Linux Foundation Open-Source Climate (OS-C)

Transition Analysis Technical Overview

January 5, 2023

Transition Analysis: Typical questions that could be answered

Is it possible to achieve a below 2 degree world working on decarbonisation right now?

What could be the technological choices and investments necessary to achieve a 2 degree world?

What would be the economic conditions and energy mix in a 2 degree world?

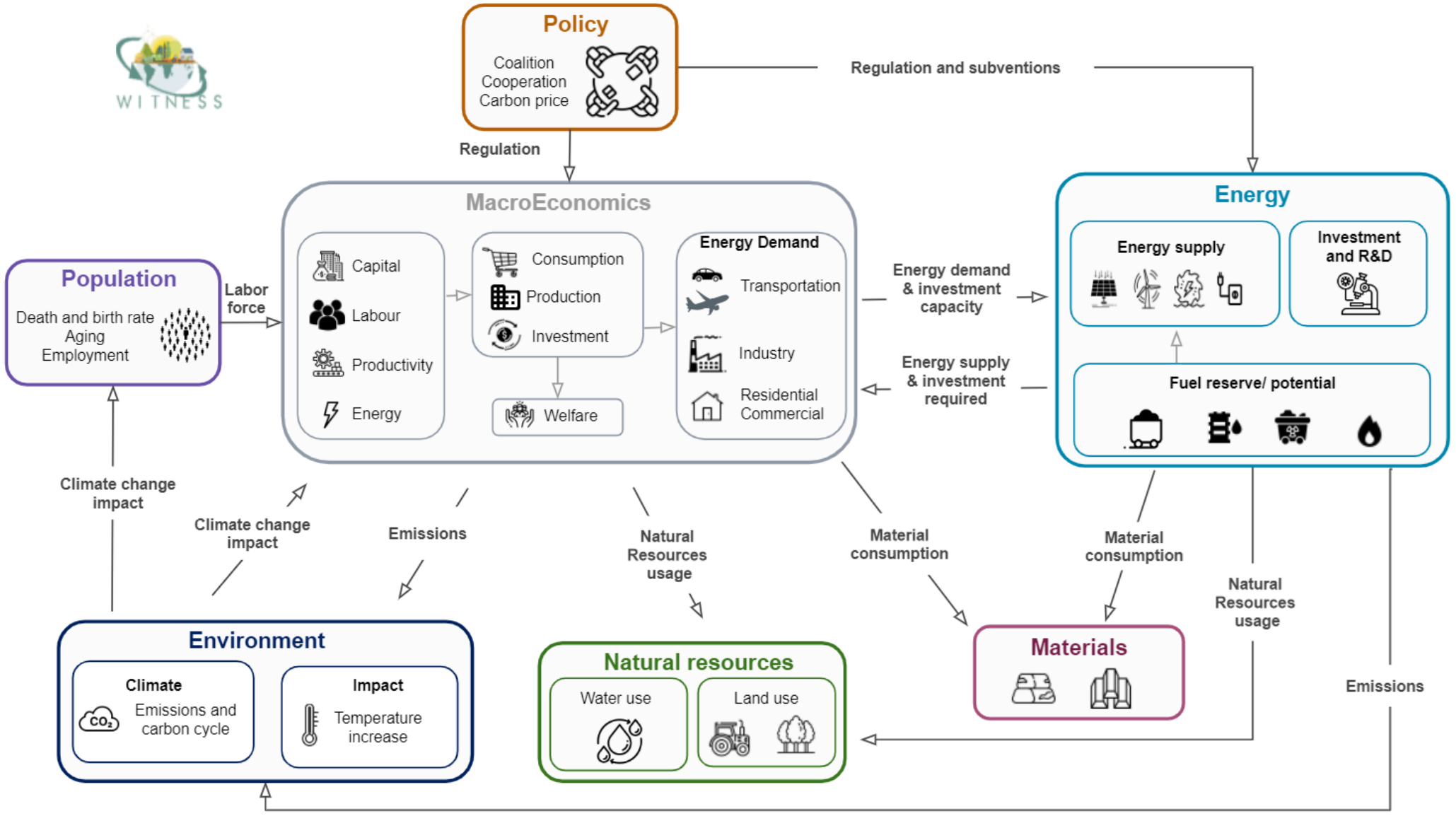
What is the effect of technological effectiveness on the speed of energy transition?

