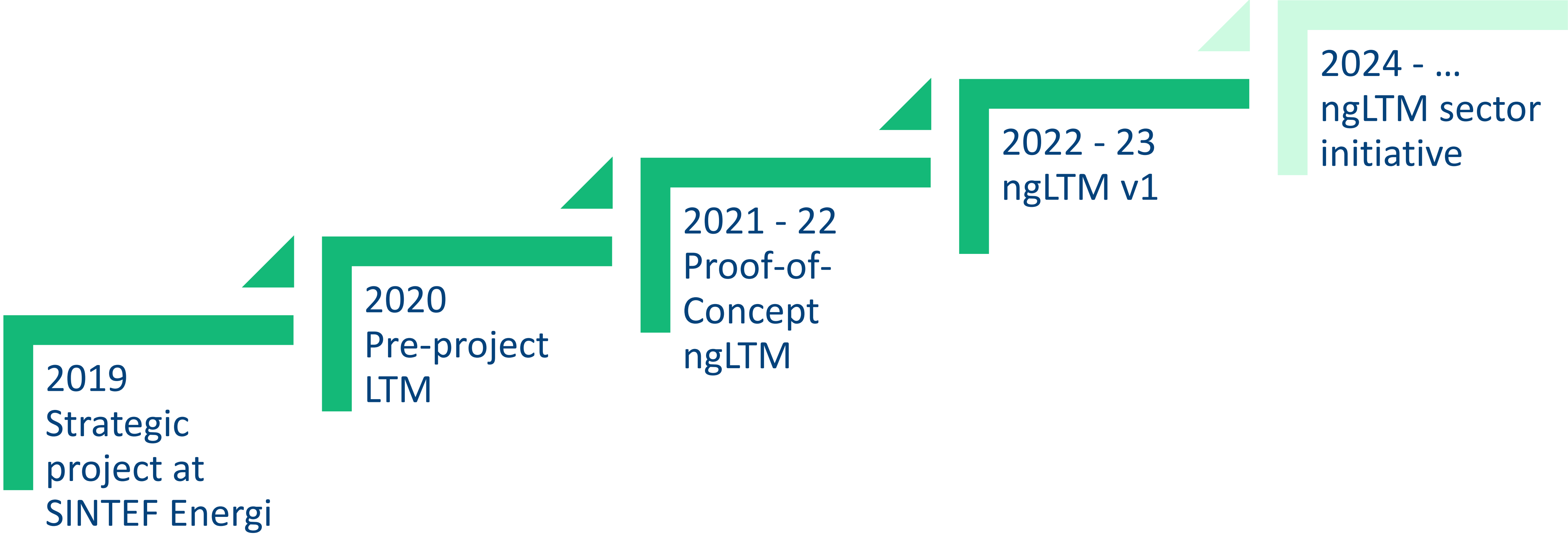
CONFIDENTIAL

ngLTM
Next generation market models

Sintef Energy Research

Energy systems

Development of the next generation of market models (ngLTM) has been realised through several stages and user engagement



