

Input-Output Analysis meets Python: a MARIO crash course

by Nicolò Golinucci PhD



POLITECNICO
MILANO 1863



eNextGen

spin off[®]
POLITECNICO DI MILANO

26/09/2024

Who we are



Lorenzo Rinaldi PhD, Consultant and Head of R&D at eNextGen

MARIO developer and post-doc at SESAM group (Department of Energy – PoliMi)



Mohammad Amin Tahavori, Head of Software at eNextGen

Main MARIO developer and currently climate and energy modeller at VITO



Nicolò Golinucci PhD, Consultant and Chief Executive Officer at eNextGen



eNextGen calculates all your company's emissions, with a focus on indirect value chain emissions (Scope 3), for effective carbon footprint reduction



Prof. Emanuela Colombo,
Prof. Matteo Vincenzo Rocco

Scientific Advisory Board



Building on our research and SESAM group contributions

The idea of building a historical memory and useful tool for taking advantage of what we learn as a group



Applied Energy

Volume 182, 15 November 2016, Pages 590-601



Energy

Volume 170, 1 March 2019, Pages 1249-1260



Journal of Cleaner Production

Volume 139, 15 December 2016, Pages 1449-1462



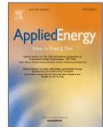
Journal of Cleaner Production

Volume 263, 1 August 2020, 121278



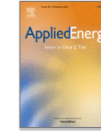
Applied Energy

Volume 194, 15 May 2017, Pages 832-844



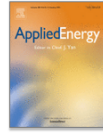
Applied Energy

Volume 274, 15 September 2020, 115300



Applied Energy

Volume 211, 1 February 2018, Pages 590-603



Applied Energy

Volume 274, 15 September 2020, 115301

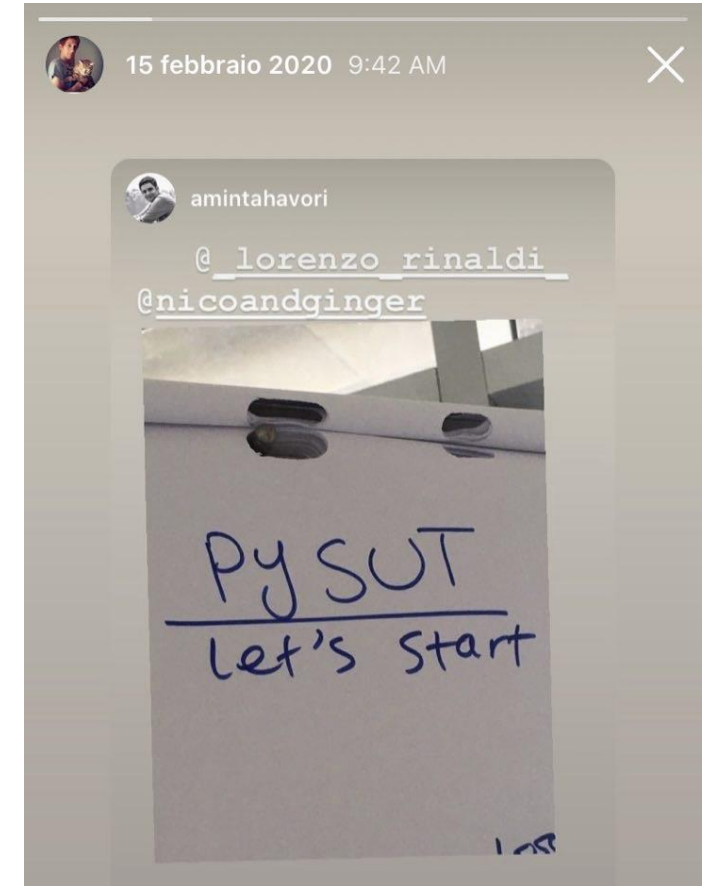


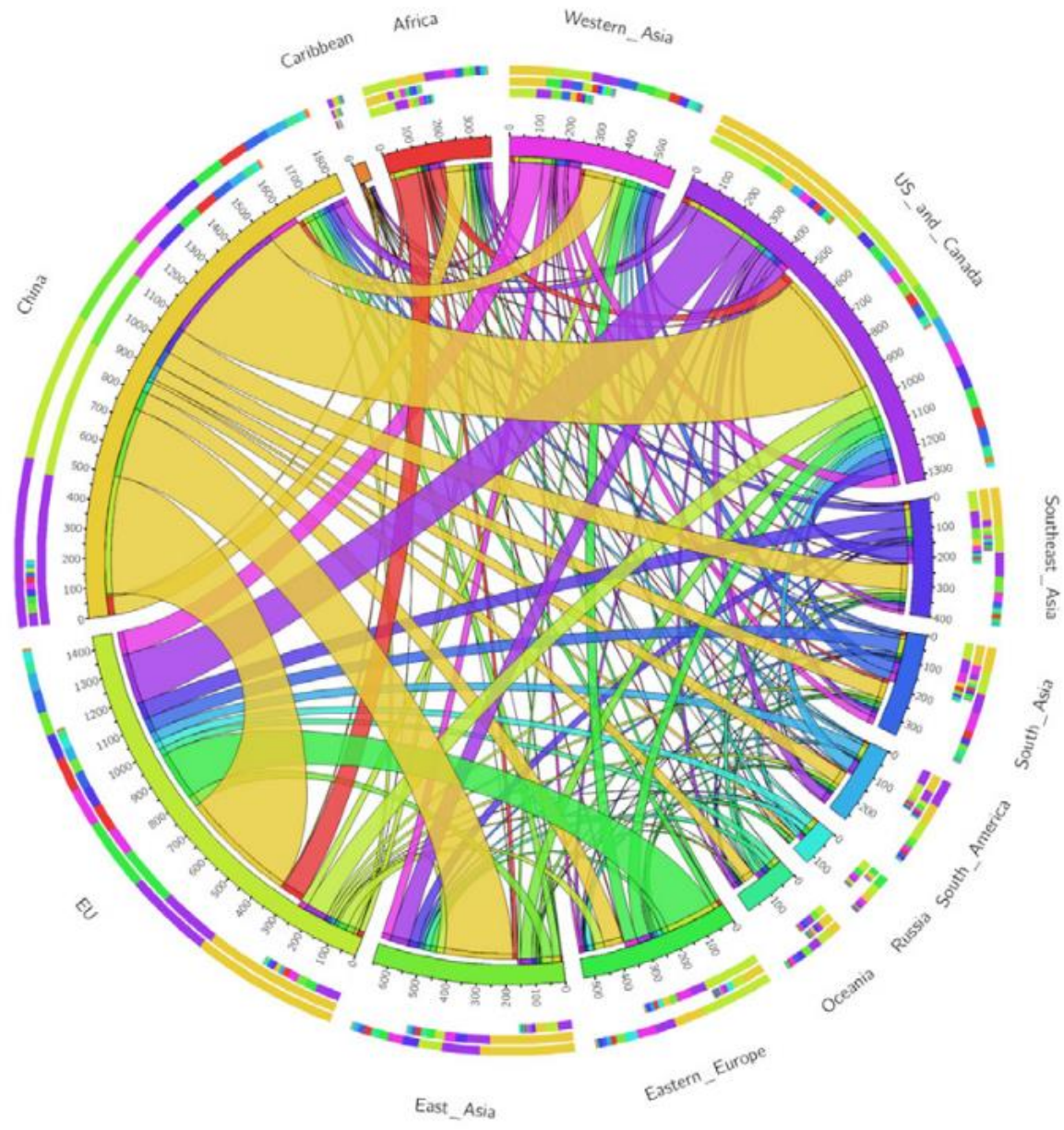
Understanding the energy metabolism of World economies through the joint use of Production- and Consumption-based energy accountings

Matteo V. Rocco ^a, Rafael J. Forcada Ferrer ^a, Emanuela Colombo ^a

Fighting carbon leakage through consumption-based carbon emissions policies: Empirical analysis based on the World Trade Model with Bilateral Trades

Matteo V. Rocco ^a, Nicolò Golinucci ^{a, b}, Stefano M. Ronco ^c, Emanuela Colombo ^a