How to split investments in the different energy technologies to achieve a low carbon world?

What would happen to the economy and human population if the world temperature goes much above 2 degree?

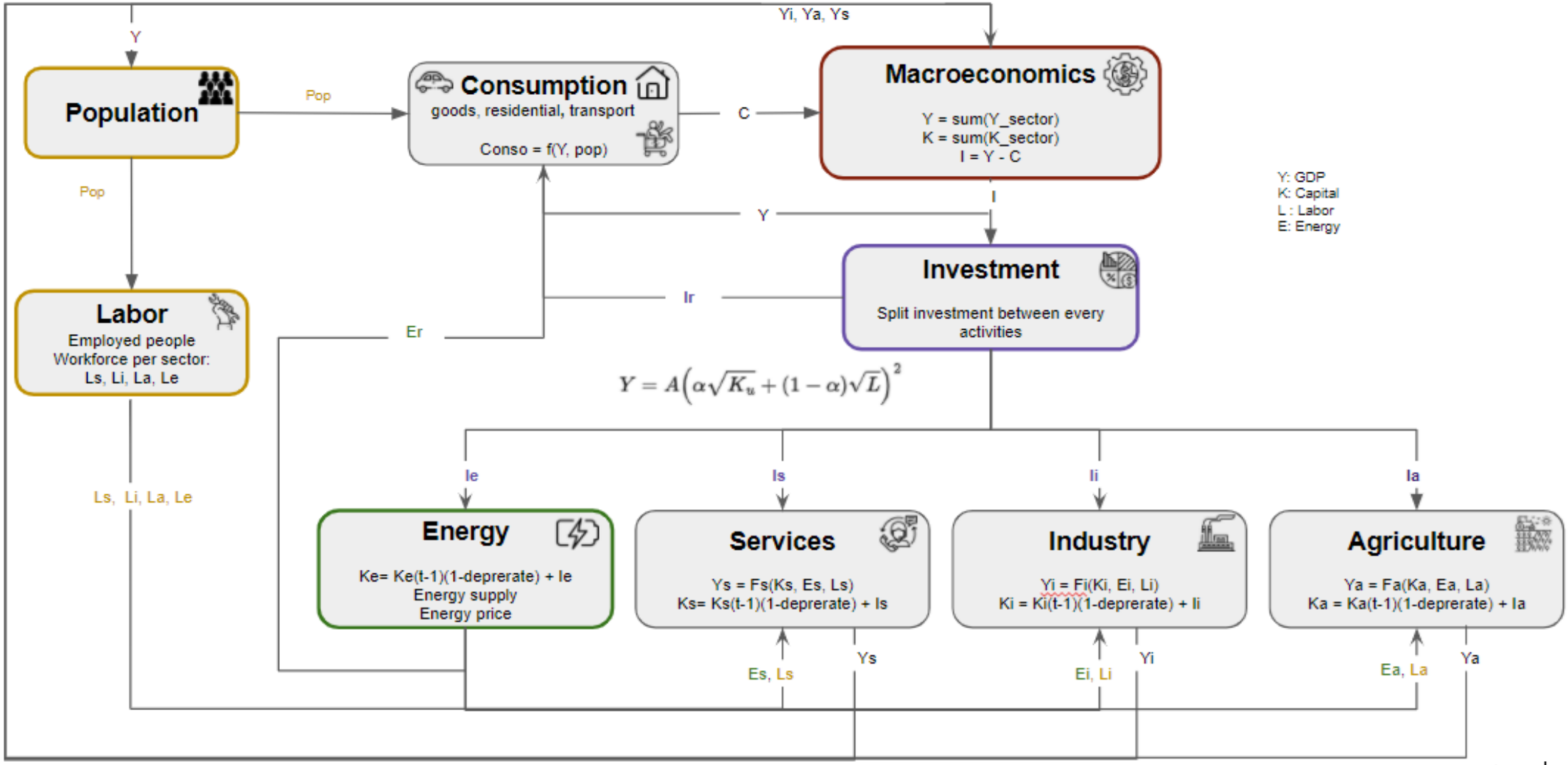
OS-C

Transition Analysis: WITNESS model



OS-C

Sectorization v0 Principles



* transport in services

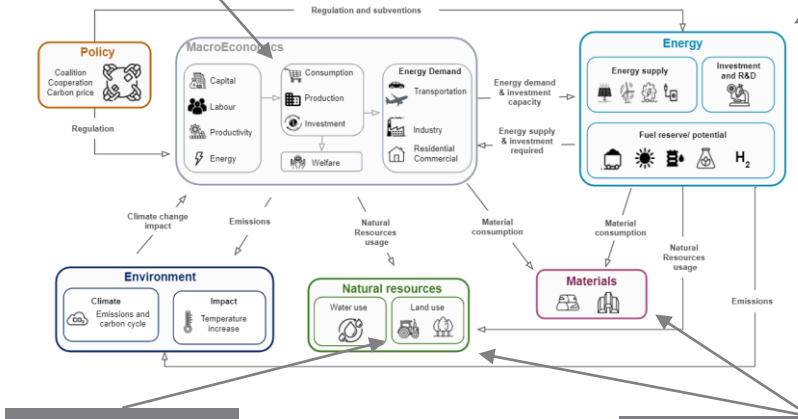
OS-C

WITNESS Roadmap Detailed Concepts

Global WITNESS

Begin sectorization of economy

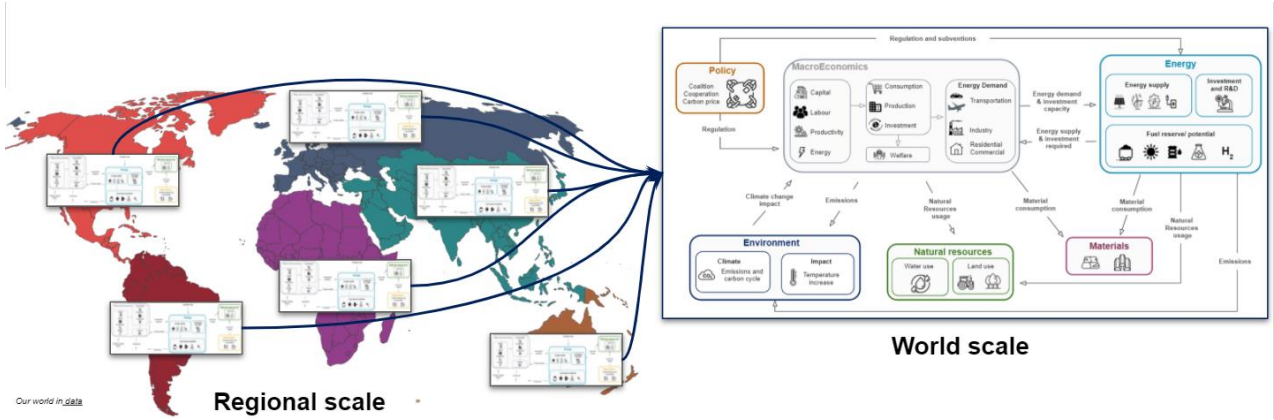
- Add missing technos
- Calibration
- Post processing



Improve land use model

Implement missing models for ressources: water, materials…

Regional WITNESS



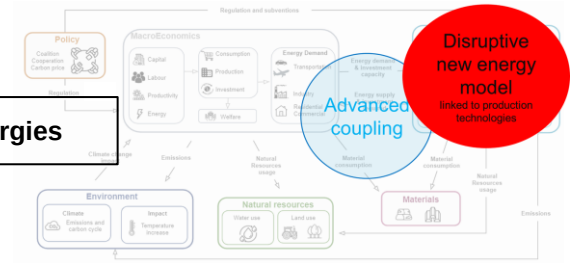
- Regional database

Step by step regionalisation:

- Regionalisation of models (energy, economics, resources, land use…)
- Exchanges between regions
- Problem formulation

OS-C

Zooming on Energy Model



40+ technologies

10+ Energies

Design variables in blue

Techno Prices
Techno Production/consumption
Techno CO2 emissions
Natural resource usage

Invest per techno (M$)

Electrolysis

WaterGasShift

Hydrogen mix

Invest Mix for each techno (%)

For a given investment plan

Macroeconomics

Invest (M$)

