This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957845

LC-SC3-ES-3-2018-2020

Integrated local energy systems (Energy islands)

# RENergetic

Community-empowered Sustainable Multi-Vector Energy Islands

Project Nº 957845

# D3.1- Description of the interim version of the ICT tools developed for energy island communities

| | |
|---|---|
| Responsible: | Stepan Gagin (UNI PASSAU) |
| Document Reference: | D3.1 |
| Dissemination Level: | Public |
| Version: | 2.0 |
| Date: | 23th February 2023 |

# Executive Summary

This offers a comprehensive view on the work done in RENergetic work package 3 (WP3) as of beginning of April 2022. Furthermore, the document is relevant for project members involved in the pilot locations as it summarizes the main concerns and ideas from the ICT point of view.

This document is the first deliverable of WP3 and describes the interim version of the ICT RENergetic system that is being developed in the project. The requirements and overall vision of the system, as well as its functionalities are described.

The RENergetic system aims to provide visualizations of the data coming from energy island systems and corresponding forecasts for the different types of users, such as visitors, residents, and managers. The data and forecasts are also used by optimization algorithms and demand response programs to maximize the share of renewables, level of autarky and self-consumption. The optimization is performed in a hierarchical manner with two levels: A global *multi-vector* optimizer and *domain-specific* optimizers. Domain-specific optimizers solve an optimization problem tailored to a specific sub-system of the energy island. Examples for these sub-systems or domains are the heating domain, the electricity domain or electrical vehicle charging.

From a software perspective, the RENergetic system is designed as a service-oriented architecture. That is to say the system is built as a set of modular services that each perform some specific functionality and may communicate with other modules of the system. The RENergetic system is deployed in the supercomputing platform provided by a project member (Poznan Supercomputing and Networking Centre).

The development process in the project adapts the *Scrum* framework and other agile methods. Thus, the functionalities of the systems are formulated as user stories that are grouped into epics (i.e., collections of user stories related to a specific domain or module of the RENergetic system). The requirement analysis utilizes the Smart Grid Architecture Model (SGAM) approach, which is also used to ensure replicability of the system. This deliverable contains descriptions of the vision, mock-ups and details about algorithms for the following functionalities in the RENergetic system:

- Common information model.
- Forecasting.
- Demand response for electrical vehicle charging.
- Demand response for heat domain.
- Heat supply optimization.
- Local waste heat optimization.
- Interactive platform.

The objective of RENergetic is to demonstrate the viability of so-called ‘urban energy islands’. Energy islands seek to achieve the highest possible degree of self-sustainability with regards to the supply of its energy demand, be it electricity or heat through local renewable resources. At the same time, an urban energy island may offer ancillary services to the public grid surrounding it.

These islands place the consumer at the centre of the energy transition, giving them an active part in energy communities capable of producing their own energy, sharing the surplus with the rest of the public grid and optimizing consumption. RENergetic will demonstrate that Urban Energy Islands increase both the number of renewables in these areas and the energy efficiency of local energy systems. RENergetic will demonstrate the viability of this energy islands in three site pilots, each of them of a different nature: New Docks, a residential area in Ghent – Belgium, Warta University Campus in Poznan, Poland and San Raffaele Hospital and its investigation and research campus in Segrate-Milan, Italy. The impact of the Urban Energy

Islands is assured as technical, socio-economic, and legal / regulatory aspects are considered while safeguarding economic viability.

RENergetic will be carried out over the stretch of 42 months involving 14 European partners: Inetum (Spain, France, and Belgium), Clean Energy Innovative Projects and Gent University (Belgium), Poznan University of Technology, Veolia and Poznan Supercomputing and Networking Center (Poland), Ospedale San Raffaele, Comune di Segrate and University of Pavia (Italy), Energy Kompass GMBH (Austria), the University of Mannheim and the University of Passau (Germany), University of Stuttgart (Germany) and Seeburg Castle University (Austria).

# Contributors Table

| DOCUMENT SECTION | AUTHOR(S) | CONTRIBUTOR(S) TO RESULTS | REVIEWER(S) |
|---|---|---|---|
| **I Introduction** | Stepan Gagin, Dominik Danner (UNIPASSAU) | | Hermann de Meer (UNIPASSAU)<br>Marcin Adamski (PSNC) |
| **II.1. Requirement Analysis SGAM** | Michael Niederkofler (EK)<br>Benedikt Kirpes (UNIMA) | Ammar Alyousef (UNIPASSAU) | |
| **II.2. RENergetic System Logical Overview** | Stepan Gagin (UNIPASSAU)<br>Tomás García, Ubaldo Serna Pérez (Inetum ES) | Elias Goller, Daria Musikhina (UNIPASSAU) | |
| **II.3. RENergetic System Component View** | Tomás García, Ubaldo Serna Pérez (Inetum ES)<br>Tomek Ciesielczyk (PSNC) | Ammar Alyousef (UNIPASSAU) | |
| **II.4. Deployment** | Tomás García, Ubaldo Serna Pérez (Inetum ES)<br>Tomek Ciesielczyk (PSNC)<br>Stepan Gagin (UNIPASSAU) | Raúl Parra Ruiz (Inetum ES),<br>Marek Błażewicz (PSNC), Bogdan Ludwiczak (PUT) | |
| **II.5. Security Considerations** | Tomás García, Ubaldo Serna Pérez (Inetum ES) | | |
| **III Optimization Concept** | Dominik Danner, Armin Stocker (UNIPASSAU) | Ammar Alyousef, Stepan Gagin (UNIPASSAU)<br>Michael Niedermeier (EK) | |
| **IV.1. Common Information Model** | Tomek Ciesielczyk (PSNC)<br>Dominik Danner (UNIPASSAU) | Tomás García, Raúl Parra Ruiz (Inetum ES) | |
| **IV.2. Forecasting** | Daniele Baranzini (OSR) | Adrián Carrasco Revilla (Inetum ES)<br>David Bouret (Inetum BE) | |
| **IV.3. Demand Response for Electric Vehicle Charging** | Matthias Strobbe (UGent) | | |
| **IV.4. Demand Response for Heating Domain** | Sonja Klingert (UNIMA)<br>Adrian Carrasco (Inetum ES) | Mona Bielig (UNIMA)<br>Łukasz Malewski, Radosław Górzeński (PUT)<br>Daria Musikhina (UNIPASSAU) | |
| **IV.5. Heat Supply Optimization** | Lieven Demolder (CEIP)<br>Łukasz Malewski (PUT) | Radosław Górzeński (PUT) | |

| | | | |
|---|---|---|---|
| **IV.6. Local Waste Heat Optimization** | Franciszek Sidorski, Ariel Oleksiak (PSNC) | Łukasz Malewski, Radosław Górzeński (PUT) | |
| **IV.7. Interactive Platform** | Sonja Klingert (UNIMA) | Celina Kacperski (UNIMA) | |

# Table of Contents


I. INTRODUCTION ........ 11
I.1. Purpose and Organization of the Document ........ 11
I.2. Development Methodology ........ 11
I.3. Scope and Audience ........ 12

II. ICT ARCHITECTURE OF RENERGETIC SYSTEM ........ 14
II.1. Requirement Analysis SGAM ........ 14
II.2. RENergetic System Logical Overview ........ 16
II.3. RENergetic System Component View ........ 17
II.4. Deployment ........ 19
II.4.1. Requirements ........ 19
II.4.2. Infrastructure ........ 20
II.4.3. Strategy ........ 20
II.4.3.a. DevOps Architecture for RENergetic ........ 20
II.4.4. Tools ........ 22
II.5. Security Considerations ........ 23
II.5.1. Authentication ........ 23
II.5.2. Role-based Access ........ 24
II.5.3. API Security ........ 24

III. OPTIMIZATION CONCEPT ........ 26
III.1. Multi-vector Optimizer ........ 27
III.2. Domain-specific Optimizer ........ 28

IV. RENERGETIC FUNCTIONALITIES ........ 30
IV.1. Common Information Model ........ 30
IV.1.1. Relational Data ........ 30
IV.1.2. Time-series Data ........ 31
IV.2. Forecasting ........ 31
IV.2.1. AI Forecasting Vision ........ 31
IV.2.2. User Stories ........ 34
IV.2.3. Algorithmic View ........ 35
IV.3. Demand Response for Electric Vehicle Charging ........ 39
IV.3.1. Vision ........ 39
IV.3.1.a. Context ........ 39
IV.3.1.b. Functionality ........ 39
IV.3.1.c. Architecture ........ 40
IV.3.1.d. Stakeholders & Interactions with Smart Charging System ........ 40
IV.3.1.d.1. End User ........ 40
IV.3.1.d.2. Energy Island Operator ........ 42

IV.3.1.d.3. External Stakeholders ... 42
IV.3.2. Algorithmic View ... 43
IV.3.2.a. Introduction ... 43
IV.3.2.b. Markov Decision Process ... 44
IV.3.3. Next Steps ... 45
**IV.4. Demand Response for Heating Domain ... 46**
IV.4.1. Vision ... 46
IV.4.1.a. Related User Stories: ... 46
IV.4.2. Algorithmic View ... 48
**IV.5. Heat Supply Optimization ... 50**
IV.5.1. Vision ... 51
IV.5.1.a. Example: Importance of the Control of Heat Sources in the Ghent Pilot (CEIP) ... 51
IV.5.1.b. Example: Manual Heat Source Change in Poznan University of Technology (PUT) ... 53
IV.5.2. Algorithmic View ... 53
**IV.6. Local Waste Heat Optimization ... 55**
IV.6.1. Vision ... 55
IV.6.1.a. Example: Waste Heat within Poznań Warta Campus ... 55
IV.6.1.b. Related User Stories ... 57
IV.6.2. Algorithmic View ... 58
**IV.7. Interactive Platform ... 58**
IV.7.1. Vision ... 58
IV.7.2. User Stories and Mock-ups ... 60
**V. REFERENCES AND INTERNET LINKS ... 62**
**VI. APPENDIX ... 65**
**VI.1. CIM Relational Database ... 65**
**VI.2. Forecasting Models Registry ... 66**
**VI.3. EV Charging Experimental Results ... 67**
VI.3.1. Experimental Details ... 67
VI.3.2. Experimental Results ... 69
**VI.4. User Roles in RENergetic Platform ... 71**

# Table of Figures

Figure 1 - Organization of teams and roles in development process ... 11
Figure 2 - SGAM interoperability levels [2] ... 14
Figure 3 - Three dimensions of SGAM framework [3] ... 14
Figure 4 - Multi-energy SGAM [2] ... 15
Figure 5 - Model-Driven Architecture (MDA) engineering process [4] ... 15
Figure 6 - SGAM Level of Abstraction [4] ... 16

Figure 7 - Logical layers of RENergetic system ....................................................................17
Figure 8 - Microservice Architecture of RENergetic ICT System ..........................................18
Figure 9 - Components of RENergetic ICT System ...............................................................19
Figure 10 - Control panel with RENergetic system services at PSNC PaaS .........................20
Figure 11 - DevOps process in RENergetic ..........................................................................21
Figure 12 - Gitflow model [22]...............................................................................................22
Figure 13 - User creation in Keycloak....................................................................................24
Figure 14 – RENergetic login page.........................................................................................24
Figure 15 - Concept of Hierarchical Multi-Vector Optimization. ...........................................27
Figure 16 - Abstract overview on the data represented by the relational database of the CIM. ..........................................................................................................................30
Figure 17 - AI functions within Forecasting epic....................................................................32
Figure 18 - AI vision and Forecasting concept .......................................................................33
Figure 19 - MLOps schema reference ...................................................................................37
Figure 20 - Kubeflow software service (configuration point)...................................................37
Figure 21 - Python API to load and score a pre trained model for MW prediction demand ...38
Figure 22 - Python API to test model cross-generalisation performance................................38
Figure 23 - Smart Charging subsystem structure...................................................................40
Figure 24 - Manual Demand Response algorithm..................................................................41
Figure 25 - Automatic Demand Response algorithm .............................................................42
Figure 26 - An illustration of the EV charging coordination problem......................................43
Figure 27 - Heat demand response configurations ................................................................48
Figure 28 - Elements of heat demand response algorithm.....................................................50
Figure 29 - Overview of different heat sources and the expected demand of the district heating network of the Nieuwe Dokken in Ghent. ..............................................................52
Figure 30 - Expected share of the different heat sources at full development of the district (1250 inhabitant equivalent in 2027) and expected trend during the coming years. ...................52
Figure 31 - Overview of the activation of the heat pump and availability of other heat sources in the district heating network of DuCoop at the Nieuwe Dokken in Ghent (pilot CEIP). .......53
Figure 32 - A fragment of the synoptic view from the heat pump room. ...............................54
Figure 33 - SGAM for heat supply optimization .....................................................................54
Figure 34 - Heat connections within Warta campus in operation (blue, red, purple) and planned (green) ...............................................................................................................56
Figure 35 - Data center waste heat potential related to campus heating demand .................56
Figure 36: Mock-up for heat demand response with inhabitants ..........................................61
Figure 37: Mock-up for heat demand response Community Incentive .................................61
Figure 38 - Database schema ...............................................................................................65
Figure 40 - Train and test data selected in different validation methods used in experiments ..........................................................................................................................68
Figure 41 - Normalized load for RL based control policies trained with different state-action representations.................................................................................................................69

Figure 42 - Training time for RL based control policies trained with different state-action representations....70

Figure 43 - Average normalized load per iteration. Linear cost functions with $E$ = 1, i.e., optimum solution of one preceding day. ....71

Figure 44 - Normalized load for RL based control policies with linear cost functions that use $E \in \{1, 5, 10\}$. (Wilcoxon test p-values: statistically significant different for p-value<0.05)....71

# List of Tables

Table 1 - Exemplary user stories across pilots before Jira....34

Table 2 - Exemplary user stories in Jira....35

Table 3 - Example of long format data for AI algorithm consumption....36

Table 4 - Example: Input Variables for each building....49

Table 5 - Example: final score for each building ....50

Table 6 - Model registry for RENergetic Forecasting, Anomaly Detection, Root Cause Analysis, Sensitivity and Precision Energy (prioritized API models are orange coloured). Status as of 1 April 2022....66

Table 7 - Experience sets (F) generated for Experiment 1 (observability) to evaluate the effect of information provided by state-action representation on the performance of learned RL control policy. ....68

Table 8 - Experience sets (F) generated for Experiment 2 (credit assignment) to evaluate the effect of different cost functions on the performance of learned RL control policy.....69

Table 9 - User Roles in RENergetic Platform....72

# Table of Acronyms and Definitions

| Acronym | Definition |
|---|---|
| **AI** | Artificial Intelligence |
| **ANN** | Artificial Neural Network |
| **API** | Application Programming Interface |
| **BMS** | Building Management System |
| **CIM** | Common Information Model |
| **COP** | Coefficient of Performance |
| **DL** | Deep Learning |
| **EMS** | Energy Management System |
| **EV** | Electric Vehicle |
| **FOG** | Forecasting Group |
| **ICT** | Information and Communications Technology |
| **IEC** | International Electrotechnical Commission |
| **MDA** | Model-Driven Architecture |

| | |
|---|---|
| **MDP** | Markov Decision Process |
| **ML** | Machine Learning |
| **MLOps** | Machine Learning Operations |
| **MPC** | Model Predictive Control |
| **PaaS** | Platform-as-a-service |
| **PLC** | Programmable Logic Controller |
| **PSNC** | Poznan Supercomputing and Networking Center |
| **PUT** | Poznan University of Technology |
| **PV** | Photovoltaic |
| **RL** | Reinforcement Learning |
| **SGAM** | Smart Grid Architecture Model |
| **SoC** | State of Charge |
| **V2G** | Vehicle-to-grid |
| **WP3** | Work Package 3 |

# I. INTRODUCTION

## I.1. Purpose and Organization of the Document

This document is the first deliverable of work package 3 (WP3) and describes the interim version of the ICT RENergetic system that is being developed in the project. The requirements and overall vision of the system and its functionalities are described, as well as technical details about software architecture, data modelling, development and deployment strategies.

The document is structured as follows: The first chapter introduces the goals of the deliverable and present the development methodology taken in WP3. The second chapter contains information about technical details of RENergetic ICT system. It describes the approach for requirement analysis carried out by modelling system in SGAM cube in task 3.1. The approach also supports replicability of the solution solved by task 3.8. Further, this section contains the description of the output of task 3.2 - the ICT architecture and deployment tools and strategies. The third chapter is dedicated to the optimization approach in the RENergetic system. According to the tasks 3.3 and 3.4, the optimization is performed in a hierarchical manner and includes two layers, the multi-vector optimizer and domain-specific optimizers. The fourth chapter describes the proposed functionalities for the RENergetic system – a summary of results for tasks 3.3, 3.5 and 3.6. It provides current vision, mock-ups and ideas for algorithms and demand response programs, forecasting and other functionalities of the system.

## I.2. Development Methodology

The development process is adopting agile methodologies to create independent and self-organized teams. In order to handle the large number of features in the RENergetic system and their interdisciplinary nature, an organization inspired by Scrum methodology is used. This organization of teams is shown in Figure 1. The main goal of this architecture is to simplify the information exchange between different project partners involved in requirement creation, design, and implementation of the RENergetic system.

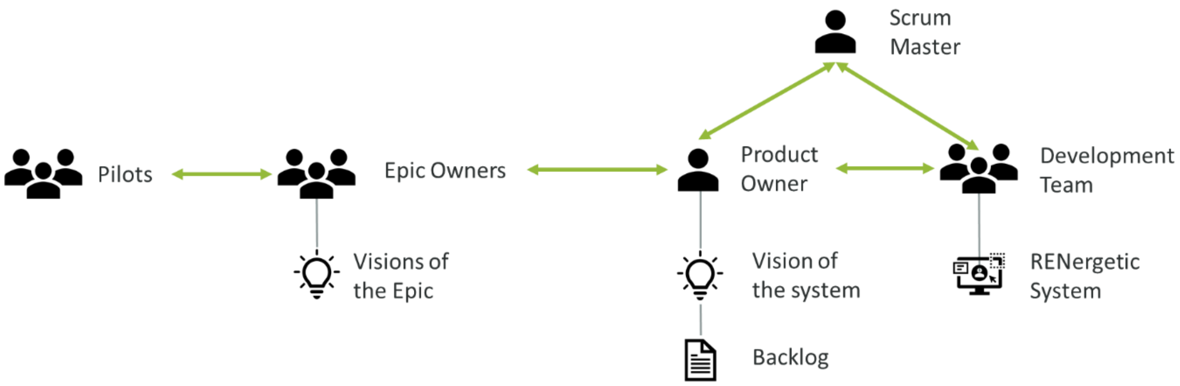



**Figure 1 - Organization of teams and roles in development process**

The functionality of RENergetic system is divided into several big parts called epics. Currently the following eight epics are considered:

- Heat demand response
- Heat supply optimization
- Local waste heat optimization
- Electric vehicle demand response
- Building electricity demand response
- Electricity supply optimization

- Interactive platform
- Forecasting

The last two epics are considered to be cross functional. The epic “Interactive platform” unifies all functionalities that should be accessible by users through graphical interfaces of web application. The epic “Forecasting” brings together functionality connected to the forecasting algorithms that are used in other epics, such as optimization, demand response and dashboards in the interactive platform.

As of April 2022, based on the needs and capabilities of the pilot sites, the project consortium decided that the development should start with the following epics:

- Heat demand response
- Heat supply optimization
- Electric vehicle demand response
- Interactive platform
- Forecasting

For each epic one responsible person, called epic owner, is assigned. This role is similar to product owner in standard Scrum, although the responsibilities of epic owner are limited only to the specific epic. This person collects requirements from all the partners in the project related to the functionality of the epic. In that way, every epic owner keeps the vision of a specific part of the system. Based on this vision, the user stories are generated, which then are transferred to the product owner.

The product owner in this organization is responsible for the global vision of the system. The product owner manages the backlog, i.e., the collection of user stories from all epics. The development team together with the product owner review these user stories. If the product owner and development team require additional information, a new communication round between epic owners and other partners is organized.

The development team is interdisciplinary and consists of members responsible for the implementation of different technical modules in the system. That is to say, it is not limited to the software developers who write code. Designers of forecasting and optimization algorithms as well as data modelling experts could be the part of this team too. Work inside the development team is organized based on the Scrum methodology, although some workflows are adapted to the specifics of the project. For instance, the daily scrum is transformed into two weekly meetings. Scrum master is a separate role that is responsible to ensures the adherence to the principles of Scrum methodology during every activity inside of the dev team.

This organization simplifies the task of transforming high level user stories into more technical user stories that could be implemented by the development team. This organization is also flexible – the epic owners could choose the way that they communicate with the partners. Another advantage is that many actions can be performed in parallel.

The functionalities described in the third chapter of this deliverable (with the exception of common information model) were considered as separate epics. These functionalities were chosen as a main priority for the development by the project consortium. The deliverable describes current progress made in each epic based on the respective user stories and their implementations by the development team.

## I.3. Scope and Audience

This deliverable provides a comprehensive description of the interim version of the ICT tools developed for energy island communities in WP3. The authors describe the development process, conceptual architecture view as well as concrete algorithmic views on different subparts of the ICT solution. It refers to scientific publications wherever possible. Since the document presents the work done in WP3 in the first project period, presented content and algorithm may change and improve in the future.

This deliverable offers a comprehensive view on the work done in WP3 so far. The document is relevant for project members involved in the pilot locations as it summarizes the main concerns and ideas from the ICT point of view.

# II. ICT Architecture of RENergetic System

## II.1. Requirement Analysis SGAM

One of the main objectives of RENergetic is to design and develop a reference system architecture for energy islands. Standardization and interoperability of the systems, its sub-systems, components and business processes are key factors to enable replicability of the solutions. The system architectures developed in the project should act as a blueprint, which can be applied in the project pilots, but also for external scenarios.

Several architecture models and methodologies that seemed suitable for application in RENergetic, were analysed. Subsequently, we found that the Smart Grid Architecture Model (SGAM) is on the way to standardization by the International Electrotechnical Commission (IEC) [1] and the most mature approach. In addition, it fits very well to our energy focus, while other smart city methodologies instead consider a very broad set of verticals like health and other topics.

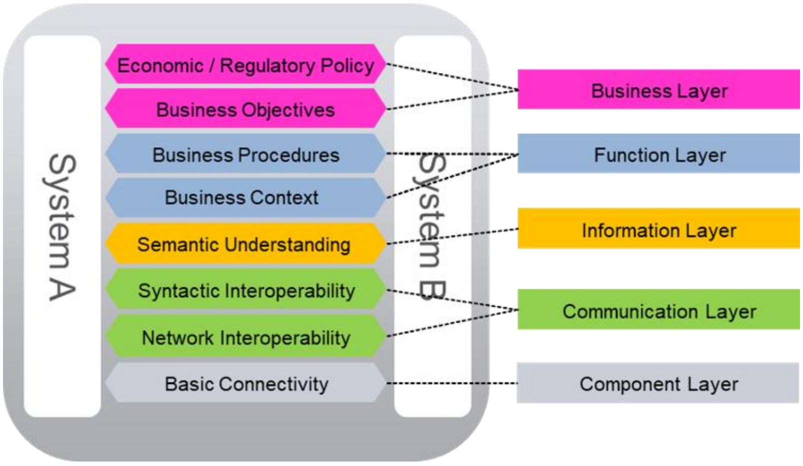



**Figure 2 - SGAM interoperability levels [2]**

The SGAM combines different perspectives on a system into the five layers business, function, information, communication and component layers. This structure enables separation of concerns. The other two dimensions, “zones” and “domains” logically separate the energy system.

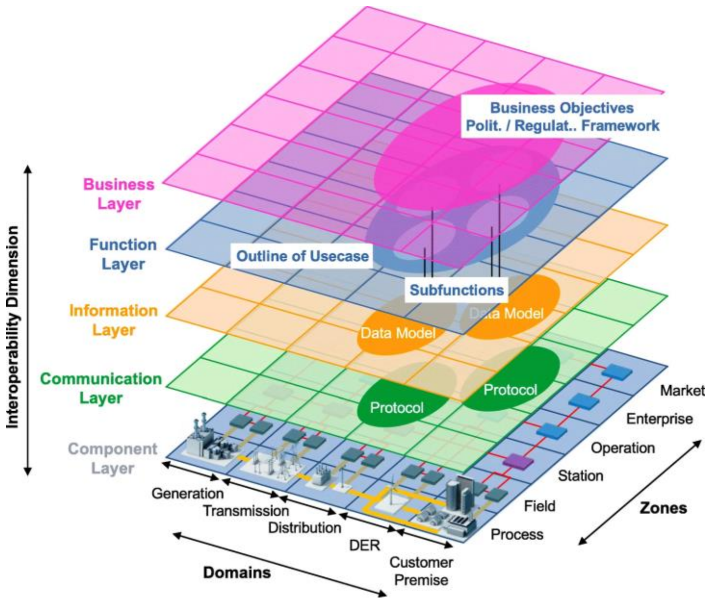



**Figure 3 - Three dimensions of SGAM framework [3]**

The SGAM is focused on the electricity sector, but within RENergetic, we also consider other energy domains, especially heat. Therefore, the multi-energy SGAM to be developed by the standardization approach (IEC SRD 63200) is chosen.

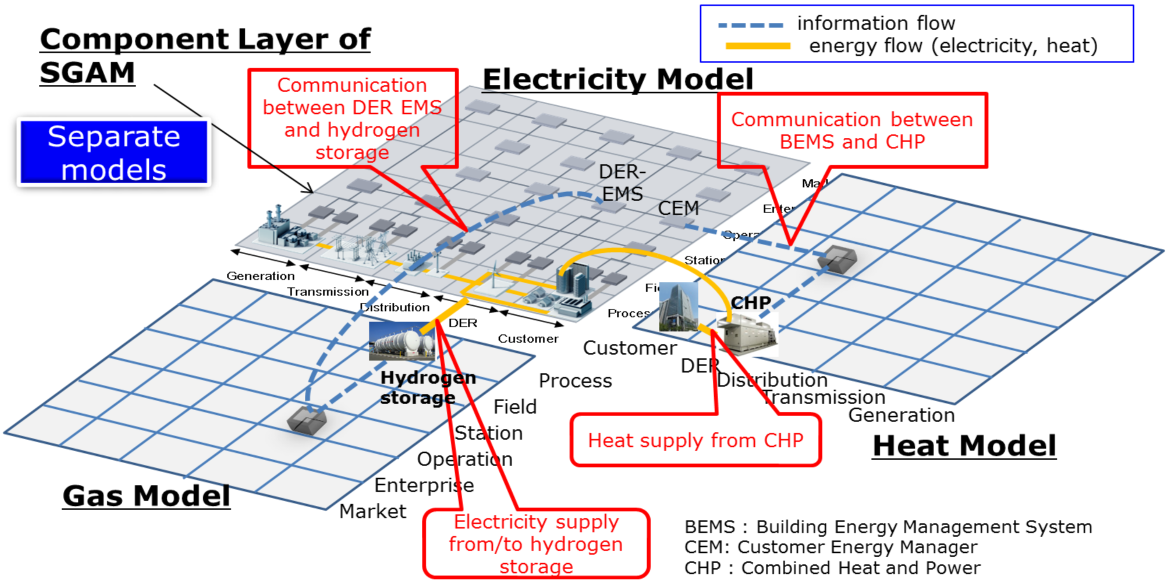



**Figure 4 - Multi-energy SGAM [2]**

Within the project, we follow an engineering process based on the SGAM, which will be described in the following. We start top-down by analysing business layer and stakeholders and bottom-up by investigating the technical situation in the pilot sites.

