Horizon 2020 – LCE-2017 - SGS

FLEXCoop

Democratizing energy markets through the introduction of innovative flexibility-based demand response tools and novel business and market models for energy cooperatives

WP6 – Semantically Enhanced DER Registry and Open Marketplace for Flexibility Sharing

D6.5 – FLEXCoop Semantically Enhanced DER Registry – Final Version

Due date: 31.05.2020 **Delivery Date:** 17.08.2020

Author(s): Karsten Isakovic (Fraunhofer), Peter Hasse (Fraunhofer)

Editor: Karsten Isakovic (Fraunhofer)

Lead Beneficiary of Deliverable: Fraunhofer

Contributors: Fraunhofer, Hypertech, Koncar

Dissemination level: Public **Nature of the Deliverable:** Demonstrator

Internal Reviewers: Eloi Gabaldon (CIMNE), Andreas Muñoz (CIRCE)

FLEXCOOP KEY FACTS

Topic:	LCE-01-2016-2017 - Next generation innovative technologies enabling smart grids, storage and energy system integration with increasing share of renewables: distribution network
Type of Action:	Research and Innovation Action
Project start:	01 October 2017
Duration:	36 months from **01.10.2017** to **30.09.2020** (Article 3 GA)
Project Coordinator:	Fraunhofer
Consortium:	13 organizations from nine EU member states

FLEXCOOP CONSORTIUM PARTNERS

Fraunhofer	Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.
ETRa	ETRA INVESTIGACION Y DESARROLLO SA
HYPERTECH	HYPERTECH (CHAIPERTEK) ANONYMOS VIOMICHANIKI
DTU	DANMARKS TEKNISKE UNIVERSITET
CIRCE	FUNDACION CIRCE CENTRO DE INVESTIGACION DE RECURSOS Y CONSUMOS ENERGETICOS
KONCAR	KONCAR - INZENJERING ZA ENERGETIKUI TRANSPORT DD
SUITE5	SUITE5 DATA INTELLIGENCE SOLUTIONS Limited
S5	SUITE5 DATA INTELLIGENCE SOLUTIONS Limited
CIMNE	CENTRE INTERNACIONAL DE METODES NUMERICS EN ENGINYERIA
RESCOOP.EU	RESCOOP EU ASBL
SomEnergia	SOM ENERGIA SCCL
ODE	ORGANISATIE VOOR HERNIEUWBARE ENERGIE DECENTRAAL
Escozon	ESCOZON COOPERATIE UA - affiliated or linked to ODE
MERIT	MERIT CONSULTING HOUSE SPRL

EXECUTIVE SUMMARY

This demonstrator deliverable describes the design and implementation of the final version of the FLEXCoop Open and Semantically Enhanced DER Registry.

The scope of this deliverable is a backend component which has no user interface by itself, the provided services are used by several other FLEXCoop components like the Aggregator Visualisation Toolkit, the Prosumer Visualisation Toolkit, the Local Demand Manager, Flexibility Forecasting and the Open Marketplace. As central service the DER Registry has been integrated as module into the FLEXCoop Middleware component developed in Task 4.3.

The FLEXCoop Open and Semantically Enhanced DER Registry allows to create, edit and query data on Distributed Energy Resources (DER) using a RESTful API. The Secure Access Control (SEAC) framework developed in Task 4.4 is applied to limit the access to authenticated users with fine granular access control rules based on the user id / role.

In order to test and showcase the functionalities of the marketplace and to obtain a more condensed demonstration environment, a minimal user interface component was developed and is made available as part of this deliverable.

Table of Contents

FLEXCOOP KEY FACTS **2**
FLEXCOOP CONSORTIUM PARTNERS **2**
EXECUTIVE SUMMARY **3**
LIST OF FIGURES **5**
LIST OF TABLES **5**
ABBREVIATIONS **6**
1. INTRODUCTION **7**
2. SOFTWARE DEMONSTRATOR MAIN INFORMATION **7**
3. RELEVANT LICENCES USED IN THE DEMONSTRATOR **7**
4. OVERVIEW OF THE PROTOTYPE/ABSTRACT **8**
5. PROGRAMMING LANGUAGE **9**
6. CONTENTS OF THE CURRENT RELEASE **9**
7. SOURCE CODE OF THE RELEASE **9**
8. RELATED DOCUMENTATION **9**
8.1. DER REGISTRY MIDDLEWARE MODULE 9
8.2. DEMONSTRATOR WEB INTERFACE 11
9. INSTALLATION GUIDE **11**
10. INTERFACES WITH OTHER COMPONENTS AND THEIR INTEROPERABILITY **12**
10.1. RESTFUL API 12
10.1.1. Data Validation *12*
10.1.2. Data Queries *13*
10.1.3. Build-In Endpoint Documentation *14*
10.2. PROSUMER TOOLKIT 15
10.3. AGGREGATOR TOOLKIT 16
11. REQUIREMENTS COVERAGE **16**
12. DEVELOPMENT AND INTEGRATION STATUS **17**
13. CONCLUSION **17**
APPENDIX A: LITERATURE **18**
APPENDIX B: REQUIREMENTS **18**

