MESSAGE$_{ix}$ Workshop - Session II
Building an Energy System Model (Part 1)

Energy Program, International Institute for Applied Systems Analysis (IIASA), Austria

MESSAGEix Workshop (online), 8 Sep 2020

The MESSAGE$_{ix}$ workshop team, Sep 2020

Behnam Zakeri

Paul Kishimoto

Oliver Fricko

Francesco Lovat

Muhammad Awais

Energy Program, International Institute for Applied Systems Analysis (IIASA), Austria

The MESSAGEix Modeling Framework

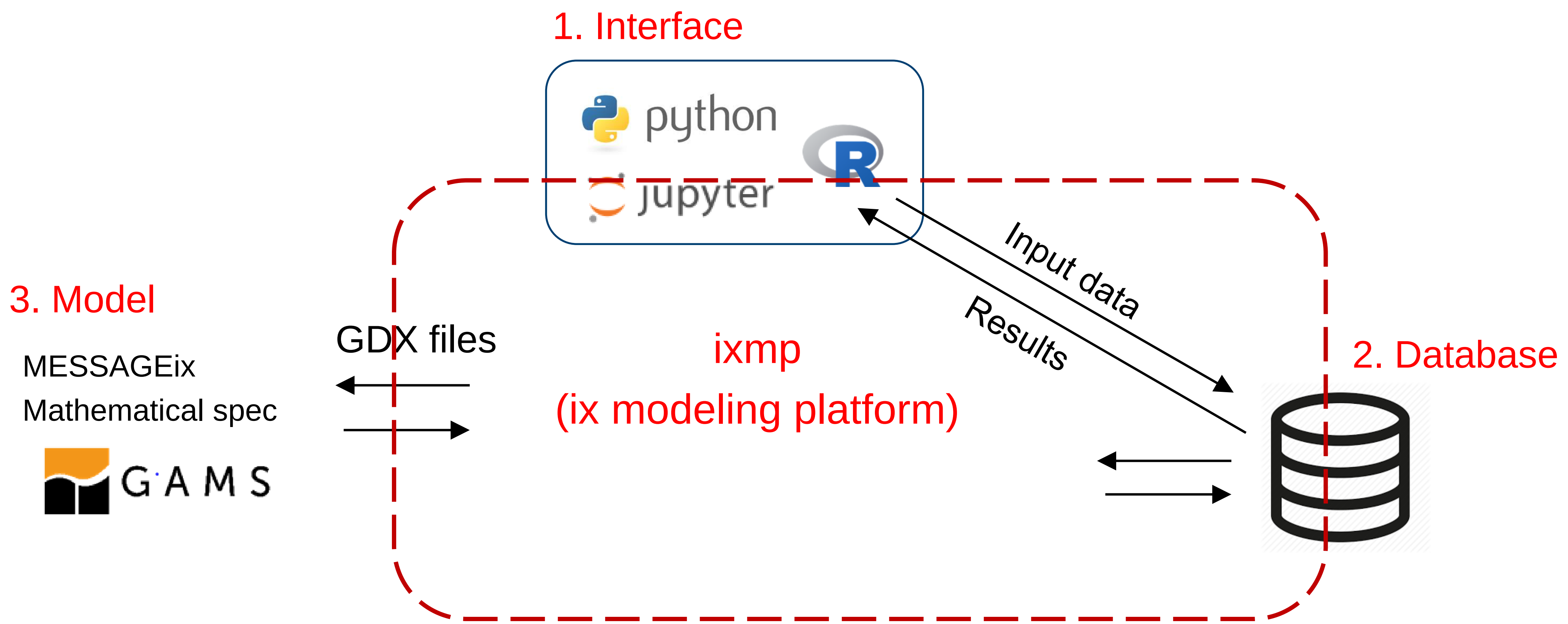
Recap...

- **MESSAGEix** is an open, version-controlled systems engineering **modeling framework**
- ix modeling ***platform*** (*ixmp*) is a data warehouse for facilitating high-powered modeling work
- python and R are the main interfaces for modelling using MESSAGEix
- MESSAGEix **mathematical model** is written in GAMS
- Documentation of the MESSAGEix model and tutorials are available online

https://docs.messageix.org

The MESSAGE$_{ix}$ framework: Workflow of modeling

Recap…



MESSAGEix framework: Building an energy system – Part 1

Agenda of this Session

- A note on optimization (if needed)
- MESSAGEix mathematical model and its structure
- Working with MESSAGEix tutorials: building a simple model

➔ Voting feature will be used to measure how much time we should spend

After this tutorial

The goal is to...

- Learn about the logic behind the MESSAGEix energy system model
- Be able to work on a MESSAGEix model using Jupyter Notebook
- Be familiar with basic terminology of a MESSAGEix model

Requirements

- MESSAGEix framework installed and running
- Knowledge on energy systems
- Patience, motivation, and curiosity

Energy Systems

Different scales: community, city, country, region, and global

- A system of energy resources, conversion/processing, transmission and distribution technologies, and services

- Technologies (cars, ...)
- Commodities (oil, gas, ...)
- Sectors (industry, transport, ...)
- Demand services (electricity, lighting, heating, ...)
- Agents (consumers, producers, Operators, etc.)