After this initial phase of system analysis, the system architecture is developed, followed by design and implementation of the system.

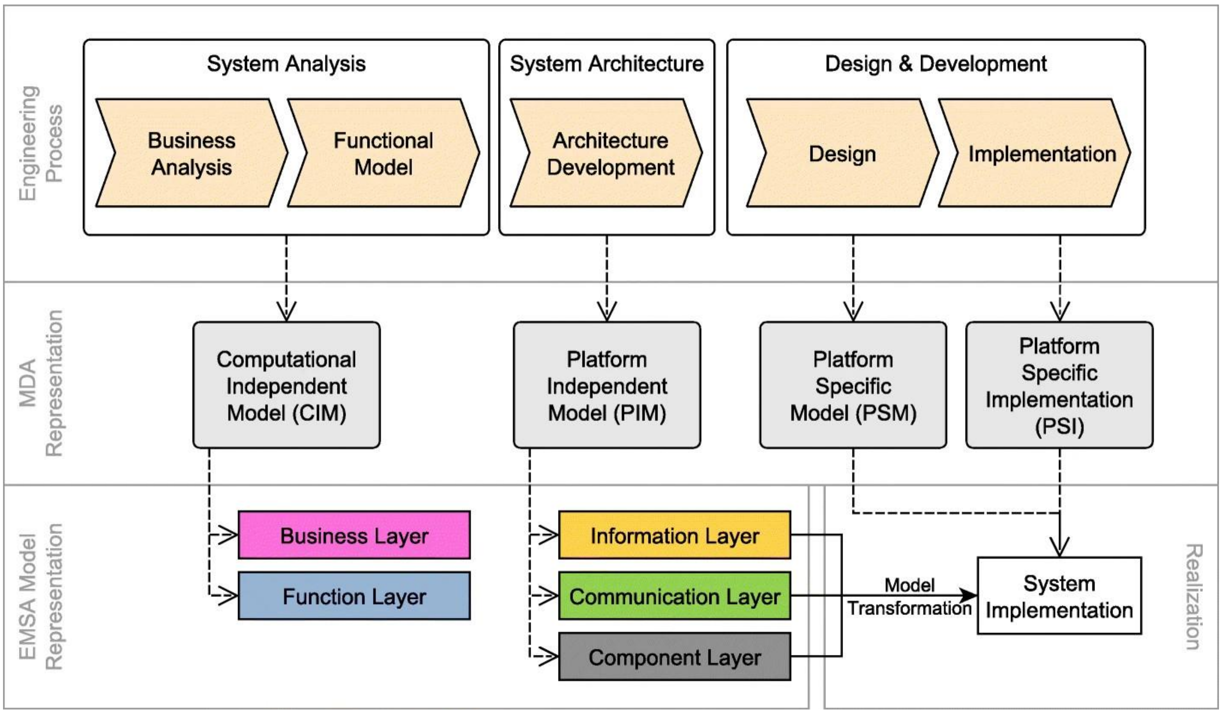



**Figure 5 - Model-Driven Architecture (MDA) engineering process [4]**

One of the main issues, when modelling the architecture of such complex system, is the level of abstraction of the model. In RENergetic, a certain level of abstraction is chosen for the different layers as shown in Figure 6.

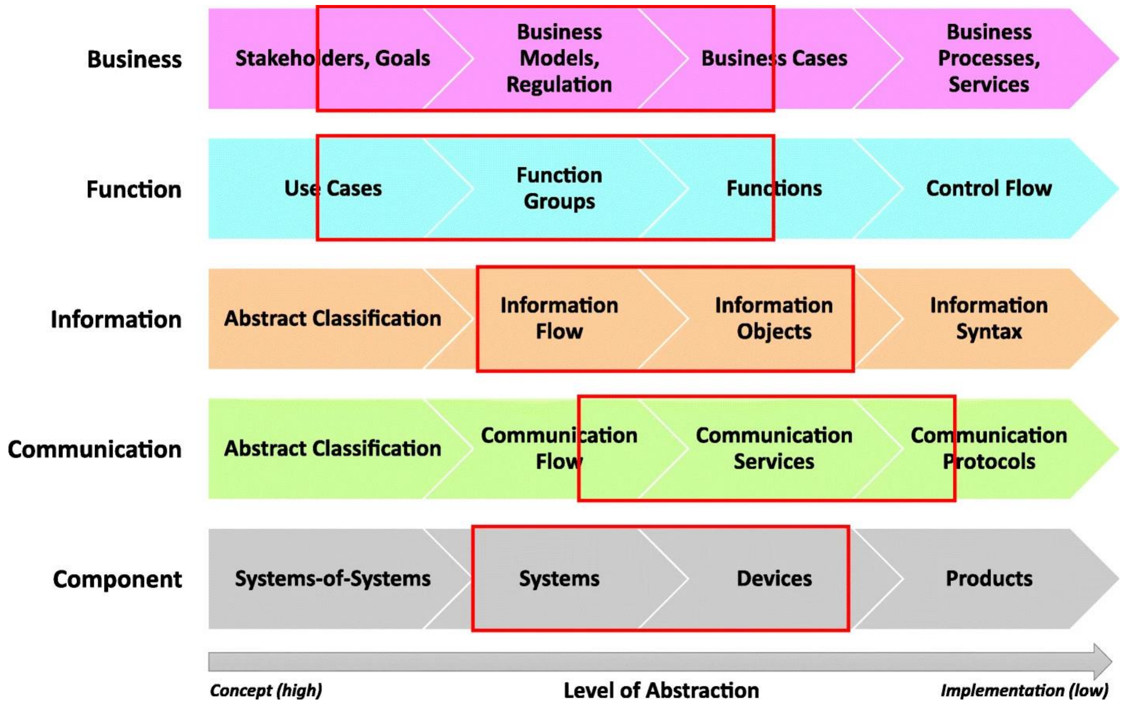



**Figure 6 - SGAM Level of Abstraction [4]**

In order to do so, a human-centric approach of user stories has been adopted that allows to communicate the needs of users in clear and simple phrases to get an understanding of all required features. These user stories are structured into overarching epics to form the basis for the requirements engineering for the RENergetic solutions. These user epics cover the electric as well as the heat domain and allow the RENergetic solutions to interact with a variety of infrastructure at the pilot and future replication sites. Each of these user epics will be mapped to a separate multi-energy SGAM cube.

The goal of this process is to identify the essential items for each user epic implementation in each of the SGAM layers as well as all necessary data interfaces between the infrastructure at the pilot sites and the RENergetic system.

This methodology also forms the basis for the replication package, which is presented in deliverable D8.1 and D8.2. There behavioural models, business models, and legal constraints will be added into the business layer of each epic's SGAM cube as well.

## II.2. RENergetic System Logical Overview

The main purpose of RENergetic ICT solution is using the data coming from the energy island systems to provide visualization of this data to users as well as to forecast energy generation and consumption. Additionally, this energy data together with forecasts allow performing optimization of the energy sources usage and demand response programs. These methods allow achieving the energy island goals, like increasing the renewable energy share and maximizing energy autarky.

That means that the system should be capable of ingesting various types of data coming from different sources in the energy island. This is not trivial task, since the architectures of different energy islands are very heterogeneous. For this reason, the project consortium decides that pilot sites are responsible for creation of a docking module that will be used to connect their respective diverse pilot system into the RENergetic system using an unified interface. Figure 7 shows the layers of RENergetic system that connect pilot systems with the end users.

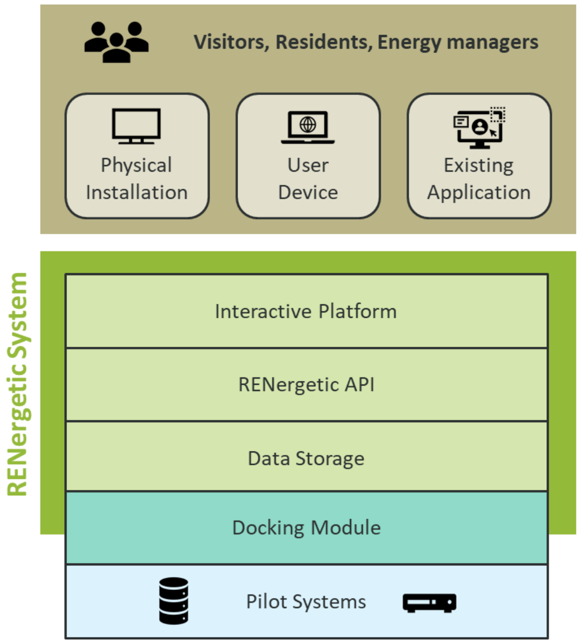



**Figure 7 - Logical layers of RENergetic system**

Users could interact with the system through the RENergetic interactive platform, a web application provided by RENergetic system. Alternatively, it is possible to integrate the RENergetic system into already existing applications in the energy islands through its Application Programming Interface (API). Furthermore, API can be used to communicate with the energy island systems through docking module, in order to transmit control signal to installed equipment.

Different types of users can use the RENergetic platform. Visitors of the energy island can receive information about energy island status, e.g., utilization of renewables, using their personal devices, displays installed in building halls or other physical installations. More information with greater detail is available to residents and associates of the energy island. This includes recommendations for changing the parameters of appliances for participation in heat demand response programs, dashboards with historic and forecasted energy data. Energy island managers receive additional information about the different energy vectors including, for example, the forecasted energy consumption and the recommended parameters and settings for energy sources.

## II.3. RENergetic System Component View

To implement identified logical layers of the RENergetic system, it is proposed to utilize microservice architecture depicted on the Figure 8. Each service is a separate software module that perform specific functionality. A service can interact with other services, the data storage and the interactive platform. The API and Access Management service is responsible for orchestrating operation of all other services. It also provides an API for communicating with external third-party systems.

A microservice architecture is proposed to implement the identified logical layers of the RENergetic system. A summary of this architecture is depicted in Figure 8. A microservice is a separate software module built to perform a specific functionality. These microservices can interact with other services, such as the data storage and the interactive platform. A special role is assigned to the “API and Access Management” microservice as it is also responsible for orchestrating the operation of all other services. Furthermore, it provides an API for communicating with external third-party systems. The data is stored in two separate databases. A time series database is used for measurement data of various sensors in the system. Yet, a relational database is better suited to model the connections of different assets at the pilot. Other functionality is related to multi-vector optimization, domain-specific optimization, forecasting or various forms of demand response, which are elaborated on in sections III and IV.

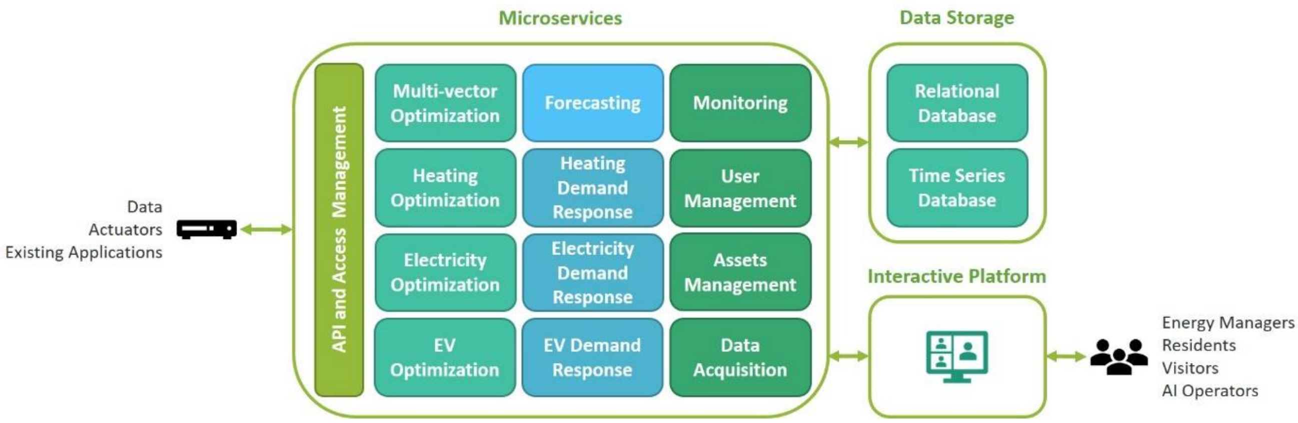



**Figure 8 - Microservice Architecture of RENergetic ICT System**

Such modular architecture enables flexible configuration of the system. This is important because not all energy islands have the necessary data or systems required for the operation of all services. In case some functionality is not needed or cannot be realised, the corresponding service can be excluded from an installation. A service-oriented architecture also simplifies extension of the system in the future. For instance, if optimization in an additional domain is required, the new service for that can be developed and easily integrated into the existing RENergetic system.

The software packages and frameworks that are used for the implementation of the RENergetic system are shown in Figure 9.

The data in the RENergetic platform is ingested by pilots' external systems – either with some open API or preconfigured scripts. Those scripts will be managed by Apache NiFi [7] (or any similar task scheduler/orchestrator). Data in the RENergetic system is stored in the InfluxDB [5] (time series database) and PostgreSQL [6]

The output from the data ingestion process will be stored into the InfluxDB database, initially in a raw way.

Apache NiFi will run Apache Spark [8] processes, which is responsible for data post–processing and normalization in InfluxDB (filling missing values, adjusting granularity, unifying units, aggregating measurements, etc.), time series inference and predictions pre generated data models.

Machine learning workflows will be deployed and managed in Kubeflow pipelines [9]. These pipelines will be scheduled to run forecasting and anomaly detection algorithms. Generated by these algorithms data is stored in the data storage (time series and relational database) that can be accessed from the Kubeflow using API.

Each microservice can be implemented using different programming languages and frameworks. Most of the microservices are implemented as Spring Boot [10] applications that provide APIs. The Spring Boot APIs are managed by WSO2 [11], which helps to secure the APIs.

RENergetic interactive platform consist of two software modules: the web application is based on Vue.js [12] and Grafana [13]. The Vue.js application is the main graphical user interface for RENergetic users. It provides access to different functionalities depending on the roles (Guest, User, Energy Manager, and Administrator) assigned to the user:

- Public dashboards
- Private dashboards
- Energy Island management (user, asset, measurement and dashboard management)

Grafana  is a tool that provides an easy way to create different types of graphs organized in dashboards that are accessible via the interactive platform. The dashboards implemented in Grafana are intended only for users with the Energy Manager role.

Keycloak [14] is used as an identity and access management solution in RENergetic. There the user roles are configured.

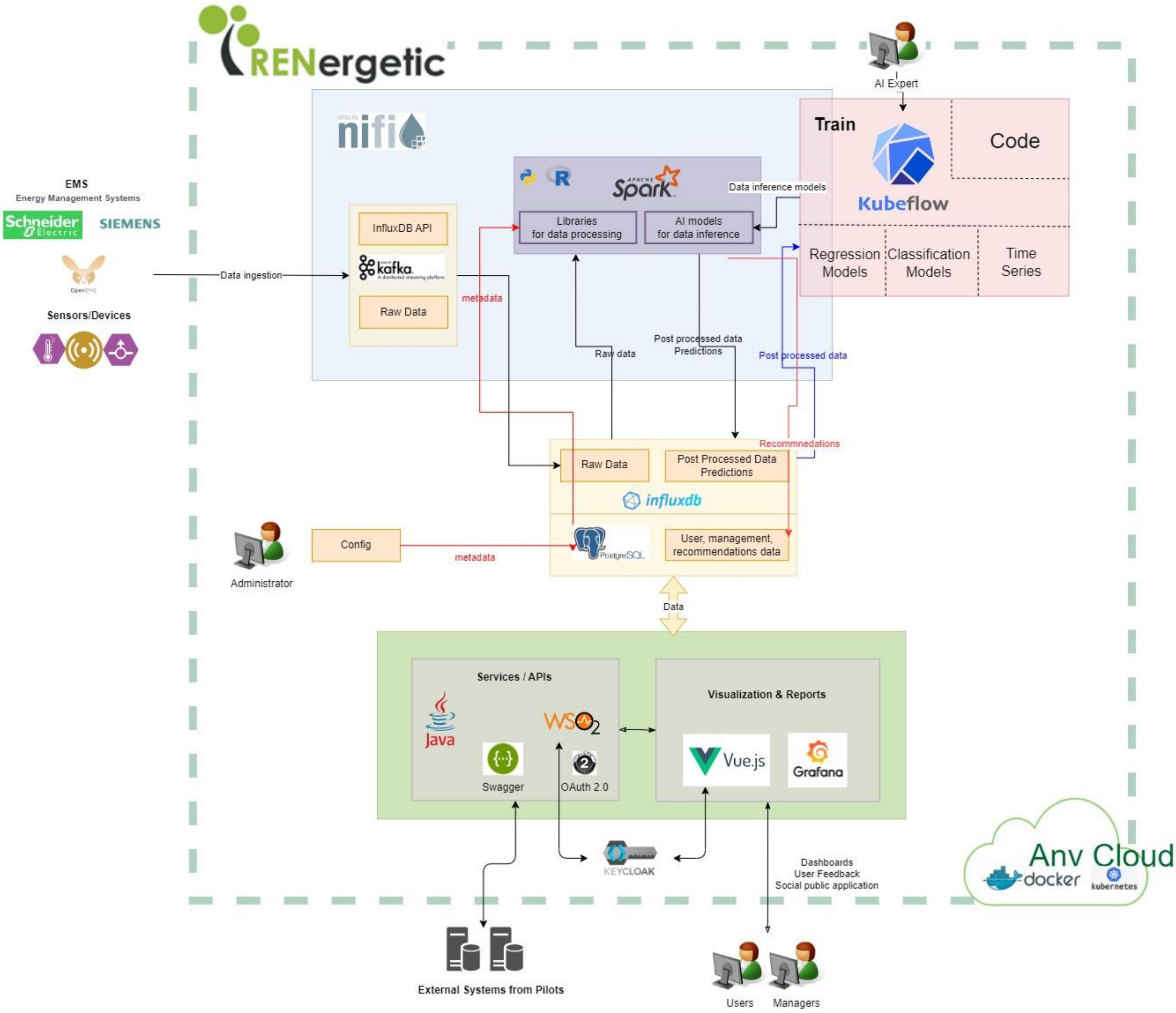



**Figure 9 - Components of RENergetic ICT System**

# II.4. Deployment

## II.4.1. Requirements

The RENergetic system is designed as a microservice architecture. The uncoupled software components, and services make the system modular and scalable. Any infrastructure where RENergetic system could be deployed should meet the following requirements:

- Support a microservice architecture.
- Support running Docker Containers [15] managed by Kubernetes [16] - near de facto standard nowadays.

## II.4.2. Infrastructure

The components of the RENergetic architecture are deployed in the Platform-as-a-service (PaaS) infrastructure provided by one of the partners in RENergetic - Poznan Supercomputing and Networking Center (PSNC).

This infrastructure is based in OpenShift [17], a containerization software product by RedHat. OpenShift makes use of OKD [18] version 3.11 – community distribution of Kubernetes. Community OpenShift platform using Kubernetes 1.11.

This kind of architecture lets us assign and scale the needed resources for each component in an independent way.

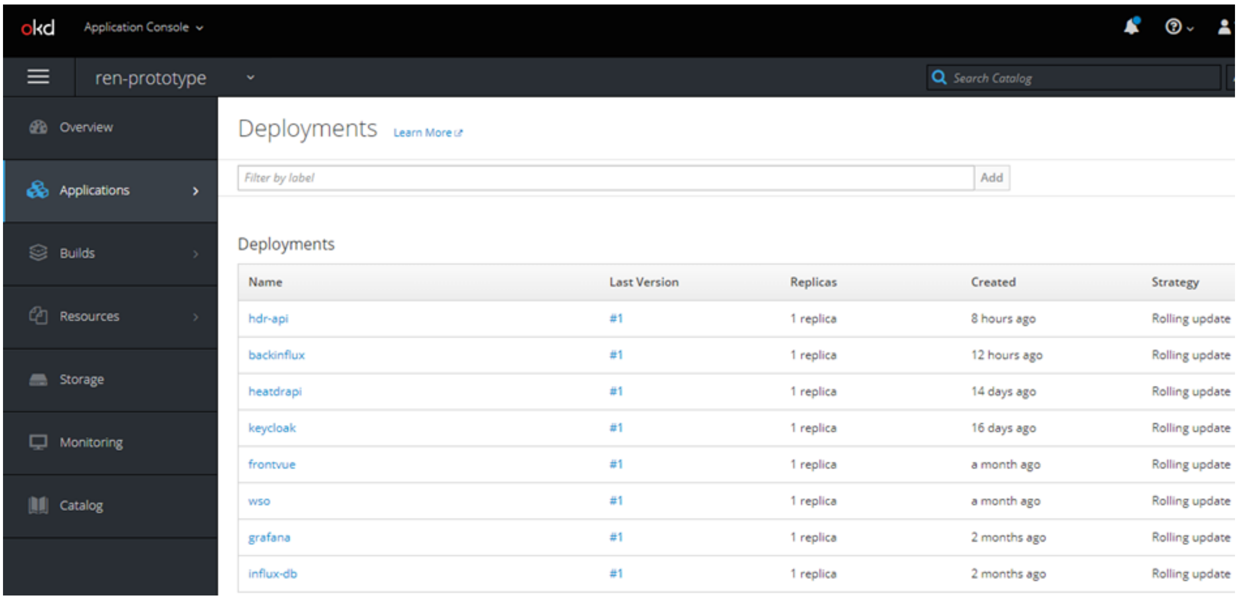



**Figure 10 - Control panel with RENergetic system services at PSNC PaaS**

PSNC provides two distinct PaaS clusters that will be used for development and production versions of the RENergetic system:

## II.4.3. Strategy

OKD offers mechanism to deploy software from the console or making use of the command-line interface. Development team have created a script that provides an easy way to deploy every component. That allow developers to redeploy the changes in the Java API backend and frontend.

### II.4.3.a. DevOps Architecture for RENergetic

**General Solution Definition**

The proposed solution is based on agile methodologies and aims to achieve a cycle of continuous integration and delivery of applications within RENergetic. In general terms, the life cycle of the applications would be as follows:

1. The cycle of integration and continuous delivery will begin when a developer with his code already implemented executes a commit.
2. At that moment, the code is committed to the version manager (Bitbucket [18]) and the work paths described in Jenkins [19] will start to be executed.
3. This will download the dependencies described in the Maven pipeline [20] and automatically run the established tests, whose evaluation determines the next step.

4. In this way, once the tests have passed, the build or artifact construction will be executed.
5. Once the artifact is built, the deployment or delivery phase of these artifacts begins. In the case of RENergetic, this delivery will be automated to achieve the continuous integration and delivery cycle.
6. Once the artifact is built, it will be stored in an artifact repository ready for use.

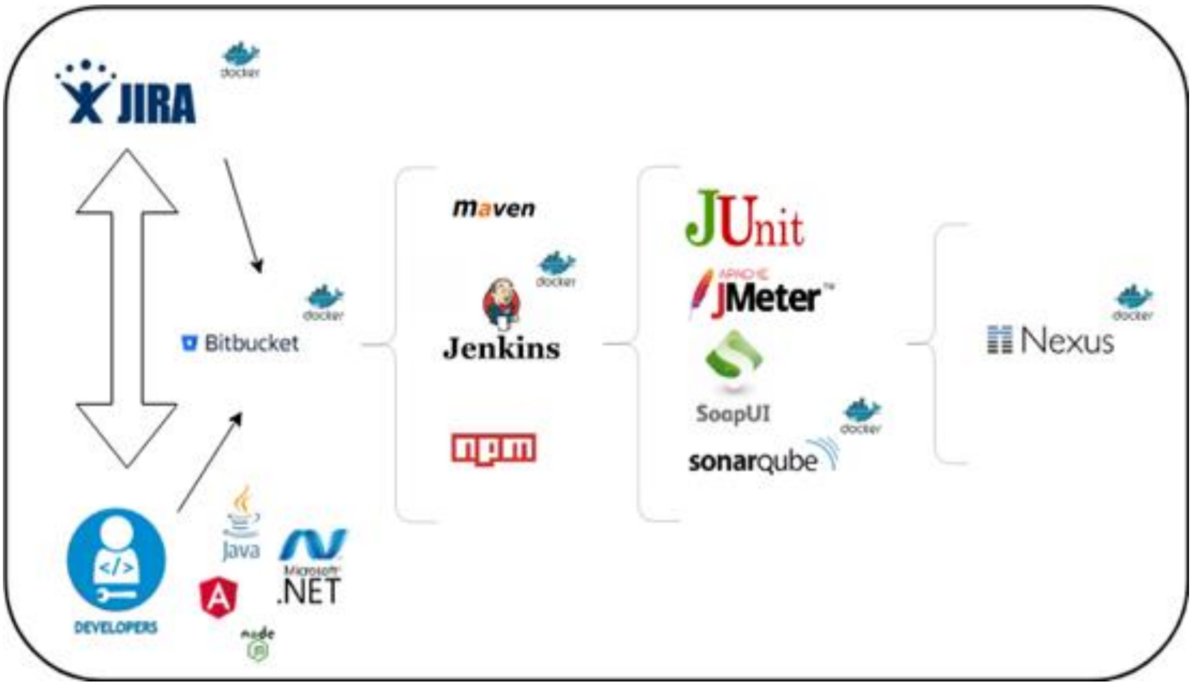



**Figure 11 - DevOps process in RENergetic**

The fundamental principles of DevOps are as follows:

- It allows to go faster in software development
- It shortens feedback loops
- It allows experimentation and continuous improvement
- It provides cultural benefits (more productive and efficient teams, and happier customers)
- It delivers business and customer value on an ongoing updates

Undoubtedly, some practices have clearly risen as critical to the realization of the fundamental principles of DevOps:

- Agile Software
- Continuous Integration (CI)
- Continuous Delivery (CD)
- Proactive monitoring
- Better communication and collaboration

Therefore, DevOps touches almost every aspect of IT management: People, practices, and automation.

**<u>Code Repositories and Workflows</u>**

Version control is a system that records changes made to a file or set of files over time so that specific versions can be retrieved later. Code repositories allow developers to work on the code without directly affecting the current version of the microservice. Once a developer has finished working on some piece of code, they commit it and then merge or rebase it to the main branch of the repository they are working on, thus updating the microservice. Workflows will be divided into branches, so that different developments can be parallelized and consolidated automatically.

