Gems: A Framework for Versatile Configuration of Models in Multi-Energy Systems Planning Studies

Dr. Thomas Bittar & Dr. Antoine Oustry, RTE

Energy Innovation Summit, Berlin
2025, June 26th

RTE – France's Transmission System Operator (TSO)

Le réseau
de transport
d'électricité

Europe's largest TSO, in figures:

- $\sim$ 106,000 km of power lines
- €5.6 billion revenue in 2024
- €40 million annual investment in R&D
- $\sim$10,000 employees

R&D Roadmap for Long-Term Power System Adequacy

- **Flexibility modelling**: seasonal storages management under uncertainty, demand-side response, reserves

R&D Roadmap for Long-Term Power System Adequacy

- **Flexibility modelling**: seasonal storages management under uncertainty, demand-side response, reserves
- **Multi-energy**: Integrated modelling & Multi-model workflows

R&D Roadmap for Long-Term Power System Adequacy

- **Flexibility modelling**: seasonal storages management under uncertainty, demand-side response, reserves
- **Multi-energy**: Integrated modelling & Multi-model workflows
- **Continuous performance increase** of *Antares Simulator*

R&D Roadmap for Long-Term Power System Adequacy

- **Flexibility modelling**: seasonal storages management under uncertainty, demand-side response, reserves
- **Multi-energy**: Integrated modelling & Multi-model workflows
- **Continuous performance increase** of *Antares Simulator*
- **Versatile modelling of energy systems: Gems project**

antaressimulator

Gems
Generic Energy Systems
Modelling Scheme

1

Introduction

Motivation: going a step further with Antares Simulator...

Antares Simulator is an open-source industrial tool for (large-scale) planning studies: RTE's official adequacy assessment and network development studies ; European studies (TYNDP).

Motivation: going a step further with Antares Simulator...

Antares Simulator is an open-source industrial tool for (large-scale) planning studies: RTE's official adequacy assessment and network development studies ; European studies (TYNDP).

As energy systems grow more complex and dynamic, we need to go a step further in terms of:

- **Versatile no-code modelling**: easily integrate new models of components without rewriting core code. *Writing and testing new models of energy system components should not require software programming skills!*

Motivation: going a step further with Antares Simulator...

Antares Simulator is an open-source industrial tool for (large-scale) planning studies: RTE's official adequacy assessment and network development studies ; European studies (TYNDP).

As energy systems grow more complex and dynamic, we need to go a step further in terms of:

- **Versatile no-code modelling**: easily integrate new models of components without rewriting core code. *Writing and testing new models of energy system components should not require software programming skills!*
- **Code stability and suitability for open-source**: prevent the simulator core from becoming overloaded with hard-coded logic.

Motivation: going a step further with Antares Simulator...

Antares Simulator is an open-source industrial tool for (large-scale) planning studies: RTE's official adequacy assessment and network development studies ; European studies (TYNDP).

As energy systems grow more complex and dynamic, we need to go a step further in terms of:

- **Versatile no-code modelling**: easily integrate new models of components without rewriting core code. *Writing and testing new models of energy system components should not require software programming skills!*
- **Code stability and suitability for open-source**: prevent the simulator core from becoming overloaded with hard-coded logic.
- **Interoperability**: interact seamlessly with external tools or formats.

... and with the **open-source** framework Gems

Gems: **G**eneric **E**nergy systems **M**odeling **S**cheme

- **A high-level modelling language** to configure and extend models of components.
- **A data structure** to represent energy systems.

Gems Interpreters

- **Development of a Gems interpreter inside *Antares Simulator*.**
- **GemsPy: a Python interpreter for Gems** to perform simulation & optimisation.

