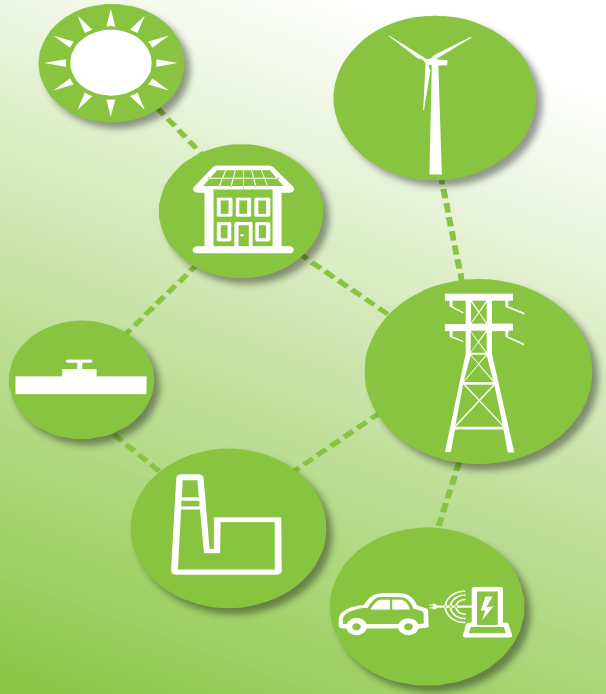
Spine



Managing data and workflow in energy system modelling

Juha Kiviluoma
Senior scientist, VTT Technical Research Centre of Finland
Senior energy systems researcher, University College Dublin

12th Jan. 2021
The Canadian Internet

VTT UCD DUBLIN KU LEUVEN KTH VETENSKAP OCH KONST ENERGY REFORM CONSULTING SERVICES

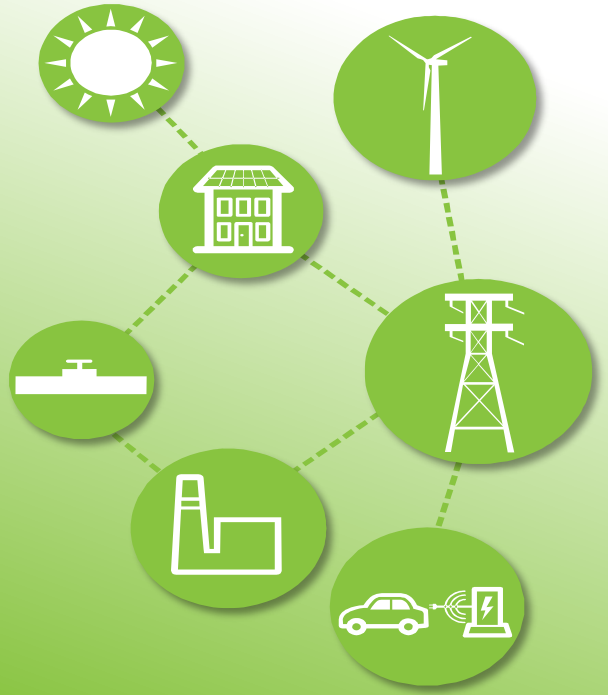
Challenges

- Data curation and data processing
  - Where did this data come from?
  - Who has made this change – and why?
- From an individualistic process to a collaborative workflow
  - ‘Well, I think the data is somewhere in my laptop…’
- Science needs replication
  - Tool specific errors – compare using same data
  - Errors in data – more eyes on the same data set
- Analysing energy systems require multiple scales
  - Temporal: from milliseconds to decades
  - Spatial: from process level to the whole globe

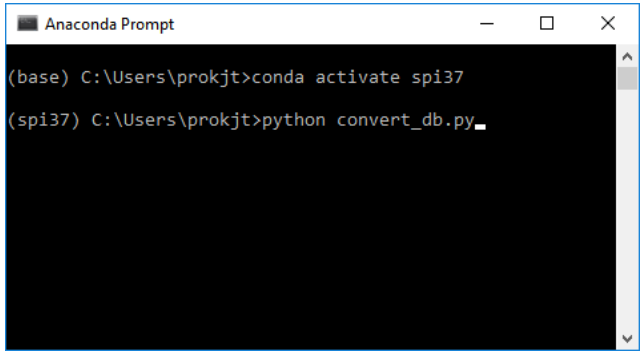
Challenges continue

- Sharing data vs. sharing data processing
  - Confidence in data?
  - Improving the data processing together
- Interaction with the wider audience
  - Visualization, automatic web pages
  - Ability to change assumptions and run