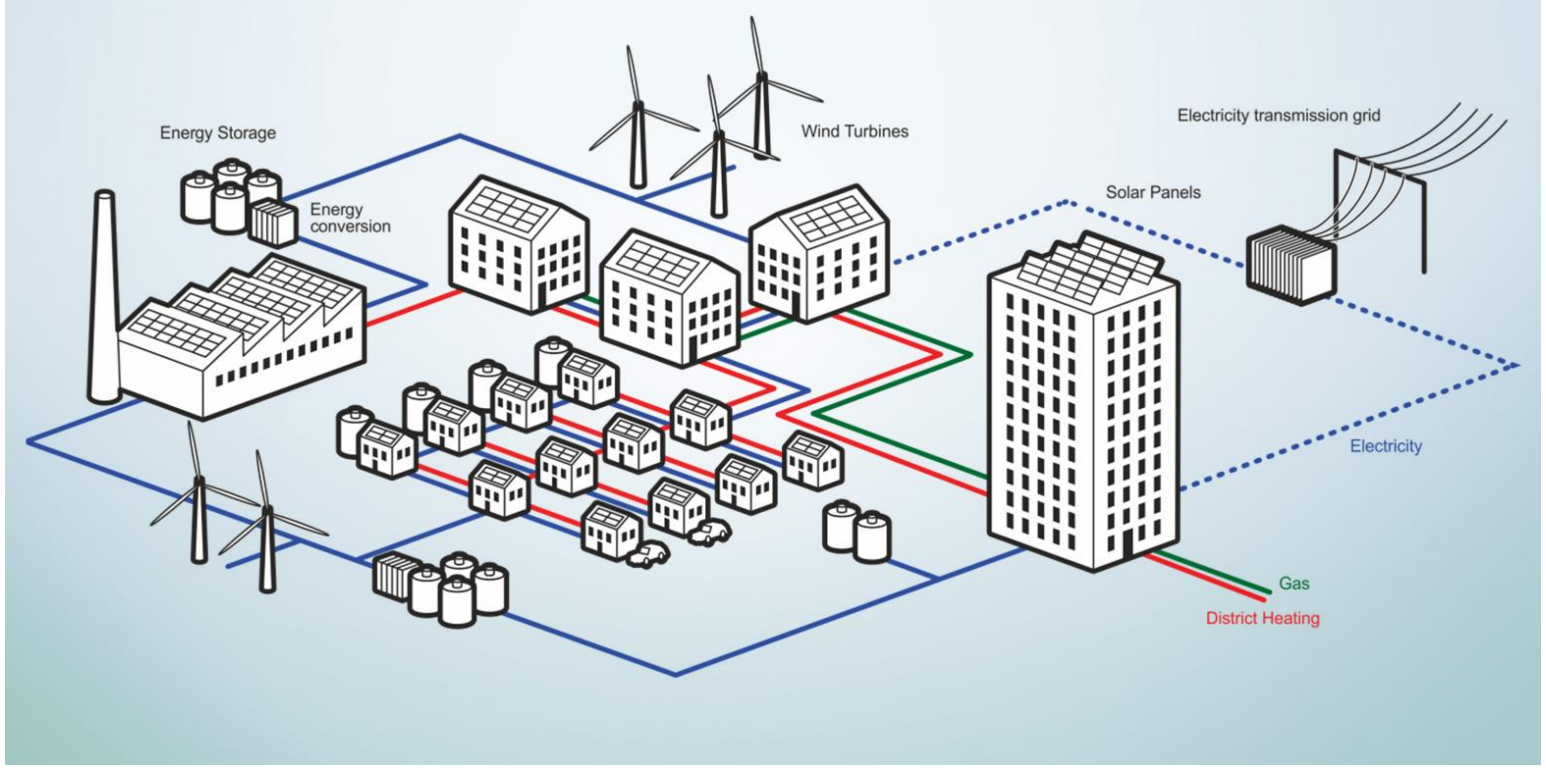


Image: ETH Zurich

Linear programming (LP)

Finding the best (optimal) solution

- The goal is to optimize a linear objective function
- There are a set of *decision variables*
- There are some constraints (bounds on or relationship between decision variables)

$$\begin{array}{ll} \text{Maximize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ \text{and} & \mathbf{x} \geq \mathbf{0} \end{array}$$

Example: the best way to commute to work

- **Decision variables:** walking, biking, bus, train, taxi, private car, car sharing
- **Objective function:** cheapest or fastest option (least environmental footprint, least walking option)
- **Constraints:** maximum 2 hours commute/day, maximum 300 euro/month, no later than 7 PM, ...
- **Feasible region**: usually there many alternative solutions but not all of them are feasible