Gems
Generic Energy Systems
Modelling Scheme

gemspy.readthedocs.io

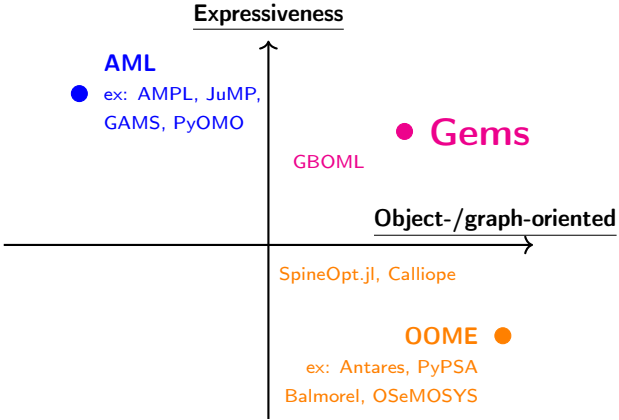
github.com/
AntaresSimulatorTeam/GemsPy

2

Presentation of the framework

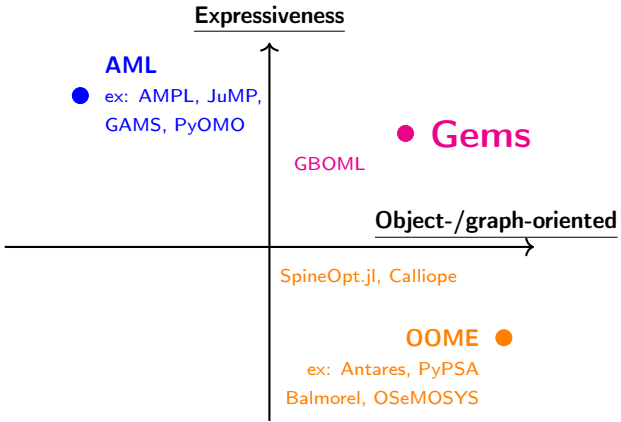
Where Does Gems Fit Among Existing Modelling Tools?

- AML: Algebraic Modelling Language
- OOME: Object-Oriented Modelling Environments



Where Does Gems Fit Among Existing Modelling Tools?

- AML: Algebraic Modelling Language
- OOME: Object-Oriented Modelling Environments

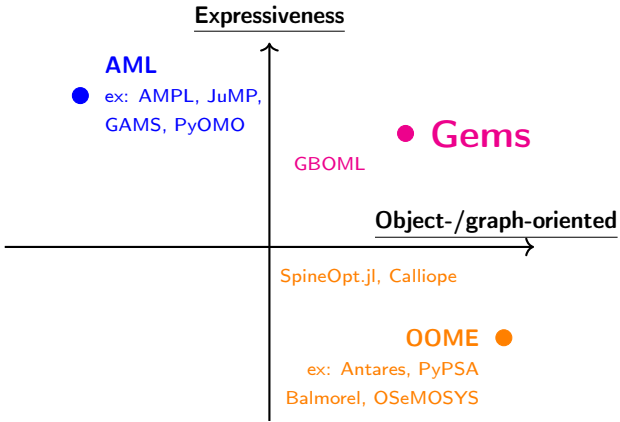


Gems: Mathematical setting

- Mixed Integer Linear Programming
- Optimisation problems over a time $\times$ scenario grid
- Deterministic or two-stage stochastic

Where Does Gems Fit Among Existing Modelling Tools?

- AML: Algebraic Modelling Language
- OOME: Object-Oriented Modelling Environments



