International Institute for
Applied Systems Analysis
IIASA www.iiasa.ac.at

TU WIEN

Lecture 4:
Developing your own energy system scenarios

Open-Source Energy System Modeling
TU Wien, VU 370.062

Dipl.-Ing. Dr. Daniel Huppmann

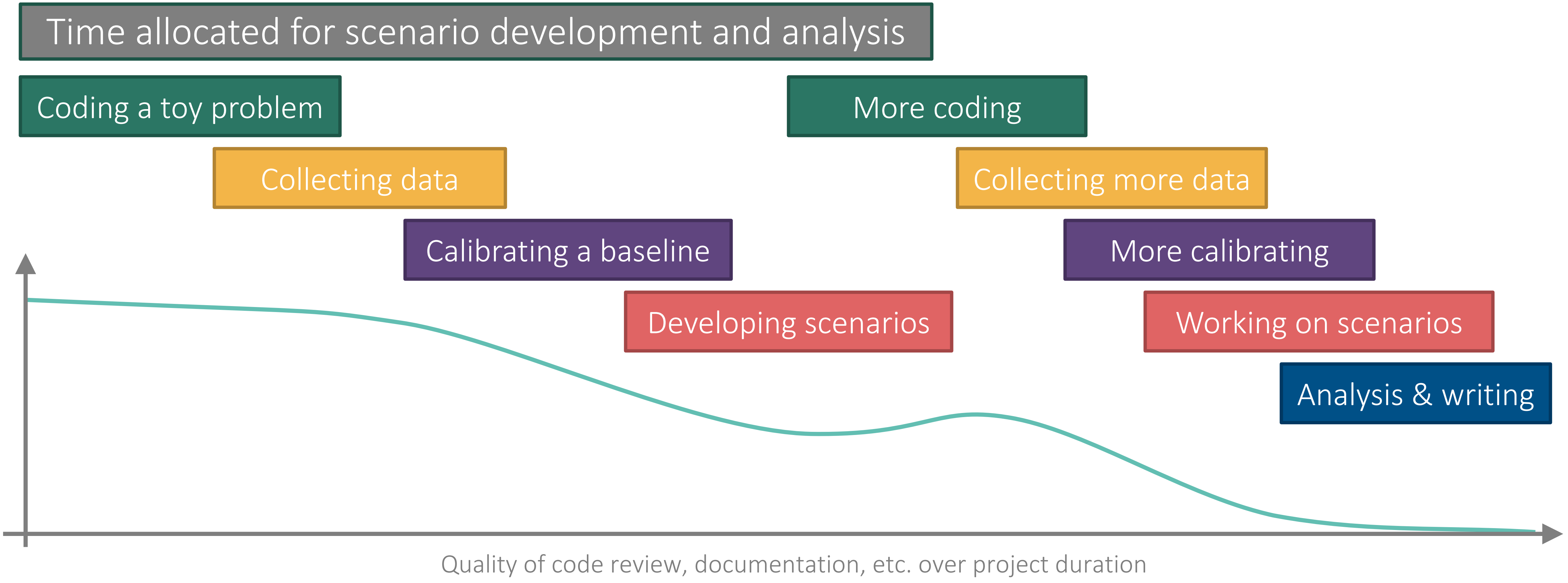
Before we get started...

What's a "model"?

- An attempt at a definition (in the context of energy systems):
  - ⇒ A stylized representation of reality
  - ⇒ Clear definition of the system boundaries
  - ⇒ Based on a mathematical description
  - ⇒ Parametrized and solved numerically
- In practice, the terms **model** & **scenario** are used for several of the items below:
  - ⇒ Mathematical formulation – "just the **equations**"
  - ⇒ Scientific software implementing the equations (but without data) – **modelling framework**
  - ⇒ A **model** implemented in a modelling framework including full "baseline" parametrization
  - ⇒ A **scenario design** or **scenario protocol** is a narrative and parametrization of assumptions possibly relative to the baseline
  - ⇒ A **scenario** is an implementation of a scenario protocol in a model

Introduction: a typical modelling project

Open-source tools (can) increase the efficiency of modelling, scenario development, analysis, and writing



Problems with open-source scientific software

There are many concerns that open-source projects deliver sub-par quality compared to closed-source tools

List of drawbacks:

- …?
- …?
- …?

⇒ It’s just a question of committed resources…

⇒ Overall, the downsides & risks are (pretty much) the same as a close-source (commercial or academic) project

Actual issues of open-source scientific software

If the quality of open-source projects depends on resources, how do we make sure that projects get adequate support?

A few ideas on how to improve collaboration:

- ⇒ Make open-source required by funding agencies
- ⇒ Change the expectation in the community
- ⇒ Look around for existing projects rather than start from scratch…

Challenges

- ⇒ In particular for early-career researchers,
  how to get recognition for contributions to other projects?
- ⇒ Open-source doesn't mean high-quality scientific software

Rationale for best-practice scientific programming

Following best-practice principles in your work
will give you more time to do better research

Modelling and scientific analysis is usually a “constant prototyping” exercise

⇒ ”Just adding one more feature” often breaks existing functionality

⇒ Dependencies (open-source packages) change over time

⇒ Models and tools are too complex to immediately notice changed behaviour

Who has not yet experienced the panic & stress
from a model not solving shortly before a deadline…?

