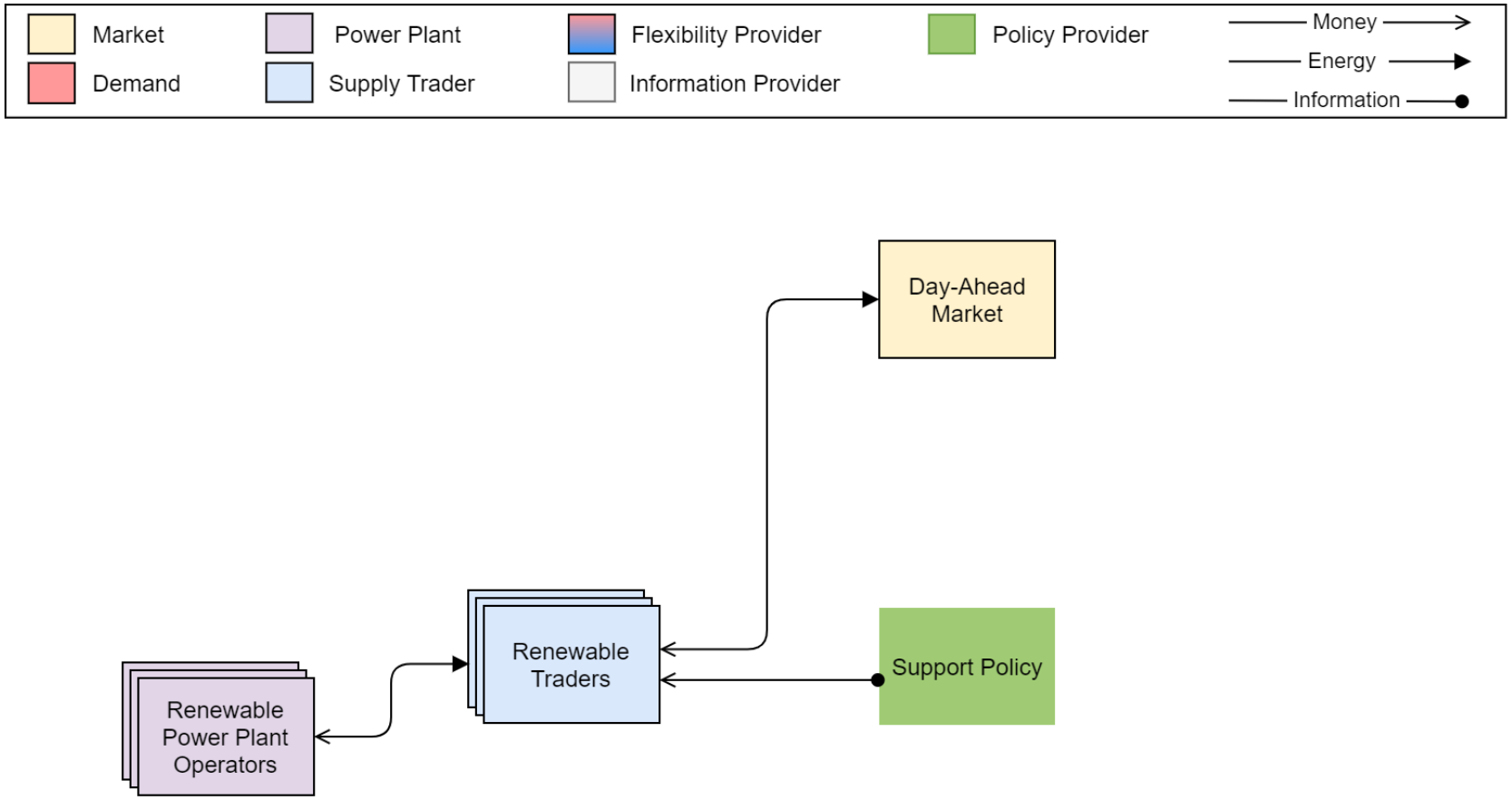
AMIRIS Interactions
Renewables

Power Plant Operator
- Calculate marginal cost
- Dispatch power plants

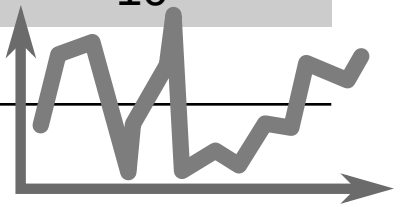
Renewable Trader
- Create bid
- Request support

Support Policy
- Calculate support tariffs
- Provide support funding

Day-Ahead Market
- Clears Market



Renewables
Power Plant Operator

Input parameter	Value
EnergyCarrier	WindOn
InstalledPowerInMW	1000
OpexVarInEURperMWH	10
YieldProfile	

Actions

1) *Calculate power potential*
2) *Calculate marginal costs*
3) *Send marginals to Trader*
4) *Receive assignment*
5) *Dispatch plants*