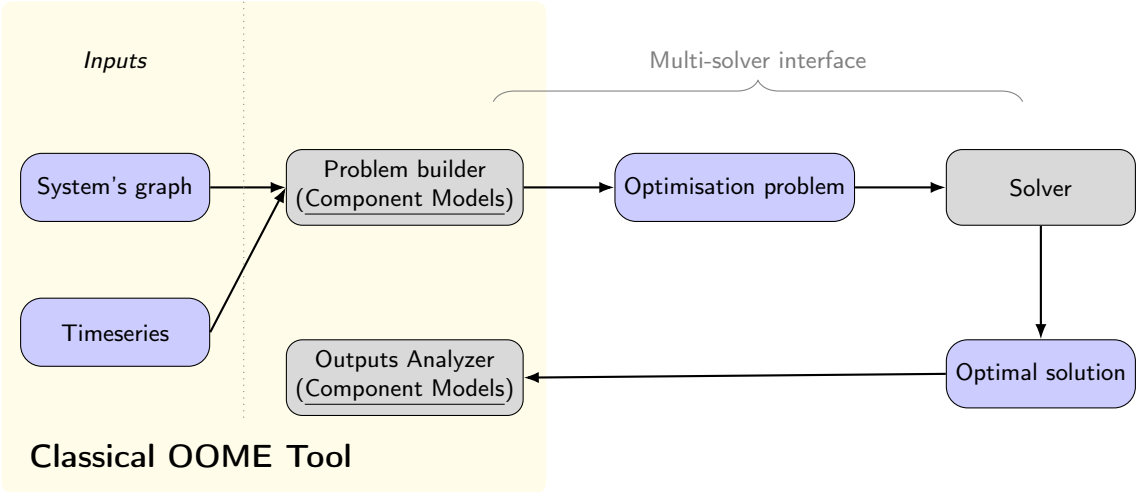
Gems: Mathematical setting

- Mixed Integer Linear Programming
- Optimisation problems over a time $\times$ scenario grid
- Deterministic or two-stage stochastic

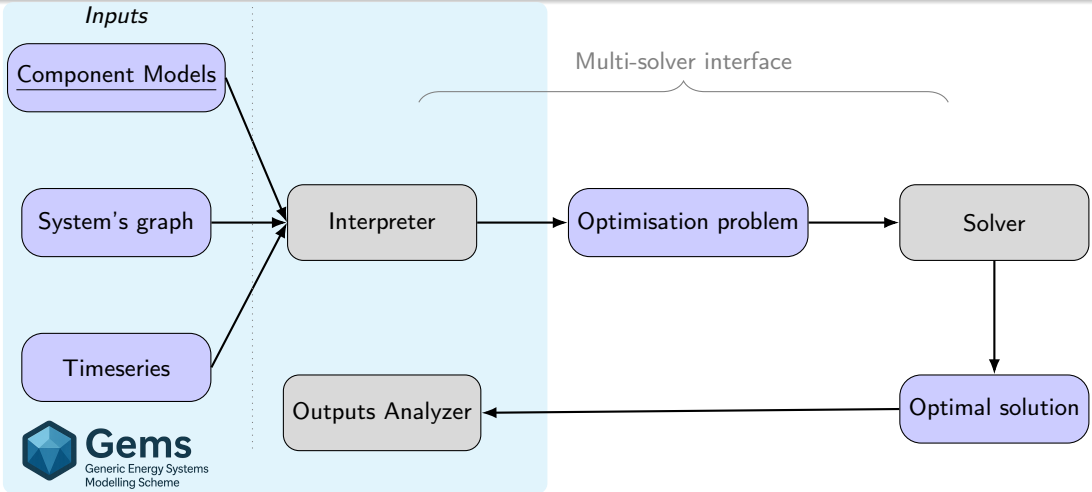
Gems: Main targeted applications

- Adequacy assessment
- Dispatch/production cost simulation
- Capacity expansion planning