2019
Strategic project at SINTEF Energi

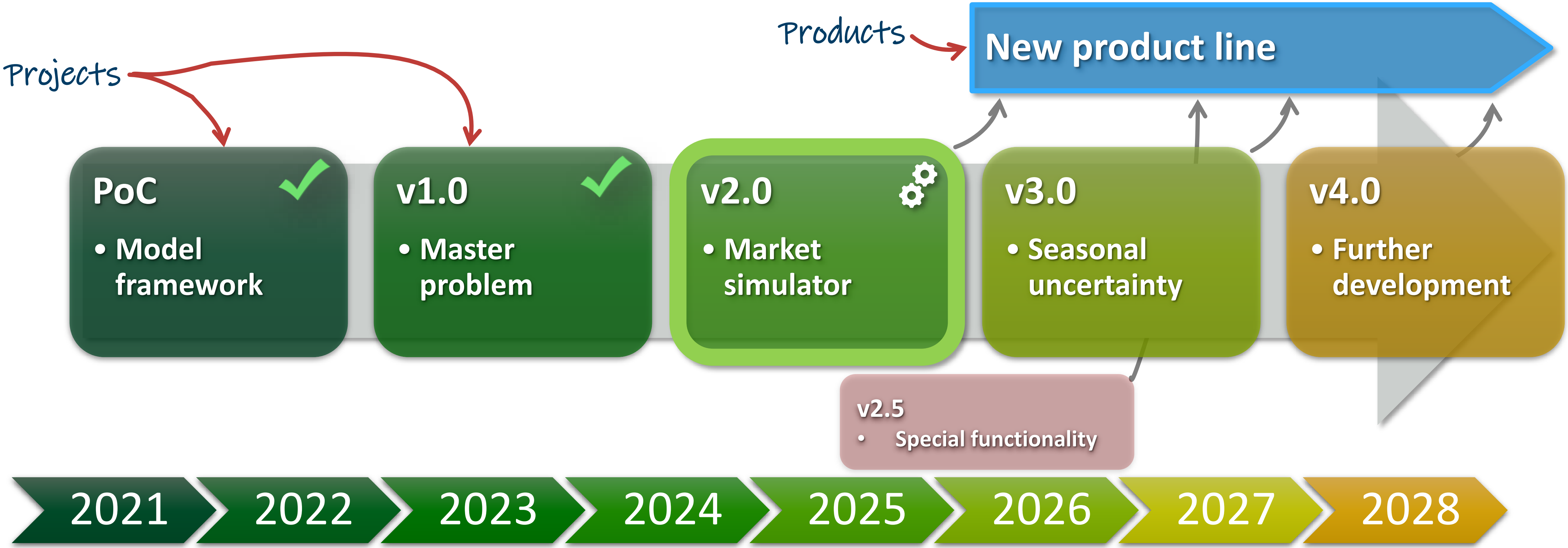
2020
Pre-project LTM

2021 - 22
Proof-of-Concept ngLTM

2022 - 23
ngLTM v1

2024 - …
ngLTM sector initiative

ngLTM is a development project planned and executed in several phases



Projects

Products

New product line

PoC ✓
- Model framework

v1.0 ✓
- Master problem

v2.0
- Market simulator

v3.0
- Seasonal uncertainty

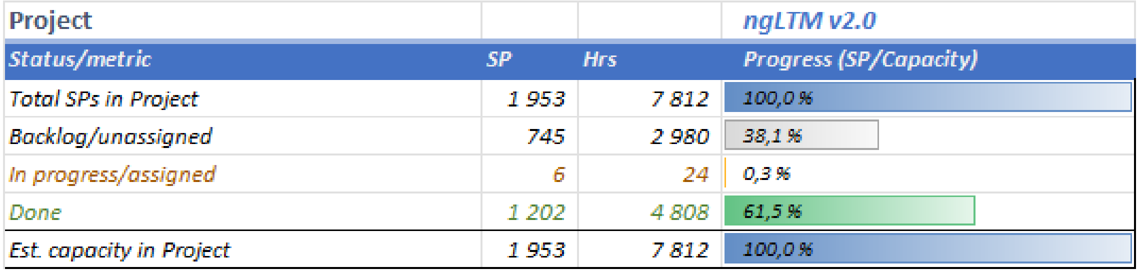
v4.0
- Further development

v2.5
- Special functionality

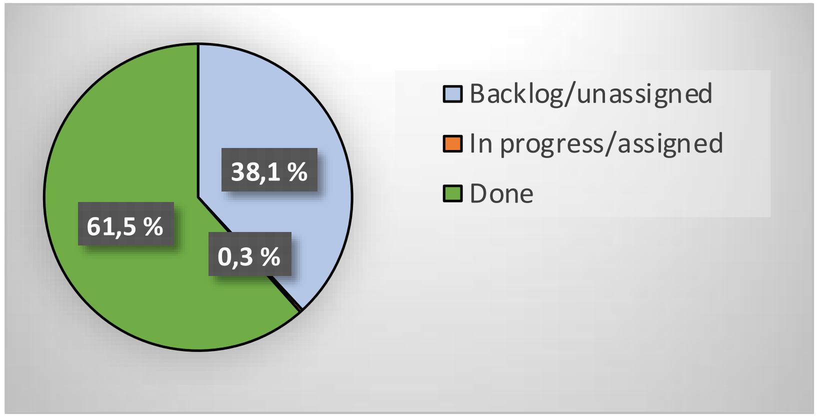
2021 | 2022 | 2023 | 2024 | 2025 | 2026 | 2027 | 2028

ngLTM v2 – Overall status Week 18 - 2025

- The development project has good progression without large incidents
- The largest part of functionality in version 2 is implemented and under testing
- We are on the way to provide a new product

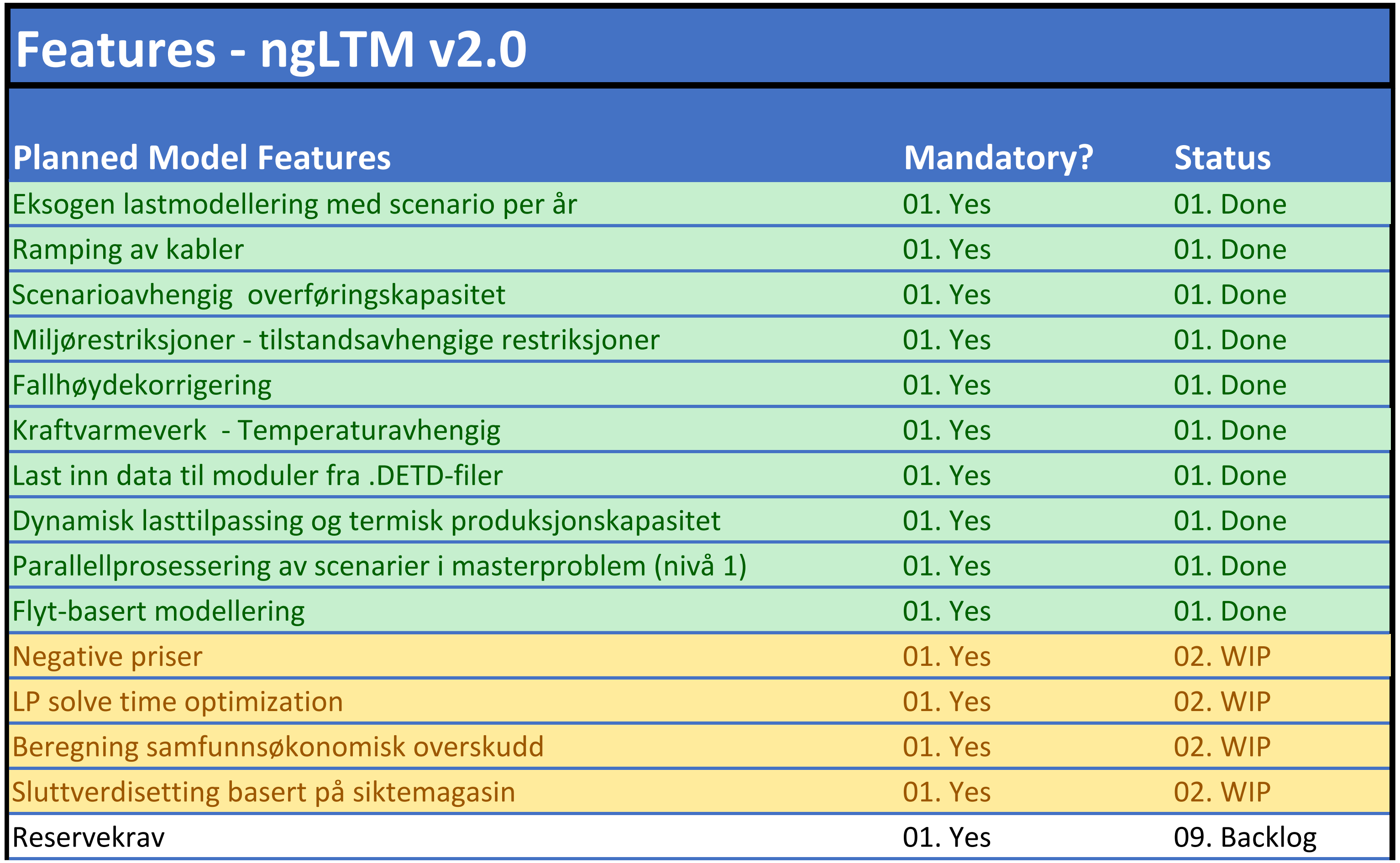


Project			*ngLTM v2.0*
Status/metric	***SP***	***Hrs***	***Progress (SP/Capacity)***
Total SPs in Project	*1 953*	*7 812*	*100,0 %*
Backlog/unassigned	*745*	*2 980*	*38,1 %*
In progress/assigned	*6*	*24*	*0,3 %*
Done	*1 202*	*4 808*	*61,5 %*
Est. capacity in Project	*1 953*	*7 812*	*100,0 %*