Linear programming (LP) (reminder)

Applications of LP

- Production management
- Personnel management
- Marketing management
- Resource/ inventory management
- Blending problem, etc

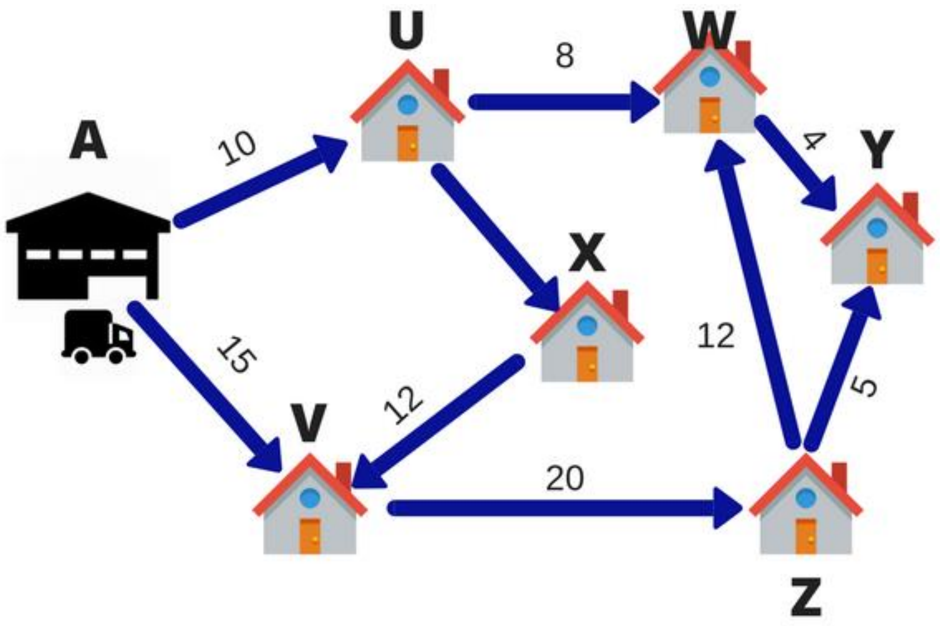


image: deepai.org

Principle:

Maximizing the utility (use of resources)

Or Minimizing the cost of meeting a service ➔ MESSAGEix

MESSAGEix: a model for investment and planning

Minimizing total discounted cost of the system

- **Objective:** The least cost option for meeting certain services (demand) ➔ min $cT \cdot x$
- **System:** a network of technologies (processes), resources, and commodities (products)
- **Cost of the system**: installing/maintaining ***capacity***, cost of ***activity*** (O&M), taxes, emission penalties, land use costs (if any), etc.
- Constraints: maximum use of a technology, growth/decline rates of activity, capacity factor, etc.