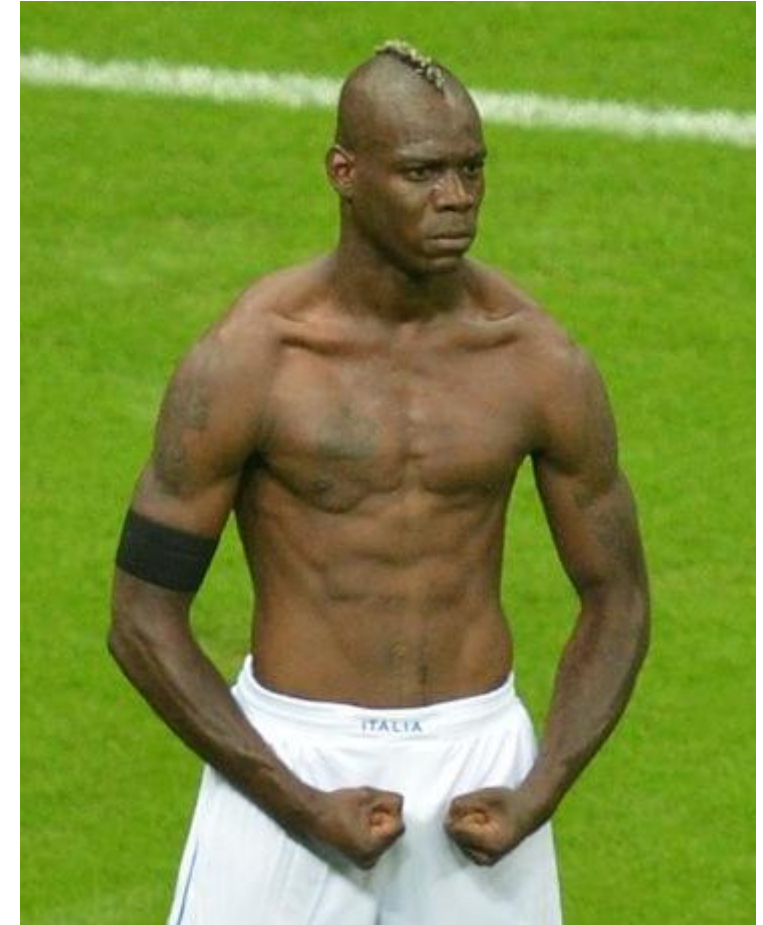
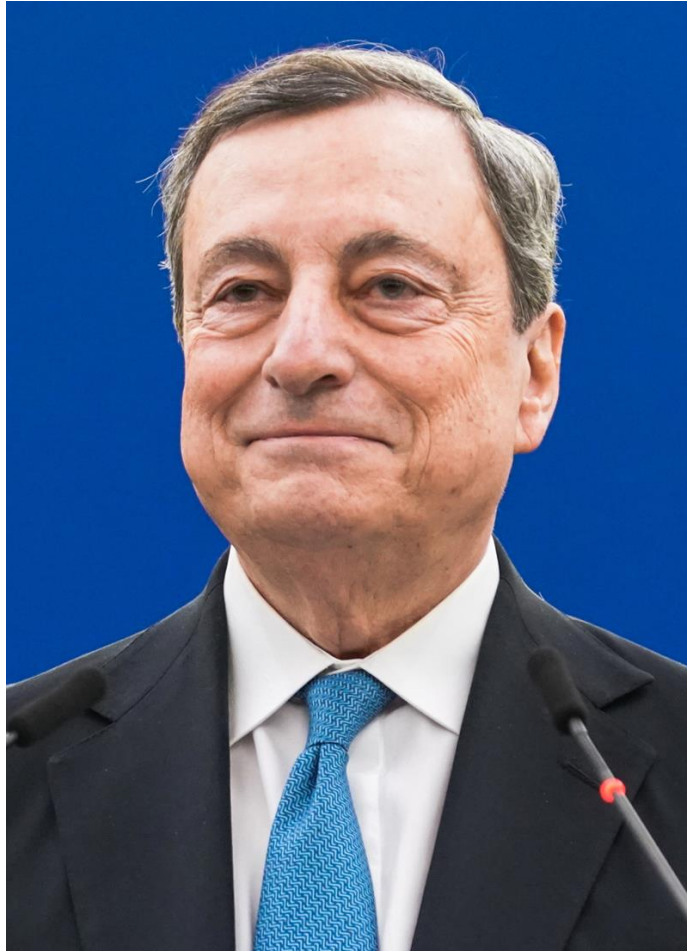


Multifunctional Analysis of Regions through Input-Output





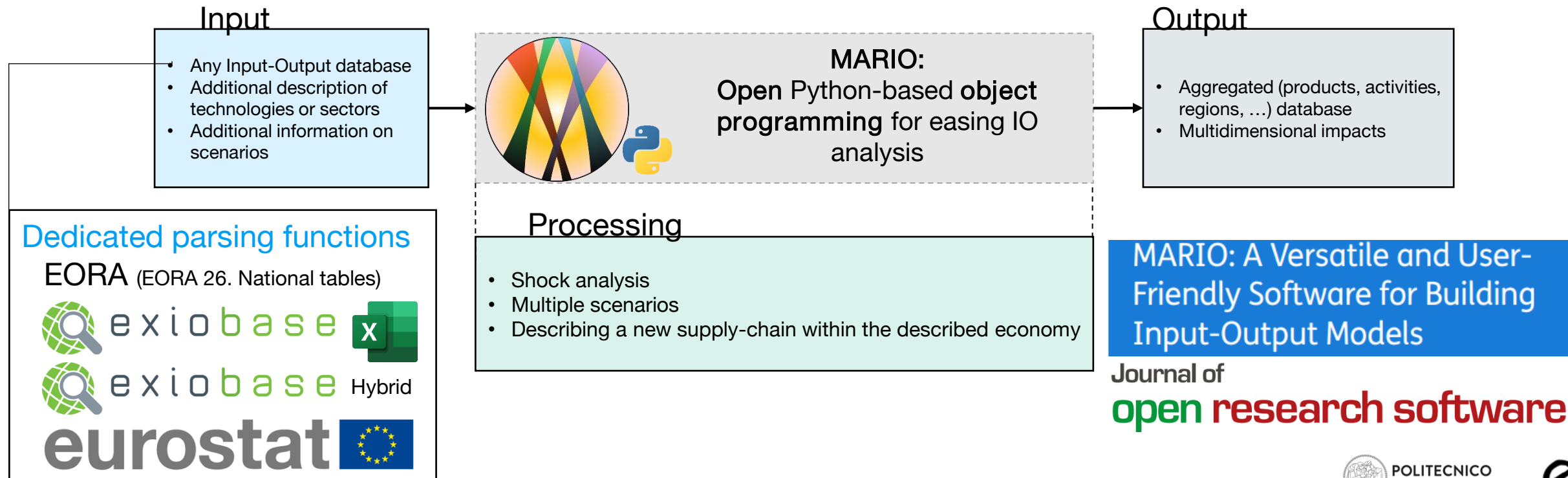
Making MRIO open-source software easy to remember through other famous Marios



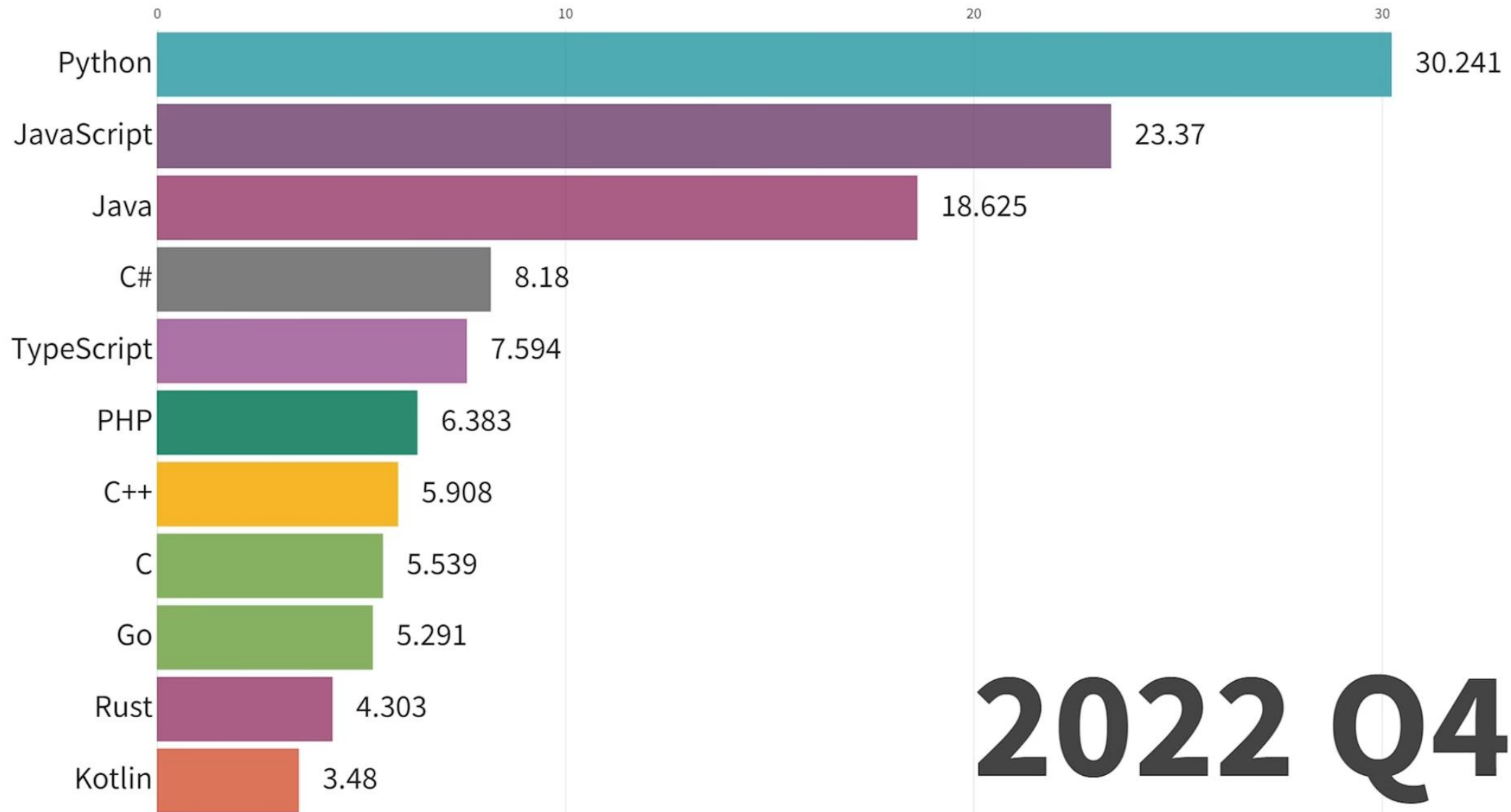
What is MARIO

MARIO is an open-source Python package, a framework aiming at easing input-output analysis operations and modelling

