FlexMeasures Technical Steering Committee

June 22 2023

Agenda

- Welcome & Short introduction to FlexMeasures
- The latest release: v0.14
- A quick tour of the new reporting architecture
- The new FlexMeasures-Client (Python)
- An update on current work on heating optimization
- Q&A

LF ENERGY FlexMeasures

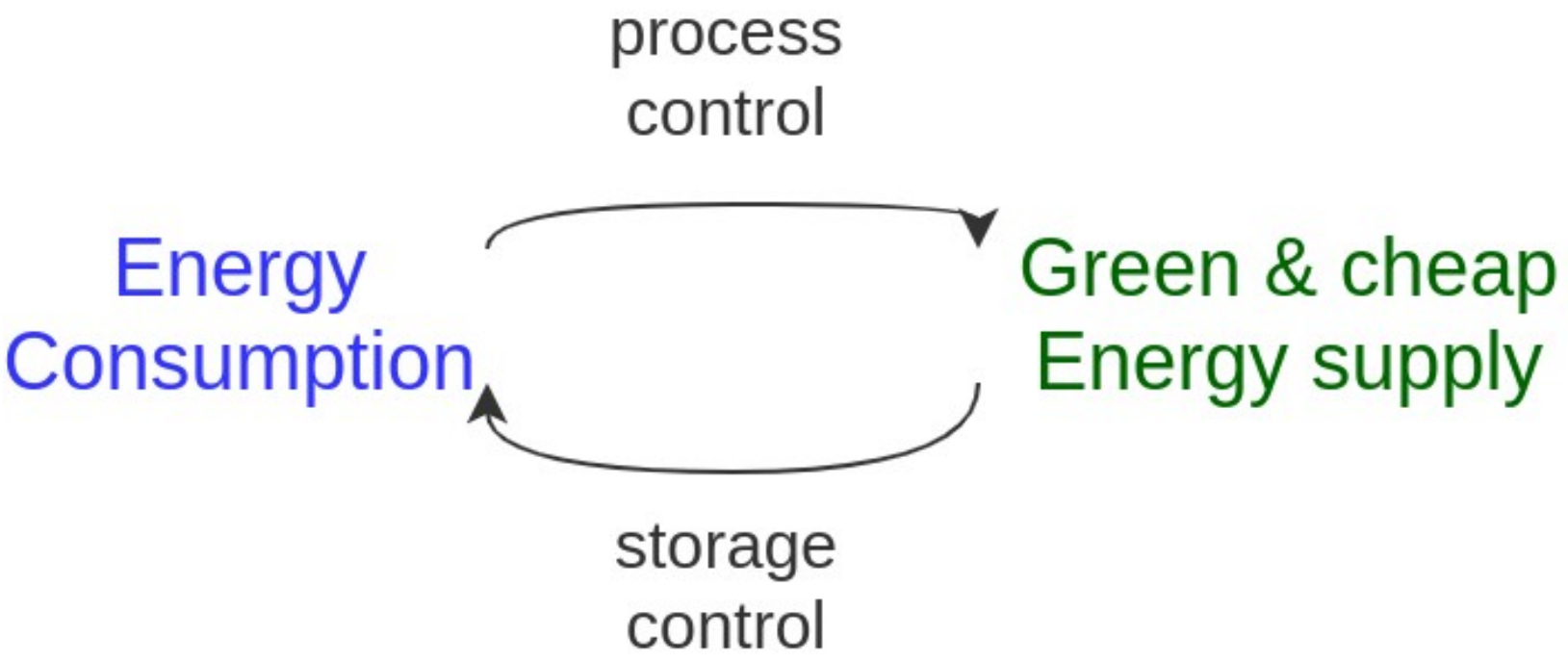
FlexMeasures is the intelligent & developer-friendly EMS to support real-time energy flexibility apps.

Go green in daily operations, stay in control.