Technology for a better society

Most of the features are implemented or work in progress

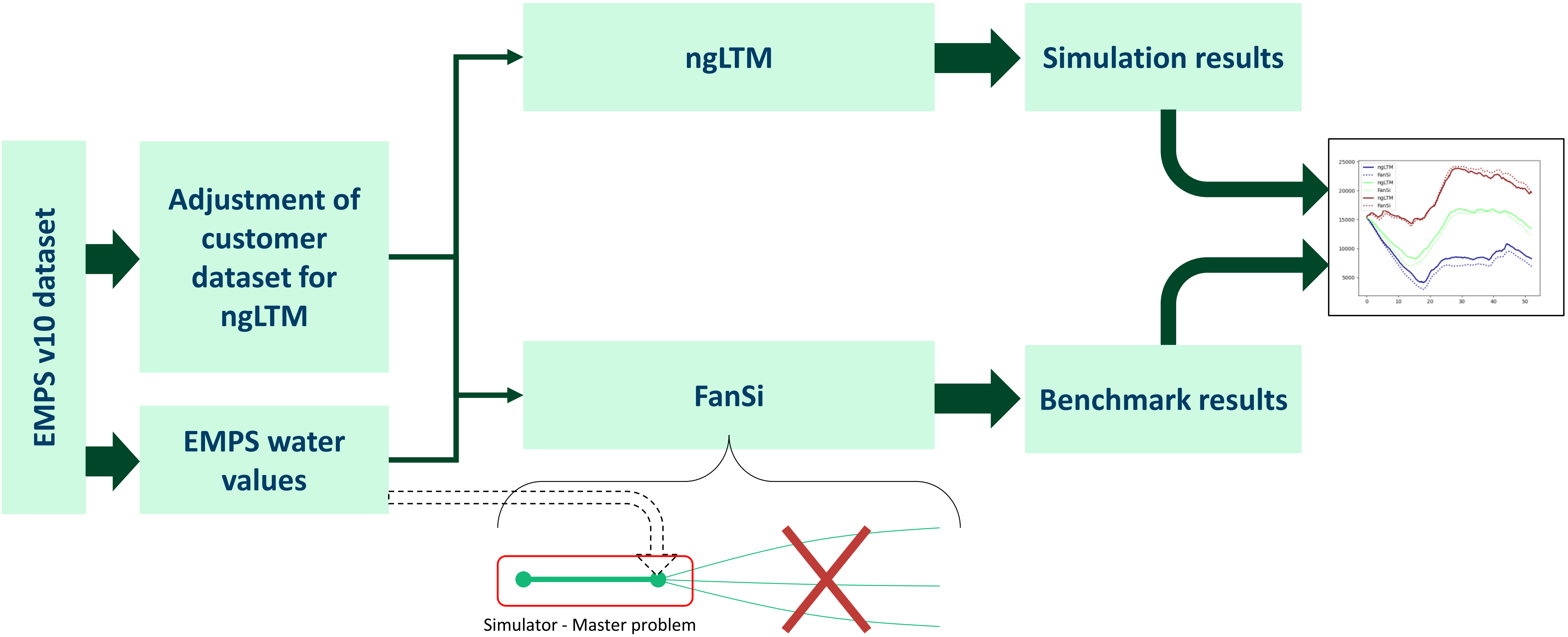


Features - ngLTM v2.0

Planned Model Features	Mandatory?	Status
Eksogen lastmodellering med scenario per år	01. Yes	01. Done
Ramping av kabler	01. Yes	01. Done
Scenarioavhengig overføringskapasitet	01. Yes	01. Done
Miljørestriksjoner - tilstandsavhengige restriksjoner	01. Yes	01. Done
Fallhøydekorrigering	01. Yes	01. Done
Kraftvarmeverk - Temperaturavhengig	01. Yes	01. Done
Last inn data til moduler fra .DETD-filer	01. Yes	01. Done
Dynamisk lasttilpassing og termisk produksjonskapasitet	01. Yes	01. Done
Parallellprosessering av scenarier i masterproblem (nivå 1)	01. Yes	01. Done
Flyt-basert modellering	01. Yes	01. Done
Negative priser	01. Yes	02. WIP
LP solve time optimization	01. Yes	02. WIP
Beregning samfunnsøkonomisk overskudd	01. Yes	02. WIP
Sluttverdisetting basert på siktemagasin	01. Yes	02. WIP
Reservekrav	01. Yes	09. Backlog