Spine



Solutions

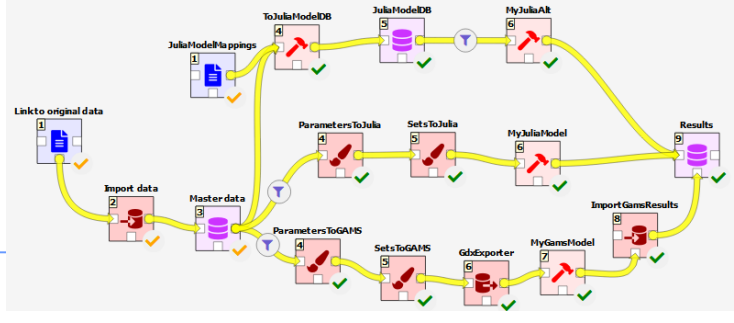


Pieces of code

- Flexibility
- Expert user

Workflow management

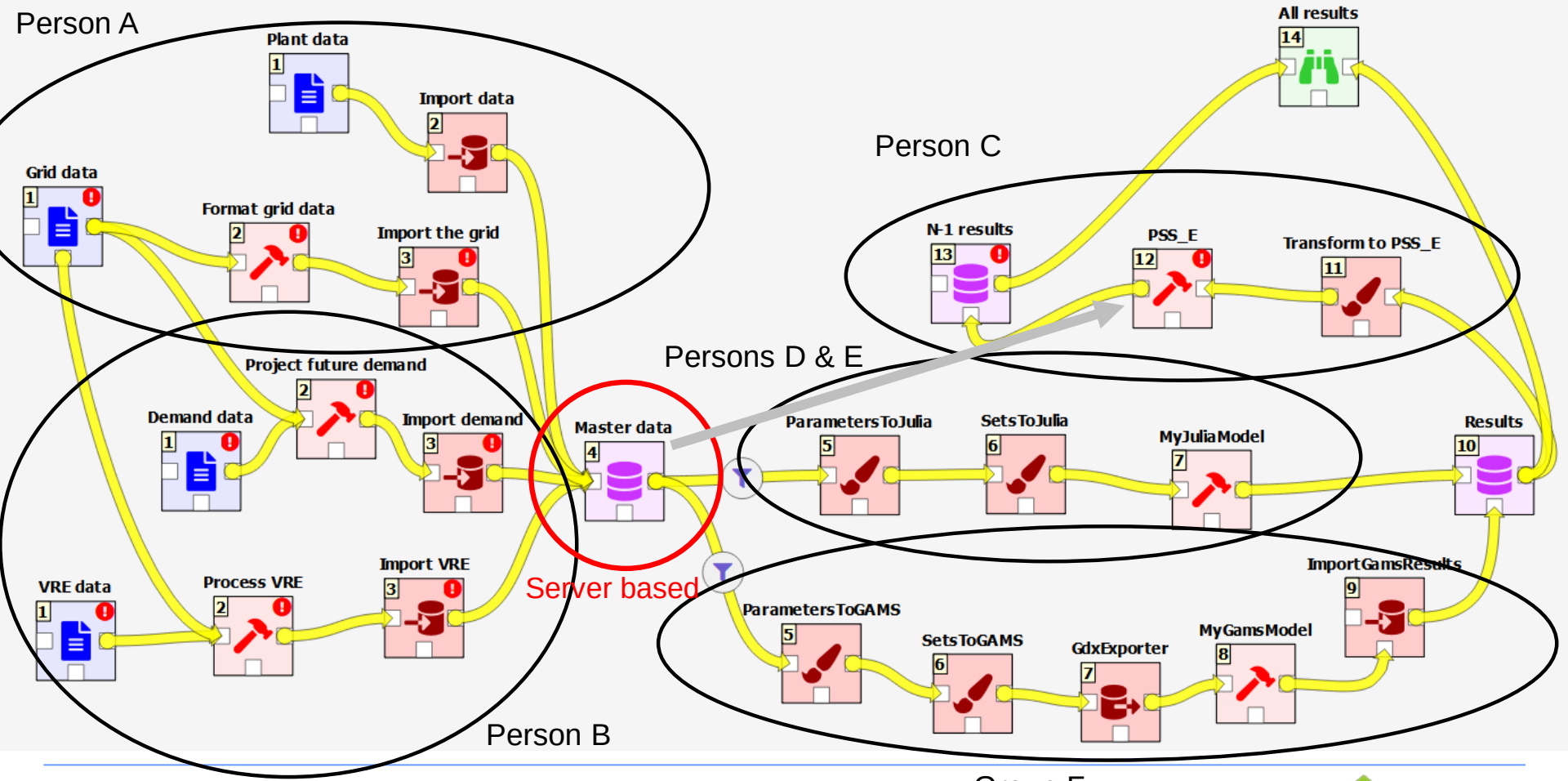
- Easy for regular user
- Flexibility (may require expert skills)