- Smart industry
- Smart city

LF ENERGY FlexMeasures

The matching challenge



LF ENERGY
FlexMeasures

FlexMeasures - simple

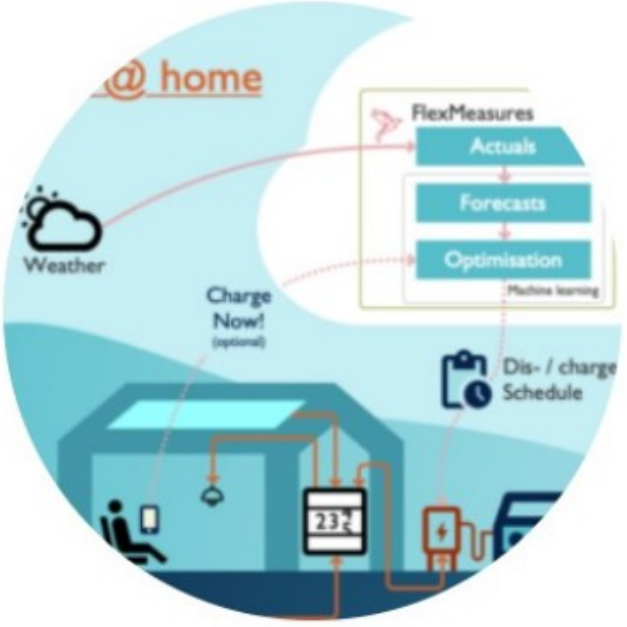
LFENERGY
FlexMeasures

Use case: SteerOnCO_2 at Rijnland Water Board

We help water board Rijnland to only run their centrifuges for sludge dehydration when the CO_2 footprint in the grid is low.

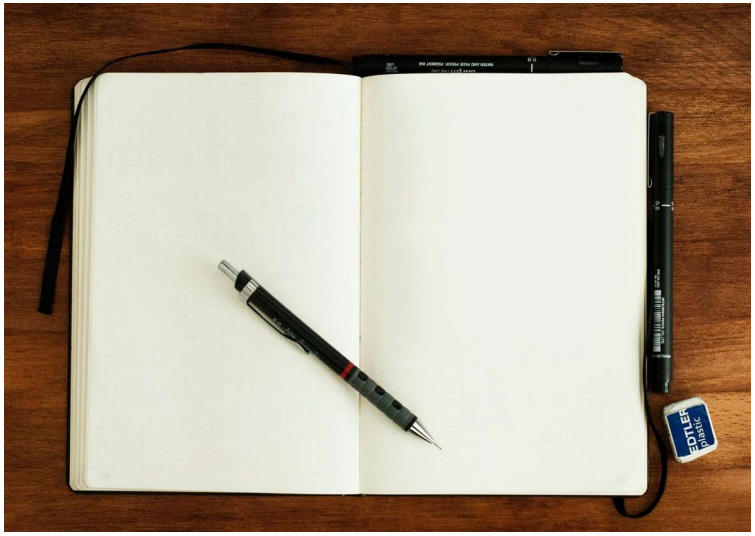
Use case: SteerOnPrice & SteerOnSolar at V2G@Home

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



Version 0.14 is out

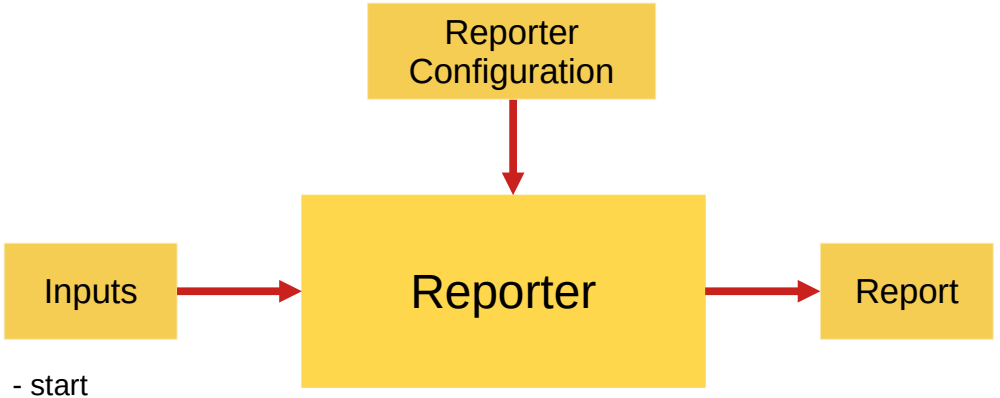
- new reporting capabilities
- begin supporting optimization of heat storage
- a bunch of developer support features



https://flexmeasures.io/014-reporting-power/

LF ENERGY FlexMeasures

The new reporting architecture



- start
- end
- resolution
…

Classes

- Reporter
- PandasReporter
- TibberReporter
- AggregatorReporter

Benefits:
- minimize code repetition
- propagate code re-use
- standard interface for data computations

Commands:

```
flexmeasures add report
flexmeasures show reporters
```

```
result.head()
                                                                                                  event_value
event_start               belief_time                       source       cumulative_probability
2023-04-13 00:00:00+00:00 2023-06-22 22:53:14.825937+09:00 FlexMeasures 0.5                          292.0
2023-04-13 01:00:00+00:00 2023-06-22 22:53:14.825937+09:00 FlexMeasures 0.5                          277.0
2023-04-13 02:00:00+00:00 2023-06-22 22:53:14.825937+09:00 FlexMeasures 0.5                          270.0
2023-04-13 03:00:00+00:00 2023-06-22 22:53:14.825937+09:00 FlexMeasures 0.5                          262.0
2023-04-13 04:00:00+00:00 2023-06-22 22:53:14.825937+09:00 FlexMeasures 0.5                          260.0
sensor: TibberReportSensor, event_resolution: 1:00:00
```