LIST OF FIGURES

Figure 1: Component integration overview for Demonstrator D6.5 8

Figure 2: oAuth Providers and Tokens 10

Figure 3: Cerberus JSON schema definition for DER endpoint 13

Figure 4: SwaggerUI DER Registry API documentation 14

Figure 5: Prosumer Toolkit: Device List 15

Figure 6: Prosumer Toolkit: Device Consumption and Flexibility 15

Figure 7: Aggregator Toolkit: Device List 16

LIST OF TABLES

Table 1: Requirements Coverage 16

Table 2: Development and integration status 17

ABBREVIATIONS

AGR	AGGREGATOR
API	Application programming interface
CIM	Common Information Model
D	Deliverable
DER	Distributed Energy Resources
DR	Demand-Response
LDEM	Local Demand Manager
OSB	Open Smart Box
OpenADR	Open Automated Demand Response
REST	Representational State Transfer
SEAC	Security Access Control
T	Task
UI	User Interface
URI	Uniform Resource Identifier
VPP	Virtual Power Plant
WP	Work Package

1. INTRODUCTION

The deliverable provides the design and respective tools for publishing and advertising available DERs by the prosumer along with their capabilities (geographical information, power characteristics, etc), while enabling the sub-sequent discovery and introduction in VPP configuration by the Aggregators/Energy Cooperatives. The deliverable is the final outcome of the Task 6.1 “Open and Semantical Enhanced DER Registry”.

2. SOFTWARE DEMONSTRATOR MAIN INFORMATION

Title/Designation

- D6.5 FLEXCoop Semantically Enhanced DER Registry – Final Version

Versioning/Version

- Release Version 1.0

Release Date

- 31.05.2020

3. RELEVANT LICENCES USED IN THE DEMONSTRATOR

- PYTHON 3.7.4 - PSF LICENSE - http://docs.python-eve.org/en/latest/license.html
- Python Eve 0.9.2 – BSD License - http://docs.python-eve.org/en/latest/license.htmlc
- Python Eve Swagger 0.0.11 - https://github.com/pyeve/eve-swagger/blob/master/LICENSE
- Python flask 1.0.2 - https://flask.palletsprojects.com/en/1.1.x/license/
- Python Cerberus 1.2 – ISC License - http://docs.python-cerberus.org/en/stable/license.html
- Python PyMongo 3.8.0 - Apache Software License (Apache License, Version 2.0)
- MongoDB 4.0.10 - AGPL v3.0 - https://www.mongodb.com/community/licensing

Further dependencies of the software mentioned above can be extracted from the section requirements in the appendix.

4. OVERVIEW OF THE PROTOTYPE/ABSTRACT

The DER Registry module for the FLEXCoop Middleware implements the DER Registry related logic as a module for the FLEXCoop Middleware developed in T4.3[1]. The DER Registry enables the Aggregators to discover the devices and their data published by the Prosumers. The SEAC framework developed in T4.4[2] is applied by the middleware and extended in the DER Registry module to allow other components of the FLEXCoop solution secure access of the DER Registry related information stored in the Middleware.

