FlexMeasures

Rapid & scalable energy flexibility services for/by ESCOs

LF Energy TAC meeting, 2 Nov 2021
"Flexibility is the grid’s ability to manage variability and volatility to balance supply and demand."

ACCENTURE

"Demand flexibility uses communication and control technology to shift electricity use across hours of the day."

ROCKY MOUNTAINS INSTITUTE
energy flex specialists

Where's your data?

Scaling is hard!

asset owners & operators

Who are you?
ESCOs are the bridges we need!

energy flex specialists

relationship data

asset owners & operators
Our customers:
Energy Service Companies (ESCOs)

Examples:
- Metering companies
- Real estate developers
- Microgrid developers
- Car charging station operators
- Business parks
- Energy cooperatives

Market size: USD 29 billion, growing 8% / year
ESCos & implicit distributed flexibility

- Production
- Process
- Emergency
- Generators
- Cooling
- Systems
- Heat Pump
- Solar
- Electric Vehicle
- Airco System
- Storage systems

Active Customer

ESCo

Supplier

DSO

control & advise

variable costs

variable costs
ESCos & explicit distributed flexibility
Many services?

The needs in energy flexibility service implementation change with:

- type of use
- type of customer
- sector
- connection/grid
- markets
- storage?
- culture!
- etc.
Energy flexibility services: crucial, but expensive.

ESCos want to become active, but fear vendor lock-in or high costs.
FlexMeasures: Design goal

Build real-time energy flexibility services, *rapidly* and *scalable*.
On top of open source.
How we use FlexMeasures as a middleware.

**Energy Flexibility Services**
- **GripOnGas**: Track avoidable gas consumption
- **E-Mission**: Reduce CO2 footprint of processes
- **V2G@Home**: EV charging living lab
- **BVP**: Balance portfolios

**Platform Middleware**
- FlexMeasures: ML models, API, security, plugin support, multi-tenancy, plotting, developer docs...

**Data integrations**
- **Metering Companies**
- **Weather Services**
- **Markets**
FlexMeasures value adds

1. Integrate data multiple times per day or hour:
   - Support for real-time updates
   - Forecasting for the upcoming hours
   - Schedule optimization

2. A data model to model uncertainty in forecasts and outcomes accurately.

https://github.com/SeitaBV/timely-beliefs

3. Reduce software development costs:
   - Well-documented API
   - Plugin support
   - Plotting support
   - Multi-tenancy
FlexMeasures in the LFE context

New target group:

ESCos

Possible integrations:

• OpenLEADDR
• ShapeShifter
• OpenEEMeter
Resources

- https://github.com/SeitaBV/flexmeasures/
- https://flexmeasures.readthedocs.io
- https://flexmeasures.io
- https://seita.nl/core-technology/flexmeasures/
- https://seita.nl/services/
Thank you.
Seita: journey & team

2016-2017: Academic spin-off
(initial idea: energy pricing)

2018-2019: Apply data skills as consultants,
first ESCO relationship.

2020: Open-source FlexMeasures

2021: First energy flexibility services,
second ESCO client.

2022: Scale 1st service, grow team,
start 2nd service

Nicolas Höning
- Web/Cloud engineering lead
- PhD in smart grid mechanisms
- Ex-data engineer @ Senfal / Vattenfal

Felix Claessen
- Data science lead
- Ex-smart grid researcher
- USEF expert
FlexMeasures and USEF

Already supported by FlexMeasures (with hands-on experience):

- Meter Data and Price Data (but also generally any type of Sensor Data)
- D-Prognoses (i.e. day-ahead meter data prognoses)
- UDI Events (description of available flexibility from individual devices)
- Device Messages (which tell devices what to do, usually in response to a UDI event)

Not yet officially supported by FlexMeasures (so far only simulations with these concepts):

- Flex Requests
- Flex Offers
- Flex Orders
- Flex Settlements
Plugins: Getting started

S cookiecutter https://github.com/SeitaBV/flexmeasures-plugin-template

plugin_name [Your plugin name, e.g. 'My Plugin']:
plugin_slug [a new service]:
module_name [a new service]:
description []: Providing flexible scheduling to X customers in region Y:
author_name []: Nicolas Höning
author_email []: nicolas@seita.nl
plugin_url []:
minimal_flexmeasures_version [0.7.0]:
api_blueprint [y]:
ui_blueprint [y]: n
cli_blueprint [y]:

S ls A\new\service
a_new_service Makefile README.md requirements run_mypy.sh setup.cfg setup.py

S ls A\new\service/a_new_service
api cli __init__.py

S cd A\new\service

S pytest
Test session starts [platform: linux, Python 3.8.10, pytest 6.2.4]
rerun: /home/nicolas/workspace/seita/My Plugin/A new service plugins sugar:0.9.4, requests-mock:1.9.3, flask:12.0, cov:2.12.1
collecting:
  a_new_service/api/tests/test_api.py ✓
50% [≈] 1/2 passed
  a_new_service/cli/tests/test_cli.py ✓
100% [≈] 1/1 passed

Results (0.07s):
  2 passed
Quickstart

This section walks you through getting FlexMeasures to run with the least effort. We'll cover making a secret key, connecting a database and creating one user & one asset.

Note

Are you not hosting FlexMeasures, but want to learn how to use it? Head over to our tutorials, starting with posting data.

Install FlexMeasures

Install dependencies and the platform itself:

```
pip install flexmeasures
```
Code hygiene

- require higher pip-tools version
  - deploy-to-staging #141: Commit 2a6a3c7b pushed by rhromeing

- require higher pip-tools version
  - lint-and-test #795: Commit da23b45b pushed by rhromeing

- Enable plugin list to be an app creation param, use...
  - deploy-to-staging #140: Commit d3d550b9 pushed by rhromeing

- Enable plugin list to be an app creation param, use...
  - lint-and-test #794: Commit 3c2b000f pushed by rhromeing

- do not rely on a actual secret_key file in testing / CI
  - lint-and-test #793: Commit c03a6c7f pushed by rhromeing

- add changelog entry
  - lint-and-test #791: Commit 2e5d7a44 pushed by rhromeing

- document caveats when testing plugins
  - lint-and-test #791: Commit 200954b2 pushed by rhromeing

- Prepare changelogs for v0.6.1 release
  - lint-and-test #790: Commit 37b4467d pushed by Piotr

- Backport PR #127: Add release date (#127)
  - lint-and-test #789: Commit 3a38998d pushed by Piotr

- get_or_create_source might create data source wit...
  - lint-and-test #788: Commit 64b2a0f8 pushed by Piotr

- flake8

- Black

- mypy
What if we could build for energy flexibility, what WordPress has become for web publishing?

A technology to raise the standard by which every small ESCo in the world can approach this problem.
Our business model: Subscriptions via SaaS

- **Status:**
  - 1st ESCo partner (25K end customers)
  - 1st end customer paying subscriptions