Energy mix Investments distribution

Hydrogen Prices
Hydrogen Production/consumption
Hydrogen CO2 emissions

Multi-Disciplinary Analysis (MDA)

For each techno

Techno Prices
Techno Production/consumption
Techno CO2 emissions
Natural resource usage

Each energy mix

Total Energy Net production (TWh)
Final Energy prices ($/MWh)

Energy CO2 emissions (kg/MWh)
Energy prices ($/MWh)

Overall energy mix

MacroEconomics

Material/Natural resources usage

Total CO2 emissions (Gt)

Ressources

Environment

- ➔ The energy model is a MDA
- ➔ Total production and CO2 emissions are computed following a given investment plan
- ➔ Electrolysis will be net zero emissions only if electricity mix is zero emissions

OS-C

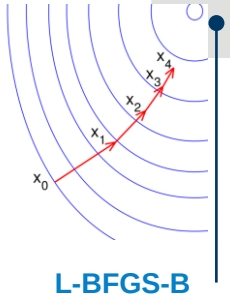
WITNESS Optimization Example

Objective	*maximize* **welfare** and *minimize* **CO2 emissions**
Design Variables	**technology investment mixes** *(from 2020 to 2100)*
Constraints	*(from 2020 to 2100)*: total energy production > energy lower bound net energies production > energies demand liquid fuel + H2 prod + H2 liquid production > % total production solid fuel + electricity + biomass production > % total production hydropower production < hydropower production in 2020 H2 liquid production > %H2 total production available land > land demand (for forest, agriculture,…)

Adjoint based gradient computation

1 function evaluation
1 adjoint system
instead of
241 functions evaluations
per iteration

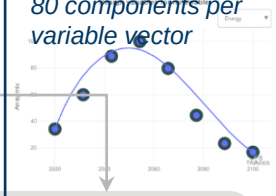
Optimization solver



L-BFGS-B

B-splines

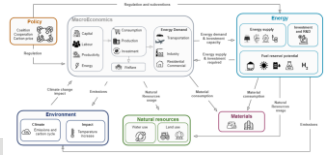
8 poles per variable vector
instead of
80 components per variable vector



DesignVar

MDA Analysis

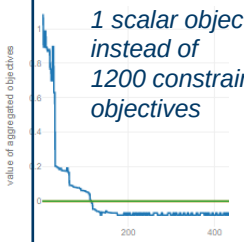
NewtonRaphson solver
~30 iterations
~8 minutes