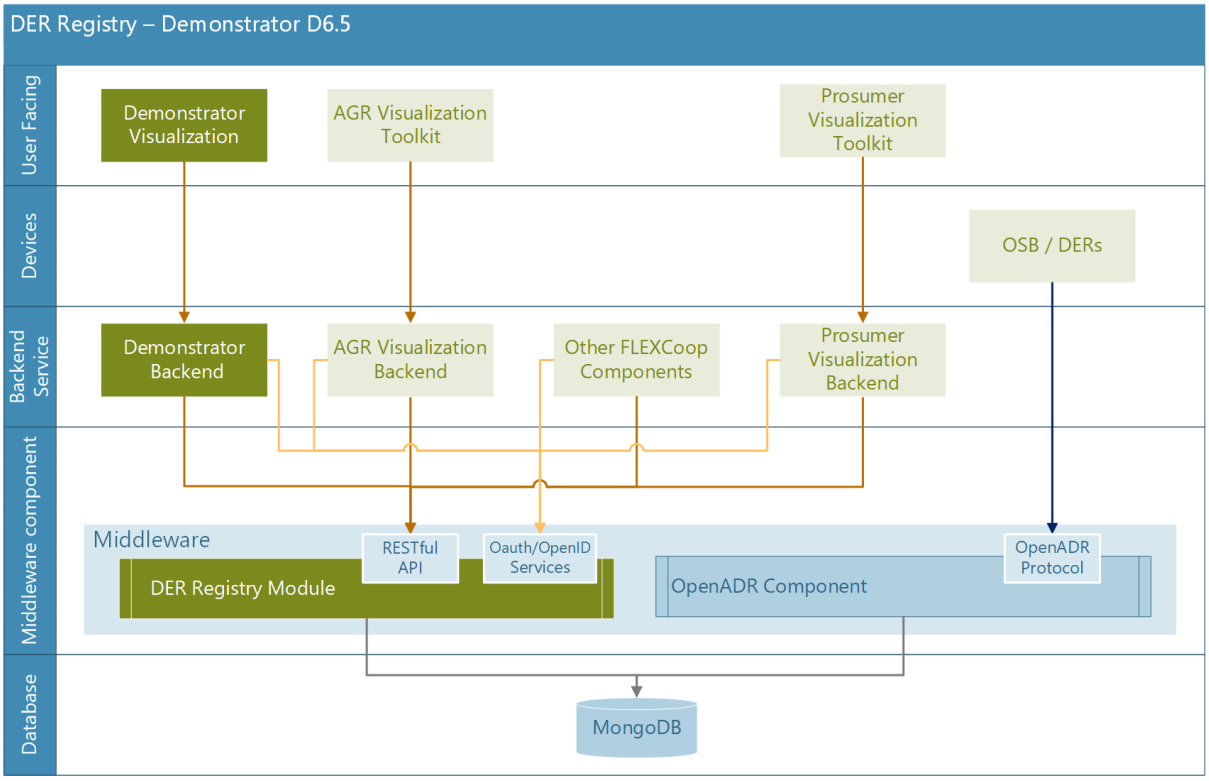

Figure 1: Component integration overview for Demonstrator D6.5

The block diagram in Figure 1 illustrates the interplay of the DER Registry module with other FLEXCoop system components. Those components implemented within the scope of this deliverable are highlighted with a green background.

Distributed Energy Resources installed at the Prosumers facilities are connected to the FLEXCoop Open Smart Box (OSB) described in D4.6 [2] . The OSB uses the Open Automated Demand Response (OpenADR) protocol to communicate device states and device registration to the Middleware OpenADR component described in D4.8 [4].

[1] Task 4.3: Middleware Configuration and Big Data Analytics Platform

[2] Task 4.4: Information Security, Access Control and Data Privacy Mechanism

Registering new devices creates new database entries in the main MongoDB of the FLEXCoop Middleware. These new device entries are then picked up and handled by the DER Registry module to make them available for authenticated parties via RESTFul API.

To realize a self-contained demonstrator showcasing the functionalities of the DER Registry, a simple web interface (UI) was developed that showcases listing and searching of devices.

5. PROGRAMMING LANGUAGE

The demonstrator as well as the underlying FLEXCoop component are developed in Python 3.7.x.

6. CONTENTS OF THE CURRENT RELEASE

The implemented components covered by this deliverable realize a self-contained demonstrator showcasing the DER Registry interactions using its RESTful API. The supplemental web interface allows simple interaction with the DER registry aligned with the use cases of D2.9 [1].

7. SOURCE CODE OF THE RELEASE

The source code of the release is available from the FLEXCoop GIT Repository using the following addresses (user account required):

- https://gitlab.fokus.fraunhofer.de/FlexCoop/middleware_rest/tree/master/modules/devices
- https://gitlab.fokus.fraunhofer.de/FlexCoop/demonstrator_admin_view

8. RELATED DOCUMENTATION

This chapter describes the data privacy management implemented in the DER Registry module and the supplemental demonstrator web interface.

8.1. DER Registry Middleware Module

The DER Registry is a module in the FLEXCoop Middleware component that is described in D4.8 [4]. The Middleware is based on the Python EVE framework providing the RESTful API and an extension for oAuth/OpenID services for security access control (SEAC) described in D4.9 [5].

All RESTful API request to the Middleware need to be authenticated by a valid oAuth token signed by one of the FLEXCoop oAuth providers.

The block diagram of Figure 2 shows several users accessing the FLEXCoop system through either the Aggregator or Prosumer Toolkit [7][9] and the associated oAuth providers of the aggregators that are used for user authentication. The digital signed oAuth tokens generated during login is stored in the Toolkits backend components and all requests of the backends to the Middleware resulting from user interaction will contain the users oAuth token.

The right side of the diagram illustrates that FLEXCoop components acquire the mandatory oAuth token from a dedicated Middleware oAuth provider component.