A monolithic program

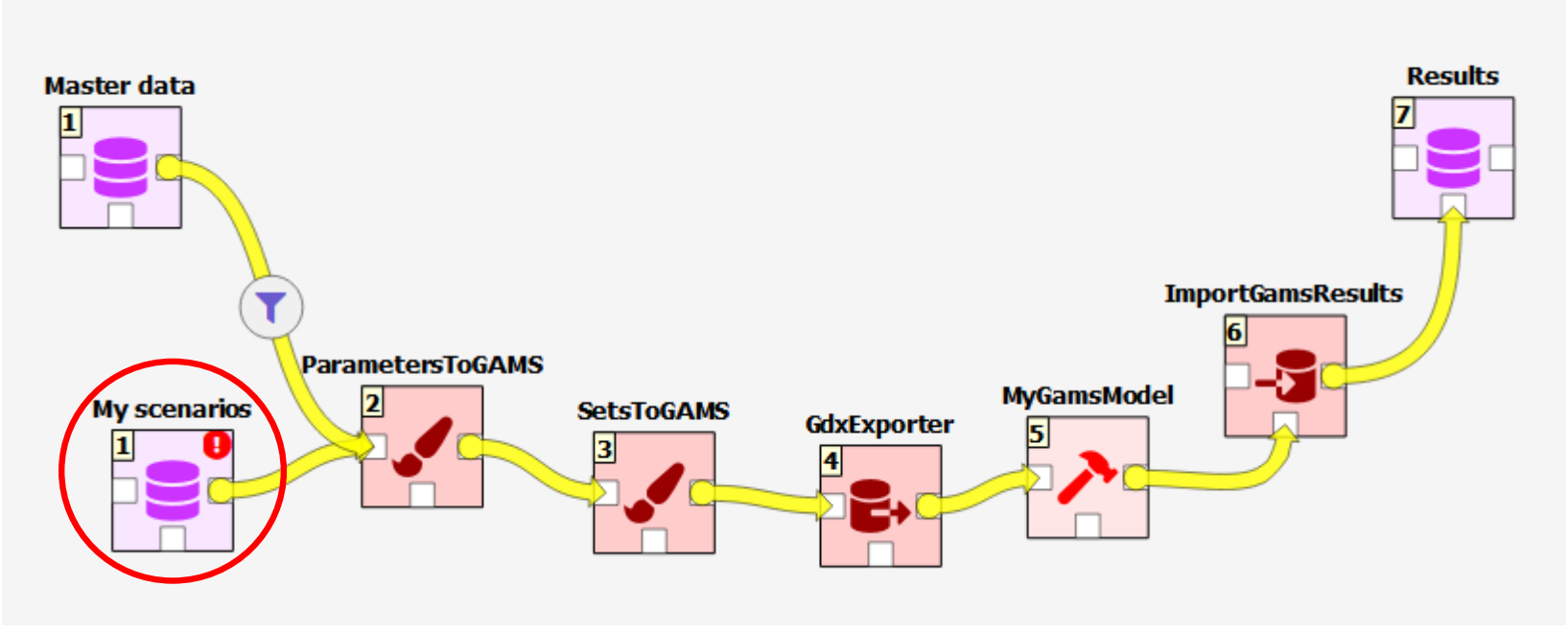
- May be easy to learn
- You're dependent

Spine


Person A
Plant data
1
Import data
2
Grid data
1
Format grid data
2
Import the grid
3
Person C
All results
14
N-1 results
13
PSS_E
12
Transform to PSS_E
11
Persons D & E
Project future demand
2
Demand data
1
Import demand
3
Master data
4
ParametersToJulia
5
SetsToJulia
6
MyJuliaModel
7
Results
10
VRE data
1
Process VRE
2
Import VRE
3
Server based
ParametersToGAMS
5
SetsToGAMS
6
GdxExporter
7
MyGamsModel
8
ImportGamsResults
9
Person B
Group F

Spine

Group F



Plug-ins!

GAMS

Integrated Development Environment

Local program

- It works
- Ownership
- Complex workflows difficult to share

Workflow management

- Runs on your own machine from open source code
- Computation can be outsourced
- Share the workflow
  - Partial projects
  - Version control
  - Server-based databases

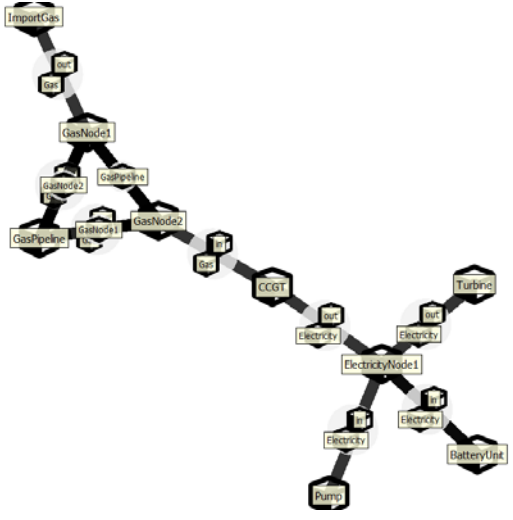
Spine

Cloud application

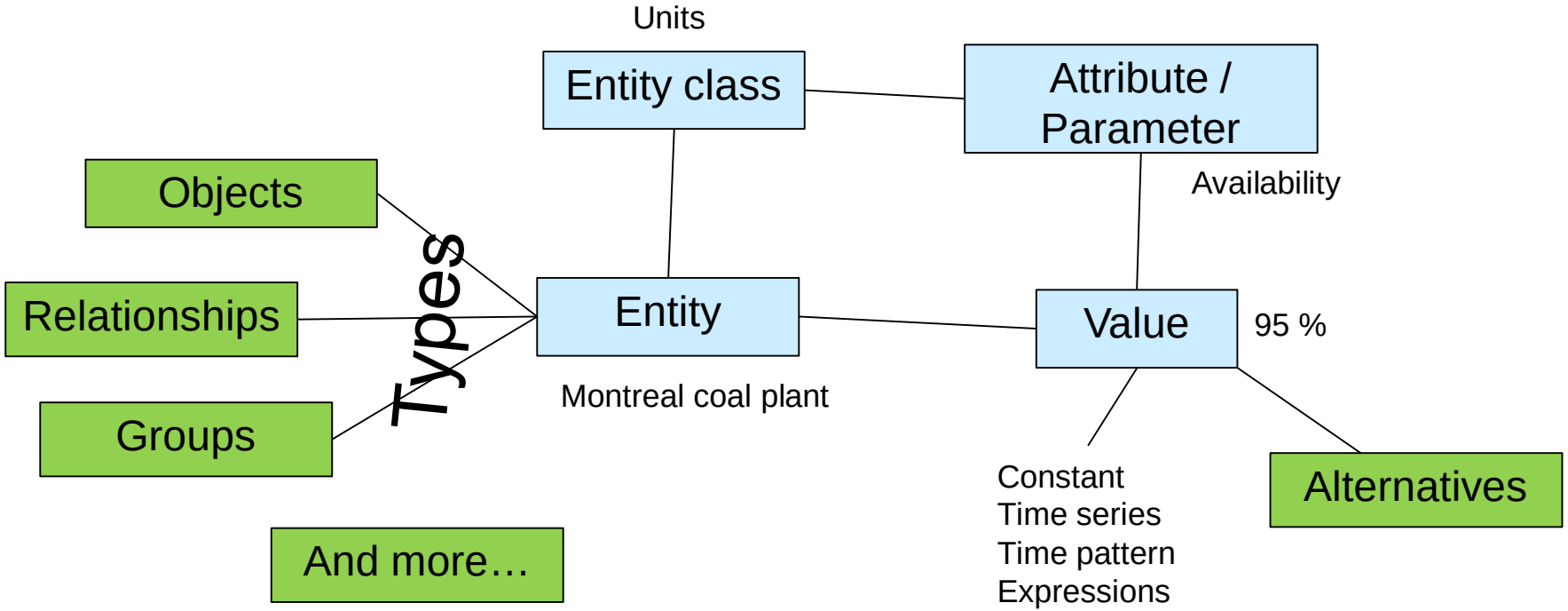
- Outsourcing:
  - Maintenance
  - Computation
  - Ownership
- Allows shared workflow