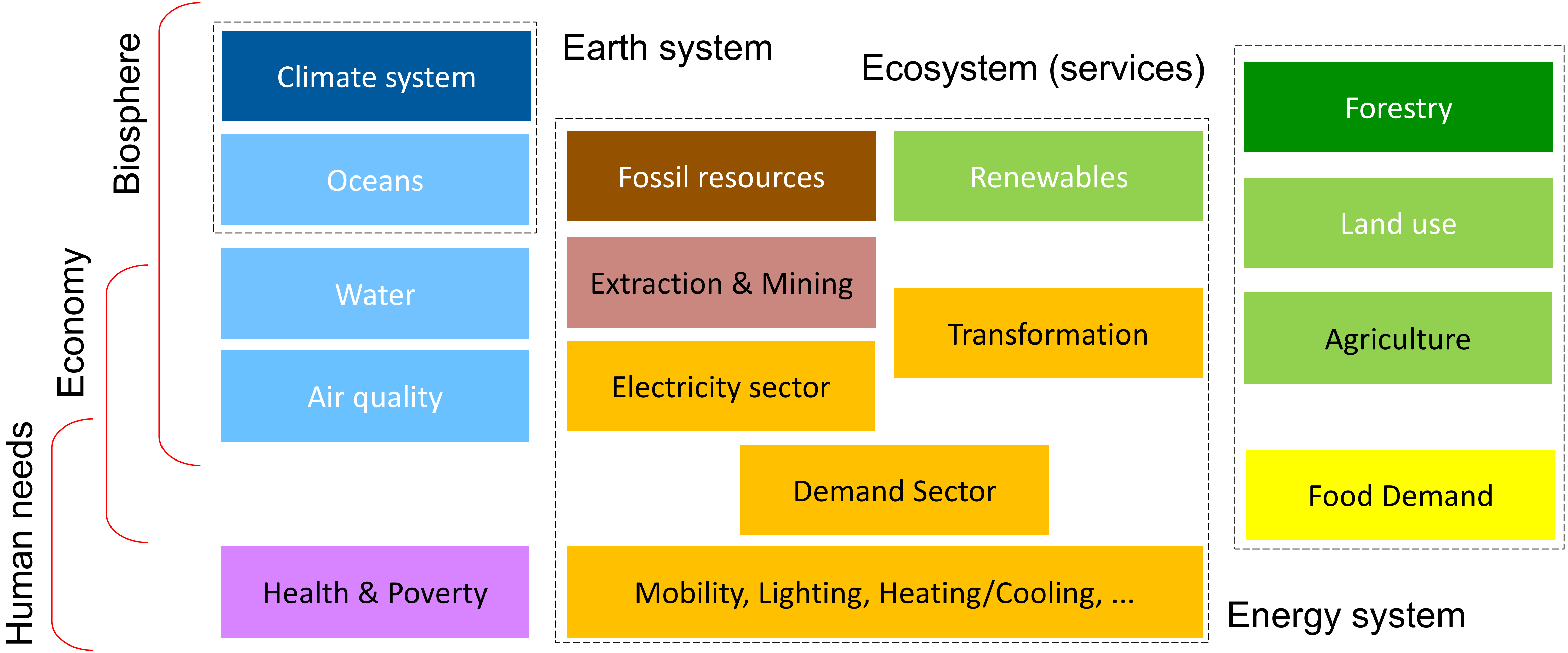
Following best-practice principles…

⇒ Guards against models and tools failing to work (as expected)

⇒ Helps you to understand *your own thinking* a few months later

Some practical considerations for starting model development

Make a conscious choice concerning the system boundaries of your work



Relevant open-source energy modelling frameworks

There are numerous well-maintained options – don't start from scratch…

- OSeMOSYS: https://osemosys.org
- GENeSYS-MOD: http://www.osemosys.org/genesys-mod.html
- MESSAGEix: https://docs.messageix.org
- PyPSA: https://pypsa.org
- Calliope: https://callio.pe
- Spine: https://www.spine-model.org

All of these frameworks have tutorials, examples, active user support via a forum, …

Please don't start a new model!

Part 2

A high-level overview of the open-source energy system model MESSAGE$_{ix}$

The MESSAGE$_{ix}$ framework: Goals and Vision

An integrated modeling platform for x-cutting analysis

An effort started in 2016 – and still ongoing...

Goal: Develop a platform for streamlined modeling

⇒ using state-of-the-art tools for data processing,

⇒ building versatile & powerful mathematical models,

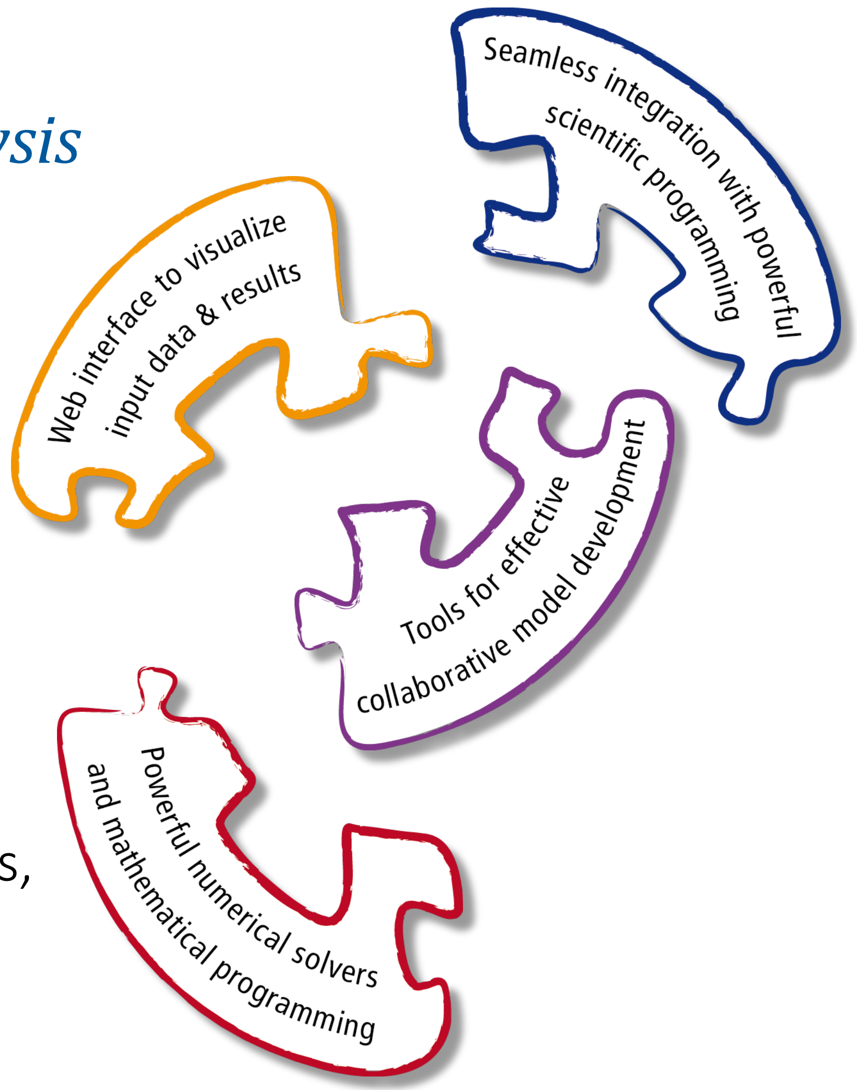
⇒ applying best practice of collaborative research