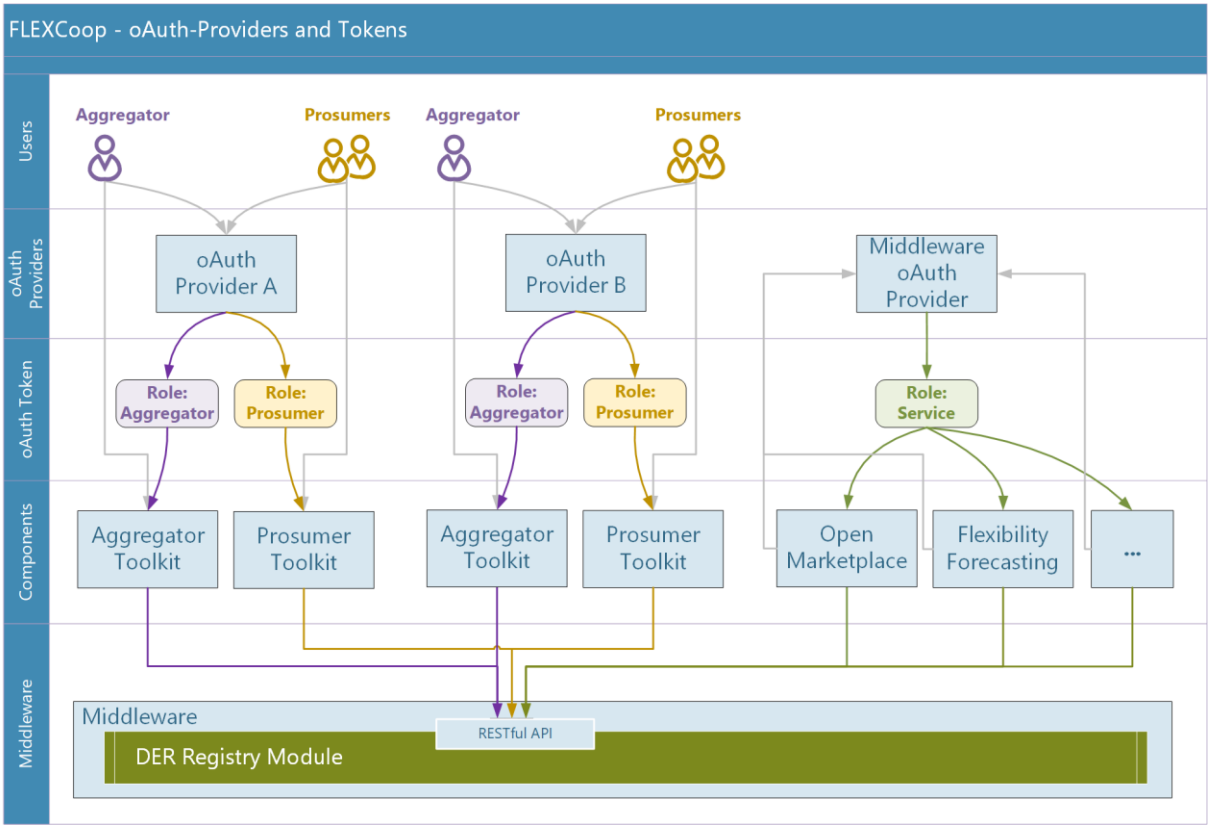

Figure 2: oAuth Providers and Tokens

The DER Registry module utilizes the EVE event hook feature to extend the access rights management to data privacy management, where each authenticated party can only access data it is allowed to.

On data access, the user role of the oAuth token is inspected

- oAuth Token with *"Aggregator"* role can only retrieve or modify DER devices that are associated with the requesting aggregator
- oAuth Token with *"Prosumers"* role can only retrieve or modify DER devices that belong to the requesting prosumer.
- oAuth Token with *"Service"* role authenticated as FLEXCoop components that can retrieve and modify all DER devices.

The event hook mechanism of the EVE framework is also used to act on newly added device database entries which were added by the OpenADR component.

8.2. Demonstrator Web Interface

This demonstrator component offers UI functionalities to access the DER Registry in different roles.

On the demonstrator portal a selection of the FLEXCoop oAuth providers allows to log into the system using either prosumers or aggregators credentials. After login in, the accounts role determines the content and layout presented in several UI panels.

The Devices panel lists devices associated with the user and allows filtering for specific criteria.

The demonstrator is implemented as Python flask webserver serving and updating the UI via REST endpoints. Authentication is applied via OAuth2/OpenID. The demonstrator backend component queries its data from the DER Registry and Open Marketplace module of the Middleware using the offered REST services and authorizes by supplying the authentication token of the user together with its requests.

9. INSTALLATION GUIDE

All components of this demonstrator can be installed in a Python 3.7 environment by cloning their GITs. This can either be done natively or inside of a Python virtual environment. The demonstrator can be deployed in container environments to support cloud integration and continuous integration developments (e.g. OpenShift Autobuild function).