Data structure should be generic

- The workflow should not care what data goes through
- Needs to include the structure of the data
  - E.g. Power plant B is in location X
  - It's a graph
- The interface can display and manipulate any data without changes to the interface
- The tools can be made generic (when it makes sense)
  - E.g. a missing values replacer



Spine Toolbox generic data structure (EAV with classes and entity types)

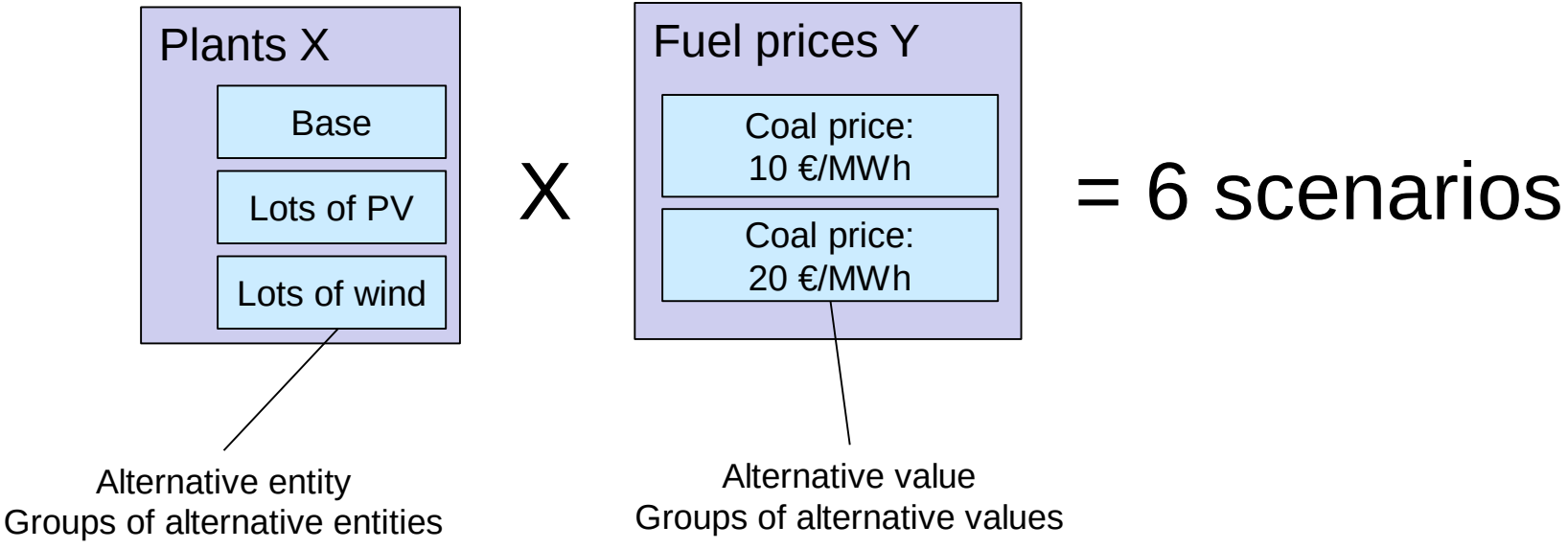


Passing data between tools

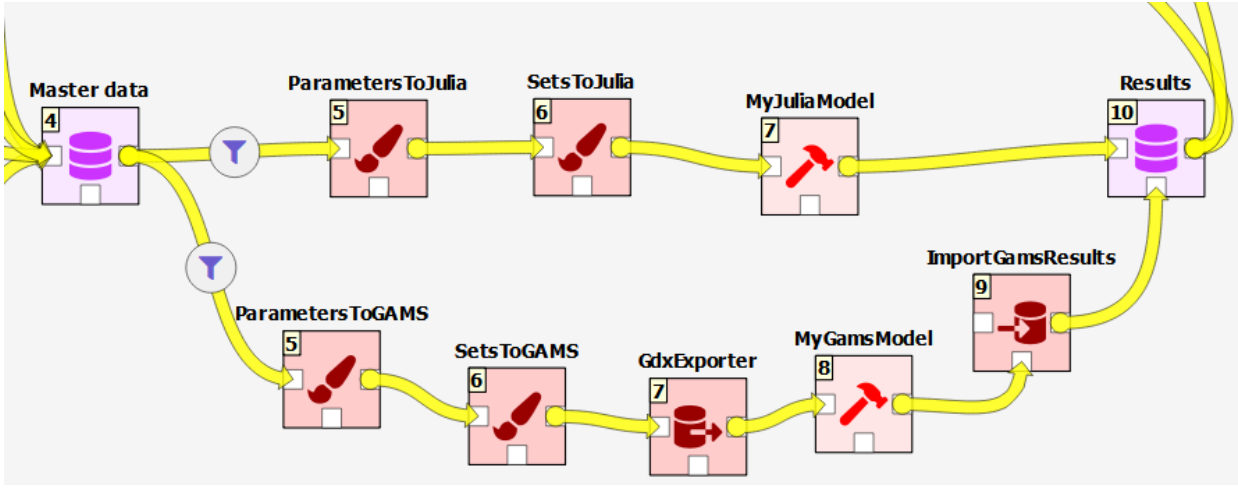
- The generic data structure of Spine in combination with the API to the Spine Database enables a standard way to pass data between tools
- Moving to a standard allows tools to communicate more easily

- Data can also be passed by other means (e.g. files)
- Spine Toolbox includes a generic data importer
  - Excel, CSV, GDX, JSON and more can be added
- Generic data exporter is work in progress

Alternatives, scenarios and recipes



- Models have different data requirements
  - Investment model: less detail, but investment related data too
  - Power system stability model: high grid detail, but only snapshots in time
- Spine Toolbox supports tools, features and methods to filter what is sent to which tool from the same database
  - Work in progress



Nomenclature / ontology

- How do you like to call things?
- How does your different models call things?
- Is there a shared nomenclature available?
- Are there other differences (data form or data structure)?

- Spine Toolbox does not force you to use anything
- Make your own nomenclature or use a shared nomenclature
- Transformations need to be maintained (from the common nomenclature to tool specific nomenclature)
- A common nomenclature makes life a lot easier (lot less transformations needed)
- Spine Toolbox has internal support for common transformation operations
  - …work in progress
  - …and you can do anything with Python underneath

Importing data

- Need to be careful – easy to overwrite your work