Vision: Facilitate integration of models & scientific analysis

... between different disciplines and fields including economics, engineering, geophysical, and social sciences

... across spatial and temporal levels of disaggregation

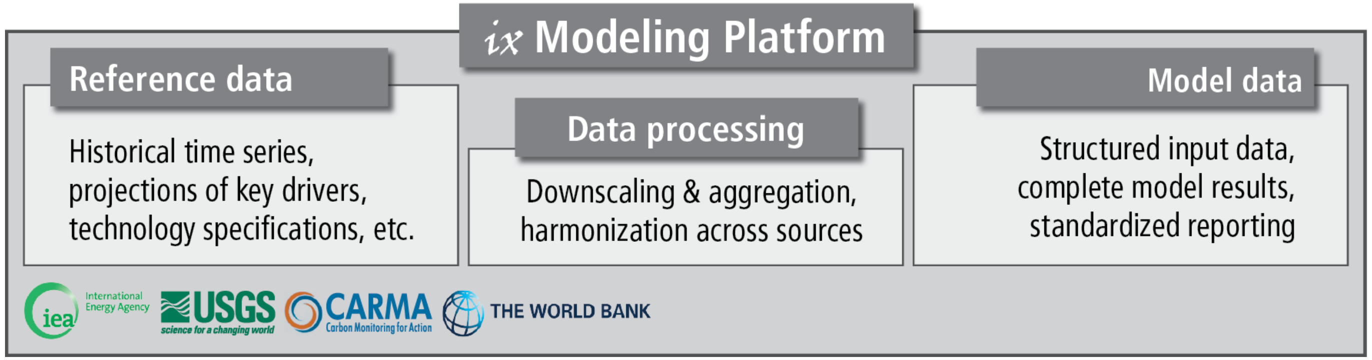
... while guaranteeing the highest level of transparency and scientific reproducibility for a wide audience



Key features of the *ix* modeling platform

The MESSAGE$_{ix}$ framework: Data management

A central data management warehouse



Good data management is crucial for modeling & scientific analysis:

… version-controlled and traceable input data for model development

… reference data for calibration and verification

… efficient workflows based on standardized data processing tools and a common data interface

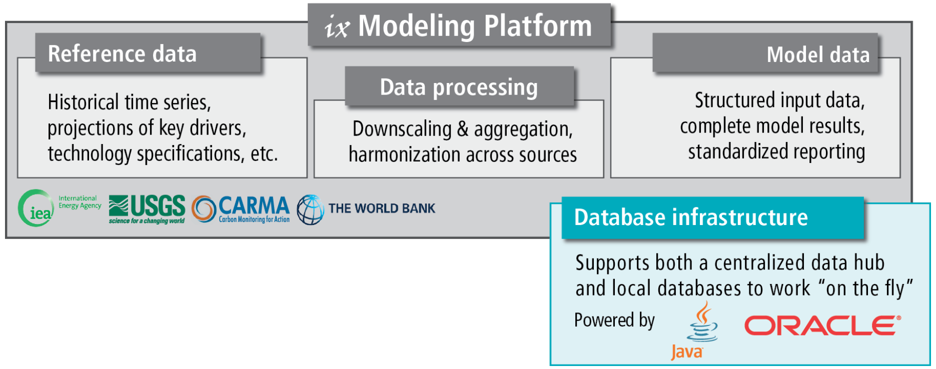
The MESSAGE$_{ix}$ framework: Database backend

Supported by a high-performance database architecture

The platform...

... is based on a Java interface as gateway to the data

... supports both an ORACLE database backend for high-performance, collaborative modeling and local, file-based databases for getting started or working "on the fly"