WITNESS model

Lagrangian objective formulation

1 scalar objective
instead of
1200 constraints and 160 objectives



FuncManager

key numbers

MDO
65 disciplines
4240 design variables
265383 variables
1200 constraints

MDA
63 disciplines
25064 coupling variables
262715 variables

MDO
65 disciplines
424 design variables
265383 variables
0 constraint

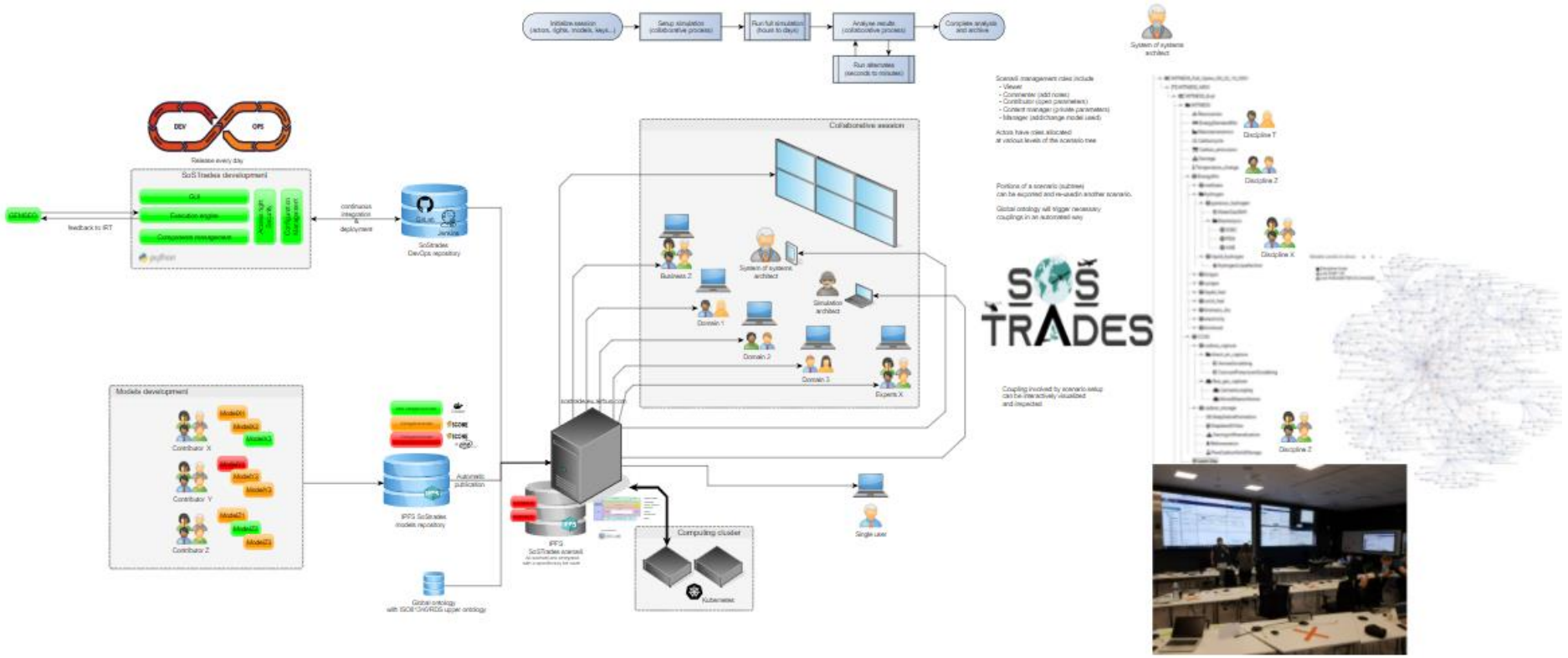
MDA
63 disciplines
25064 coupling variables
262715 variables

solved in ~10 hours

OS-C

SoSTrades Collaborative Solution Overview

python™

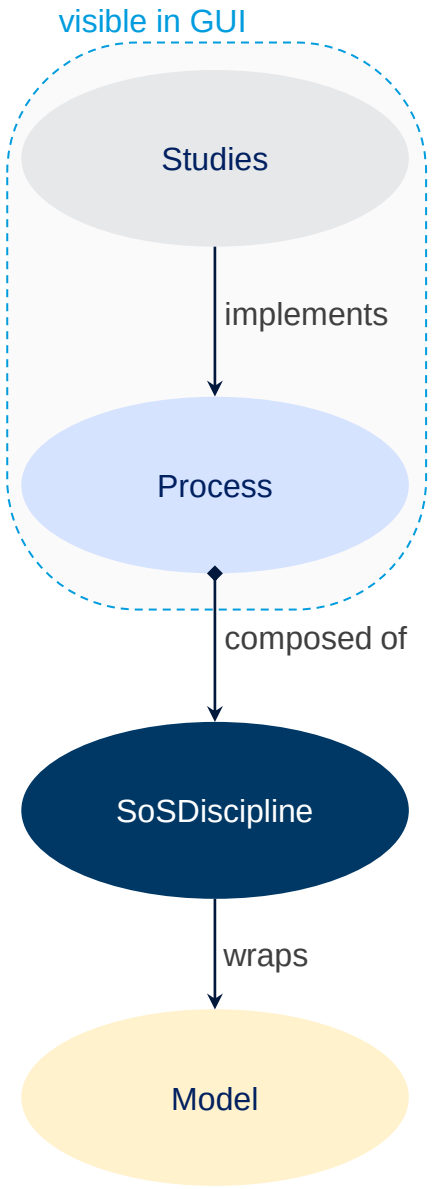


based on: GEMSEO A Generic Engine for Multidisciplinary Scenarios, Exploration and Optimization