ngLTM v2 Testing

ngLTM is tested thoroughly on customer datasets with FanSi as a benchmark



EMPS v10 dataset

Adjustment of customer dataset for ngLTM

EMPS water values

ngLTM

FanSi

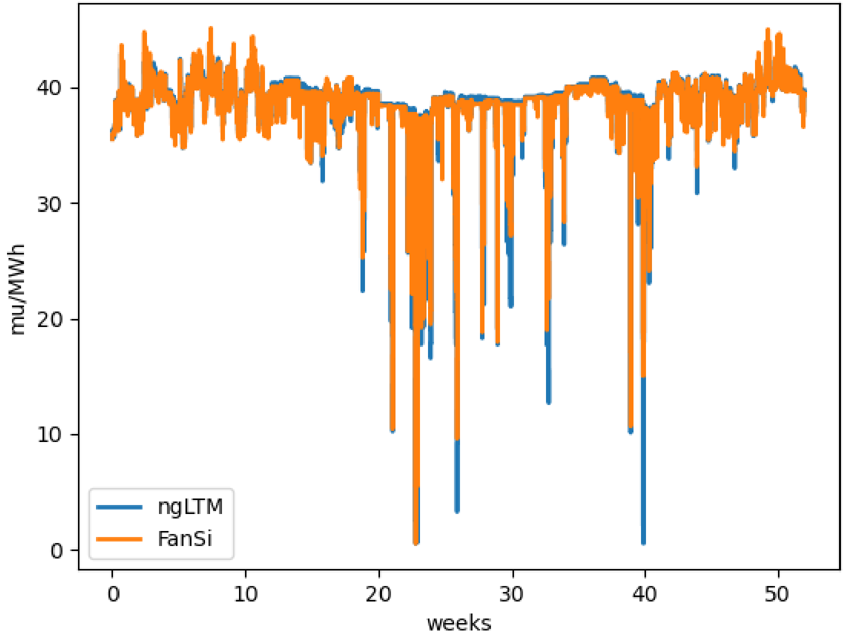
Simulation results

Benchmark results

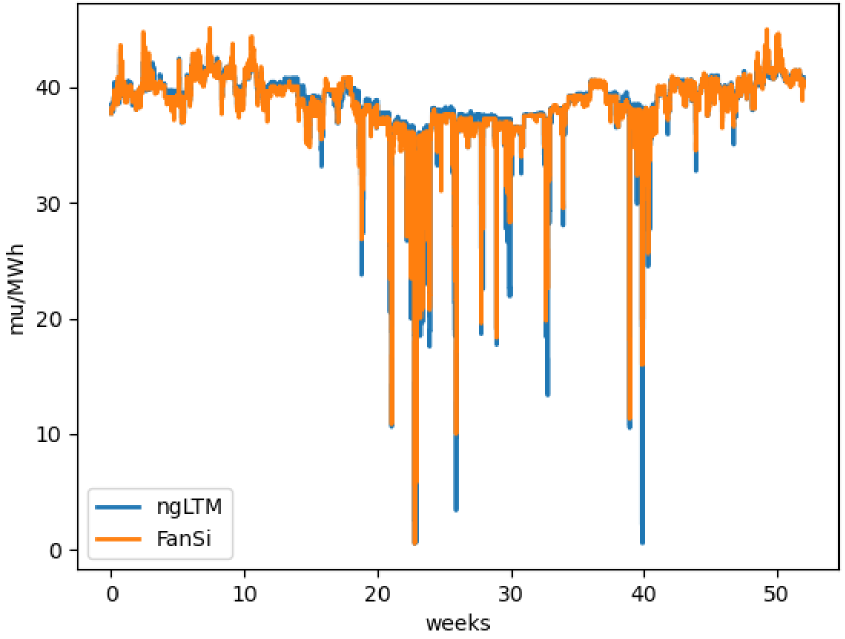
Simulator - Master problem

Comparison of power prices from testing with SINTEF's Nordic dataset

NO 5



NO 1

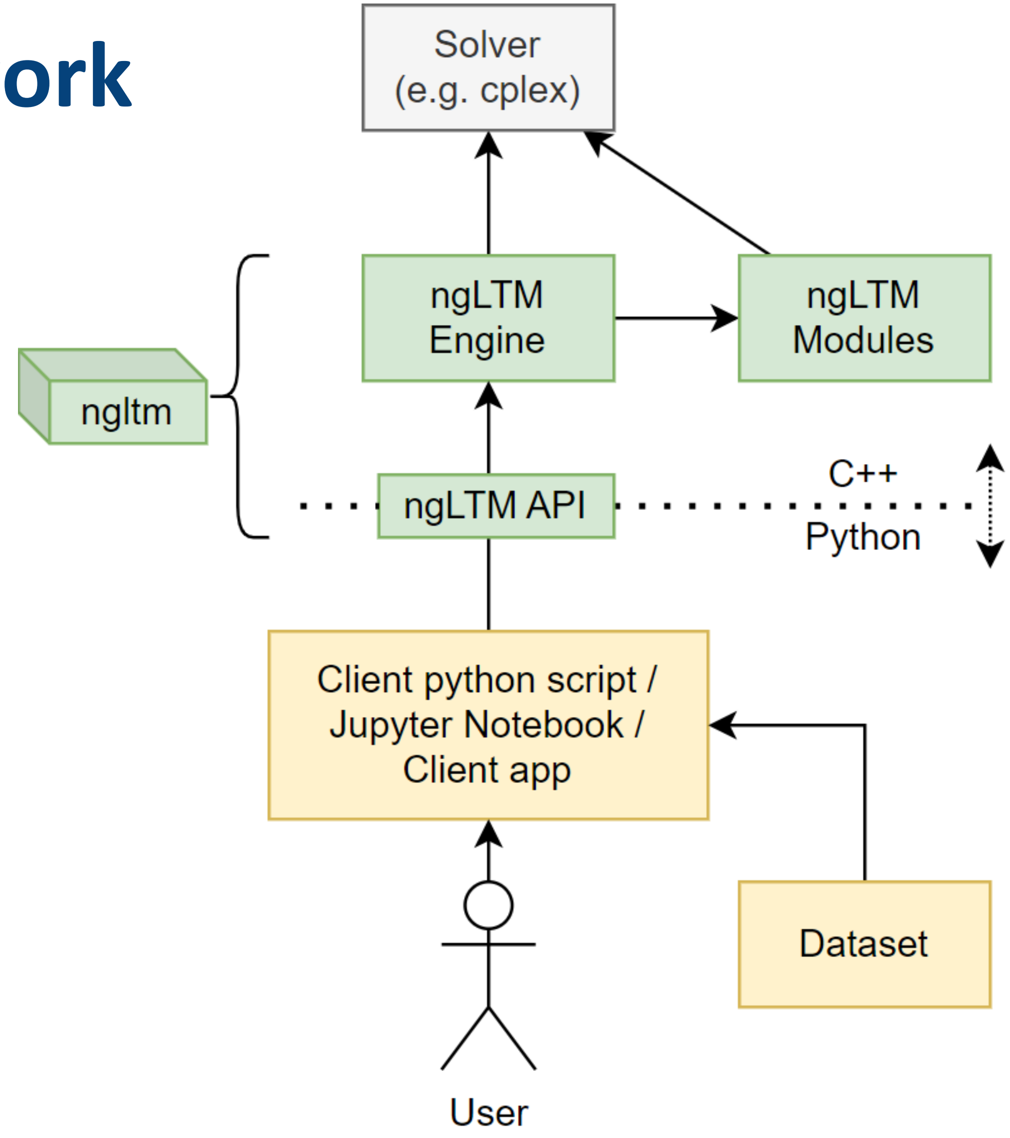


ngLTM v2 Demonstration

The ngLTM model framework

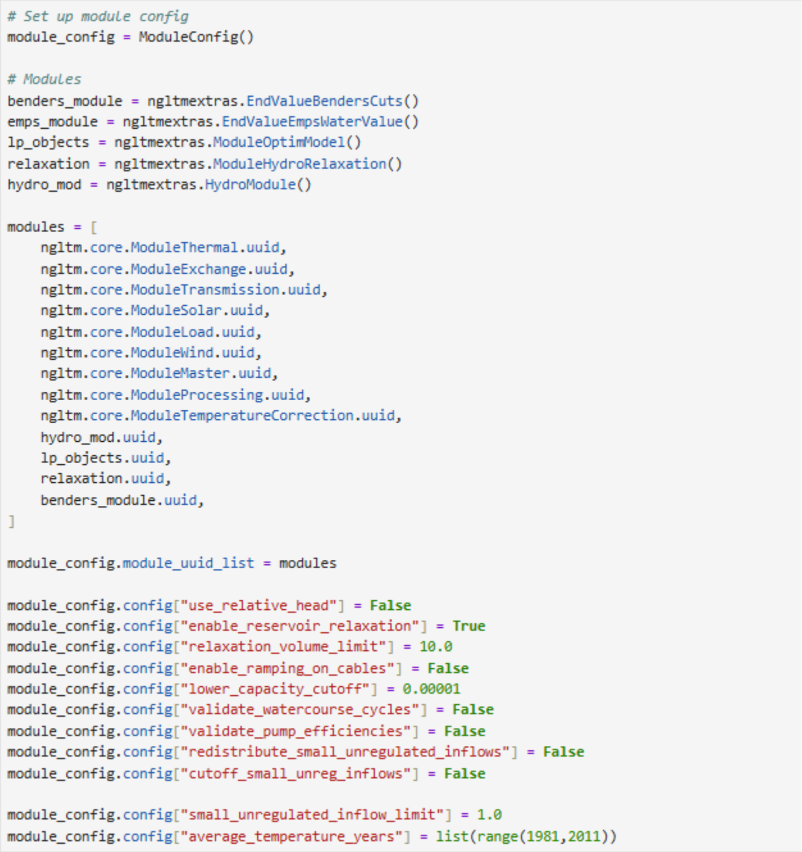
- **C++ framework with modularity in mind**
  - Efficient and fast code for core features
  - Loose coupling between features with interfaces
  - Domain Specific Language (DSL) for LP problems
  - "Future proofing", easy to adapt new tech

- **Framework exposed to Python**
  - Well known programming language
  - Easy to use

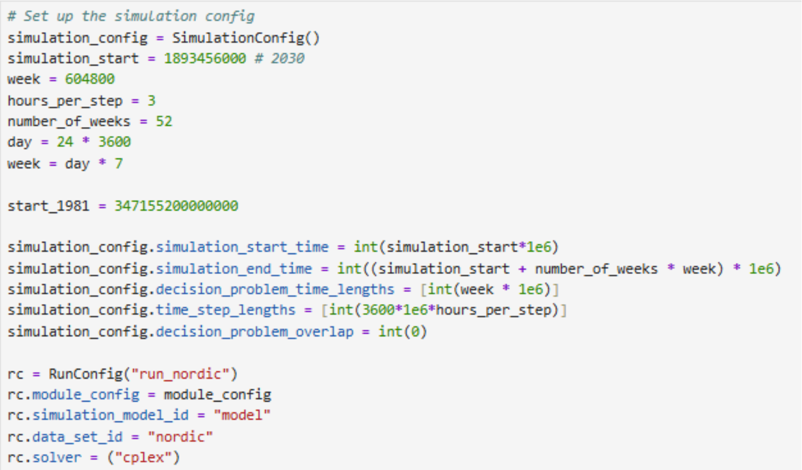


Out with the old

- A year ago the scripts for setting up and running ngLTM could get fairly large.
- Focused work on making the API robust, intuitive and easy to use.
- Moving and storing configurations should also be possible.



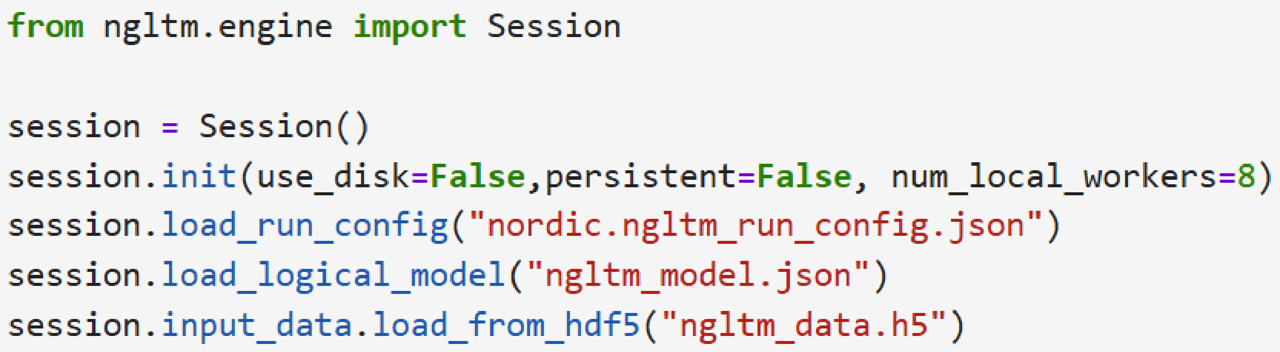
```python
# Set up module config
module_config = ModuleConfig()

# Modules
benders_module = ngltmextras.EndValueBendersCuts()
emps_module = ngltmextras.EndValueEmpsWaterValue()
lp_objects = ngltmextras.ModuleOptimModel()
relaxation = ngltmextras.ModuleHydroRelaxation()
hydro_mod = ngltmextras.HydroModule()

modules = [
    ngltm.core.ModuleThermal.uuid,
    ngltm.core.ModuleExchange.uuid,
    ngltm.core.ModuleTransmission.uuid,
    ngltm.core.ModuleSolar.uuid,
    ngltm.core.ModuleLoad.uuid,