- Currently Spine Toolbox imports to a new alternative (one entity can have alternative values for the same data)
- In future, we need to offer users more freedom to choose how to treat incoming data

Time series / array data

Object parameter value

object_class_name	object_name	parameter_name	alternative_name	value	database
grid_node	node 1	elec_demand	base	Map	Master data
grid_node	node 2	elec_demand	base	Map	Master data
				None	Master data

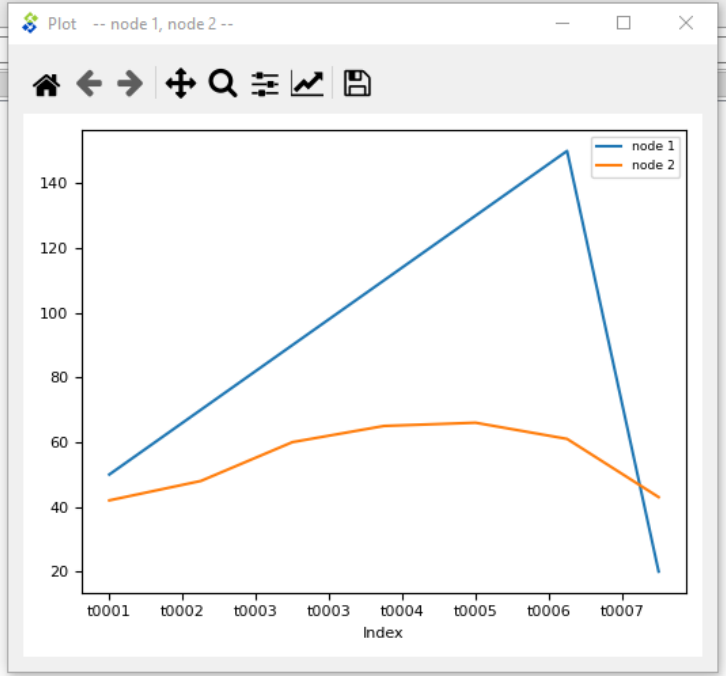
Pivot table

	parameter	elec_demand
	grid_node	
	node 1	Map
	node 2	Map

aster_data\Master data.sqlite

Pivot table

(X)		
grid_node	node 1	node 2
index		
t0001	50.0	42.0
t0002	70.0	48.0
t0003	90.0	60.0
t0004	110.0	65.0
t0005	130.0	66.0
t0006	150.0	61.0
t0007	20.0	43.0



Currently data in JSON - need to enable binary formats

Spine

Metadata

- Spine Toolbox supports metadata and importing metadata
  - Work in progress
- Metadata is complicated
- Same metadata may apply to multiple entities or to multiple parameter values
- Metadata can be partially same for multiple data objects
- How to avoid replication
- How to keep history (when someone changes one data item and others with same metadata remain unaffected)
- Common use case: user changes multiple records at once
- Spine Toolbox works with commits
  - Suggest to insert a commit message
  - Undo possible before commit
- Thinking how to maintain full history of data – needs to be user choice

The power of Python and Julia

- Spine Toolbox has an API in Python and in Julia
  - Direct access to the Spine databases from these languages
  - Embed Python or Julia scripts
- Python has almost everything already in open source
- Julia promises to be very fast

Jupyter Notebooks as Toolbox items (WIP)

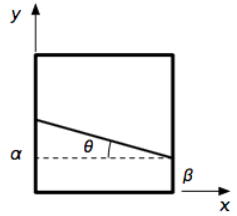
jupyter symblock (autosaved)

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

Three sides wetted

Reorient our point of view to make the analysis easier. Start by assuming that three sides of the block are at least partially immersed. Assume also that $\rho \leq 1/2$ and $\theta \leq 45°$.