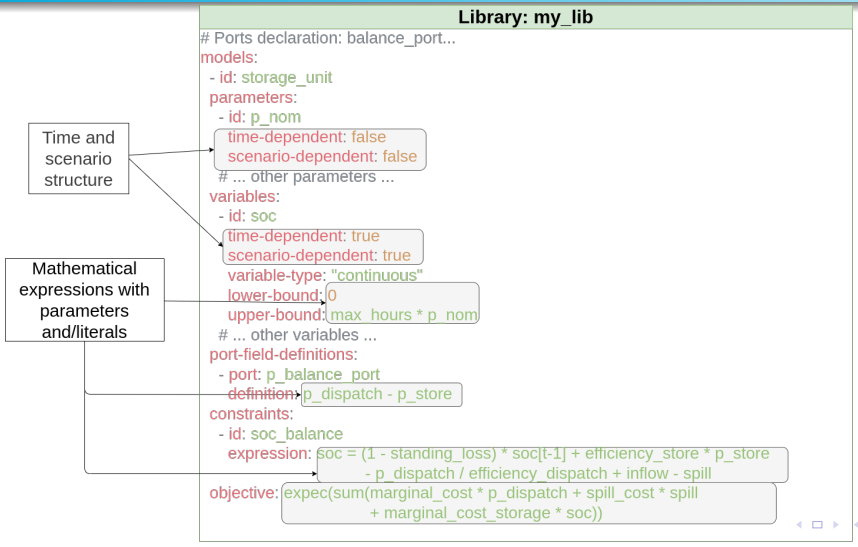
Classical OOME Architecture



Gems Architecture: Models go out of the software's code.

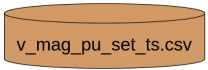


Component models: language and file structure



System's graph and timeseries data

Timeseries

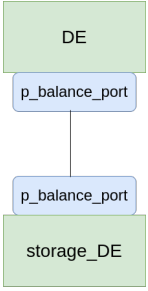


System file

```
system:
  components:
  - id: storage_DE
    model: my_lib.storage_unit
    parameters:
    - id: p_nom
      value: 7526.22
    # other parameters value...
  - id: DE
    model: my_lib.bus
    parameters:
    - id: v_mag_pu_set
      value: v_mag_pu_set_ts
    # other parameters value...

  connections:
  - component1: DE
    port1: p_balance_port
  - component2: storage_DE
    port2: p_balance_port
```

System visualization



Reference libraries of models (under construction)

Principle: build and share ready-to-use model libraries. May be used separately or together.

Antares (legacy)

`antares_models.yml`

- Area
- Thermal cluster
- Load
- Hydro Storage
- Short-term Storage
- [...]

Reference libraries of models (under construction)

Principle: build and share ready-to-use model libraries. May be used separately or together.

Antares (legacy)
`antares_models.yml`

- Area
- Thermal cluster
- Load
- Hydro Storage
- Short-term Storage
- [...]

PyPSA
`pypsa_models.yml`

- Bus
- Load
- Generator
- Storage Unit
- Link
- [...]

Reference libraries of models (under construction)

Principle: build and share ready-to-use model libraries. May be used separately or together.

Antares (legacy)
`antares_models.yml`
- Area
- Thermal cluster
- Load
- Hydro Storage
- Short-term Storage
- [...]

PyPSA
`pypsa_models.yml`
- Bus
- Load
- Generator
- Storage Unit
- Link
- [...]

Tool X
`toolx_models.yml`
- supply
- demand
- storage
- conversion
- transmission
- [...]

R&D Project H2
`h2_models.yml`
- H_2 Area
- Demand
- Pipe
- Electrolyzer
- SMR
- [...]

3

Illustration

Two examples illustrating the benefits of Gems

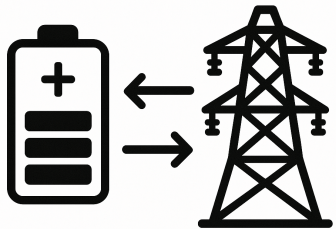
1. **Versatility**: "No-code" Model Adaptation

2. **Interoperability**: Loading and simulating a PyPSA study with Gems

1. Example of "No-code" Model Adaptation: Storage Units

Context: Fictitious simultaneous charging and discharging in simulations

- Unlike most real-world storage (e.g. batteries), classical storage models (e.g. Antares and PyPSA) allow for simultaneous charging and discharging.
- Whenever there is energy surplus, it is then "optimal" to use this mode to dissipates surplus renewable energy. **This is an issue to model energy surplus evacuation (and negative prices).**



Storage model adaptation with Gems:

a. Classical storage model: description and simulation

b. Model adaptation and simulation, without any dev.