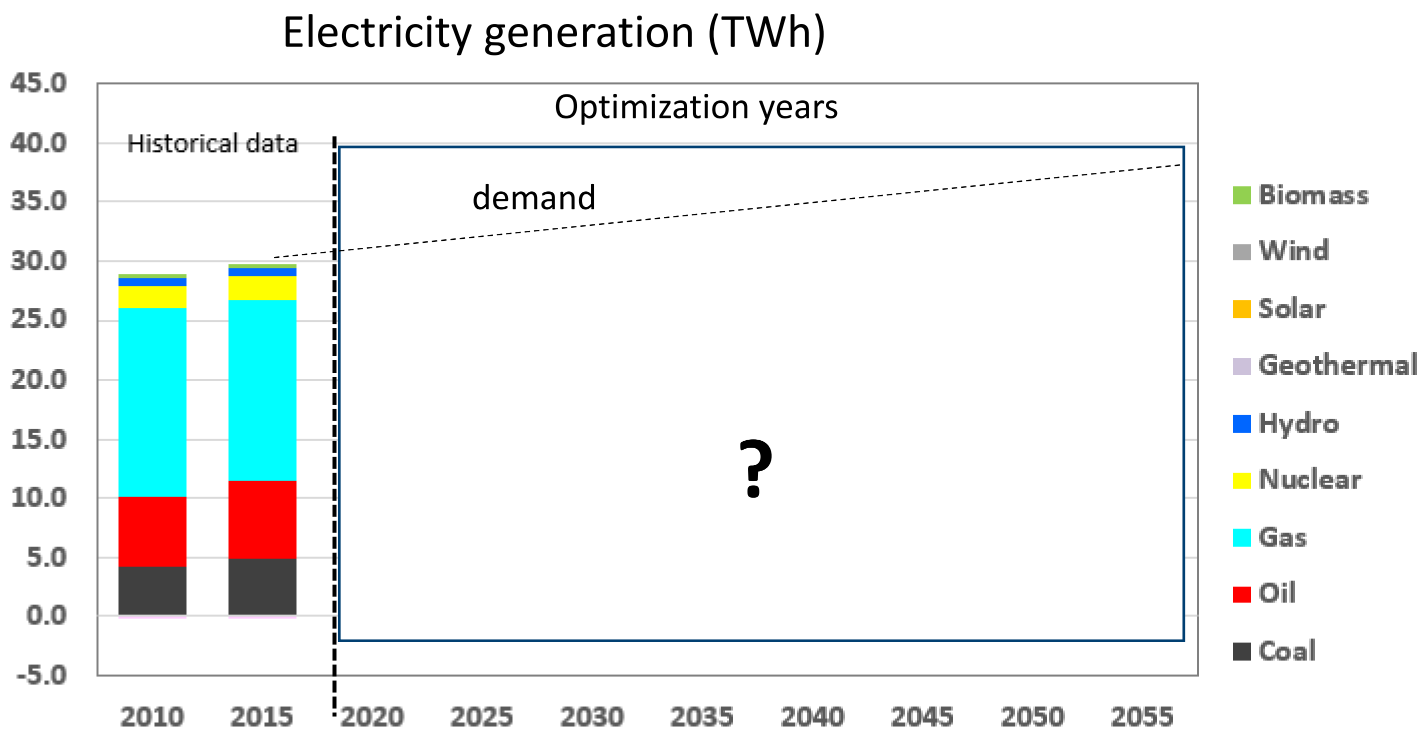
➔ $s.t. A \cdot x \leq b$

A note on "capacity" and "activity" (MESSAGEix formulation)

- **Capacity:** installed units of a technology (e.g. 150 MW power plant)
- **Activity:** operation of that technology (e.g., 800 GWh)
- **Reminder:** capacity ≠ activity   capacity * capacity factor = activity