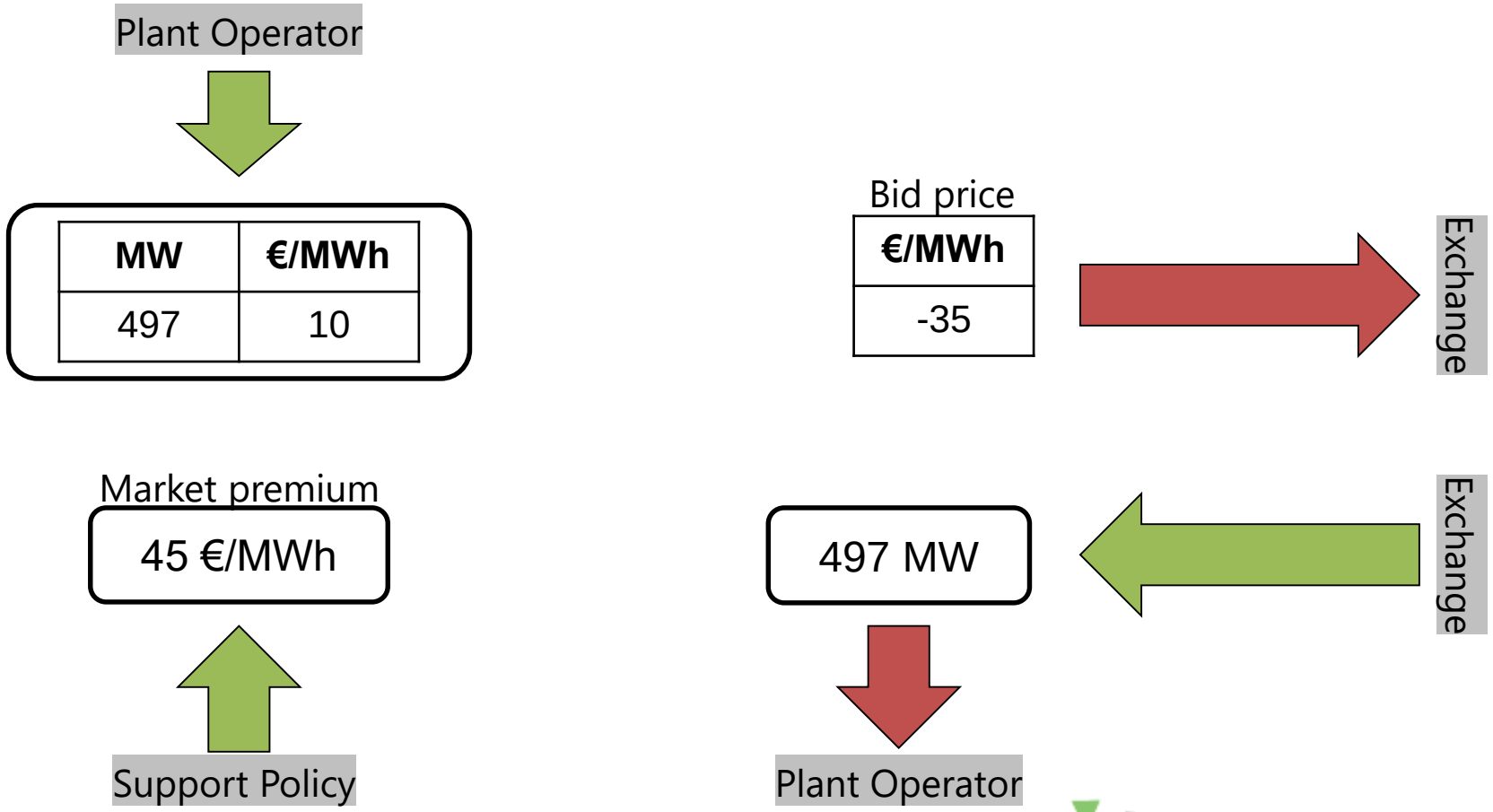
MW	€/MWh
497	10

497 MW

Renewables
Trader

Actions

1) *Receive marginal costs*
2) *Check support instrument*
3) *Derive bid*
4) *Send bids to Exchange*
5) *Receive awards*
6) *Forward power to operator*



Renewables
Support Policy

Actions

1) *Register clients*
2) *Track power prices*
3) *Track feed-in potentials*
4) *Calculate variable tariffs*
5) *Provide support*

Trader

Carrier	Instrument
PV	Feed-in tariff
WindOn	Market premium

Exchange

12 €/MWh

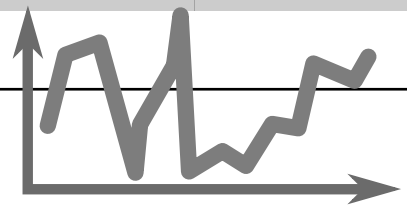
Plant Operator

Carrier	Feed-in
PV	0 MW
WindOn	497 MW

Carrier	Instrument	Support
PV	Feed-in tariff	90 €/MWh
WindOn	Market premium	45 €/MWh

