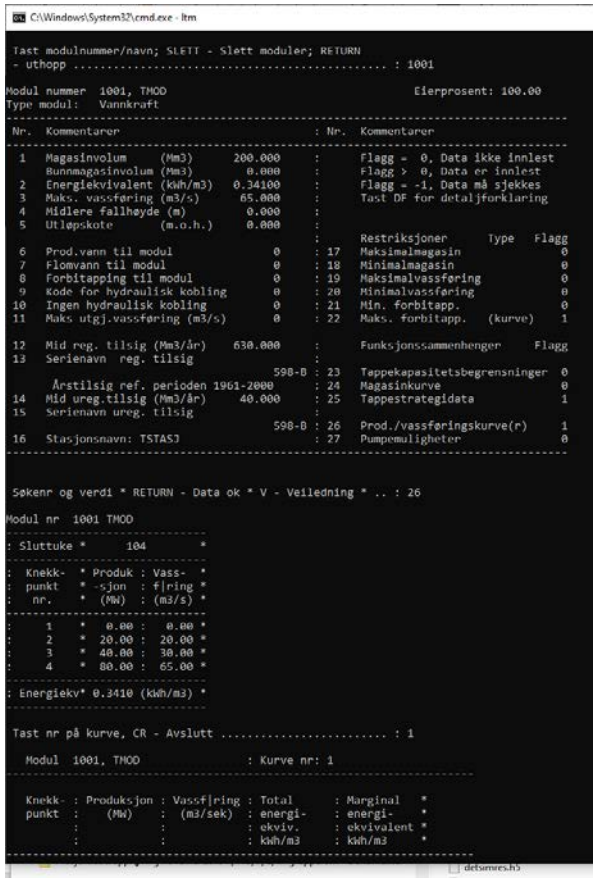
ProdRisk API

Hans Olaf Hågenvik

Agenda

- Introduction
- ProdRisk user interfaces
- Examples
- What's missing?