- MARIO stands for **M**ultifunctional **A**nalysis of **R**egions through **I**nput-**O**utput
- It handles some of the most well-known Input-Output or Supply and Use tables with dedicated parsing functions...but it virtually handles whatever table, including customized ones
- It is based on Python and available on [Github](#) ([it-is-me-mario/MARIO](https://github.com/it-is-me-mario/MARIO)), [Zenodo](#) ([record/5879383](https://zenodo.org/record/5879383)) and comes with a [full documentation](#) (including tutorials and installation guide - mario-suite.readthedocs.io)



Why Python?



2022 Q4

Anacondas, pythons...and spyders from Mars jupyter

If you are not familiar with Python all this names will make you feel confused

Anaconda: a distribution of Python and R that simplifies package management and deployment. It includes many pre-installed libraries and tools for data science and machine learning.

Python: a high-level programming language known for its readability and versatility, widely used for web development, data analysis, artificial intelligence, and more.

Spyder: an integrated development environment (IDE) for Python, included with Anaconda, designed for scientific computing. It provides features like code editing, debugging, and interactive testing.

Jupyter: an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text. It is especially popular for data analysis and visualization.



Anacondas, pythons...and spyders from Mars jupyter

You can even use **Visual Studio Code (VSCode)**: a free, open-source **code editor** developed by Microsoft. It supports multiple programming languages and offers features like debugging, version control, and extensions for additional functionality. It is popular among developers for its flexibility, ease of use, and integration with various development tools.

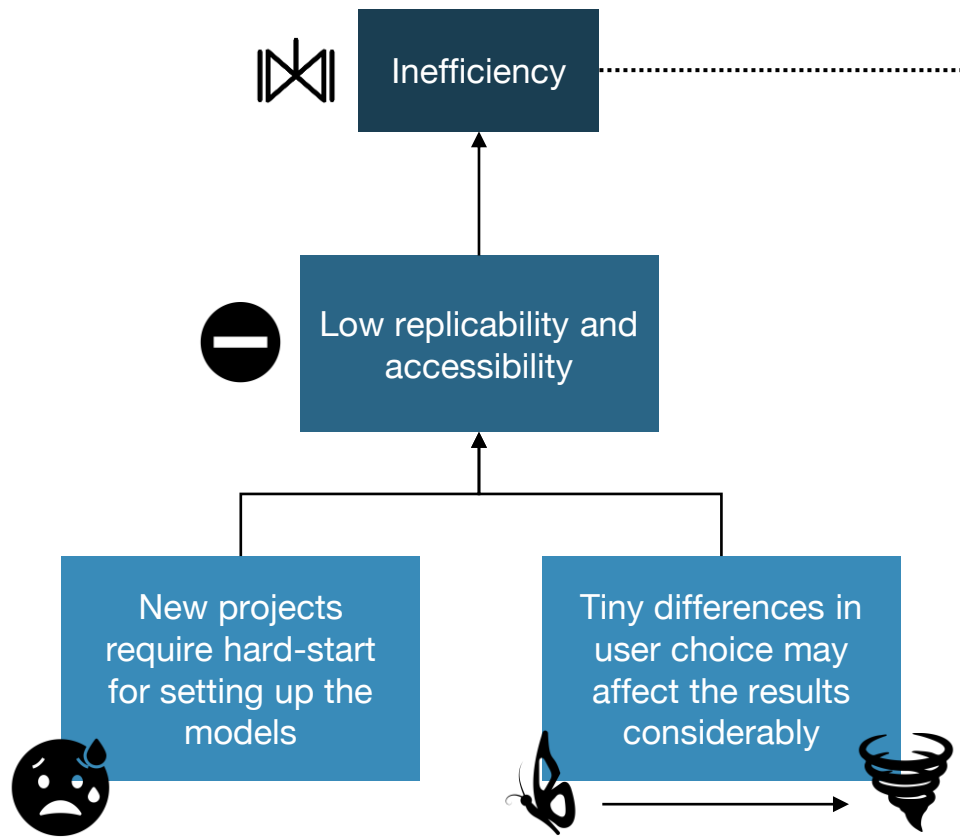
Environment: an **isolated setup** where specific Python versions and **packages** are installed. It ensures that projects have the necessary dependencies without conflicts, making development and deployment more manageable.

Python Package: a **collection** of Python **modules** that provide specific functionality or tools. Packages are used to extend the capabilities of Python, and they can be easily installed and managed.



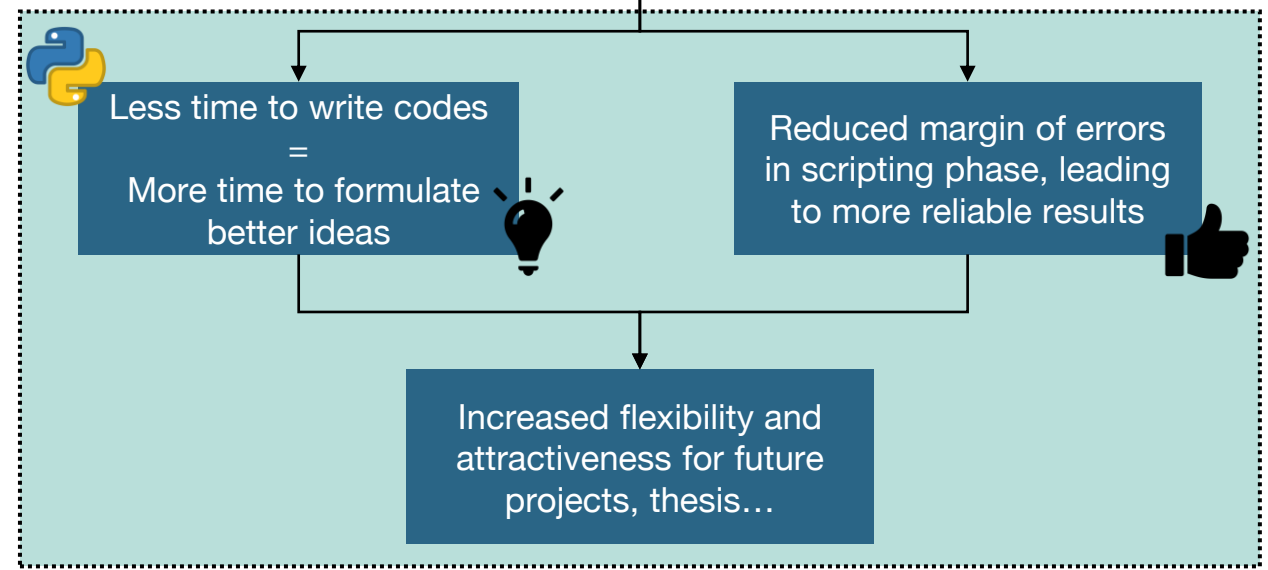
Why do we need an open-source package for input-output analysis?