Trader

Demand
Trader

Input parameter	Value
ValueOfLostLoad	3000
DemandSeries	

Actions

1) *Create bid*
2) *Send bid(s) to Exchange*

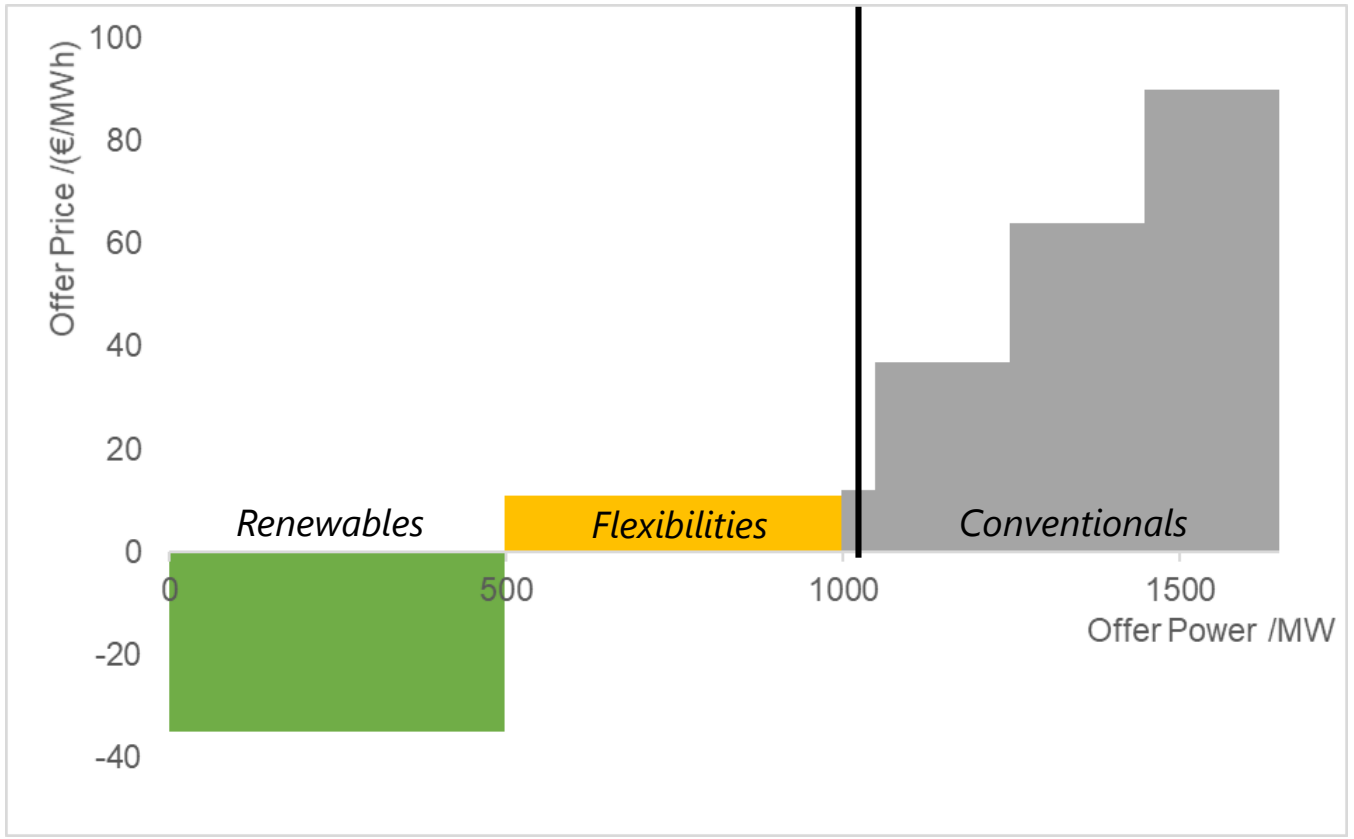
MW	€/MWh
1017	3000

Exchange

Energy Exchange
Market Clearing

Actions

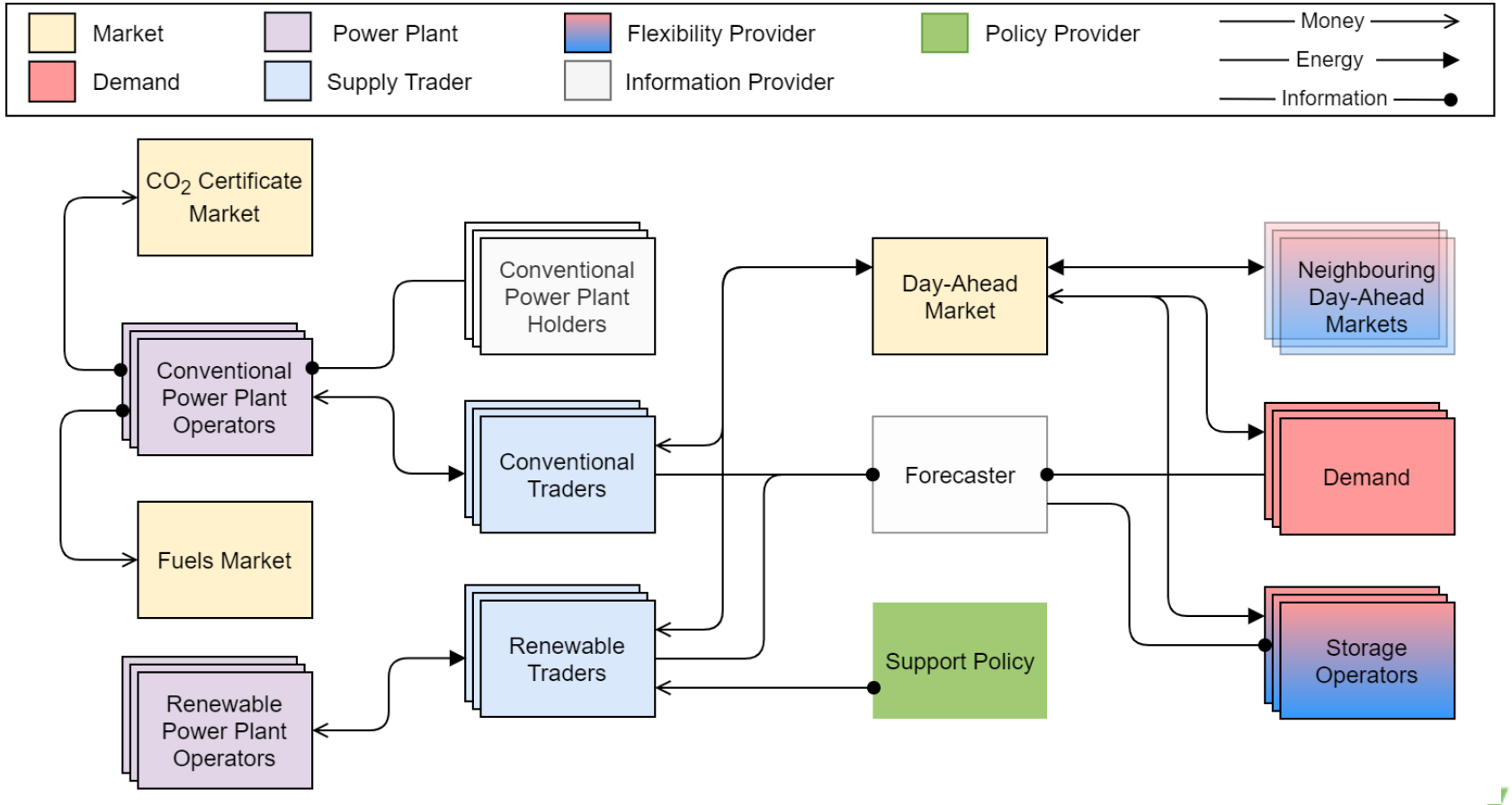
1) *Receive bids*
2) *Clear market*
3) *Send awards*



MW	€/MWh
497	12
500	12
20	12

Trader

AMIRIS Agents
Overview



AMIRIS

Setup

Setup
Requirements

- Java JDK 11

```
(base) PS C:\> java --version
openjdk 11.0.9.1 2020-11-04
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.9.1+1)
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.9.1+1, mixed mode)
```

  - Obtain from, e.g., https://adoptium.net/

- Python 3.8 / 3.9

```
(base) PS C:\> python --version
Python 3.9.7
```

  - Obtain from, e.g., https://github.com/conda-forge/miniforge#mambaforge

Setup
AMIRIS

- Create environment

```
(base) PS C:\> mamba create -n AmirisEnv python=3.8
```

- Activate environment

```
(base) PS C:\> conda activate AmirisEnv
```

- Install *amirispy*

```
(AmirisEnv) PS C:\> pip install amirispy
```

- Create folder:

```
(AmirisEnv) PS C:\> mkdir amiris; cd amiris
```

- Install *AMIRIS*:

```
(AmirisEnv) PS C:\amiris> amiris install
```

Setup
Files

examples ← configuration files

amiris-core_2.0.0-alpha.8-jar-with-dependencies.jar ← AMIRIS executable

fameSetup.yaml ← ignore today!