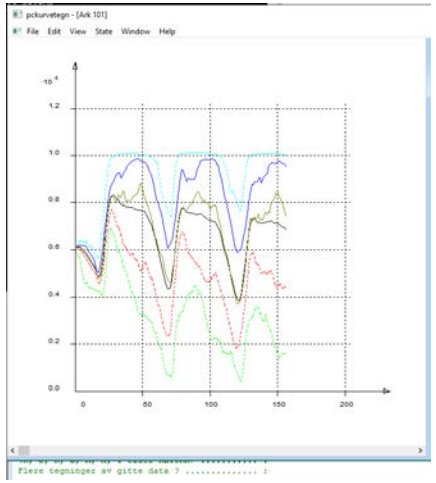
Introduction



ltm.exe / med.exe

ProfTemp	10.02.2019 20:53	File folder	
detsimres.h5	15.07.2019 14:50	H5 File	1 150 KB
DYNMODELL.SIMT	10.02.2019 20:56	SIMT File	15 KB
EFFEKTPROFIL.ARCH	16.05.2018 11:00	ARCH File	12 KB
ENMRES.h5	15.07.2019 14:50	H5 File	1 804 KB
hdf5_copywrite.dat	15.07.2019 14:45	DAT File	2 KB
historical.h5	10.02.2019 20:56	H5 File	111 KB
HKORR.SDDP	21.06.2019 10:14	SDDP File	2 KB
indvan.prd	15.07.2019 14:50	PRD File	1 KB
KUTT.SDDP	15.07.2019 14:50	SDDP File	9 976 KB
LASTPROFIL.ARCH	16.05.2018 11:00	ARCH File	1 KB
LPtags	15.07.2019 13:22	File	1 KB
LTM_license_Sintef.dat.LTM_log	12.11.2021 19:48	LTM_LOG File	17 KB
LtmSystem.xml	10.02.2019 20:58	XML Document	2 KB
MAGVOL.SDDP	21.06.2019 10:14	SDDP File	3 KB
miljo.bat	06.11.2019 14:57	Windows Batch File	1 KB
model.h5	21.12.2020 12:09	H5 File	22 KB
NY_HKORR.SDDP	15.07.2019 14:50	SDDP File	2 KB
NY_MAGVOL.SDDP	15.07.2019 14:50	SDDP File	3 KB
PARAMETER.TRAN	10.02.2019 20:56	TRAN File	1 KB
PRISAVSNITT.DATA	16.05.2018 11:00	DATA File	2 KB
PRISREKKE.PRI	16.05.2018 11:00	PRI File	123 KB
Prisrekke.PRISMOD	10.02.2019 20:56	PRISMOD File	58 KB
prodr.dat	10.02.2019 20:56	DAT File	48 KB
prodrisk.CPAR	12.06.2018 12:43	CPAR File	2 KB
prodrisk.exe	15.07.2019 13:12	Application	35 928 KB
residualstoy.dat	15.07.2019 14:45	DAT File	8 KB
RunInfo.txt	12.11.2021 19:48	Text Document	2 KB
ScenarioData.h5	10.02.2019 20:56	H5 File	271 KB
STYREFIL-PB027.EFI	15.07.2019 14:50	EFI File	4 KB
TEMPPROFIL_HODE.ARCH	10.02.2019 20:53	ARCH File	1 KB