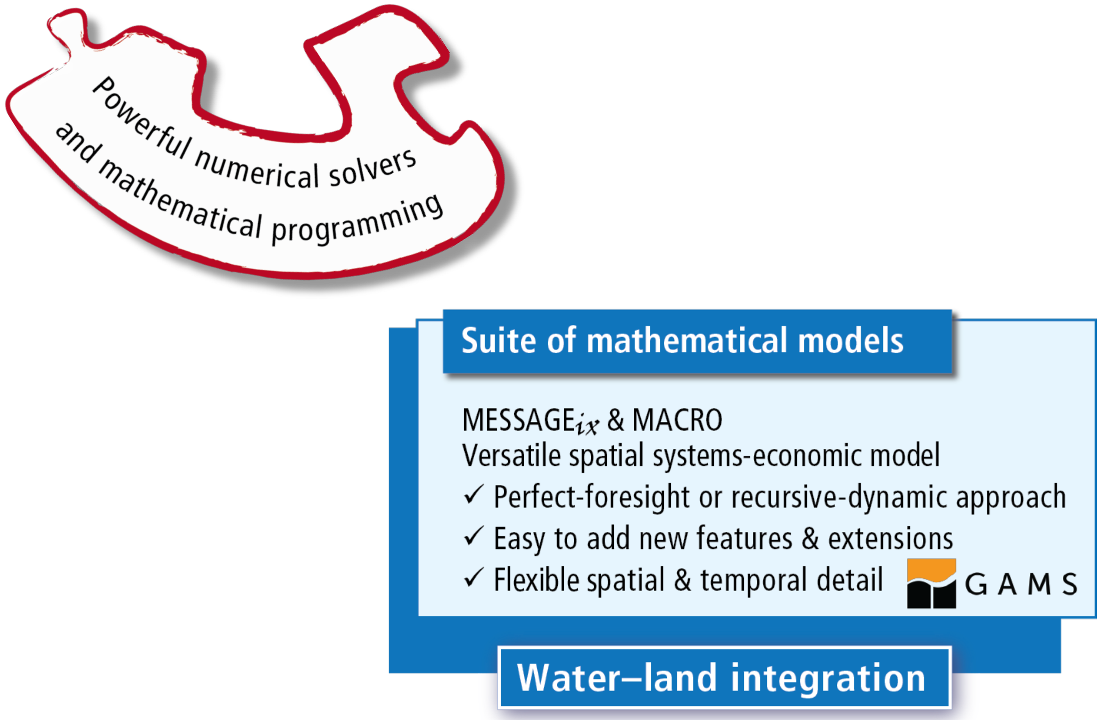


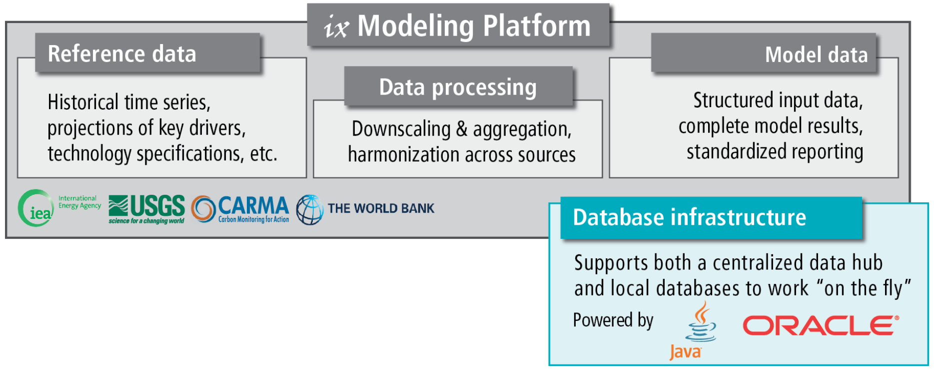
The MESSAGE$_{ix}$ framework: Integration with GAMS

Connected to high-performance numerical programming

The platform has an interface to GAMS, a versatile software for mathematical programming and optimization.

⇒ MESSAGE$_{ix}$ is the first model fully integrated with the *ix* modeling platform...





The MESSAGE$_{ix}$ framework : Scientific programming

Interfaces to scientific programming for advanced users

```
In [1]: import ixmp

In [2]: # launch the IX modeling platform
        using the local default database
        mp = ixmp.Pl

In [3]: model = "Au
        scen = "bas
        annot = "st

        scenario =

        annotation=
             scheme=

In [4]: horizon = r
        firstyear =

In [5]: scenario .a
        scenario .a
        "firstmodel

In [6]: country = "
        ds.add_set(
```

python

```
#-----------------------------------------
# load package

require('Rixmp')

# launch the IX modeling platform
mp <- Platform()

#-----------------------------------------
# specify the model and scenario name

model <- "canning problem"
scen  <- "standard"

#-----------------------------------------
# load a datastructure from the database

scenario <- mp$Scenario(model, scen)

#-----------------------------------------
# retrieve the demand as a dataframe

demand <- scenario$par("demand")
```

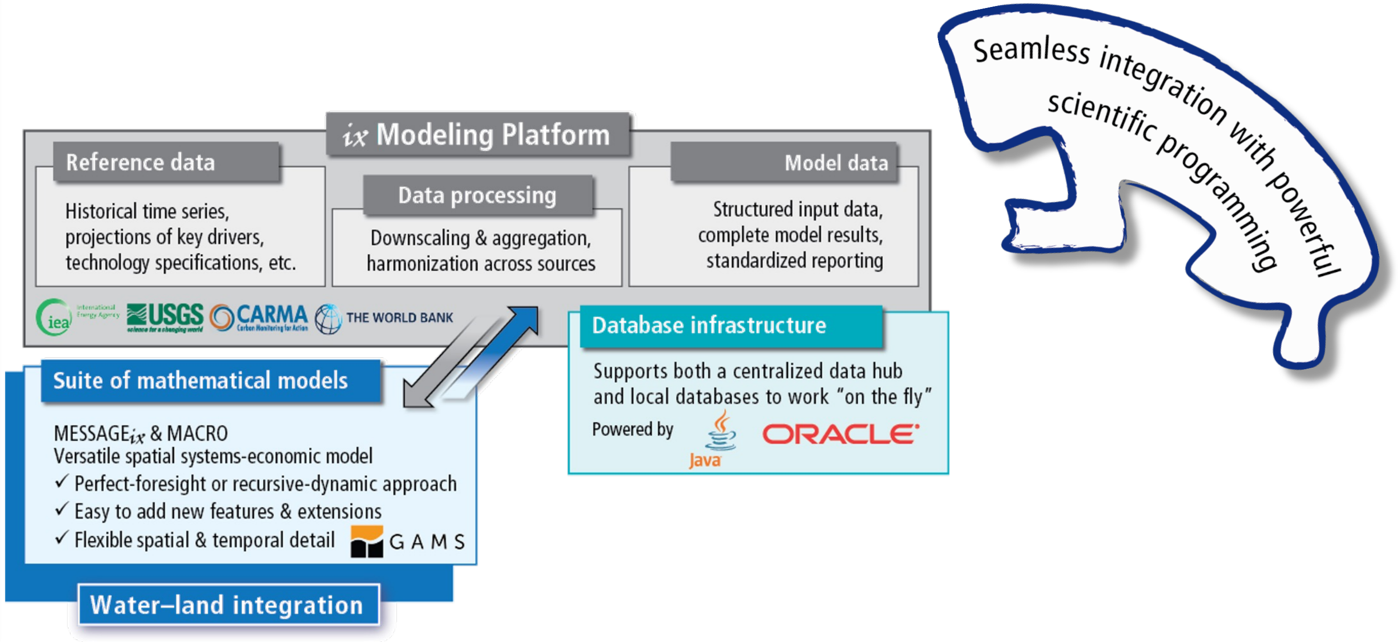
Scientific programming API

Seamless integration with powerful, open and flexible scientific programming languages

✓ Efficient implementation of workflows

✓ Standardized interface for data processing

python™ R



The MESSAGEix framework: Collaborate research

Geared towards best-practice in collaborative research

The platform facilitates collaborative model development

... through comprehensive data version control

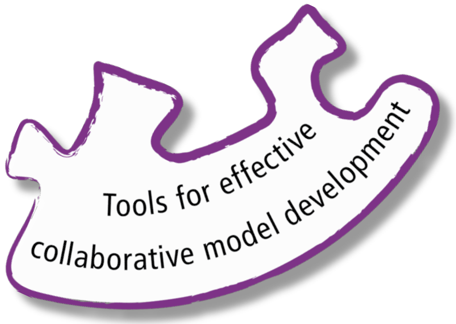
... by moving to “script-based” data processing & analysis