In [1]:
```
from sympy import *
init_printing()
```

In [2]:
```
alpha, beta, rho, x, y, theta = symbols('alpha beta rho x y theta')
```

Geometric equations:

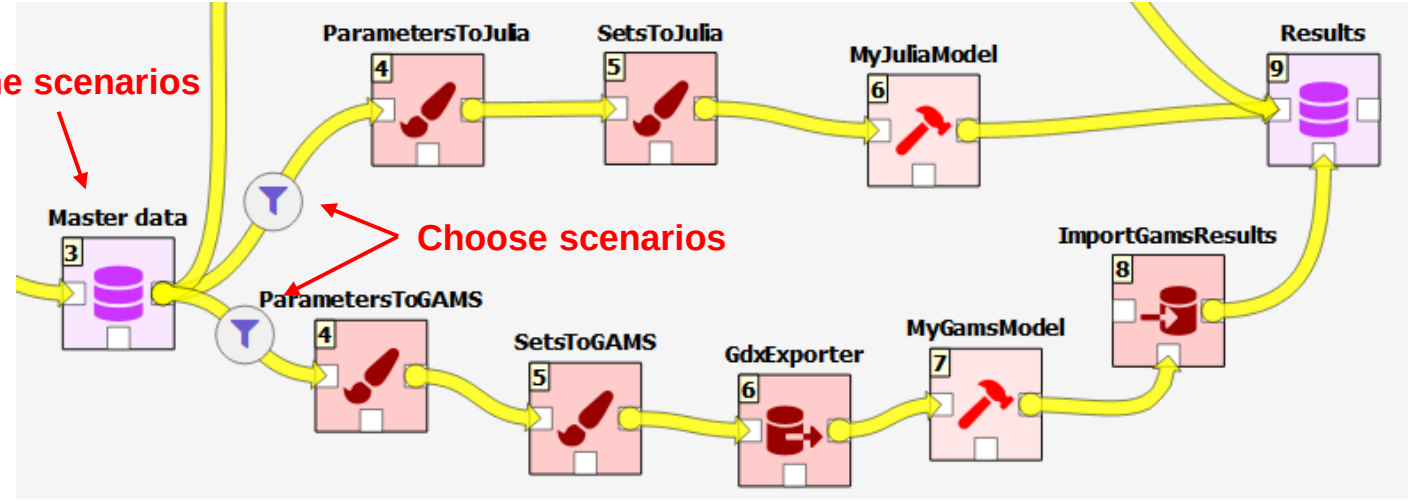
In [3]:
```
e1 = rho - (alpha + beta)/2
e2 = tan(theta) - (alpha - beta)
s12 = solve((e1, e2), (alpha, beta))
```

In [4]:
```
s12
```

Out[4]: $\left\{\alpha : \rho + \frac{1}{2}\tan(\theta), \quad \beta : \rho - \frac{1}{2}\tan(\theta)\right\}$

Spine Engine parallelizes workflows

- Spine Toolbox does not execute workflows (it stays responsive)
- Spine Engine uses Dagster to execute
  - To be cloud / computing cluster enabled
- Parallelization and headless execution

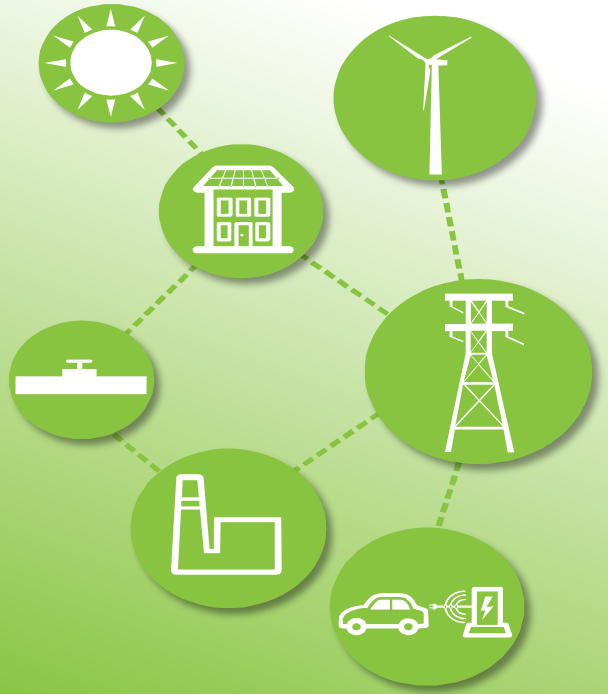


Spine Engine parallelizes: 2 tracks times x scenarios

Spine Toolbox is in early deployment mode

- At VTT we have started migrating data and tools to run on Spine Toolbox
- KU Leuven is using it national projects
  - Building Julia models
- KTH using it in MSc and PhD student projects
- Our open source GAMS model Backbone has 20-30 users in 5 countries
  - We have data conversion from Backbone to Spine Toolbox format and are starting to test using it in real projects
  - Nordic Energy Research funded FasTen and Amber projects (actual co-modelling between Baltic partners and VTT
- EU project TradeRES will be using Spine Toolbox
  - Orchestrating 2 optimization models and 4-5 agent based models to study electricity market design for high wind and solar futures
  - All running from a single database
  - First tests underway – data migration to happen next
- Spine project has 9 more months – documentation and building missing features (we're still well resourced for those 9 months)
- Trying to get new funding also for development (EU calls forthcoming, also involved with G-PST)
- Very happy to partner with resources from elsewhere – this is open source

