



D5.1 - 1st Release of DIGITISE DT Solutions and Tools for Consumers Engagement in Marketplace Environments

Project number	101160671
Project acronym:	DIGITISE
Project title:	Digital Innovative cross-sector services for Greater citizen Integration in a just energy Transition, and Societal Empowerment

Work Package	WP5: DIGITISE DT Solutions and Tools for Consumers Engagement in Marketplace Environments
Responsible Partner	UBITECH
Official Submission Date	30/11/2025
Actual Submission Date	30/11/2025
Type	Other
Dissemination Level	Public
Reviewers	MIW (VOLTIVA), CIRCE
Version	1.3



Funded by
the European Union

Versioning and contribution history

Version	Date	Author(s)	Notes
0.1	31/7/2025	Eleftheria Petrianou (UBITECH)	Accepted Table of Contents
0.2	31/10/2025	Kostas Tsatsakis (SUITE5)	Initial input of Section 2.1
0.3	7/11/2025	Vlasios Koutsos (UCD), Eleftheria Petrianou (UBI), Titos Georgoulakis (UBI)	Initial input of Sections 2.2 and 2.3
0.5	14/11/2025	Eleftheria Petrianou (UBI), Titos Georgoulakis (UBI), Kostas Tsatsakis (SUITE5), Vlasios Koutsos (UCD)	Updated contributions
1.0	20/11/2025	Eleftheria Petrianou (UBI), Titos Georgoulakis (UBI), Kostas Tsatsakis (SUITE5), Vlasios Koutsos (UCD)	Document version ready for review – Executive Summary, summary tables, Introduction, Conclusion, References added
1.1	25/11/2025	Elena Ruiz (VOLTIVA), Boniface Mselle (CIRCE)	Reviewed Document – Addressed feedback
1.2	28/11/2025	Eleftheria Petrianou (UBI), Titos Georgoulakis (UBI), Kostas Tsatsakis (SUITE5), Vlasios Koutsos (UCD)	Version ready to be submitted
1.3	30/11/2025	Eleni Tsironi (UBI)	Submitted version

Copyright notice

© Copyright 2024–2027 by the DIGITISE Consortium

This document contains information that is protected by copyright. All Rights Reserved. No part of this work covered by copyright hereon may be reproduced or used in any form or by any means without the permission of the copyright holders.

Disclaimer

The information and views set out in this report are those of the authors and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

Preface

Funded by the European Commission under Grant Agreement number 101160671, DIGITISE is a project focused on enhancing the digital literacy and empowerment of consumers and prosumers in the energy sector. By integrating advanced technologies and fostering active engagement in digital energy activities and markets, DIGITISE aims to play a crucial role in the global energy transition.

Executive Summary

The current document (D5.1 – “1st Release of DIGITISE DT Solutions and Tools for Consumers Engagement in Marketplace Environments”) presents the status of the development actions taken in the scope of WP5 tasks and activities towards the 1st version of the energy and non-energy applications and tools for consumer engagement in marketplace environments. Based on the extracted functional requirements of D2.1 and the architectural design of D2.2, the deliverable provides comprehensive documentation of the four applications and tools of WP5, including the progress made for the 1st version and a development plan until the 2nd and final version.

Starting from Task 5.1 – Household Digital Twin, which aims to facilitate the optimization of the DIGITISE energy services, the realization of the different components is described, presenting the adoption of the open-source digital twinning solution of Arras Energy – GridLAB-D. Exploiting the capabilities of GridLAB-D and further enhancing them with novel parametrization and device synthetic consumption pattern generation methods, has led to the delivery of an initial version of the DIGITISE Digital Twin application which poses solid foundation for the next version and for supporting the DIGITISE applications’ implementation and evaluation.

The Task 5.2–Aggregator Portfolio Management and Virtual Power Plant Configuration Application provides insights into the implementation of its 1st release, focusing on the development of a powerful VPP and portfolio management tool offered to aggregators. Its core functionalities, like management, clustering and grouping of distributed energy resources (DERs) as well as reporting mechanisms and an intuitive dashboard have already set a strong basis for the next release and further market integration.

As for Task 5.3 – Flexibility Marketplace Environment, whose main objective is to actively engage prosumers and aggregators in the flexibility market transactions, the development plan showcases a clear implementation path, where progress is reported for the 1st version regarding the configuration of flexibility assets, the flexibility contract introduction and negotiation and the basic smart contracting mechanisms, setting the ground for the flexibility monetization functionalities to be added in the final release.

And finally, regarding Task 5.4–Behavioural Profiling and Insights Application, which aims to extract valuable insights into consumers’ behaviour and habits, incorporating various data, from real data and forecasts to digital twin simulation outcomes, significant advancements have been documented. Trend analysis, lifestyle clustering of customers and API-based data sharing are among those, with a concrete plan of enhancements and updates for the 2nd release described in the current document.

Table of Contents

Executive Summary	4
Introduction.....	9
Scope of the document.....	9
Relation with other deliverables.....	10
Structure of the document.....	10
1 DIGITISE Digital Twins	12
1.1 DIGITISE Household Level Digital Twin.....	12
1.1.1 Overview.....	12
1.1.2 Delivered Functionality.....	12
1.1.3 Assumptions and Constraints	18
1.1.4 Development plan for next version.....	19
2 DIGITISE End-User Applications.....	21
2.1 Aggregator Portfolio Management and Virtual Power Plant (VPP) Configuration Application.....	21
2.1.1 Overview.....	21
2.1.2 Delivered Functionality.....	22
2.1.3 Assumptions and Constraints	29
2.1.4 Development plan for next version.....	30
2.2 Flexibility Marketplace Environment.....	31
2.2.1 Overview.....	31
2.2.2 Delivered Functionality.....	32
2.2.3 Assumptions and Constraints	40
2.2.4 Development plan for next version.....	41
2.3 Behavioural Profiling and Insights Application	42
2.3.1 Overview.....	42
2.3.2 Delivered Functionality.....	43
2.3.3 Assumptions and Constraints	48
2.3.4 Development plan for next version.....	48
3 Conclusions.....	50
References.....	51

List of Figures

Figure 1 – Extracted consumption cycle of a washing machine from the refit dataset.....	17
Figure 2 – Synthetically generated consumption pattern of a washing machine that is utilized in the digital twin, scaled and time stretched to match target appliance characteristics.....	17
Figure 3 – Example simulation output of a household in Athens, with dummy data Configurations, showcasing appliance consumption in kwh.....	18
Figure 4 – Example metadata json file of the scenario simulation request	18
Figure 5 DIGITISE VPP Login Page.....	25
Figure 6 DIGITISE VPP Portfolio Overview.....	25
Figure 7 DIGITISE VPP Portfolio Overview – Assets Selection.....	26
Figure 8 DIGITISE VPP Clusters Overview	27
Figure 9 DIGITISE VPP Clusters Overview – filtering.....	27
Figure 10 DIGITISE VPP Clusters Overview – Insights.....	28
Figure 11 DIGITISE VPP Clusters Overview – KPIs	28
Figure 12 DIGITISE VPP Strategies Overview	29
Figure 13 – User Login page of the Flexibility Marketplace.....	35
Figure 14 – Assets listing from prosumer view.....	35
Figure 15 – assets filtering from prosumer view (same for any user)	36
Figure 16 – assets searching from prosumer view (same for any user).....	36
Figure 17 – Asset Edit capability for prosumer	37
Figure 18 – Asset preview capability for aggregator.....	37
Figure 19 – Contracts listing from Aggregator view (searching and filtering possible for all users).....	38
Figure 20 – New contract creation from aggregator view	38
Figure 21 – Contract publication and signing feature for aggregator.....	39
Figure 22 – Contract preview and signing prompt for prosumer	40
Figure 23 – Connection to Rabby wallet pop-up	40
Figure 24 – User 36 activity over a year per hour.....	46
Figure 25 – Least and most active users respectively between 13:00 and 18:00, on a daily basis	46
Figure 26 – Top-10 users according to lifestyle {early birds, night owls}, with instance counts across a year-long duration	47
Figure 27 – Energy-conscious instances per user and High-consumption users over a year	47
Figure 28 – Average temperature per month of user 36 household, over a year	48

List of Tables

Table 1 – Features of Household Digital Twin for 1st version.....	13
Table 2 – Requirements mapping with features of the Household Digital Twin.....	13
Table 3 – Features of Household Digital Twin for the 2nd version	20
Table 4 – Features of VPP Configuration app for 1 st version	22
Table 5 – Requirements mapping with features of VPP Configuration app.....	22
Table 6 – Features of Virtual Power Plant (VPP) Configuration Application for 2nd version	30
Table 7 – Features of Flexibility Marketplace for 1st version	32
Table 8 – Requirements mapping with features of Flexibility Marketplace app.....	32
Table 9 – Features of Flexibility Marketplace application for the 2nd version.....	42
Table 10 – Features of Behavioural Profiling app for 1st version.....	43
Table 11 – Requirements mapping with features of Behavioural Profiling app	44
Table 12 – Features of Behavioural Profiling application for the 2nd version.....	49

Abbreviations

Abbreviation	Full Name
DoA	Description of Action
DT	Digital Twin
WP	Work Package
API	Application Programming Interface
IoT	Internet of Things
AI	Artificial Intelligence
HVAC	Heating, Ventilation, and Air Conditioning
TMY	Typical Meteorological Year
GLM	GridLAB-D Model file
SQLite3	Structured Query Language Lite (database)
DLT	Distributed Ledger Technology
DR	Demand Response
VPP	Virtual Power Plant
DER	Distributed Energy Resources
TSO	Transmission System Operator
DSO	Distribution System Operator
PoA	Proof of Authority (blockchain consensus mechanism)
ETH	Ethereum (cryptocurrency)
SSO	Single Sign-On

KPIs	Key Performance Indicators
BESS	Battery Energy Storage System
PV	Photovoltaic
EV	Electric Vehicle
LEC	Local Energy Community
CSV	Comma-Separated Values (data format)
JSON	JavaScript Object Notation (data format)

Introduction

Scope of the document

As, based on the DoA (Description of Action) one of the target goals of DIGITISE is to further empower aggregators and mainly their customers, who act as consumers and prosumers, and support their engagement into the world of advanced energy market designs and data- and intelligence-driven technologies that have emerged into the energy sector, Work Package 5 comes to address this by delivering a set of tools and applications dedicated to the below solutions:

1. The DIGITISE Household Digital Twin (Task 5.1), that exploits the power of the various data available along with the AI analytics offered in the DIGITISE ecosystem and combines them with the knowledge of physical aspects of households, in order to further optimize the DIGITISE services' operations with added intelligence-driven insights.
2. The DIGITISE Aggregator Portfolio Management and Virtual Power Plant (VPP) Configuration application (Task 5.2), that promotes flexible assets into novel flexibility and demand-response (DR) products, as a consequence of the prosumers' participation in flexibility transactions through the Flexibility Marketplace application. The VPP configuration as well as the strategy design capabilities of the application are based on advanced AI flexibility analytics of DIGITISE, fostering the integration of recent technological trends in the energy and flexibility markets.
3. The DIGITISE Flexibility Marketplace (Task 5.3), which serves as the means to facilitate the establishment of bilateral agreements between prosumers and aggregators / Local Energy Communities (LEC) managers in the scope of flexibility contracts' set-up, activation, settlement and remuneration, leveraging the most recent Distributed Ledger Technologies (DLT) and allowing the monetization of the flexibility that is offered through the configuration of the respective prosumer assets.
4. The DIGITISE Behavioural Profiling and Insights application (Task 5.4), that analyzes and examines the customers' consumption profiles and usage trends for extracting valuable insights based on consumption and sensor data as well as on AI behavioural profiling analytics results made available at the DIGITISE Data Space. This enriched information on consumers' behavioural profiles can be further exploited by service providers to provide additional market products and enroll consumers in new data economy schemas.

The aforementioned applications and tools are further described in the current deliverable, focusing on the technological aspects of their implementation, aiming to present in a clear and comprehensive way the development roadmap followed and the

latest status of the applications' advancements towards the project's first milestone. The objective of the deliverable is to showcase that by reaching the first milestone, the tools and services of WP5 are in a mature state in terms of development and have set a clear path towards their final version. For that reason, apart from each tool's components and functionalities descriptions, in order to offer an overall picture of the WP5 activities' status, a verification of requirements coverage completes the descriptions, but most importantly details on the technologies utilized, along with implementation steps and assumptions made for the 1st release, are provided.

Relation with other deliverables

Since D2.1 – “DIGITISE End-User Requirements and Socio-Economic Analysis” [1] and D2.2 – “DIGITISE Evaluation Protocol, Novel Market Models and Architecture Design” [2] have preceded the current deliverable D5.1, they have set the basis for the WP5 tools' design. Conceptualization of tasks has been expressed through the definition of the relevant use cases and the extraction of user and technical requirements related to each application and are presented in detail in D2.1. Following the work of D2.1 and stepping upon the use cases and requirements related to each application, more in-depth descriptions of the applications and their designed architecture (components and features) are provided in D2.2. This explains the constant reference to these two deliverables throughout the current document, in an effort to logically connect the designing steps that led to the 1st version of the WP5 energy and non-energy services and to showcase that these were taken into account during the implementation process.

Structure of the document

The current document is structured as follows:

- The contents span across two major chapters:
 - Chapter 1 – DIGITISE Digital Twins: It refers to the DIGITISE Digital Twinning service that is part of the “DIGITISE Baseline AI Analytics and Digital Twins Solutions” tier of the overall DIGITISE architecture.
 - Chapter 2 – DIGITISE End-User Applications: It covers the three applications that belong to the “DIGITISE end-user digital solutions, applications and products” tier according to the DoA.
- Under each application chapter, the following sections can be found:
 - Overview: It includes the description of the functionalities that are planned to be covered by the different components and features of each tool, based on the architectural details described in D2.2.
 - Delivered Functionality: At this section, more details are provided for depicting the status of the development activities until the 1st version of the applications. The features that are implemented or started being developed in the 1st release period are marked accordingly as “Completed” or “Partially

Completed". Additionally, all features of each application are mapped with the corresponding requirements that derive from D2.1, so as to ensure that all of them will be fulfilled with the delivery of the full set of features by the end of the 2nd (and final) release of the applications. More details on the different technologies utilized as well as on the most recent results of the implemented functionalities can be found in the respective subsections named "Technology Stack" and "Implementation Details".

- Assumptions and Constraints: Since at this phase of the project there are several technical points that remain unaddressed among the technical but also the demo sites' partners, a list of assumptions that needed to be made or constraints that needed to be considered are documented as well in this paragraph for each tool.
- Development Plan for next version: At the last section of each application chapter, a development roadmap is presented that includes all the actions that are expected to take place so as to achieve the delivery of the 2nd version of the WP5 services.
- Conclusions: This chapter summarizes the key findings and pinpoints the main steps to be followed for the realization of the 2nd services release.

1 DIGITISE Digital Twins

1.1 DIGITISE Household Level Digital Twin

1.1.1 Overview

The Household Digital Twin is a pivotal tool supporting the operations of DIGITISE applications, offering users the ability to effectively experiment with diverse simulation scenarios based on digital twin models of the project's pilot households. The implementation provides three core components that work together to enable comprehensive household energy simulation and management.

Household Simulation Engine: The Simulation Engine enables the configuration and execution of household simulation models in a dedicated GridLAB-D environment. GridLAB-D is an open-source power system simulation tool developed by the U.S. Department of Energy to model complex distribution networks with high fidelity. It integrates electric power flow, building energy use, distributed energy resources (DERs), and control systems into a single simulation environment. Unlike traditional grid simulators, GridLAB-D uses an agent-based modeling approach, allowing each component—such as loads, appliances, inverters, and market controllers—to behave autonomously and interact dynamically [3]. Each household has its own customized model incorporating physical characteristics (e.g. household static data), and scenario-specific data. The engine processes dynamic appliance consumption patterns, HVAC control strategies, solar generation, and weather conditions to produce detailed energy simulations.