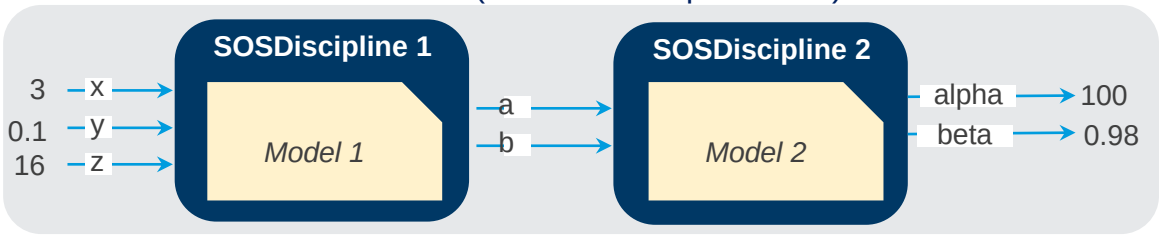
https://gemseo.readthedocs.io/en/stable/

OS-C

Wrapping | Key SoSTrades Objects

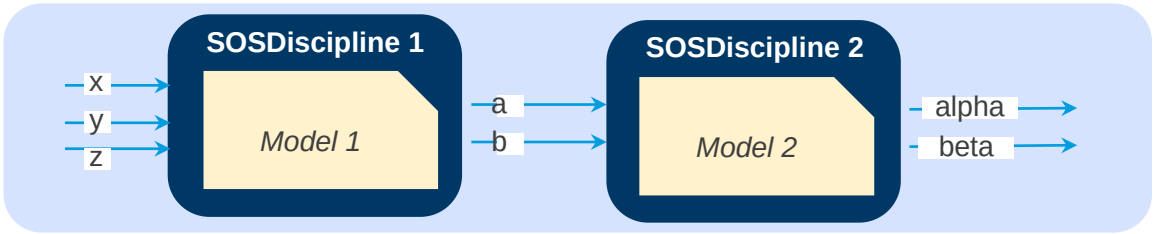


Studies (or instance of processes)



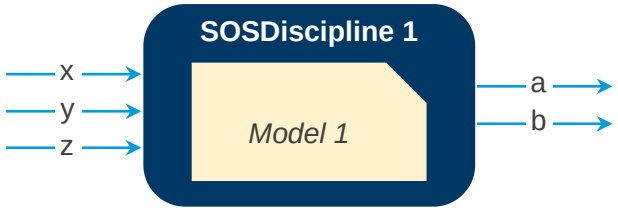
A Study is a **process instantiated with given input data,** and **output data** if the study has run

Process



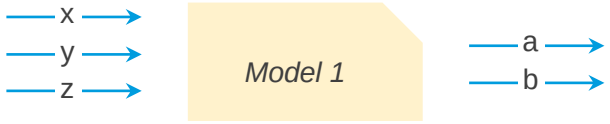
A **Process** is used to **couple** multiple disciplines together inside **SoSTrades**

SoSDiscipline (model wrapper)



