

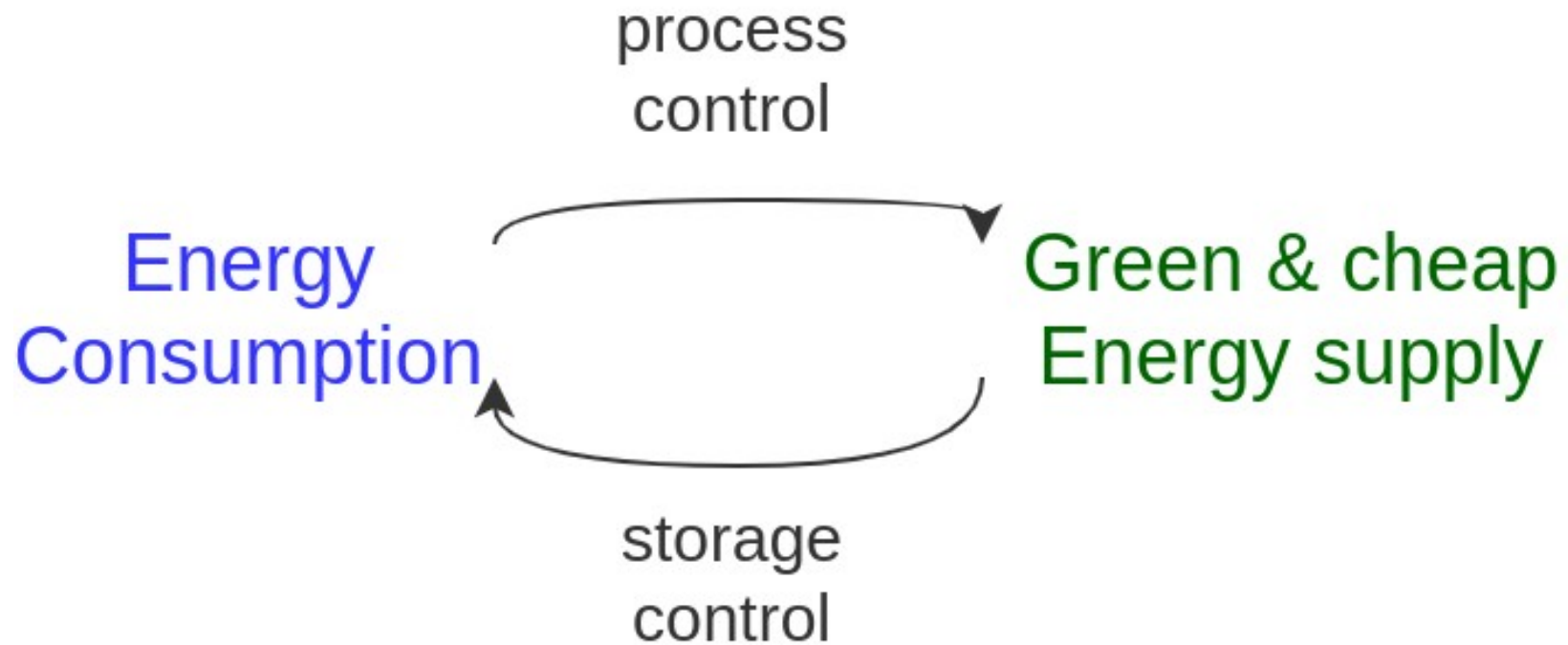
FlexMeasures Technical Steering Committee

February 15, 2024

Agenda

- Welcome & Short introduction to FlexMeasures
- What's new in v0.19?
- Roadmap for v0.20 — rate limiting? What more?
- Q&A

The matching challenge



FlexMeasures is the intelligent & developer-friendly EMS to support real-time energy flexibility apps.

Go green in daily operations, stay in control.

- Smart industry
- Smart city

FlexMeasures - simple



Use case: SteerOnCO₂ at Rijnland Water Board

We help water board Rijnland to only run their centrifuges for sludge dehydration when the CO₂ footprint in the grid is low.



Use case: SteerOnPrice & SteerOnSolar at V2G@Home

We optimize (dis)charging of Nissan Leaf cars with Wallbox chargers to save costs and use solar power, with zero user interaction needed.