Configuration Engine: The Configuration Engine orchestrates all activities before and after simulation execution. When a simulation request is received, it gathers and formulates relevant data and parameters, including unit conversions from SI/metric to GridLAB-D units, generation of dynamic appliance consumption patterns from historical templates, creation of HVAC schedules, and location-specific TMY (Typical Meteorological Year) file management. The engine triggers scenario execution and collects, transforms, and makes simulation outputs available for retrieval.

API Connector: The API Connector provides RESTful interfaces for external users and applications to interact with the Digital Twin. Users can request simulation executions, retrieve results, manage household configurations, and access HVAC scheduling capabilities. The connector is designed for integration with the DIGITISE Data Space and other ecosystem components.

1.1.2 Delivered Functionality

The current status of the development for the 1st version of the Household Digital Twin, is presented per feature at the table below.

TABLE 1 – FEATURES OF HOUSEHOLD DIGITAL TWIN FOR 1ST VERSION

Feature ID	Feature Name	Development status for v1
SE-01	Configuration of household models	Partially Complete. Mock household models with basic parametrization.
SE-02	Execution of simulation scenarios	Complete.
CE-01	Formulation of the simulation’s input and output	Partially Complete. Initial input formulation complete.
CE-02	Setup of the simulation parameters	Partially Complete. Basic implementation.
CE-03	Triggering of the execution of a simulation	Complete.
CE-04	Storing of the simulation results	Partially Complete. Preliminary results storing process.
AC-01	Simulation execution and results request interface	Partially Complete. Currently supporting simulation execution.
AC-02	Data Space Communication Interface	To be developed.

The requirements that are described in D2.1 and need to be covered during the development of the Household Digital Twin are listed below and are mapped with the corresponding tool features that fulfill them.

TABLE 2 – REQUIREMENTS MAPPING WITH FEATURES OF THE HOUSEHOLD DIGITAL TWIN

Requirement	Description	Mapped Feature(s)
PRO-12	Ensure integration of real-time and historic measurements from various household data sources (i.e. appliance consumption, DERs, IoT sensors, etc)	AC-02, CE-01, CE-02
PRO-13	Support the integration of AI baseline analytics for incorporating any user- and energy-related aspects in the household digital twins, i.e. DER generation forecasts or occupancy profiling analytics	AC-02, CE-01, CE-02
PRO-14	Consider physical aspects of household assets in energy demand(/household) simulations	SE-01, CE-02

<i>PRO-15</i>	Allow the configuration of the digital twins with several parameters, i.e. social, energy or economics	SE-01, CE-02, AC-02
<i>PRO-16</i>	Provide an easily accessible interface for interaction with the Digital Twin (request and retrieval of digital twin simulation results) for updating or reviewing the simulated scenarios at any time	AC-02, AC-01
<i>PRO-17</i>	Exploit and extend open-source solutions for the implementation of the Digital Twins	CE-01, SE-01
<i>PRO-18</i>	Facilitate the implementation and evaluation of the DIGITISE services and tools	AC-01, CE-02, SE-01

At this phase of the implementation, the functionalities cover the simulation execution process, the core household model configuration and parametrization aspects, along with main capabilities for interaction with the Digital Twin – basic API-based simulation configuration and execution requests. Although there are features to be finalized, while others are planned to be delivered at the 2nd version, i.e. results’ database storing and API-based query and retrieval, all requirements of D2.1 are properly mapped with the full set of features with the intention to ensure that the development plan is in accordance with the main objectives of the project.

1.1.2.1 Technology Stack

The Household Digital Twin is implemented as a containerized microservices architecture using the following technologies:

Backend Framework: Python 3.11 using FastAPI for the API Connector and Configuration Engine, providing asynchronous request handling, automatic OpenAPI documentation, and Pydantic-based data validation. Dedicated Python scripts for unit conversion, JinJA2 library for GLM (Gridlab-D Models) dynamic file creation and synthetic data generation. Organized file system structure with scenario-specific directories containing GLM files, generated CSV inputs (appliance pattern consumption, HVAC schedules), TMY(Typical Meteorological Year) files, simulation results, and metadata JSON files [4].

Simulation Engine: GridLAB-D 4.2.0 running in a dedicated Ubuntu-based Docker container, providing industry-standard power distribution simulation capabilities with support for residential loads, solar generation, and HVAC systems.

Databases: PostgreSQL 13 for configuration and results metadata storage, utilizing JSONB columns for flexible configuration storage and enabling efficient querying and versioning.

Containerization: Docker Compose orchestrating multiple services (API, GridLAB-D simulator, PostgreSQL databases) with shared volumes for data exchange. The application can be deployed on any Docker-capable infrastructure, with all services configured via environment variables. The API serves requests on configurable IPv4/IPv6 addresses and ports, with comprehensive logging for debugging and monitoring.

1.1.2.2 Implementation Details

The implementation is divided into three cooperating services: the Configuration Engine, the Household Simulation Engine and the API Connector.

The Configuration Engine is backend oriented and built using FastAPI through Python. In order to build the household models, several household configurations are required that are input parameters to the GLM (GridlabD-Model) files which is the file type used by Gridlab-D to perform simulations. These parameters vary from static household data information (floor_area, ceiling height, building orientation, window wall area etc.) to more specific information related to the HVAC systems (such as heating_system_type, cooling_system_type, design_heating_capacity etc.), the domestic water heaters (tank_volume, tank_diameter, heating_element_capacity etc.), specific household appliance information (washing machines, dishwashers, ranges, lighting etc.) and where available PV Panels, EV-Chargers, and BESS (Batteries). This information is expected to be provided by the different pilots, and the configurations are created through the **POST /configs** endpoint and saved in a PostgreSQL configs database (JSONB type column) which enables flexible schemas and versioning. Preliminary work has been conducted to gather the aforementioned demo sites configurations (static household information, appliance specific data, etc.). A similar process has been conducted by WP6 to gather specific demo sites appliance information that can be utilized by the DIGITISE tools.

The rationale behind this architecture was the limitation of Gridlab-D to configure scenarios that are dynamic and scalable. GLM files are plain text files structured in a specific way to be recognized by Gridlab-D. To enable the dynamic configuration of the different households and the configuration of the different scenarios, we utilized the JinJA2 library in Python that allows dynamic template-based file generation, supporting conditional logic for different household configurations, appliance types and scenarios.

To perform a simulation, the API connector is utilized. First the user specifies the household_ID, pilot_name, simulation start_time and stop_time, interval length (in seconds) and the output properties they wish to receive (e.g. HVAC total load, household air temperature, any appliance's load, and/or any other parameter Gridlab-D supports as output), assuming that the household configurations have already been created.

This information is parsed to the **POST /simulations** endpoint, which is part of the Configuration Engine, in order for the simulation scenario to be created. Using this endpoint, a dedicated folder with the {scenario_id} name is created, along with the GLM

file with the appropriate household configurations obtained from the configs database and replaced on the file.

If specific appliance data are available for the selected simulation timespan (through dedicated household meters), they are obtained and converted to CSV's so that GLM files utilize them. In case any forecasted data are produced through the DIGITISE AI Analytics are available, they can also be used for simulations regarding future dates. In cases of no device data availability, we synthetically generate the consumption patterns of the target appliances that we have no available data.

In cases where data are synthetically generated, we utilize a scaling algorithm along with the REFIT NILM (Non-Intrusive Load Monitoring) dataset [4] appliance measurements as well as the desired target device nominal power, average duration of usage and target device activations per day. Moreover, we capture the appliance activations (e.g. washing machines) from the REFIT DATASET, and we time stretch these to match the desired device's target average duration and scale its nominal power to match the desired ones. We evaluated four different synthesis methods such as template scaling (Time + Amplitude Rescaling) [5], time-warping, statistical weighted averaging, and cycle mixing, using the metric Pearson correlation coefficient which measures the linear similarity between two time-series (shape similarity) [6]. Among these, template scaling was selected because it achieved the highest shape fidelity (Pearson-R) while the results remained robust on the different appliances with our small template set.

An example of this process is presented in the Figures below. Figure 1 showcases the extracted consumption of a washing machine from a household of the REFIT dataset, where the appliance was active for 70 minutes, with a sample rate of 7 seconds and a Nominal Power of 2.3kW. Figure 2 shows the synthetically generated washing machine pattern with different specifications, namely target appliance duration 90 minutes, and nominal power of 2kW with a requested simulation timestep of 15 minutes, which was generated using the aforementioned method. Similar processes take place for other household appliances where needed. The output of this process is a CSV file of the target appliance, which contains x number of random appliance activations per day, where x is specified during the scenario creation process of the Configuration Engine.

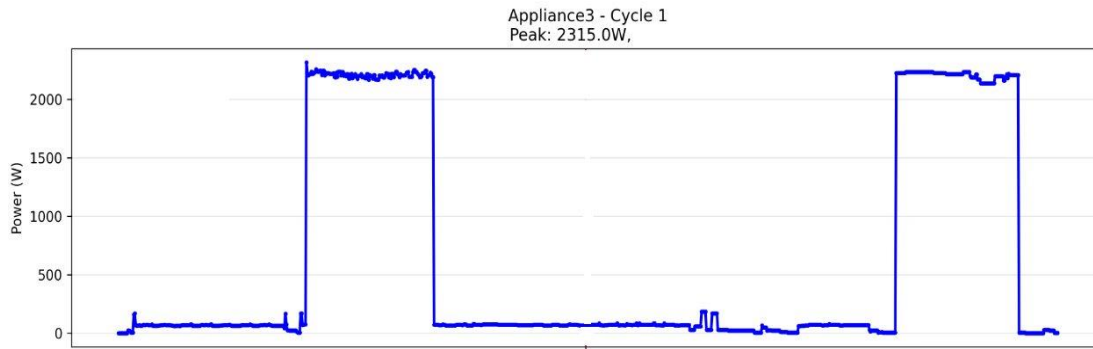


FIGURE 1 – EXTRACTED CONSUMPTION CYCLE OF A WASHING MACHINE FROM THE REFIT DATASET

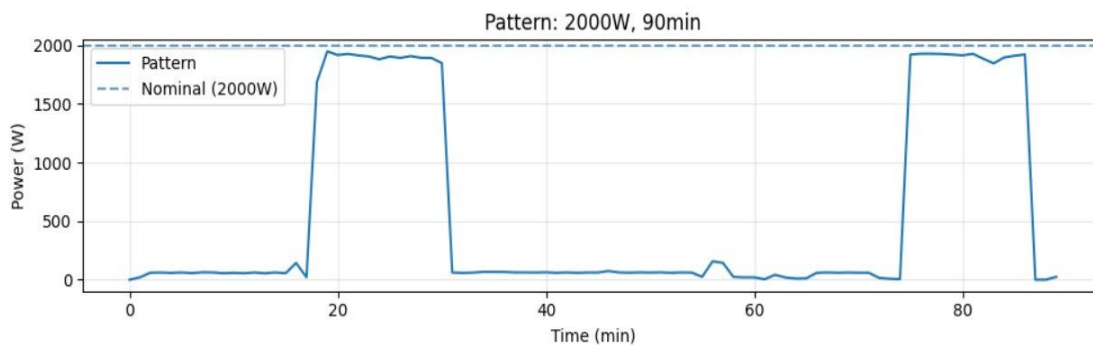


FIGURE 2 – SYNTHETICALLY GENERATED CONSUMPTION PATTERN OF A WASHING MACHINE THAT IS UTILIZED IN THE DIGITAL TWIN, SCALED AND TIME STRETCHED TO MATCH TARGET APPLIANCE CHARACTERISTICS

The retrieved CSV's are added to the specified scenario folder, along with a tmy2 file which contains the typical meteorological year data of the selected pilot's location and are also included in the GLM file. Finally, a JSON metadata file is created and stored in the same folder which contains the relevant simulation scenario metadata.

Following the scenario creation, the **POST /simulations/{scenario_id}/execute** endpoint is utilized to perform the requested simulation, which is part of the Simulation Engine. This endpoint utilizes the command 'gridlabd <file_name.glm>' which triggers the simulation inside a docker container where Gridlab-D is installed. When the simulation is finished, the results are exported to a {scenario_id}.csv in the same scenario folder, and all the information requested from the output properties are recorded with the specified recording interval rate. The results are then returned to the user as CSV's along with the JSON scenario metadata file through the **GET /download/results/{result_id}** endpoint of the Configuration Engine. All the API endpoints mentioned in this implementation are used internally by the configuration and simulation engine. The API Connector handles the requests and formulates appropriate inputs to these engines (through these endpoints).

Figure 3 showcases the requested outputs of a simulation scenario, namely the consumption (in kW) of an example household in Athens with dummy configurations, including major load appliances such as the HVAC system, washing machine, water heater (main_wh), dishwasher and indoor and exterior lighting. The washing machine and dishwasher consumption is synthetically generated and preset to 1 random activation per day, the lighting and the water heaters follow predefined schedules and the HVAC load behaves based on preset heating and cooling setpoints along with Athens TMY data. Figure 4 represents the scenario metadata JSON file of the requested scenario simulation, where the simulation was set to run for an entire week with an interval rate of 60 seconds.

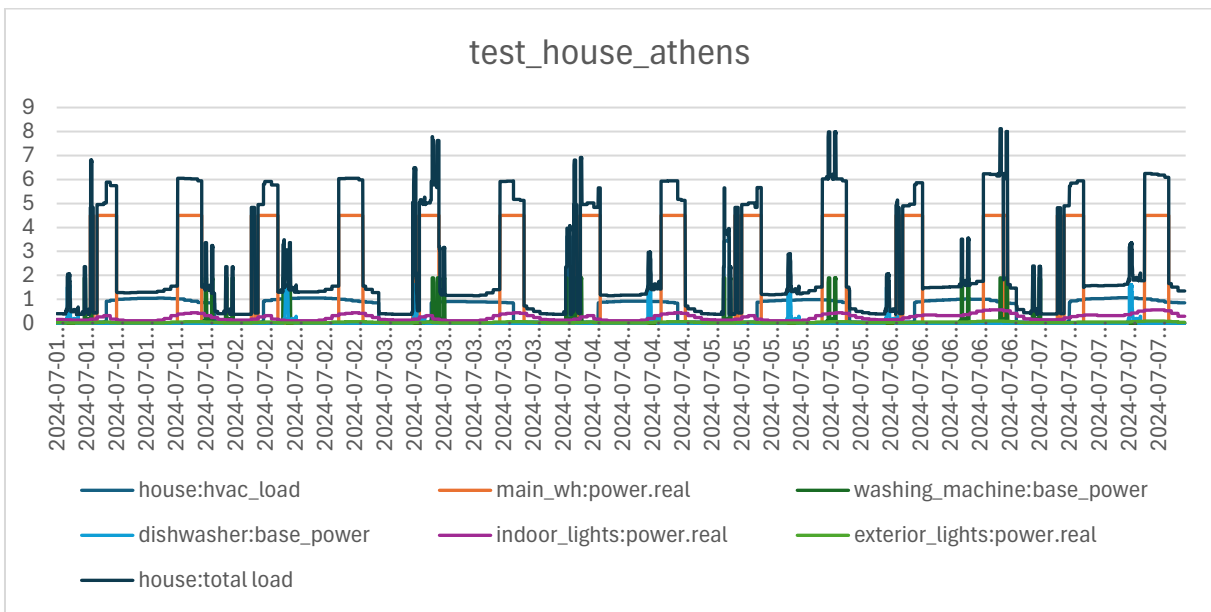


FIGURE 3 – EXAMPLE SIMULATION OUTPUT OF A HOUSEHOLD IN ATHENS, WITH DUMMY DATA CONFIGURATIONS, SHOWCASING APPLIANCE CONSUMPTION IN KWH

```

{
  "scenario_id": "d2a3fd4f-10a5-4a5f-8897-18e0ab1b9d48",
  "config_id": "7fd9a50f-0d25-4495-b91b-67ee30f4e7dc",
  "config_name": "house_athens_complete_example",
  "start_time": "2024-07-01 00:00:00",
  "stop_time": "2024-07-07 23:59:00",
  "created_at": "2025-11-06T12:55:54.536565",
  "glm_file": "scenario_d2a3fd4f.glm",
  "output_file": "results_d2a3fd4f.csv",
  "generated_csvs": [
    "dishwasher_consumption.csv",
    "washing_machine_consumption.csv"
  ],
  "generation_errors": {},
  "scenario_dir": "./data/scenarios/house_athens_complete_example/d2a3fd4f-10a5-4a5f-8897-18e0ab1b9d48"
}