    ngltm.core.ModuleWind.uuid,
    ngltm.core.ModuleMaster.uuid,
    ngltm.core.ModuleProcessing.uuid,
    ngltm.core.ModuleTemperatureCorrection.uuid,
    hydro_mod.uuid,
    lp_objects.uuid,
    relaxation.uuid,
    benders_module.uuid,
]

module_config.module_uuid_list = modules

module_config.config["use_relative_head"] = False
module_config.config["enable_reservoir_relaxation"] = True
module_config.config["relaxation_volume_limit"] = 10.0
module_config.config["enable_ramping_on_cables"] = False
module_config.config["lower_capacity_cutoff"] = 0.00001
module_config.config["validate_watercourse_cycles"] = False
module_config.config["validate_pump_efficiencies"] = False
module_config.config["redistribute_small_unregulated_inflows"] = False
module_config.config["cutoff_small_unreg_inflows"] = False

module_config.config["small_unregulated_inflow_limit"] = 1.0
module_config.config["average_temperature_years"] = list(range(1981,2011))
```



```python
# Set up the simulation config
simulation_config = SimulationConfig()
simulation_start = 1893456000 # 2030
week = 604800
hours_per_step = 3
number_of_weeks = 52
day = 24 * 3600
week = day * 7

start_1981 = 347155200000000

simulation_config.simulation_start_time = int(simulation_start*1e6)
simulation_config.simulation_end_time = int((simulation_start + number_of_weeks * week) * 1e6)
simulation_config.decision_problem_time_lengths = [int(week * 1e6)]
simulation_config.time_step_lengths = [int(3600*1e6*hours_per_step)]
simulation_config.decision_problem_overlap = int(0)

rc = RunConfig("run_nordic")
rc.module_config = module_config
rc.simulation_model_id = "model"
rc.data_set_id = "nordic"
rc.solver = ("cplex")
```