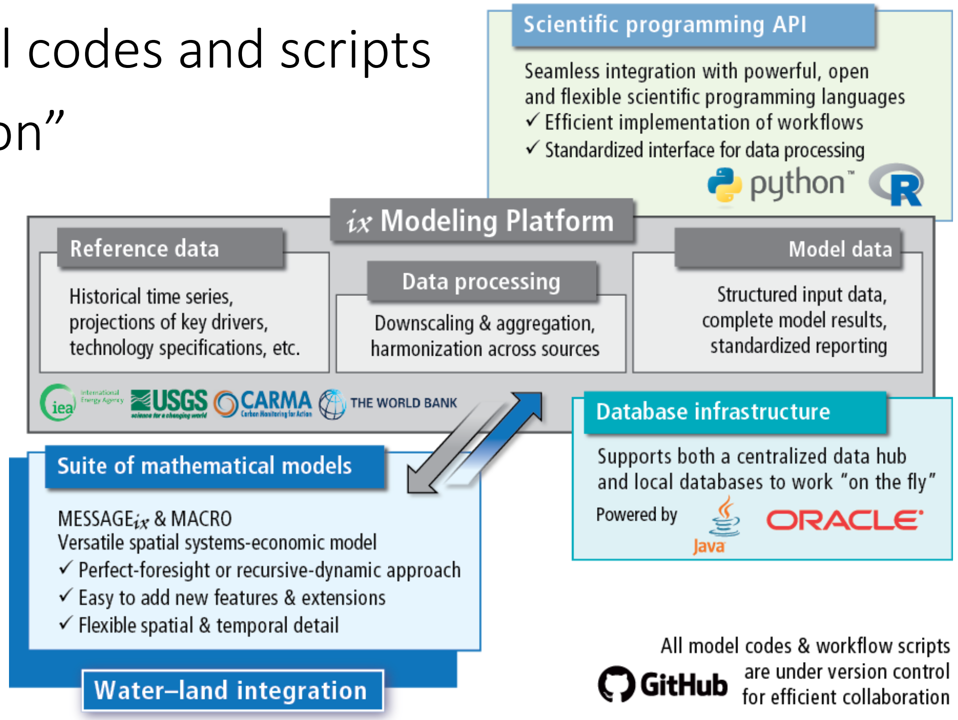
... using full version control of all model codes and scripts

... implementing “continuous integration”

⇒ automated testing of new features to ensure stable code base

GitHub





The MESSAGE$_{ix}$ framework: Documentation

Implementing tools for comprehensive documentation

The framework ensures transparency and intelligibility through "auto-documentation" of all codes & packages on readthedocs.org

⇒ Documentation of all scientific programming packages using 'sphinx'

⇒ Documentation of the mathematical equations generated automatically from LaTeX mark-up in the GAMS code

GAMS

```
*----------------------------------------------------------------------------------------------------
***
* Technology section
* ------------------
*
* Technical and engineering constraints
* ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
*
* Equation CAPACITY_CONSTRAINT
* """"""""""""""""""""""""""""
* This constraint ensures that the actual activity of a technology at a node/time cannot exceed available (maintained)
* capacity summed over all vintages, including the technology capacity factor :math:`capacity\_factor_{n,t,y,t}`.
*
*  .. math::
*     \sum_{m} ACT_{n,t,y^V,y,m,h}
*         \leq duration^H_{h} \cdot capacity\_factor_{n,t,y^V,y,h} \cdot CAP_{n,t,y^V,y}
*         \quad t \ \in \ T^{INV}
*
* where :math:`T^{INV} \subseteq T` is the set of all technologies
* for which investment decisions and capacity constraints are relevant.
***
CAPACITY_CONSTRAINT(node,inv_tec,vintage,year,time)$( map_tec_time(node,inv_tec,year,time)
        AND map_tec_lifetime(node,inv_tec,vintage,year) )..
    sum(mode$( map_tec_act(node,inv_tec,year,mode,time) ), ACT(node,inv_tec,vintage,year,mode,time) )
        =L= duration_time(time) * capacity_factor(node,inv_tec,vintage,year,time) * CAP(node,inv_tec,vintage,year) ;
```

Scientific programming API

Read the Docs

International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

master

Search docs

- Installation
- Tutorials
- MESSAGEix framework overview
- Python & R API
- Mathematical specification
  - Sets and mappings definition
  - Parameter definition
  - Mathematical formulation (core model)
    - Notation declaration
    - Objective function
    - Regional system cost accounting function
    - Resource and commodity section
    - Technology section
      - Technical and engineering constraints
      - Constraints representing renewable integration
      - Constraints for addon

Read the Docs   v: master

Equation STOCKS_BALANCE

This constraint ensures the inter-temporal balance of commodity stocks. The parameter $commodity\_stocks_{n,c,l}$ can be used to model exogenous additions to the stock

$$STOCK_{n,c,l,y} + commodity\_stock_{n,c,l,y} = duration\_period_y \cdot \sum_h STOCK\_CHG_{n,c,l,y,h} + STOCK_{n,c,l,y+1}$$

Technology section

Technical and engineering constraints

The first set of constraints concern technologies that have explicit investment decisions and where installed/maintained capacity is relevant for operational decisions. The set where $T^{INV} \subseteq T$ is the set of all these technologies.