```

FIGURE 4 – EXAMPLE METADATA JSON FILE OF THE SCENARIO SIMULATION REQUEST

1.1.3 Assumptions and Constraints

The first version (v1) of the application has been developed with the following assumptions and constraints:

- **Unit System:** We assume that partners provide their configuration parameters in SI Units. Gridlab-D accepts most parameters in the hybrid metric system (American metric system), so unit converter scripts were applied on the configurations before the GLM files creation, as well as right after the results retrieval.
- **Weather Data:** The weather data used by the GLM files and consequently at the simulations, are tmy2 files data, which utilize temperature and relevant humidity data from 2012–2024 where each month from every year in the files is randomly used to formulate one typical year worth of data. (e.g. January 2020, February 2013, March 2017, etc..). Each pilot site is represented by a different tmy2 file [7].
- **HVAC Load Calculation:** GridLAB-D's *house* model is based on an Enhanced Equivalent Thermal Parameter (ETP) representation of the building's thermal dynamics [8]. It computes HVAC load by dynamically solving the building's thermal balance using its built-in *house* and *hvac* modules from the static household configurations provided. The model evaluates heat gains and losses through the building envelope, internal gains, infiltration, and weather conditions at each simulation timestep. Based on the thermostat setpoints, insulation characteristics, and equipment efficiency, GridLAB-D determines the required heating or cooling power needed to maintain indoor temperature. The resulting HVAC electricity consumption is therefore an outcome of the building's thermal response, the HVAC system performance curves, and the external weather driver (through tmy files), all of which are updated continuously throughout the simulation. The Occupant load is also considered as internal gains in the household. We have implemented several presets for the heating and cooling setpoints, although these can be changed during the scenario request if needed. In cases where meter data for the HVAC load are available, we bypass the ETP model and insert the HVAC load as a CSV.
- **Appliance Data Availability:** In order to mimic the demo sites household behaviour as close as possible to the reality, appliance consumption data are required. In practice, most pilot sites do not provide fully instrumented households with dedicated meters for each appliance. To ensure that the Digital Twin remains functional, and capable of producing meaningful simulation outputs even in the absence of real appliance measurements, synthetic data generation is adopted for the required devices. The synthetic data generation process allows the system to approximate the expected behaviour of appliances by scaling, adapting, and time-stretching prototype cycles extracted from REFIT in order to match the desired nominal power ratings, typical usage durations, and activation frequencies associated with the target household.

1.1.4 Development plan for next version

For the next version of the Household Dital Twins, several actions are planned:

- Integration of the actual household static data from the demo partners to be available for configurations creation as well as scenario creation and simulation execution.
- Integration with available appliance data from the demo sites metered data as well as the forecasted data from the different DIGITISE tools (real time and historic).
- Results request interface to be developed as part of the API Connector in cooperation with the Configuration Engine, as well as results output format to be finalized.
- Authentication process for the secure usage of the Digital Twin.
- Support for DER assets usage such as solar generation systems, vehicle chargers, battery energy storage systems (BESS) with charge/discharge rates.

TABLE 3 – FEATURES OF HOUSEHOLD DIGITAL TWIN FOR THE 2ND VERSION

Feature ID	Feature Name	Notes
SE-01	Configuration of household models	Integration with demo sites household data.
SE-02	Execution of simulation scenarios	Complete.
CE-01	Formulation of the simulation’s input and output	Finalize simulation output.
CE-02	Setup of the simulation parameters	Finalize the setup of the simulation parameters.
CE-03	Triggering of the execution of a simulation	Complete.
CE-04	Storing of the simulation results	Implement database integration.
AC-01	Simulation execution and results request interface	Implement flexible querying of the simulation results through the database.
AC-02	Data Space communication Interface	To develop an interface for the secure, correct and robust communication with the DIGITSE Data Space.

2 DIGITISE End-User Applications

2.1 Aggregator Portfolio Management and Virtual Power Plant (VPP) Configuration Application

2.1.1 Overview

The Aggregator Portfolio Management and Virtual Power Plant (VPP) Configuration Application represents an advanced digital platform which enables the aggregation and real-time management of Distributed Energy Resources (DERs) for commercial optimization. It functions as a complete toolset which allows energy aggregators along with VPP operators and utilities and grid operators (TSOs/DSOs) to realize the flexibility value of battery storage and renewable generation together with controllable loads. The primary function of the application enables the conversion of multiple DER assets into a unified dispatchable virtual power plant which delivers market services and grid support functions.

Through DIGITISE platform data infrastructure, the application delivers two main differentiators which include a modular customizable design and AI analytics engine that enables users to adapt VPP configurations for various market prospects and operational requirements.

The application contains multiple interconnected components which support the entire VPP management lifecycle according to the D2.2 architecture definition.

- DER Onboarding and Management: A foundational component for configuring individual DERs. It includes asset templates to streamline the integration of different technologies (e.g., batteries, solar PV, EV chargers) as part of the portfolio of the VPP → Linked to the **PMC-03** as defined in D2.2.
- Portfolio Configuration and Visualization: An interactive dashboard that allows users to group individual DERs into distinct portfolios or VPPs. Users can define specific objectives for each VPP (e.g., geo-temporal or business -oriented) and visualize the capacity, location, and real-time status of their assets on a map-based interface → Linked to the **PMC-01/ PMC-02** as defined in D2.2.
- AI-Powered Portfolio Clustering Engine: This component utilizes machine learning models to generate accurate VPP clusters for key variables, including renewable energy generation (solar, wind), load consumption, and the available upward/downward flexibility from the aggregated portfolio over different time horizons. → Linked to the **PMC-01/ PMC-02** as defined in D2.2.
- Real-Time Optimization and Dispatch: The core intelligence of the application. This engine uses advanced algorithms to calculate the optimal dispatch schedule for the VPP based on grid service requests, and the physical constraints of the

aggregated assets, ensuring maximum value extraction while respecting technical limits. → Linked to the **VPP-01/ VPP -02** as defined in D2.2.

- DSO services Integration and Bidding Module: This component provides the interface to connect with energy markets. It automates the process of submitting bids and offers for flexibility services and reports the dispatch instructions back to the optimization engine → Linked to the **VPP-01** as defined in D2.2.
- Analytics and Performance Reporting: A comprehensive module for tracking the operational performance of the VPP. It provides detailed reports on service delivery, asset utilization allowing for continuous strategy refinement. → Linked to the **VPP-02** as defined in D2.2.

2.1.2 Delivered Functionality

The development status for the first version (v1) of the application, corresponding to the current status of the work as follows (linked with the functionalities mentioned above):

TABLE 4 – FEATURES OF VPP CONFIGURATION APP FOR 1ST VERSION

Feature ID	Feature Name	Development status for v1
<i>PMC-01</i>	Geo-Temporal Flexibility Asset Clustering	Completed
<i>PMC-02</i>	Business-Oriented Asset Grouping	Completed (baseline)
<i>PMC-03</i>	Custom Asset Selection for Portfolio Structuring	Completed
<i>VPP-01</i>	Flexibility Services for DSOs	Partially Completed (Integration with primary DSO service needs)
<i>VPP-02</i>	Dynamic Allocation of VPP Flexibility Assets	Completed (Core Reporting)

Overall, the listed functionalities provide comprehensive coverage for the aggregator requirements (AGG-01 through AGG-10) as defined in D2.1. Here is a breakdown of how the requirements are covered:

TABLE 5 – REQUIREMENTS MAPPING WITH FEATURES OF VPP CONFIGURATION APP

Requirement	Description	Mapped Feature(s)
<i>AGG-01</i>	Classify assets (contracted params)	PMC-02: Business-Oriented Asset Grouping
<i>AGG-02</i>	Classify assets (actual params)	PMC-03: Custom Asset Selection
<i>AGG-03</i>	Segment sources (spatial/temporal)	PMC-01: Geo-Temporal Flexibility Asset Clustering

		PMC-02: Business-Oriented Asset Grouping
AGG-04	Filter flexibility	PMC-01: Geo-Temporal Flexibility Asset Clustering PMC-02: Business-Oriented Asset Grouping PMC-03: Custom Asset Selection
AGG-05	UI for classification/segmentation	PMC-01: Geo-Temporal Flexibility Asset Clustering PMC-02: Business-Oriented Asset Grouping PMC-03: Custom Asset Selection
AGG-06	Define VPPs (characteristics)	VPP-01: Flexibility Services for DSOs
AGG-07	Configure VPPs (economic terms)	VPP-01: Flexibility Services for DSOs
AGG-08	Define VPPs (operational timeframes)	VPP-01: Flexibility Services for DSOs
AGG-09	Manual VPP reconfiguration (UI)	VPP-02: Dynamic Allocation of VPP Flexibility Assets
AGG-10	Monitor VPP performance (UI)	VPP-02: Dynamic Allocation of VPP Flexibility Assets

2.1.2.1 Technology Stack

The application is built on a modern, scalable, and secure technology stack designed for real-time data processing and robust performance.

- **Frontend:**
 - **Framework:** Vue.js for a responsive and interactive user interface [9].
 - **UI Components:** Tailwind CSS for a consistent and professional design system [10].
 - **Data Visualization:** D3.js for rendering complex charts, graphs, and map-based asset visualizations [11].
- **Backend:**
 - **Language & Framework:** Python [12] with FastAPI [13] for building high-performance, asynchronous APIs to support the management of the data required for the application

- **AI/ML Libraries:** Scikit-learn [14] for developing and deploying forecasting and optimization models, this is very important for the operation of the clustering techniques delivered as part of the functionality.
- **Database:**
 - **Time-Series Data:** Elastic [15] for efficient storage and querying of high-frequency operational data from DERs (flex related data series, energy data series etc.)
 - **Relational Data:** PostgreSQL [16] for managing user data, asset metadata, and VPP configurations.
- **Infrastructure & DevOps:**
 - **Architecture:** Microservices architecture to ensure modularity and scalability. The different services are deployed as different modules
 - **Containerization:** Docker [17] for services packaging in order to enable the easy deployment of the services

2.1.2.2 Implementation Details

As mentioned above, the DIGITISE Flexibility Analytics Suite is a web-based application designed for business stakeholders, primarily Aggregators, participating in the DIGITISE ecosystem. Its core purpose is to provide advanced tools for managing, analyzing, and dispatching a portfolio of distributed flexible assets. This tool empowers aggregators to:

- Organize and segment large portfolios of flexible assets.
- Analyze the static and time-series characteristics of individual assets and grouped clusters.
- Monitor and review flexibility campaigns triggered by third-party entities (e.g., Network Operators).
- Analyze the optimization and dispatch strategies.

The screenshots from the 1st version of the app are presented below taking into consideration the features presented above.

Access to the application is secured and streamlined through the DIGITISE project's standard Single Sign-On (SSO) authorization process. This familiar procedure requires the user to provide their personal account credentials (username and password), just as they would for any other application within the DIGITISE platform. This central mechanism handles all security, authentication, and authorization, granting seamless access upon successful verification.

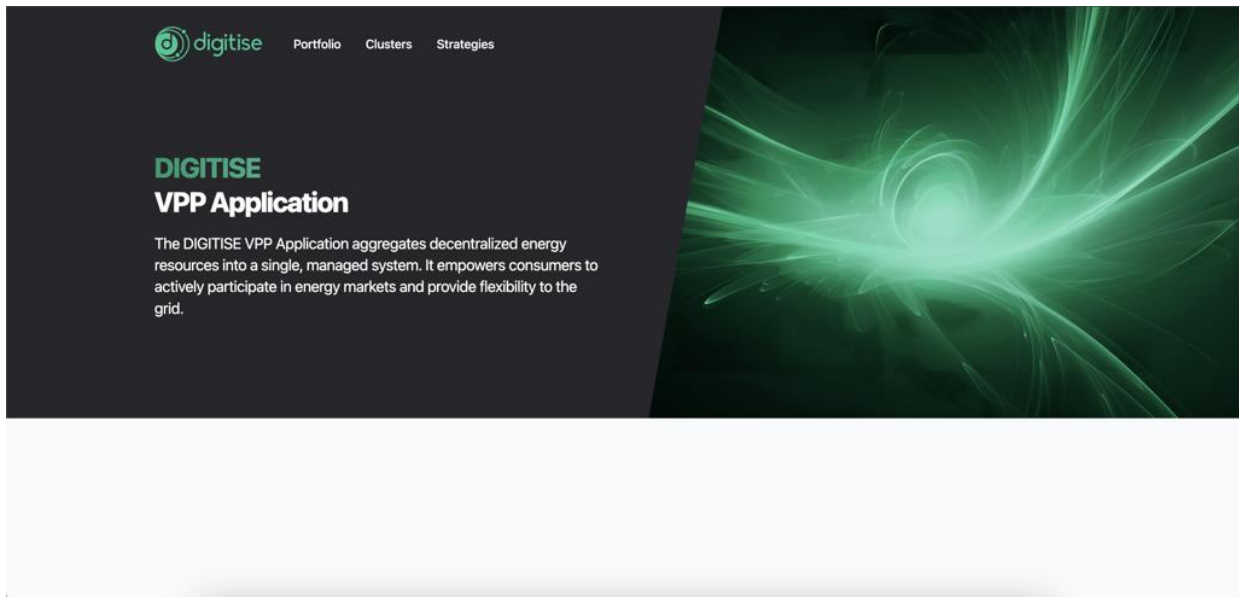


FIGURE 5 DIGITISE VPP LOGIN PAGE

Upon logging in, the aggregator is presented with the Portfolio Management module, which provides a comprehensive overview of their complete asset portfolio. The application provides filtering capabilities to manage and search the asset list. This filtering relies on a multi-dimensional classification system, allowing users to define highly specific criteria:

- **Operational Criteria:** Users can filter based on the physical and technical attributes of the assets, such as spatial filtering (location) or other operational characteristics (e.g., asset type, power rating).
- **Business Criteria:** Users can filter based on commercial parameters, such as participation in different contractual agreements, active business periods, or other contractual limitations.

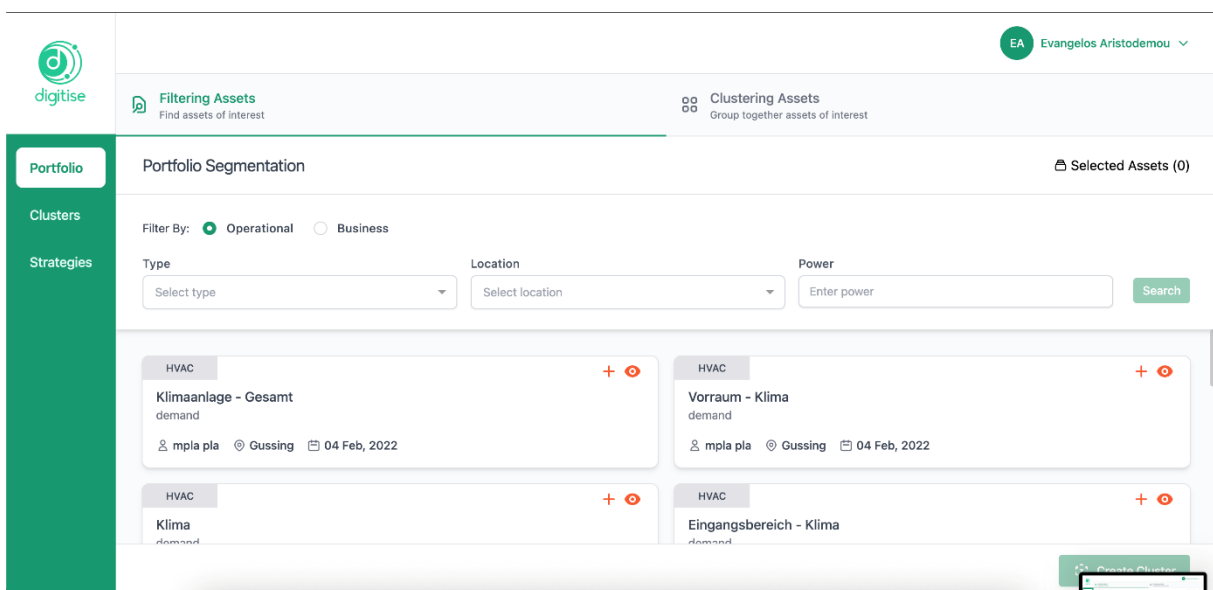


FIGURE 6 DIGITISE VPP PORTFOLIO OVERVIEW

A key feature of the suite is the ability to create logical 'clusters' or 'groups' from the asset list. After applying the desired filters, the user can select a specific set of assets from the refined list to form a new cluster. The application then prompts the user to assign a custom name and a short description for easy identification and future use.