TILSIG.SDDP	15.07.2019 14:45	SDDP File	5 KB
Tilsigsddp.CPAR	12.06.2018 12:43	CPAR File	1 KB

Input/output files



pckurvetegn

FANTASI.ENMD

Simuleringsperiode: Fom. uke 01 2002 tom. uke 52 2004

Resultatene er referert til prosentiler av tilsig

Økonomi fom. uke 01 2002 tom. uke 52 2004

	ENHET	MIDDEL	0 %	25 %	50 %	75 %	100 %
	År		1940	1946	1951	1947	1948
Tilsig	GWh	34053.6	26555.1	31739.2	33405.4	35059.9	42404.3
Flom	GWh	1471.8	985.3	842.9	1080.6	1464.5	3178.8
Forbruk til pumping	GWh	471.3	325.1	450.0	354.1	453.9	337.8
Gevinst ved pumping	GWh	620.5	474.8	588.9	517.4	613.2	539.9
Netto pumpeenergi	GWh	149.3	149.7	138.9	163.3	159.3	202.1
Startmagasin	GWh	6096.4	6096.4	6096.4	6096.4	6096.4	6096.4
Sluttmagasin	GWh	6805.7	4973.4	7083.8	5859.8	10024.7	9832.5
Levert vannkraft	GWh	31941.8	26842.7	30047.7	32724.6	29826.4	35691.6
DISKONTERT TIL UKE 01							
REALRENTE 0.00 %							
Fastkraft ettersprsel	Mkr	6900.0	6900.0	6900.0	6900.0	6900.0	6900.0
Ikke levert fastkraft	Mkr	0.0	0.0	0.0	0.0	0.0	0.0
Utkoblbart salg	Mkr	0.0	0.0	0.0	0.0	0.0	0.0
Salg spotmarked	Mkr	1533.4	1967.7	1710.6	1864.9	1538.8	958.7
Sum inntekter	Mkr	8433.4	8867.7	8610.7	8764.9	8438.8	7858.7
Kontraktrettigheter	Mkr	0.0	0.0	0.0	0.0	0.0	0.0
Utkoblbart kjøp	Mkr	0.0	0.0	0.0	0.0	0.0	0.0
Kjøp spotmarked	Mkr	2033.2	4455.6	2531.9	1988.5	2442.6	1623.2
Sum kostnader	Mkr	2033.2	4455.6	2531.9	1988.5	2442.6	1623.2
Netto inntekter	Mkr	6400.2	4412.2	6078.8	6776.4	5996.2	6235.5
Netto inntekter spot	Mkr	-499.8	-2487.8	-821.2	-123.6	-903.8	-664.5