And in with the new

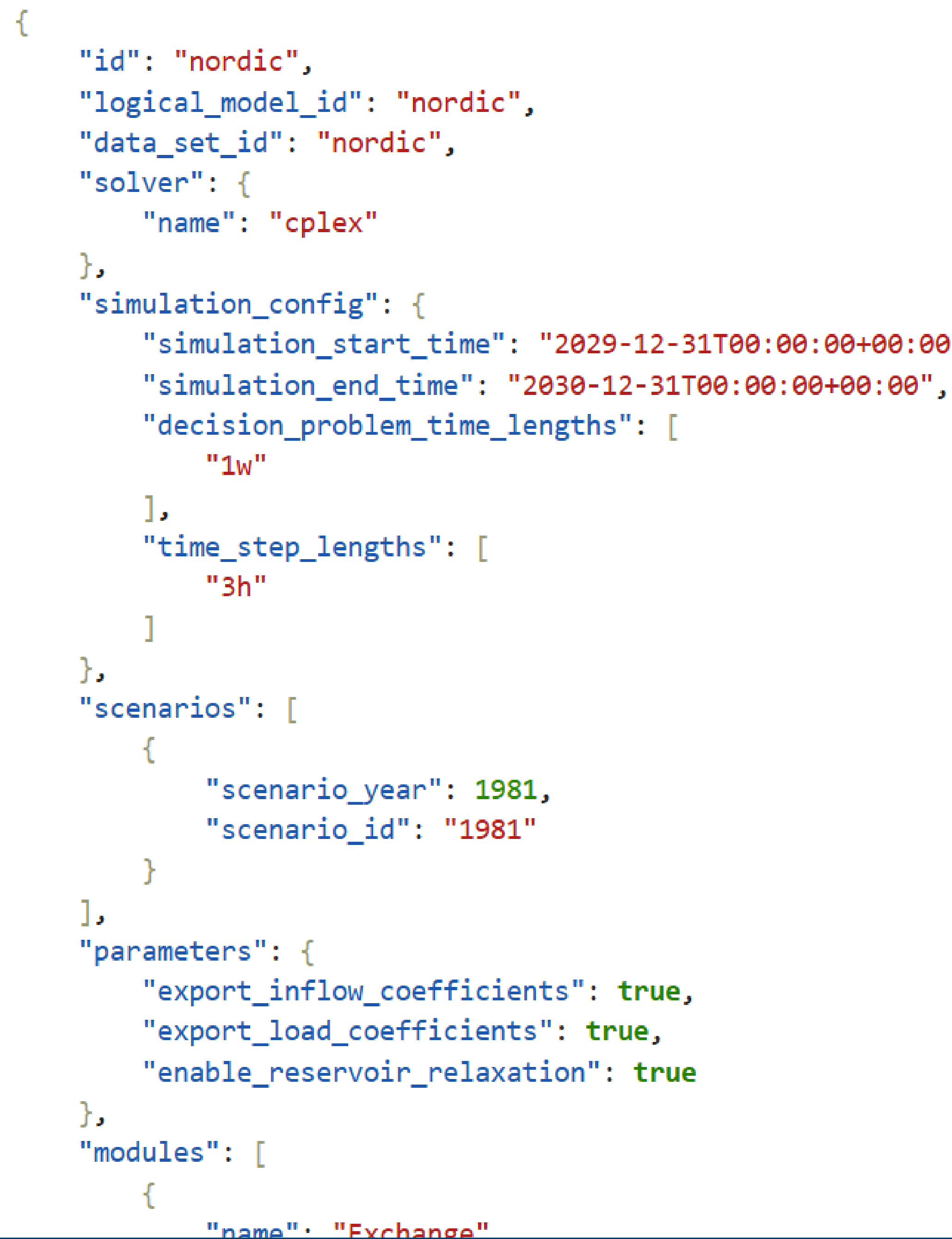
- Session; a convenient way to set up and run ngLTM.
- Access to all API functions if adjustments are needed.
- Greatly simplified interaction with the model.



```
from ngltm.engine import Session

session = Session()
session.init(use_disk=False,persistent=False, num_local_workers=8)
session.load_run_config("nordic.ngltm_run_config.json")
session.load_logical_model("ngltm_model.json")
session.input_data.load_from_hdf5("ngltm_data.h5")
```