The workflow followed in the Gitflow model is as follows:

- By default, you have two branches: *master* and *develop*. With *develop* being the branch where everything happens and *master* being the code version deployed in production.
- From *develop* branch, new feature branches are created for the implementation of new features and bug fixes that are not in *master* branch. At any time, it should be possible to switch from *develop* to *master* branch, which means that the new features should not be developed in *develop* branch, but in the respective feature branches.
- To go from *develop* to *master* branch one must create an intermediate release. This branch is used as an intermediate point to fix bugs before moving to production and is versioned Major or Minor.
- There can be several relay versions maintained at the same time.
- If a critical bug is discovered in *master* branch, hotfix branches are used to fix production.

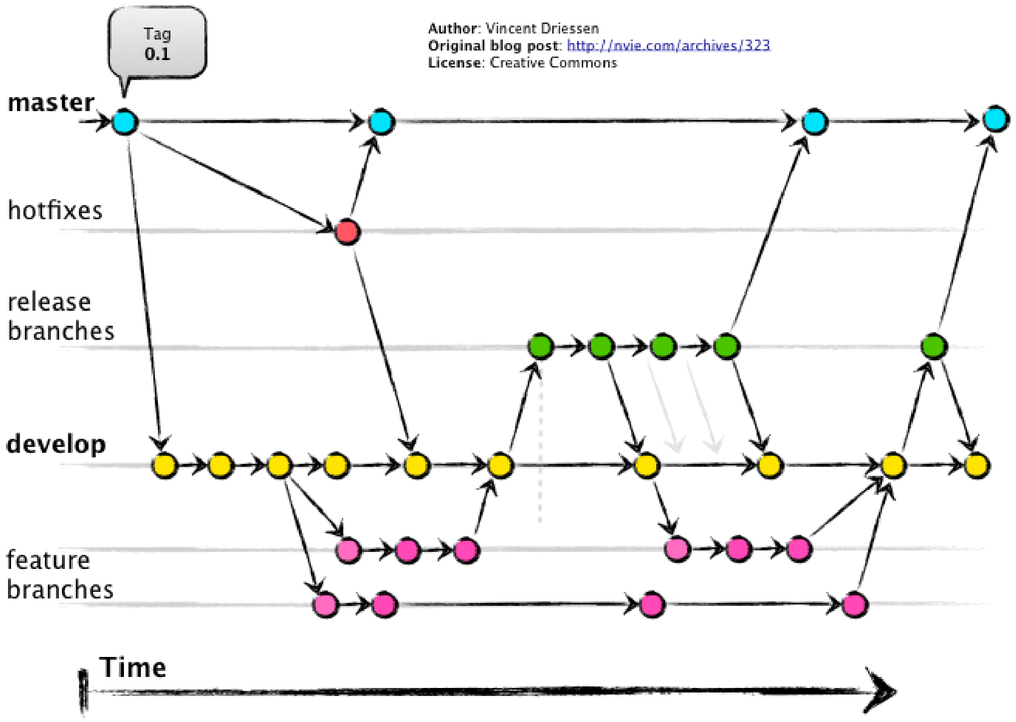



**Figure 12 - Gitflow model [22]**

## II.4.4. Tools

To organize the project from the development perspective we are making use of different tools:

**<u>Jira</u>**

Used to hold the backlog of user stories and to organize and follow up the Agile methodology applied in the project [23]

- Backlog
- Create sprints
- Lifecycle of the tasks
- Reporting

**Confluence**

The site where content related to the development tasks is created, being a reference place for the Dev Team Members [24]

**Bitbucket**

Code repository based on Git that lets us to have a centralized place to:

- Share our developments between all the Dev Team Members
- Versioning the software
- Create branches to organize the collaborative work.
- Apply CI/CD mechanisms to automate tasks.

**Jenkins**

In today's DevOps world, continuous delivery and deployment is critical to delivering high-quality software products faster than ever. It helps developers build and test software continuously. Essentially, Jenkins integrates development lifecycle processes of all types, including build, document, test, package, stage, deploy, static analysis, and more.

Jenkins works as follows:

- A developer commits the code to the source code.
- The Jenkins server checks the repository at regular intervals for changes.
- Jenkins detects changes that have occurred in the source code.
- Jenkins will make those changes and start preparing a new version. If the build is correct, then Jenkins implements the built-in.

You can configure the pipeline (the script to execute) to create the build with several steps: Prepare, test (unit tests and integration tests), package, publish, deploy. If the built-in is successful, Jenkins sends the created artifacts to the repository.

**SonarQube**

SonarQube [25] is "an automatic code review tool to detect bugs, vulnerabilities, and code smells in your code". It allows sharing rules between all developers and set goals regarding the development.

## II.5. Security Considerations

RENergetic system is going to implement security in different places:

- Authentication
- Role-based access
- API security

### II.5.1. Authentication

The authentication to be able to access Kubeflow and the RENergetic system will be managed by Keycloak [14], an open-source identity and access management tool. A **realm** in Keycloak is the equivalent of a tenant. It allows creating isolated groups of applications and users. Users can be created directly in Keycloak as it is shown on Figure 13. But it could be possible to enable the self-registration.

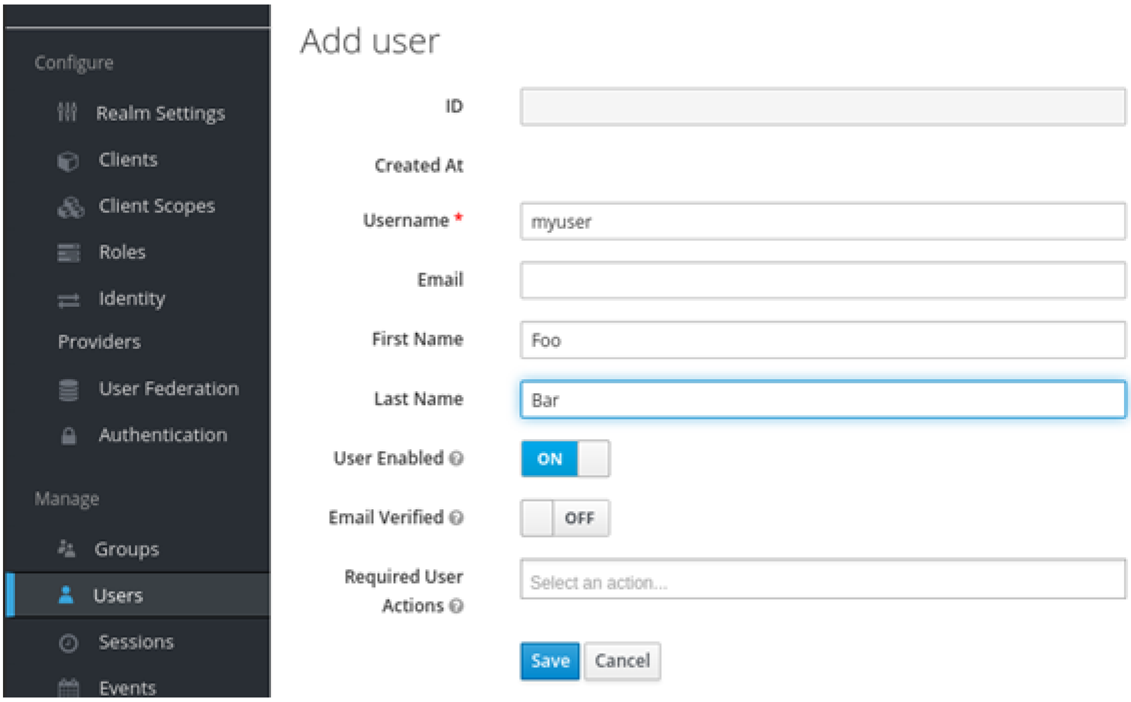



**Figure 13 - User creation in Keycloak**

At the end, RENergetic will implement its user management functionality and will interact with Keycloak through its own API. Moreover, Keycloak offers mechanisms to create a customized login page shown on Figure 14.

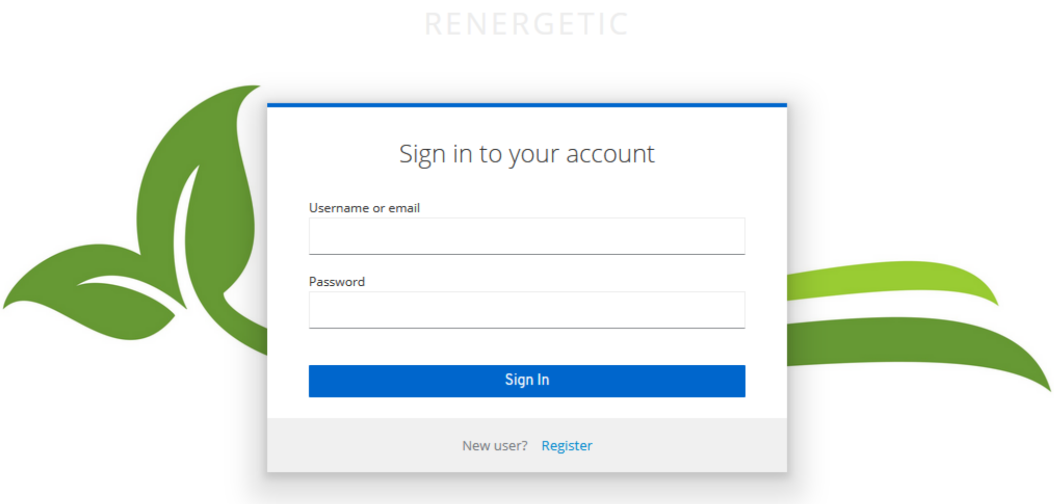



**Figure 14 – RENergetic login page**

## II.5.2. Role-based Access

Functionalities in the RENergetic system will be defined for different roles (Energy Manager, Resident, etc.). Then the users that will be created/register into the RENergetic system have to be assigned to the needed role(s) in order to be able to interact with the platform properly. Keycloak is used to configure the roles, manage their assignment, and retrieve them to apply restrictions through its API.

## II.5.3. API Security

Backend APIs are managed and exposed by the WSO2 API Manager. WSO2 makes it possible to integrate our APIs and manages them in order to keep them versioned and secured. It offers different securitization options. In the current development stage, the **Api Key** option is used that enables the access via a static token.

The next step will be to change the application-level security to **OAuth2**, making use of Keycloak as key manager of the dynamic tokens. WSO2 also allows configuring CORS policies, for example, to restrict the access from some IPs and the allowed HTTP methods.

# III. OPTIMIZATION CONCEPT

This chapter details the concept of optimization within the RENergetic solution. The main *objectives* (some of them are overlapping) are:

- **Maximize share of renewables**. Any energy generation asset (no matter if it is electricity or heat domain, or both), can provide its energy to a certain degree of renewability.
- **Maximize autarky**. Optimally supply the consumption from energy sources within the grid, either by shifting flexible consumption or by scheduling local generation.
- **Maximize self-consumption.** Optimally use the generated energy within the energy island by minimizing energy flow through external grid connections.
- (**Minimize CO2 emission.** CO2 emission is highly coupled with renewability of the energy share hence is already part of the first objective.)
- (**Minimize cost.** Cost minimization of energy generation is coupled with higher self-consumption values, autarky levels and reduced external connections.)

In order to reach the above-mentioned goals, the RENergetic system can control/suggest operation of different assets from different domains, including Electric Vehicle (EV) charging, battery energy storage system, Photovoltaic (PV) system, heat-pump operation, Heat demand response signals, electricity demand response signals, etc. However, with this set of different decision variables, a lot of limitations come along. Among others, there are technical *constraints* for different domains, such as voltage limitations in the power grid, resource availability limitations, such as waste-heat availability, storage constraints, such as storage capacities of heat and electrical energy, or controllability limitations arising from the controllable assets, such as EV charging control, heat-pump operation modes and combined heat and power plant operation plans. Additionally, involvement of end-users, e.g., via manual demand response schemas, requires behaviour modelling of the users to incorporate their reactions in the system. All these limitations must be considered by the constraints of the optimization problem.

Due to the requirement for replication of the solution, the optimization cannot model all domains and their constraints systematically. For example, it is not desired to have an exact power grid model with line impedance and topology, because this would limit the replicability of the model, due to high data and information requirement. Instead, *model-free* solutions or *highly abstract* representations of the actual energy island are desired, as long as they still capture the necessary energy island functionalities. Due to the interconnection of different energy domains (electricity, heat, mobility, etc.) the overall problem is highly constrained, mainly due to the interconnectors between the domains, e.g., heat pumps, and the huge amount of possible decision variables interactions.

In order to achieve *scalability* and keep the *global multi-vector* optimization as *abstract* as possible, we propose a *hierarchical concept* for the optimization, depicted in Figure 15. The global multi-vector optimizer will fix the energy flow at inter-connectors between domains, e.g., *charging stations*, *heat pumps* or *combined heat and power plants*. While *domain-specific* solvers will try to optimize the energy usage within their domain, respecting the interconnector decision by the multi-vector optimizer. Therefore, only aggregated generation and demand profiles, as well as flexibility in both (generation and demand) are required. In a second step, the domain-specific optimizer will take care of the internal domain-specific assets, e.g., which electric vehicle will charge at which time, what heat demand response signals are sent to whom, or where to place ancillary services in the power grid domain.

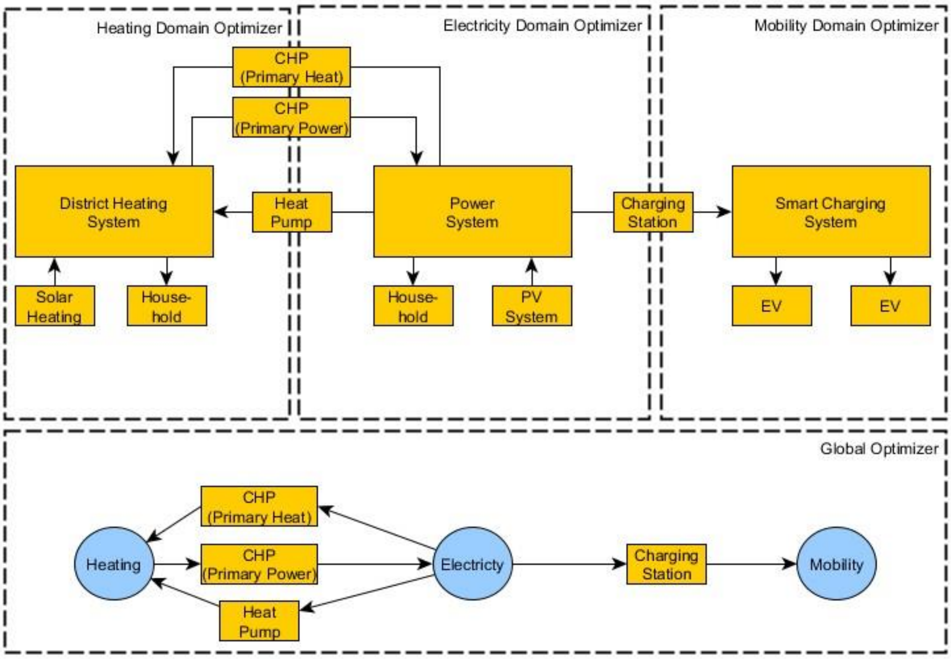



**Figure 15 - Concept of Hierarchical Multi-Vector Optimization. First, the global multi-vector optimizer fixes the domain interconnectors (CHP, heat pump, charging station). Second, the domain-specific optimizers control their local assets using domain-specific knowledge.**

The hierarchical concept performs the following three steps:

1. Retrieve aggregated fixed load and generation (non-controllable, which must be served, e.g., baseload in heat and electricity grid), as well as flexibility potential (flexible loads and generation, e.g., battery or heat storage, manual/automatic heat/electricity demand response potential).
2. Optimize inter-domain energy flow by fixing the operation of the inter-connectors (whether they are switched on/off or the operation mode if possible) and target values for aggregated flexible load and generation.
3. Domain-specific optimizer optimizes the energy usage within its domain, following the flexible aggregated profile (this is important, because some inter-connectors, e.g., heat-pumps, are operated heat-driven, hence higher heat demand will increase the electricity consumption, which may lead to sub-optimal energy island operation in the electricity domain). The domain-specific optimizer should base its decision on the same global objectives (e.g., when scheduling the generation units), otherwise there is again the risk of sub-optimal energy island operation. However, the domain-specific knowledge and single disaggregated assets (and potentially higher time resolution in the electricity domain) can be used to reach the goal.

From the RENergetic system architecture point of view (Figure 8), the functionalities related to the optimization are shared over multiple microservices. The multi-vector optimizer has a dedicated microservice in the architecture. Similarly, for each domain-specific optimizer a separate microservice is introduced.  To ensure correct operation of the algorithms these services require information from other parts of the system, such as forecasting or heat demand response. The output of the optimizers is stored in the data storage and can be shown on the interactive platform dashboards.

## III.1. Multi-vector Optimizer

As already outlined above, the multi-vector optimizer fixes the operation of the domain inter-connectors. Therefore, an abstract representation of the different domains and its connected assets is required. The desired *common information model*, which can represent the required information, is discussed in detail in Section IV.1. Most important to know are the domain-interconnectors, which according to the name, are part of at least two managed domains of the energy island. For example, a heat pump consumes electrical energy while producing thermal energy in form of heat or cooling (air conditioning system). Another example is a combined heat and power plant, where external energy carriers (gas, methane, coal, etc.) are used to produce both thermal and electrical energy at once. In this way, the total efficiency of the plant is improved compared to energy generation in one single domain.

In general, it may be enough to capture only the bigger inter-connector units, such that at least the biggest portion of the energy flow among domains is optimized. This can improve calculation speed and may be required in energy islands, where connection to all inter-connectors is possible. In addition, multiple inter-connectors that share the same domains and can operate in the same operation modes may be aggregated. This reduces the number of decision variables on the multi-vector optimization problem. The following provides a non-exclusive list of possible domain inter-connectors:

- Heat pump
- Combined heat and power plant
- Charging station
- Gas boiler (if connected to a gas distribution network, which is also managed by the energy island)

Note that the multi-vector optimizer operates on the greatest common time resolution. For instance, if heat is planned/operated in 3-hour slots and the electricity grid is planned/operated in 15 min, the multi-vector optimizer plans for 3-hour slots. Because heating infrastructure has higher inertia, a planned heat-pump operation profile with a given heat energy demand for the 3-hour block, may be fine-tuned be the electricity domain optimizer in a faster time scale. For example, the operation profile of the heat pump (force switch off/on) during the 3 hours can be optimized by the electricity domain optimizer as long as in total the same amount of energy is transferred between the electricity and heat domain.

The main optimisation strategies offered above are clearly approached from an (integer) linear programming perspective with objective functions and constraints modelling the specifications. Such approach, primary for this research could be accompanied by probabilistic modelling and stochastic methods derived from AI and centred on machine learning from the data outputs of all available domain optimizers. These developments will be reported in deliverable D3.2.

## III.2. Domain-specific Optimizer

The domain-specific optimizers have two main tasks: (1) provide information on the aggregated domain-specific fixed energy demand and generation, as well as their flexible counterparts. Additionally, they may specify constraints on flexibility, e.g., energy restrictions. (2) After the multi-vector optimizer has fixed the inter-connector operations, the domain-specific optimizer is in charge of optimizing the detailed operation of the single assets connected to the domain. For this task, domain-specific knowledge, e.g., requirement and placement of ancillary services for the power grid or temperature control of district heating systems can be utilized.

The most important thing to remember is that every domain should try to stick to the inter-connector operation profiles, e.g., if there is heat generation from the combined heat and power plant, the heat-domain optimizer should try to consume this heat by its own assets. Because the flexibility potential of each domain is already shared with the multi-vector optimizer in the first step, energy profiles at the inter-connectors are guaranteed to have a feasible solution in the domain as well. Note that if the profile cannot be achieved, e.g., there is deviation from the planned profile; this will result in less optimal energy island operation. In the worst case, if the deviation becomes too big, there is a need for re-planning by the multi-vector optimizer.

The exact nature of the domain-specific optimizer differs between domains and is specified in Sections IV.3. , IV.4. and IV.5. Note some domains require additional synchronization on a higher time resolution. For instance, EV charging and electrical domain need to synchronize their consumption/generation profiles in higher time resolution, which can be achieved by first fixing one domain (electric vehicle charging), and adjust the electrical domain on the consumption profile of the electric vehicle charging domain. This order may be determined by the availability of flexibility in the domains, e.g., if there are only a few electric vehicles, they should be optimized first. If there are plenty electric vehicle, the mobility domain offers more flexibility and may adapt to the requirements of the power grid.

# IV. RENERGETIC FUNCTIONALITIES

## IV.1. Common Information Model

A *Common Information Model* (CIM) defines the structure and the way to store the data of an energy island. With the CIM in place, all components of the RENergetic system have a common way to retrieve and store data, which also supports interoperability. In general, we distinguish between structural, e.g., structure of the energy community, users, etc., and time-series data, e.g., load and generation profiles and their forecasts. This data is stored in two different types of the databases: the structural data is stored in a relational database (PostgreSQL), while time-series data is stored in a time-series database (InfluxDB).

### IV.1.1. Relational Data

In general, the CIM must represent the different energy infrastructures of the energy island, e.g., the electricity grid, district heating or cooling networks. In addition to that, any asset that is connected to these infrastructures and, hence has an impact on the energy island, is required to be known. Among others, these assets contain any controllable and non-controllable loads/generations, the inter-connector (heat pump, combined heat and power plant, EV charging station) and external grid connections. Any connection of an asset to an infrastructure may be attached with a measurement, e.g., consumption and generation time series or static upper/lower limits. The constant limits are stored in the relational database, whereas for the time-series data and the relational database only stores a pointer to the time-series. Finally, users need to interact with the system. Therefore, the user must be modelled in the relational database, as well as their connection to some assets, e.g., the asset the user may control such as the energy consumption of his flat/building. Figure 16 summarizes the requirements of the relation database.

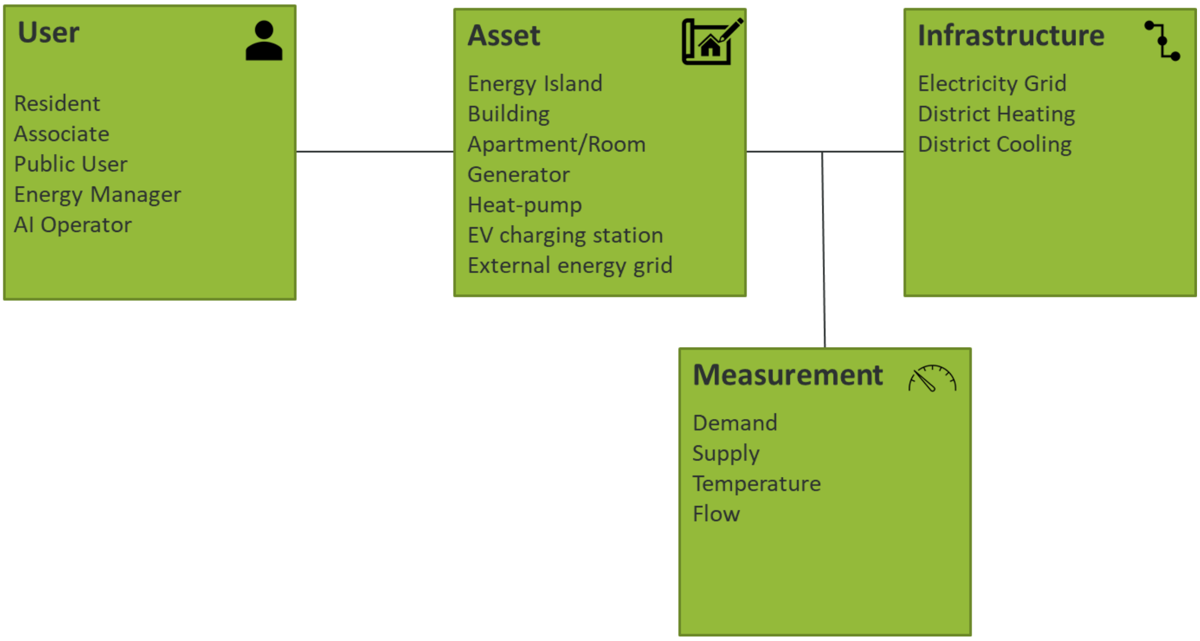



**Figure 16 - Abstract overview on the data represented by the relational database of the CIM.**

The exact database schema can be found as part of the code and is included in Appendix VI.1. The main information stored in the relational database is listed in the following:

- *measurement_type:* Description of physical parameters and their units.
- *measurement* & *measurement_details*: The *measurement* table reflects either single time-series instance in the InfluxDB, and *measurement_details* extends *measurement*

with additional properties, like visualisation colour, available predictions, or aggregation functions.

- *asset:* An *asset* reflects any infrastructure object in the energy island, like heaters, PV panels, rooms, buildings etc. Assets can be connected together via links and can be extended with additional *key-value* properties, which are specific for given asset.
- *heatmap* & *areas:* Stores details of a 2D graphical map, which determines objects' location on the map for visualization. A *heatmap* contains areas, which are polygons linking asset, dashboards and other heatmaps.
- *information_panel* & *information_tile:* An ordered tile-like (information tiles) grid layout used to present information either in full page or single popup.
- *demand:* Table containing schedule of energy management recommendations.
- *notification:* Table containing details about information, warning and error, and which user should be notified about it.

### IV.1.2. Time-series Data

The RENergetic system need to deal with a lot with time-series data. Therefore, all data that varies over time is stored in the time-series database InfluxDB that allows efficient access to the data. The InfluxDB has the following data structure.

- *bucket*: A bucket is a location where data is stored, and all time-series of a bucket share the same retention period. We will create one bucket for each energy island named with the *energy_island_name*.
- *measurement*: A measurement describes the data stored in the associated fields. We will use abstract measurement names, *e.g., outside temperature, energy produced, heat consumed, water flow.*
- *field*: Fields are stored as key-value pairs, it is used to record the actual data values (including the timestamp of the measurement). We will use the names of physical measurements, *e.g. power, energy, temperature* as the field-keys.
- *tags*: Tags are key-value pairs that are used to enrich the data with additional metadata. Queries checking the value of tags are typically fast, because tags are indexed. We will use tags for, e.g., *location, prediction model, prediction time window, aggregation function and window.*

## IV.2. Forecasting

Artificial intelligence-based methods that are covered by this epic are planned to be implemented as a separate forecasting microservice (Figure 8). It uses data storage to obtain historic time series data from energy island. The output of the forecasting service is also stored in the data storage of the RENergetic system. It can be used to generate various dashboards in the interactive platform.