Synthetic “problem tree” based on lessons learnt



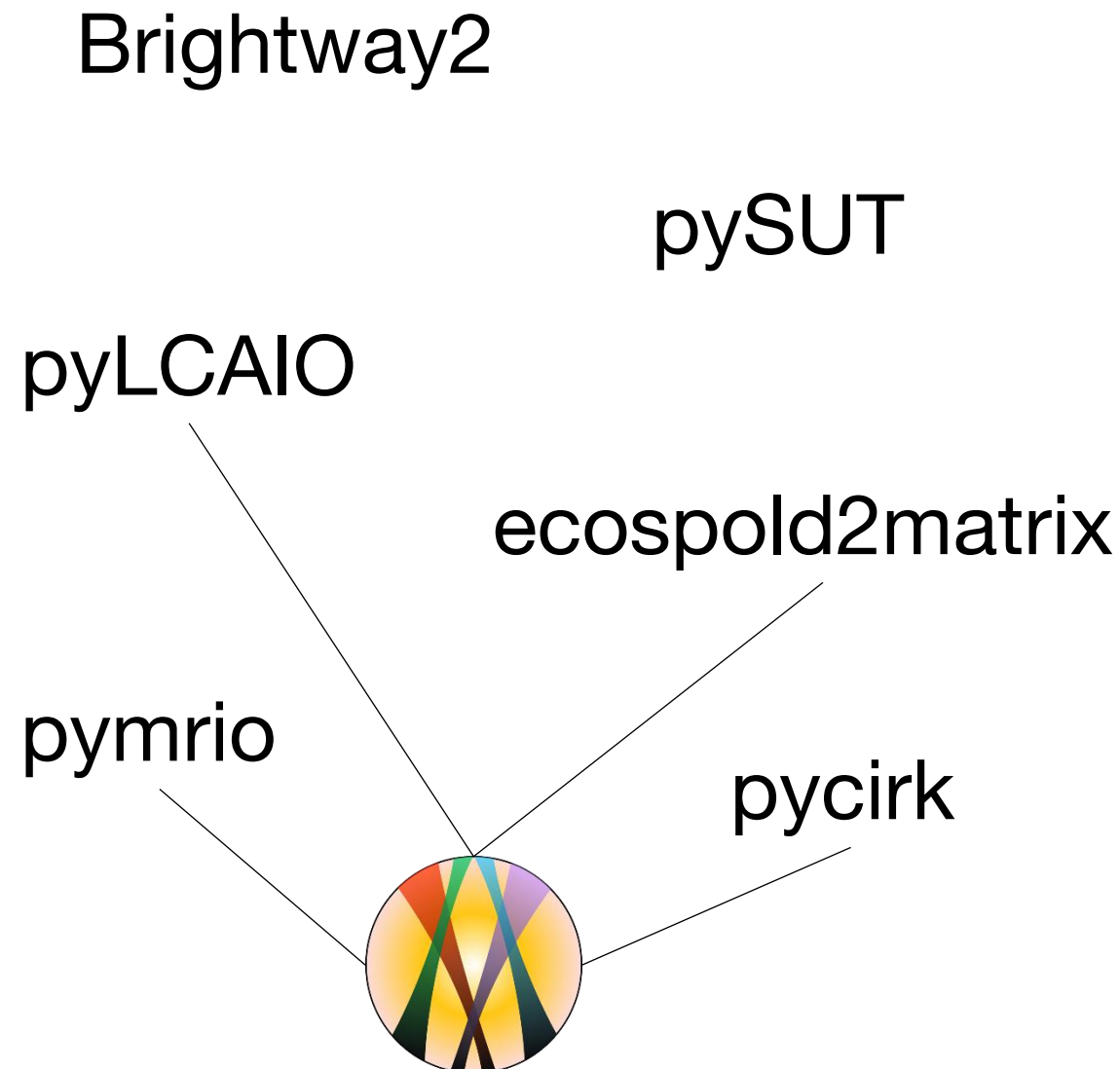
Solution?

Adoption of an integrated tool to support model generation



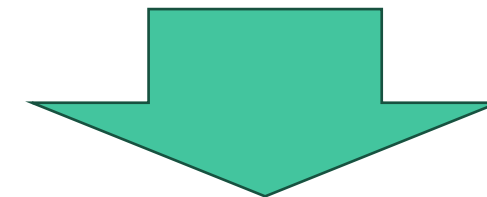
Bringing contribution to standardize IOA practices

There are many Industrial Ecology open tools. Why did we develop MARIO?



Main limitations of open science

- Lack of comprehensiveness
- Each model is an island
- While in the learning process, starting from scratch is the most effective way to learn programming

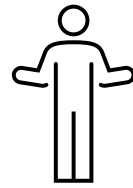


Possible solutions

- Integrations
- Documentation
- Teaching
- YouTube tutorials

How OOP works?

Objective Oriented Programming (OOP) makes our life easier



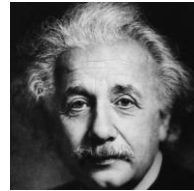
Human as a **class**

Properties

- ✓ Age
- ✓ Job
- ✓ Sex
- ✓ Nationality
- ✓ Languages

Functionalities

- ✓ Travel
- ✓ Migrate
- ✓ Learn Language
- ✓ Change Job
- ✓ Get older



People as a **instances**

- ✓ Age: 30
- ✓ Job: Startupper
- ✓ Sex: Male
- ✓ Nationality: Italian
- ✓ Languages: Italian
- ✓ Status: Alive



- ✓ Age: 76
- ✓ Job: Professor
- ✓ Sex: Male
- ✓ Nationality: German
- ✓ Languages: German
- ✓ Status: Died



Migrate



Migrate



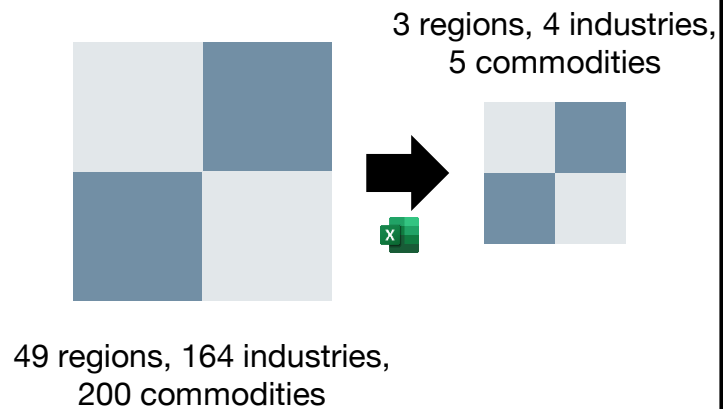
- ✓ Nationality: Italian, **American**
- ✓ Languages: Italian, **English**

What is MARIO – Processing

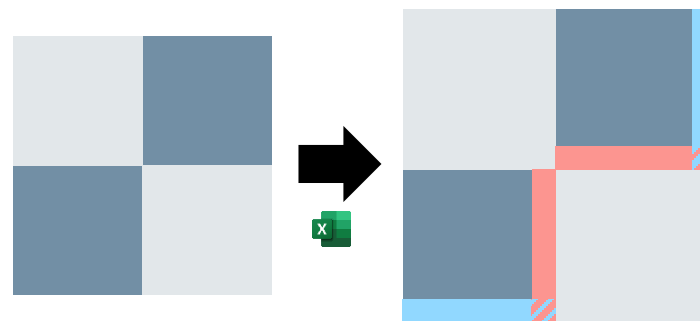
MARIO scales up the navigation and application of input-output tables without requiring advanced programming skills and supporting the user via Excel interface

It allows to:

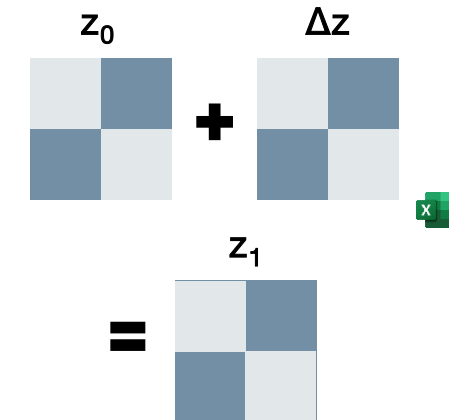
Aggregate tables



Add new **activities** and/or **commodities**

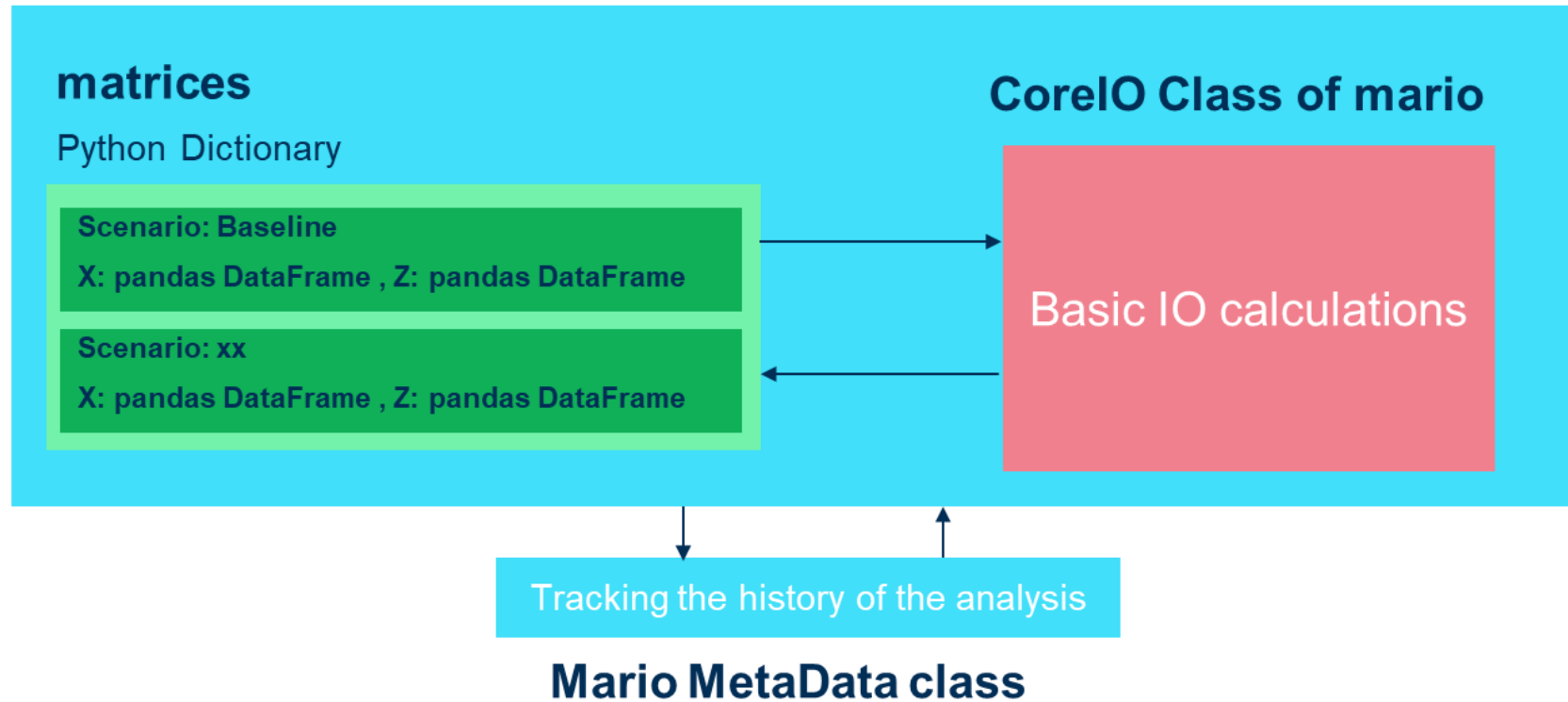


Implement shocks



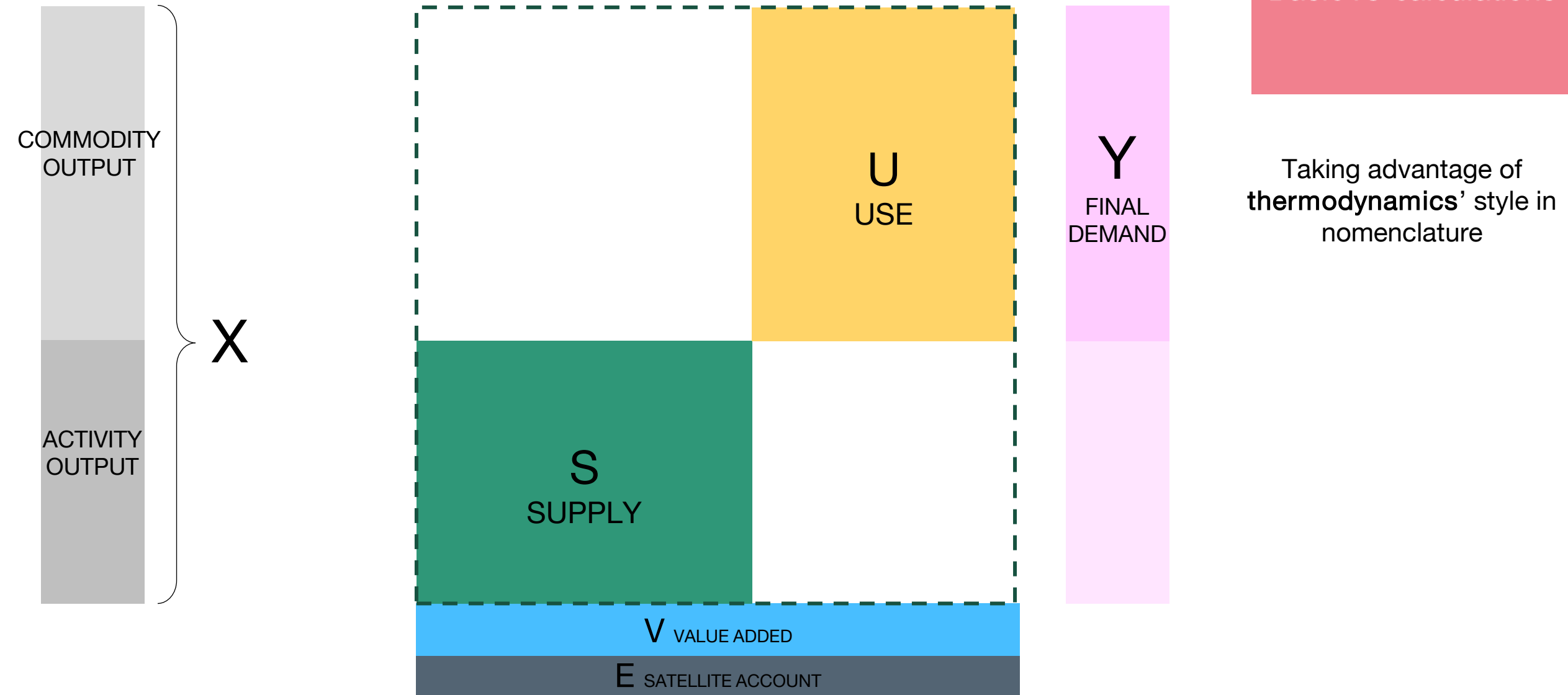
Code structure

Database Class of mario

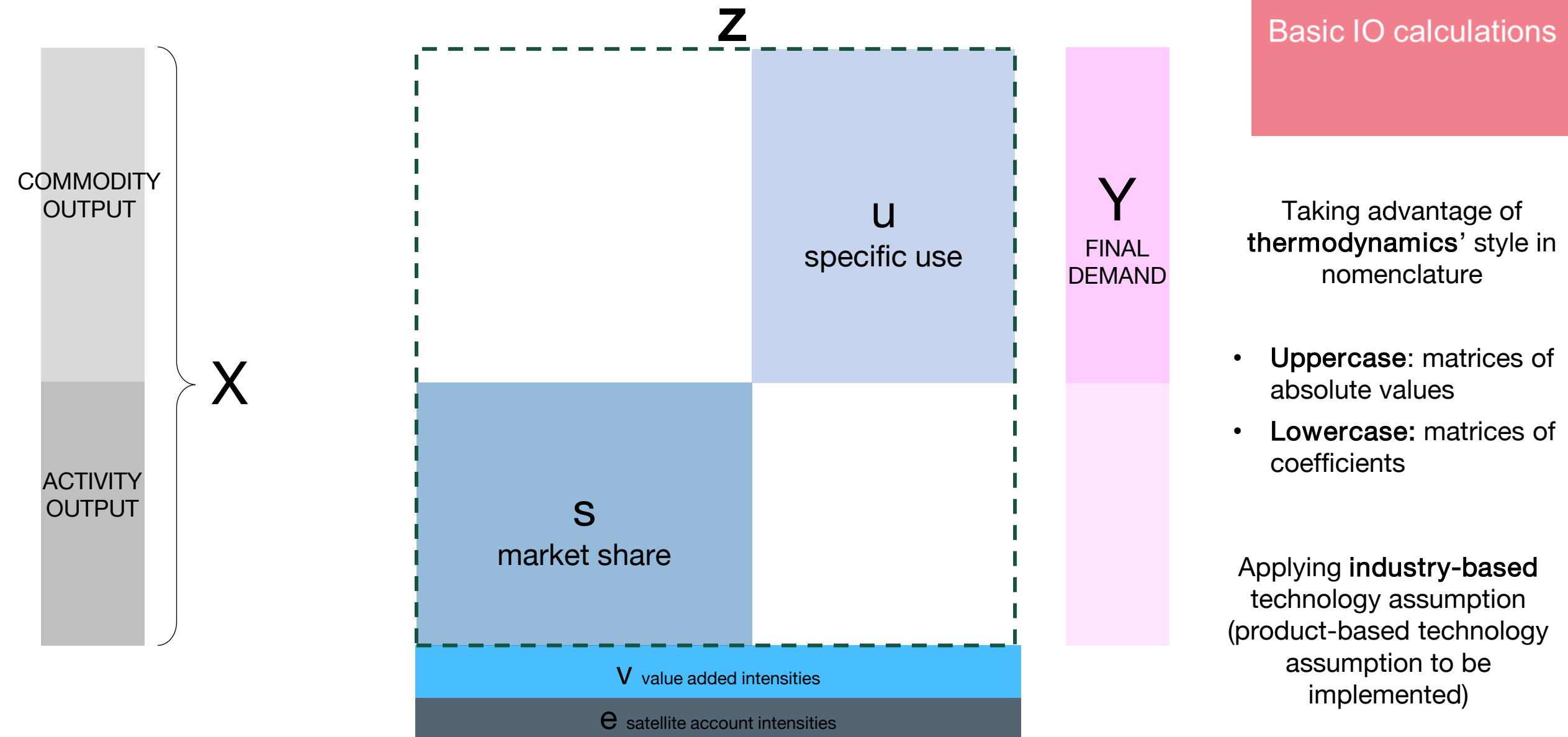


- Handling **any unit** of measure initialized when parsing the table
- Handling **scenarios** thus easing results exploration (function query)