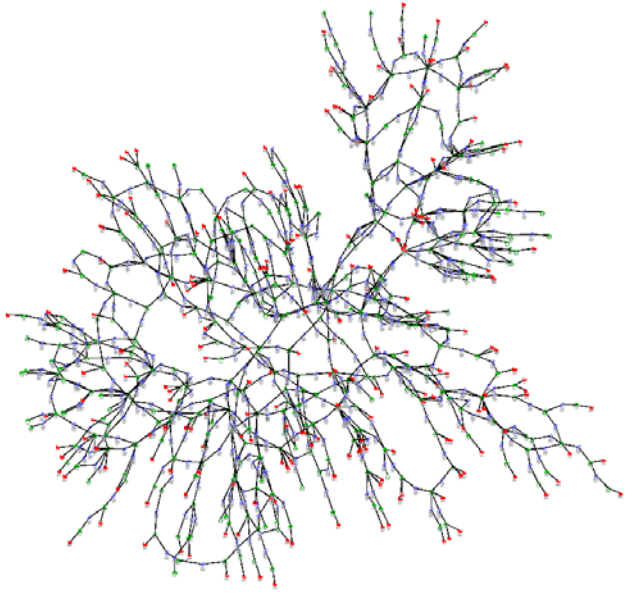
Spine



Spine case studies
to demonstrate capabilities

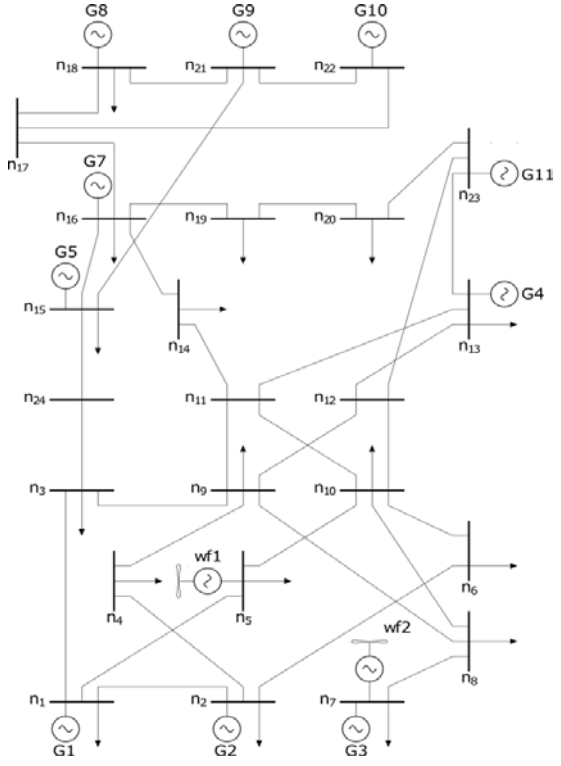
A1 Irish dispatch study with power flows

- The case study aims at replicating the functionality previously implement in Energy Reform's Epiphron software.
- The model includes three control areas, several units, one storage, various fuels.
- It is a rolling horizon unit commitment model.
- The current Spine implementation relies on Spine Toolbox and uses a port of Epiphron to Julia following Spine Model style.



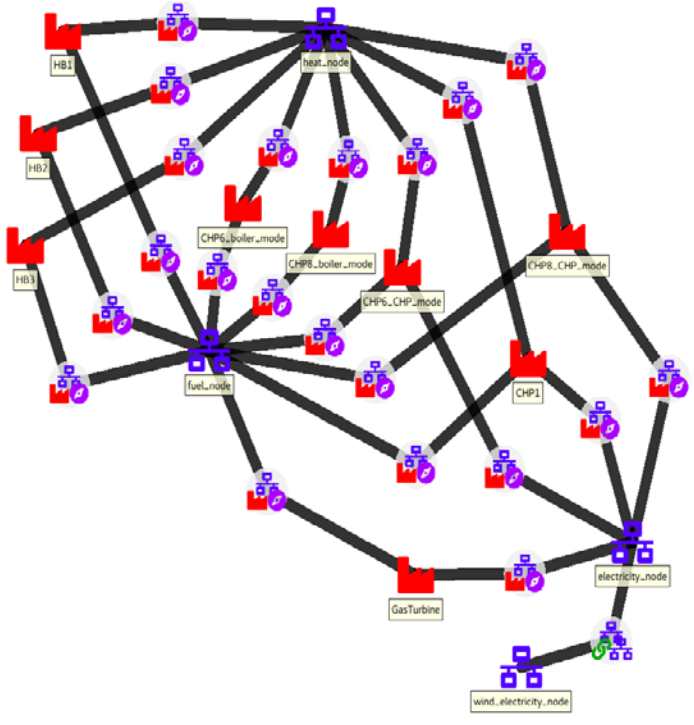
A2 Belgian gas grid study with pressure driven gas transfer

- The purpose of this study is to reproduce a gas transmission system with pressure-driven gas transfer. Moreover, the gas transmission system is linked to the electricity system.
- Many gas variables and equations were included in a Spine Model fashion, using the *extend* functionality.
- Results using Spine Model are well aligned with the original model.