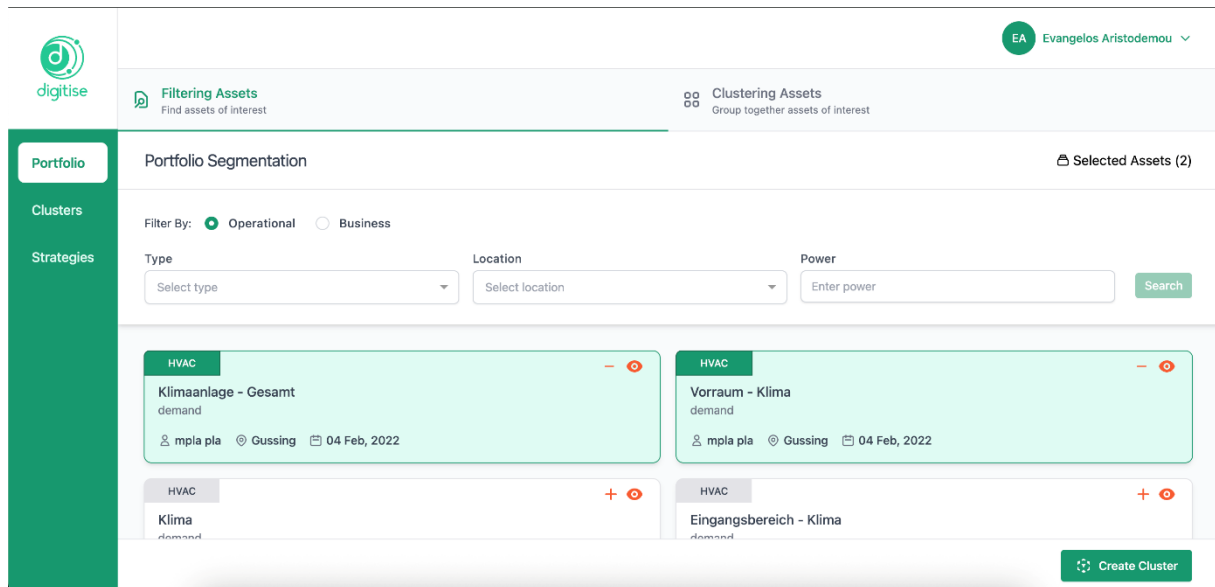


FIGURE 7 DIGITISE VPP PORTFOLIO OVERVIEW – ASSETS SELECTION

A dedicated view within the application manages all clusters defined by the business actor. From this list, the user has several options:

- View: See the list of specific assets participating in a group.
- Edit: Reshuffle or modify the list of assets within that cluster.
- Delete: Remove the cluster definition.

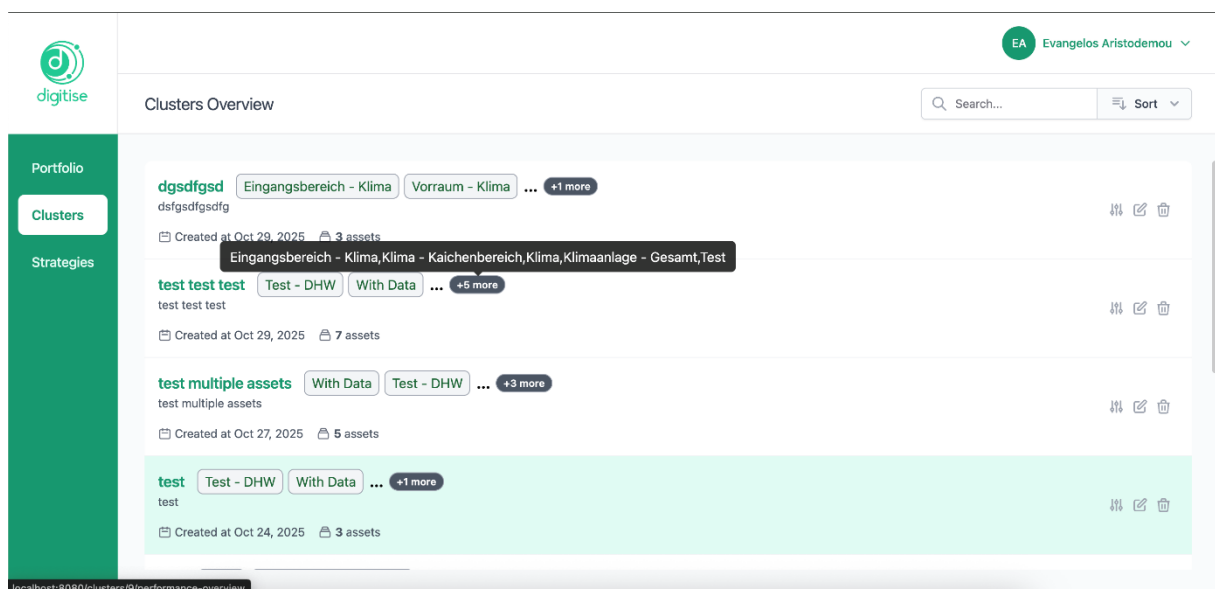


FIGURE 8 DIGITISE VPP CLUSTERS OVERVIEW

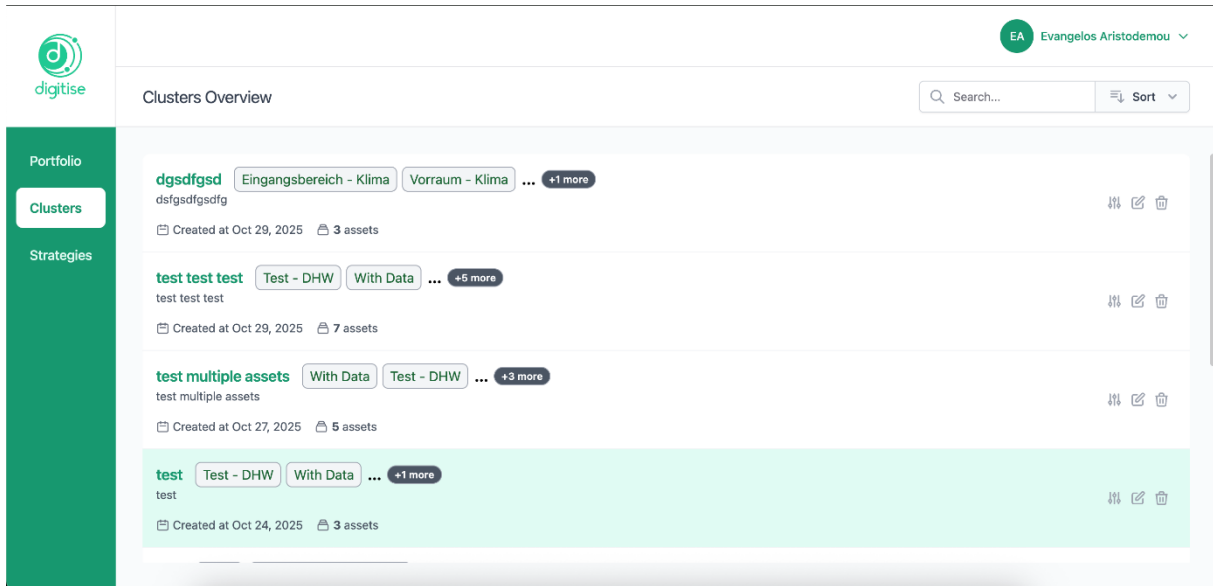


FIGURE 9 DIGITISE VPP CLUSTERS OVERVIEW – FILTERING

The tool provides multi-level drill-down capabilities for detailed analysis:

- **Individual Asset View:** The business actor can select any individual asset to gain deep insights into its flexibility-related characteristics. The tool presents detailed information, including:
 - Static Data: Nominal power, ramp rates, flexibility capacity, etc.
 - Dynamic Data: Time-series-based information (e.g., historical consumption/generation) presented in visualizations.
- **Cluster-Level View:** The user can select a specific group to analyze its collective properties. This view provides:
 - Time-series flexibility information for each individual asset within the group.
 - A tabular visualization offering insights into the collective performance of the group over an aggregated period. This table highlights key metrics such as total flexibility potential and maximum flexibility potential for the entire cluster.

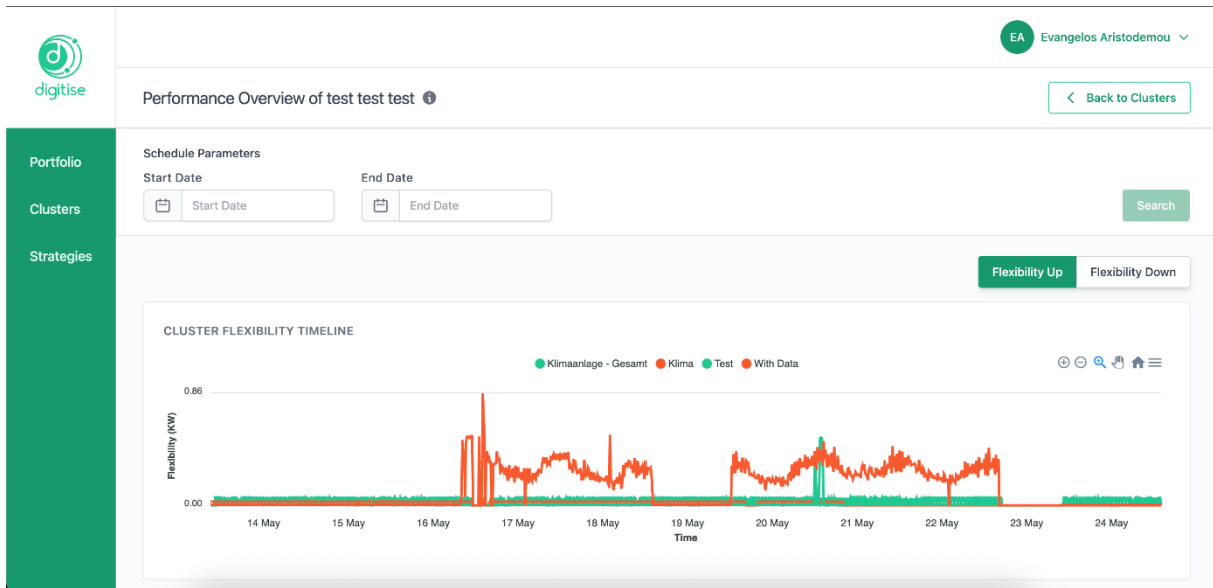


FIGURE 10 DIGITISE VPP CLUSTERS OVERVIEW – INSIGHTS

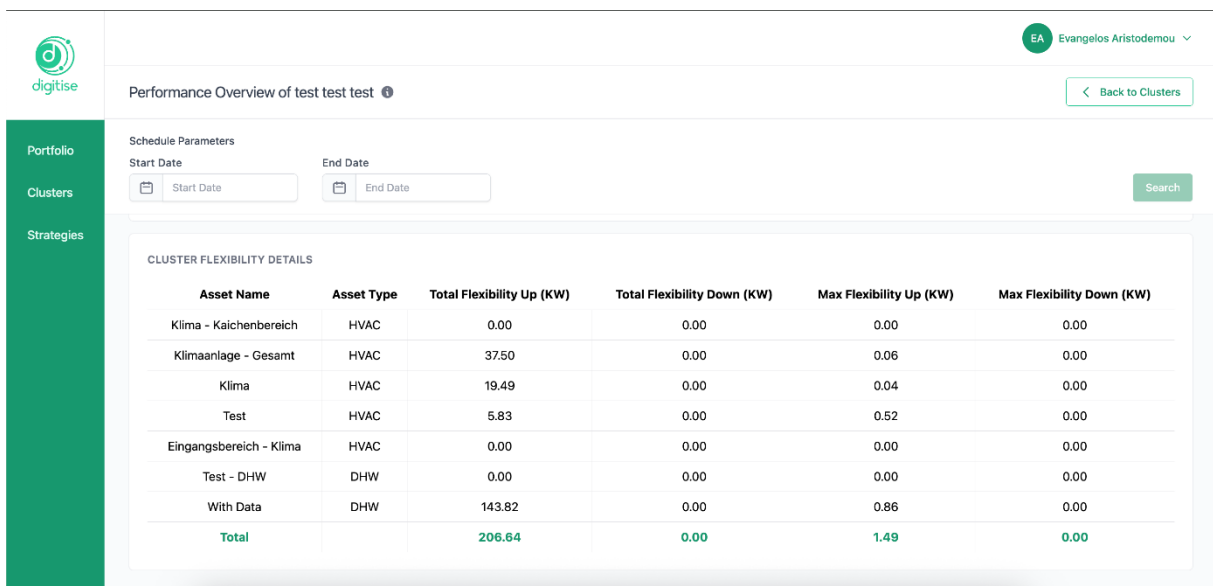


FIGURE 11 DIGITISE VPP CLUSTERS OVERVIEW – KPIs

Complementary to portfolio management, this module provides visualizations related to the analysis performed by the central DIGITISE Virtual Power Plant (VPP) component. A 'Strategies Tab' provides a dashboard of all flexibility events. The business stakeholder can access the full list of events (campaigns) triggered by third-party entities, such as network operators.

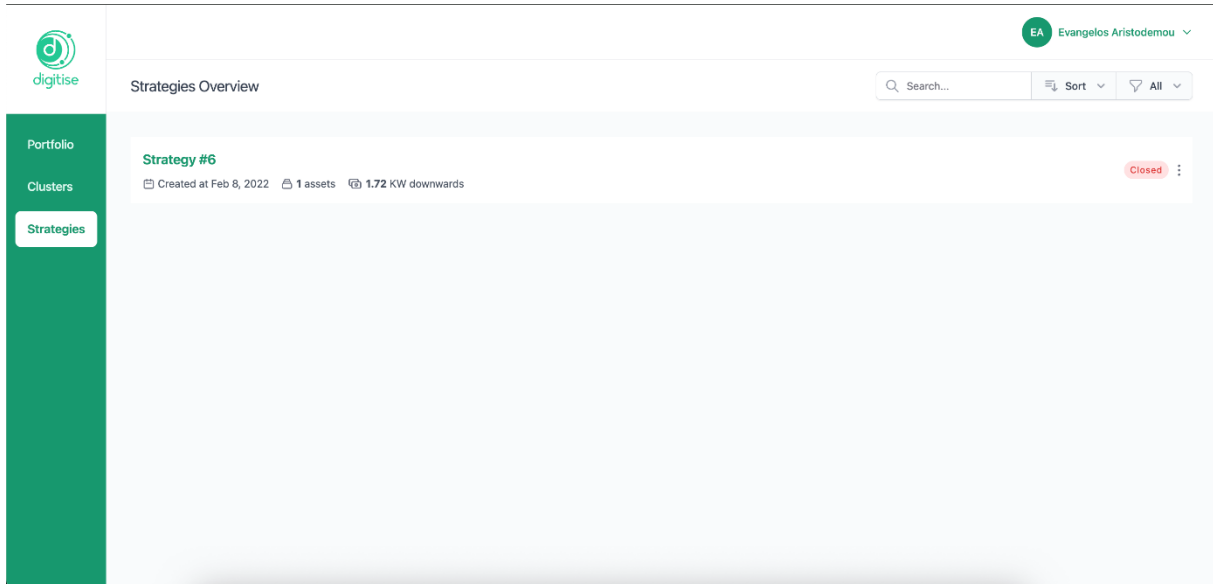


FIGURE 12 DIGITISE VPP STRATEGIES OVERVIEW

From the overview list, the end-user can select a single strategy to get detailed insights on its two primary components: the external request and the internal response.