The MESSAGE$_{ix}$ framework : Investment planning

From historical activity/capacity to model years

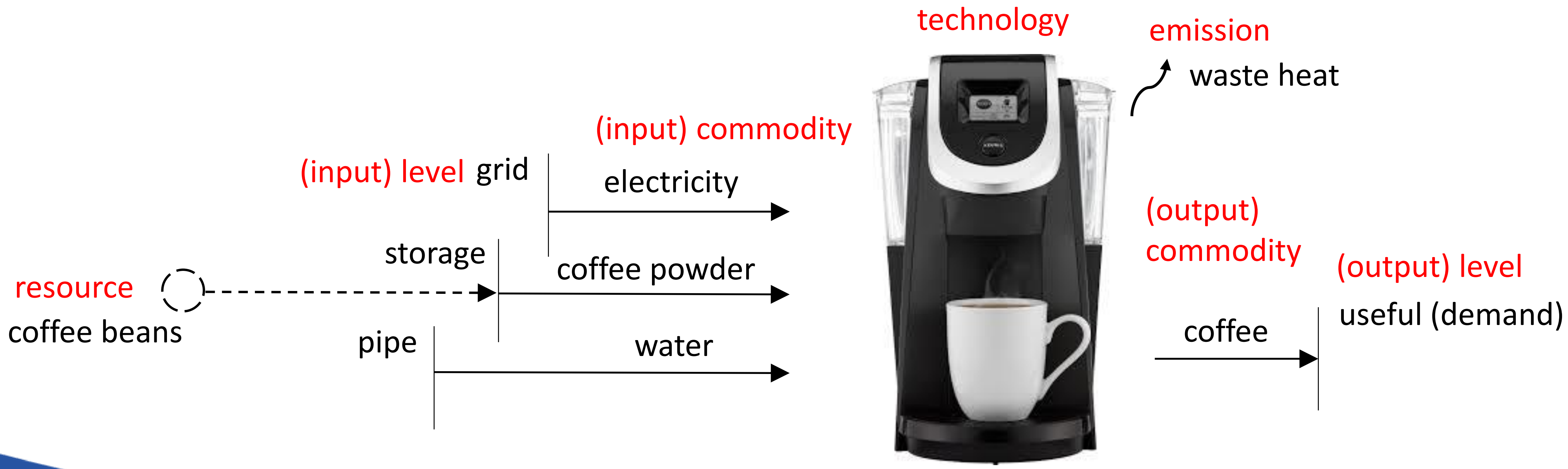


MESSAGEix: a technology-based model

Technologies and resources meet demand

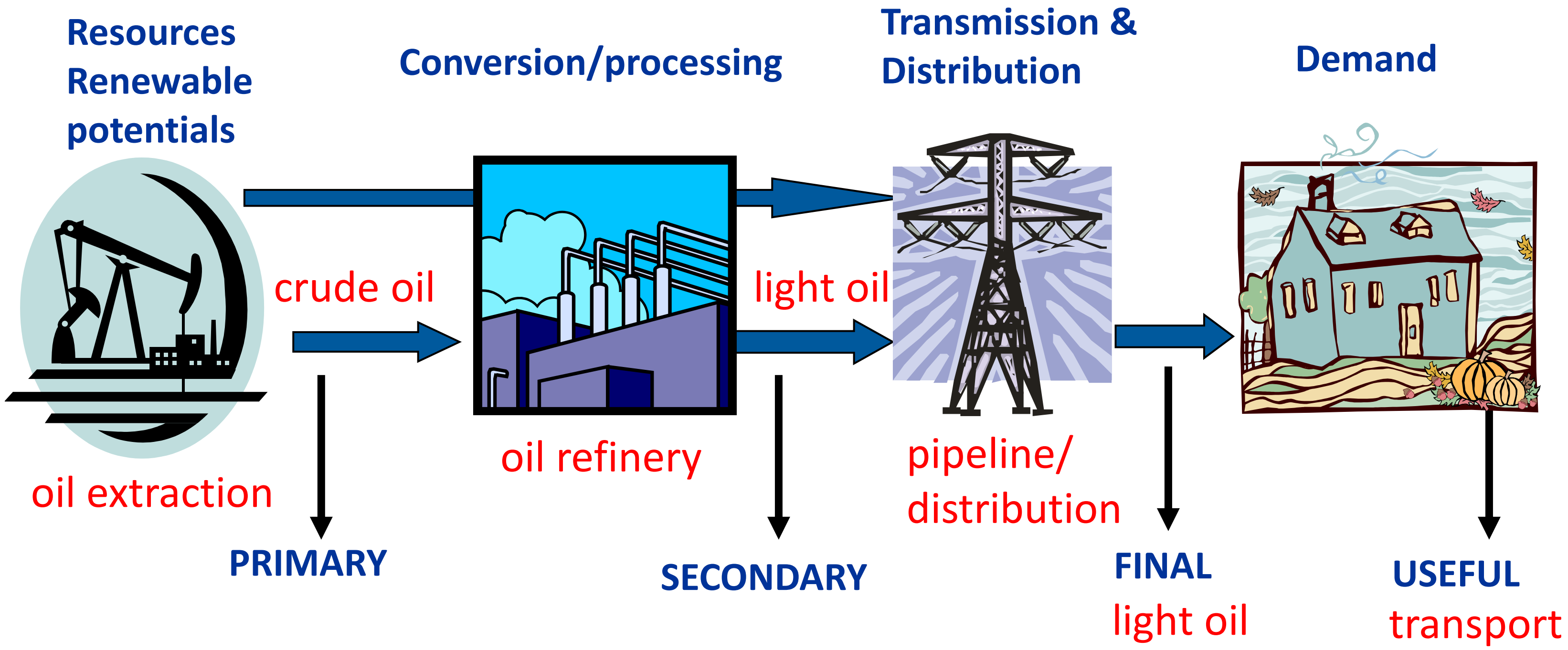
- Example **technologies:** electric car, reactor, pipeline, power plant, building, ship, industrial process

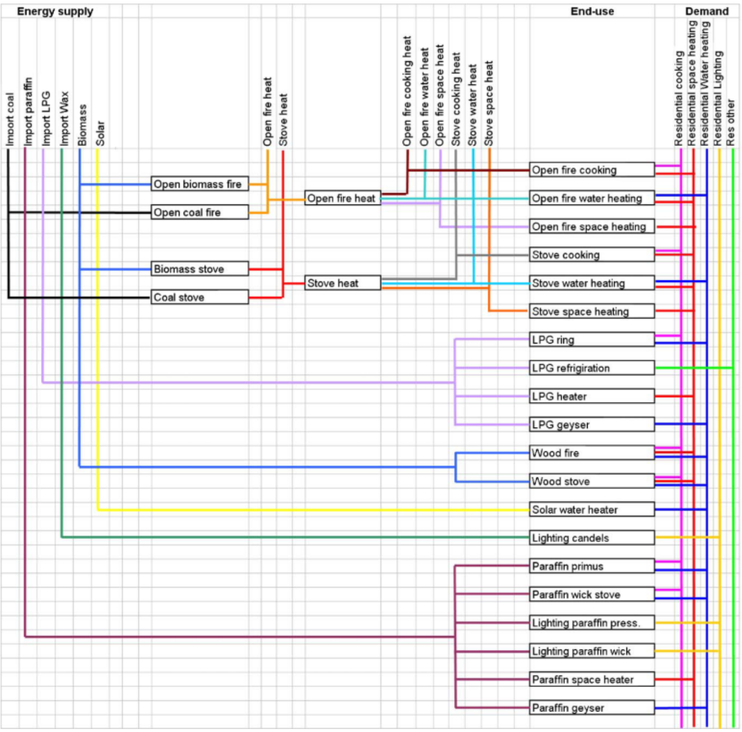
A sample technology: coffee machine



MESSAGEix: a technology-based model (2)