Equation CAPACITY_CONSTRAINT

This constraint ensures that the actual activity of a technology at a node cannot exceed available (maintained) capacity summed over all vintages, including the technology capacity factor $capacity\_factor_{n,t,y,t}$.

$$\sum_m ACT_{n,t,y^V,y,m,h} \leq duration\_time_h \cdot capacity\_factor_{n,t,y^V,y,h} \cdot CAP_{n,t,y^V,y} \quad \forall t \in T^{INV}$$

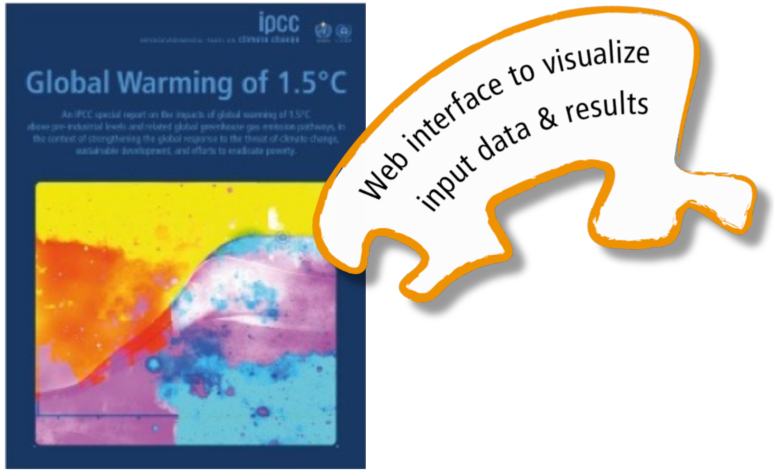
Equation CAPACITY_MAINTENANCE_HIST

The following three constraints implement technology capacity maintenance over time to allow early retirement. The optimization problem determines the optimal timing of retirement, when fixed operation-and-maintenance costs exceed the benefit in the objective function.

The MESSAGE$_{ix}$ framework: Interactive web user interface

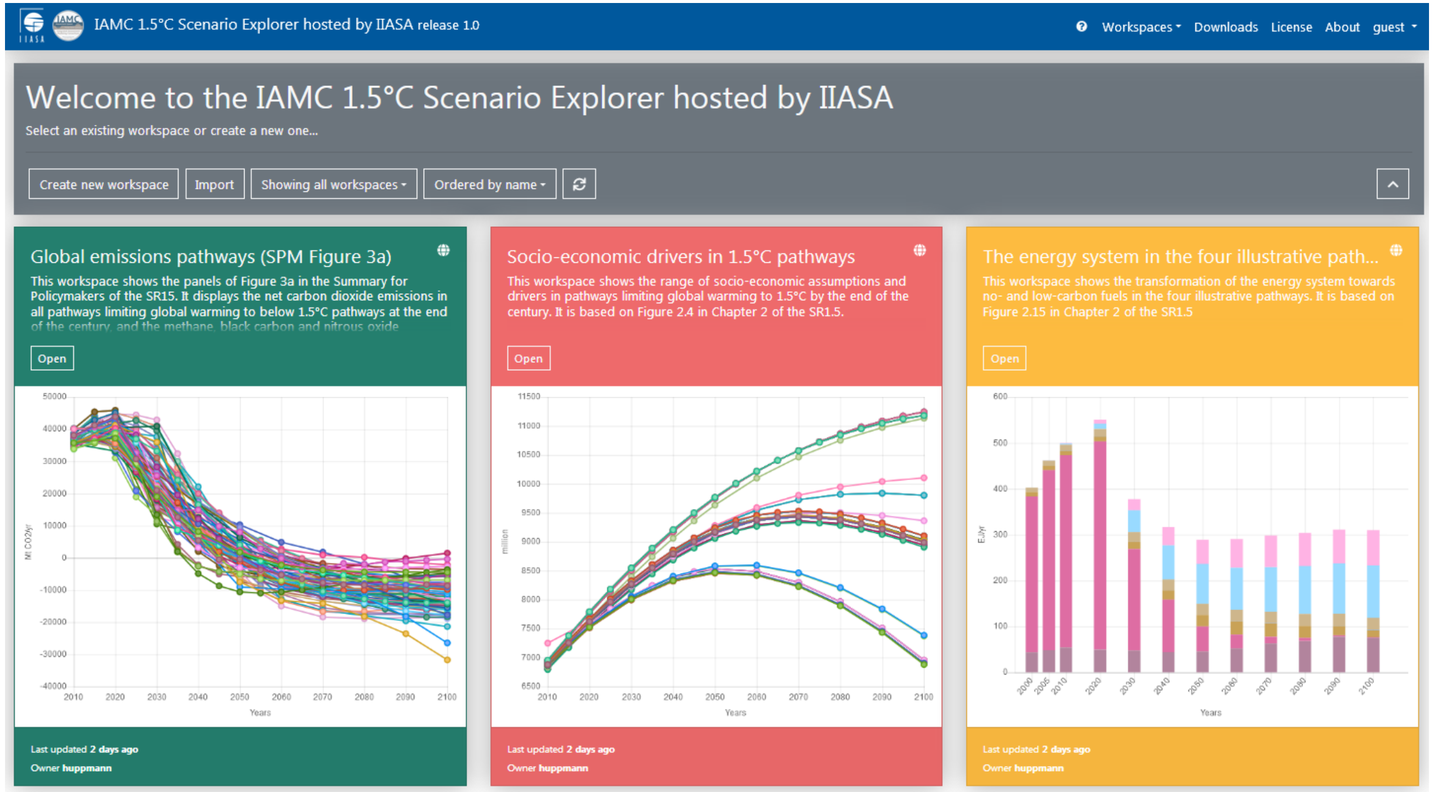
An intuitive gateway to modeling data for researchers and a wider audience

The “IAMC 1.5°C Scenario Explorer” presenting an ensemble of pathways supporting the IPCC SR15 assessment is powered by the web user interface of the *ix* modeling platform