1. Campaign Request: This details the parameters of the flexibility request as triggered by the third-party network operator. It includes information about the requested flexibility at each specific time step of the campaign.
2. VPP Response: This shows the results of the DIGITISE VPP's optimization process for this specific event. It presents the flexibility-related parameters for the pool of assets that were selected for activation, detailing the requested flexibility per asset at each specific time step of the portfolio strategy.

The user can also drill into the history of closed campaigns to retrieve information associated with asset performance during those specific events. This analysis view includes activation details and actual performance of the different flexible assets under that specific business request. As a side functionality, the end-user of the tool (from the management section of the strategies list) may also remove one or more strategies from the campaigns overview list. The aforementioned functionality is listed as the implementation of the 1st version of the application in the context of DIGITISE project.

2.1.3 Assumptions and Constraints

The first version (v1) of the application has been developed with the following assumptions and constraints:

1. Limited Market Integration: v1 focuses on integration with specific ancillary service markets Integration with a broader view of flexibility products/ services is planned for a future release.

2. Defined DER Protocols: The initial version supports a defined set of common DERs of interest in the project. Support for additional DER types will be added in subsequent versions.
3. Core UI Functionality: The user interface provides all essential functionality for VPP configuration and monitoring. Advanced UI customization and user-defined reporting features are limited in v1, with enhancements especially in the strategies tab to be provided in the final version.
4. Scalability Scope: Performance and scalability have been validated for VPP portfolios considering a limited number of DER assets. Further stress testing and optimization for larger-scale VPPs are part of the v2 development plan.

2.1.4 Development plan for next version

The roadmap for the next major version (v2) will focus on expanding functionality, enhancing intelligence, and improving scalability based on feedback from early adopters. Key development steps include:

1. Expanded Market Integration through Developed modules for participation in local flexibility markets operated by DSOs. Better alignment with flexibility services /needs from DSOs
2. Broader DER Support as to increase the library of supported DERs to allow for more diverse and heterogeneous VPP portfolios.
3. The forecasting and optimization algorithms to support strategies implementation need to develop self-learning features along with better probabilistic forecasting capabilities and automated anomaly detection tools for connected asset monitoring.
4. The platform requires thorough architectural optimization together with extensive load testing to validate its performance for large-scale VPP deployments and guarantee high system availability.

It is evident that while a mature version of the app was made available in the 1st version, updates and enhancements will be performed in the 2nd version. Below, the full list of features to be supported is listed (as per D2.2).

TABLE 6 – FEATURES OF VIRTUAL POWER PLANT (VPP) CONFIGURATION APPLICATION FOR 2ND VERSION

Feature ID	Feature Name	Notes
<i>PMC-01</i>	Geo-Temporal Flexibility Asset Clustering	Finalization and integration with actual data
<i>PMC-02</i>	Business-Oriented Asset Grouping	Finalization and integration with actual data

<i>PMC-03</i>	Custom Asset Selection for Portfolio Structuring	Incorporation of fine-grained models for the VPP formation
<i>VPP-01</i>	Flexibility Services for DSOs	Expanded Market Integration
<i>VPP-02</i>	Dynamic Allocation of VPP Flexibility Assets	Fine grained optimization and fine tuning of the UI

2.2 Flexibility Marketplace Environment

2.2.1 Overview

The Flexibility Marketplace of the DIGITISE project is coming to equip prosumers and aggregators (LEC managers) with the means to perform flexibility-related transactions in a secure and trustworthy environment. Prosumers are empowered to offer flexibility for each of their assets and aggregators are able to propose new contracts for exploiting the offered flexibility in future transactions with the system operators. The implementation of the application steps upon the novel distributed ledger technology (DLT) for sealing each transaction on the blockchain and offers transparency over the settlement of the established contracts by publishing relevant remuneration information on the blockchain.

All operations of the Flexibility Marketplace are realized by its three components, as described in D2.2: The **Marketplace User Interface** (MUI), which is the main point of interaction with the application for its users, the **Flexibility Configuration and Remuneration Manager** (FCR) that is responsible for all the back-end operations of the application and finally the **DLT Smart Contract Engine** (SCE) which allows all the blockchain-related transactions to be performed either through the UI or the back-end processes. The different functionalities covered by these components and their respective features are presented below.

A flexibility contracting procedure stands at the core of all operations of the Flexibility Marketplace. For this to be possible, configuration of prosumers' assets is required by their side for flexibility to be offered and assets to be able to be included in flexibility contracts (covered by features MUI-01, MUI-02 and FCR-02). A contract can be then introduced by an aggregator, providing a set of contract terms (i.e. contract period, included assets, activation schedules, remuneration prices), which are in turn verified by the related prosumer who either directly agrees on the terms or proposes alternatives. When an agreement is reached between the two parties, the contract is published and signed on the blockchain. The introduction, negotiation and establishment of smart flexibility contracts is covered by features MUI-01, MUI-03, MUI-04, FCR-01 and SCE-01. Finally, after the activation of related DR events, and in accordance with the provided settlement results by the VPP Configuration application, a validation and verification process is run in order to define the respective compensation for the prosumers. This procedure includes

exploitation of measured consumption retrieved from the DIGITISE Data Space and relevant calculated baseline data, that can be used to prove whether the system operator’s request has been fulfilled or not, resulting in revenues or penalties for the prosumers. The defined amounts are made available to both prosumers and aggregators through the UI, in the form of detailed tables and intuitive visualizations, for further monitoring of their performance in terms of flexibility delivery and financial gains. As mentioned, for transparency reasons, the calculated amounts are logged on the chain through the DLT Engine. The remuneration-related processes are covered by features MUI-05, FCR-03, FCR-04 and SCE-02.

2.2.2 Delivered Functionality

The current status of the development for the 1st version of the Flexibility Marketplace application, is presented per feature at the table below.

TABLE 7 – FEATURES OF FLEXIBILITY MARKETPLACE FOR 1ST VERSION

<i>Feature ID</i>	<i>Feature Name</i>	<i>Development status for v1</i>
<i>MUI-01</i>	Flexible assets and contracts listing	Partially Completed – Main listing and filtering
<i>MUI-02</i>	Flexible asset configuration by prosumers	Completed
<i>MUI-03</i>	Flexibility contract initiation and negotiation of terms	Completed
<i>MUI-04</i>	Triggering of smart flexibility contract publishing and signing	Completed
<i>FCR-01</i>	Flexibility contract management (creation and update)	Completed
<i>FCR-02</i>	Flexibility asset management	Completed
<i>SCE-01</i>	Smart contract deployment and signing	Completed

The requirements that are described in D2.1 and need to be covered during the development of the Flexibility Marketplace are listed below and are mapped with the corresponding application features that fulfill them.

TABLE 8 – REQUIREMENTS MAPPING WITH FEATURES OF FLEXIBILITY MARKETPLACE APP

<i>Requirement ID</i>	<i>Description</i>	<i>Mapped Feature(s)</i>
<i>AGG-11</i>	Provide a flexibility marketplace environment to allow prosumers to configure their flexible assets’ characteristics, i.e. the offered capacity	MUI-01, FCR-02

AGG-12	Provide aggregator with the ability to get a full list of flexible assets or search/filter them based on provided criteria, i.e. the type of the flexible asset	MUI-01, FCR-02
AGG-13	Equip an aggregator with the ability to introduce a new flexibility contract, where all the necessary contract details are defined in a standardized way	MUI-03, FCR-01
AGG-14	Prosumers are allowed to negotiate contract terms introduced by an aggregator by providing an altered version of the flexibility contract with edited details	MUI-03, FCR-01
AGG-15	Provide aggregators the possibility to publish and sign a smart flexibility contract on the Blockchain after both partners have agreed on terms	MUI-04, SCE-01
AGG-16	Allow prosumers to sign a smart contract through the flexibility marketplace environment	MUI-04, SCE-01
AGG-17	Support the flexibility market with automated mechanisms for monitoring and calculating remuneration amounts, exploiting also relevant AI analytics for baselining	FCR-03, FCR-04, SCE-02
AGG-18	Provide aggregators with a UI to monitor their contracts' performance and relevant analytics at LEC level, in the form of event based as well as weekly and monthly reports	MUI-05, FCR-04
AGG-19	Allow prosumers, through an intuitive UI, to monitor their individual contracts' performance along with insightful analytics in the form of event based as well as monthly reports	MUI-05, FCR-04
AGG-20	Allow prosumers to be prioritized in the flexibility contract introduction process based on their assets' flexible capacity	MUI-01, MUI-03, FCR-02

From the main functionalities described in the previous section, those features related to the process of remuneration definition, logging and visualization are those which are not included in the development plan for the 1st version of the Flexibility Marketplace. However,

they are included in the requirements mapping table for clarity and coherence. The rest, that cover the asset configuration, contract introduction and negotiation, and relevant smart contracts interactions are practically fully developed during the current version, with only small adjustments expected, so as to set a strong basis for the integration with the DIGITISE Data Space as well as other applications (i.e. VPP Configuration application), and for the revenues and penalties calculation procedure.

2.2.2.1 Technology Stack

For the implementation of the three components of the Flexibility Marketplace application the following technologies were utilized in order to deliver a web-based application:

- **Marketplace User Interface:** Angular is used as the main development framework for the application UI. The web3.js [18] library is exploited for integrating with the blockchain network and the browser-based Ethereum wallet (Rabby) [19]. For the different visualizations to be offered, d3.js [11] and ApexCharts.js [20] libraries will be exploited.
- **Flexibility Configuration and Remuneration Manager:** As the main backend component, FCR Manager is developed with the Python-based [12] Django web framework [21]. The related django-rest-framework.py library [22] is used on top of that, offering REST API-like capabilities for communication with both internal and external components but also for data exchange purposes. The web3.py library [23] is also exploited for interacting with the blockchain network and the smart contracts. For storing any data, i.e. during the asset configuration as well as contract introduction and negotiation phases, a PostgresDB [16] serves as a relational database for the Marketplace backend operations.
- **DLT Smart Contract Engine:** The stack used for the DLT SC Engine includes a private Proof-of-Authority network, configured for testing purposes in the scope of the DIGITISE project, based in go-ethereum (geth) [24]. Smart contracts that are used for any blockchain transactions are written using the Solidity language [25] and are compiled and tested with the exploitation of Hardhat framework [26].

The deployment of the components on the production server happens in a containerized way, in the form of microservices architecture, using Docker and docker-compose [17].

2.2.2.2 Implementation Details

The Flexibility Marketplace is offered in two forms, one for prosumers and one for aggregators, with slight differences between the two versions, but enough in order to separate the application functionalities and actions to be available according to each type of stakeholder. Below, screenshots from the application's user interface are provided. It should be considered that most of the functionalities described below are implemented in coordination with the main backend component (Flexibility Configuration and Remuneration Manager), which receives object creation, update or retrieval requests and

after proper validation it proceeds with serving the requested actions, which include interaction with the application’s database.

Since the Flexibility Marketplace stands as a web application, it can be reached by both kinds of users through a web browser and accessed with the usage of valid credentials (Figure 13).

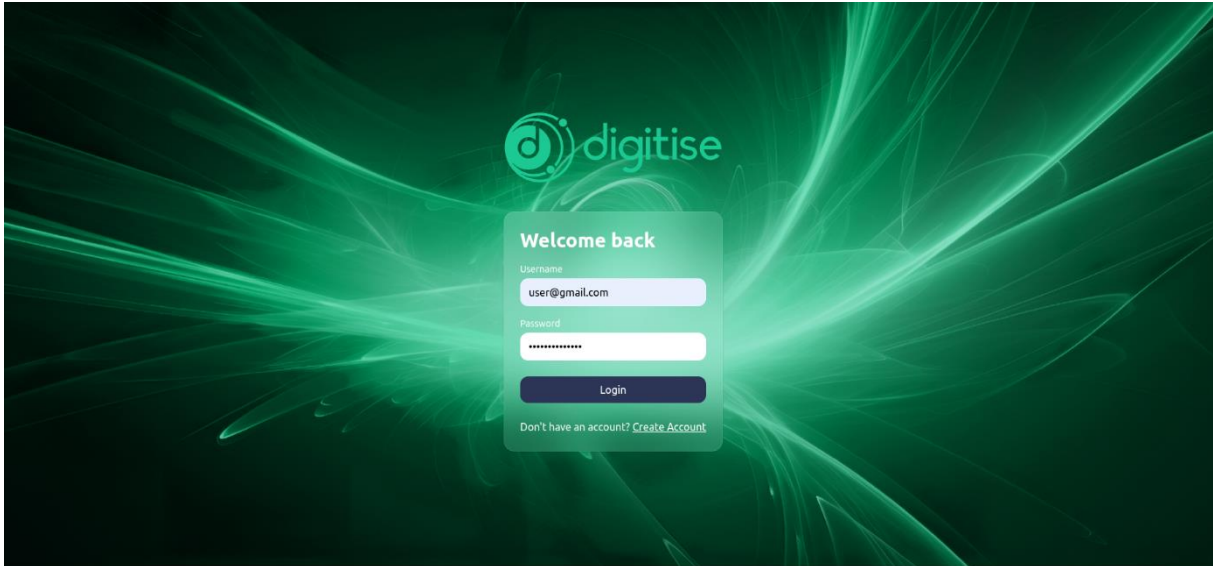


FIGURE 13 – USER LOGIN PAGE OF THE FLEXIBILITY MARKETPLACE

A logged-in prosumer is able to see a list of all flexible assets in their disposal (Figure 14), filter and sort the provided list (Figure 15), but also search for assets by providing several criteria, i.e. the asset’s location (Figure 16). This asset listing feature is also provided to the aggregator, so as to be able to monitor the flexible assets of their portfolio.

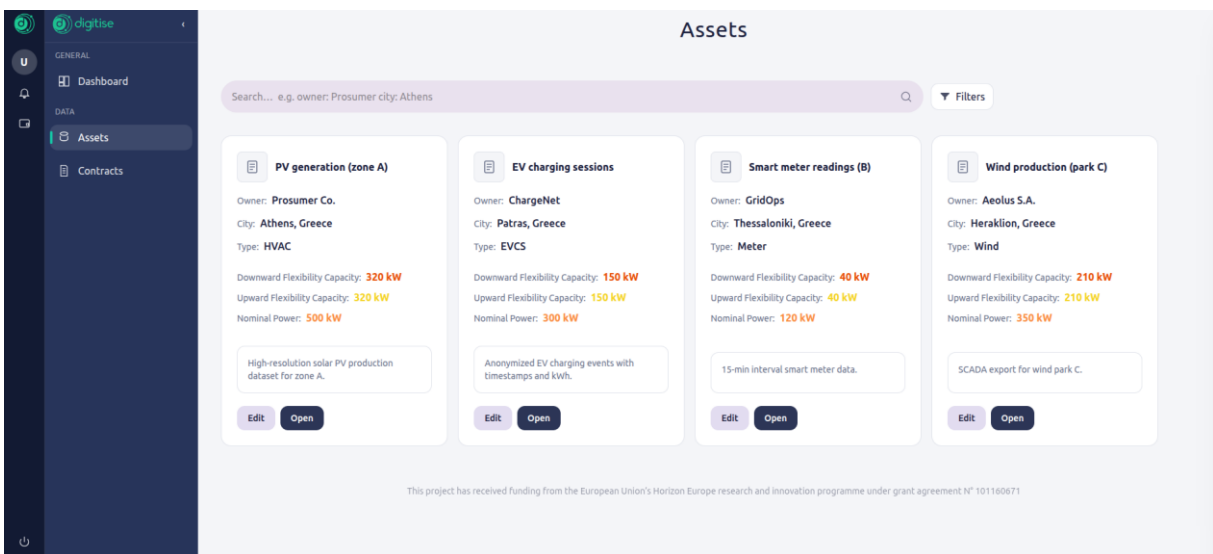


FIGURE 14 – ASSETS LISTING FROM PROSUMER VIEW

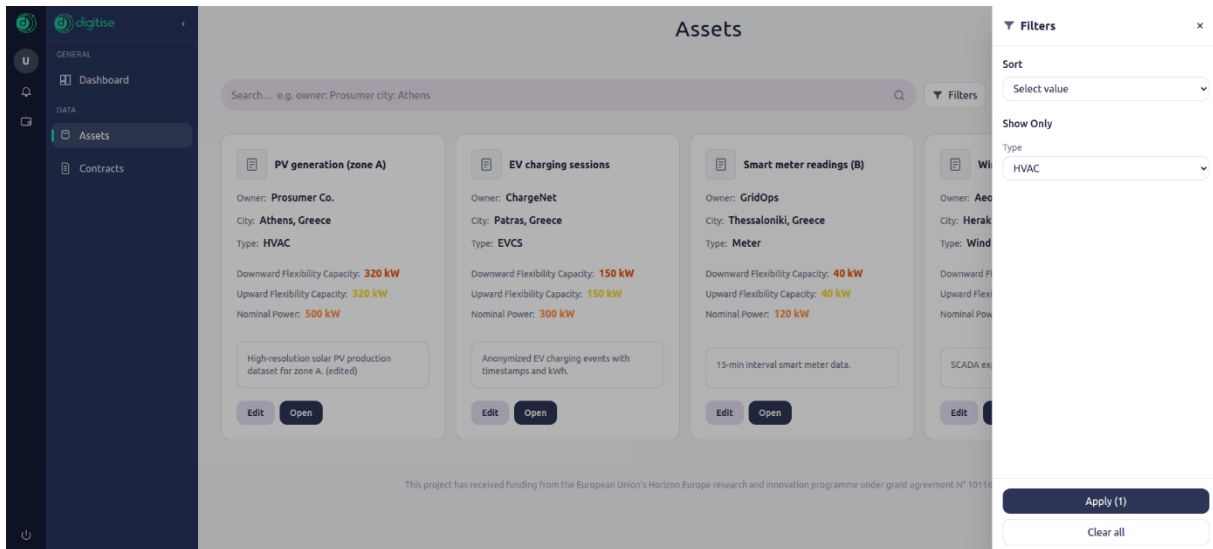


FIGURE 15 – ASSETS FILTERING FROM PROSUMER VIEW (SAME FOR ANY USER)