Ease to use is a focus point for API

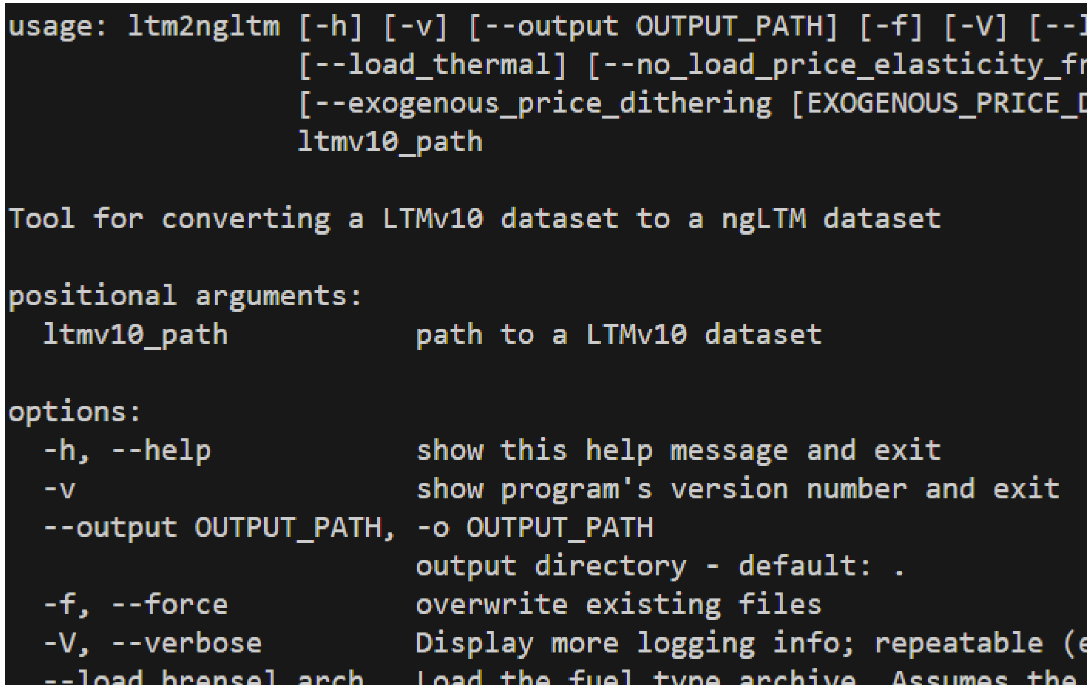
- Configurations can be stored as json files.
- Can also be set (and edited) directly through the Python API.



```
{
    "id": "nordic",
    "logical_model_id": "nordic",
    "data_set_id": "nordic",
    "solver": {
        "name": "cplex"
    },
    "simulation_config": {
        "simulation_start_time": "2029-12-31T00:00:00+00:00
        "simulation_end_time": "2030-12-31T00:00:00+00:00",
        "decision_problem_time_lengths": [
            "1w"
        ],
        "time_step_lengths": [
            "3h"
        ]
    },
    "scenarios": [
        {
            "scenario_year": 1981,
            "scenario_id": "1981"
        }
    ],
    "parameters": {
        "export_inflow_coefficients": true,
        "export_load_coefficients": true,
        "enable_reservoir_relaxation": true
    },
    "modules": [
        {
            "name": "Exchange"
```

Tools for LTMV10 data sets

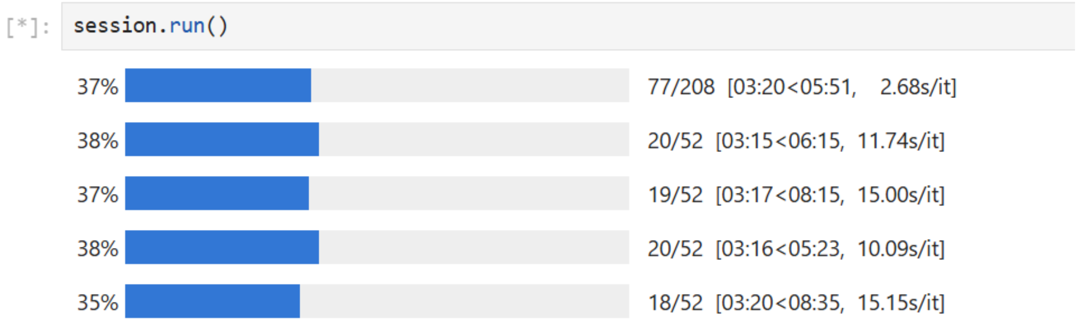
- Conversion tools for converting LTMV10 data sets.
- Converts to a format easily digested by ngLTM.
- Based on loader scripts which can be run with Python.



```
usage: ltm2ngltm [-h] [-v] [--output OUTPUT_PATH] [-f] [-V] [--l
                 [--load_thermal] [--no_load_price_elasticity_fr
                 [--exogenous_price_dithering [EXOGENOUS_PRICE_D
                 ltmv10_path

Tool for converting a LTMv10 dataset to a ngLTM dataset

positional arguments:
  ltmv10_path           path to a LTMv10 dataset

options:
  -h, --help            show this help message and exit
  -v                    show program's version number and exit
  --output OUTPUT_PATH, -o OUTPUT_PATH
                        output directory - default: .
  -f, --force           overwrite existing files
  -V, --verbose         Display more logging info; repeatable (e
  --load_brensel_arch   Load the fuel type archive. Assumes the
```

Running ngLTM

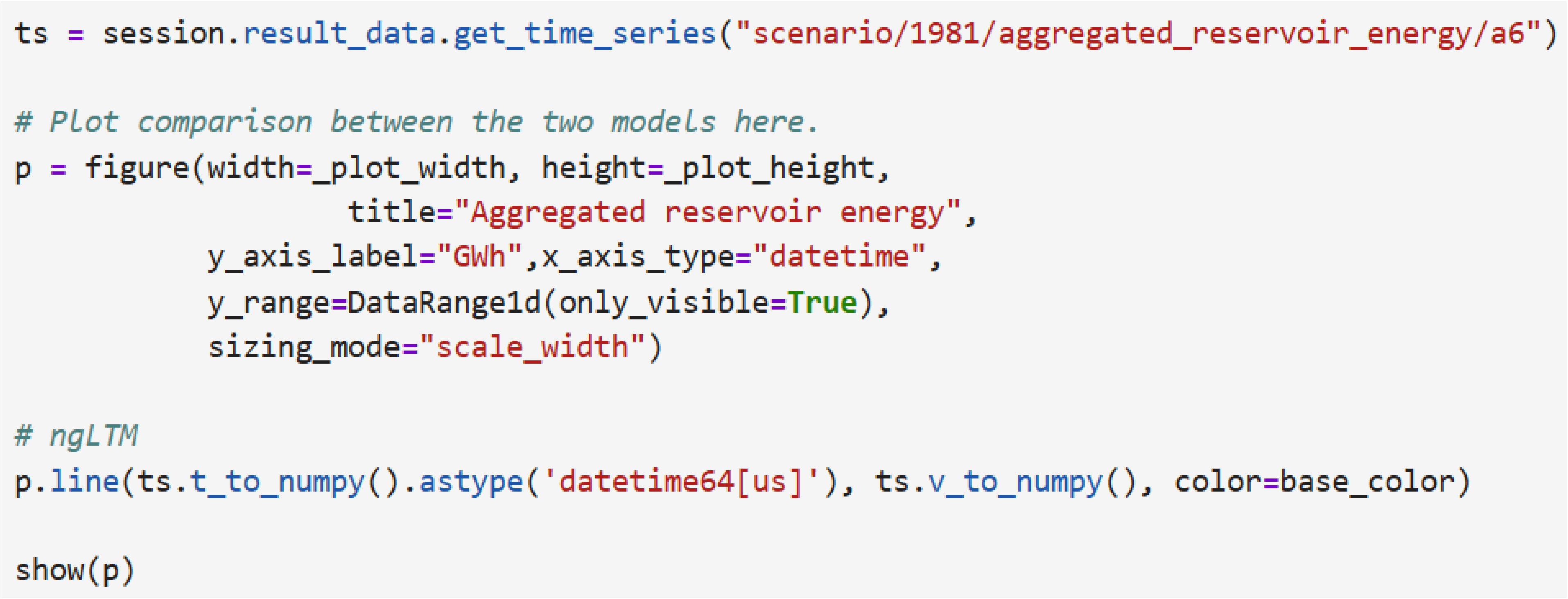
- Notebook example: Running ngLTM with session.
- Local parallelization of scenario years with threads
- MPI implementation is also available



```
[*]: session.run()
```