ET

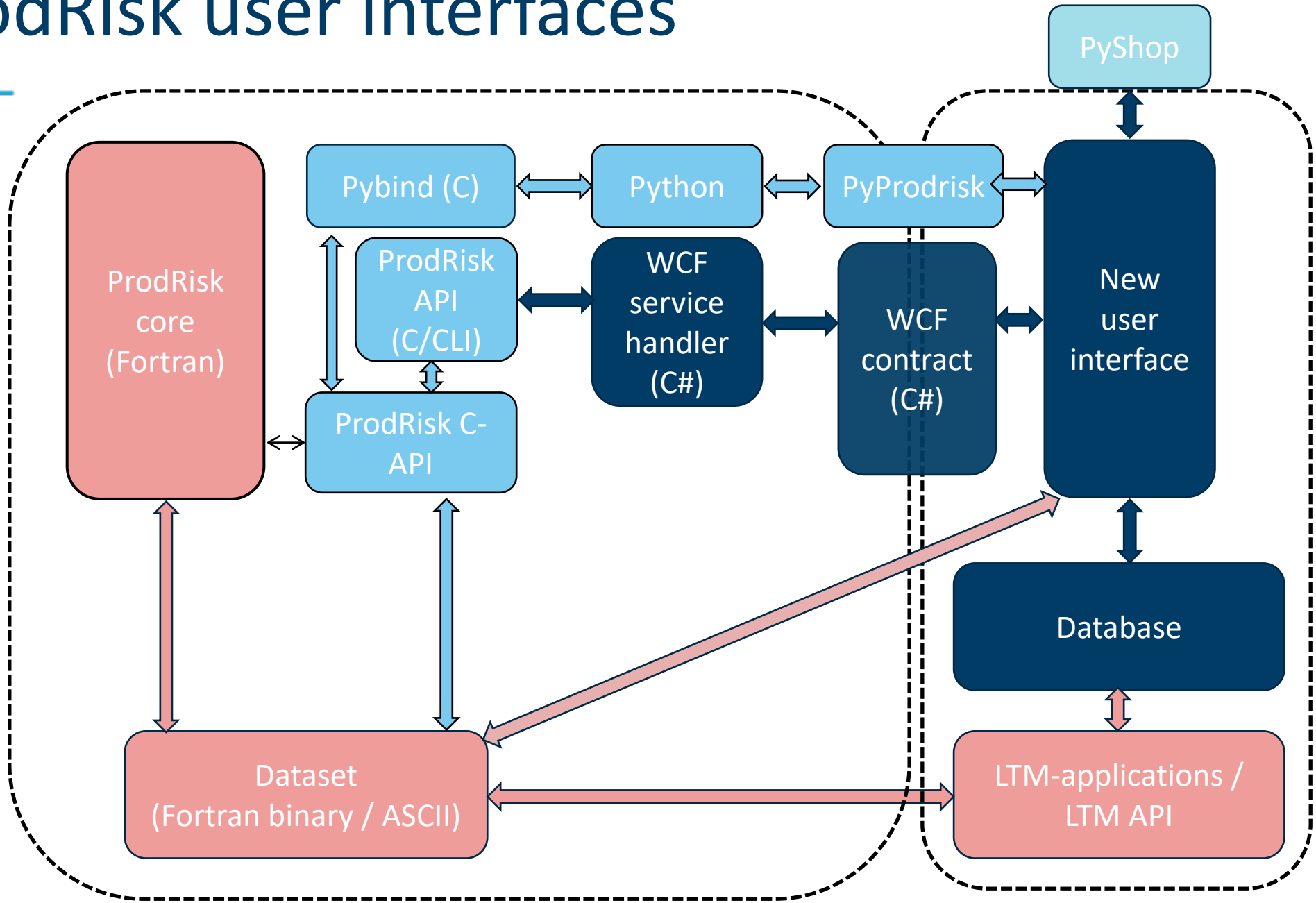
ProdRisk API

```python
from pyprodrisk import ProdriskSession

# --- create a new session ---

prodrisk = ProdriskSession(license_path='', silent=False, log_file='')

# --- configure settings for the session ---

prodrisk.set_optimization_period(
    pd.Timestamp("2021-01-04"),
    n_weeks = 156
)
```

```python
mod.refVol.set(pd.Series(name=0.0,
                         index = [prodrisk.start_time + pd.Timedelta(weeks=i) for i in [0,18,24,52,70,76,104,122,128,156]],
                         data=[45.0, 10.0, 90.0, 100.0, 10.0, 90.0, 100.0, 10.0, 90.0, 100.0]))


mod.PQcurve.set(pd.Series(name=50.0, index=[0, 20.0, 40.0, 80.0], data=[0, 20.0, 30.0, 65.0]))
mod.energyEquivalentConst.set(0.341)
mod.maxDischargeConst.set(65.0)
mod.maxProd.set(80.0)
mod.maxBypassConst.set(10000.0)
mod.topology.set([0, 0, 0])

mod.startVol.set(90.0)
```

```python
rsv_vols = mod.reservoirVolume.get()

ltm_res.plot_percentiles(rsv_vols, "Volume [Mm3]", "", percentiles_limits=[0, 25, 50, 75, 100])
```