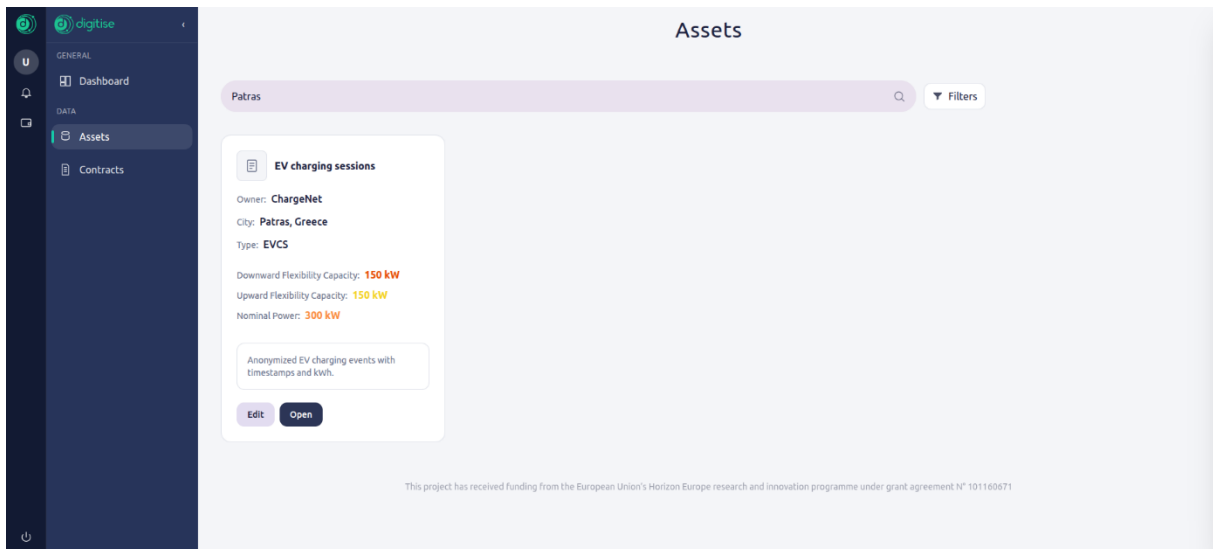


FIGURE 16 – ASSETS SEARCHING FROM PROSUMER VIEW (SAME FOR ANY USER)

One major difference, in comparison to aggregators, is that prosumers are also allowed to configure their assets, by defining the amount of flexibility offered to be available during the assets' participation in DR programs, along with a few additional customization capabilities (Figure 17). This is the first step for prosumers to start engaging with aggregators in flexibility transactions and actively participate in the energy market. As can be seen in Figure 18, aggregators can only preview the details of an asset but not alter them.

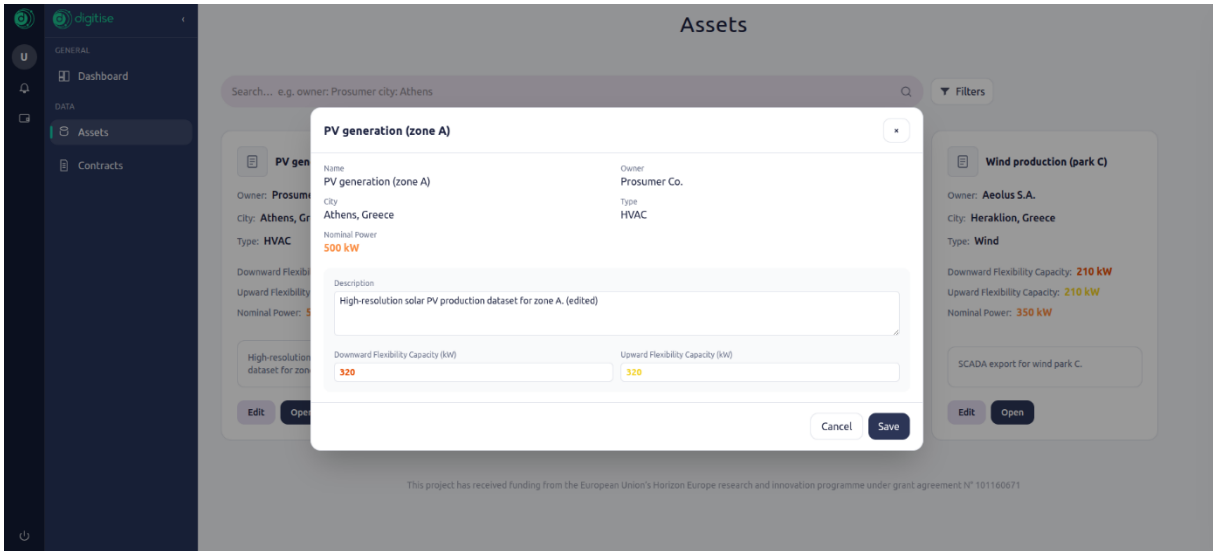


FIGURE 17 – ASSET EDIT CAPABILITY FOR PROSUMER

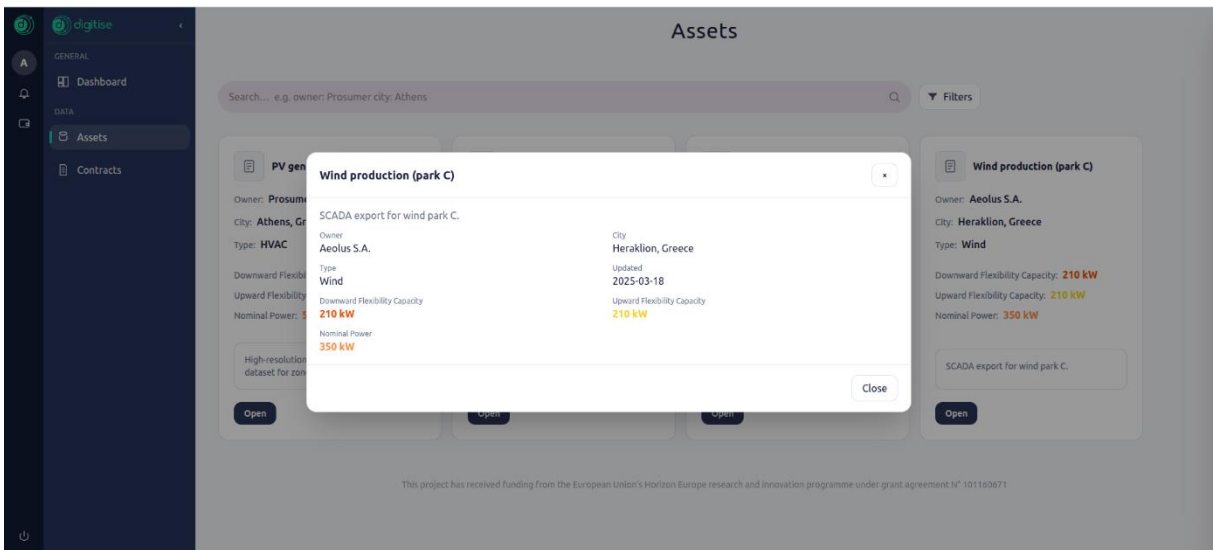


FIGURE 18 – ASSET PREVIEW CAPABILITY FOR AGGREGATOR

The next set of application features refers to the functionalities of contract handling and management. Both prosumers and aggregators can see a list of the contracts they are participating, along with quick-view, useful information regarding, among others, the participants, creation dates and statuses (Figure 19). Search and filtering capabilities are also offered as part of the contract listing functionality.

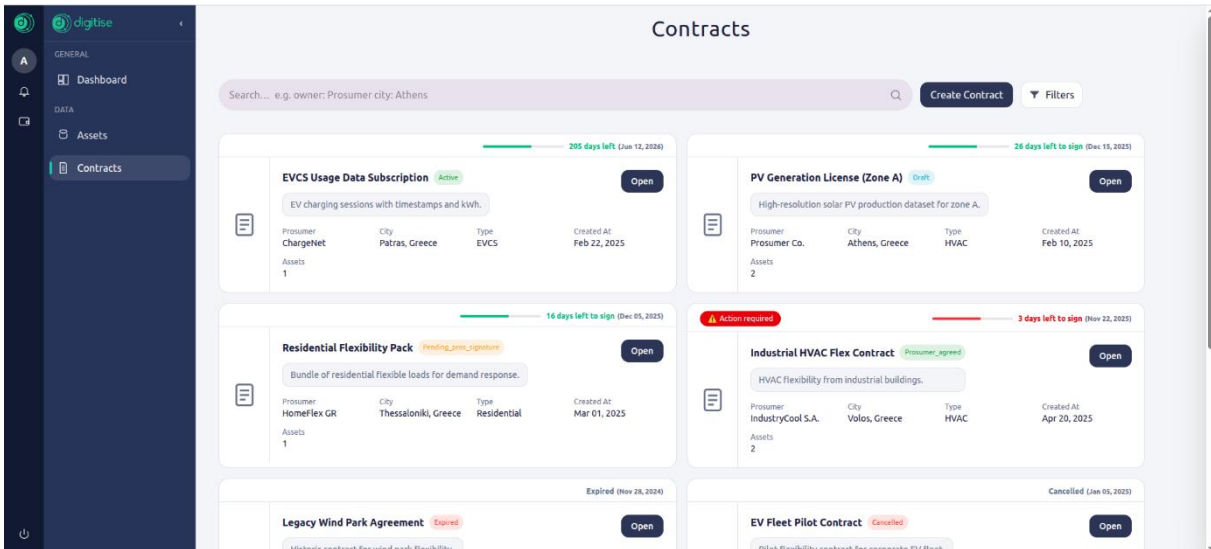


FIGURE 19 – CONTRACTS LISTING FROM AGGREGATOR VIEW (SEARCHING AND FILTERING POSSIBLE FOR ALL USERS)

In the case of the aggregator’s view of the Flexibility Marketplace, an additional feature that differentiates their role from those of the prosumer, is the ability to initiate a new contract (Figure 20). This process includes the definition of the contract terms, by setting up a contract name, description and deciding on the prosumer with whom to establish the agreement. The rest of the remaining terms include the participating assets, contract duration, activation schedules and financial details, based on which the relevant remuneration amounts will be determined after the contract settlement.

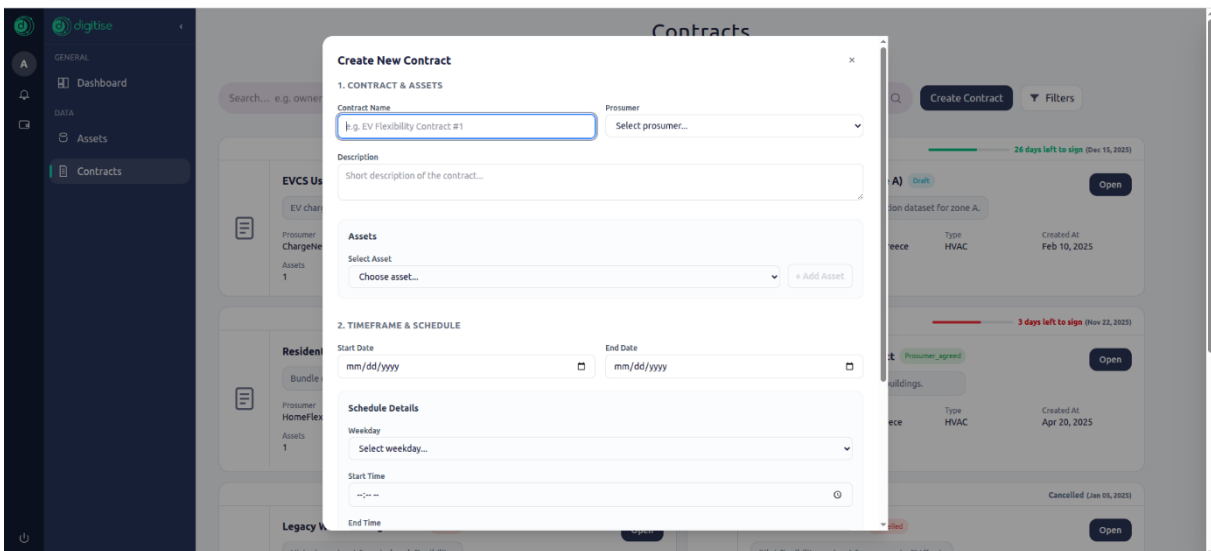


FIGURE 20 – NEW CONTRACT CREATION FROM AGGREGATOR VIEW

The process continues with the contract details review by the prosumer, and the direct agreement or proposal of counter terms based on the ones originally introduced by the aggregator. Once the contract terms have been finalized and agreed by both parties, the

contract is ready to be published on the blockchain, and at the same time signed, by the aggregator (Figure 21).

The smart contract utilized for this purpose, as it is already mentioned, is written in Solidity, where several variables and functions have been properly designed to make sure that only specific accounts can interact with them or retrieve contract information.

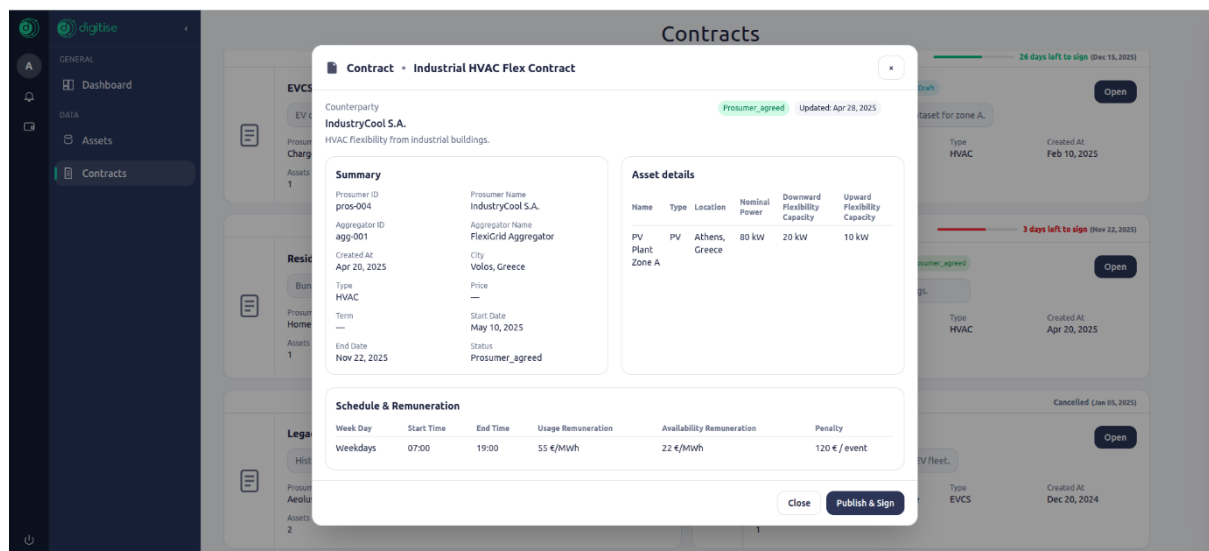


FIGURE 21 – CONTRACT PUBLICATION AND SIGNING FEATURE FOR AGGREGATOR

Once this procedure is completed successfully, the prosumer is responsible for sealing the agreement by signing the smart contract on the blockchain from his side (Figure 22). For signing the smart contract, but also for deploying it on the blockchain by the aggregator, a connection to the user’s Ethereum wallet is required (Figure 23). The usage of personalized wallets ensures that any sensitive information of the users’ blockchain account is totally detached from the application for further security and trustworthiness. Only their public account addresses are used for their relationship with the published contracts.