### IV.2.1. AI Forecasting Vision

**General vision on AI approach**

The epic Forecasting gives an energy island the ability to learn patterns in the historical data and provide temporal forward projections in the future about one or more target variables of interest (i.e., one series of measurements in the time series database). Such predicted variables being, for instance, energy island (heat or electrical) energy consumption and supply, renewability of an energy source in percent or specific indexes for building efficiency. The application is evidently cross-sectional to the three Pilots available.

The forecasting methods and the algorithms that generate such forecasts, will produce new data and information. This is used to either inform managers' or citizens' decision-making processes or to directly (automatically) affect and modify the behaviour of an energy actuator in a technical system. In the former case, the forecasting acts as a pure decision support system, in the latter case as automatic Artificial Intelligence (AI) controller of the behaviour of a technical energy system.

Notably in both cases, the capability to do forecasting requires data availability and MLOps (Machine Learning Operations) that deliver, orchestrate, and train ML models iteratively on streams of incoming data on a regular basis.

It is evident that this Forecasting epic is fundamentally a horizontal capacity affecting all the other epics due to its all-purpose applicability and serving across multiple functions and cross-epic objectives. Wherever anticipating a response to mitigate risk or optimise solutions is required, so is the forecasting and machine learning.

It is paramount to stress that this epic does not limit to pure forecasting function, but several other AI skills are in place.

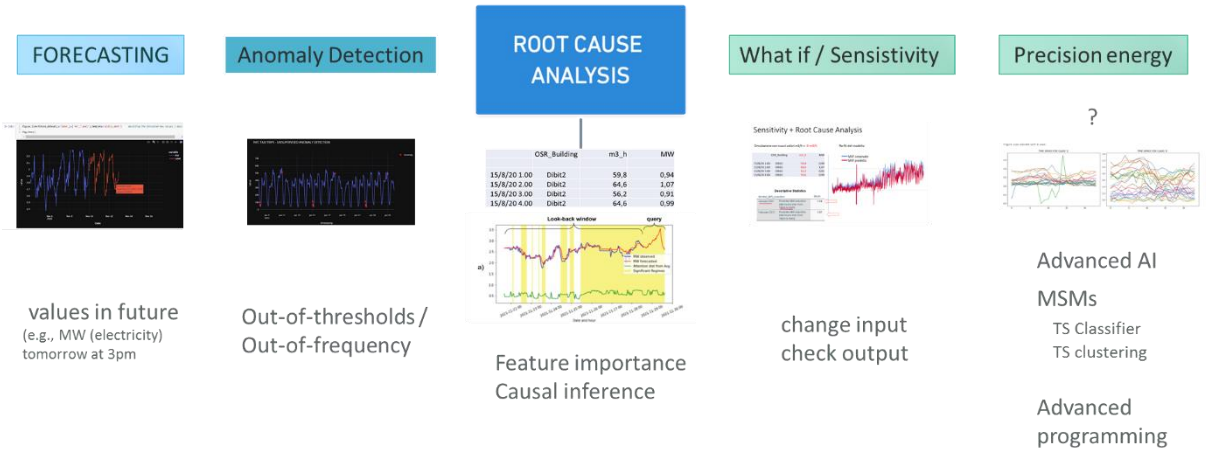



**Figure 17 - AI functions within Forecasting epic**

In fact, as shown in Figure 17, together with Forecasting, other support functions like Anomaly Detection, Root Cause Analysis, Sensitivity and Precision Energy are accounted for by the Forecasting epic.

In this sense, the pure prediction of values (classic forecasting) is supported and enriched by AI algorithms dedicated to further detect:

1) Out-of-threshold values (e.g., over-peaking values)
    a. The target variable values are being monitored to detect projected too high (or too low) values.
    b. The detection algorithm is nominal or embedded into the same forecasting algorithm.
2) Feature importance and causal inference (causal factors)
    a. In presence of exogenous features supporting the time series prediction, only those features important to the task are retained and identified.
    b. In some AI algorithms like Temporal Fusion Transformers the very shape (i.e., profile of the target variable in the time series) can provide evidence on the importance of historical time segments with more weight to the prediction.
    c. Together with detection of important exogenous features or time windows for prediction, some causal inference can be assessed both in model-testing or model building approach interventions. This has relations with sensitivity analysis as well.

3) Predicted output change due to simulated input change ('what if analysis')
    a. A trained and tested AI model (algorithm) can be tested by simulation.
    b. Simulation consists of perturbing input randomly (model building hypothesis) or modify input selectively (model testing) to aim at assessing impact on the predicted output.
4) Meta-predictions or forward oriented precision classifications/predictions based on time-aware energy profile characteristics (precision energy models)
    a. A doubly indexed time series classification is operated either from single or from multiple data sources. The first time-index is the number instances repeated over time. The second time-index is the lag-order of the time series itself.
    b. After a successful time series classification (a dedicated algorithm) with a fixed time series lag-order as by the second index, the classified time series are pooled together backwards and ordered by the first time-index to generate a single composite time series.
    c. The composite time series is then forecasted itself by a dedicated algorithm.
    d. The temporal forward prediction from the preceding step is in turn classified itself by the original pre-trained time series classifier. The result is a classification based on the expected shape or *time series profile* of the target variable over time.
    e. The result is a precise classification prediction based on future forecasted states or expected future energy profiles.

**<u>AI specific approach</u>**

The additional four approaches above (bullets 1) to 4)) together with standard forecasting methods, are expected to generate a strong leverage on RENergetic AI capability. An AI business and operational capacity positioned to replicate RENergetic services over any new Pilot involved in the future.

A core vision for forecasting methods in RENergetic is depicted by the following Figure 18.

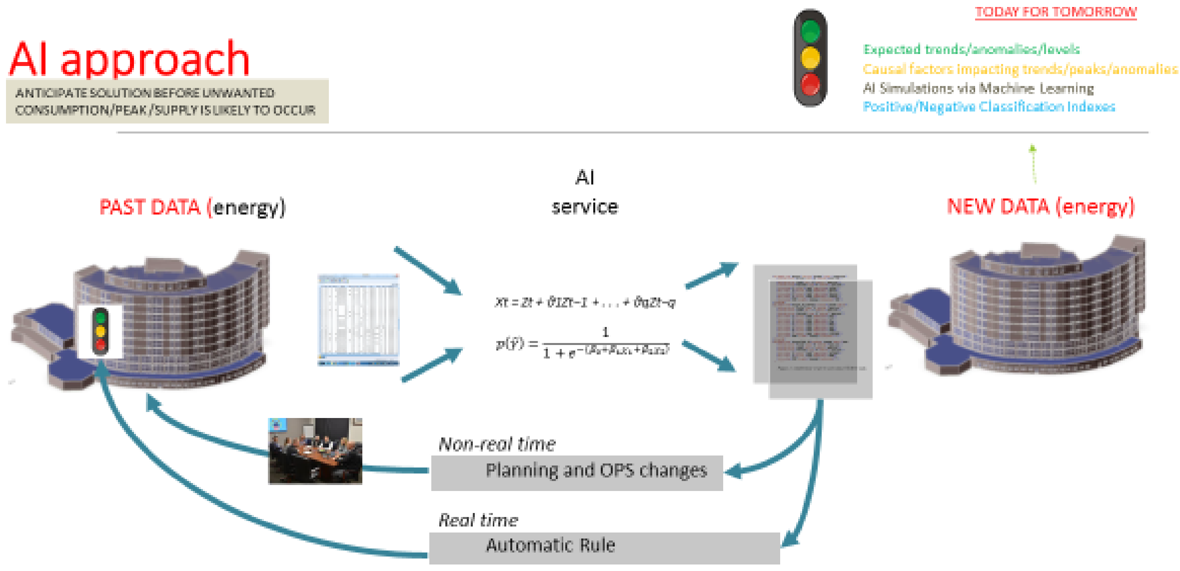



**Figure 18 - AI vision and Forecasting concept**

According to Figure 18, historical data are streamed and dedicated forecasting. Machine Learning (ML) and deep learning (DL) algorithms, are trained on such data according to selected Pilot user stories.

## IV.2.2. User Stories

As of 1 April 2022, a limited number of User Stories per Pilot are available for direct implementation. Each Pilot has responsibility to implement AI skills according to the core objective of each user story.

Each pilot thus can enrich a single story (or multiple stories) with any AI approaches described in Figure 17 above: Forecasting, Anomaly Detection, Root Cause Analysis, Sensitivity and Precision Energy for single or multi-energy source target. In fact, a single story may require forecasting and root cause analysis or simulation according to the narrative in the same user story.

**User Story Setup**

The following simplified Table 1 summarizes some exemplary key user stories across pilots in terms of roles and narrative as of 1st April 2022. Some of such stories are then converted into a final DevOps environment Jira for software transformation.

**Table 1 - Exemplary user stories across pilots before Jira**

| OVERALL STATUS | ROLES | STATUS | GENERIC VIEW |
|---|---|---|---|
| **G01** | Waste heat generation from industry/ Factory | | |
| **G-1-1 POZNAN** | *PSNC technical manager + PUT technical manager* | | *I want to forecast heat flux to PSNC-DC waterloop generation (short term) to perform further analytics and comparisons with historical data. The goal of this analysis is to come up with a strategy to supply as much waste heat as possible in order to avoid having extra heat dissipation cost / maximise profits / perform heat demand response.* |
| **G-3 OSR** | Lead energy manager in OSR (GSD) | READY | *I, as an energy manager of GSD (allocated in OSR) want to forecast the expected MW heat energy demanded by total OSR (Dibit 2 + Dibit 1 + Dimer + Others) in order to spot undesirable trends in MW volumes with risks of being forced to buy energy from national grid or check if surplus MW energy (predicted not to be consumed) could be rendered available to re-distribute or to re-sell.* |
| **G-3 OSR** | Lead energy manager in OSR (GSD) | READY | *I, as an energy manager of GSD (allocated in OSR) want to forecast the heat building inefficiency in OSR with REN-EI Index. This index will anticipate of 1 to 5 days the risk (and associated energy trend pattern) of a building (or more buildings aggregated) to deviate too much from the in-period norm of expected energy heat consumption. The objective is to intercept any signal of expected inefficiency before it occurs in order to enact a proactive response.* |

Not all targeted user stories above will enter Jira environment and selection is based on priority criteria and timing/operational constraints of WP3.

**JIRA implements**

As of 1 April 2022, the following Forecasting epic user stories in Table 2 are being under development and implementation in Jira. More implementations will occur in the course of 2022 and early 2023.

**Table 2 - Exemplary user stories in Jira**

| Pilot | Type of AI | AI action | Description | Roles |
|---|---|---|---|---|
| **Any Pilot** | Forecasting | Forecasted Building Heat Demand for Energy Manager | Forecasts of building heat demand in MW (or other source) with forward horizons between 1 to 3 days. Spot undesirable trends in the expected heating demand | Energy Managers |
| **Any Pilot** | Forecasting | Notifications for Energy Manager about Reaching Heat Demand Thresholds | Detect anomalies in upper peaks levels for MW or other energy source | Energy Managers |
| **Any Pilot** | Algorithm control | Access to Kubeflow for AI Operator | Access to Kubeflow to edit and modify algorithms | AI Operator |

At this stage of developments in Jira the Forecasting and Anomaly Detection, type of AI is available. Other AI algorithmic solutions like Sensitivity, Root Cause Analysis or advanced Precision energy classification are due in the late 2022.

### IV.2.3. Algorithmic View

**Fundamental AI Algorithmic Process[1]**

The general objective functions required by any algorithm in RENergetic forecasting is driven by a clear bias-variance trade off approach applied to any Machine Learning model were:

any k*th* AI RENergetic algorithm

$k = \{1, 2, 3, \ \dots n\}$ expressing an approximation function

$\widehat{F(\dot{x})}$

mapping $x$ observed input vectors to $y$ observed output responses, shall assume

$$\widehat{F(\dot{x})} = \arg\max(y_{obs} \cong y_{pred})$$

to maximise asymptotic approximation between observed $y_{obs}$ and predicted $y_{pred}$ response values: a minimal *bias.* This is operated by some dedicated loss function minimization of the form

$$\arg\min \quad L(y, F(x))$$

which is intended to reduce and contain cross-generalization error performances across measurements and trials: a minimal *variance*.

[1]Mathematical modelling by D. Baranzini, 2022 (personal communication on 27/04/22).

Overall, the RENergetic AI methods are expected to maintain high accuracy and high consistency/reliability across applications. That is, an adequate bias-variance trade off.

**Model registry**

In the forecasting group (FOG) Group folder in the RENergetic SharePoint [26] a model registry folder, called 'Python APIs Registry' contains the continuously updating list of available AI estimators or machine learning algorithms in use (or in future use) for the AI applications to embed in Kubeflow MLOps orchestrator according to Epic needs and deployments. This list is reported in Table 6 in Appendix VI.2. .

**Data**

The data to serve AI algorithms are considered in the light of energy sources (primarily heat, electric, renewable sources). Data for machine learning processes consists in a two-way matrix form with data in long format structure as shown in Table 3 below.

**Table 3 - Example of long format data for AI algorithm consumption**

| Date time Index | Building | Water Temp in | Water Temp out | Hot Water m3/h | MW | m3 | MWh |
|---|---|---|---|---|---|---|---|
| **15/8/20 1.00** | Dibit2 | 87 | 73,2 | 59,8 | 0,94 | 1025204,69 | 22606,2 |
| **15/8/20 2.00** | Dibit2 | 87,8 | 73,3 | 64,6 | 1,07 | 1025263,56 | 22606,2 |
| **15/8/20 3.00** | Dibit2 | 87,4 | 73,2 | 56,2 | 0,91 | 1025314,38 | 22606,2 |
| **15/8/20 4.00** | Dibit2 | 86,7 | 73,3 | 64,6 | 0,99 | 1025381,56 | 22606,2 |

In particular, each column denotes a variable of interest either being the target of prediction (e.g., MW values) or a variable supporting the prediction of the target (e.g., Hot Water m3/h). Each row instead denotes the time index for each feature in the columns. As of 1st April 2022 the data variables and time indexes are contained in the file.xlsx in the FOG folder.

**MLOps schema**

The MLOps schema represents the machine learning operations required to run the RENergetic AI process from data acquisition to model scoring and model drift evaluation. All MLOps steps are represented in Figure 19 below and are all implemented into the chosen MLOps software service Kubeflow as summarized in Figure 20.

All Kubeflow pipelines are easily controlled by the AI Operator in python coding for easer programming, processing re-configuration and scaling up across Epics. Kubeflow is critical ML coordination module within the RENergetic platform software solution.

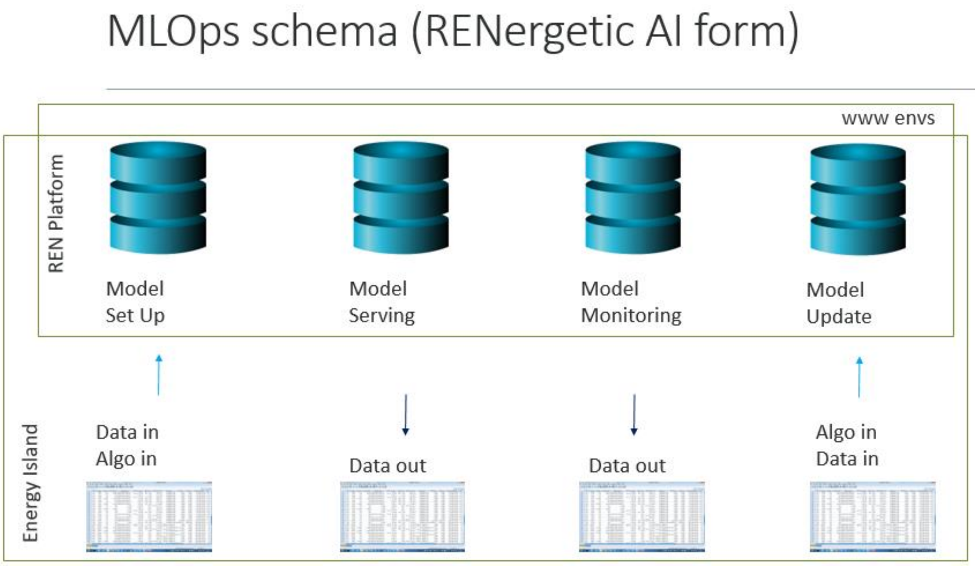



**Figure 19 - MLOps schema reference**

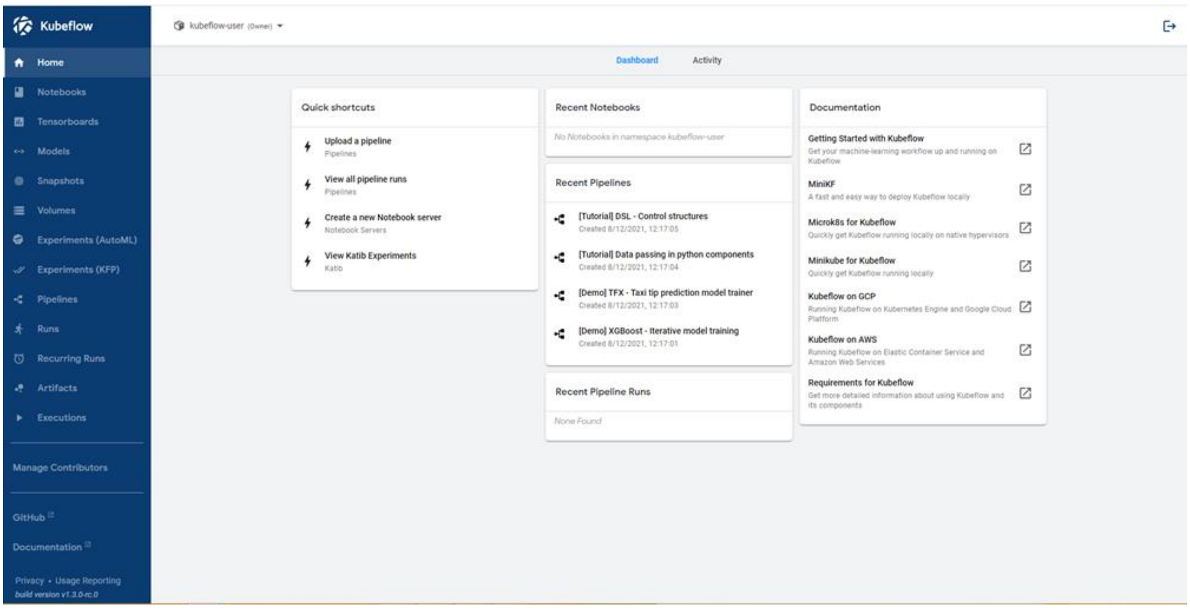
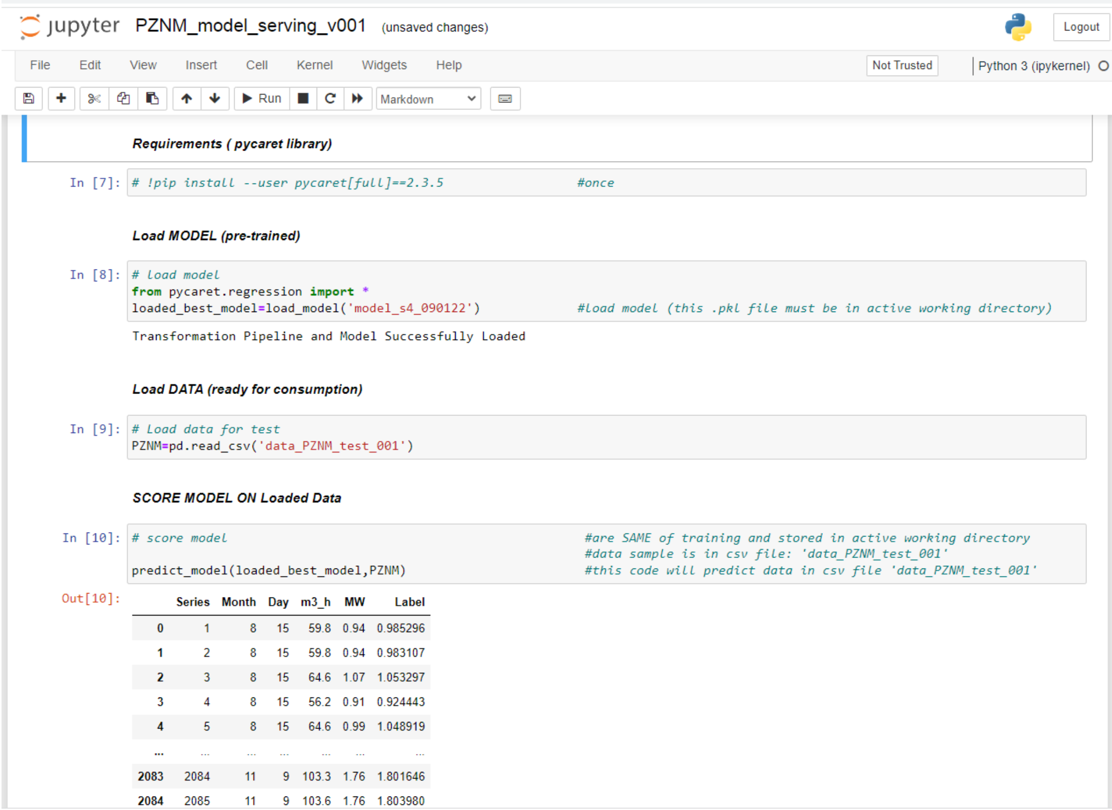



**Figure 20 - Kubeflow software service (configuration point)**

## <u>ML models as Python APIs</u>

To demonstrate the AI application core elements some python interpreter code is provided in Figure 21 and Figure 22 below. The Python code represented in Figure 21 specify the Python coding imports for:

1) Pythion API pre-trained model and relative API to score it in Kubeflow
2) Python APIs to monitor performance according to MAE, RMSE, MAPE and SMAPE metrics



jupyter PZNM_model_serving_v001 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3 (ipykernel)

Run Markdown

***Requirements ( pycaret library)***

In [7]:
```
# !pip install --user pycaret[full]==2.3.5                 #once
```

***Load MODEL (pre-trained)***

In [8]:
```
# load model
from pycaret.regression import *
loaded_best_model=load_model('model_s4_090122')             #load model (this .pkl file must be in active working directory)
```

```
Transformation Pipeline and Model Successfully Loaded
```

***Load DATA (ready for consumption)***

In [9]:
```
# Load data for test
PZNM=pd.read_csv('data_PZNM_test_001')
```

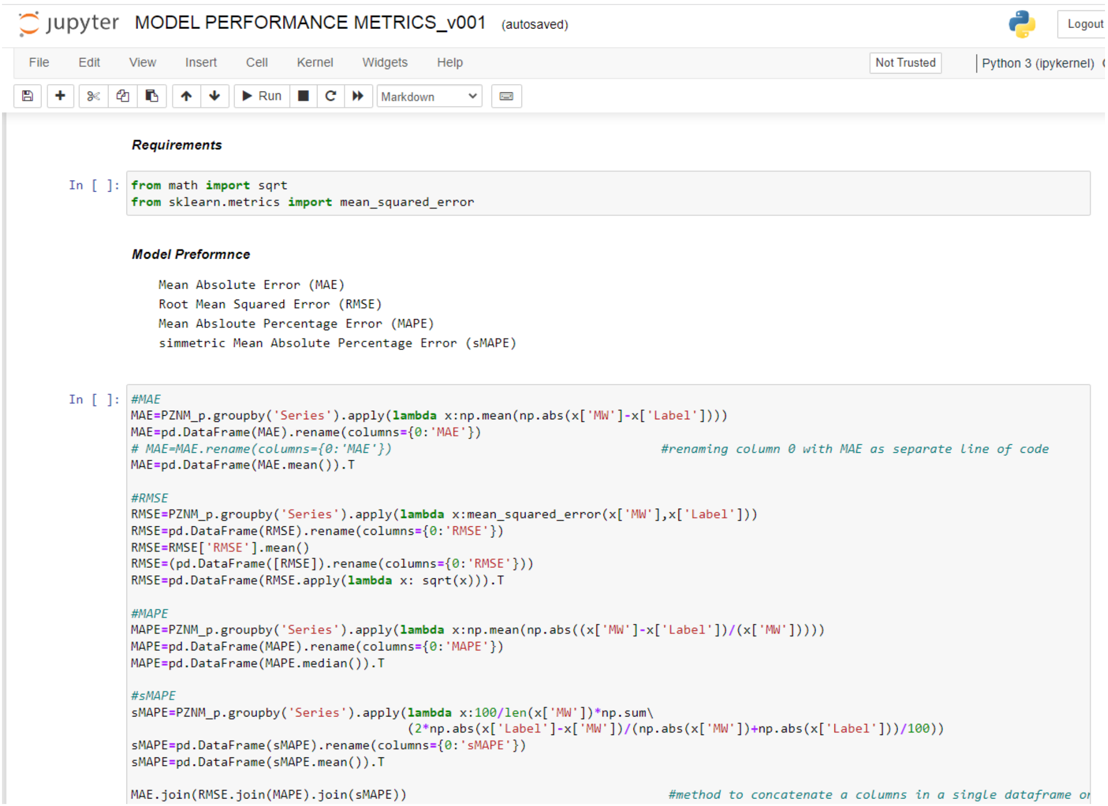
***SCORE MODEL ON Loaded Data***

In [10]:
```
# score model                                                #are SAME of training and stored in active working directory
                                                             #data sample is in csv file: 'data_PZNM_test_001'
predict_model(loaded_best_model,PZNM)                        #this code will predict data in csv file 'data_PZNM_test_001'
```

Out[10]:

| | Series | Month | Day | m3_h | MW | Label |
|---|---|---|---|---|---|---|
| 0 | 1 | 8 | 15 | 59.8 | 0.94 | 0.985296 |
| 1 | 2 | 8 | 15 | 59.8 | 0.94 | 0.983107 |
| 2 | 3 | 8 | 15 | 64.6 | 1.07 | 1.053297 |
| 3 | 4 | 8 | 15 | 56.2 | 0.91 | 0.924443 |
| 4 | 5 | 8 | 15 | 64.6 | 0.99 | 1.048919 |
| ... | ... | ... | ... | ... | ... | ... |
| 2083 | 2084 | 11 | 9 | 103.3 | 1.76 | 1.801646 |
| 2084 | 2085 | 11 | 9 | 103.6 | 1.76 | 1.803980 |

**Figure 21 - Python API to load and score a pre trained model for MW prediction demand**



jupyter MODEL PERFORMANCE METRICS_v001 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3 (ipykernel)

Run Markdown

***Requirements***

In [ ]:
```
from math import sqrt
from sklearn.metrics import mean_squared_error
```

***Model Preformnce***

```
Mean Absolute Error (MAE)
Root Mean Squared Error (RMSE)
Mean Absloute Percentage Error (MAPE)
simmetric Mean Absolute Percentage Error (sMAPE)
```

In [ ]:
```
#MAE
MAE=PZNM_p.groupby('Series').apply(lambda x:np.mean(np.abs(x['MW']-x['Label'])))
MAE=pd.DataFrame(MAE).rename(columns={0:'MAE'})
# MAE=MAE.rename(columns={0:'MAE'})                          #renaming column 0 with MAE as separate line of code
MAE=pd.DataFrame(MAE.mean()).T

#RMSE
RMSE=PZNM_p.groupby('Series').apply(lambda x:mean_squared_error(x['MW'],x['Label']))
RMSE=pd.DataFrame(RMSE).rename(columns={0:'RMSE'})
RMSE=RMSE['RMSE'].mean()
RMSE=(pd.DataFrame([RMSE]).rename(columns={0:'RMSE'}))
RMSE=pd.DataFrame(RMSE.apply(lambda x: sqrt(x))).T

#MAPE
MAPE=PZNM_p.groupby('Series').apply(lambda x:np.mean(np.abs((x['MW']-x['Label'])/(x['MW']))))
MAPE=pd.DataFrame(MAPE).rename(columns={0:'MAPE'})
MAPE=pd.DataFrame(MAPE.median()).T

#sMAPE
sMAPE=PZNM_p.groupby('Series').apply(lambda x:100/len(x['MW'])*np.sum\
                                     (2*np.abs(x['Label']-x['MW'])/(np.abs(x['MW'])+np.abs(x['Label']))/100))
sMAPE=pd.DataFrame(sMAPE).rename(columns={0:'sMAPE'})
sMAPE=pd.DataFrame(sMAPE.mean()).T

MAE.join(RMSE.join(MAPE).join(sMAPE))                       #method to concatenate a columns in a single dataframe or
```

**Figure 22 - Python API to test model cross-generalisation performance**

All data, snippet and python API codes are generated by Dr. Baranzini Daniele in OSR as of 1 April 2022.