1. Example of "No-code" Model Adaptation: Storage Units

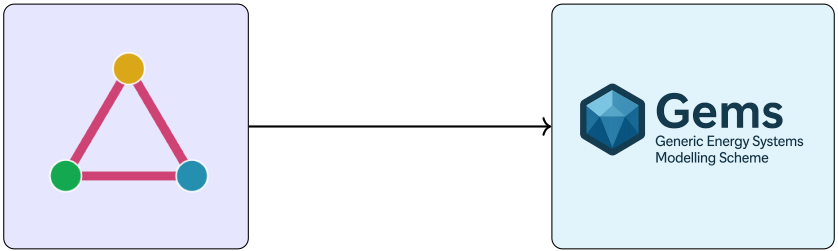
a. Base case: Linear model

```
library:
  models:
    - id: short-term-storage
      parameters: …
      variables:
        - id: p_injection
          lower-bound: 0
          upper-bound: injection_nominal_capacity
        - id: p_withdrawal …
        - id: level …
      ports: …
      port-field-definitions: …
      constraints:
        - id: initial_level
          expression: level[0] = initial_level
        - id: Level equation
          expression: level[t+1] = level +
            efficiency_injection * p_injection
            - efficiency_withdrawal * p_withdrawal + inflows
```

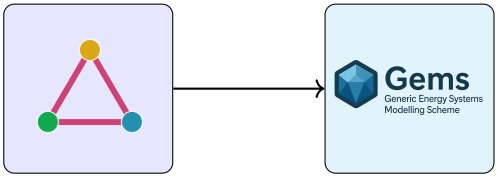
b. Adapted case MILP model

```
library:
  models:
    - id: short-term-storage
      parameters: …
      variables:
        - id: p_injection
          lower-bound: 0
          upper-bound: injection_nominal_capacity
        - id: p_withdrawal …
        - id: level …
        - id: is_charging      You, 16 hours ago • Update demo
          lower-bound: 0
          upper-bound: 1
          variable-type: integer
      ports: …
      port-field-definitions: …
      constraints:
        - id: initial_level
          expression: level[0] = initial_level
        - id: Level equation
          expression: level[t+1] = level +
            efficiency_injection * p_injection
            - efficiency_withdrawal * p_withdrawal + inflows
        - id: is_charging_cst
          expression: p_injection
            <= is_charging * injection_nominal_capacity
        - id: is_withdrawing_cst
          expression: p_withdrawal <= (1 - is_charging) *
          injection_nominal_capacity
```

2. Loading and simulating a PyPSA study with Gems



2. PyPSA to Gems converter data converter: state of progress



Validation tests (numerical comparison):

- ✓ Small PyPSA examples (CI tests)
- ✓ Medium-size PyPSA-Eur test cases
- ✎ Large-scale PyPSA-Eur test cases

PyPSA component support status

PyPSA Component	State of Progress
Bus	✓
Load	✓
Generator	✓
Storage Unit	✓
Store	✓
Link	✓
Global Constraint	✓
Line	✎
Transformer	✎
Shunt Impedance	✎

4

Roadmap and Conclusion

Roadmap for Gems and its integration in Antares Simulator

Gems
Generic Energy Systems Modelling Scheme

2023
Start of conception

2024
POC for Gems interpreter

2025
`GemsPy` is out!
+ Interoperability demonstration

Functionality extensions

2025
Gems interpreter in Antares Simulator (incl. hybrid mode)

2026
Integration in Antares Web (interface)

antaressimulator

Key Takeaways

The open-source framework Gems, in a nutshell

- A high-level language that takes the mathematical models out of the code! Self-contained input data.
- Generic and stable interpreters (`GemsPy` and soon in Antares Simulator), with no implicit modelling choices.
- Favors versatile advanced modelling and interoperability.

`gemspy.readthedocs.io`

`github.com/AntaresSimulatorTeam/GemsPy`

Key Takeaways

`gemspy.readthedocs.io`

`github.com/AntaresSimulatorTeam/GemsPy`

The **open-source** framework Gems, in a nutshell

- A high-level language that takes the mathematical models out of the code! Self-contained input data.
- Generic and stable interpreters (`GemsPy` and soon in Antares Simulator), with no implicit modelling choices.
- Favors versatile advanced modelling and interoperability.