Code structure – Calculations



Code structure – Calculations



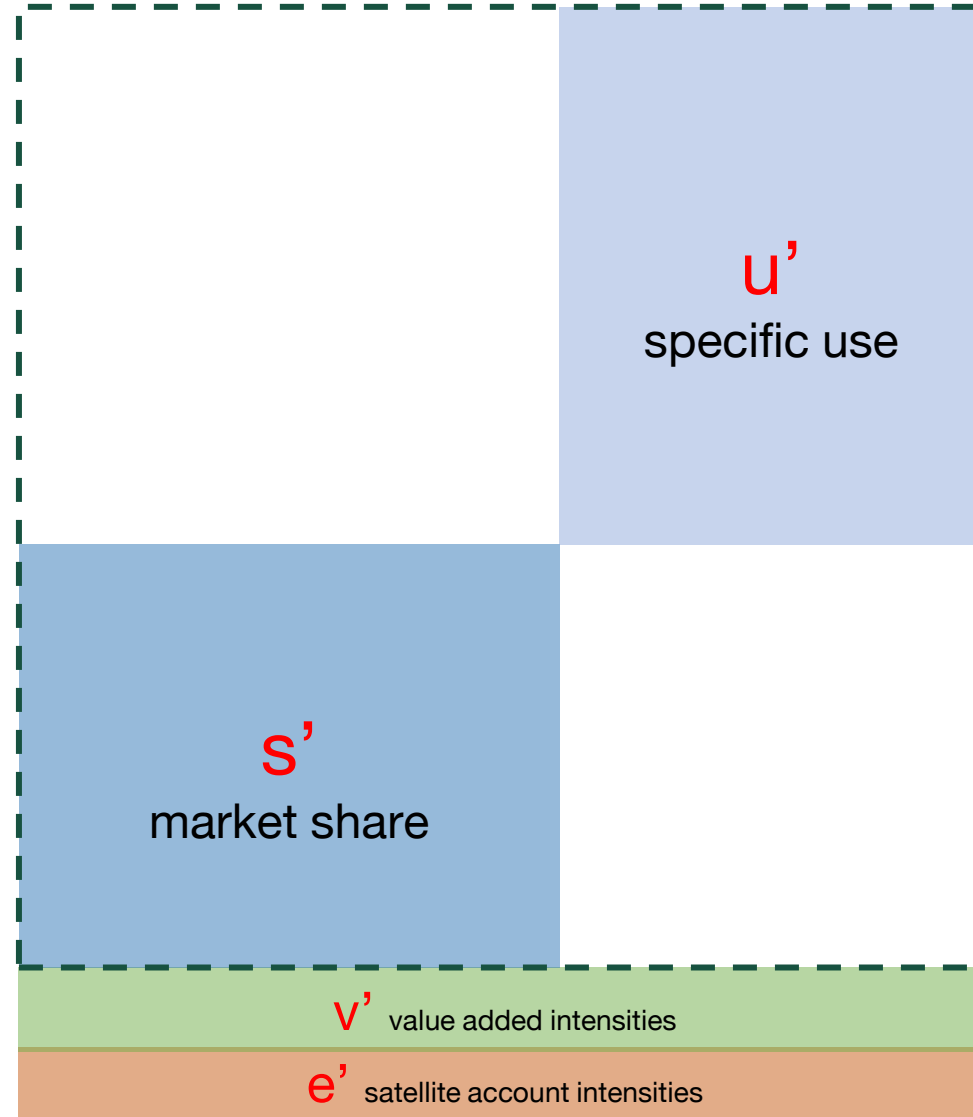
Code structure – Calculations

z'

COMMODITY
OUTPUT

ACTIVITY
OUTPUT

X'



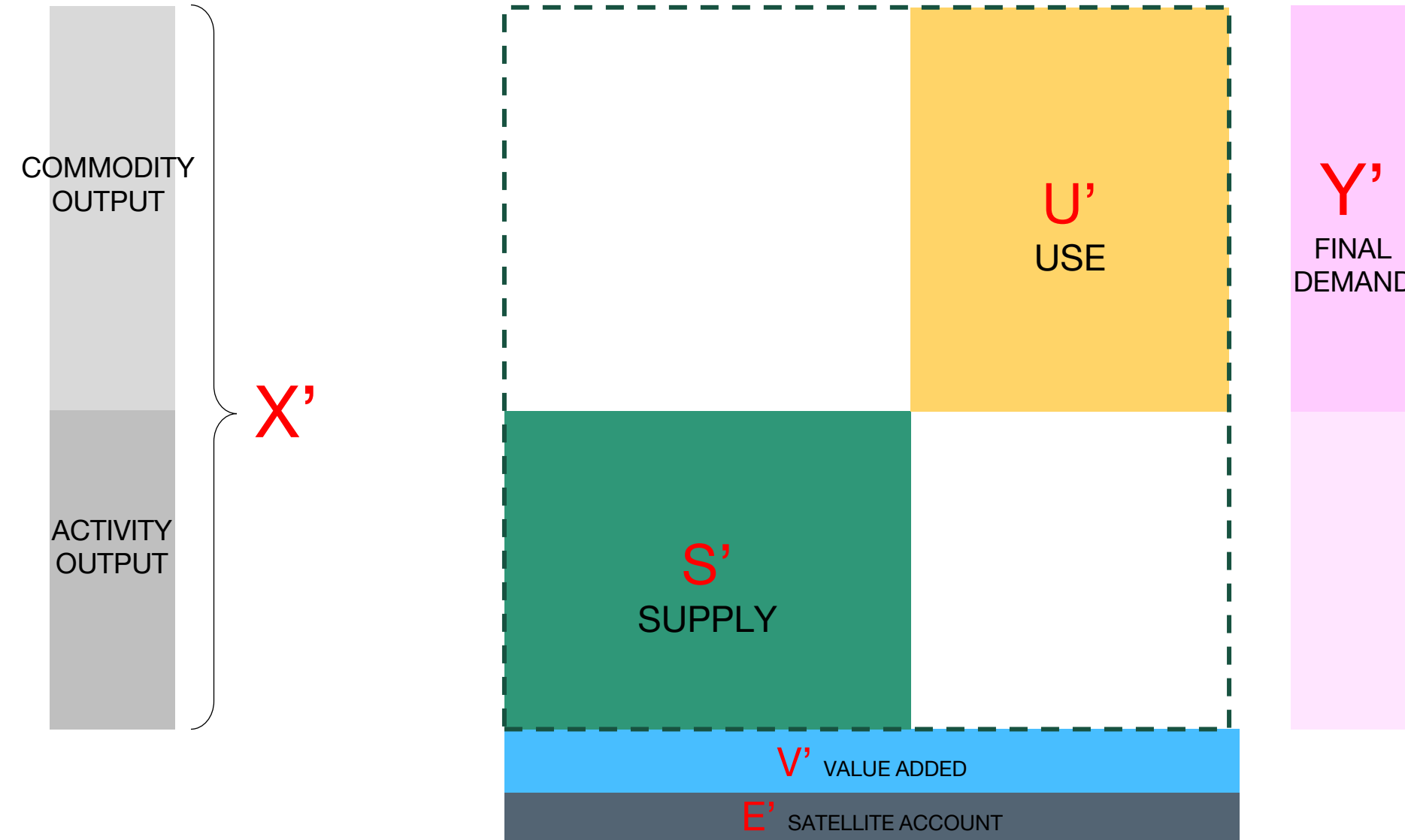
Y'
FINAL
DEMAND

Basic IO calculations

Performing shock
analysis...

Code structure – Calculations

Z'