# IV.3. Demand Response for Electric Vehicle Charging

Functionalities of manual and automated EV demand response described in this section form the dedicated microservice EV Demand Response in the RENergetic architecture (Figure 8).

## IV.3.1. Vision

### IV.3.1.a. Context

Standard EV charging means that an EV immediately starts charging when connected to the charging station until the battery is full, without interruption, and at the nominal power level. When *smart charging* is possible and allowed by the end user, charging could be altered in different ways:

- The start of the charging could be shifted**.**
- Charging could be temporarily interrupted, possibly several times.
- Charging (temporarily) at lower power levels could be possible.
- Energy from the car battery could even flow towards the charging station and local grid (vehicle-to-grid or V2G).

Smart charging provides flexibility towards the electricity system, which could be used for different scenarios, among others cost optimization via dynamic pricing or by selling flexibility on energy markets, maximizing the use of (locally generated) renewable energy or avoiding local grid congestions.

### IV.3.1.b. Functionality

EV demand response can operate in two main modes:

- **Manual**: The users agree to receive notifications on the best moments to charge their car. These notifications are generated by a scheduling algorithm that tries to meet as good as possible a certain objective (typically from an energy island management perspective) taking all relevant current and forecasted context info into account. Hence, there is no direct control of the charging station. When the car is connected, it immediately starts charging until a full battery is reached or the car is disconnected.
- **Automatic:** User connects car to charging station, indicates a deadline by when the charging should be ready and the current state of charge. The user could also indicate additional preferences/constraints such as a minimum State of Charge (SoC) to be reached, to use only (local) green energy above a certain SoC, etc. An algorithm then remotely controls the charging process to meet as good as possible a certain objective (from an energy island management perspective) taking the provided constraints and other relevant current and forecasted future context info into account.

User experience and intuitive visualization is key, so the core smart charging service should be complemented with user-oriented services to

- Monitor the charging process**,** also providing info on obtained rewards by offering flexibility (lower cost, positive impact on climate, …)
- The ability to easily set preferences such as whether or not flexible charging is allowed (possibly depending on the time of the day/day of the week), minimum SoC to be reached, whether or not only renewable energy should be used for charging (possibly from a certain SoC level on), if notifications can be sent to indicate good moments to charge, etc.
- Check current and expected availability of charging infrastructure and the possibility to book a charging spot.

This epic is linked to the forecasting epic as for an optimal charging control forecasts will be needed on the expected energy needs of the charging stations, other loads within the energy island, generation capacity in the neighbourhood, expected occupancy of the available charging infrastructure, etc.

## IV.3.1.c. Architecture

Figure 23 gives an overview of the Smart Charging subsystem. The specific optimization objective will be defined by the multi-vector optimization algorithm and then translated to concrete actions for the charging of EVs.

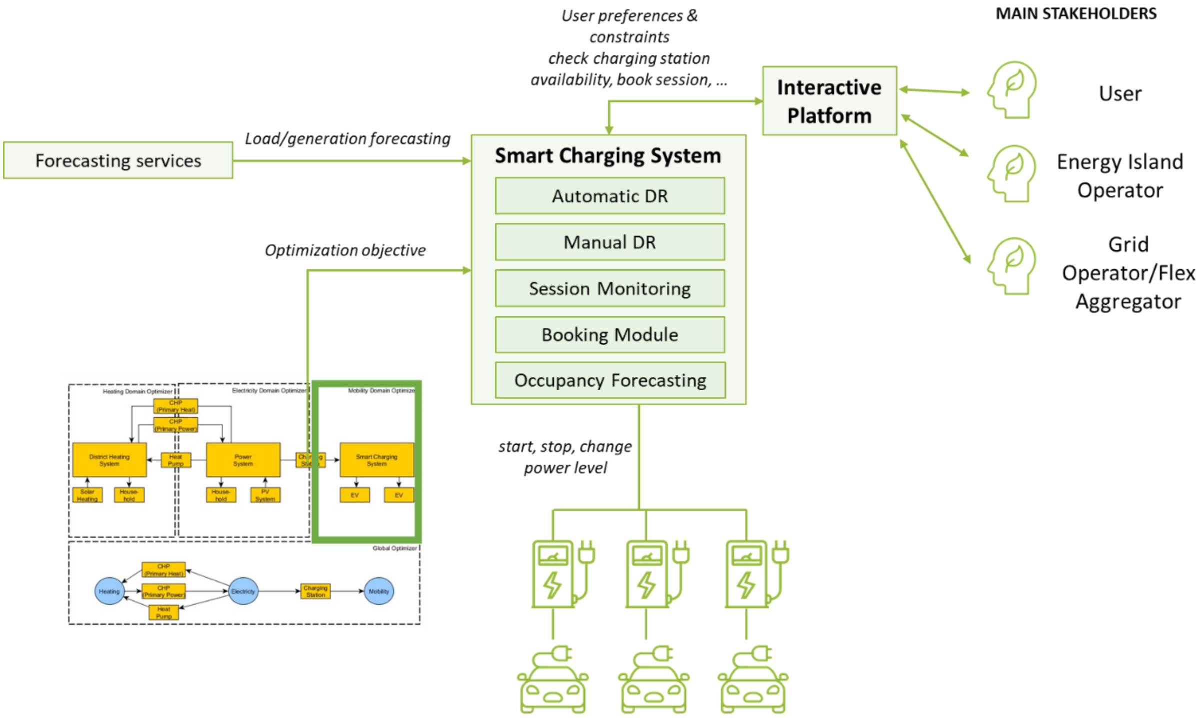



**Figure 23 - Smart Charging subsystem structure**

## IV.3.1.d. Stakeholders & Interactions with Smart Charging System

### *IV.3.1.d.1. End User*

**<u>Manual Demand Response</u>**

Users can see via the interactive platform the best moments to charge their car and/or they can receive direct notifications when there is a 'good' moment to charge. When they connect the car, it immediately starts charging until a full battery is reached. So, no remote control of the charging stations is needed. The user might also receive a notification to unplug the car earlier if e.g., no green energy is available anymore to charge. This division in good/bad moments is generated by a smart charging scheduling algorithm and takes into account user preferences and the local context.

Good moments could be (depending on the provided user preferences):

- **Community** moments which are good moments to charge from an energy management perspective for the energy island (for users that are open to support the local energy island with flexibility)
- **Green** moments when abundant (local) renewable energy is available (when charging with renewable energy is the primary concern)
- **Quiet** moments when there are certainly charging stations available (when availability of charging infrastructure is the main concern of the user)

Intuitive visualization of these good/bad moments to charge is key. E.g., colour indications could be used on a timeline (green/yellow/red). Note that functionality might be needed to avoid that on ‘suggested charge moments’ more people want to charge than there is available infrastructure. Booking functionality would be interesting, but probably difficult to realize if no direct control of the charging stations is possible. Possible alternatives are to show expected occupancy together with the good/bad moments timeline or even use expected occupancy as parameter to determine good/bad moments, or to limit the number of users that receive direct notifications. Figure 24 shows the main steps for manual demand response mapped on the architecture.

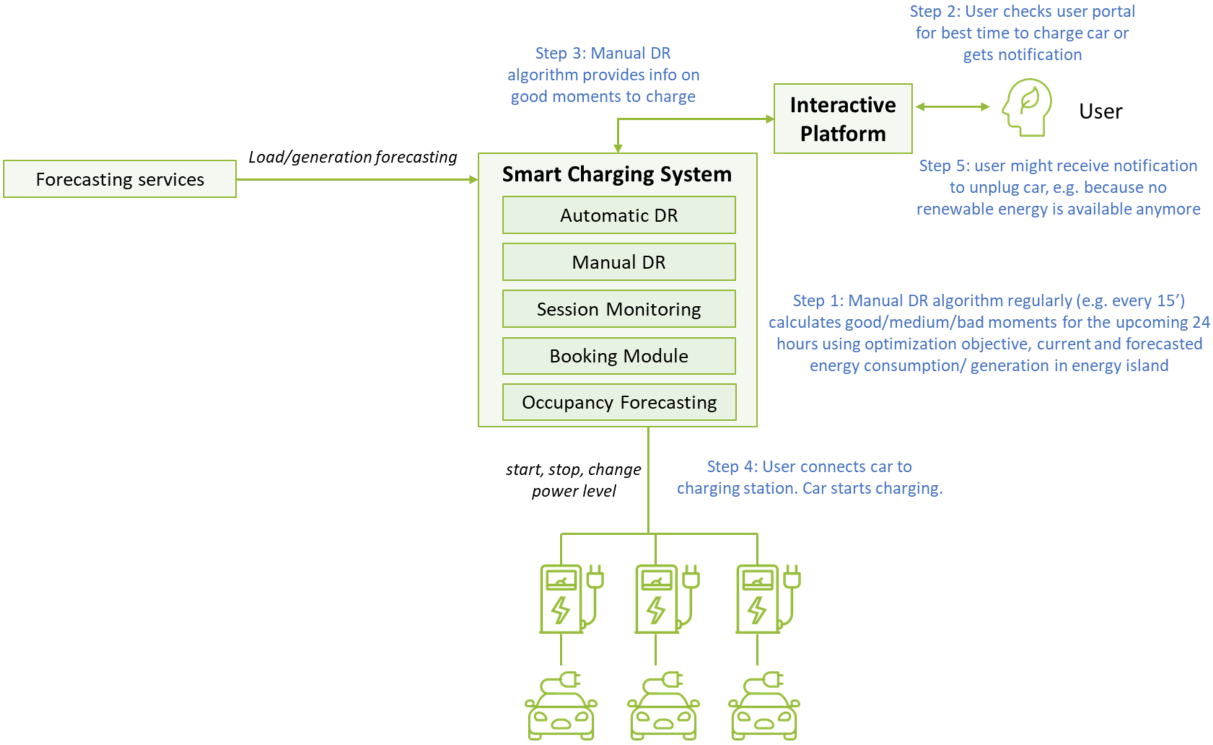



**Figure 24 - Manual Demand Response algorithm**

## Automatic Demand Response

When a user connects the car to the charging station, he/she indicates a deadline when the charging should be finished and the current state of charge via the interactive platform. Possibly extra preferences/constraints can be set (once as default, or per session), e.g., to only use green energy above a certain SoC, minimum SoC to be reached, etc. The smart control algorithm then remotely starts and stops the charging process trying to meet a certain objective (set by the energy island manager and/or the end user) and taking relevant constraints into account, making sure that the car has the requested SoC level by the set deadline. Figure 25 shows the main steps for Automatic Demand Response mapped on the architecture.

Automated Demand Response can be combined with Manual Demand Response by encouraging users to still check the portal for the best moments to charge even if they provide flexibility during the charging process itself.

## Accompanying user services

Intuitive user services needed to

- Allow easy setting of user **preferences**, both default settings as per charging session.
- Allow checking current and expected **availability** of charging infrastructure.
- Allow **monitoring** an ongoing charging session.

- Provide **feedback** on obtained rewards by charging at a suggested moment and/or by providing flexibility during the charging process. WP2 investigates which types of incentives (financial, environmental, combination of both) work best.
- Allow possibly **booking** a charging spot.

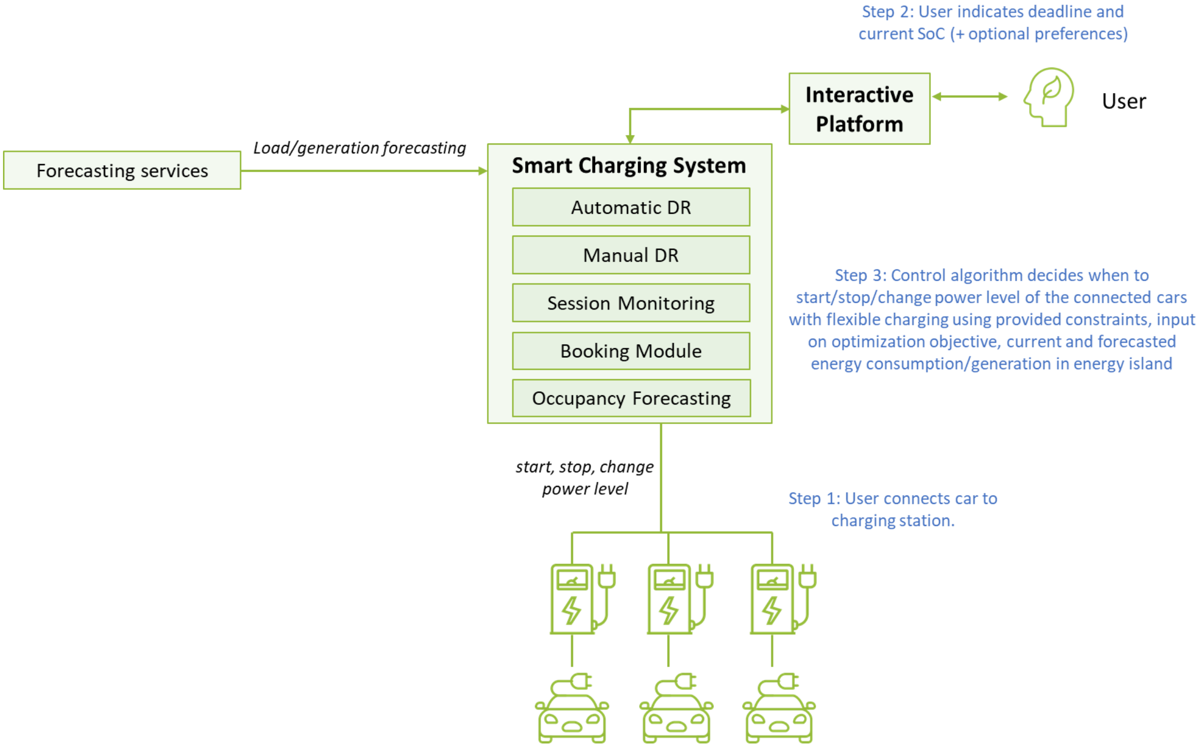



**Figure 25 - Automatic Demand Response algorithm**

### *IV.3.1.d.2. Energy Island Operator*

The energy island operator is responsible to set the objective for the energy island from which the specific objective for the EV charging system is derived. Example objectives could be to balance local supply/demand in the island, minimize total energy costs, avoid grid congestion, maximize profit by selling flexibility to external aggregators, etc. Furthermore, for the energy island operator it is important to provide relevant monitoring information, including:

- Energy consumption of the charging stations and load predictions.
- Occupancy of the infrastructure (e.g., how often is the infrastructure fully occupied?).
- Fraction of users that provide flexibility, at which moments during the day/week, etc.
- Obtained benefits in comparison with a scenario without smart charging.

### *IV.3.1.d.3. External Stakeholders*

Energy island flexibility (including flexibility from the EV charging subsystem) could be sold to external stakeholders such as the local grid operator or flexibility aggregators. The **grid operator** might be interested in the effect on energy offtake/injection by the energy island due to demand response services. The grid operator could provide input on grid constraints or (expected) congestion issues, which could be taken into account to define the objective for the energy island. A grid operator could even directly use available local flexibility to solve local congestion issues. A **Flexibility aggregator** could include flexibility from the energy island in its portfolio and sell it on energy markets (R1/R2/R3 reserve, imbalance market, …) and reimburse the energy island operator. An aggregator would be interested to have info on currently available and expected flexibility and should have access to an interface to activate this flexibility.

## IV.3.2. Algorithmic View

### IV.3.2.a. Introduction

To realize demand response control algorithms different approaches could be used. Traditionally often model predictive control (MPC) algorithms are used [27] [28], where an optimization problem is solved using a predefined model. However, deployment of such model-based demand response algorithms is limited due to uncertainties associated with the assumed models and lack of scalability and generalizability [29] [30].

Model-free approaches circumvent the aforementioned challenges by formulating the problem using a Markov decision process (MDP) where the optimum policy is learned by an agent interacting with the environment [31] [32] [33] [34]. The agent receives a reward/cost in each interaction and is trained to maximize/minimize the long-term rewards/costs. The figure below illustrates the overall goal to jointly coordinate a number or EV charging stations for a particular objective (load flattening in this case).

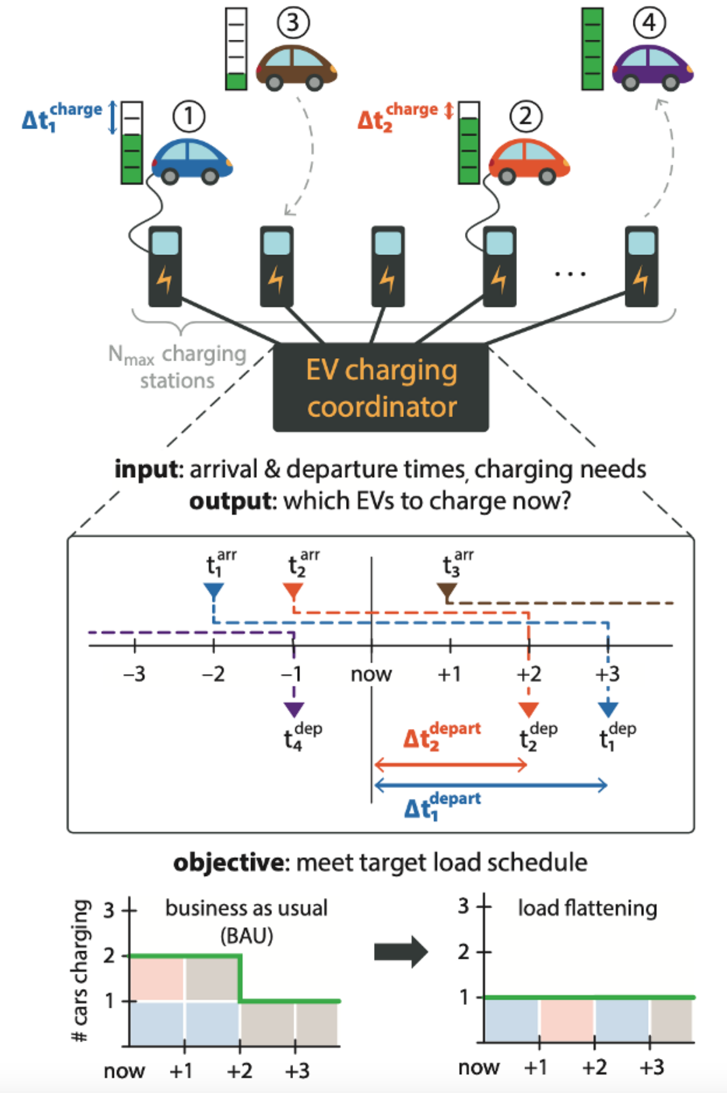



**Figure 26 - An illustration of the EV charging coordination problem. Two cars currently connected (and remaining so for ∆tdepart), with indicated arrival and departure times (tarr and tdep, measured in timeslots) as well as charging needs (noted as time left t)**

In previous work [31], we provided a proof-of-concept for Reinforcement Learning (RL) based demand response for joint EV coordination (with MDP formulation that has a quadratic cost function and state-action representation).

Our recent research as part of RENergetic addresses the real-world implementation and scalability challenge of RL-based control, by defining and exploring state-of-the-art MDP formulations. More specifically, we, (i) defined new compact state-action representations that scale linearly with system capacity and coordination horizon in contrast to exponential scaling in [31], (ii) proposed computationally linear cost functions compared to the quadratic cost function in [31], and (iii) we studied the impact of our cost functions on RL based control policy optimization (by evaluating the computation time per iteration in the FQI algorithm). The next sections give an overview of the different components of our improved approach and evaluation results of the new MDP formulations can be found in Appendix VI.3.

## IV.3.2.b. Markov Decision Process

**State representations**

An individual EV charging session is characterized by the (i) EV arrival time, (ii) EV departure time ($\Delta t^{depart}$), (iii) required energy, and (iv) charging power. The required charging time ($\Delta t^{charge}$) is computed by dividing the required energy with the charging power. A state representation is defined using the information from these features. At time step *t*, the number of EVs in the system is $N^S$, and the available information can be summarized as:

$$Vt\left\{\left(\Delta t_1^{depart},\ \Delta t_1^{ch\arg e}\right),\dots,\left(\Delta t_{N_S}^{depart},\ \Delta t_{N_S}^{ch\arg e}\right)\right\}$$

The underlying idea of charging demand coordination is to exploit the available flexibility in the system. This flexibility, i.e., how much charging can be delayed, is represented by $\Delta t^{flex} = \Delta t^{depart} - \Delta t^{charge}$. The state can thus also be defined as:

$$V_t' = \left\{\left(\Delta t_1^{flex}\right),\ \dots,\left(\Delta t_{N_S}^{flex}\right)\right\}$$

In our previous research [31] we used a matrix state representation using set $V_t$ with 2 parameters ($\Delta t^{depart}$, $\Delta t^{charge}$). We now propose to use a more compact vector state representation using set $V_t'$ with only 1 parameter ($\Delta t^{flex}$).

The matrix state representation summarizes all the information available from the environment and results in a fully observable setting. The vector state representation only summarizes the information about flexibility, and results in a partially observable setting. Yet it still is highly relevant for making charging decisions and this representation results in a smaller number of states in the state space of the problem compared to the matrix state representation.

**Action representations**

Our agent needs to decide which EVs to charge and which ones to delay in a certain state. This decision will be based on the available flexibility. EVs that offer similar flexibility will be considered together. Actions are represented by a vector $u_s$ where the element at position *d*, given by $u^d_s$, provides the number of EVs to charge of the set of EVs that thus have the same amount of flexibility.

The total number of EVs with a certain amount of flexibility is $N^d$. Thus, the element $u^d_s$ of the action vector will be a number in {0, ..., $N^d$} . For example, $N^d = 3$ means that 3 cars offer the same flexibility, and $u^d_s$ is the number of cars that will be charged, and thus lies in {0, 1, 2, 3}.

We scale the elements of action $u_s$ to be numbers in [0,1], representing the fraction of cars that we will charge, i.e., we divide $u_s$ by $N^d$ (amount of cars with a certain amount of flexibility) or $N^{max}$ (total number of charging stations). In our previous research [31], we divided action $u_s$ by $N^d$ to estimate a locally scaled action. To improve interpretability, we now also divide action $u_s$ by $N^{max}$ to estimate a globally scaled action, which keeps the scaling factor fixed.

**Cost Function**

The objective (e.g., flattening the aggregated EV charging load) will be achieved through defining a cost function denoted as C(s, $u_s$, s') quantifying the utility of a transition from state s to s' by taking action $u_s$. The cost function will be related to the charging load. Power consumed from all EVs by taking action $u_s$ in state $x_s$ will be represented by P($x_s$, $u_s$):

$$P(x_s,\, u_s) = \sum_{d=0}^{s_{max}-1} N^d\, u_s^d$$

In our previous work [35], we defined a cost function based on the squared power consumption. We now define cost functions that utilize the information from optimum policies of the preceding days (where we have information for all EV sessions). This is based on the assumption that days with similar EV session characteristics (arrivals, departures and required energy) will have similar optimal solutions. The optimal solution, i.e., power consumption for each timeslot, for a prior day can be calculated using an all-knowing optimum policy (e.g., by formulating the problem as a quadratic optimization problem). The power consumed from all EVs under optimal policy coordination is represented by $P^{opt}(t,e)$, where *t* is timeslot and *e* represents the corresponding episodic day. For preceding $E$ episodic days, the consumed powers from episode *e-1* to *e-E* can be summarized in the following set:

$$P_{t,e,E}^{opt} = \{P^{opt}(t, e-1),\ \ldots,\ P^{opt}(t, e-E)\}$$

By utilizing the information about the optimal policy available in this set, we define two cost functions. These cost functions are based on the absolute difference between power consumed, i.e., $P^{\pi}(x_s,\, u_s)$, and either the average or the median of set $P^{opt}_{t,e,E}$.

$$C_{l,a}^{E}(s,\, u_s,\, s') = \left|P^{\pi}(x_s,\, u_s) - avg\left(P_{t,e,E}^{opt}\right)\right|$$

$$C_{l,m}^{E}(s,\, u_s,\, s') = \left|P^{\pi}(x_s,\, u_s) - median\left(P_{t,e,E}^{opt}\right)\right|$$

The second term of these cost functions is the average/median of power consumption by optimum policy coordination for the preceding $E$ episodic days. Minimizing the long-term cost calculated from these cost functions translates to reducing the deviation of the current charging policy (π) from the optimum policies of preceding episodes. Hence, the current charging policy (π) learns to mimic the behaviour, and subsequently the objective, of these optimum policies. These cost functions will be effective for any coordination objective (e.g., cost-saving, peak-shaving, etc.), as the current charging policy approximates the optimum policies of preceding episodes, which can be trained for any objective.

**State-Action Value Function**

Solving the MDP means finding an optimum control policy. The policy can be identified by evaluating a state-action value function, i.e., the Q-function, and selecting the action that minimizes it at each time step. The Q-function corresponding to the optimum policy can be calculated if the transition probabilities between states are known. However, these are unknown in our setting, hence we use a learning algorithm to approximate the optimum Q-function using batch reinforcement learning.