Next steps:
- YAML
- Reporters are DataSources
- Separate reporters from reports

LF ENERGY FlexMeasures

AggregatorReporter

Sum the power flows of the PV (sensor #1) and consumption (sensor #2).

```
{
    "beliefs_search_configs": [
        {
            "sensor": 1,
            "source" : 1,
            "alias" : "pv"
        },
        {
            "sensor": 2,
            "source" : 2,
            "alias" : "consumption"
        }
    ],
    "method" : "sum",
    "weights" : {
        "pv" : 1.0,
        "consumption" : -1.0
    }
}
```

PandasReporter

Sum the values of sensors 1 & 2 for each time period and resample to 2h resolution.

```
{
  "beliefs_search_configs" : [
    {
      "sensor" : 1,
      "event_starts_after" : "2023-04-10T00:00:00 00:00",
      "event_ends_before" : "2023-04-10T10:00:00 00:00",
    },
    {
      "sensor" : 2,
      "event_starts_after" : "2023-04-10T00:00:00 00:00",
      "event_ends_before" : "2023-04-10T10:00:00 00:00",
    },
  ],
  "transformations" : [
    {
      "df_input" : "sensor_1",
      "df_output" : "sensor_1_source_1",
      "method" : "xs",
      "args" : ["@source_1"],
      "kwargs" : {"level" : 2},
    },
    {
      "df_input" : "sensor_2",
      "df_output" : "sensor_2_source_1",
      "method" : "xs",
      "args" : ["@source_1"],
      "kwargs" {"level" : 2},
    },
    {
       "df_output" : "df_merge",
        "df_input" : "sensor_1_source_1",
        "method" : "merge",
        "args" : ["@sensor_2_source_1"],
        "kwargs" : {"on" : "event_start", "suffixes" : ("_sensor1", "_sensor2")},
    },
    {
      "method" : "resample", "args" : ["2h"]
    },
    {
      "method" : "mean"
    },
    {
      "method" : "sum",
      "kwargs" : {"axis" : 1}
    }
  ],
  "final_df_output" : "df_merge",
}
```

The new flexmeasures-client

An async client to communicate with the Flexmeasures API and save local EMS developers time

Current functionality:

- Authentication
- Listing available sensors
- Listing assets
- Adding measurements to sensors
- Triggering and retrieving schedules

Built-in behaviour:

- Polling (e.g. while Flexmeasures is calculating schedules)
- Automatic reauthorization when the access token expires

Next steps:

- creating assets, sensors
- logging and improved error handling

LF ENERGY FlexMeasures

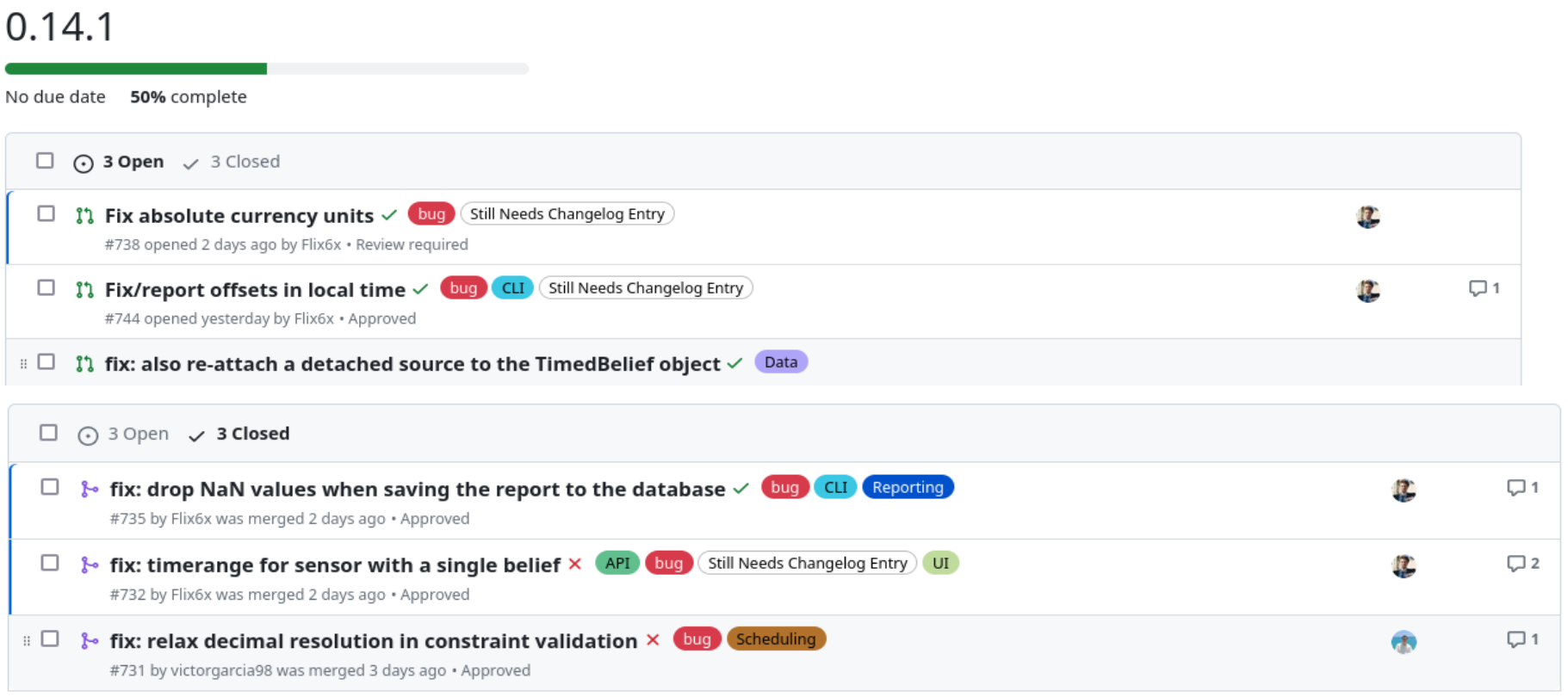
```python
import asyncio

from flexmeasures_client.client import FlexMeasuresClient as Client

EMAIL = "admin@admin.nl"
PASSWORD = "admin"


client = Client(
    email=EMAIL,
    password=PASSWORD,
    host="localhost:5000",
)


async def my_script():
    await client.trigger_and_get_schedule(
        sensor_id=1,
        start="2023-06-18T10:00:00+00:00",
        duration="PT45M",
        soc_unit="MWh",
        soc_at_start=50,
        soc_targets=[
            {
                "value": 100,
                "datetime": "2023-06-20T11:00+02:00",
            }
        ],
        consumption_price_sensor=3,
    )

asyncio.run(my_script())

```

Ongoing: heating optimization

- Storage scheduler supports losses over time (#679) → new storage-efficiency parameter
- S2 protocol implementation in FlexMeasures client
- Multiple maxima and minima constraints in scheduling (#680) → usage forecasts can be modelled for scheduling (heat) buffers

LF ENERGY FlexMeasures

Version 0.14.1 is on the way



LF ENERGY FlexMeasures

Q&A

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

LF ENERGY FlexMeasures

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] Congestion support (e.g. for DSOs in GOPACS)
- [2023] Sector coupling (optimize e-mobility and heating in one site)
- [2024] VPP (optimize multiple sites towards one market)

Roadmap – projects

- [Q1 2023] **More powerful algorithm configurations**, to support more use cases and more custom situations (e.g. research). For scheduling, as well as for forecasting. [work has started in Q4 2022]
- [Q2 2023] **KPIs support** (e.g. reporting of daily totals), customizable
- [Q2 2023] **Scheduling algorithm for heat buffering**
- [Q2 2023] Allow for **annotations on time series**, e.g. to model processes and operator feedback. [work has started in 2022]
- [Q4 2023] **Build out the flexibility modelling,** by supporting Fraunhofers Energy Flexibility Data Model (EDFM) and TNO's S2
- [tbd] **Authorization model for allowing "super-accounts"** to manage other accounts (e.g. for ESCos) or add data to them (e.g. meter data companies).
- [tbd] **Smarter monitoring.**
- [tbd] **Scheduler compatible with ShapeShifter** (based on USEF flex trading protocol)
- [tbd] **Better plotting support** (via API/vega-lite), for plugins to define their own plots which are then made available in the FlexMeasures API (usable in custom frontends).
- [tbd] Better tooling to **work well at scale** (e.g. support load balancing, db sharding etc). Also using Docker to scale up more flexibly (e.g. in Kubernetes).

LF ENERGY FlexMeasures

https://flexmeasures.io/roadmap/

Resources – do get in touch!

- https://github.com/FlexMeasures/flexmeasures/
- https://www.flexmeasures.io
- https://lists.lfenergy.org/g/flexmeasures
- https://twitter.com/flexmeasures
- LF Energy Slack: #flexmeasures

LF ENERGY FlexMeasures

FlexMeasures - integration