We're open to **collaboration** around Gems!

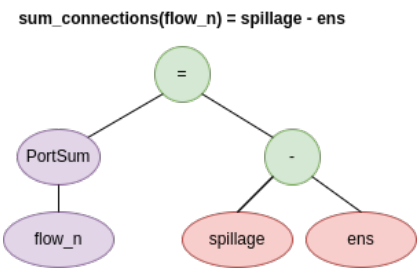
- Gems functionality extensions, advanced performance
- Interoperability with your favorite tool/chain of tools
- Structuration of reference libraries of models
- Graphical user interface

Thank you! Any question?

Appendix

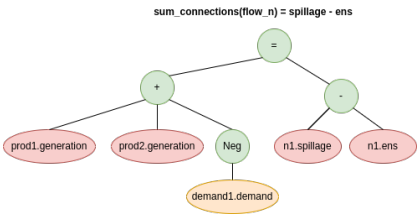
The Gems Python interpreter, in a nutshell

- **Input:** Libraries + System description + Data + Optimisation context
- Representation of expressions as **Abstract Syntax Trees** (ASTs)
- **Visit the ASTs** to perform operations on expressions



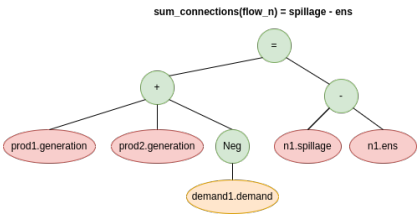
The Gems Python interpreter, in a nutshell

- **Input:** Libraries + System description + Data + Optimisation context
- Representation of expressions as **Abstract Syntax Trees** (ASTs)
- **Visit the ASTs** to perform operations on expressions
  - **Substitution of ports** by their expressions (Libraries + System description)



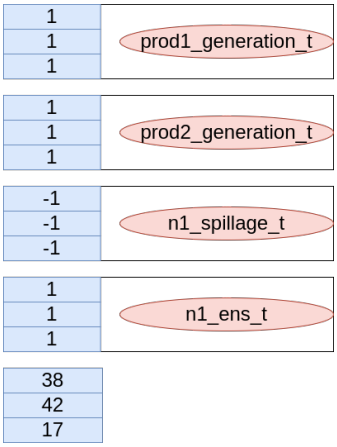
The Gems Python interpreter, in a nutshell

- **Input:** Libraries + System description + Data + Optimisation context
- Representation of expressions as **Abstract Syntax Trees** (ASTs)
- **Visit the ASTs** to perform operations on expressions
  - **Substitution of ports** by their expressions (Libraries + System description)
  - **Automatic deduction** of number of variables/constraints to instantiate (Using time/scenario structure + Optimisation context)



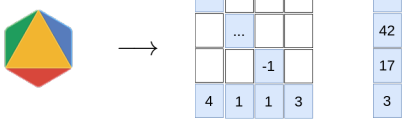
The Gems Python interpreter, in a nutshell

- **Input:** Libraries + System description + Data + Optimisation context
- Representation of expressions as **Abstract Syntax Trees** (ASTs)
- **Visit the ASTs** to perform operations on expressions
  - **Substitution of ports** by their expressions (Libraries + System description)
  - **Automatic deduction** of number of variables/constraints to instantiate (Using time/scenario structure + Optimisation context)
  - **Substitute parameters expressions** (in constraints coefficients) with numerical data



The Gems Python interpreter, in a nutshell

- **Input:** Libraries + System description + Data + Optimisation context
- Representation of expressions as **Abstract Syntax Trees** (ASTs)
- **Visit the ASTs** to perform operations on expressions
  - **Substitution of ports** by their expressions (Libraries + System description)
  - **Automatic deduction** of number of variables/constraints to instantiate (Using time/scenario structure + Optimisation context)
  - **Substitute parameters expressions** (in constraints coefficients) with numerical data
- **Build** the optimisation problem