**Batch Reinforcement Learning**

In batch reinforcement learning algorithms, optimization is performed on data collected in past experiences rather than online interactions from the environment. We use the historical EV data (arrivals, departures, and required energy) and a random policy to collect past experiences. Each experience is defined in terms of (i) an initial state *s*, (ii) the action taken $u_s$, (iii) the resulting state *s'* after taking the action, and, (iv) the associated costs $C(s, u_s, s')$. An experience set denoted by *F* contains tuples $(s, u_s, s', C(s, u_s, s'))$ and is generated based on the state representation, action representation, and cost function.

We use the Fitted Q-iteration [36] algorithm to learn the optimum Q-function from *F*. A fully connected Artificial Neural network (ANN) is used as function approximation.

### IV.3.3. Next Steps

Next steps for this epic include the definition of user stories and mock-ups to start implementing the different required services for demonstration of this epic in the pilots in Ghent and Segrate. Furthermore, validation of the algorithmic approach will be performed with pilot data from the pilots and investigation of other objective functions (e.g., maximizing self-consumption instead of load flattening).

# IV.4. Demand Response for Heating Domain

A Heat DR microservice (Figure 8) is responsible for demand response functionalities in heating domain described in this section. This microservice interacts with Data Storage to retrieve injected data and outputs of the Forecasting service to generate the demand response recommendations and store them in the databases according to the defined CIM.

## IV.4.1. Vision

As for the case of EV demand response heat demand response can be implemented in two ways:

- As automated demand response where the interaction level with the end user (i.e., the non-professional user as e.g., residents, the list with all user roles considered in RENergetic system is in Appendix VI.4. ) is restricted to configuration settings.
- As manual demand response where there is a more or less continuous interaction with the end user, either via push or pull communication.

Heat demand response is characterized by two issues: a high level of inertia and the absence of realistic physical constraints with regards to the availability of power sources. Inertia leads to an extremely high level of latency between the actuation of a power steering knob that affects the temperature in the distribution grid and the impact on the room temperature of the inhabitants of the energy island. This increases the scope for automated demand response, as changes on the supply side take a long time to affect people's comfort, and by nature demand response activations are only temporary. However, this depends to a high degree on the specific use case at hand including the corresponding heating technology as different technologies are characterized of, among others, different latencies.

In the heating domain, theoretically, at all times a temporary gap between supply and demand might always be closed by increasing or decreasing the controllable energy sources that are from the point of view of the energy island "infinite" e.g., oil or gas for a boiler. This simplifies the application of demand response both in an active and in a reactive version without having to bother about very short-term physical issues (contrary to electricity demand response).

At the same time, heating is a particularly sensitive issue for people affected by the system, as comfort and the associated fear of loss of comfort and control play a major role here. This has to be taken into account when designing opt in options or defining the technical actuators of demand response. For instance, should in the case of manual demand response, thermostats be controlled externally, the way to do this and the boundaries of this interference need to be transparently discussed with inhabitants and determined by them.

For the case of heat demand response, the trigger metric might be some cost ratios between the different energy sources, where cost can be CO2 or other cost. It might alternatively be the expected availability of waste heat or a request from the district heating operator. This is feasible for both automated and manual demand response elements; if the solution contains elements of both, a threshold between the two elements could be defined by a temporary range.

### IV.4.1.a. Related User Stories:

User stories are differentiated into both the various roles that are necessary to implement heat demand response and into the two general approaches "automated" vs. "manual" demand response. It is worth noting that automated demand response can be treated as an automatic operation of the system that requires manual confirmation by the technical manager. Such approach is caused by safety reasons and after successful and positive experience can be deployed as a fully automated solution. Here, in order not to inflate the document, only some user stories for the end users and for the technical managers are extracted.

The main use stories for **end users for automated demand response** are:

- I in my role as a resident want the automatic demand response system to operate within my comfort zone in order to ensure my comfort.
- I in my role as a resident prefer a possibility to override the automatic demand response somehow when the temperature is not comfortable in order to secure my own comfort and keep some kind of control
- I in my role as a resident want access to information why my heating is optimized at peak times in order to understand what is happening within my home and heating system.

The main user stories for **technical managers for automated demand response** are:

- I in my role as a technical energy manager want to receive recommendations about when, where and how to change temperature settings in my building heating systems upon some forecasted change in a heat source.
- I in my role as a technical energy manager want to an automated activation of temperature setting in my building heating systems (when, where, how) upon some forecasted change in a heat source.
- I in my role as a technical energy manager want live & forecasted information on heat supply and demand to decide when to activate the automatic demand response system in order to ensure correct reactions in terms of time and quantity.

The main user stories for **end users for manual demand response** are:

- I in my role as a resident want constant life and forecasted information about the availability of heat supply (e.g., waste heat) to be able to adapt my behaviour based on information in order to save money & live more sustainably
- I in my role as a resident want to get notifications that give me an easy understandable signal (e.g., traffic light) and short information (or recommendation when it is red) which tells me when it is best to use my heating or adapt my heating behaviour in order to understand the system better and be able to contribute to sustainability.
- I in my role as a resident want to get individual incentives or rewarding information when I comply with recommendations and actively decide for a change in my heating behaviour in order to feel valued for my contribution.
- I in my role as a resident want to get community incentives or rewarding information when I comply with recommendations.

The main user stories for technical managers **for manual demand response** are:

- I in my role as a technical energy manager want live & forecasted information on heat supply and demand to track the system operation according to my requirements including information (traffic light) displayed to end user.

Figure 27 gives a graphical overview of the different configurations that can be applied based on the sites' characteristics. For instance, individual incentives are only feasible in so far as there are individual data available that keep track of the degree to which the desired behaviour (heat adaptation) has been implemented. If such data is not available, there can be only a community incentive or no incentive.

This shows that the implementation of heat demand response is very specific of each site and has to be configured not only with regards of the desired version of demand response but also with regards to feasibility which is mostly determined by data availability.

In order to make user stories more graspable, additionally mock-ups were developed.

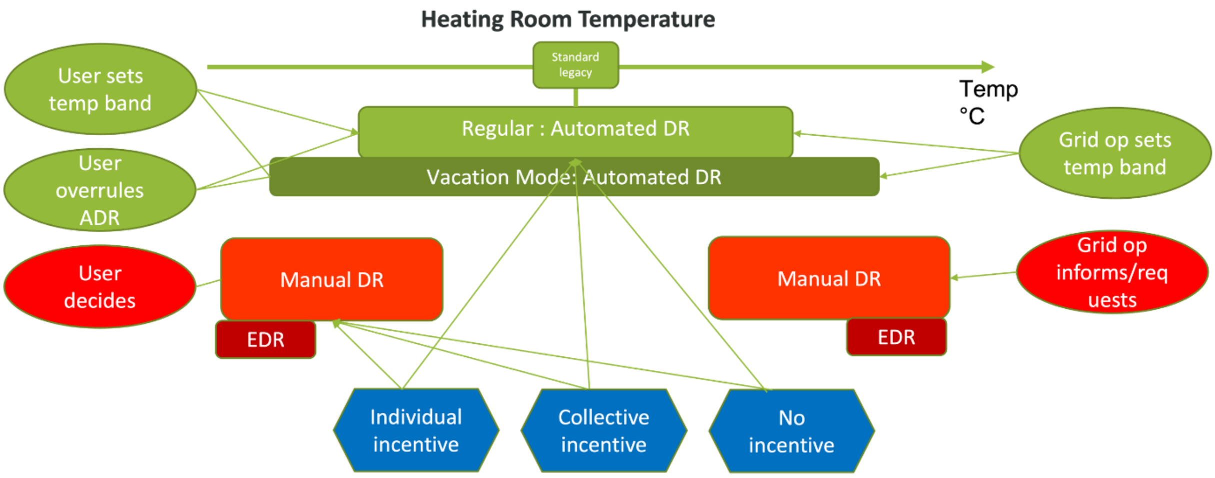



**Figure 27 - Heat demand response configurations**

## IV.4.2. Algorithmic View

For a manual demand response is key that the information is given to the user both when the situation requires but also when the user can best help to improve the renewability of the energy island. We achieve this by dividing the process in two conditions.

- Is the energy island producing energy in a non-renewable way?
- Is the user in question one that could more easily fix this situation?

This is a very simple view of the algorithm developed. We solve the first question by using a custom-made renewability score and the second by using a consumption algorithm to recommend actions.

**Renewability Score**

The metric for the renewability score is calculated as a ratio between the energy that comes from renewable sources ($E_{Ren}$) and the total energy produced($E_{Tot}$) within the island.:

$$REN_{score} = \frac{E_{Ren}}{E_{Tot}}$$

In which $MW_r$ stands for the Megawatt power generated by renewable sources while $MW_t$ is the total Megawatt generation. How we determine the megawatt power generated by those sources. We define for this a sum:

$$MW_r = \sum_N MW_i * R_i$$

$R_i$ is the renewability score of each source, which can be determined as follows:

- **Fixed (F)** - the value for this source is fixed and cannot be changed in time. An example of this would be a full gas boiler which Renewability score would be 0, or a solar thermal, which has a value of 100.
- **Variable** - the value is not fixed in time. Depending on conditions or characteristics, the value of its renewability will change. How this value change creates two different types:
- **Calculated (V-C)** -these sources mix two or more types of sub-sources, which are fixed (in terms of renewability score) energy sources. We could store them separately but that could cause a DB management issue as we plan to store values for the tools in data frames. Therefore, calculations will be done in previous steps to have to values returned: MW produced and R percentage of them from Ren Sources.
- **Inferred (V-I)** - These sources are not a mix of sources but rather a conditioned environment. For instance, if the conditions are in a certain way, then the source is fully renewable and in others not.

The V-I case could be a bit more complicated as maybe there is not enough data to stablish a full score, then it may be considered as a fixed source with 50 as value, so that is not fully renewable. Also, the case of values for which there is an estimation of the renewability as a function of outside conditions that has a specific formula or regression model.

**Temperature Actuation Recommendation**

This is for the case in which the Renewability score for the energy island has fallen below the threshold given by the energy manager. Once this happens, the recommendation to lower the temperature will be given. This is not a recommendation to the entire island but just to select ENVIRONMENT, understanding as environment the most granular separation of areas on the energy island, as an example of this, if the most granular data we have access to it at building level, a different recommendation will be given to each user inside that building.

The action of sending the recommendation is given into three different metrics:

- **Excess consumption based on forecast:** Based on the predictions of the model related to the consumption of the environment, the idea of this metric is to detect abnormal behaviour. We could think of this as a binary variable (Excess/Normal), but we could expand the definition to a 0-10 score. As the excess could be infinite, I suggest we define the excess limit as 2 times the forecasted consumption, meaning if the consumption is 2 times or more the forecasted value then the metric will be a 10, if it is the value or less then the metric is 0. In between, we normalize the variable between 1 and 2 times the forecasted value.
- **Representative consumption:** This metric tries to score the sources by consumption levels, again the environment consuming the most globally will be assigned a 10, the least a 0, with normalization happening in between.
- **Energy Rules:** This is a metric to be given by the energy manager to each environment. If a lower score is given, then the system will not recommend to that user and 10 will surely recommend. This is later reflected in the Energy Manager Scores, which shall be determined based on practical experience or usability.

We unite these metrics via a weighted average. The weights given to each metric is also decided by the energy manager. That way there is a chance for the energy manager to decide how the recommendation goes outside the calculations that we are performing.

As an example, the following situation is presented.

The energy manager has marked 6, 10, 10 as weights for the metrics. This way the energy manager considers that both his own score (Metric 3) and the representative consumption (Metric 2) should be more important than the excess based on forecast.

**Table 4 - Example: Input Variables for each building**

| Environment | Forecast | Real | Total Island | Manager Score |
|---|---|---|---|---|
| **Building A** | 1,58 MW | 1,78 MW | 6,8 MW | 8 |
| **Building B** | 1,69 MW | 2,35 MW | 6,8 MW | 5 |
| **Building C** | 2,78 MW | 2,67 MW | 6,8 MW | 10 |

This way Building C would get a recommendation to lower its temperature.

The view of this score is static, this means that if a building does not follow the recommendation that is not factored into the scores. A way to change this is to use the Energy Manager Score to prioritize buildings which have an easier way to change its energy behaviour and the weights of the scores. For instance, if the weights are [0,0,10] that means that regardless of the results from metrics 1 and 2 the recommendation will be given as the Energy Manager considers.

**Table 5 - Example: final score for each building**

| Environment | Metric 1 | Metric 2 | Metric 3 | Final Score |
|---|---|---|---|---|
| **Building A** | 1,27 | 0 | 8 | 3,77 |
| **Building B** | 7,81 | 6,77,6 | 5 | 6,33 |
| **Building C** | 0 | 10 | 10 | 7,69 |

This example gives a unique recommendation to one building, but this can be changed to show the recommendation to other buildings, so that if the main building has a higher score but is not improving its situation, then the optimal Renewability score will be achieved via other buildings.

Outside of this scope, given access to more data, it would allow us to make a more specific temperature recommendation, close to a real value for the user to input.

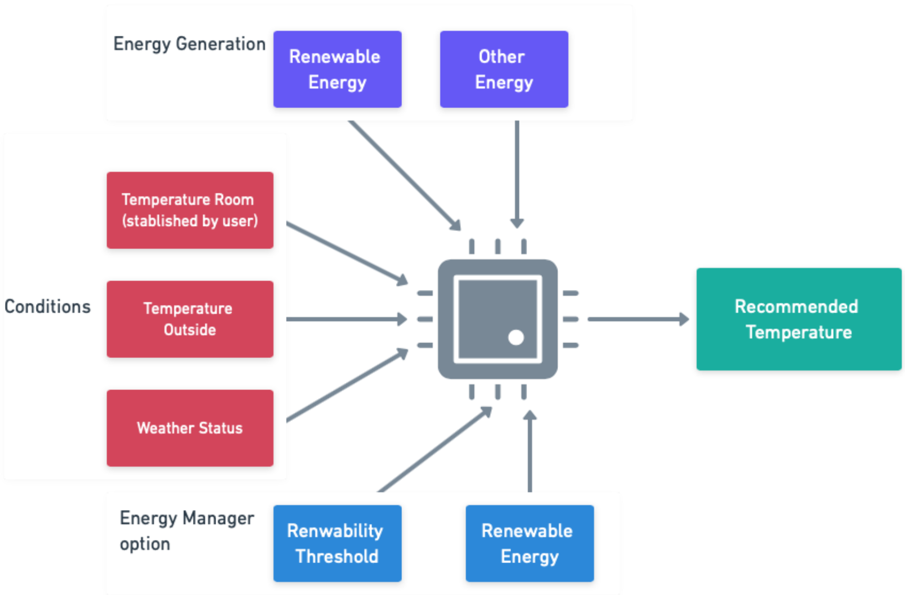



**Figure 28 - Elements of heat demand response algorithm**

The algorithmic description above is a preliminary version, which will be refined in the next steps of the project. The main focus of the current version is to include considerations towards preferences of users and of the energy manager. Open points are to align this with the global and domain-specific optimization schemes regarding interfacing with the global optimizer and the interaction with manually controlled or automatically controlled temperature. To achieve this the exclusive prioritization of one building over the others might have to be relaxed to a distribution of the required reaction based on the relative scores. Further, the triggering of this algorithm can be interpreted as a dead band in which no recommendations are sent to avoid rapid deactivation cycles of heat resources. These rapid activations can be harmful due to the large inertia of these resources. Further refinement of the output signals is possible towards a concrete temperature signal over lose recommendations if more data is reliably available.

## IV.5. Heat Supply Optimization

Functionalities described in this section are implemented in multiple microservices in the RENergetic architecture (Figure 8). Together with Local Waste Heat Optimization (Section IV.6. , heat supply optimization form the domain-specific optimization service that also supports multi-vector optimization service. Forecasting algorithms proposed in this section are planned to be included to the system as a part of the Forecasting microservice.

## IV.5.1. Vision

- Balance heat supply from various sources in order:
  - to reduce the usage of fossil fuels and CO2 emission
  - to reduce the cost of district heating

Other user stories that can be developed during this epic, but are considered as a lower priority:

- Operational quality monitoring**:**
  - Constantly monitor relevant system parameters to ensure service quality and reliability
  - Ensure reliability (continuous operation) and quality of service (fulfilling demands) for all relevant systems
  - Detect 'anomalies' in operational data (unusual operation, faulty equipment, non-optimal user behaviour) to avoid abnormal operation and / or excess energy consumption
  - Predict energy usage in different part of the system in order to improve heat supply balance/optimise energy mix
  - Increased system efficacy – making the system more independent from external disruptions.
  - Have predictions in every other aspect of the heating system operation: weather, building occupancy, heat accumulation in the system (buildings, network)
  - Know in advance prognosed heat demand in specific buildings / facilities (interfaces with users / facility managers)
  - Make sure that the quality of room/apartment heating suffices the requirements from my tenants in order not to get complaints
- Optimize hydraulic management of local heating network in order to:
  - Minimise pumping costs (and related $CO_2$ emission)
  - Utilise the ability to accumulate heat in the system in order to smooth demand peaks and improve efficiency.
  - Optimise heat supply temperature.

This last user story could play an important role in optimizing the sustainability and the economics of the district heating network, but it is more difficult to control these hydraulic parameters through a cloud platform or energy management system (EMS). In practice, we will try to evaluate hydraulic efficiency by analysing parameters and performance of the district heating network (also based on the results of the forecasting epic), as well as adapting programmable logic controller (PLC) settings or consider changes to hardware equipment (settings).

### IV.5.1.a. Example: Importance of the Control of Heat Sources in the Ghent Pilot (CEIP)

Achieving a higher share of waste heat and heat from heat pumps in the energy mix is essential to operate district heating networks in a sustainable and economically viable way. The production profile of these heat sources however does not always fit the demand in the district heating network; increasing the need to fill in remaining demand with unsustainable heat sources (e.g., peak gas boilers).

Two strategies can be enabled to increase the efficiency of waste heat or sustainable sources: adapt the demand profile to the production profile of these heat sources (1), a strategy that will be further developed in the epic “heat demand response” (see Section IV.4. ). Another strategy could be to control the availability of the waste heat production or sustainable sources (2).

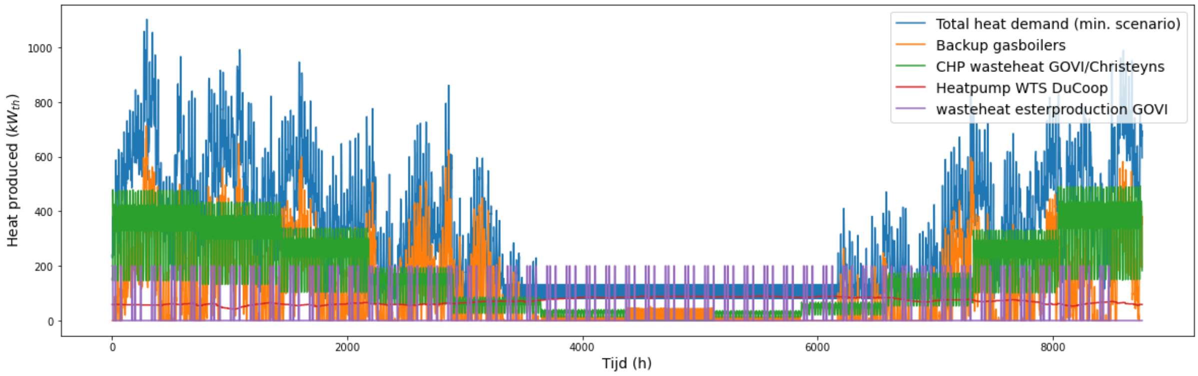



**Figure 29 - Overview of different heat sources and the expected demand of the district heating network of the Nieuwe Dokken in Ghent.**

If the sum of the ‘sustainable’ (industrial waste heat (max. 700kW$_{th}$), heat pump (max. 125kW$_{th}$), CHP (max. 600kW$_{th}$)) heat sources exceeds the demand, some of the excess waste heat cannot be used effectively. If there is a shortage of heat, peak boilers fill in the remaining demand.

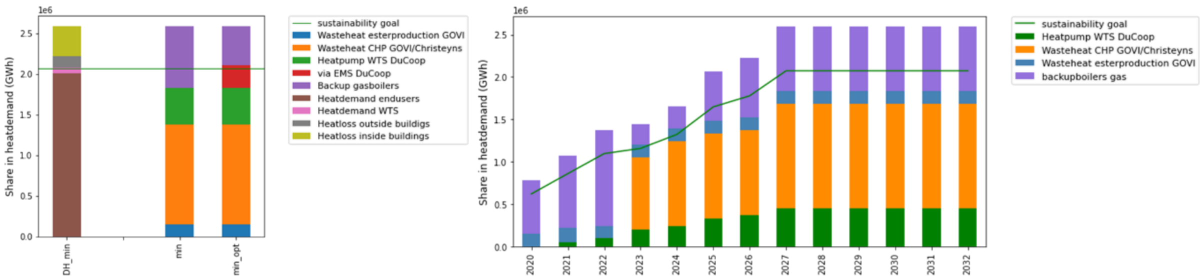



**Figure 30 - Expected share of the different heat sources at full development of the district (1250 inhabitant equivalent in 2027) and expected trend during the coming years.**

Optimization of heat sources using an EMS (see red bar at the left) can be crucial to achieve the sustainability goals of the district heating network.

In the Ghent pilot, the availability of industrial waste heat from the ester production cannot be controlled, as this is dependent of external processes, which are not managed by CEIP or the district heating network operator (DuCoop). The wastewater heat pump takes its heat from the effluent of the local wastewater treatment, which is operated by DuCoop and has the ability to buffer effluent wastewater in the decentralized wastewater treatment plant.

Not all targeted user stories above will enter the Jira environment and selection is based on priority criteria and timing/operational constraints of WP3.

- There is a residual heat demand on the district heating network.
- Heat provided by the heat pump is the cheapest source of heat energy. Calculation goes as follows:
  - Calculate the coefficient of performance (COP) for each operable stage, currently based on historical data.
  - Calculate the heat price for each operable stage, using the electricity price, the COP and the electric power consumed/stage.
- When there is solar or stored energy available, take this into account and recalculate the price of the stages that can (partly) be powered with ‘free’ energy:
  - The set point of the heat sink (in this case buffer tank 3) is not reached.

- There is sufficient heat available in the buffer containing the water that heat is extracted from (buffer flow and T)

- The EMS is active (communication with EMS or default)

Planned expansion: EMS controls the following set points of the heat pump:

- Heat pump operation mode [0, 1, 2]
- Compression stage 1 [0, 1]
- Compression stage 2 [0, 1]
- Compression stage 3 [0, 1]
- Delivery temperature [°C]

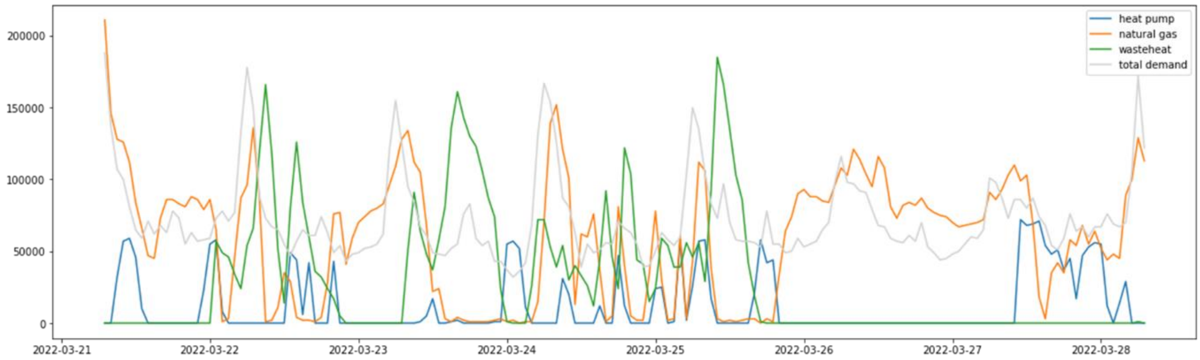



**Figure 31 - Overview of the activation of the heat pump and availability of other heat sources in the district heating network of DuCoop at the Nieuwe Dokken in Ghent (pilot CEIP).**

Since February 2022, the heat pump is controlled to work only when there is sufficient demand, and no other sustainable heat source is fulfilling the demand at the same time. Using the EMS, the heat that the heat pump produces is put to better use; and the heat pump is working on a more optimal temperature level.

### IV.5.1.b. Example: Manual Heat Source Change in Poznan University of Technology (PUT)

Two heat sources are available in the building of the Faculty of Chemical Technology: heat pumps and district heating system. Currently, in the building management system (BMS) there is a static calculation of the profitability of the source (yellow box in the ).

As part of RENergetic, we plan to automate these calculations and base them on real measurement data, not manufacturer data.

## IV.5.2. Algorithmic View

- Forecast energy prices (also from electricity in the case of heat pumps), waste heat availability, technical parameters (e.g., COP)
- Forecast grid prices and specific $CO_2$-emissions (e.g., from transmission system operator datasets or Energy market data)
- Prioritize waste heat usage.
- Optimise operational parameters in order to increase efficiency of waste heat recovery, heat pumps, boilers etc. (in most cases it means lowering supply and return temperatures and correcting process water flows in part of the systems)

Not all targeted user stories above will enter Jira environment and selection is based on priority criteria and timing/operational constraints of WP3.