A **SoSDiscipline** is used to **wrap** a model inside **SoSTrades**

Model



A user model is available

Wrapping | Make your first SoSTrades wrapper of model

I/O variables definition

User model call

Post-processing definition

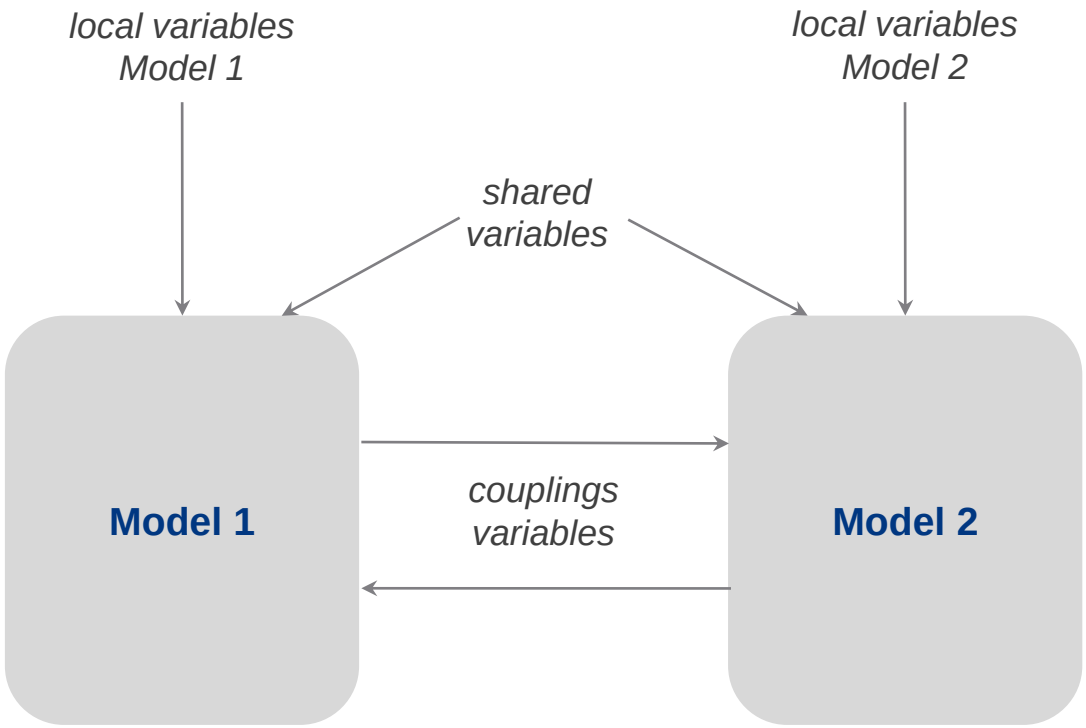
```python
class Disc1(SoSDiscipline):
    _maturity = 'Fake'

    DESC_IN = {'a': {'type': 'float', 'visibility': 'Shared', 'namespace': 'ns_a'},
               'b': {'type': 'int'}}

    DESC_OUT = {'x': {'type': 'float',
                      'visibility': 'Shared', 'namespace': 'ns_x'}}
```

```python
    def run(self):
        a, b = self.get_sosdisc_inputs(['a', 'b'])
        # call models
        model1 = Model1(a, b)
        x = model1.compute()
        dict_values = {'x': x}
        # put new field value in data_out
        self.store_sos_outputs_values(dict_values)
```

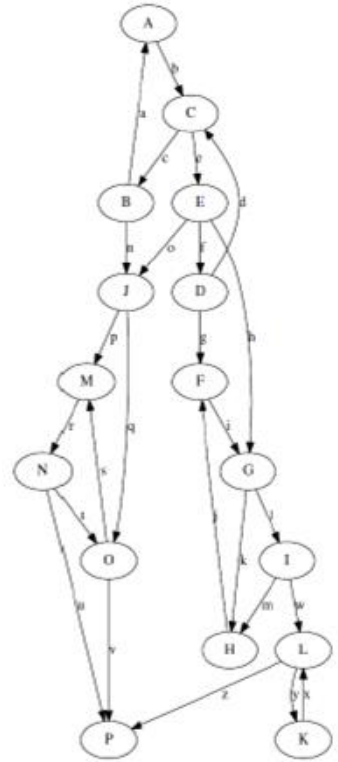
```python
    def get_chart_filter_list(self):

    def get_post_processing_list(self, filters=None):

        instanciated_charts = []

        # Overload default value with chart filter
        if filters is not None:
            for chart_filter in filters:
                if chart_filter.filter_key == 'graphs':
                    charts_list = chart_filter.selected_values

        if 'chart 1' in charts_list:

            chart_name = 'chart 1'

            output1 = self.get_sosdisc_outputs('x')
            input1 = self.get_sosdisc_inputs('a')
            new_chart = TwoAxesInstanciatedChart('input1', 'output1',
                                                 chart_name=chart_name)
            serie = InstanciatedSeries(
                [input1], [output1], '', 'bar')
```

OS-C

Coupling | Multi-Disciplinary Analysis (MDA)



According to the discipline I/O names, ***couplings variables are automatically identified*** *and* ***multi-disciplinary analyses automatically built****.*