Version 0.19 is (almost) out

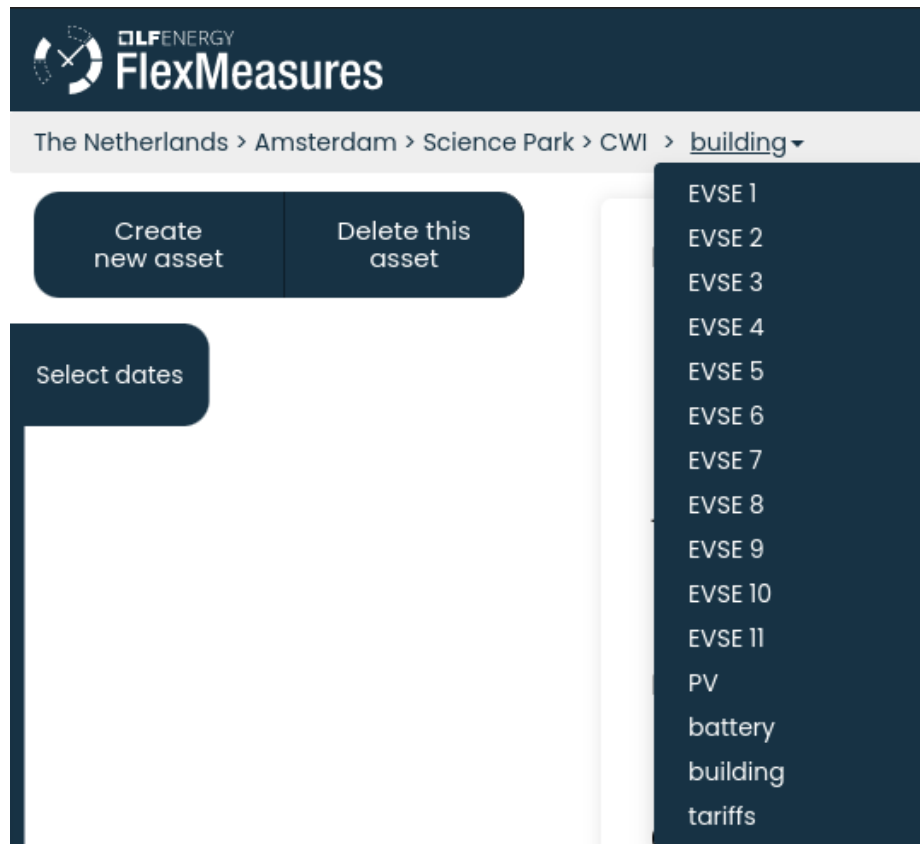
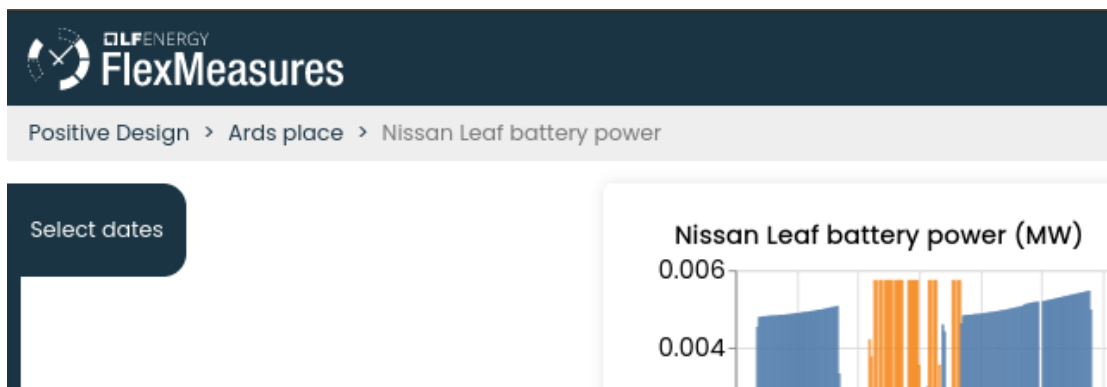
- Asset structure visualization
- SoC-minima and maxima as ranges
- Support Flask 2.3
- Support for ARM-based processors (e.g. Macbooks)
- Faster database queries of time series data
- Support SQLAlchemy 2.0

https://docs.sqlalchemy.org/en/20/changelog/migration_20.html

<https://blog.miguelgrinberg.com/post/what-s-new-in-sqlalchemy-2-0>



Asset structure visualization



Asset structure visualization

All sensors for Large Office Building

Show records

Filter records:

Name	Unit	Resolution	Entity address
Solar 1	kWh	15 minutes	ea1.2024-02.localhost.fml.I5276
Solar 2	kWh	15 minutes	ea1.2024-02.localhost.fml.I5277
SOC Target 1	kWh	15 minutes	ea1.2024-02.localhost.fml.I5278
SOC Target 2	kWh	15 minutes	ea1.2024-02.localhost.fml.I5279
SOC Target 3	kWh	15 minutes	ea1.2024-02.localhost.fml.I5280
SOC Target 4	kWh	15 minutes	ea1.2024-02.localhost.fml.I5281
SOC Target 5	kWh	15 minutes	ea1.2024-02.localhost.fml.I5282
Departure 1	s	15 minutes	ea1.2024-02.localhost.fml.I5283
Departure 2	s	15 minutes	ea1.2024-02.localhost.fml.I5284
Departure 3	s	15 minutes	ea1.2024-02.localhost.fml.I5285