examples/

- Austria2019
- Germany2019
- Simple

} three example scenarios

Examples/Simple/

- contracts
- timeseries
- LICENCE.md
- scenario.yaml ← Important file: Defines what is happening in simulation
- schema.yaml

Setup
Run AMIRIS

```
(AmirisEnv) PS C:\amiris> amiris run
usage: amiris run [-h] --jar JAR --scenario SCENARIO
                  [--output OUTPUT]
amiris run: error: the following arguments are required: --jar/-j, --scenario/-s
```

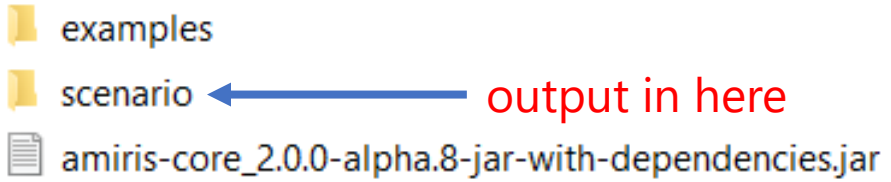
Required arguments

- -j AMIRIS executable
- -s Scenario file

```
(AmirisEnv) PS C:\amiris> amiris run -j .\amiris-core_2.0.0-alpha.8-jar-with-dependencies.jar
 -s .\examples\Simple\scenario.yaml
```

Console output

```
14:18:38 – PRINT – Start running AMIRIS
Starting up 1 processes.
Warm-up completed after 1 ticks.
04.10.2023 14:18:39:: Simulation completed! Ran 219 ticks in 258 ms
14:18:40 – PRINT – Successfully executed AMIRIS. See your results in '.'
```



Setup
Redirect output

```
(AmirisEnv) PS C:\amiris> amiris run -h
usage: amiris run [-h] --jar JAR --scenario SCENARIO [--output OUTPUT]

optional arguments:
  -h, --help            show this help message and exit
  --jar JAR, -j JAR     Path to 'amiris-core_<version>-jar-with-dependencies.jar'
  --scenario SCENARIO, -s SCENARIO
                        Path to a scenario yaml-file
  --output OUTPUT, -o OUTPUT
                        Directory to write output to
```

use this

```
(AmirisEnv) PS C:\amiris> amiris run -j .\amiris-core_2.0.0-alpha.8-jar-with-dependencies.jar
 -s .\examples\Simple\scenario.yaml -o simple
```

- examples
- simple ← output now in here

Setup
Results

- ConventionalPlantOperator.csv
- ConventionalPlantOperator_DispatchedP...
- ConventionalPlantOperator_VariableCost...
- ConventionalTrader.csv
- DemandTrader.csv
- EnergyExchange.csv
- NoSupportTrader.csv
- VariableRenewableOperator.csv

AgentId	TimeStep	TotalAwardedPowerInMW	ElectricityPriceInEURperMWH
1	01.01.2021 00:00	12431	267.4721054
1	01.01.2021 01:00	11416	262.9066734
1	01.01.2021 02:00	11163	260.8119727
1	01.01.2021 03:00	11036	257.4786831
1	01.01.2021 04:00	11192	256.4702082
1	01.01.2021 05:00	12177	256.2193284
1	01.01.2021 06:00	12685	256.2193284
1	01.01.2021 07:00	15222	259.7771467
1	01.01.2021 08:00	16491	260.2935264
1	01.01.2021 09:00	17125	257.9859146
1	01.01.2021 10:00	17378	255.7190453
1	01.01.2021 11:00	16997	255.4696391
1	01.01.2021 12:00	16237	257.2258181
1	01.01.2021 13:00	15476	256.4702082
1	01.01.2021 14:00	15222	259.5197279
1	01.01.2021 15:00	14968	262.3798356
1	01.01.2021 16:00	15095	265.3039864
1	01.01.2021 17:00	15729	265.8426993
1	01.01.2021 18:00	16491	264.7674623
1	01.01.2021 19:00	17505	263.1708901
1	01.01.2021 20:00	18012	260.035079
1	01.01.2021 21:00	17251	250.822094
1	01.01.2021 22:00	16744	0
1	01.01.2021 23:00	14968	0

AMIRIS

Parametrisation

Parametrisation

Scenario: Main config file to bundle all simulation properties

Open: examples/Germany2019/scenario.yaml

scenario.yaml

schema	*input validation*
general properties	*simulation start/end*
agents	*agents and their parameters*
contracts	*agent interactions*

File name is arbitrary

Split scenario.yaml into separate files, e.g. schema, contracts, etc., and join them using ***!include****, see https://gitlab.com/fame-framework/fame-io#split-and-join-multiple-yaml-files*

Parameterisation
General Properties

- Define
  - start and end of simulation
  - which random seed to use

```
GeneralProperties:
  RunId: 1          <-- ignore
  Simulation:
    StartTime: 2018-12-31_23:58:00
    StopTime: 2019-12-31_23:58:00
    RandomSeed: 1
  Output:           <-- ignore
```

*FAME's time definition **always** uses 365 days / 8760 hours per year, see also https://gitlab.com/fame-framework/wiki/-/wikis/architecture/decisions/TimeStamp*

YAML is indentation-based (2 spaces)

Parameterisation
Agents

- Define
  - agents
  - their type, ID, and attributes.