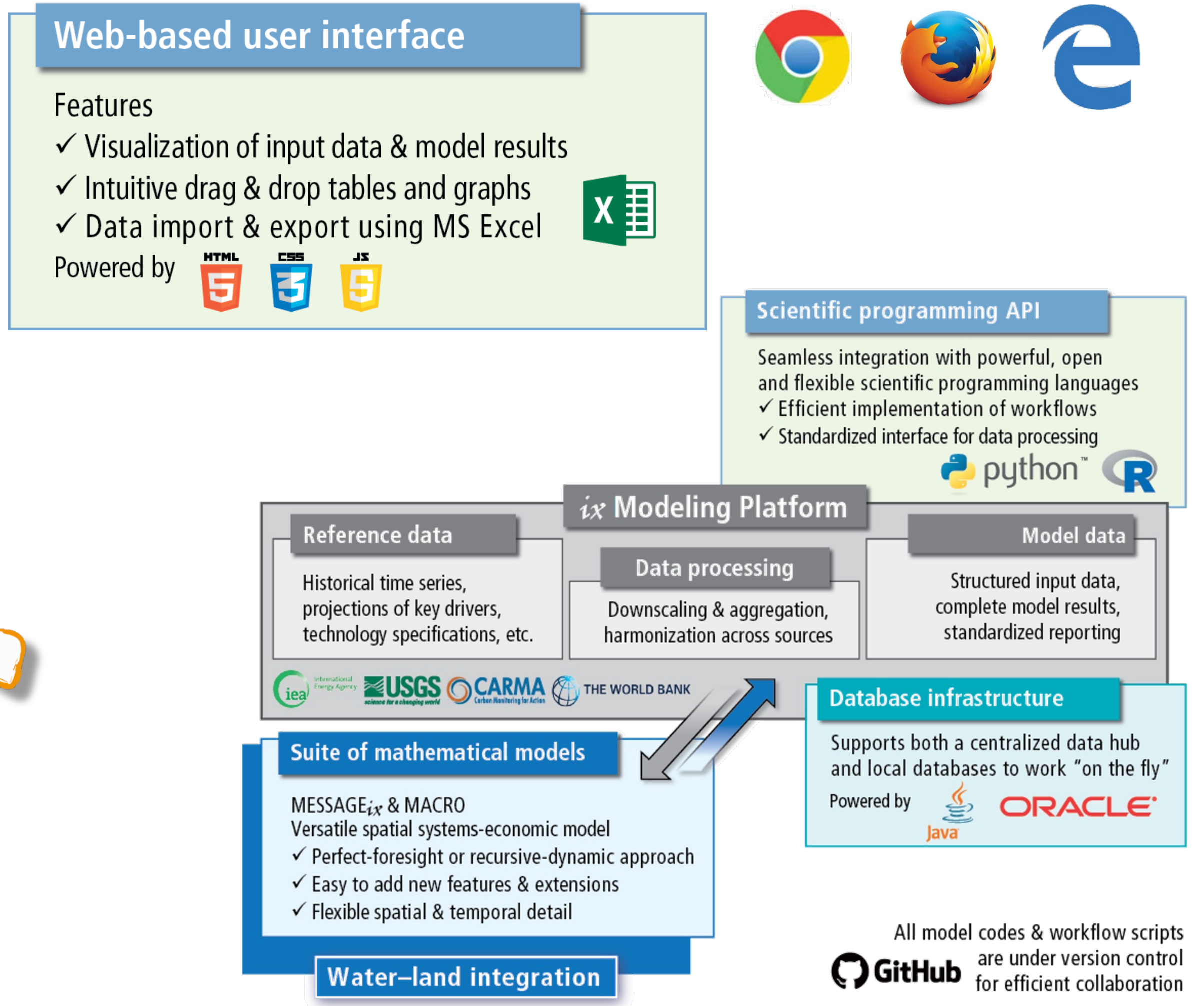
Special Report on *Global Warming of 1.5°C*
(IPCC SR15, http://www.ipcc.ch/report/sr15/)

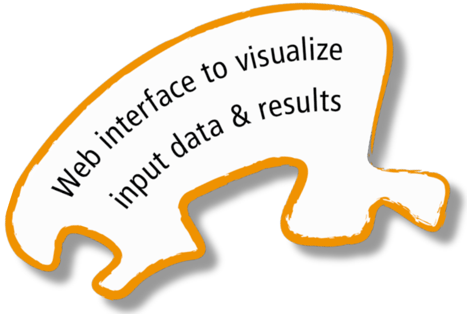
Visit the Scenario Explorer at https://data.ene.iiasa.ac.at/iamc-1.5c-explorer



The MESSAGE$_{ix}$ framework: Interactive web user interface

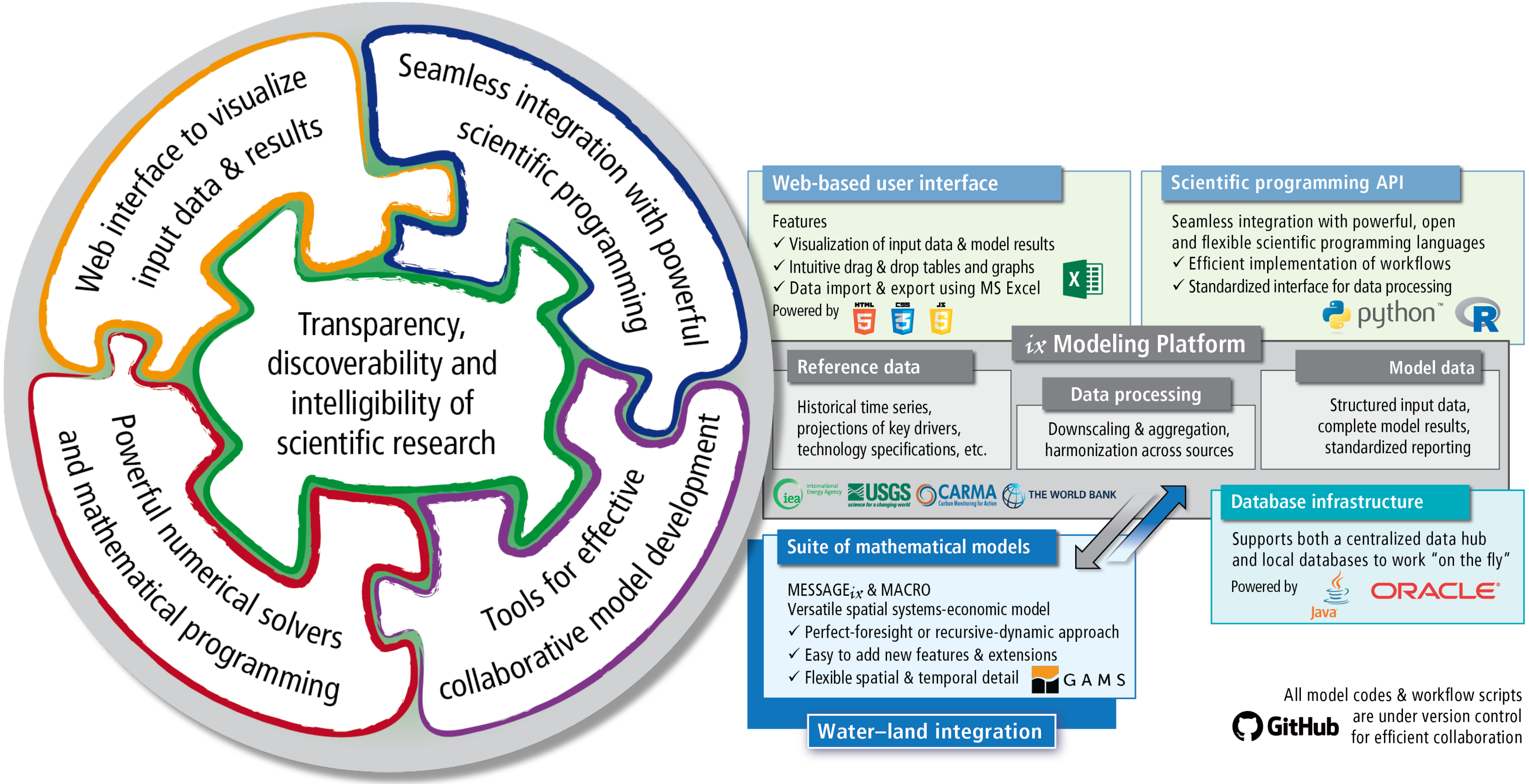
An intuitive gateway to modeling data for researchers and a wider audience