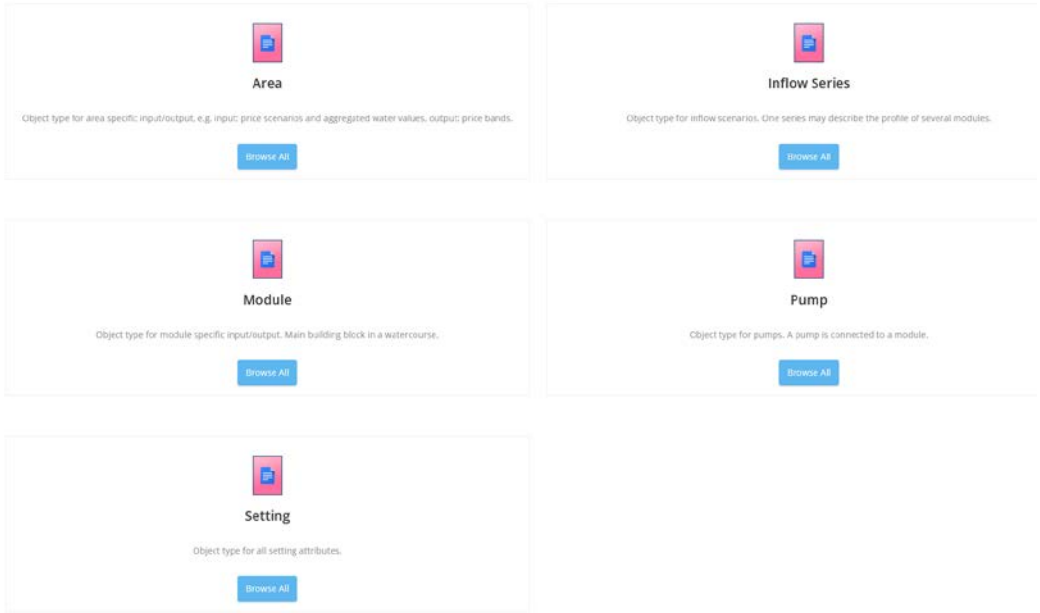
ProdRisk user interfaces



SINTEF

ProdRisk API input/output data

Objects



Attributes

Module

Attribute name	Data type	Unit	Time resolution	Size	Default values	Input Output	Description
maxVol	TXY	Mm3	weekly	nSimWeeks		(I)	Time-dependent maximal volume
minVol	TXY	Mm3	weekly	nSimWeeks		(I)	Time-dependent minimal volume
refVol	TXY	%	weekly	nSimWeeks		(I)	Time-dependent reservoir guidline curves (for buffer reservoirs).
energyEquivalent	TXY	kWh/m3	weekly	nSimWeeks		(I)	Time-dependent total energy equivalent
maxDischarge	TXY	m3/s	weekly	nSimWeeks		(I)	Time-dependent maximal discharge
minDischarge	TXY	m3/s	weekly	nSimWeeks		(I)	Time-dependent minimal discharge
minBypass	TXY	m3/s	weekly	nSimWeeks		(I)	Time-dependent minimal water bypass
maxBypass	TXY	m3/s	weekly	nSimWeeks		(I)	Time-dependent maximal water bypass
rampingDown	TXY	Mm3/week	weekly	nSimWeeks		(I)	Maximal decrease in reservoir volume per week[1]
rampingUp	TXY	Mm3/week	weekly	nSimWeeks		(I)	Maximal increase in reservoir volume per week[1]
maintenance	TXY	MW	weekly	nSimWeeks		(I)	Capacity unavailable for power generation[2]
reservoirVolume	STXY	Mm3	weekly	nSimWeeks per scenario		O	Output reservoir volume
overflow	STXY	m3/s	weekly	nSimWeeks per scenario		O	Output overflow
localInflow	STXY	m3/s	weekly	nSimWeeks per scenario		O	Output inflow
production	STXY	MW	price period	nSimWeeks* nPricePeriods per scenario		O	Output production

SINTEF

Examples

- Input data from different sources
- Investment analysis
- Maintenance scheduling
- ProdRisk-SHOP simulator

Input data from different sources

```python
def build_prodrisk_model(LTM_input_folder, model_name, n_weeks=156, start_time="2030-01-07"):
    # INITIALIZE PRODRISK API #
    prodrisk = ProdriskSession()
    prodrisk.set_optimization_period(pd.Timestamp(start_time), n_weeks=n_weeks)

    # BUILD MODEL#

    add_inflow_series(prodrisk, LTM_input_folder, model_name)

    add_modules(prodrisk, LTM_input_folder, model_name)

    add_area_object(prodrisk, LTM_input_folder)

    add_settings(prodrisk, LTM_input_folder)

    set_start_vols(prodrisk, LTM_input_folder, model_name)

    add_module_restrictions(prodrisk, LTM_input_folder)

    return prodrisk
```

```python
def add_modules(prodrisk, data_dir, model_name):
    model_file = h5py.File(data_dir + "model.h5", 'r')

    system_path = 'hydro_data/' + model_name.upper()

    module_data = model_file.get(system_path + '/Module_data')
    for i in range(module_data.size):
        mod = prodrisk.model.module.add_object(module_data[i]['res_name'].decode("iso-8859-1"))

        mod.name.set(module_data[i]['res_name'].decode("iso-8859-1"))
        mod.plantName.set(module_data[i]['plant_name'].decode("iso-8859-1"))
        mod.number.set(module_data[i]['res_id'])
        mod.ownerShare.set(module_data[i]['owner_share'])
        mod.regulationType.set(module_data[i]['reg_res'])
```