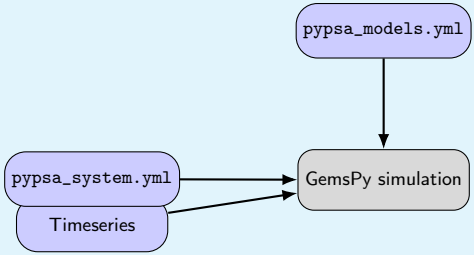
Loading and simulating a PyPSA study with Gems

Gems

pypsa_models.yml

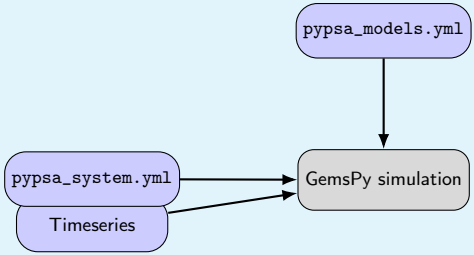
Loading and simulating a PyPSA study with Gems

Gems

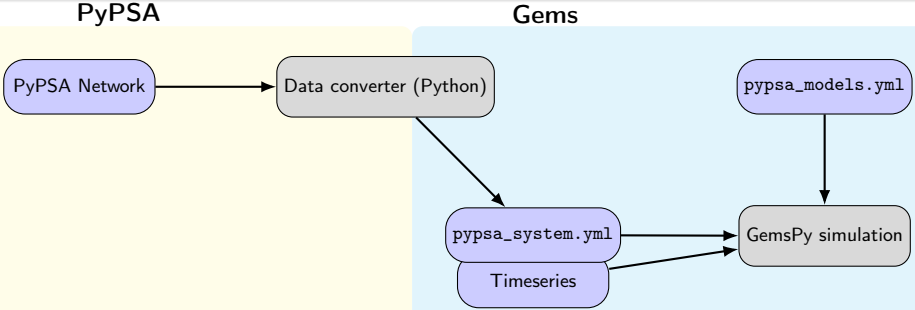


Loading and simulating a PyPSA study with Gems

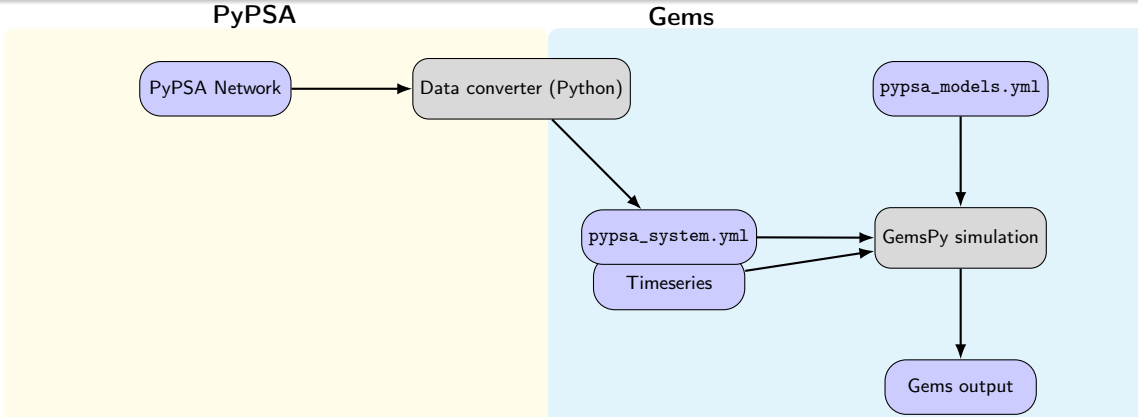
