FleXible user-CEntric Energy poSitive houseS

Deliverable 3.3: EXCESS Flexibility Analytics Module

LC-EEB-03-2019

New developments in plus energy houses (IA),

Proposal No. 870157

Title	Document Version
D3.3 EXCESS Flexibility Analytics Module	1.0

Project Number	Project Acronym	Project Title
H2020-870157	EXCESS	FleXible user-CEntric Energy poSitive houseS

Contractual Delivery Date	Actual Delivery Date	Deliverable Type*-Security**
31 December 2021	08 February 2022	P-CO

*Type: P: Prototype; R: Report; D: Demonstrator; O: Other.

Security Class: PU: Public; PP: Restricted to other programme participants (including the Commission); RE: Restricted to a group defined by the consortium (including the Commission); CO: **Confidential, only for members of the consortium (including the Commission).

Responsible	Organisation	Contributing WP
Konstantinos Latanis	Suite5	WP3

Abstract
This deliverable describes the design and development of the EXCESS Data Analytics Framework, which comprises the Comfort Profiling, the Generation Forecasting, the Demand Forecasting, the Dynamic VPP Configuration, the Context-Aware Flexibility Profiling and Analytics, the Flexibility Analytics Visualizations and the Energy Consumption Visualizations components. The EXCESS Data Analytics Framework receives data from the EXCESS Data Management Platform and performs various analytics through dedicated algorithms in order to provide visualizations to EXCESS end users and facilitate the optimal control of devices and loads in the demo sites' buildings of the EXCESS project towards realizing the PEB concept.

Keywords
EXCESS Data Analytics Framework, comfort profiling, generation forecasting, demand forecasting, flexibility profiling, VPP configuration, algorithms, visualizations, dashboards

Revision	Date	Description	Author (Organisation)
V0.1	20.12.2021	*Draft*	Konstantinos Latanis (S5)
V0.2	26.01.2022	*Contribution to section 4*	Miltos Gakis (CGSoft)
V0.3	26.01.2022	*Contribution to section 5*	Michael Kernitzkyi (JR)
V0.4	01.02.2022	*Contribution to section 7*	Koen Allaerts (VITO)
V0.5	03.02.2022	*Contributions to all sections and draft ready for internal review*	Konstantinos Latanis (S5)
V0.9	07.02.2022	*Feedback from internal review received*	Ingrid Kaltenegger (JR), Reinhard Mayr (NETx)
V1.0	08.02.2022	*Deliverable ready for submission to EC*	Konstantinos Latanis (S5)

This project has received funding from the
European Union's Horizon 2020 Research and Innovation Programme
under Grant Agreement № 870157.

More information available at www.positive-energy-buildings.eu

EXECUTIVE SUMMARY

The deliverable D3.3 “EXCESS Flexibility Analytics Module” constitutes a direct outcome of the Task 3.3 “Core ICT platform services” and Task 3.4 “Flexibility analysis and forecasting component”, documenting the activities that have driven the design and development of the EXCESS Data Analytics Framework.

The EXCESS Data Analytics Framework comprises the part of the EXCESS system where the data collected from the demo sites’ buildings of the project by the EXCESS Data Management Platform, are analyzed through specific forecasting and flexibility algorithms in order to produce meaningful results that are digested by the rest of the components of the EXCESS system. Such analytics provide input for the EXCESS Visualizations that offer valuable insights to the end users of the project, while they also enable the operation of the MPC components in the demo sites’ buildings towards the optimal control of the devices and loads and the subsequent achievement of the PEB concept.

This deliverable documents the different components of the EXCESS Data Analytics Framework, namely the Comfort Profiling component, the Demand Forecasting component, the Generation Forecasting component, the Dynamic VPP Configuration component, the Context-Aware Flexibility Profiling and Analytics component, the Flexibility Analytics Visualizations and the Energy Consumptions Visualizations.

The deliverable D3.3 comprises an accompanying document of the first release of the EXCESS Data Analytics Framework and has received input from the deliverable D3.1 “EXCESS ICT Architecture Blueprint”, while it will offer input for the activities of WP4 “PEB implementation and monitoring” towards showcasing the operation of demo sites’ building using the EXCESS system. The feedback obtained from the testing activities of WP4 will be accommodated in the second version of the deliverable D3.3, which will describe the final release of the EXCESS Data Analytics Framework in M42 of the project, including also any necessary enhancements and additional functionalities.