It is important to note that the blockchain network, to which the smart contracts are published and through which any interaction with them is performed, is private PoA network, utilized only in the scope of the DIGITISE project. If the users have not been connected before to the network with their wallet, they are prompted to add the network to their wallet for all the upcoming transactions. Any (ETH) funds used for the different blockchain transactions on this private network are totally fictitious.

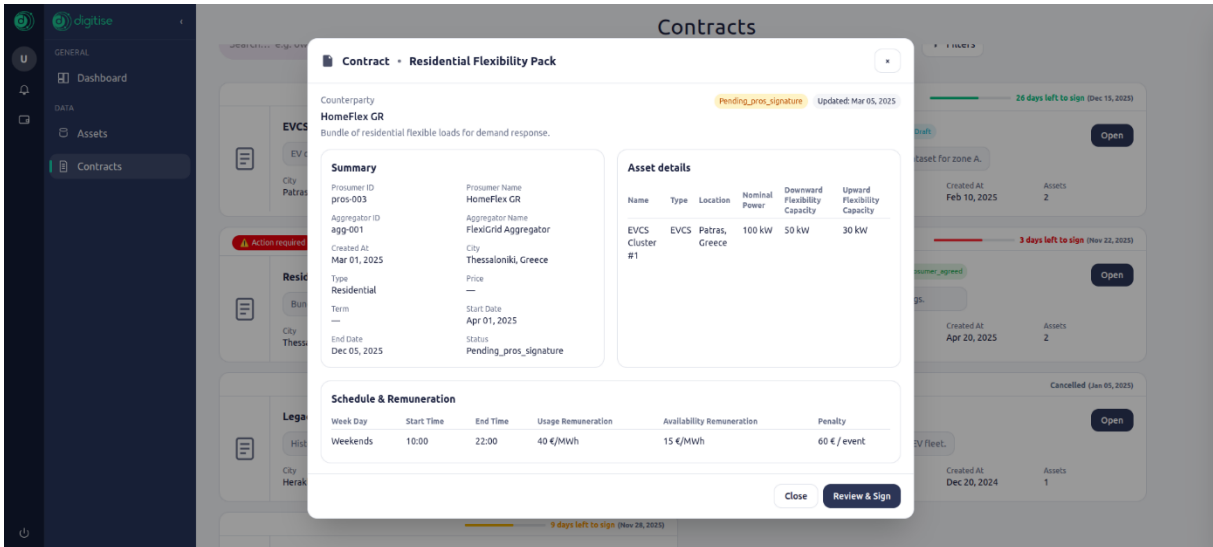


FIGURE 22 – CONTRACT PREVIEW AND SIGNING PROMPT FOR PROSUMER

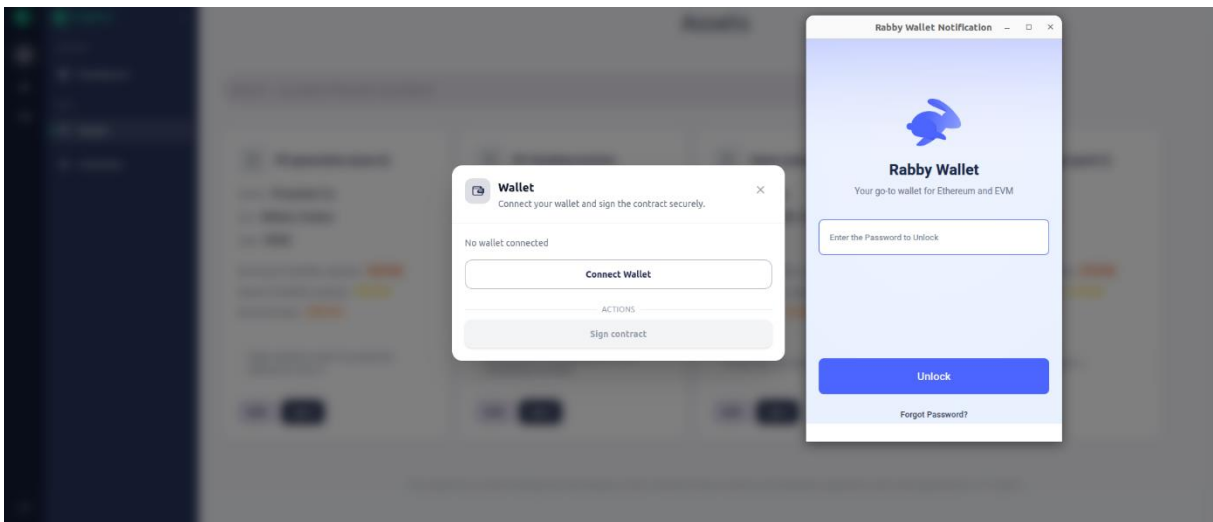


FIGURE 23 – CONNECTION TO RABBY WALLET POP-UP

With the signing process performed by both parties, the contract is considered as “Active”. The terms of the contract are expected to be utilized by the VPP Configuration application, as the next step of the prosumers’ participation in the energy market.

2.2.3 Assumptions and Constraints

As of the 1st version of the Flexibility Marketplace application, the following points have been considered during implementation of the delivered features:

1. Integration with actual data: Currently, since the integration with the prosumers’ actual device static data, which would provide us with the information needed for the flexible assets’ configuration, mock data have been used for testing and validation. Relevant WP6 data landscaping activities and a static data collection process has also been initiated as part of the WP5 work, to ensure that the

information needed for the different functionalities of the application will be available.

2. Test users: for implementation and testing reasons, fictional users are created (both prosumers and aggregators) and are linked to specific test blockchain accounts. Further actions are expected to be performed regarding the authentication process once it is defined by the different application owners.
3. Contract terms: The values used as contract parameters regarding the prices of the different amounts (flexibility request coverage, deficiency and availability) are also synthetic. For the final version, we assume that during the application usage by the aggregators, the latter could define those values after consulting the community's performance and the general flexibility market trends of the respective country.

2.2.4 Development plan for next version

For the next version of the Flexibility Marketplace there are two major actions planned: the first is the integration with actual demo data, which includes both static information of the flexible devices per prosumer as well as dynamic consumption data from the devices. Furthermore, the data integration will include consumption forecast results as well as demand-response (DR) event request data along with settled flexibility amounts for each event. All the above will lead to the finalization of flexible asset configuration and management procedures but also to the realization of the second major functionality, which is the remuneration definition process. The latter will combine all this data to calculate the respective amounts, which in turn will be made available through the UI for visualization, but will also be logged on the blockchain.

Moreover, as already mentioned, test users are currently utilized for the validation of the implemented features, so it is expected that, for the 2nd version, the authentication system which is planned to be utilized will provide user information that will be made available from the demo partners. This, in combination with the data integration, will further lead to any customizations needed for each demo site.

It should be noted that the features presented in Table 9 as completed already cover the planned functionalities, although some effort is expected in case several adjustments or enhancements are needed (especially after integration with the demo data).

Below, the full list of features to be supported (as per D2.2) is provided.

TABLE 9 – FEATURES OF FLEXIBILITY MARKETPLACE APPLICATION FOR THE 2ND VERSION

Feature ID	Feature Name	Notes
<i>MUI-01</i>	Flexible assets and contracts listing	Finalization of filtering or asset grouping capabilities and integration with actual data
<i>MUI-02</i>	Flexible asset configuration by prosumers	No major updates are expected. Integration with actual asset static data is pending.
<i>MUI-03</i>	Flexibility contract initiation and negotiation of terms	Only visual enhancements (if any) to be provided.
<i>MUI-04</i>	Triggering of smart flexibility contract publishing and signing	Polishing or enhancement of Ethereum wallet integration (if needed).
<i>MUI-05</i>	Contracts performance visualizations	To be developed for the 2 nd version.
<i>FCR-01</i>	Flexibility contract management (creation and update)	No updates. Possibility for performance optimization (if needed).
<i>FCR-02</i>	Flexibility asset management	In accordance with MUI-02 – Integration with actual asset data pending for possible refinements.
<i>FCR-03</i>	API-based data exchange	To be developed for the 2 nd version.
<i>FCR-04</i>	Definition of prosumers remuneration amounts	To be developed for the 2 nd version.
<i>SCE-01</i>	Smart contract deployment and signing	No updates expected.
<i>SCE-02</i>	Logging of remuneration amounts	To be developed for the 2 nd version.

2.3 Behavioural Profiling and Insights Application

2.3.1 Overview

The scope of the work in Task 5.4 is to design behavioural profiling services for residential consumers. These services will not only give information to prosumers regarding their behavioural patterns but also provide them with insights regarding their trends and how they compare against others. Additionally, for the energy providers, high-level overviews will be generated, giving insights into their clients. More specifically, the aim of this task is to deliver a suite of functionalities with two main purposes: (i) extract from household-collected data behavioural patterns, lifestyle indicators, and contextual trends, and (ii) allow authorised users and third parties to leverage these insights for other, personalized services, enhance the consumers’ experiences, and participate in the data economy.

The behavioural profiling and insights application has three main components. The user interacts with the application through an Interface that provides historical and real-time views for a variety of behavioural data. The outcomes of the app are made visible to the users to view trends, clustering data, and personalized insights on their behaviour. While this first version of the application currently supports some visualization functionality, importantly, for the second and final version of the application, all user interaction and visualization will happen through and via the Dashboard. Second, the AI Behavioural Profiling Engine is the most integral part of the application, as it serves as its intelligence core. It analyses user behaviour and household usage patterns, and occupancy trends. At the same time, a key aspect in the design and implementation of this service is the real-time integration and analysis of IoT-based data (e.g., sub-metering, energy consumption, environmental conditions) combined with Household Digital Twin models and personal analytics. This can in turn enable clustering, and behavioural modelling, outlining actions related to appliance usage, environmental awareness, lifestyle routines, and influenced by prosumer preferences. Importantly, all entities have tailored access rights to the output of the application, ensuring that profiling and data-sharing are conducted in a secure, ethical, and user-empowering manner.

During the 1st release, the behavioural profiling and insights application realizes a subset of the functionalities expressed in D2.2. First, it supports configurable trend generation across various aspects of the household. Additionally, the user can utilize the application to see their respective ranking across other users in pre-defined labels characterizing their behaviour. Last, for the energy providers, initial clustering is supported. Grouping essentially the prosumers into static labels. Lastly, the application exposes API endpoints for its outputs, allowing for other components/applications to utilize them for other functionalities. These are validated against placeholder authorization tokens and are ready to connect.

2.3.2 Delivered Functionality

The current status of the development for the 1st version of the Behavioural Profiling and Insights application, is presented per feature at the table below.

TABLE 10 – FEATURES OF BEHAVIOURAL PROFILING APP FOR 1ST VERSION

Feature ID	Feature Name	Development status for v1
<i>IEI - 01</i>	Software Interface for behaviour tracking and segmentation	Partially Complete. Trends across any specified category/attribute over configurable time periods.
<i>IEI - 02</i>	Visualization of lifestyle clusters and personal baseline comparisons	Partially complete. User rankings across any specified category over a period of time. User lifestyle labels {early_bird, night_owl}.

<i>BPE - 02</i>	Behavioural segmentation and clustering engine	Partially complete. User behavioural labels {energy conscious, high consumption}.
<i>DEG - 01</i>	Controlled data sharing APIs for third-party service providers	Partially complete. API(s) are validated against placeholder authorization tokens and are ready to connect.

The requirements that are described in D2.1 and need to be covered during the development of the Behavioural Profiling application are listed below and are mapped with the corresponding application features that fulfill them.

TABLE 11 – REQUIREMENTS MAPPING WITH FEATURES OF BEHAVIOURAL PROFILING APP

Requirement	Description	Mapped Feature(s)
<i>PRO-44</i>	Implement real-time monitoring and reporting for energy consumption and generation	IEI-01, IEI-02, IEI-03
<i>PRO-45</i>	Provide inputs to a user-friendly dashboard for monitoring and control of devices	IEI-01, IEI-02, IEI-03
<i>PRO-46</i>	Provide inputs to a user-friendly dashboard for monitoring sensing conditions	IEI-01, IEI-02, IEI-03
<i>PRO-47</i>	Implement historical data monitoring and reporting for energy consumption.	IEI-03
<i>PRO-51</i>	Implement real-time monitoring of energy performance and self-consumption indicators	BPE-02, BPE-03
<i>PRO-55</i>	Leverage info on frequent activities in the households and their flexibility at an individual level	BPE-01, BPE-04
<i>PRO-56</i>	Provide a comparative view with neighbours' performance	IEI-01, BPE-02, BPE-03

2.3.2.1 Technology Stack

For the initial version, a standalone application for the Behavioural profiling application has been developed. This is mainly comprised of back-end components, utilizing an internal database for the evaluation of the engine components. Below, an overview is provided regarding the supported functionalities.

The application is written in the Gleam programming language [27] targeting the BEAM virtual machine [28]. It can be deployed as a docker container using the provided Dockerfile [17]. The application can be configured to serve requests on any IPv4/IPv6

address. Currently, the application pulls the data from an API operated by VOLTIVA Energy, and indexes it by using SQLite3 [29]. It is used to perform various data analytics tasks exposed as app functionalities via APIs. All APIs are tested using Hurl [30] with nearly 100% test coverage.

2.3.2.1 Implementation Details

As mentioned above, for the development of the delivered features and functionalities for this 1st version of the application, data from VOLTIVA Energy were utilized, which proved to be a highly suitable candidate even though it is not publicly available. This dataset contains various measurements from different devices inside households. For this 1st version all the available data was considered in the development of the initial analytics and clustering algorithms. Consumption data was considered, and data preprocessing was needed and performed. At a high level, the application supports trend generation, user rankings, and user behavioural labels. As initial examples for this version the ranking was performed using the lifestyle labels {early_bird, night_owl}, and behavioural labels {energy conscious, evening routine, high consumption}. Below, screenshots from the implementation are presented alongside early-stage visualization. Figure 24 shows the number of days a user was active inside the house at each hour over a year, while Figure 25 shows the 10 least and most active users given a certain duration, respectively. Additionally, Figure 26 lists the 10 most active users between 7:00–8:00, which are given the label “early birds”, and the 10 most active users between 0:00–5:00, which are given the label “night owls”. Last, in Figure 27 for two behavioural labels (“energy-conscious” and “high-consumption”), the users falling into these categories are presented across a year. For the “energy-conscious” label we monitor users who alter their behaviour after exceeding a certain threshold (thus reporting additionally to the IDs the number of instances over a year this behaviour was exhibited), while for the “high-consumption” label we showcase the users whose average annual consumption exceeds a certain threshold. Last, Figure 28 depicts the trend of the temperature across a year, on a monthly basis, for one household.