Gems



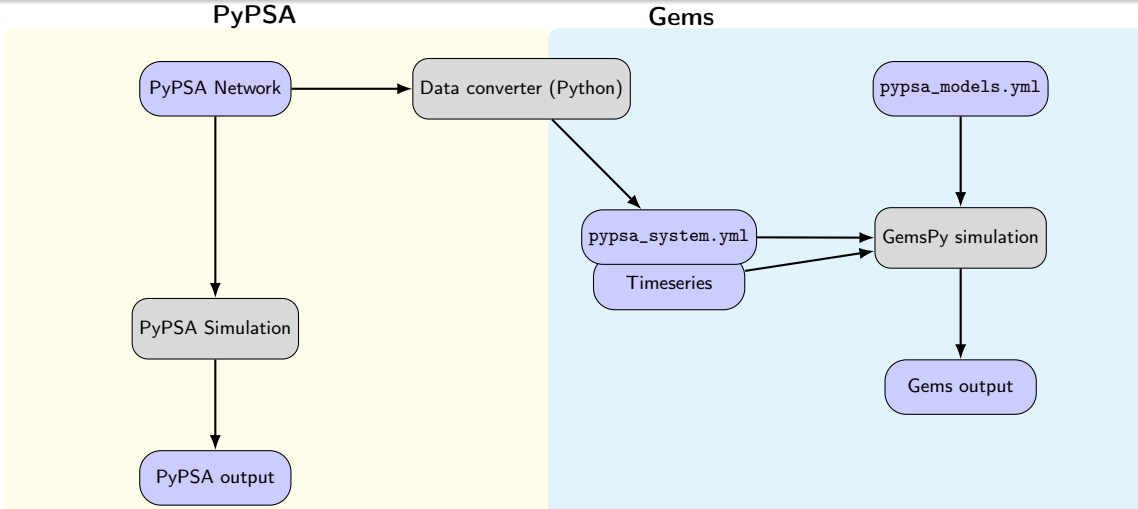
Loading and simulating a PyPSA study with Gems



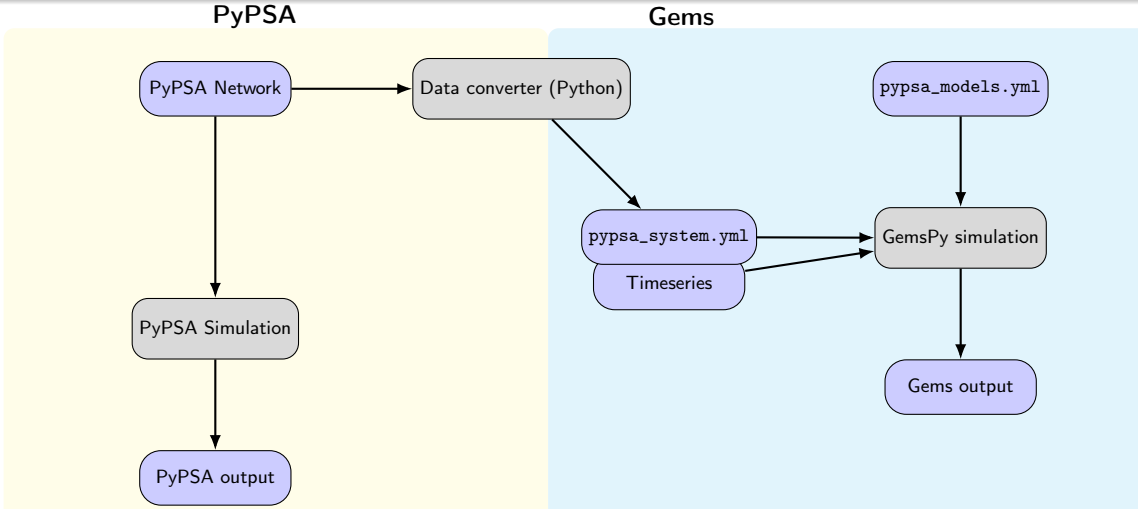
Loading and simulating a PyPSA study with Gems



Loading and simulating a PyPSA study with Gems



Loading and simulating a PyPSA study with Gems



Loading and simulating a PyPSA study with Gems