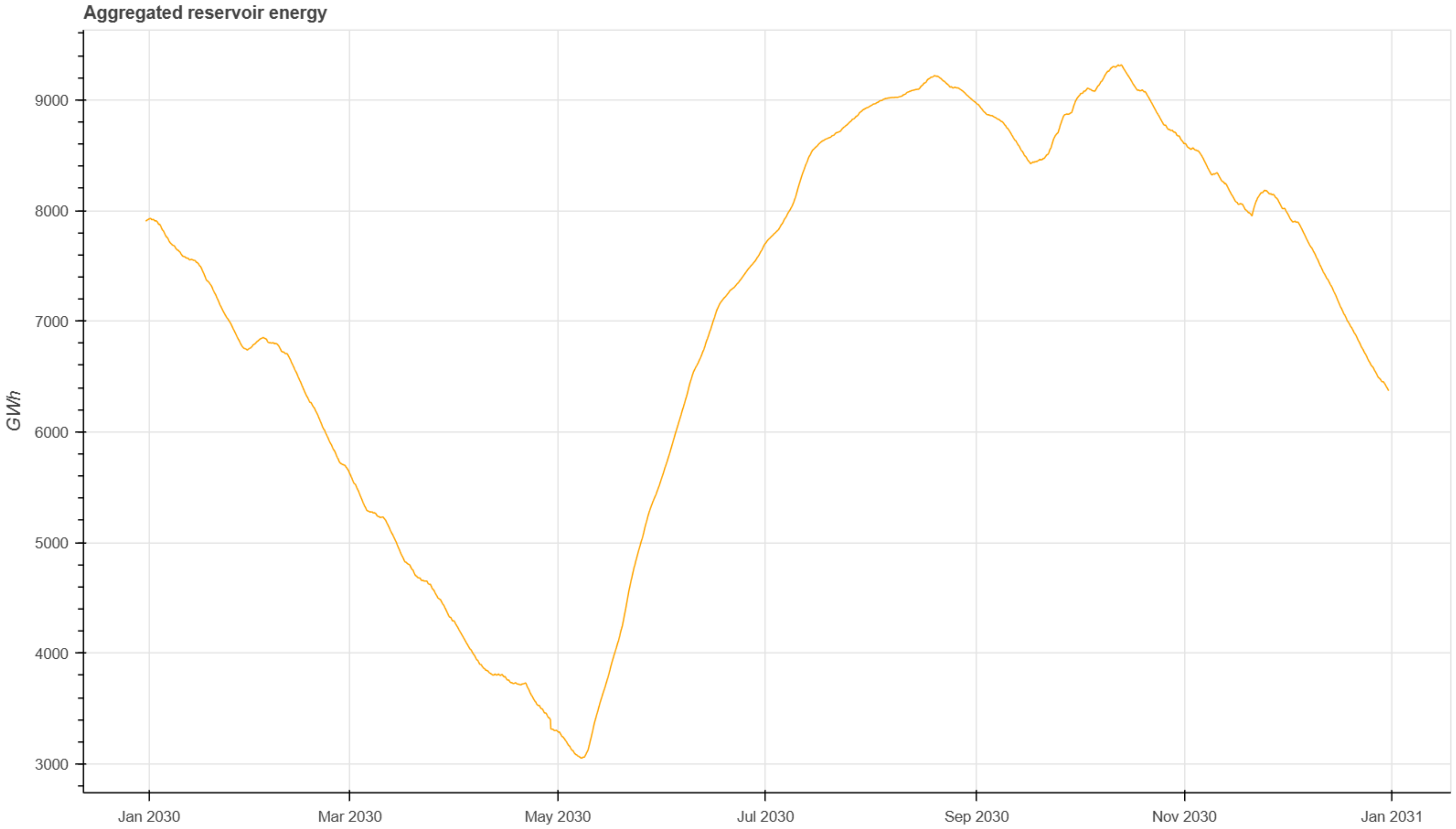
```
37%   77/208 [03:20<05:51,  2.68s/it]
38%   20/52 [03:15<06:15, 11.74s/it]
37%   19/52 [03:17<08:15, 15.00s/it]
38%   20/52 [03:16<05:23, 10.09s/it]
35%   18/52 [03:20<08:35, 15.15s/it]
```

Example: displaying results with Bokeh



```
ts = session.result_data.get_time_series("scenario/1981/aggregated_reservoir_energy/a6")

# Plot comparison between the two models here.
p = figure(width=_plot_width, height=_plot_height,
                   title="Aggregated reservoir energy",
           y_axis_label="GWh",x_axis_type="datetime",
           y_range=DataRange1d(only_visible=True),
           sizing_mode="scale_width")

# ngLTM
p.line(ts.t_to_numpy().astype('datetime64[us]'), ts.v_to_numpy(), color=base_color)

show(p)
```

Example: displaying results with Bokeh



ngLTM v2
Pilot model user

SINTEF

1950 – 2025
Technology for a
better society