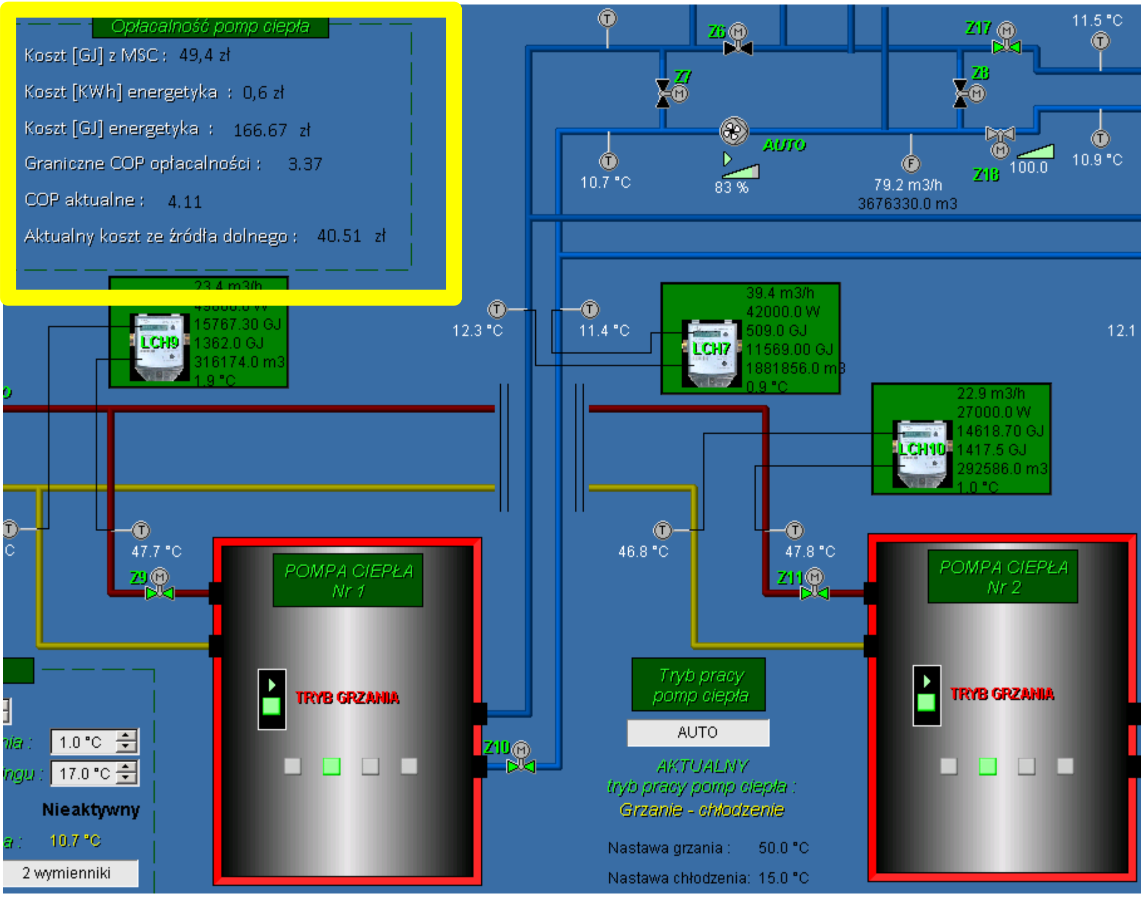



**Figure 32 - A fragment of the synoptic view from the heat pump room.**

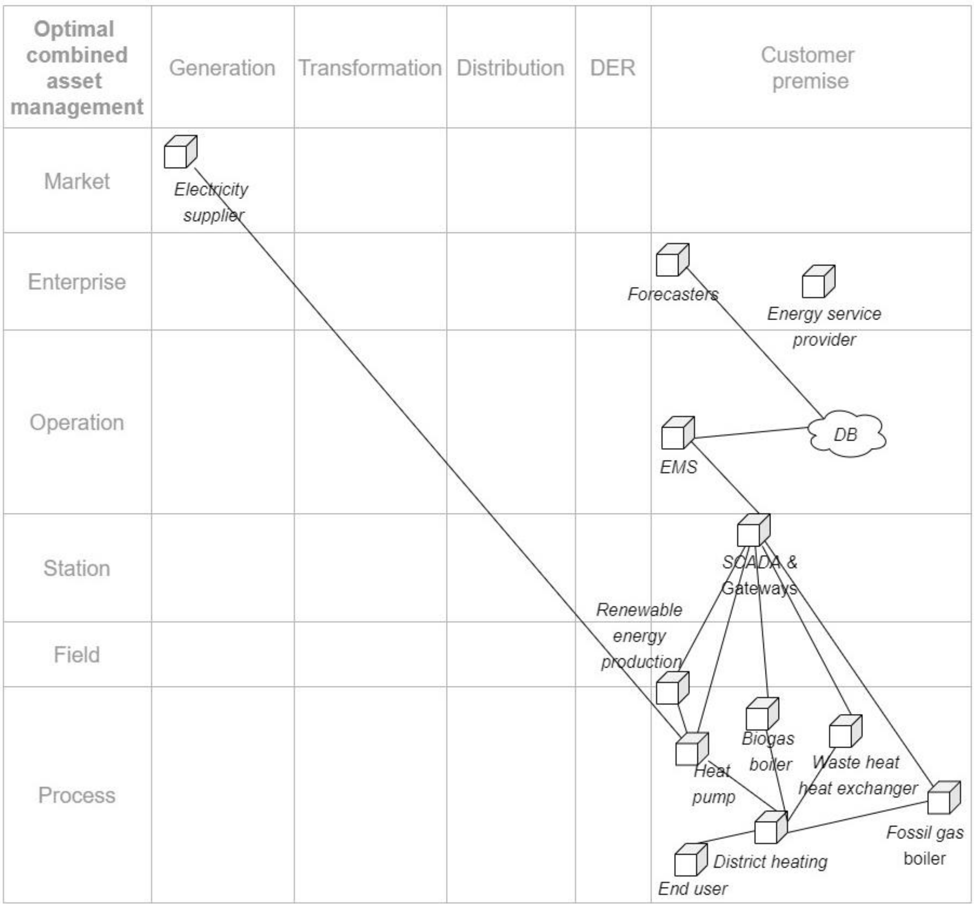



**Figure 33 - SGAM for heat supply optimization**

# IV.6. Local Waste Heat Optimization

Functionalities described in this epic are related to Heat Optimization, Forecasting and Interactive Platform microservices described by the RENergetic architecture (Figure 8). Forecasts and identified anomalies in data, as well as parameters calculated by the heat optimization service are then used to create dashboards in the Interactive Platform.

## IV.6.1. Vision

There are several definitions of waste heat. It may be misunderstood as heat from the incineration of municipal waste or from a combined heat power plant, which are not considered within this epic. Therefore, waste heat is understood as heat reused from a system in which heat generation is not the main functional purpose. So, waste heat can be classified as coming from combustion engines, data centre, system of refrigeration, water from washing, cooling and so on. One of the project pilots is the energy island connected to the district heating network with data centre and university campus. There, the main and highest impact action is related to heat reuse from the data centre. Therefore, waste heat from the data centre is treated as a separate epic and the starting point for replicability for energy islands with the supercomputing centre as the heat source with the possibility of control the parameters and generation.

It is worth emphasizing that there is significant potential to increase efficiency and heat reuse from the data centre. Based on data from 2017, the data centre consumed 416.2 billion kWh of electric energy which corresponds to 2% of world annual electricity consumption [37] . The significant part of this energy can be reused as waste heat for the local energy island.

The main parameters characterizing waste heat:

- Flux [$m^3/h$]
- Heat flux [kW]
- Return temperature [ºC]
- Supply temperature [ºC]

The user stories of local waste heat optimization from the data centre are related to the following important aspects:

- Increase data centre energy efficiency.
- Optimization of waste heat parameters from supercomputers.
- Waste heat prioritization within the energy island.
- Optimization of waste heat utilization, e.g., by preheating of building structures.
- Reduction of costs related to heat within the energy island.
- Monitoring of system parameters.
- Prediction of generation and demand for heat.
- Reducing the usage of fossil fuel units.

### IV.6.1.a. Example: Waste Heat within Poznań Warta Campus

Detailed description of this example can be found in deliverable D5.1, but here is a summary of information based on this document.

Basically, Poznań Warta campus pilot consists of three entities:

- Veolia – district heating operator and the owner of the municipal heat and power plant
- PUT – Poznań University of Technology - university campus with dormitories.

- PSNC – Poznań Supercomputing and Networking Center - data centre owner

The Figure 34 shows the pilot area and the designation of the current and planned heat connections which will be simulated and optimized within the RENergetic project.

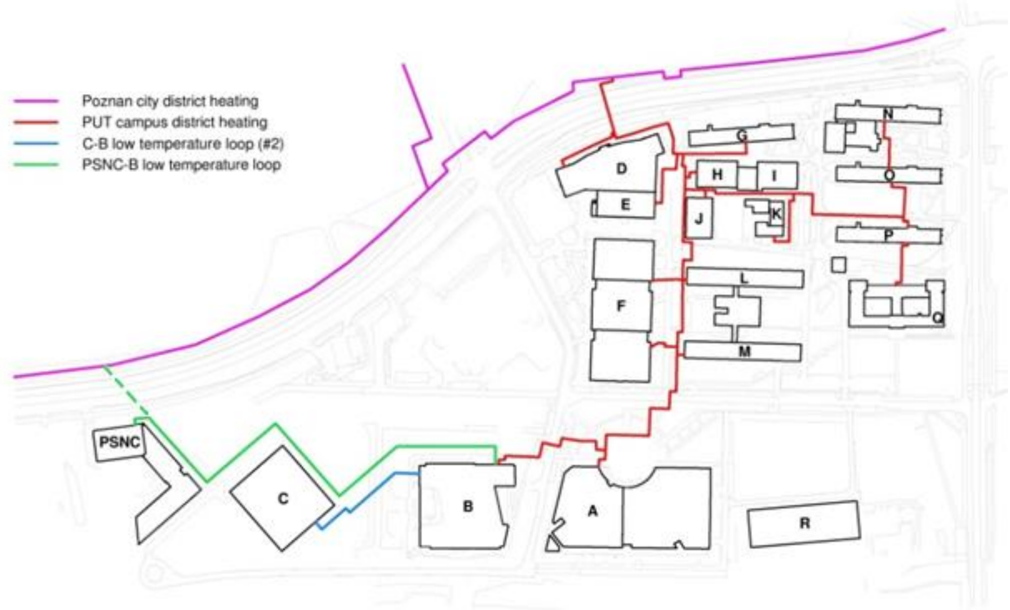



**Figure 34 - Heat connections within Warta campus in operation (blue, red, purple) and planned (green)**

As shown in the picture above, the Poznań pilot considers two possibilities of using waste heat from date centre:

1) First scenario - direct connection of the heat reuse system from the data centre to the district heating network (green dashed line)
2) Second scenario - connection of the heat reuse system from the data centre to the to the university campus (solid green line) with the possibility of selling the surplus to the district heating network.

The implementation of each of the aforementioned scenarios requires a separate optimization of parameters, in particular the temperature ones, as well as energy and economic analysis. Significant environmental benefits are obtained in both cases.

The chart below, also shown in D5.1, illustrates the exemplary analysis of the possibilities of generating heat from the data centre and its use within the campus.

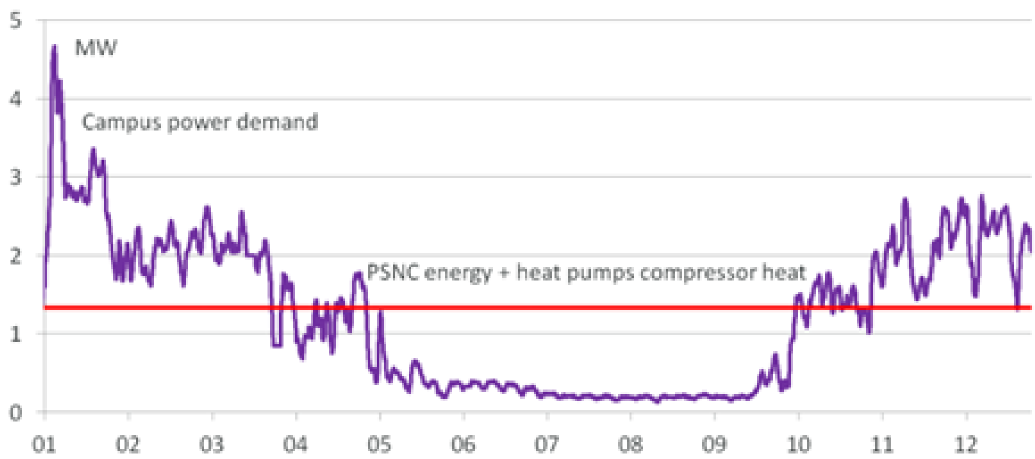



**Figure 35 - Data center waste heat potential related to campus heating demand**

Preliminary analyses showed that heat generation potential including electricity from the compressor totalling 1.3 MW/h will cover campus heat demand during the summer period and will enable the sale of excess waste heat to the district heating network at the level of 15 000 GJ/a. Heating power from PSNC equal to about 2 MW/h will cover the campus demand of around 96% and the sale of approximately 38000 GJ/a of excess waste heat to the district network.

### IV.6.1.b. Related User Stories

The main use stories for **local waste heat optimization related to data center** are:

- Monitoring
  - I in my role as a business manager I want to monitor process of supplying waste heat (focused on sales) in order to keep control of my new business model.
  - I in my role as a technical manager I want to monitor process of supplying waste heat (focused on technical parameters) in order to keep control of my new business model.
  - I in my role as a technical manager want to monitor data centre electric energy consumption in order to keep control.
- Forecasting
  - I in my role as a business manager I want to forecast heat flux to PSNC-DC waterloop generation and electric energy consumption by data centre (long term) to perform further analytics and comparisons with historical data (surplus analysis, optimization). The goal of this analysis is to come up with a strategy to supply as much waste heat as possible in order to avoid having extra heat dissipation cost and maximise profits.
  - I in my role as a business manager I want to forecast heat flux from a data centre, its load and electric energy consumption (long term). I want to obtain simulation results (cost analysis) in order to evaluate how changes of the input features (e.g., load in a data centre, coolant temperatures) would impact the predicted heat flux from a data centre. Simulations should include potential dynamic changes of a data centre state by load shifting, adjusting CPUs clock speed or inlet temperature of servers. The long-term purpose of simulations is to plan development of a data centre and new investments.
  - I in my role as a technical manager I want to forecast heat flux to PSNC-DC waterloop generation and electric energy consumption by data centre (short/long term) to perform further analytics and comparisons with historical data (surplus analysis, optimization). The goal of this analysis is to come up with a strategy to supply as much waste heat as possible in order to avoid having extra heat dissipation cost and maximise profits.
  - I in my role as a technical manager I want to review models (to get insight into importance of input features, e.g., energy consumption, load in data centre, coolant temperatures) used for prediction of heat flux to PSNC-DC waterloop generation, electric energy consumption and load (short/long term) in order to identify features or patterns with the highest impact.
- Detecting anomalies
  - I in my role as a technical manager I want to detect anomalies in load data, electric energy consumption and heat re-use system operation to avoid abnormal operation and / or excess energy consumption. Detection of such problems is especially important when new systems are integrated to a data centre and when changes in data centre operation (e.g., cooling temperatures, load) are introduced.

### IV.6.2. Algorithmic View

Local waste heat optimization from the data centre requires the following points:

- Monitoring of parameters of the waste heat reuse system, such as temperatures and heat flux.
- Forecasting tools for data like potential of waste heat generation, heat demand within energy island, heat pumps parameters, temperatures in the district heating network.
- Forecasting tools related to potential profits from the sale of waste heat in various scenarios.
- Anomaly detection tools.
- Creating a model of the data centre enabling simulations of changes in operating parameters (for example the power usage of DC due to deployment of new IT equipment) and simulation of the waste heat reuse process.

## IV.7. Interactive Platform

The interactive platform is a dedicated microservice (Figure 8) that is responsible for the visual representation of the results of other microservices stored in the databases according to the CIM.

### IV.7.1. Vision

The interactive platform epic is a cross-cutting issue as it is in some way affected by all other epics: It receives input to be communicated from heat supply optimization, heat and EV demand response, amongst others, but indirectly via those also from the forecasting epic.

It is thus a meta-use case that addresses the need for representation, feedback and system analysis of all other use cases.

To this end, the following issues need to be defined:

(1) **WHO FOR?**

(2) **WHAT FOR?**

(3) **WHAT DATA?**

(4) **WHAT VISUALIZATION?**

This implies that not only the objectives of communication need to be clarified (What – to whom) but also the granularity (both temporal and geographical) of data. In dependence of the first two issues different levels of access and interactivity need to be defined, which implies a hierarchical construction of platform pages.

The main user groups interacting with the RENergetic IT system are (more information about the roles to be found in deliverable D7.1):

**The general public**: Part of the visitors that have no invested interested in the energy island, because they are only “passers-by”, visiting the island only for a transitory period of time

**Associates:** Regularly visitors of the energy island that might take an interest in the development of the energy island as it is part of their daily activities as students or staff. Obviously, they have more rights and data issues.

**Tenants:** The part of the energy island’s inhabitants that do not own the building units that they inhabit, but who have a prolonged interest into energy island issues. They also have extended rights, as e.g., participating in manual demand response, and data privacy issues.

**Owners:** In the energy island stakeholder analysis, “owners” are the owners of building units that are – contrary to investors – also operational. Also, they have an invested interest in the energy island.

**Technical (Energy) managers:** These are the professionals operating the infrastructure of the energy island, specifically with regards to the available energy vectors, be it electricity, heat or electricity for EVs.

**Business (Energy) managers:** The professionals that are responsible for cost and benefit assessment of the envisioned energy measures. This does not necessarily only imply financial benefits.

**Sustainability evangelist:** The sustainability evangelist – if this role is defined in the energy island – represents sustainability issues, for the case of RENergetic, mainly regarding energy.

This analysis leads to a basic structure of the interactive platform along two lines: one for the private end-users and one for professionals. The reason is that those two user groups have a different set of skills with regards to graphs and data interpretation. However, in order to avoid double work, whenever possible, joint pages are to be implemented.

As the analysis shows, specifically for end-users, the platform needs to be organized in a hierarchical way where the general public has the lowest set of viewing rights and owners are on the other end of the range. As mentioned, managers will be supplied with a separate line of interactive pages:

For the general public and visitors:

**Information**

- Energy efficiency and energy data on building / area level.
- Visual designed feedback and developments.
- Tips for energy behaviour.

**Interaction**

- Opportunity to learn more.
- Opportunity to ask Questions.
- Feedback for constant updates.

**To-Keep-In-Mind**

- Rebound effects.
- Easily understandable for lays.
- Create attention, provide a touch point.
- What about non-digitals?

For residents:

**Qualitative Information**

- Notifications.
- Visual Feedback.
- Classification in comparison to similar households.

**Quantitative Information**

- Actual energy efficiency & energy consumption data.
- Data comparison to other time points.
- Costs & cost reductions.

**To-Keep-In-Mind**

- Rebound effects.
- Negative emotions, reactance.
- Reversed effects due to comparison.
- Easily understandable for lays.

For managers

**Information**

- At the consumption side and the generation side, together with their interaction
- About forecasting (quantitative info), proposed optimized schedules/plans.
- Also, recommendations, alerts, tips, qualitative info based in optimizers.
- Maybe other data (special events...other predictors)

**Interaction**

- Very usable, graphical, drill down, filter, zoom in
- Information to take decisions and actions.
- Historical info is useful to understand the new info.
- But also, you can input some data/actions to be automated
- Push notifications/reports available.

**To-Keep-In-Mind**

- Not to overlap features already available at the EMS/BMS level
- Focus on making the algorithms useful.
- Specific role for “Administrator”, someone taking care that everything is working OK

The final question to be answered is on which **device** the communication should take place. Among the pilots, the most agreed solution is either an online web-based platform with and without restricted access sections. In some cases, e.g., for manual heat demand response in Ghent, the communication should take place via a screen in the apartments, for other specific functionalities as e.g., manual heat demand response with students at PUT, communication is divided between representation of information on a web-based platform and emails or an app to send adaptation requests.

## IV.7.2. User Stories and Mock-ups

The analysis above results in a set of different sets of user stories aimed at the different user groups of the interactive platform. These are too many, to be displayed here, however, the ones relating to all users of the interactive platform are:

- **IP-0:** Anybody interacting with the Energy Island can access the interactive platform as a visitor using a web page or smartphone app, including the possibility to receive push notifications/emails/messages.
- **IP-1:** Anybody interacting with the platform should be restricted to layers of information - layer 1 generic, simple summary of energy expenditure/emissions of the island; layer 2 accessible via building-wide login, specific but still simple summary of energy expenditure of the building; layer 3, personalized log-in, for access to personal energy expenditure if data is available.
- **IP-2:** Energy information display should be possible for different aggregation levels (sources, energy vectors, geographical aggregation, time). Aggregation levels will be displayed dependent on rights of roles.

In order to visualize the interactive platform, as a first step of the WP3 implementation task, the prioritized interactive platform user stories were poured into a system of mock-ups. Whenever possible, these are based on already existing examples in order to avoid re-inventing the wheel; in other cases, as e.g., heat demand response, clearly new designs have to be created. Two examples that as a next step will be tested with final end-users, are shown here. These mock-ups are subject to ongoing discussions and will most probably be changed due to internal or external feedback.

Figure 36 shows the different communication elements with the final end-users. Green rectangles signify buttons, red rectangles signify relations to other pages of the mock-up system or explanations of different text. This mock-up is the "generic" version that combines all options, which can be mixed and matched by pilots and replicators – of course impacting the elements of the back end necessary to provide the chosen functionality. The mock-up consists of 5 elements: the representation of the renewable share (greenish circle), the recommendation in which direction to adjust temperature, up or down (white rectangle with arrow), the detailed recommendation including temperature (white rectangle with temperatures; not chosen by any pilot), the buttons and an "Away mode" that asks people to turn off the heating in case they leave the building for an extended period of time (greenish rectangle). Also, the latter has not been chosen by any pilot.

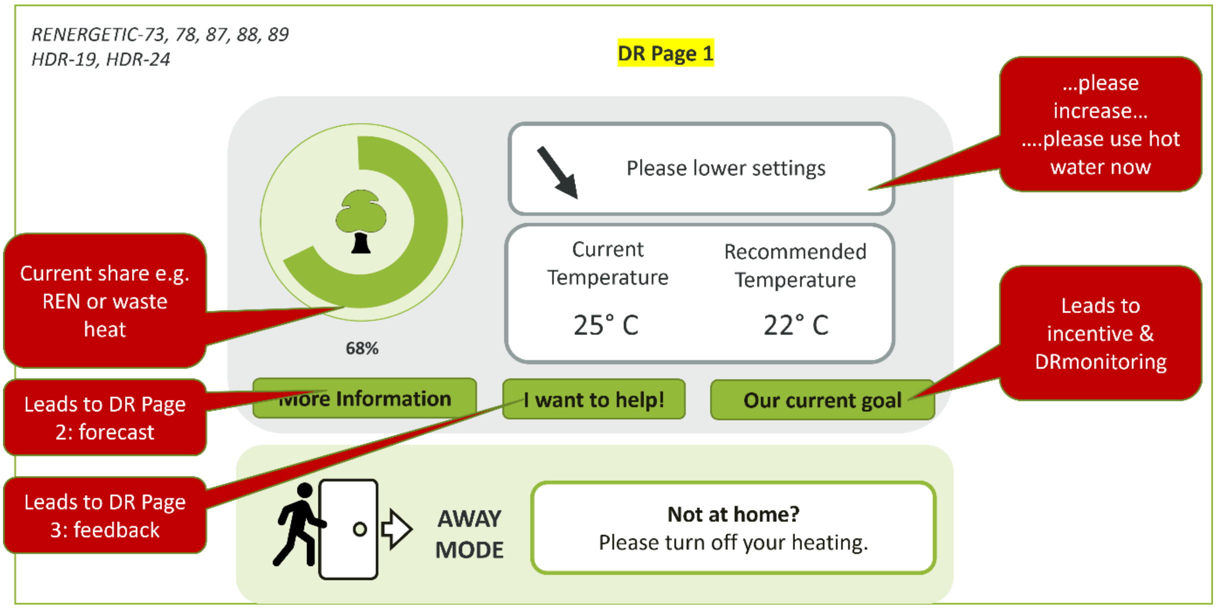



**Figure 36: Mock-up for heat demand response with inhabitants**

Also, Figure 37 displays a part of the heat demand response mock-up series, as they have been created from scratch, not building on existing tools. It shows the degree to which a common objective of the current heat demand response event has been achieved with the accumulated feedback from the participants, either per active acknowledgement (see Figure 36) or per monitoring from the part of the management. The rectangle on the right-hand side represents a community incentive that has to be determined by the pilot or replicator, e.g., a barbecue organized for the community.

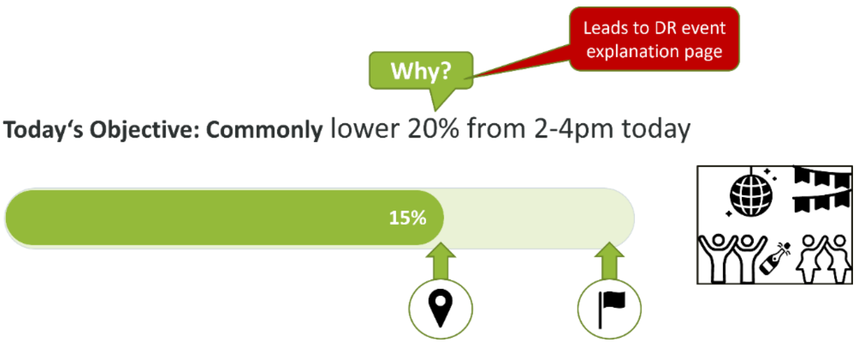



**Figure 37: Mock-up for heat demand response Community Incentive**

# V. References and Internet Links


[1] IEC, “Home - SyC Smart Energy,” [Online]. Available: https://syc-se.iec.ch/.

[2] IEC, “IEC SRD 63200 – SGAM basics,” [Online]. Available: https://syc-se.iec.ch/deliveries/sgam-basics/.

[3] CEN-CENELEC-ETSI Smart Grid Coordination Group, “Smart Grid Reference Architecture,” November 2012. [Online].

[4] B. Kirpes, P. Danner, R. Basmadjian, H. d. Meer and C. Becker, “E-Mobility Systems Architecture: a model-based framework for managing complexity and interoperability,” *Energy Informatics,* vol. 2, no. 1, pp. 1-31, 2019.

[5] “InfluxDB: Open Source Time Series Database,” [Online]. Available: https://www.influxdata.com/.

[6] “PostgreSQL: The world's most advanced open source database,” [Online]. Available: https://www.postgresql.org/.

[7] “Apache NiFi,” [Online]. Available: https://nifi.apache.org/.

[8] “Apache Spark - Unified engine for large-scale data analytics,” [Online]. Available: https://spark.apache.org/.

[9] “Kubeflow,” [Online]. Available: https://www.kubeflow.org/.

[10] “Spring Boot,” [Online]. Available: https://spring.io/projects/spring-boot.

[11] “WSO2 - Next-generation technologies to future-proof your business,” [Online]. Available: https://wso2.com/.

[12] “Vue.js - The Progressive JavaScript Framework,” [Online]. Available: https://vuejs.org/.

[13] “Grafana: The open observability platform,” [Online]. Available: https://grafana.com/.

[14] “Keycloak,” [Online]. Available: https://www.keycloak.org/.

[15] “Docker,” [Online]. Available: https://www.docker.com/.

[16] “Kuberneters,” [Online]. Available: https://kubernetes.io/.

[17] “Red Hat OpenShift makes container orchestration easier,” [Online]. Available: https://www.redhat.com/en/technologies/cloud-computing/openshift.

[18] “OKD.io,” [Online]. Available: https://www.okd.io/.

[19] “Bitbucket,” [Online]. Available: https://bitbucket.org/product.

[20] “Jenkins,” [Online]. Available: https://www.jenkins.io/.

[21] "Maven – Welcome to Apache Maven," [Online]. Available: https://maven.apache.org/.

[22] L. Mezzalira, "Git Flow vs Github Flow," [Online]. Available: https://lucamezzalira.com/2014/03/10/git-flow-vs-github-flow/.