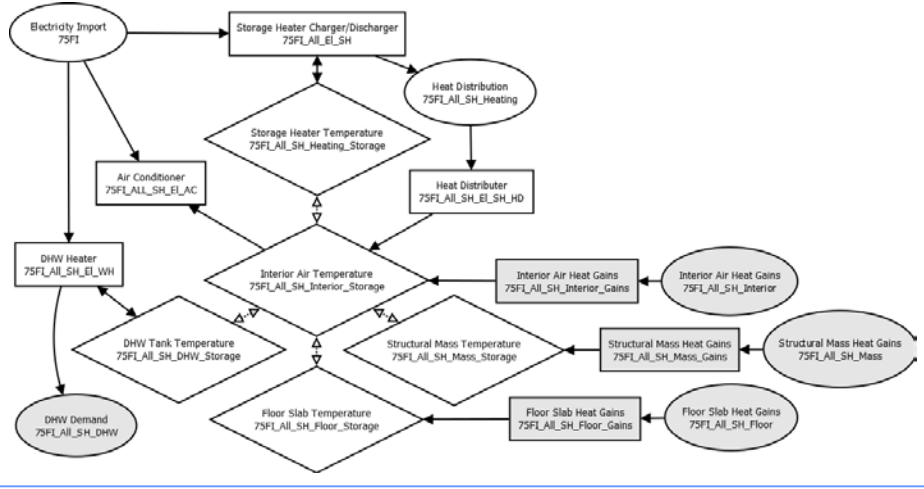
A3 District heating study of Stockholm

- The purpose of this case study is to simulate one year of operation of a subset of Stockholm's district heating system.
- The system includes 1 extraction condensing steam turbine, 2 back-pressure turbines, 1 gas turbine and 3 heat boilers.
- Ramp constraints and different types of start are neglected at the moment.
- Results using Spine Model are well aligned with the original model.



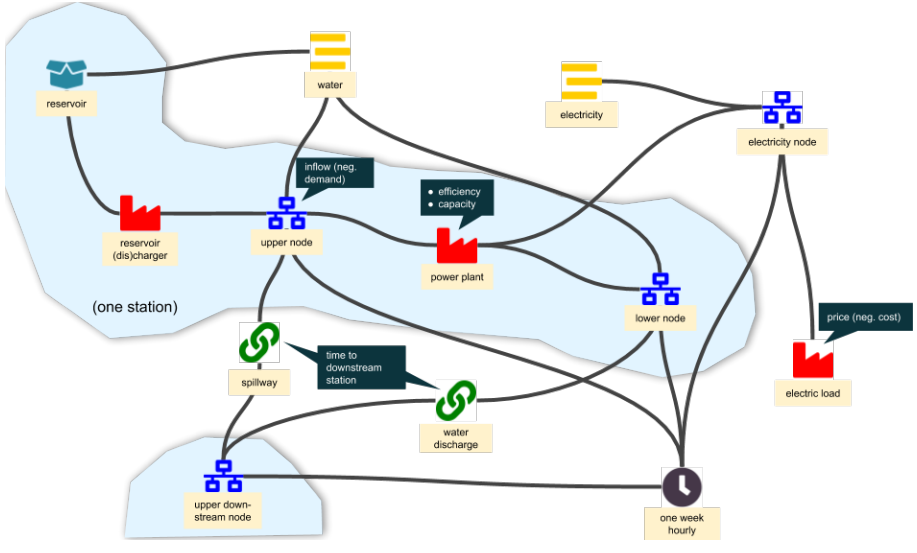
A4 Cost optimisation study with building heat physics

- This case study compares the performance of Spine Model against Backbone for a rolling unit commitment and economic dispatch optimisation of the Finnish energy system in 2020, including widespread flexible residential electric heating.
- Results using Spine Model are well aligned with the original Backbone model.



A5 Hydro power study with river systems

- The purpose of this study is to simulate one week of operation of the Skellefte river including fifteen power stations.
- Results have benn validated and are well aligned with the reference model.



Ongoing case studies

- B1 - Spatial aggregation of nodal systems
- B2 - Temporal aggregation of time periods
- B3 - Planning a low emission transport sector
- B4 - Biomass resource constraints
- B5 - Industrial energy use
- C1 - Market design for integrated energy systems
- C2 - Power grid investments under uncertainty
- C3 - Integrated energy system planning with high operational detail

Resources

- Spine Toolbox repository: https://github.com/Spine-project/Spine-Toolbox
- Spine Toolbox Windows release executables: https://drive.google.com/drive/folders/1t-AIIwRMl3HiYgka4ex5bCccI2gpbspK
- Spine project website: http://www.spine-model.org/

- Highly flexible energy system modelling framework SpineOpt in Julia (https://github.com/Spine-project)
- Energy and power system modelling framework Backbone in GAMS (https://gitlab.vtt.fi/backbone)
- IRENA FlexTool in Excel/Mathprog (https://www.irena.org/energytransition/Energy-System-Models-and-Data/IRENA-FlexTool)

Spine: Open source toolbox for modelling ~~integrated energy systems~~

- Project part funded by the Horizon 2020 program of the European Union
- LCE-05-2017 - Tools and technologies for coordination and integration of the European energy system
- 4 year project commenced October 2017 with a €3.7m budget
- 5 Partners, collaboration with NREL & DTU

www.spine-model.org

spine_info@vtt.fi