- Supported data types:
  - integer, floating point, enums, timeseries

In YAML, dash is used to denote lists

```yaml
Agents:
  - Type: EnergyExchange
    Id: 1
    Attributes:
      DistributionMethod: SAME_SHARES
      GateClosureInfoOffsetInSeconds: 11

  - Type: CarbonMarket
    Id: 3
    Attributes:
      OperationMode: FIXED
      Co2Prices: "./timeseries/co2_price.csv"

  - Type: FuelsMarket
    Id: 4
    Attributes:
      FuelPrices:
        - FuelType: LIGHNITE
          Price: 5.00
          ConversionFactor: 1.0
        - FuelType: NATURAL_GAS
          Price: "./timeseries/natural_gas_cost.csv"
          ConversionFactor: 1.0
```

All agents are explained in AMIRIS-Wiki: https://gitlab.com/dlr-ve/esy/amiris/amiris/-/wikis/Classes/Classes

*Every agent **must** have a unique ID within the simulation. This is how agents address each other.*

Execute
Germany2019

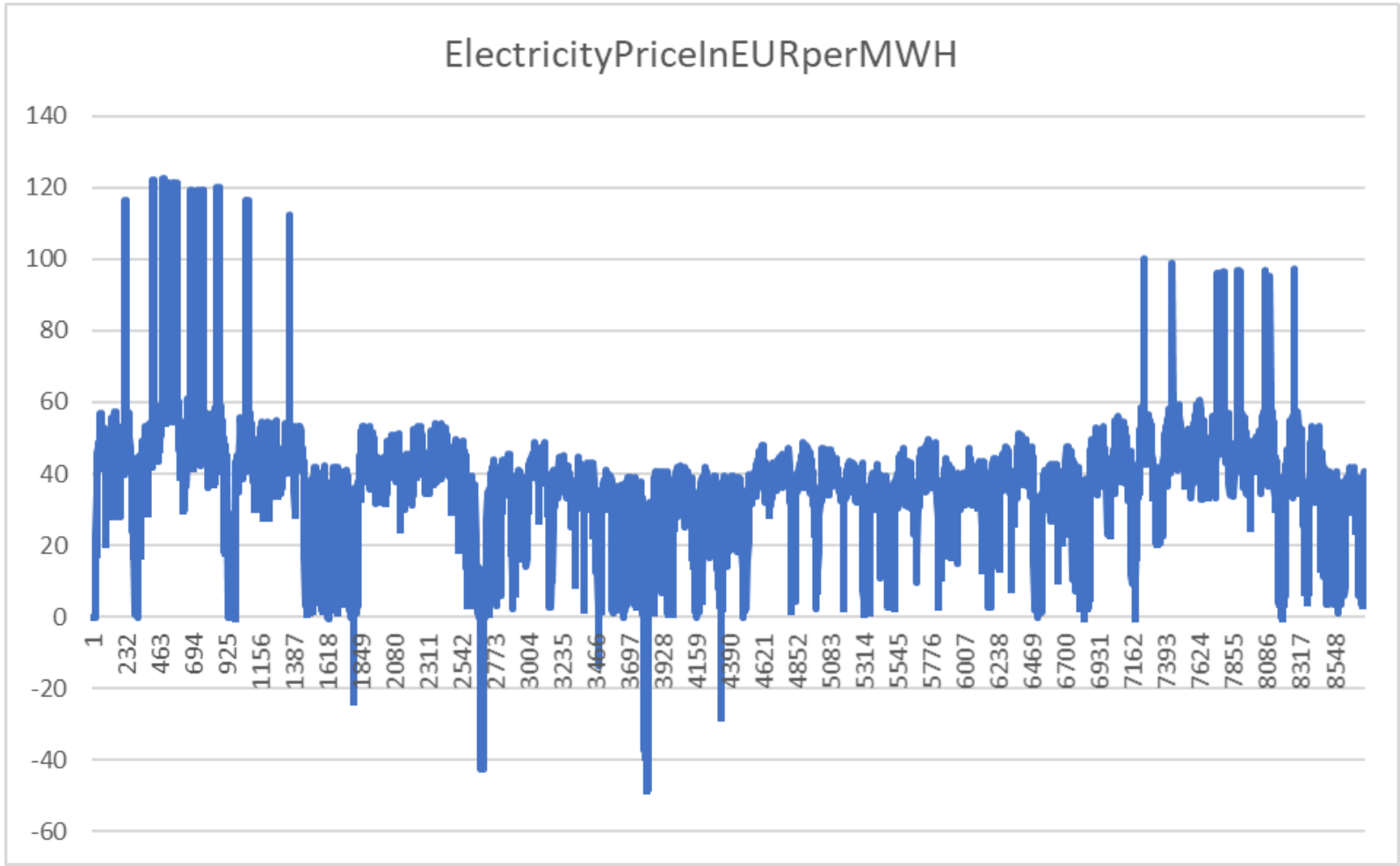
```
(base) PS C:\amiris> amiris run -j .\amiris-core_2.0.0-alpha.8-jar-with-dependencies.jar
 -s .\examples\Germany2019\scenario.yaml -o germany
```

examples

germany ← output in here

simple

→ Plot electricity price time series in “EnergyExchange.csv”



Parameterisation
Contracts

Define **when** agents send **what** data to **which** other agents

You **can** define agents without contracts:
they simply do nothing. There is no check!

Parameterisation
Contracts: Nice to know

Open: examples/Germany2019/contracts/conventionals.yaml

- Simulations often require *many* contracts!
- Contracts are often *similar*!
- Short notations available:
  - 1:N → one sender to multiple receivers
  - N:1 → one receiver from multiple senders
  - M[1:1] → m times one sender to **one** receiver

- Sender / receiver lists *repeat* often!
- Use YAML anchors to replace similar lists
  - Define: &anchorName <something>
  - Reference: *anchorName

```yaml
AgentGroups:
  - &builders  [2000, 2001, 2002, 2003, 2004, 2005]
  - &traders   [1000, 1001, 1002, 1003, 1004, 1005]
  - &operators [500, 501, 502, 503, 504, 505]
  - &exchange 1
  - &carbonMarket 3
  - &fuelsMarket 4
  - &forecaster 6

Contracts:
################################
# -- PlantBuildingManagement -- #
################################
  - SenderId: *builders
    ReceiverId: *operators
    ProductName: PowerPlantPortfolio
    FirstDeliveryTime: -60
    DeliveryIntervalInSteps: 31536000

##############################
# -- Forecast Preparation -- #
##############################
  - SenderId: *forecaster
    ReceiverId: *traders
    ProductName: ForecastRequest
    FirstDeliveryTime: -26
    DeliveryIntervalInSteps: 3600
```