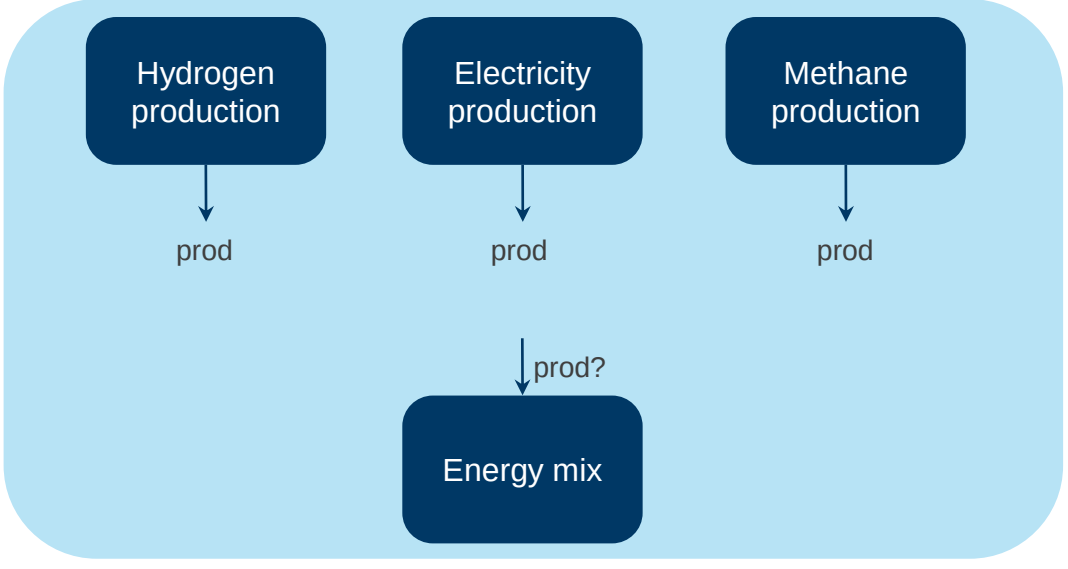
GEMSEO

A Generic Engine for Multidisciplinary Scenarios,
Exploration and Optimization

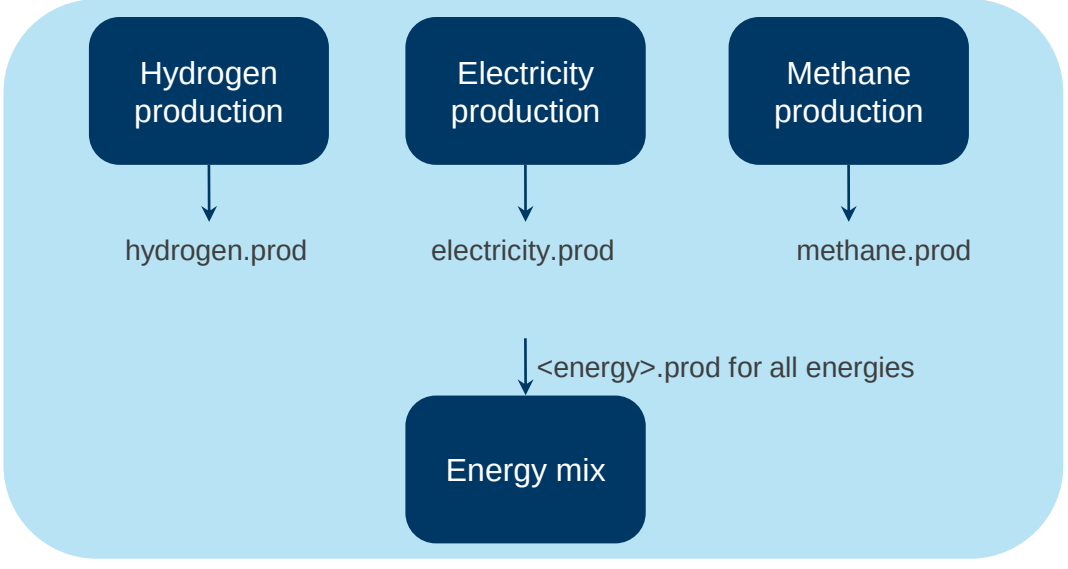
Coupling | Introduction to Namespaces

*In order to be able to **distinguish different i/o with the same names** and **avoid undesired couplings**, the **namespace definition** has been introduced*

Example without namespaces definition



Example with namespaces definition



Namespace declaration in discipline i/o grammars

```
class Disc1(SoSDiscipline):
    _maturity = 'Fake'

    DESC_IN = {'a': {'type': 'float', 'visibility': 'Shared', 'namespace': 'ns_a'},
               'b': {'type': 'int'}}

    DESC_OUT = {'x': {'type': 'float',
                      'visibility': 'Shared', 'namespace': 'ns_x'}}
```