A flexible representation of processes technologies and interlinkages



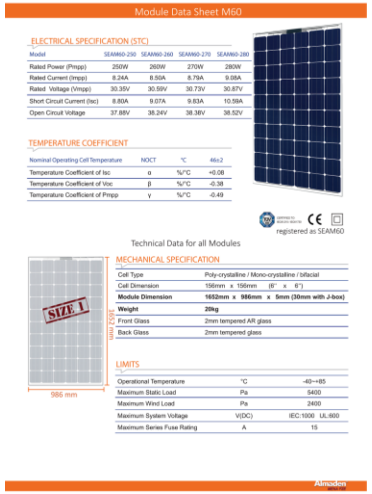


MESSAGEix: Mathematical notation (GAMS)

Sets:

- List of elements for building a model
- Example: technology, commodity, node, emission

set	technology	commodity
members	Solar PV, pipeline	gas, heat, steel,



Parameters:

- Defining quantities (specification), *e.g., lifetime, efficiency, costs*
- Defining relationships between sets, *e.g., input and output of technology*
- Defining constraints, *e.g., bounds and growth rates*

EQUATIONS:

- Relationship between sets, parameters, VARIABLES, etc.
- Building the model

VARIABLES:

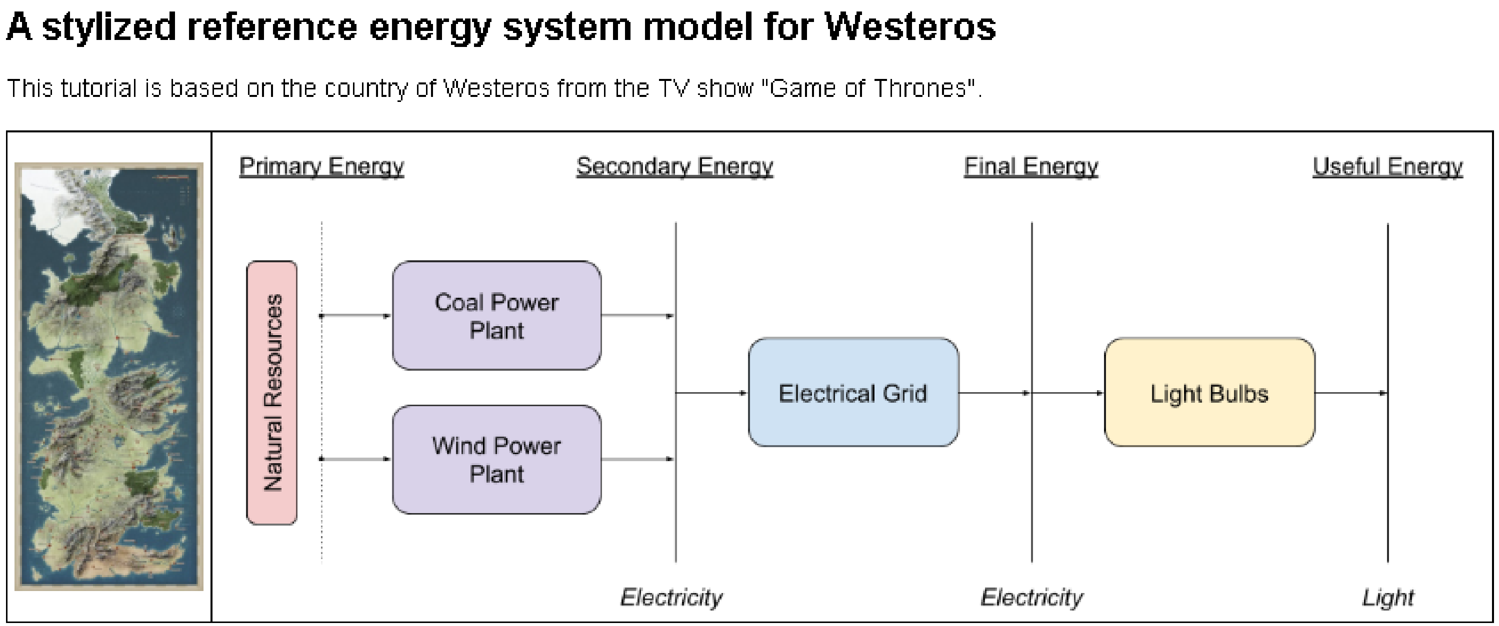
- decision variables to meet the objective and constraints, e.g., ACT, CAP