anchors

comment

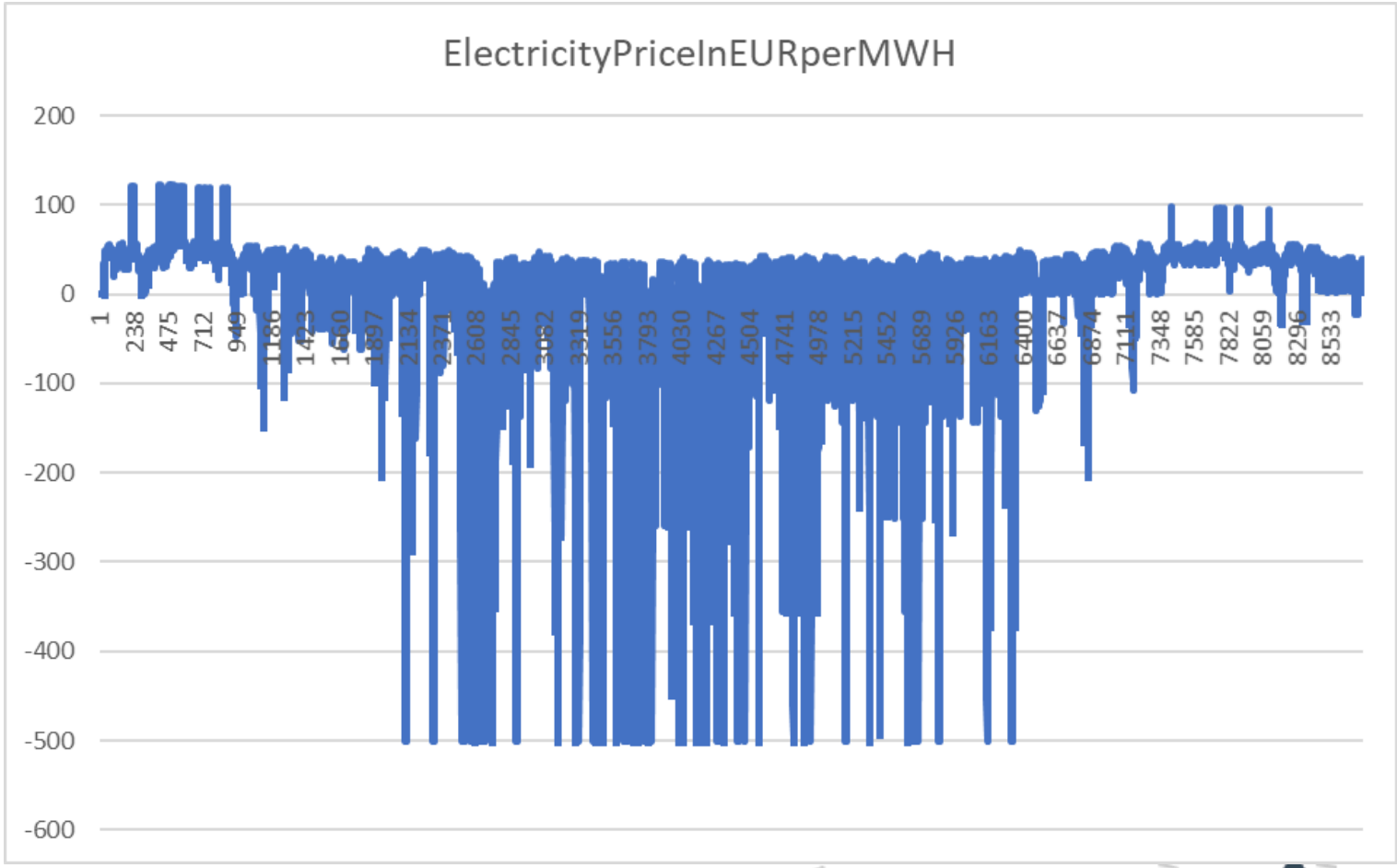
reference

AMIRIS

Parameter Experiments

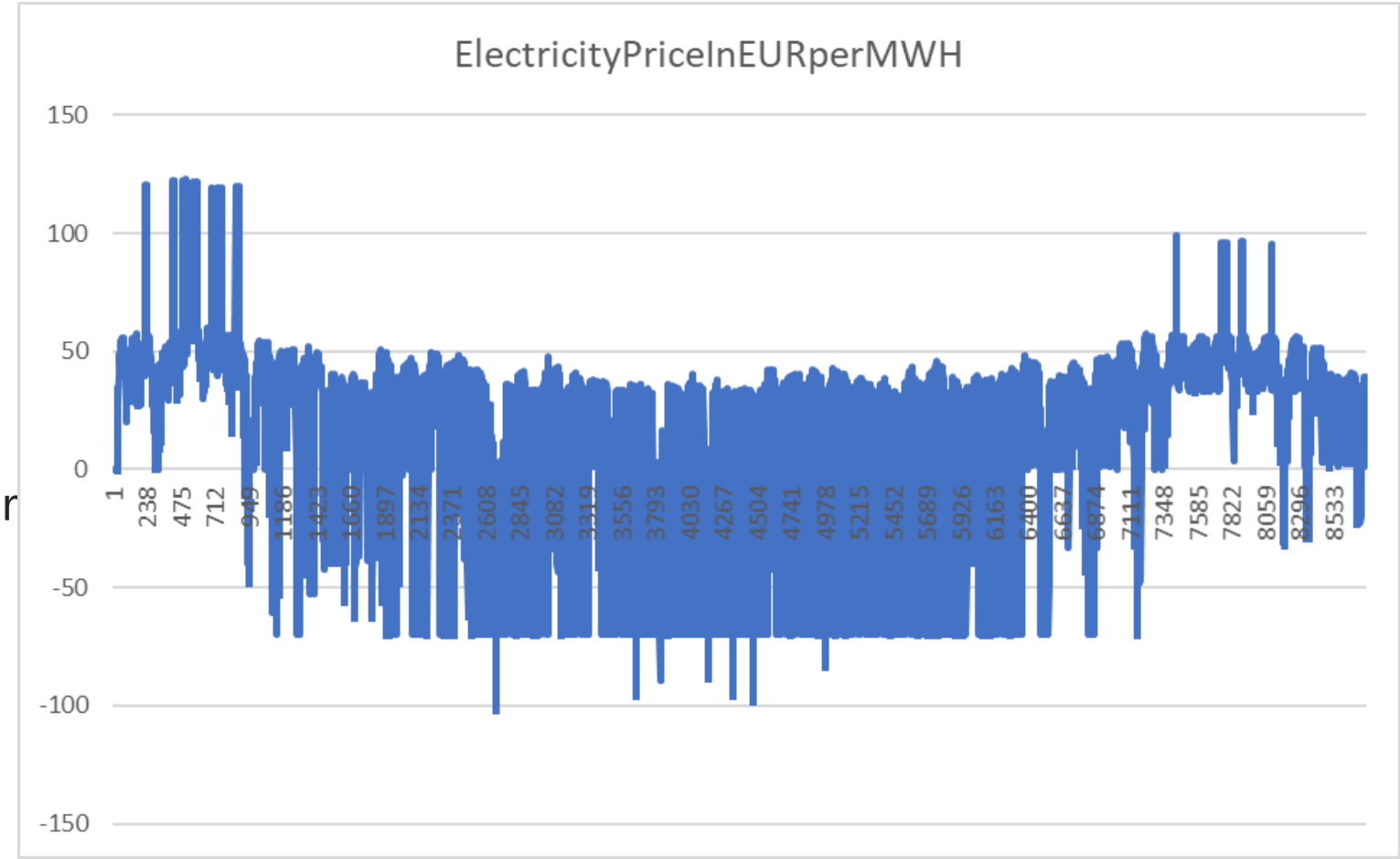
Application
Add PV Agent

1. Copy & rename *scenario* file
2. Add PV *Agent*
3. Choose Unique *ID*
4. Copy & rename *Contracts* folder
5. Redirect Contracts in scenario
6. Add *PV* Contracts
7. Rerun
8. Evaluate Electricity Prices



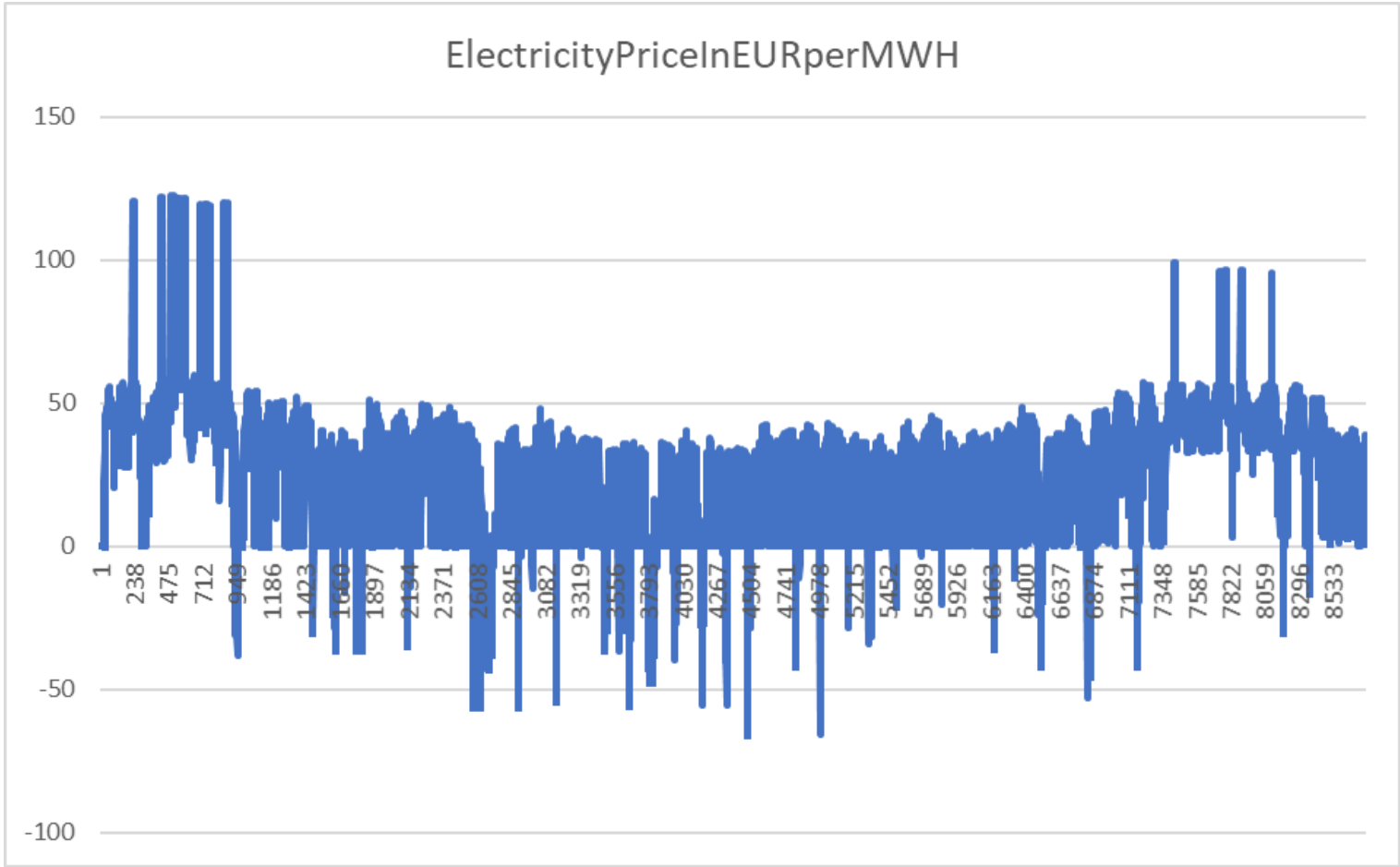
Application
Change Support Instrument: Fixed Premium

1. Copy & rename *scenario* file
2. Update *SupportPolicy*:
    1. Add *Set* "OtherPV"
    2. *MPFIX: Premium*
3. Update *PV Agent*
    1. Change *Set*: OtherPV
    2. Change *Support Instrument*: MPFIX
4. Copy & rename *Contracts* folder
5. Redirect Contracts in scenario
6. Switch PV Agent Contracts
7. Rerun
8. Evaluate Electricity Prices