The MESSAGE$_{ix}$ framework

Facilitating transparency and reproducibility of research



Working with the MESSAGE$_{ix}$ framework

Practical considerations where MESSAGE$_{ix}$ differs from other frameworks

Installation:

⇒ When installing public release versions via pip or anaconda, you don't need to worry

⇒ To get the bleeding-edge developments, make sure that you install the corresponding branches from the GitHub repositories `ixmp` and `message_ix`

⇒ Known issue on Mac: `versioneer` is sometimes confused,
delete installation from `site-packages` directory manually if necessary

Your scientific workflow:

⇒ Don't re-run your scenario assessment notebooks over and over again,
because this will create a new scenario instance in the database every time

⇒ Instead, remove the `` `version=new` `` argument to load an existing scenario
and adapt the script accordingly

Working with the MESSAGE$_{ix}$ framework

Practical considerations where MESSAGE$_{ix}$ differs from other frameworks

Integration with GAMS:

⇒ The GAMS code is installed (copied) to the Python `site-packages` directory, so if you make changes in your `git` folder, it won't have any effect on your model run

⇒ This actually makes a lot of stuff simpler for the Python installation (say @gidden and @khaeru)

⇒ But you can set your `git` folder as the model folder (i.e., where the `message_ix` package looks for the MESSAGEix-GAMS code)

using this command line interface (CLI):

```
$ messageix-config --model_path /path/to/model
```

Part 2

How to start developing your own energy system scenarios?

Considerations for developing a new (energy system) model

What do you need to build an energy system

- A “reference energy system” (RES)
  ⇒ The technologies, commodities, levels
- Regional specification
- Time horizon
- Assumptions (projections)
  ⇒ Costs (investment, capacity, variable)
  ⇒ Demand for energy and other commodities
  ⇒ Bounds on trade, diffusion of new technologies, etc.
- Policies on emissions (taxes, bounds) and sustainable development policies

To make learning MESSAGEix more fun, we developed
a suite of tutorials based on the TV show “Game of Thrones”

GAME OF THRONES

Part 3

Some considerations on modelling

More practical considerations for starting model development

Choose an appropriate methodology for the research question at hand

Commonly used methodologies:

⇒ Optimization: determine the system that is optimal according to a metric

⇒ Equilibrium: determine the system as a result of interacting agents

⇒ Simulation: determine the system given some decision rules

Dealing with uncertainty:

⇒ Deterministic optimization (perfect foresight):

all future states (exogenous parameters) are known at the beginning of the model horizon

⇒ Stochastic optimization:

all future states along an "uncertainty tree" are known, including probabilities of each branch

⇒ Myopic (rolling horizon) optimization:

- decisions in period y are taken under some assumptions about the future;
- move to period $y+1$ and repeat, with (possibly altered) assumptions about periods $[y+2, \ldots]$

Yet more practical considerations for starting model development

There are many issues that a self-critical modeller should consider...

- Model uncertainty:

  ⇒ Is the approach appropriate? Are results dependent on the methodology?

- Parameter uncertainty:

  ⇒ How much confidence can you have on input assumptions?

- Model horizon and level of temporal/spatial disaggregation:

  ⇒ What is the intended scope of analysis? Beware of the “end-of-horizon”-effect!

- Model simplifications for numerical tractability and comprehensibility:

  ⇒ What are appropriate trade-offs between having a high level of detail vs. loosing focus?
  E.g., variable renewables require infrastructure for system stability – assumption or result?

- System boundaries and model closure:

  ⇒ Are the assumptions to “close” the model valid?
  E.g., for a national electricity model, you need to make assumptions about import/export

Methods to evaluate the robustness of results

Think hard about testing your model behaviour

Methods for validation:

- Sensitivity analysis:
  Structured variation of key input parameters to understand the impact on results
  ⇒ Relatively easy to do, but you can never do sensitivity assessment for all parameters...
- Multi-criteria analysis:
  Include multiple dimensions in the objective function, solve model with different weights
  ⇒ Requires some work, still prone to modelling artefacts
- "Modelling to generate alternatives"
  Re-solve a model to get a different solution within some additional bounds
  ⇒ Very elegant, but requires substantial effort to implement

  Further reading: Joseph F. DeCarolis. Using modeling to generate alternatives (MGA) to expand our thinking on energy futures. *Energy Economics* 33(2):145-152, 2011. doi: [10.1016/j.eneco.2010.05.002](https://doi.org/10.1016/j.eneco.2010.05.002)

Thank you very much for your attention!

Dr. Daniel Huppmann

Senior Research Scholar – Energy, Climate, and Environment Program

International Institute for Applied Systems Analysis (IIASA)
Schlossplatz 1, A-2361 Laxenburg, Austria

huppmann@iiasa.ac.at

@daniel_huppmann

@daniel_huppmann@mastodon.social

www.iiasa.ac.at/staff/daniel-huppmann