The Python dependencies can be found in the *requirement.txt* files in the corresponding git repositories and in Appendix A of this document. E.g. the following command will install the requirements.

```
pip install -r requirements.txt
```

The DER Registry is implemented as sub-module of the middleware component, the required MongoDB and its environment variables are specified in the README.md of the GIT. At least one OAuth provider needs to be present and configured in the Middleware. The following command will start the DER registry:

```
python run.py
```

The Demonstrator UI configures its oAuth providers in a json config file, the middleware URI is specified using an environment variable. The demonstrator UI is started using:

```
python app.py
```

For more information regarding installation, build and configuration read the `README.md` in the components GIT.

10. INTERFACES WITH OTHER COMPONENTS AND THEIR INTEROPERABILITY

An overview of the interconnections of the DER Registry with other FLEXCoop components is shown in figure 1.

- The Flexibility Forecasting component described in D5.6 [6] queries the DERs and their states to compute predictions of the power consumption.
- The Aggregator Monitoring Platform described in D5.9 [7] allows Aggregators to filter for specific devices to combine them into Demand Request campaigns.
- The Prosumer Toolkit described in D6.7 [9] allows Prosumers to list all the devices and the devices states and to change their availability to the Open Marketplace.
- The Open Marketplace component described in D6.6 [8] validates the contracts proposed by the Aggregator regarding the availability of the Prosumers DERs to the marketplace and their availability at the timespan specified in the contract.

10.1. RESTful API

The RESTful API of the DER Registry is realized by the Middleware described in D4.8 [4]. The Python EVE framework used in the Middleware is powered by the Python Flask, Cerberus and MongoDB Frameworks.

10.1.1. Data Validation

Python EVE uses Cerberus to define API endpoints, the supported methods and the data structures consumed and produced by the resource endpoints. The data validation bases on the Cerberus schema definition ensures that only valid data structures can be created via the RESTful API endpoint.

The *devices* Cerberus schema definition show in Figure 3 was refined during the integration phase. Compared to the version described in the preliminary deliverable, the final version now includes detailed status information and a *device_name* field. Other fields not requested by any component were removed, for instance the *cluster_id* field.

The Common Information Model (CIM) of FLEXCoop described in D4.7 [3] was updated with these changes.

```
{
```

```
  "item_title": "devices",
  "id_field": "device_id",
  "item_lookup_field": "device_id",
  "item_url": "regex(\"[a-zA-Z0-9-]+\")",
  "cache_control": "max-age=10,must-revalidate",
  "cache_expires": 10,
  "resource_methods": [ "GET" ],
  "item_methods": [ "GET", "PATCH" ],
  "schema": {
    "device_id"      : { "type": "uuid", "minlength": 36, "maxlength": 36,
                         "required": true, "unique": true },
    "device_name"    : { "type":"string", "required":true, "default":"NO NAME" },
    "status"         : {
        "type": "dict",
        "schema": {
            "setPoint"        : { "type":"float" },
            "x-fanspeed"      : { "type":"float" },
            "mode"            : { "type":"string", "allowed":[ "1", "2", "3", "4", "5" ] },
            "operationState"  : { "type":"string", "allowed":[ "0", "1" ] }
        } },
    "account_id"     : { "type":"uuid", "minlength":36, "maxlength":36, "required":true },
    "ven_id"         : { "type":"uuid", "minlength":36, "maxlength":36, "required":true },
    "aggregator id"  : { "type":"string", "minlength":3, "maxlength":256, "required":true },
    "device_class"   : { "type": "string",
                         "allowed": ["hvacDevice", "lightingLoad", "dhwDevice", "otherLoad"],
                         "required": true },
    "availability"   : { "type":"string", "allowed":[ "yes", "no", "" ], "required":true },
    "marketplace_availability"    : { "type":"string", "allowed":[ "yes", "no" ],
                                      "required":false },
    "dr_availability": { "type":"string", "allowed":[ "yes", "no" ], "required":false },
    "location"       : { "type":"point", "minlength":1, "maxlength":36, "required":false },
    "device_type"    : { "type": "string", "allowed": [ "Demand", "Generation", "Storage" ],
                         "required": false },
    "max_capacity"   : { "type":"float", "required":false },
    "available_capacity"          : { "type":"float", "required":false }
  }
}
```

Figure 3: Cerberus JSON schema definition for DER endpoint

10.1.2. Data Queries

The EVE framework allows to retrieve multiple documents with support for complex query filters, sorting and projections.

These features can be used (for instance) to find all domestic hot water devices near a specific location that are available to the marketplace (query) sorted by their capacity (sort) and only return their *device_id* and *max_capacity* (projection) with a single RESTful API call.