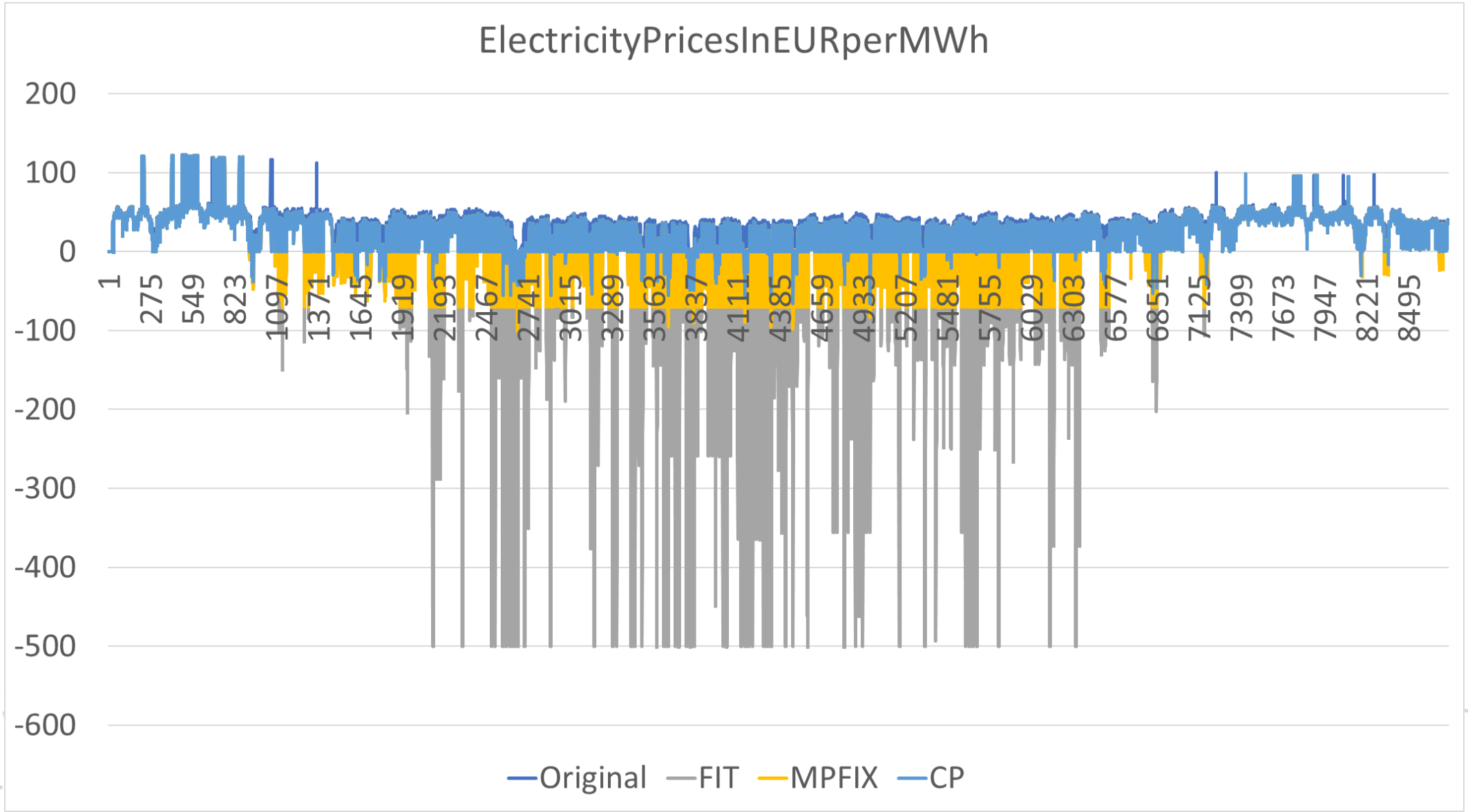


Application
Change Support Instrument: Capacity Premium

1. Copy & rename *scenario* file
2. Update *SupportPolicy*:
    1. In *Set* "OtherPV"
    2. Change to *CP*
3. Update *PV Agent*
    1. Change *Support Instrument*: CP
4. Rerun
5. Evaluate Electricity Prices



Overall experiment comparison



Application
Experiment Summary

- Support Policy had high impact on electricity prices (with limited flexibility)
- Renewables became price setting (bidding at opportunity costs)
- Opportunity cost depended on the employed support instrument

Disclaimer

- This was merely an experiment, not consistent scenario work
- Support paid out at negative prices due to historic regulations

AMIRIS

What do you think?

AMIRIS

Final Remarks

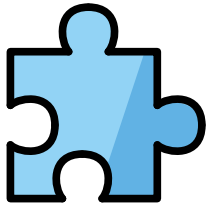
AMIRIS
Following FAIR4RS Principles

Findable

- DOI
- Wikipedia
- COMSES
- HECI
- OEP
- openmod
- Website

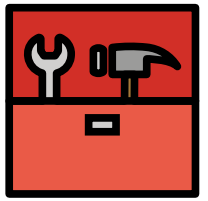
Accessible

- GitLab
- PyPI
- Zenodo



Interoperable

- API
- Workflow tools
- CSV
- YAML



(Re-)usable

- Apache 2.0
- REUSE
- Wiki
- Javadoc
- Win/Mac/Linux
- Scalable (H)PC

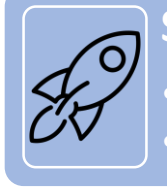
will upload presentation here

Key Indicators

Users
- 12 confirmed external user
- 4 bugs reported

PhD candidates
- 4 internal
- 3 external

Visibility
- 14k views on Wikipedia
- 9k views on openmod



Software
- 39 releases
- 60k downloads

Barker et al. (2022). 10.1038/s41597-022-01710-x

AMIRIS
See also

Website	https://dlr-ve.gitlab.io/esy/amiris/home/
Gitlab project	https://gitlab.com/dlr-ve/esy/amiris/amiris
Open mod forum	https://forum.openmod.org/tag/amiris
Wiki	https://gitlab.com/dlr-ve/esy/amiris/amiris/-/wikis/home
Javadoc	https://dlr-ve.gitlab.io/esy/amiris/amiris/
Zenodo	https://zenodo.org/communities/amiris
E-Mail	amiris@dlr.de

FAME Framework https://gitlab.com/fame-framework

Visit our website

Imprint

Topic:	AMIRIS: Installation, Execution and Market Design Parametrisation
Date:	October 12th 2023
Author:	Christoph Schimeczek
Institution:	Institute of Networked Energy Systems, German Aerospace Center
Images:	DLR (CC BY-NC-ND 3.0)
	TradeRES icon: TradeRES consortium (No license)