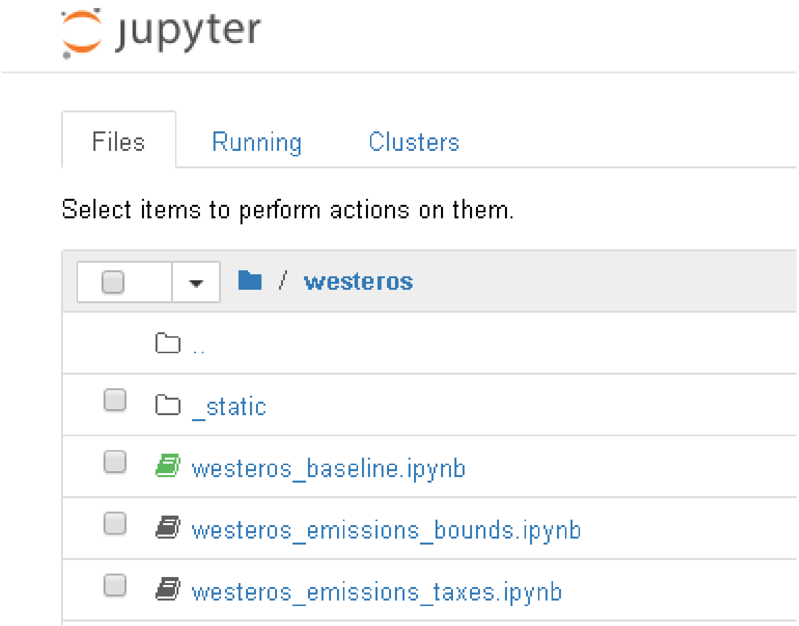
Images from: https://edyeazul.blogspot.com/ (above) and https://weburbanist.com/wp-content/uploads/2015/11/arc-kit-2.jpg (below)

Working with tutorials

Building an energy system from scratch

- Locate your tutorial folder in your machine
- Then, open a command window and call ***jupyter notebook***
- Navigate to the folder for Westeros tutorials and open the baseline





Building a MESSAGEix model

Different steps of modeling

- Creating a new scenario (or loading an existing one)
- Declaring required sets (*node, technology, commodity, level*, etc.)
- Defining required parameters (adding numeric data, relating sets to each other, etc.)

  - demand

  - techno-economic parameters (lifetime, efficiency, investment cost, O&M cost, etc.)

  - bounds and dynamic constraints (growth rates, diffusion rates of technologies)
- Solving the model
- Postprocessing and plotting

Building a MESSAGEix model (2)

Minimum information for building a model

- Sets: *technology*, *node*, *commodity*, *level*, *mode* (of operation), *year*, *time* (sub-annual timesteps)

➔ Some sets have default values in the model (for example, sub-annual timesteps *time*: ‘year’)

- Parameters

  - ***demand***

  - ***output*** **(of technologies)**

➔ In MESSAGEix efficiency of technologies is defined with two parameters: *input, output*

eff = output/input

Working with MESSAGEix scenarios

A short note on model/scenarios

- Loading the ixmp platform (connection to the database):

```
mp = ixmp.Platform()
```

- Creating a new scenario:

```
my_scen = message_ix.Scenario(mp, model, scenario, version='new')
```

Python Class

Modeling platform

Your model/scenario identifiers

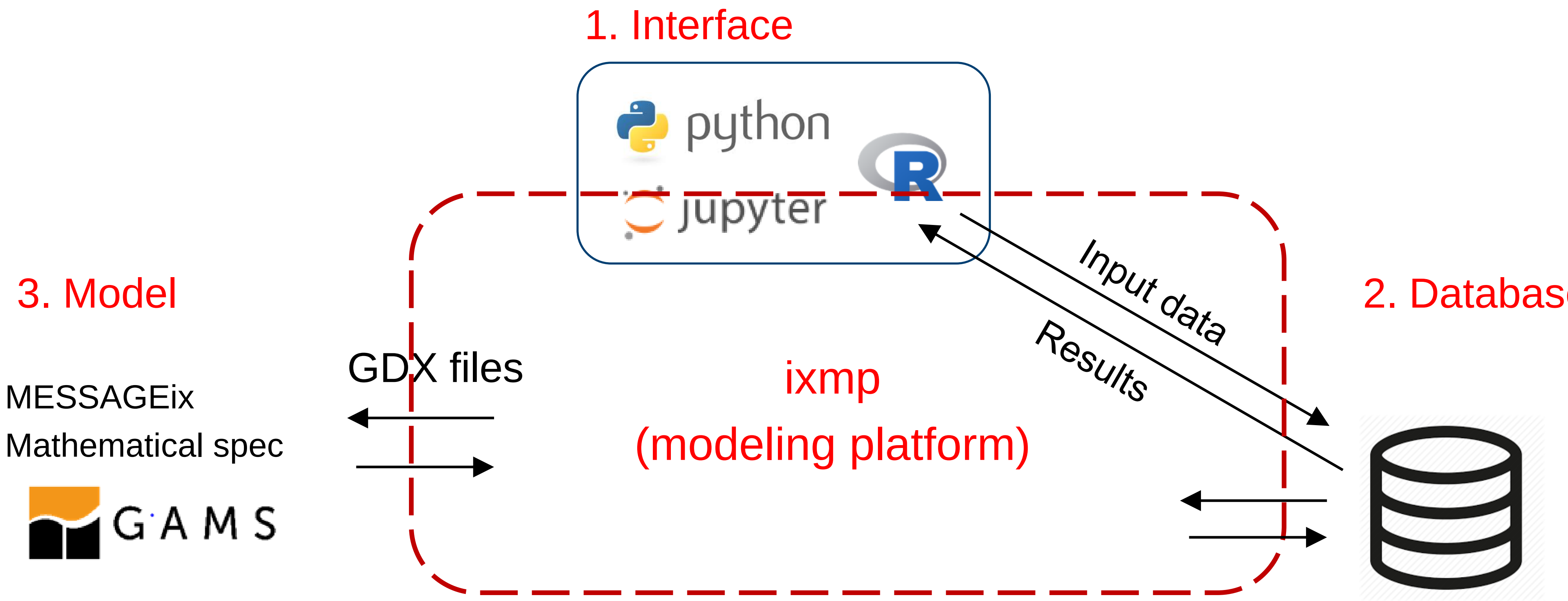
Example: `model = 'building energy system', scenario = 'baseline' (or 'low efficiency')`