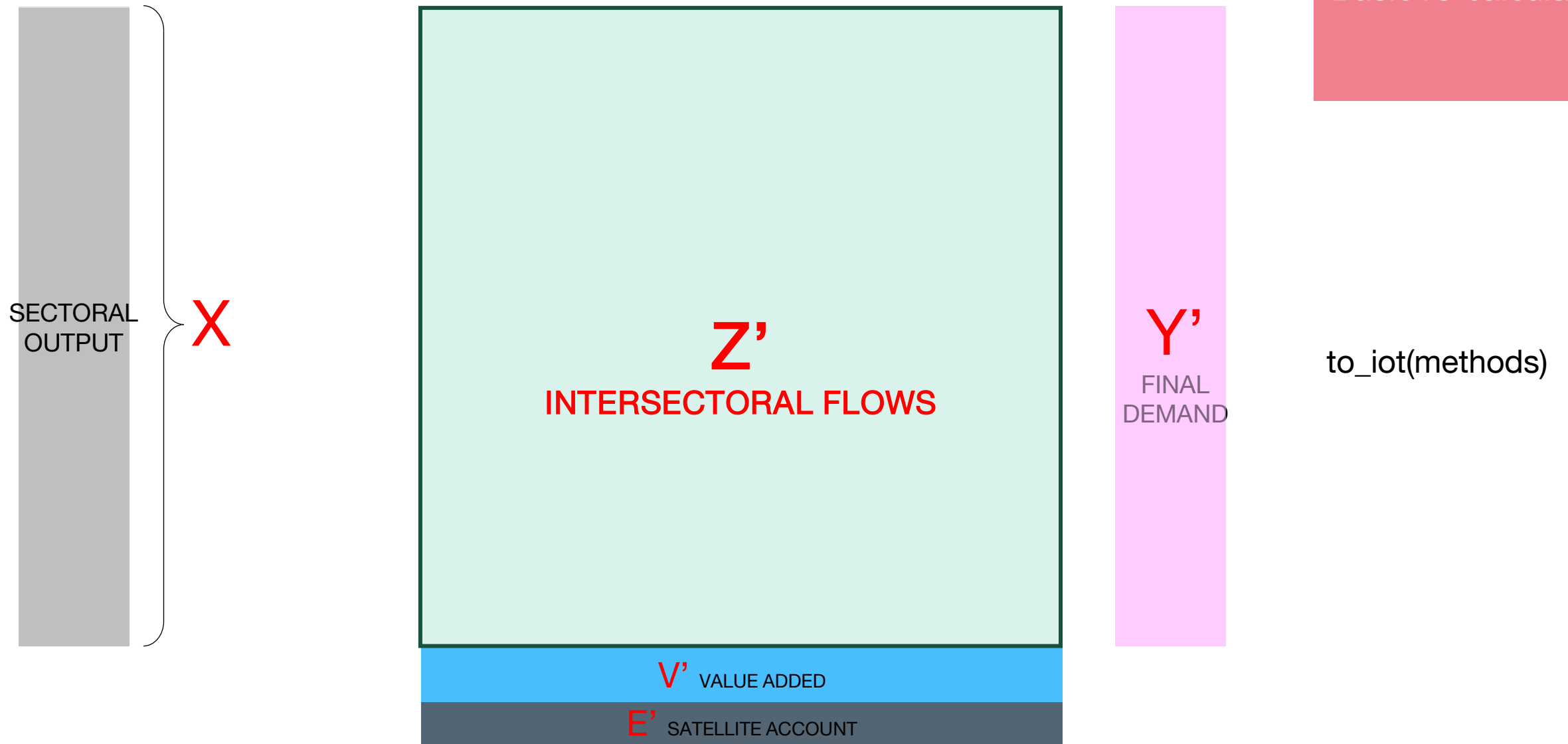
Basic IO calculations

Performing shock analysis...

... and assessing its impacts

shock_calc

Code structure – Calculations



You don't like our nomenclature?

Basic IO calculations

(nobody likes other people's nomenclature)

Good for you!

You can use the new MARIO v3 features in the settings, for using your own terminology!



Terminology

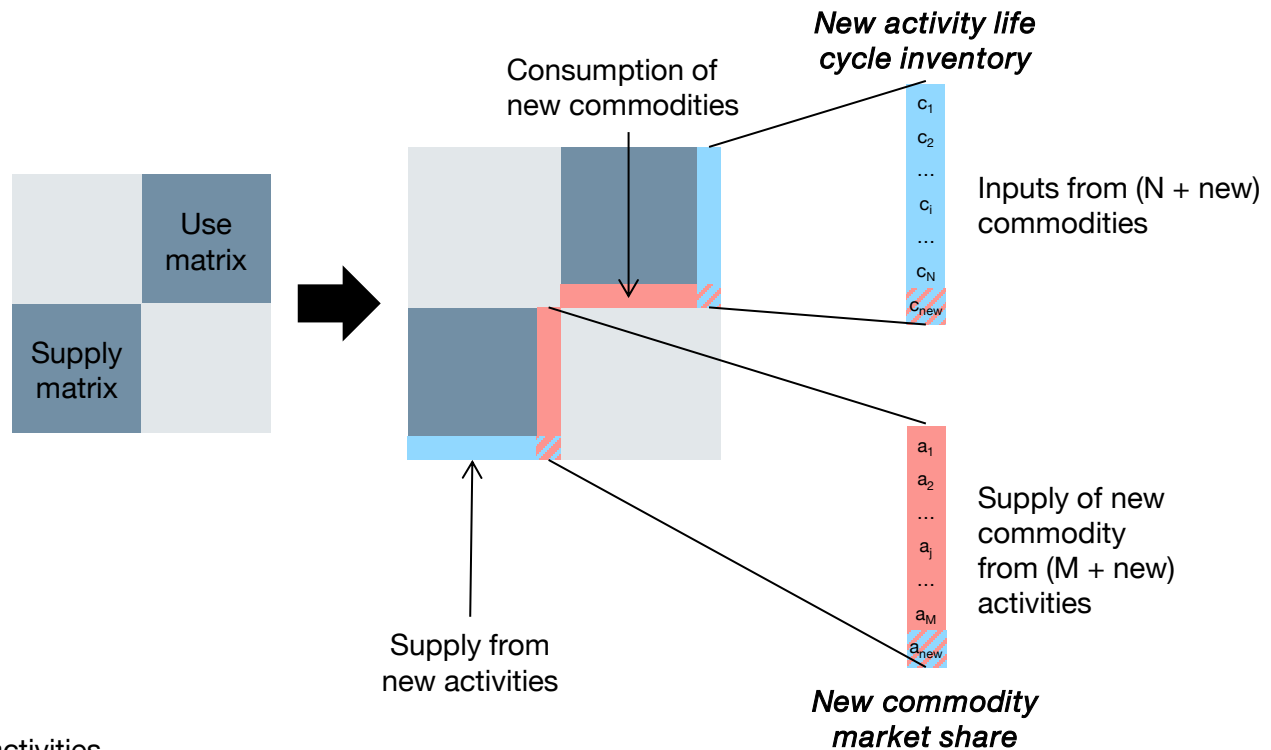
In the lack of consistent terminology for IO systems in the scientific community, MARIO uses its own customized variable names. MARIO follows a thermodynamic way of nomenclature which:

- Uppercase letters represents Flows
- Lowercase letters represents Coefficients

Following table represents the variables and their explanations in MARIO:

variable name	also known as	extended name
Z	T	Intersectoral transaction flows matrix
z	A	Intersectoral transaction coefficients matrix
w	L	Leontief coefficient matrix
Y	F	Final demand matrix

Adding new sectors



Python Dictionary

Scenario: Baseline

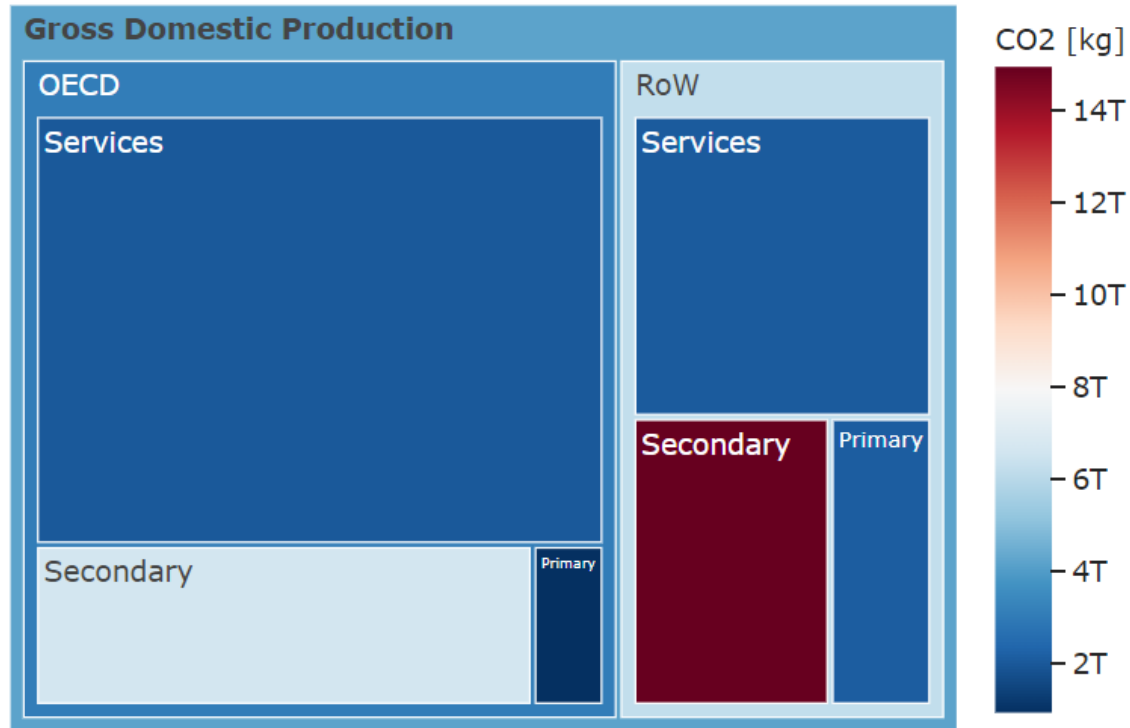
X: pandas DataFrame , Z: pandas DataFrame