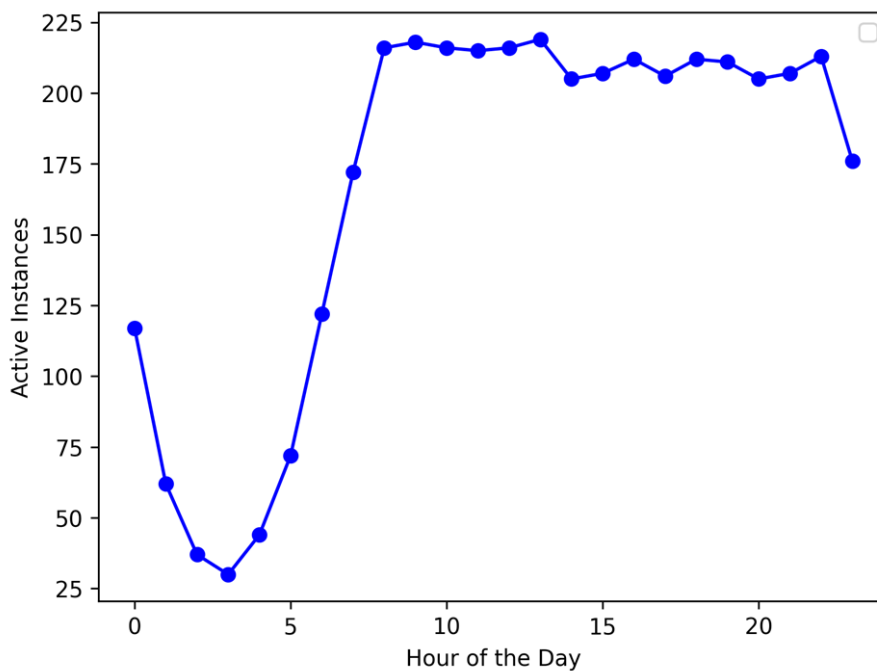


FIGURE 24 – USER 36 ACTIVITY OVER A YEAR PER HOUR

#	user	count
0	MIW-U0032	34
1	MIW-U0028	53
2	MIW-U0024	96
3	MIW-U0034	143
4	MIW-U0033	150
5	MIW-U0025	223
6	MIW-U0031	227
7	MIW-U0036	227
8	MIW-U0029	234
9	MIW-U0026	286

#	user	count
0	MIW-U0030	314
1	MIW-U0035	291
2	MIW-U0026	286
3	MIW-U0029	234
4	MIW-U0036	227
5	MIW-U0031	227
6	MIW-U0025	223
7	MIW-U0033	150
8	MIW-U0034	143
9	MIW-U0024	96

FIGURE 25 – LEAST AND MOST ACTIVE USERS RESPECTIVELY BETWEEN 13:00 AND 18:00, ON A DAILY BASIS

#	user	count	#	user	count
0	MIW-U0030	276	0	MIW-U0026	269
1	MIW-U0026	266	1	MIW-U0030	216
2	MIW-U0035	256	2	MIW-U0035	190
3	MIW-U0036	222	3	MIW-U0036	185
4	MIW-U0029	194	4	MIW-U0029	174
5	MIW-U0025	182	5	MIW-U0025	142
6	MIW-U0033	152	6	MIW-U0031	129
7	MIW-U0034	139	7	MIW-U0033	77
8	MIW-U0031	119	8	MIW-U0028	76
9	MIW-U0028	92	9	MIW-U0024	56

FIGURE 26 – TOP-10 USERS ACCORDING TO LIFESTYLE {EARLY BIRDS, NIGHT OWLS}, WITH INSTANCE COUNTS ACROSS A YEAR-LONG DURATION

#	user	count
0	MIW-U0022	182
1	MIW-U0023	1
2	MIW-U0024	28
3	MIW-U0025	462
4	MIW-U0026	24
5	MIW-U0027	183
6	MIW-U0028	240
7	MIW-U0029	87
8	MIW-U0030	2
9	MIW-U0031	364
10	MIW-U0033	2
11	MIW-U0034	24
12	MIW-U0035	23
13	MIW-U0037	39

0	MIW-U0023
1	MIW-U0032
2	MIW-U0035

FIGURE 27 – ENERGY-CONSCIOUS INSTANCES PER USER AND HIGH-CONSUMPTION USERS OVER A YEAR

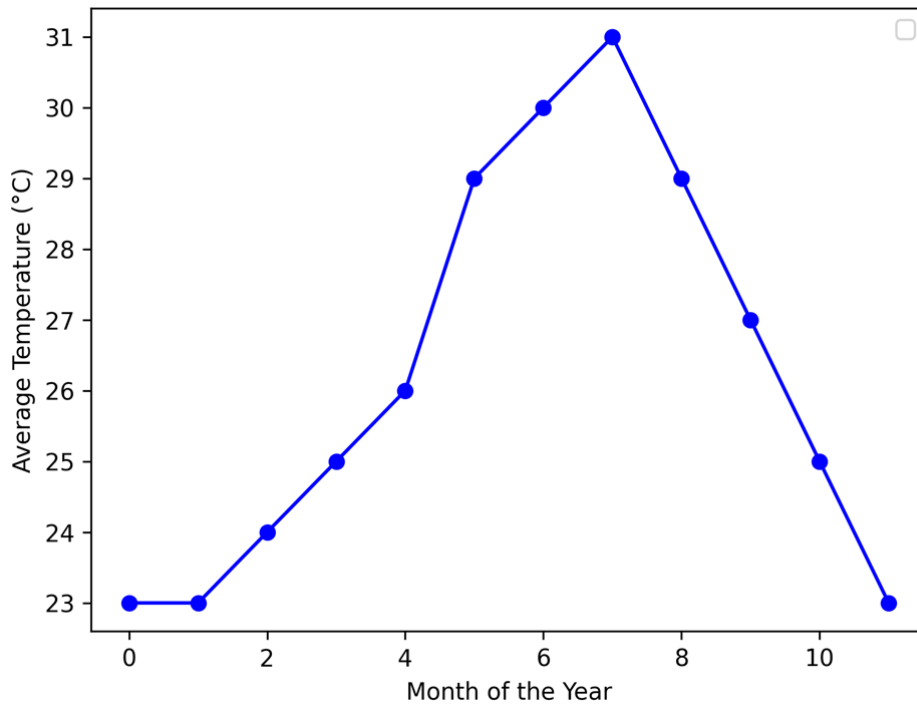


FIGURE 28 – AVERAGE TEMPERATURE PER MONTH OF USER 36 HOUSEHOLD, OVER A YEAR

2.3.3 Assumptions and Constraints

1. **Connectivity:** Data Space and Dashboard integration are missing currently.
2. **Evaluation:** The application has been tested for scalability and performance based on its supporting initial-version operations on data sourced from an existing dataset. Further testing is needed upon access to real-time data.
3. **Functionality:** The engine of the application provides the user with essential functionality in terms of historical trends, clustering, and static labelling.

2.3.4 Development plan for next version

For the application to be fully functional and connected to the DIGITISE ecosystem, the components need to be fully developed, enabling the realization of the remaining features as well. As specified in D2.2, first, the Interface needs to be able to support additional insights for the household residents, incorporate more behavioural labels and extract easy-to-comprehend reports. Second, the profiling engine component needs to be further developed. Utilization of the Digital Twins is planned, to further enhance the underlying profiling models. Moreover, the application will support identification of new patterns in behaviour and enrich the user’s behaviour profiles accordingly. Lastly, clustering over the users will become fully developed to map with the needs of the prosumers and provide insights into their everyday routines.

For technological aspects of the implementation, the following observation can be made: A fusion of techniques is needed to realize the desiderata for this application. Various ML

models will be explored for classification and prediction of the behaviour of the users. These predictions can then be used as input for anomaly detection and assist in generating clusters dynamically. Additionally, occupancy trends can be extracted, which can lead to better and more efficient energy management of the household. Below, the full list of features to be supported is listed (as per D2.2).

TABLE 12 – FEATURES OF BEHAVIOURAL PROFILING APPLICATION FOR THE 2ND VERSION

Feature ID	Feature Name	Notes
<i>IEI - 01</i>	Software Interface for behaviour tracking and segmentation	To be completed in v2. Routine trends to be finalized.
<i>IEI - 02</i>	Visualization of lifestyle clusters and personal baseline comparisons	To be completed in v2. Enrichment of the clustering mechanisms and labels.
<i>IEI - 03</i>	Attribute inference reports	To be developed in v2.
<i>BPE - 01</i>	Real-time fusion of IoT data streams with Digital Twin models	To be developed in v2.
<i>BPE - 02</i>	Behavioural segmentation and clustering engine	To be completed in v2. Develop models (using the provided data and specification). Incorporate personalization.
<i>BPE - 03</i>	Lifestyle profiling and behavioural characteristic extraction	To be developed in v2.
<i>BPE-04</i>	Continuous profile enrichment based on observed data and seasonal dynamics	To be developed in v2.
<i>DEG-01</i>	Controlled data sharing APIs for third-party service providers	To be completed in v2. Remove placeholders and validate with real tokens.

3 Conclusions

The first release of the DIGITISE Digital Twin solutions and tools represents a major milestone in the project's mission to empower consumers and prosumers in the energy transition. In the basis of knowledge enrichment over the households and consumers there are two applications under development. First it is the Household Digital Twin, which provides a robust simulation environment for household energy modelling, enabling optimization of consumption, flexibility, and integration of renewable resources. Next comes the Behavioural Profiling and Insights application which delivers a personalized consumption analysis and lifestyle characterization, helping households and energy providers better understand behavioural patterns and improve engagement in data and energy markets. On the other hand, the Flexibility Marketplace and the Aggregator Portfolio Management and VPP Configuration applications are here to directly engage prosumers in the energy market economy. The first one establishes a secure and transparent environment for prosumers and aggregators to negotiate and settle flexibility contracts, leveraging blockchain-based smart contracts for trust and accountability. The VPP Configuration application introduces advanced DER clustering and optimized dispatch scheduling mechanisms, allowing distributed energy resources to be aggregated into virtual power plants that deliver market services and grid support.

Together, these tools form a coherent ecosystem that enhances consumer engagement, supports flexibility transactions, and strengthens the integration of renewable energy resources. While this first release focuses on foundational features, the roadmap for subsequent versions emphasizes on integration with actual (historical and real-time) data, expanded market participation and interactions between applications for exchanging data and insights, and exploitation of enriched AI-driven analytics. Overall, "D5.1 – 1st Release of DIGITISE DT Solutions and Tools for Consumers Engagement in Marketplace Environments" highlights the progress made development-wise during the 1st release period and lays the groundwork for the first demonstration period and the initial validation process for the WP5 applications and tools.

References

- [1] DIGITISE HE, "D2.1 – DIGITISE End-User Requirements and Socio-Economic Analysis," 2025.
- [2] DIGITISE HE, "D2.2 – DIGITISE Evaluation Protocol, Novel Market Models and Architecture Design," 2025.
- [3] "GridLAB-D," U.S. Department of Energy (DOE), 2017. [Online]. Available: <https://www.gridlabd.org/>. [Accessed November 2025].
- [4] D. (. S. L. (. S. V. (. Murray, "REFIT: Electrical Load Measurements (dataset)," 16 June 2016. [Online]. Available: <https://pureportal.strath.ac.uk/en/datasets/refit-electrical-load-measurements-cleaned> . [Accessed June 2025].
- [5] C. K. C. H. M. a. E. W. Klemenjak, "A synthetic energy dataset for non-intrusive load monitoring in households.," *Scientific Data*, vol. 7, no. 1, p. 108, 2020.
- [6] H. & S. V. & S. L. & S. P. Rashid, "Evaluation of Non-intrusive Load Monitoring Algorithms for Appliance-level Anomaly Detection," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.
- [7] L. K. D. B. C. Lawrie, "Development of Global Typical Meteorological Years (TMYx)-Region 6 Europe," Climate.OneBuilding.Org, [Online]. Available: https://climate.onebuilding.org/WMO_Region_6_Europe/default.html#gsc.tab=0. [Accessed November 2025].
- [8] "GridLAB-D Document Archives – Technical Support for Residential Module," U.S. Department of Energy (DOE), [Online]. Available: <https://www.gridlabd.org/documents/archives.stm>. [Accessed June 2025].
- [9] Vue.js Community, "Vue.js – The Progressive JavaScript Framework," 2016. [Online]. Available: <https://vuejs.org/>. [Accessed November 2025].
- [10] Tailwind CSS, "Tailwind CSS – Rapidly build modern websites without ever leaving your HTML," [Online]. Available: <https://tailwindcss.com/>. [Accessed November 2025].
- [11] D3.js, "d3.js – The JavaScript library for bespoke data visualization," 2011. [Online]. Available: <https://d3js.org/>. [Accessed November 2025].
- [12] Python Software Foundation, "Python programming language," 1991. [Online]. Available: <https://www.python.org/>. [Accessed November 2025].
- [13] S. Ramírez, "FastAPI – modern, fast, web framework for building APIs with Python," 2018. [Online]. Available: <https://fastapi.tiangolo.com/>. [Accessed November 2025].
- [14] G. V. A. G. V. M. Fabian Pedregosa, "Scikit-learn: Machine Learning in Python," INRIA, 2010. [Online]. Available: <https://scikit-learn.org/>. [Accessed November 2025].

- [15] Elastic, "Elasticsearch – OPEN SOURCE SEARCH, ANALYTICS, AND AI PLATFORM," 2010. [Online]. Available: <https://www.elastic.co/elasticsearch/>. [Accessed November 2025].
- [16] PostgreSQL , "PostgreSQL – a powerful, open source object–relational database system," 1996. [Online]. Available: <https://www.postgresql.org/>. [Accessed November 2025].
- [17] Docker, "Docker – Accelerated, Containerized Application Development," Docker Inc., 2013. [Online]. Available: <https://www.docker.com/>. [Accessed November 2025].
- [18] Web3.js, "Web3.js – TypeScript libraries for Interaction with EVM Chain," 2014. [Online]. Available: <https://web3js.org>. [Accessed November 2025].
- [19] Rabby, "Rabby Wallet – Wallet for Ethereum and EVM," DeBank, 2021. [Online]. Available: <https://rabby.io/>. [Accessed November 2025].
- [20] Juned Chhipa, Brian Lagunas and ApexCharts Open–source Community, "ApexCharts.js – Modern & Interactive Open–source Charts," 2018. [Online]. Available: <https://apexcharts.com/>. [Accessed November 2025].
- [21] Django Software Foundation, "Django – High–level Python web framework for quick app development," 2005. [Online]. Available: <https://www.djangoproject.com/>. [Accessed November 2025].
- [22] "Django REST framework – A powerful and flexible toolkit for building Web APIs," Encode OSS Ltd, 2011. [Online]. Available: <https://www.django-rest-framework.org/>. [Accessed November 2025].
- [23] Ethereum Foundation, "web3.py – A Python library for interacting with Ethereum," 2016. [Online]. Available: <https://web3py.readthedocs.io/en/stable/>. [Accessed November 2025].
- [24] Ethereum Foundation, "go–ethereum – Official Go implementation of the Ethereum protocol," Free Software Foundation, Inc., 2015. [Online]. Available: <https://geth.ethereum.org/>. [Accessed November 2025].
- [25] Ethereum Foundation, "Solidity – A statically–typed programming language for developing smart contracts," Argot Collective, 2015. [Online]. Available: <https://www.soliditylang.org/>. [Accessed November 2025].
- [26] Nomic Foundation, "Hardhat – Ethereum development environment for professionals," [Online]. Available: <https://hardhat.org/>. [Accessed November 2025].
- [27] Louis Pilfold, "Gleam – A friendly language for building type–safe, scalable systems," 2016. [Online]. Available: <https://gleam.run/>. [Accessed November 2025].
- [28] Erlang/OTP, "The BEAM–Erlang’s virtual machine," 1998. [Online]. Available: <https://www.erlang.org/blog/a-brief-beam-primer/>. [Accessed November 2025].

[29] SQLite Consortium members, "SQLite – A C-language library that implements an SQL database engine," 2000. [Online]. Available: <https://sqlite.org/>. [Accessed November 2025].

[30] Orange Open-Source, "Hurl – A command line tool that runs HTTP requests," Orange S.A., [Online]. Available: <https://hurl.dev/>. [Accessed November 2025].