- Loading an existing scenario:

```
my_scen = message_ix.Scenario(mp, model, scenario, version=1)
my_scen = message_ix.Scenario(mp, model, scenario)
```

The MESSAGE$_{ix}$ framework : Workflow of modeling

Recap…

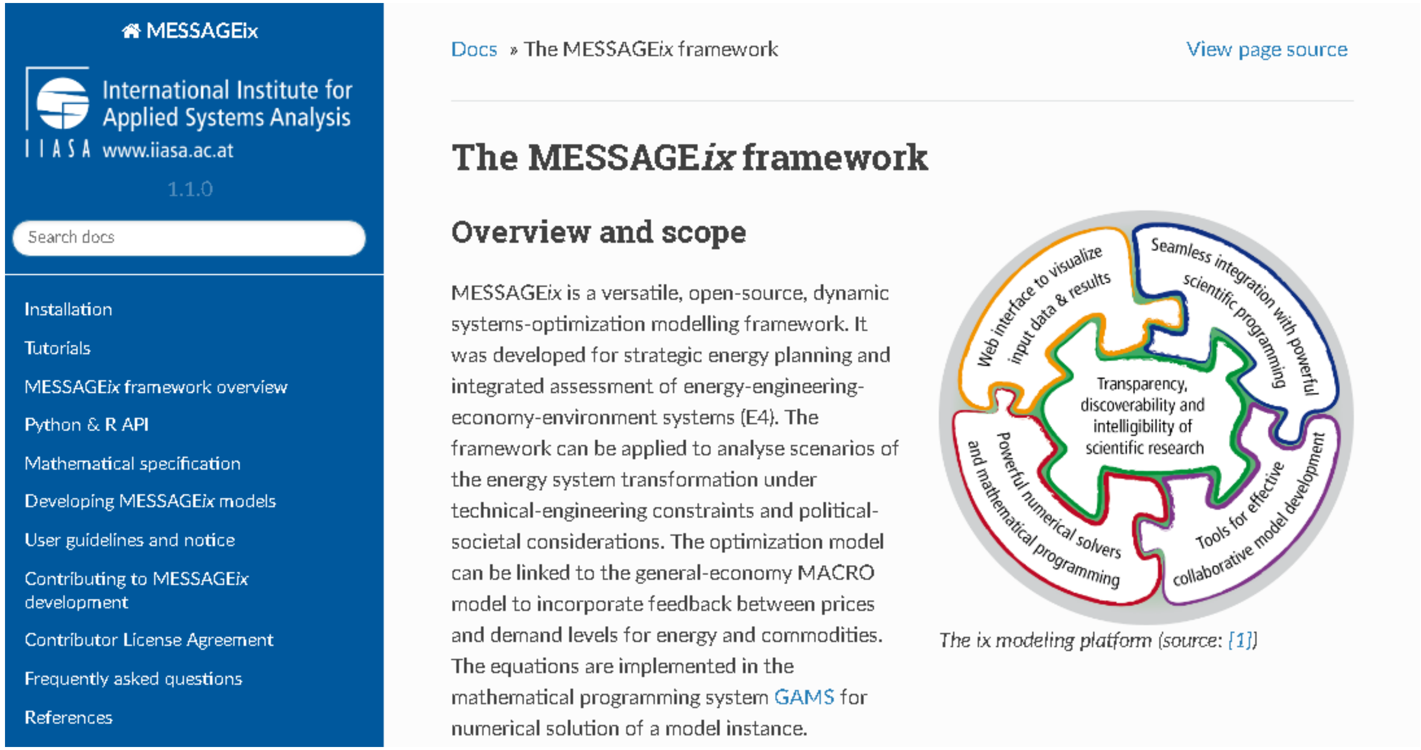


MESSAGEix Website: main source of information

- Main page:
  ⇒ https://MESSAGE.iiasa.ac.at

- Open-source GitHub repository:
  ⇒ https://github.com/iiasa/message_ix
  (contribution guide)

- Tutorials and examples



Thank you very much for your attention!

Dr. Behnam Zakeri

Research Scholar – Energy Program

International Institute for Applied Systems Analysis (IIASA)
Laxenburg, Austria

zakeri@iiasa.ac.at