[23] "Jira," [Online]. Available: https://www.atlassian.com/software/jira.

[24] "Confluence," [Online]. Available: https://www.atlassian.com/software/confluence.

[25] "SonarQube: Code Quality and Code Security," [Online]. Available: https://www.sonarqube.org/.

[26] "RENergetic SharePoint," [Online]. Available: https://gfi1.sharepoint.com/sites/RENergetic/Documentos%20compartidos/Forms/AllItems.aspx.

[27] N. Dkhili, J. Eynard, S. Thil and S. Grieu, "A survey of modelling and smart management tools for power grids with prolific distributed generation," *Sustainable Energy, Grids and Networks,* vol. 21, p. 100284, 2020.

[28] D. Dongol, T. Feldmann, M. Schmidt and E. Bollin, "A model predictive control based peak shaving application of battery for a household with photovoltaic system in a rural distribution grid," *Sustainable Energy, Grids and Networks,* vol. 16, pp. 1-13, 2018.

[29] N. Sadeghianpourhamami, T. Demeester, D. F. Benoit, M. Strobbe and C. Develder, "Modeling and analysis of residential flexibility: Timing of white good usage," *Applied energy,* vol. 179, pp. 790-805, 2016.

[30] T. A. Nakabi and P. Toivanen, "Deep reinforcement learning for energy management in a microgrid with flexible demand," *Sustainable Energy, Grids and Networks,* vol. 25, p. 100413, 2021.

[31] N. Sadeghianpourhamami, J. Deleu and C. Develder, "Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning," *IEEE Transactions on Smart Grid,* vol. 11, pp. 203-214, 2019.

[32] Z. Wan, H. Li, H. He and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Transactions on Smart Grid,* vol. 10, pp. 5246-5257, 2018.

[33] T. M. Hansen, E. K. P. Chong, S. Suryanarayanan, A. A. Maciejewski and H. J. Siegel, "A partially observable markov decision process approach to residential home energy management," *IEEE Transactions on Smart Grid,* vol. 9, pp. 1271-1281, 2016.

[34] M. v. Otterlo and M. Wiering, "Reinforcement learning and markov decision processes," in *Reinforcement learning*, Springer, 2012, pp. 3-42.

[35] M. Lahariya, N. Sadeghianpourhamami and C. Develder, "Reduced state space and cost function in reinforcement learning for demand response control of multiple EV charging stations," in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 2019.

[36] M. Riedmiller, "Neural fitted Q iteration--first experiences with a data efficient neural reinforcement learning method," in *European conference on machine learning*, 2005.

[37] Q. Huang, S. Shaob, H. Zhanga and others, “Development and composition of a data center heat recovery system and evaluation of annual operation performance,” *Energy, Volume 189, 116200,* 2019.

[38] M. Riedmiller, “10 steps and some tricks to set up neural reinforcement controllers,” in *Neural networks: tricks of the trade*, Springer, 2012, pp. 735-757.

[39] P. J. Huber, “Robust estimation of a location parameter,” in *Breakthroughs in statistics*, Springer, 1992, pp. 492-518.

[40] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski and others, “Human-level control through deep reinforcement learning,” *nature,* vol. 518, pp. 529-533, 2015.

# VI. APPENDIX

## VI.1. CIM Relational Database

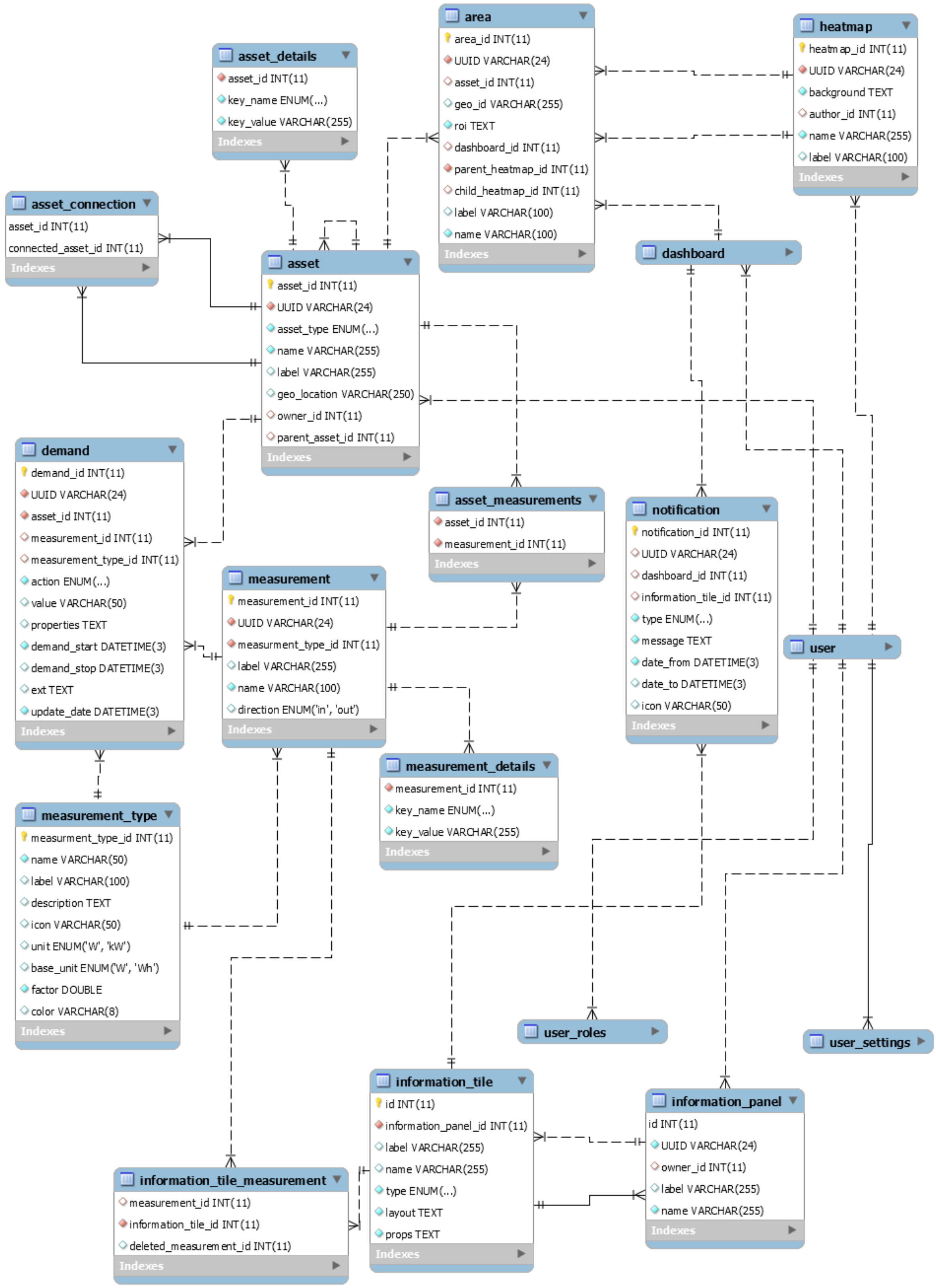



**Figure 38 - Database schema**

## VI.2. Forecasting Models Registry

**Table 6 - Model registry for RENergetic Forecasting, Anomaly Detection, Root Cause Analysis, Sensitivity and Precision Energy (prioritized API models are orange coloured). Status as of 1 April 2022**

| API | URL | Description |
|---|---|---|
| **PyCaret** | https://pycaret.org/ | Provides both statistical ML and DL modelling |
| **Sktime** | https://www.sktime.org/en/stable/index.html | Forecasting + classification + clustering |
| **tslearn** | https://tslearn.readthedocs.io/en/stable/index.html | Forecasting + classification + clustering |
| **Darts** | https://github.com/unit8co/darts | Python – from statistical or DL approaches |
| **Arrow** | https://github.com/arrow-py/arrow | Data engineering for timeseries |
| **greykite** | https://github.com/linkedin/greykite | LinkedIn to time series |
| **tsfresh** | https://tsfresh.readthedocs.io/en/latest/ | Python – automate exogenous features |
| **pyts** | https://pyts.readthedocs.io/en/stable/ | A Python Package for Time Series Classification |
| **ROCKET (+mini-rocket)** | https://github.com/angus924/rocket | In sktime |
| **tsai** | https://timeseriesai.github.io/tsai/ | AI time series best complex DL approach (colab examples) |
| **Cesium** | https://cesium-ml.org/ | Advanced platform for timeseries |
| **Featuretools** | https://featuretools.alteryx.com/en/stable/ | automated feature engineering |
| **Statsmodels** | https://www.statsmodels.org/stable/tsa.html | Statistical models (e.g., ARIMA SARIMAX etc...) |
| **pmdarima.arima.AutoARIMA** | https://alkaline-ml.com/pmdarima/index.html | Grid search for model order for pdq (AR, differenceing and q) and seasonal model order for PDQs |
| **Prophet** | https://facebook.github.io/prophet/ | Facebook forecasting |

| | | |
|---|---|---|
| **Neuralprophet** | https://github.com/ourownstory/neural_prophet | Facebook forecasting |
| **Orbit** | https://eng.uber.com/orbit/ | Uber - forecasting |
| **Merlion** | https://github.com/salesforce/Merlion | Python forecasting |
| **Pastas** | https://pastas.readthedocs.io/en/latest/index.html | Python forecasting |
| **pyflux** | https://github.com/RJT1990/pyflux | Python forecasting |
| **Temporal Fusion Transformers** | https://www.sciencedirect.com/science/article/pii/S0169207021000637; GitHub example: https://github.com/h3ik0th/TFT_darts/blob/main/TFT_2g6_gpu.ipynb | Python, TensorFlow, other APIs for Deep Learning forecasting |
| **N-BEATS** | https://towardsdatascience.com/n-beats-unleashed-deep-forecasting-using-neural-basis-expansion-analysis-in-python-343dd6307010; GitHub example: https://github.com/h3ik0th/ES_energy_Transformer/blob/main/NBEATS_energy_03.ipynb | Python, TensorFlow, other APIs for Deep Learning forecasting |

## VI.3. EV Charging Experimental Results

### VI.3.1. Experimental Details

**Data and Model Specifications**

Our dataset to train the RL based control policy is derived from real-world data collected by ElaadNL since 2011, from 2500+ public charging stations, from which we selected the data for 2015. We represent this data in an episodic format, such that each episodic 'day' starts at 7 am and ends 24 hours later (the day after at 7 am). Further, we assume an empty car park at the end of each episode (all EVs leave the charging stations). A terminal state stabilizes the learning process in FQI that adopts a neural network-based function approximation [38]. The time granularity is set to $\Delta t^{slot}$ = 2 h. We jointly coordinate $N^{max}$ = 10 charging stations, i.e., at most 10 EVs can be connected simultaneously.

For training the RL agent, we start by creating the experience sets *F* containing past experiences for multiple episodes. For each episode, we start from the first state of a day and randomly choose an action from the set of possible actions in each state and observe the next state and the associated state transition cost until the terminal state is reached. This single sequence of states and actions is referred to as a trajectory. Each transition in this trajectory is saved in the experience set in the form of tuples (s, $u_s$, s', C(s, $u_s$, s')). We randomly generate 5000 trajectories for each episode.

An ANN architecture is used to estimate the Q-function from the experience set *F* using FQI. This network consists of an input layer and 2 hidden layers with ReLU activation functions. There are 128 and 64 neurons in the first and second hidden layers respectively. The output layer has a single neuron with a linear activation function. The input of the network is a vector that is created by combining the state and action. We use Huber loss [39] instead of mean-squared-error for improving the stability in learning in our algorithm [40].

**Experiments**

**Table 7 - Experience sets (F) generated for Experiment 1 (observability) to evaluate the effect of information provided by state-action representation on the performance of learned RL control policy.**

| Experience Set | State Representation | Action Representation | Cost Function |
|---|---|---|---|
| ***F1*** | Matrix | Locally Scaled | Quadratic |
| ***F2*** | Vector | Locally Scaled | Quadratic |
| ***F3*** | Matrix | Globally Scaled | Quadratic |
| ***F4*** | Vector | Globally Scaled | Quadratic |

*Experiment 1 Observability:* Evaluation of the impact of different state-action representations. We generate 4 different experience sets, summarized in the table above. Each experience set is used to learn a control policy.

The state-action representations affect the space complexity of an experience set (*F*), and the learning speed of the control policy. To compare different *state-action* representations, we perform an increasing window validation where the size of training datasets is different (see figure below). The training datasets are generated from {30, 60, ..., 270} episodes, and we test on the immediate next 30 episodes.

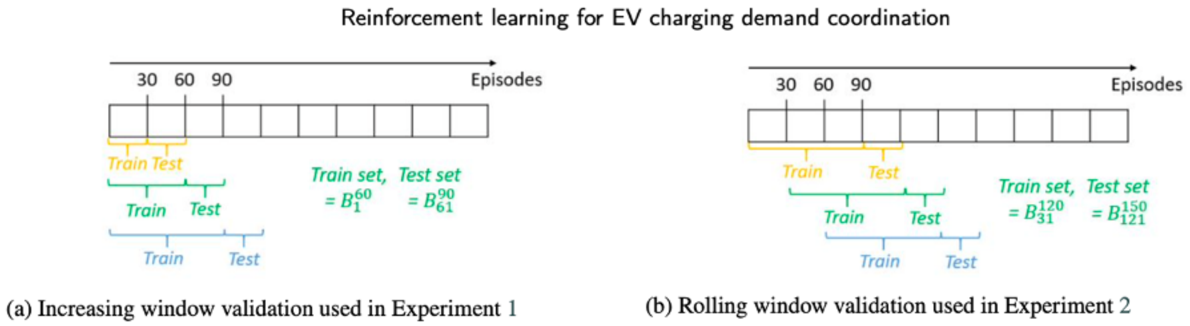



**Figure 39 - Train and test data selected in different validation methods used in experiments**

*Experiment 2 Credit assignment:* Investigation of the impact of the cost definition on the training and performance of the learned optimum policy. Credit is assigned to each transition (taking an action on a given state) based on the defined cost function in the MDP formulation. Based on the different cost functions defined we generate 3 different experience sets, summarized in the table above. Each experience set is used to train a control policy.

In the case of different cost functions, the space complexity of the experience set is not affected. Linear cost functions are dependent on the optimal charging policies for the preceding $E \in \{1, 5, 10\}$ days. To train the policies on the same size of data, but with different preceding days, we evaluate the cost functions using a rolling window validation where the size of training sets is kept fixed (see figure above). Data were used for weekdays, which have similar EV session characteristics.

**Table 8 - Experience sets (F) generated for Experiment 2 (credit assignment) to evaluate the effect of different cost functions on the performance of learned RL control policy.**

| Experience Set | State Representation | Action Representation | Cost Function |
|---|---|---|---|
| ***F5*** | Vector | Globally Scaled | Quadratic |
| ***F6*** | Vector | Globally Scaled | Linear (Average) |
| ***F7*** | Vector | Globally Scaled | Linear (Median) |

**Performance Evaluation**

To evaluate the performance of the learned policy, we use a metric defined as normalized load, which is relative to the load achieved by the optimal policy (obtained from solving the problem as an all-knowing quadratic optimization problem). A normalized load of 1 means that the optimal policy is reached.

Furthermore, for performance comparison we include the normalized load for (i) BAU: a business-as-usual policy characterized by continuously charging each EV upon arrival, and (ii) Heur: a discrete-action heuristic policy that assumes that individual EVs are charged uniformly over their entire connection time.

*Policy Training Time:* defined as the time it takes for an RL agent to be trained. During FQI, we run 12 iterations to train the ANN for each selected training dataset. We record the time for these iterations for all the learned policies.

## VI.3.2. Experimental Results

**Observability: State-Action Representation**

We evaluate policies trained on different state-action representations by analysing the results from Experiment 1. The figure below provides the normalized load comparison for control policies for different MDPs. Each box is constructed from 30 normalized loads calculated for each episode in the test set. We also perform a Wilcoxon signed-rank test on these normalized loads to quantify statistically significant difference among different control policies (significant for p-values ≤ 0.05). Using RL based demand coordination provides 30%-50% improvement in performance compared to the BAU control policy, depending on the training set size and the underlying state-action representation in the MDP formulation.

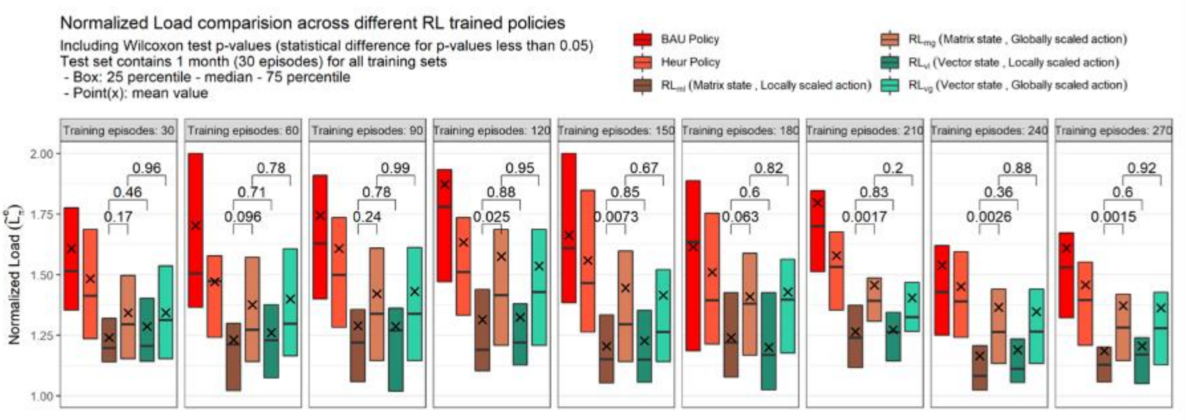



**Figure 40 - Normalized load for RL based control policies trained with different state-action representations. Each box is constructed from normalized loads of 30 episodes in the test set. (Wilcoxon test p-values: statistically significant difference for p-value < 0.05)**

Locally scaled actions train a better performing control policy compared to globally scaled actions. The reason is that locally scaled actions explicitly calculate the percentage of EVs in each flexibility bin, which helps in learning a superior policy, in contrast to globally scaled actions where this information is implicit. This performance gains increases with the increase in training data size. For a provided action representation, both matrix and vector state representations have similar performance. Furthermore, the performance of the control policies improves with an increase in the training data size.

Training time depends on the MDP formulation as each state-action representation has different space complexity. Utilizing vector state representation in MDP formulation leads to linear space complexity compared to the quadratic space complexity in matrix states representations. This results in a reduction in max training time of control policies trained using vector state representations, as shown in the figure below. Training time increases with the number of episodes in the training data. We note that local scaling of actions decreases the training time compared to global scaling, and the lowest training times are reported for the vector state, locally scaled actions policy, 30% less than for the training time of the matrix state – globally scaled actions policy.

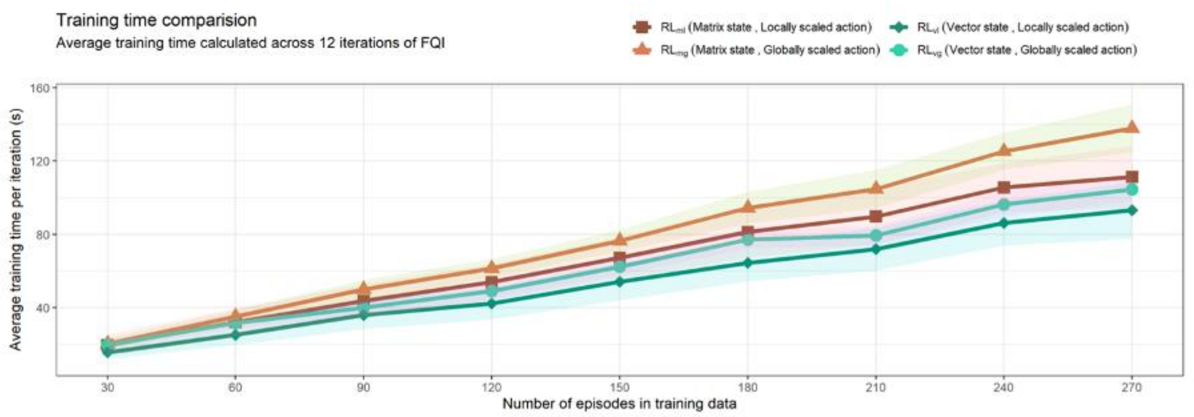



**Figure 41 - Training time for RL based control policies trained with different state-action representations. (Points and solid lines: Average value calculated across all validation sets, Shaded area: 25 to 75 percentile)**

**<u>Impact of Cost Functions Definitions</u>**

The performance of a trained control policy depends on the cost function, where an effective cost function helps to achieve faster convergence by providing informative rewards. To study the effect of cost functions on performance and convergence, we analyse the results of Experiment 2. We use the FQI algorithm to train control policies using MDP formulations characterized by the different cost functions defined above. The figure above compares the average normalized load incurred in different control policies. Control policies are evaluated in the test set after each iteration, and the average normalized load is calculated for all episodes in all validation sets. Note that linear cost functions are based on the optimum solutions of preceding $E$ episodes, and we choose $E$ = 1 for this comparison. Similar results are observed for other values of $E$. A policy based on quadratic cost function takes 8-10 iterations for the FQI algorithm to converge, whereas the policies trained on linear cost functions take 3-4 iterations to converge. Furthermore, we notice after a single iteration, both control policies based on linear cost functions perform much better than the policy trained with a quadratic cost function.

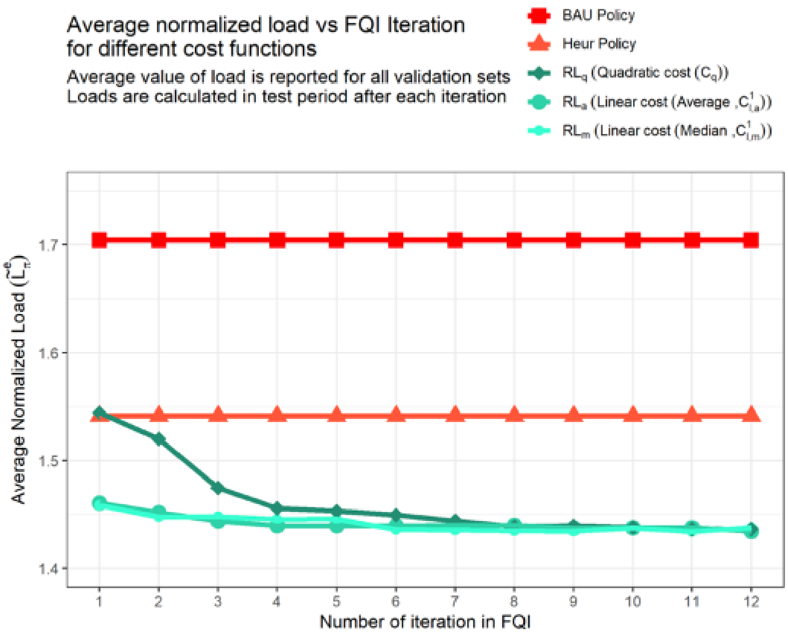



**Figure 42 - Average normalized load per iteration. Linear cost functions with $E$ = 1, i.e., optimum solution of one preceding day.**


**Figure 43 - Normalized load for RL based control policies with linear cost functions that use $E$ ∈ {1, 5, 10}. (Wilcoxon test p-values: statistically significant different for p-value<0.05)**

The figure above shows the normalized load for control policies trained for linear cost functions with $E$ ∈ {1, 5, 10} (average/median of optimum solutions is calculated for preceding $E$ episodes). We notice similar performance for average and median based linear cost functions. Policies trained with average/median of preceding 10 episodes has similar performance to a policy trained with information of a single preceding episode.

## VI.4. User Roles in RENergetic Platform

The following table contain current list of the roles. These are subject to ongoing discussions, and will most probably be changed due to internal or external feedback.

**Table 9 - User Roles in RENergetic Platform**

| ID | Role | Description |
|---|---|---|
| **EC-O** | Energy Consumption - Owner | final end consumer owning an apartment/building; most consumer rights |
| **EC-T** | Energy Consumption - Tennant | final end consumer, renting; needs information for behaviour change |
| **EC-A** | Energy Consumption - Associate | final end consumer, spending a lot of time; needs information for behaviour change |
| **EC-PB** | Energy Consumption - Passer-By | final end consumer, temporary, just passive recipient of information |
| **EC-EV** | Energy Consumption - EV Charger | final end consumer at charging stations, charging process-based electricity information |
| **EC** | Energy Community | Energy Island community - comprising a subset of all other roles |
| **EP_HI** | Energy Production Heat - Internal | local producer of heat, maybe of various sources, fossil & renewable, optimization of resource usage |
| **ES-HE** | Energy Trader Heat - External | external producer of heat, specific, maybe dynamic renewable share, collaboration with district heat network |
| **EP-EE** | Energy Production Electricity - Internal | local producer of electricity, maybe various sources, fossil & renewable, optimization of resource usage |
| **ES- E** | Energy Trader Electricity - External | |
| **SA-H** | Scheduling Agent - Heat | Central Heat management, technically scheduling internal production to optimize cost/$CO_2$ |
| **SA-E** | Scheduling Agent - Electricity | Central Electricity management, technically scheduling internal production to optimize cost/$CO_2$ |
| **SA** | Global Scheduling Agent | Central Management of Heat & Electricity |
| **SO-H** | System Operator - Heat Network | Operator of local heat network, important is cost optimization, grid quality maintenance, voltage control |
| **SO-E** | System Operator - Micro Grid | Operator of local heat network, important is cost optimization, hydraulic optimization |
| **DHNO** | District Heat Network Operation (Heat, external) | External district heat network: temp constraints, collaboration with external heat producer |

| | | |
|---|---|---|
| **DSO** | Distribution System Operation (Electricity, external) | External grid; maybe sending regulation service requests |
| **IS-H** | Imbalance Settlement Responsible - Heat | Central heat management, technically balancing heat demand and supply |
| **IS-E** | Imbalance Settlement Responsible - Electricity | Central electricity management, technically balancing heat demand and supply, |
| **BA-H** | Billing Agent - Heat | Preparing heat invoices for energy consumers |
| **BA-E** | Billing Agent - Electricity | preparing electricity invoices of energy consumers |
| **DP-H** | Data Provider - Heat | monitoring of heat data |
| **DP-E** | Data Provider - Electricity | monitoring of electricity data |
| **F-H** | Forecasting - Heat | forecasting of heat data |
| **F-E** | Forecasting - Electricity | forecasting of electricity data |
| **LMO-H** | Local Market Operator/Interface External Grid - Heat | provides a service whereby the offers to sell electricity are matched with bids to buy electricity |
| **LMO-E** | Local Market Operator/Interface Distribution system operator - Electricity | provides a service whereby the offers to sell heat are matched with bids to buy electricity |

RENergetic