10.1.3. Build-In Endpoint Documentation

The EVE framework used by the DER Registry provides a build-in documentation of its endpoints via SwaggerUI / OpenAPI.

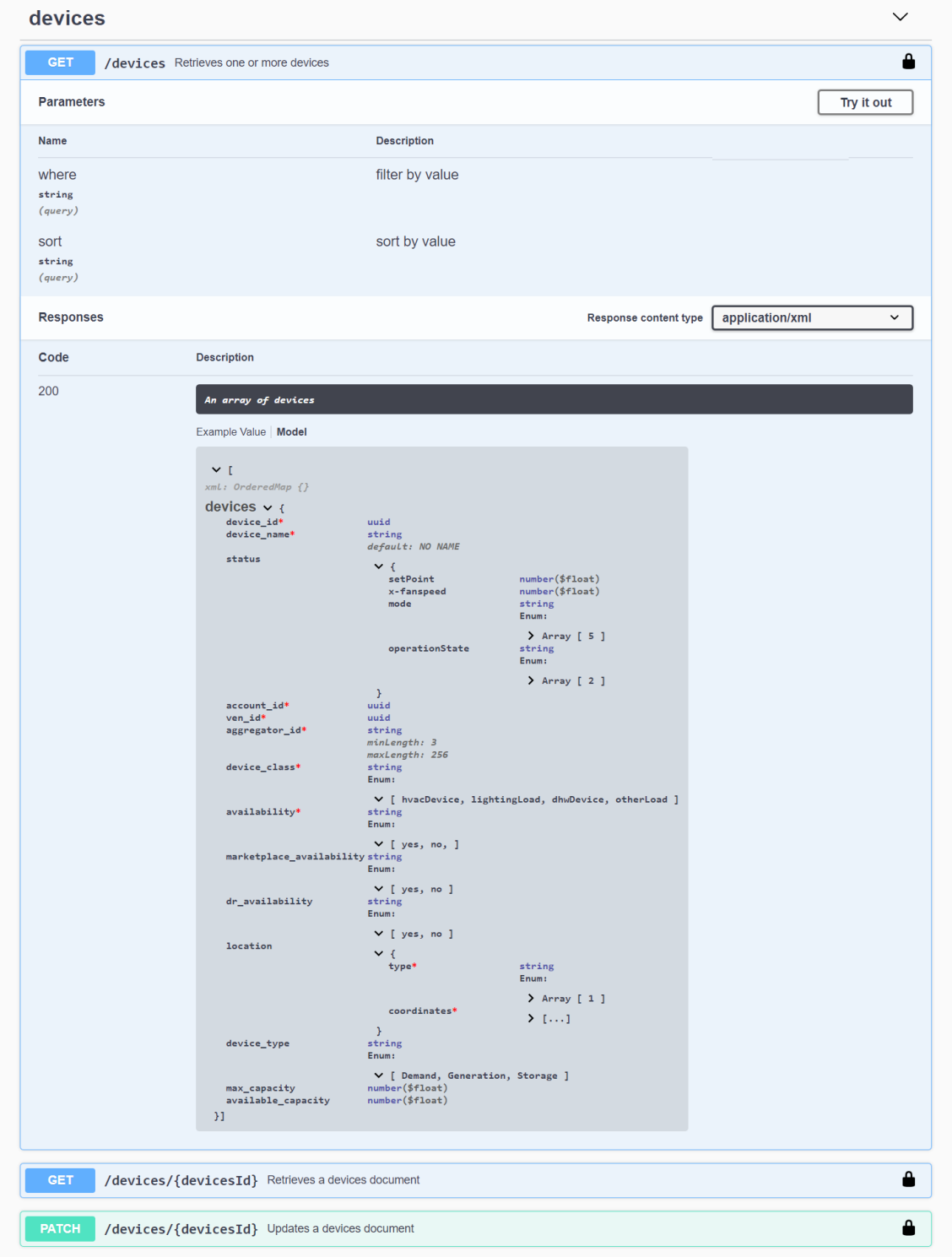

Figure 4: SwaggerUI DER Registry API documentation

10.2. Prosumer Toolkit

This subsection illustrates the DER Registry related views of the FLEXCoop Prosumer Toolkit which is described in depth in the D6.7 [9] deliverable. Figure 5 shows the device list of a Prosumer and the current status of each of the devices, Figure 6 shows graphs for device consumption and flexibilities.