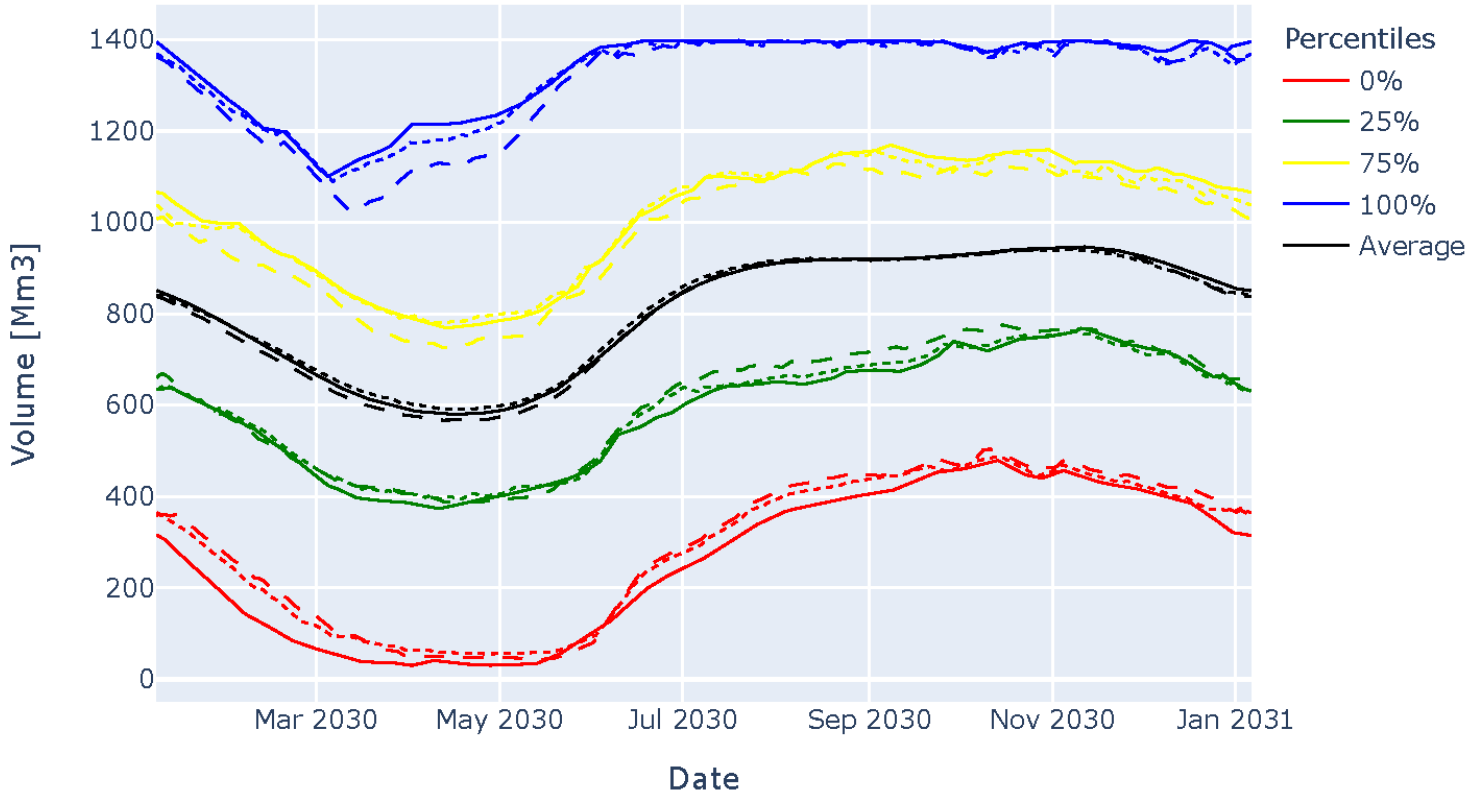
```python
def add_area_object(prodrisk, LTM_input_folder):
    area = prodrisk.model.area.add_object("my_area")

    price = get_price_from_some_source()

    area.price.set(price)

    water_values = get_water_values(LTM_input_folder, prodrisk.n_weeks)
    area.waterValue.set(water_values)

    return True
```

Investment analysis

```
prodrisk = ltm_input.build_prodrisk_model(LTM_input_folder, model_name, ADJUST_PRICE_INPUT, ADJUST_INFLOW, n_weeks=n_weeks)

add_upgrades(prodrisk, case_name)

prodrisk.run()
```

```
def add_upgrades(prodrisk, case_name):

    if case_name == "case1":
        add_case_1(prodrisk)
    elif case_name == "case2":
        add_case_2(prodrisk)
    elif case_name == "case3":
        add_case_3(prodrisk)
```

ProdRisk-SHOP simulator



What's missing?

- Functionality
  - Market data.
- Possibility to run EOPS on the same dataset
  - Water values as end value setting.
  - Used for maintenance scheduling.

SINTEF

Technology for a better society