Showing 1 to 10 out of 14 records

Previous

1

2

Next

All child assets for Large Office Building

Show records

Filter records:

Name	Location	Asset ID	Account	Sensors
baseline	LAT: 52.3780 LONG: 4.8971	4552	Toy Account	11
50kW battery	LAT: 52.3780 LONG: 4.8971	4584	Toy Account	11
smart battery	LAT: 52.3780 LONG: 4.8971	4616	Toy Account	11
curtallment	LAT: 52.3780 LONG: 4.8971	4568	Toy Account	11

SoC-minima and maxima as ranges

```
{
  "flex-model": {
    "soc-maxima": [
      {
        "value": 50,
        "datetime": "2024-02-05T11:00:00+01:00",
      },
      {
        "value": 51,
        "start": "2024-02-05T12:00:00+01:00",
        "end": "2024-02-05T13:30:00+01:00"
      },
      {
        "value": 52,
        "start": "2024-02-05T15:00:00+01:00",
        "duration": "PT1H"
      },
      {
        "value": 53,
        "duration": "PT15M",
        "end": "2024-02-05T18:30:00+01:00"
      }
    ]
  }
}
```

WIP: more helpful docs

Getting started
Get in touch
FlexMeasures Changelog

CONCEPTS

Algorithms
Security aspects
Device scheduler

TUTORIALS

Installation & First steps
Toy example: Introduction and setup
Toy example I: Scheduling a battery, from scratch
Toy example II: Adding solar production and limited grid connection
Toy example III: Computing schedules for processes
Toy example IV: Computing reports
Posting data
Forecasting & scheduling
Building custom UIs

THE IN-BUILT UI

Dashboard
Assets & data
Administration

THE API

API Introduction
Notation
Vehicle-to-grid
Version 3.0
Developer API

A quick glance

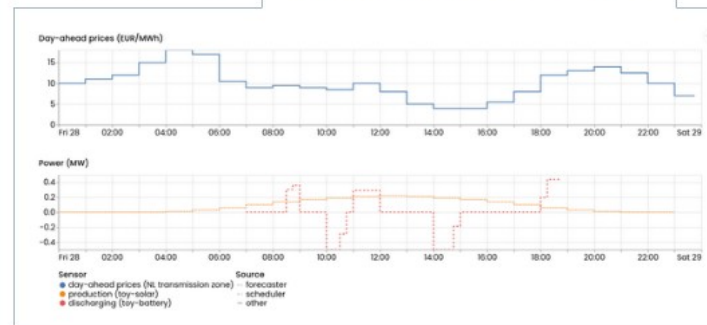
Schedules

Example code

The main purpose of FlexMeasures is to create optimized schedules. Let's have a quick glance how that looks like in the UI:

Battery optimized by price

Battery optimized by price, constrained by solar



A short explanation: This battery is optimized to buy power cheaply and sell it at expensive times - the red-dotted line is what FlexMeasures computed to be the best schedule, given all knowledge (in this case, the prices shown in blue). However, in the example on the right the battery has to store local solar power as well (orange line), which constrains how much it can do with its capacity (that's why the schedule is limited in capacity and thus cycling less energy overall than on the left).

Want to read more about the examples shown above? We discuss this in more depth at [Toy example I: Scheduling a battery, from scratch](#) and the tutorials that build on that.

What FlexMeasures does

Main functionality

Interfacing with FlexMeasures

• API

FlexMeasures runs in the cloud (although it can also run on-premise if needed, for

Scheduling: State of the art

Done

- Storage: Battery
- Storage: Heat
- Processes

Next

- Sites (start: PR#855)
- Sector coupling
- Grid services
- Order books
- VPP

Ideas for version 0.20

Focus (currently):

- Rate limiting
- RelaxedStorageScheduler
- CLI: ability to provide dynamic SoC values

(see also <https://github.com/FlexMeasures/flexmeasures/milestone/44>)

Q&A

- What are you working on?
- What is unclear?

Roadmap – Big goals

- [2022 - mature] Model & pilot e-mobility optimization (price-based, V2G)
- [2023 - started] Model & pilot heating optimization (price-based, also with heat buffers)
- [2023] Sector coupling (optimize e-mobility and heating in one site)
- [2024] Congestion support (e.g. for DSOs in GOPACS)
- [2024] VPP (optimize multiple sites towards one market)

Resources – do get in touch!

- <https://github.com/FlexMeasures/flexmeasures/>
- <https://www.flexmeasures.io>
- <https://lists.lfenergy.org/g/flexmeasures>
- <https://fosstodon.org/@flexmeasures>
- LF Energy Slack: #flexmeasures

FlexMeasures - integration