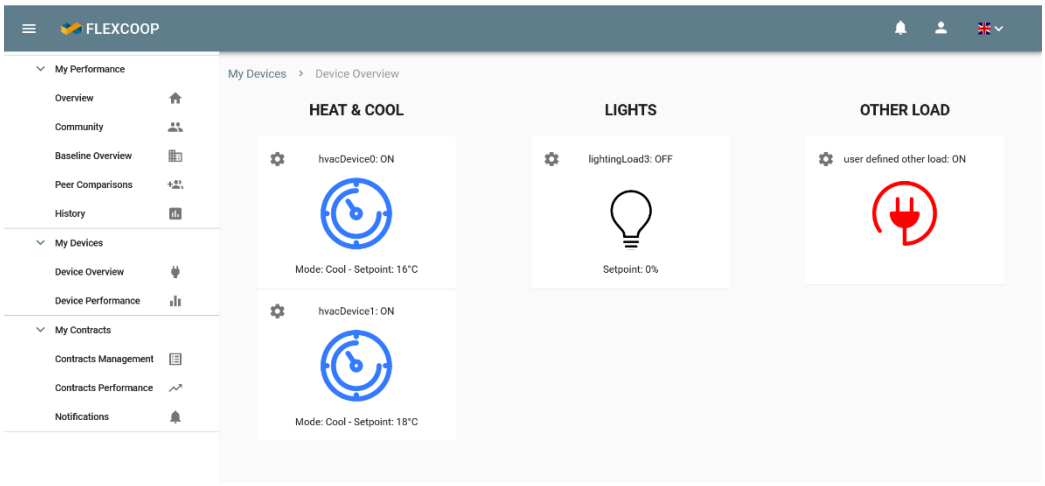

Figure 5: Prosumer Toolkit: Device List

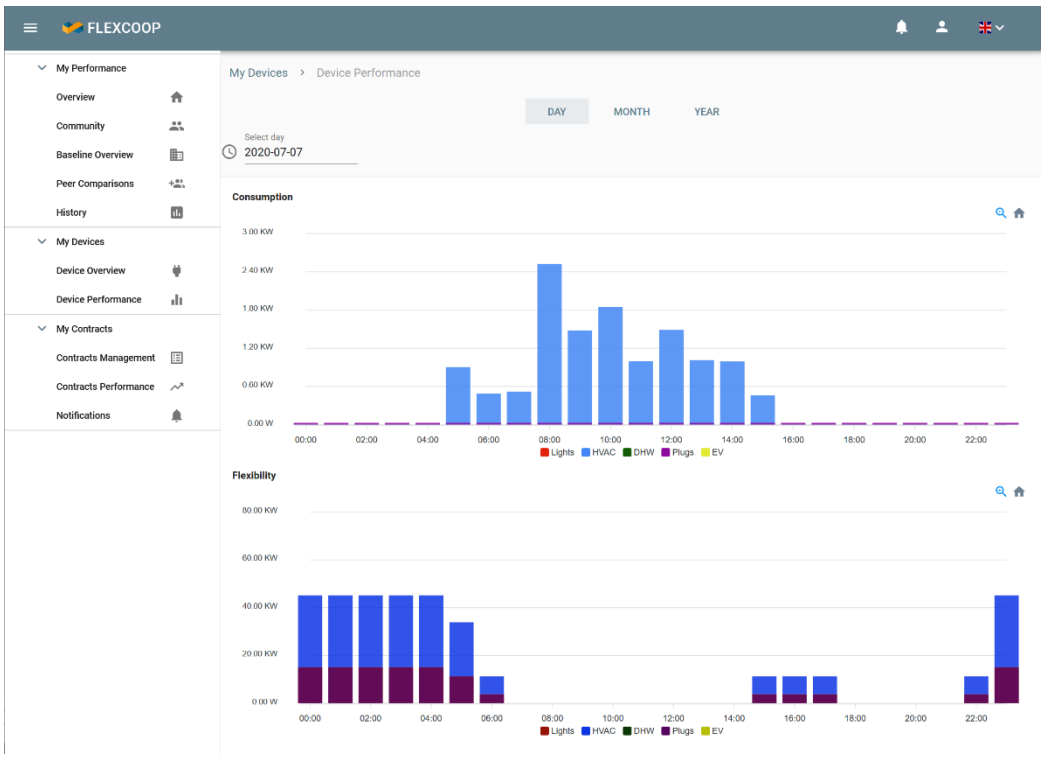

Figure 6: Prosumer Toolkit: Device Consumption and Flexibility

10.3. Aggregator Toolkit

The DER Registry view of the FLEXCoop Aggregator Visualisation Toolkit described in D5.9 [7] lists all devices available to the Marketplace. The view shown in Figure 7 allows to narrow down the list by applying several search criteria.