Scenario: xx

X: pandas DataFrame , Z: pandas DataFrame

Adding new processes and products (intended as sector for IOT or commodities and/or activities for SUT) with `add_sector`

Examples of results



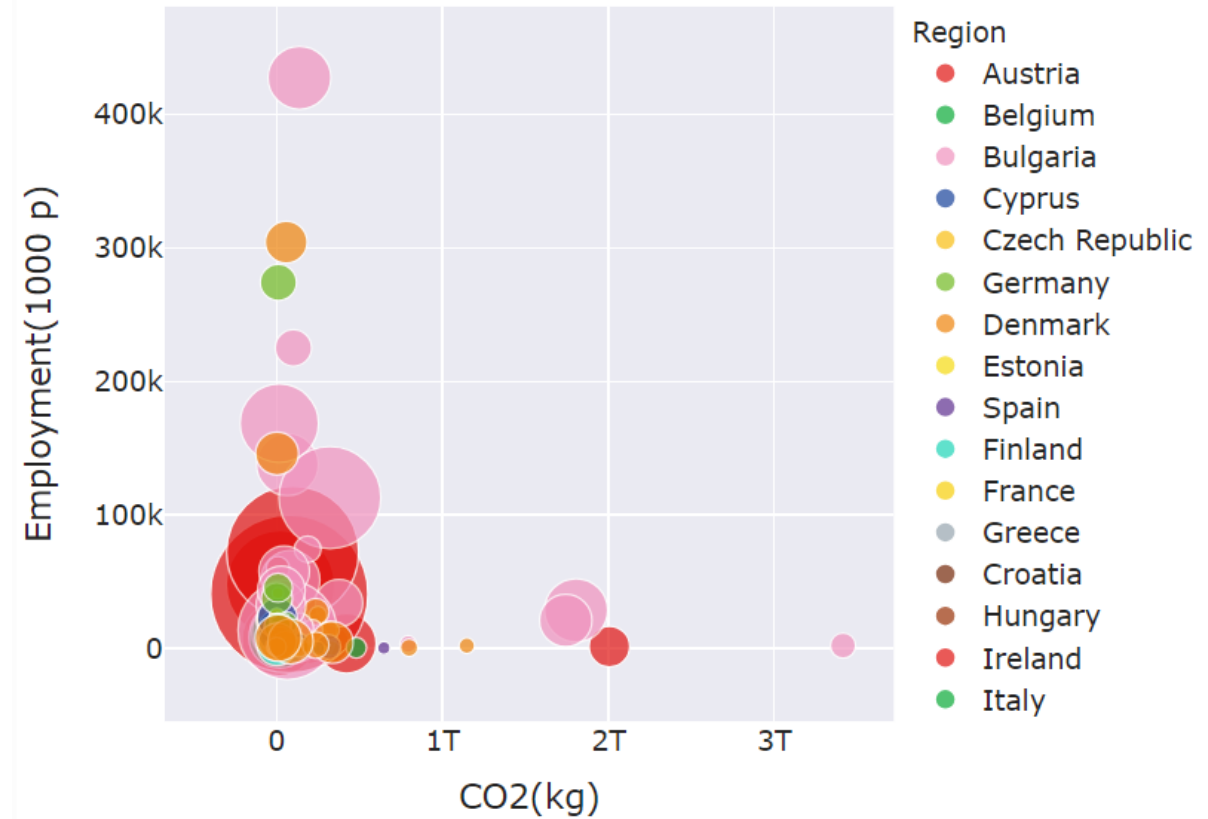
Python Dictionary

Scenario: Baseline

X: pandas DataFrame , Z: pandas DataFrame

Scenario: xx

X: pandas DataFrame , Z: pandas DataFrame



Taking advantage of **plotly** library to investigate full-scale or aggregated database

Examples of results

Python Dictionary

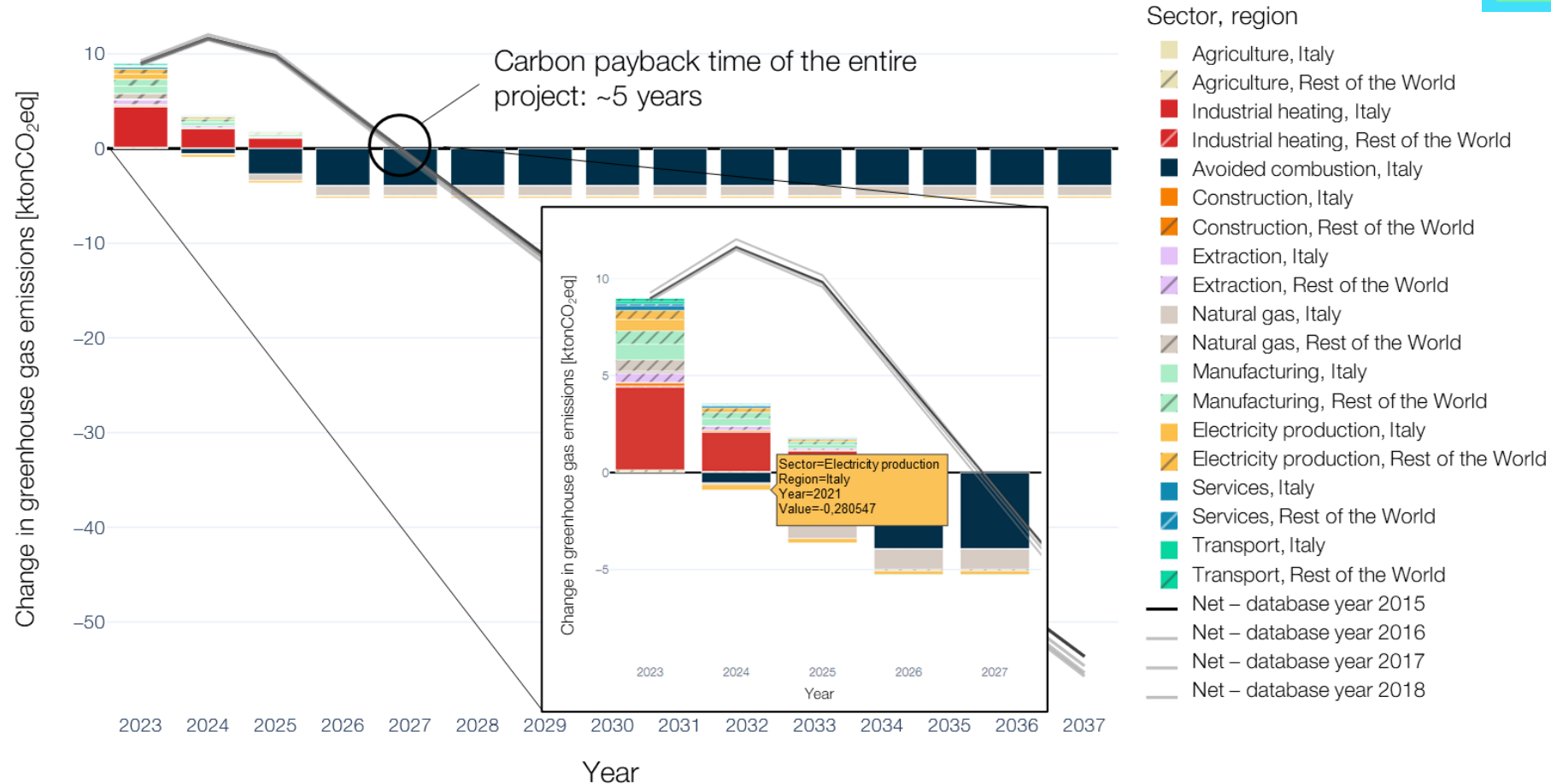
Scenario: Baseline

X: pandas DataFrame , Z: pandas DataFrame

Scenario: xx

X: pandas DataFrame , Z: pandas DataFrame

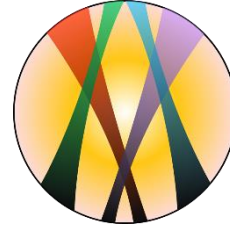
Variation in greenhouse gas by Year, Sector and region: net variation at the end of the project between -55.7 and 53.6 [ktonCO₂eq]



Taking advantage of the scenario parameter to develop comparative statics

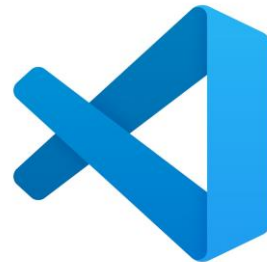
And now, a MARIO tutorial!

Installing MARIO and spyder starting from anaconda + some basic use of MARIO



We will use MARIO in spyder

You can also check the material available at: https://github.com/eNextHub/MARIO_IIOA24



Please, if you are going to use MARIO in your research, cite us at **Tahavori et al. 2023:**
MARIO: A Versatile and User-Friendly Software for Building Input-Output Models | Journal of Open Research Software (metajnl.com)

Thank you for your
attention!



POLITECNICO
MILANO 1863

www.polimi.it



eNextGen

spin off[®]
POLITECNICO DI MILANO

www.enextgen.it