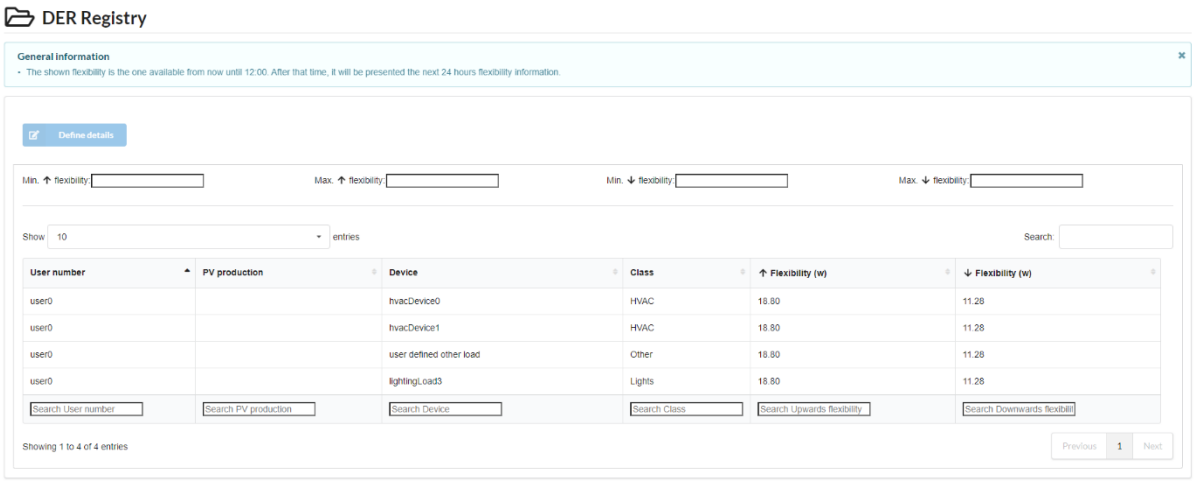

Figure 7: Aggregator Toolkit: Device List

11. REQUIREMENTS COVERAGE

The following table lists the different requirements for the deliverable and the related status of each of them.

OAuth2 / OpenID	Implemented within Task 4.4
CIM Model implementation	Implemented final CIM requirements.
DER Discovery	Implemented discovery and search feature
DER Capability Listing	Listing and search implemented
Semantically Enhance published DERs with data from the OSB	OSBs DERs are published using the OpenADR component

Table 1: Requirements Coverage

12. DEVELOPMENT AND INTEGRATION STATUS

The following table summarizes the development status of the FLEXCoop Open Registry and actions to be performed to complete it.

Current Status	Final demonstrator
Development status	Implemented
Pending development actions	None
Integration status	Integrated with the Middleware. Final CIM version implemented.
Pending integration actions	None

Table 2: Development and integration status

13. CONCLUSION

This demonstrator showcased an overview of the DER Registry of the FLEXCoop solution. All main objectives of the Registry are available and integrated in the FLEXCoop system. A minimal user interface for demonstration purposes has been implemented.

Besides the core functionality of the DER Registry, strong authentication and data privacy measures have been implemented within this demonstrator, in particular through the use of restrictive data access policy and application of the SEAC framework provided by T4.4 using the oAuth2 protocol.

APPENDIX A: LITERATURE

[1]. FLEXCoop D2.9: Framework Architecture including functional, technical and communication Specifications - Final Version
[2]. FLEXCoop D4.6: OSB Prototype - Final Version
[3]. FLEXCoop D4.7: Common Information Model - Final Version
[4]. FLEXCoop D4.8: Middleware and Big Data Management Platform – Final Version
[5]. FLEXCoop D4.9: Security Access Control (SEAC) Framework
[6]. FLEXCoop D5.6: Flexibility Forecasting, Segmentation and Aggregation Module – Final Version
[7]. FLEXCoop D5.9: Aggregator real-time Monitoring and Control Platform - Final Version
[8]. FLEXCoop D6.6: Open Marketplace for flexibility Pooling and Sharing – Final Version
[9]. FLEXCoop D6.7: Prosumer Application – Final Version

APPENDIX B: REQUIREMENTS

DER Registry Middleware Module

```
asn1crypto==0.24.0
Cerberus==1.2
certifi==2019.3.9
cffi==1.12.3
chardet==3.0.4
Click==7.0
cryptography==2.6.1
Eve==0.9.2
Eve-Swagger==0.0.11
Events==0.3
Flask==1.0.2
flask-swagger-ui==3.20.9
idna==2.8
itsdangerous==1.1.0
Jinja2==2.10.1
jwt==0.6.1
MarkupSafe==1.1.1
pandas==0.24.2
pycparser==2.19
pymongo==3.8.0
requests==2.21.0
simplejson==3.16.0
six==1.12.0
```

Demonstrator Admin View

```
flask-restplus==0.13.0
oauthlib==3.1.0
requests-oauthlib==1.1.0
requests==2.23.0
Flask-Dance==3.0.0
```