TABLE OF CONTENTS

LIST OF FIGURES

Glossary

Acronym	Full name
AC	Air Conditioner
API	Application Programming Interface
CSV	Comma-separated Values
DoA	Description of Action
DMP	Data Management Platform
DX.Y	Deliverable X.Y
EXCESS	FlEXible user-CEntric Energy poSitive houseS
HVAC	Heating Ventilation & Air Conditioning
ICT	Information, Communication and Technology
JSON	JavaScript Object Notation
LSTM	Long Short-Term Memory
ML	Machine Learning
MPC	Model Predictive Control
PEB(s)	Positive Energy Building(s)
RL	Reinforcement learning
URI	Uniform Resource Identifier
VPP	Virtual Power Plant
WP	Work Package

1 Introduction

1.1 Purpose and scope of the document

The deliverable D3.3 describes the design and implementation of the EXCESS Data Analytics Framework that consists of various components, namely the Comfort Profiling component, the Demand Forecasting component, the Generation Forecasting component, the Dynamic VPP Configuration component, the Context-Aware Flexibility Profiling and Analytics component, the Flexibility Analytics Visualizations and the Energy Consumption Visualizations. The EXCESS Data Analytics Framework accommodates a series of different analytical algorithms that provide input for the visualization dashboards of the EXCESS system, which offer valuable information to the end users of the project, and provide data to the MPC components of the demo sites' buildings for the smooth controlling of their devices and loads in order to achieve the PEB concept.

This deliverable targets at documenting the state of the art in the different areas of analytics that are exploited in the EXCESS Data Analytics Framework and define the functionalities of each component.

Within the Tasks 3.3 and 3.4 where the first release of the EXCESS Data Analytics Framework has been designed and developed, S5 has implemented the Comfort Profiling component, the Flexibility Profiling and Analytics component and the Visualization Dashboards. CGSoft has implemented the Demand Forecasting component, JR the Generation Forecasting component and VITO the Dynamic VPP configuration component. The deliverable D3.3 offers input to WP4 activities for the testing of the EXCESS system in the four demo sites' buildings of the project.

A second version of the deliverable D3.3 will be delivered in M42 of the project, documenting the final release of the EXCESS Data Analytics Framework, where the feedback from the initial operation of the demo sites' buildings will be encapsulated and any updated functionalities will be described.

1.2 Structure of the document

In order to address all the aspects relevant to the scope of T3.3 and T3.4, the present deliverable has been structured as follows:

- Section 1 introduces the work performed and the scope of this deliverable along with the deliverable's structure.
- Section 2 describes an overview of the first release of the EXCESS Data Analytics Framework.
- Section 3 presents the Comfort Profiling component.
- Section 4 presents the Demand Forecasting component.
- Section 5 presents the Generation Forecasting component.
- Section 6 presents the Context-Aware Flexibility Profiling and Analytics component.
- Section 7 prenents the Dynamic VPP Configuration component.
- Section 8 presents the Flexibility Analytics Visualizations for Aggregators.
- Section 9 presents the Energy Consumptions Visualizations for Building Managers.

- Section 10 provides a navigation to the user interfaces of the first release of the EXCESS Data Analytics Framework.
- Section 11 provides the conclusions of the work done within Tasks 3.3 and 3.4.

2 EXCESS Data Analytics Framework Overview

The EXCESS Data Analytics Framework comprises a part of the EXCESS system that enables the analysis of data coming from the EXCESS Data Management Platform in order to facilitate various operations in the EXCESS system towards the achievement of the PEB concept in the demo sites' buildings.

The different components of the EXCESS Data Analytics Framework exploit dedicated algorithms that analyze data residing in the Data Storage component of the EXCESS Data Management Platform, coming from the Distributed Information Systems of the demo sites' buildings. The analysis on these data offers input to the Visualization Dashboards of the EXCESS system (Data Visualizations Framework as sub-part of the overall Data Analytics Framework), which provide valuable information to the aggregators and the building managers of the demo sites. Moreover, the analyzed data that contain flexibility, comfort and forecasting details feed the MPC components of the demo sites' buildings in order to enable the optimal control of devices and loads in the demo sites' buildings and realize the PEB concept.

As presented in the figure below, the EXCESS Data Analytics Framework comprises the Comfort Profiling component, the Demand Forecasting component, the Generation Forecasting component, the Dynamic VPP Configuration component, the Context-Aware Flexibility Profiling and Analytics component and encapsulates the Flexibility Analytics Visualizations and the Energy Consumptions Visualizations.

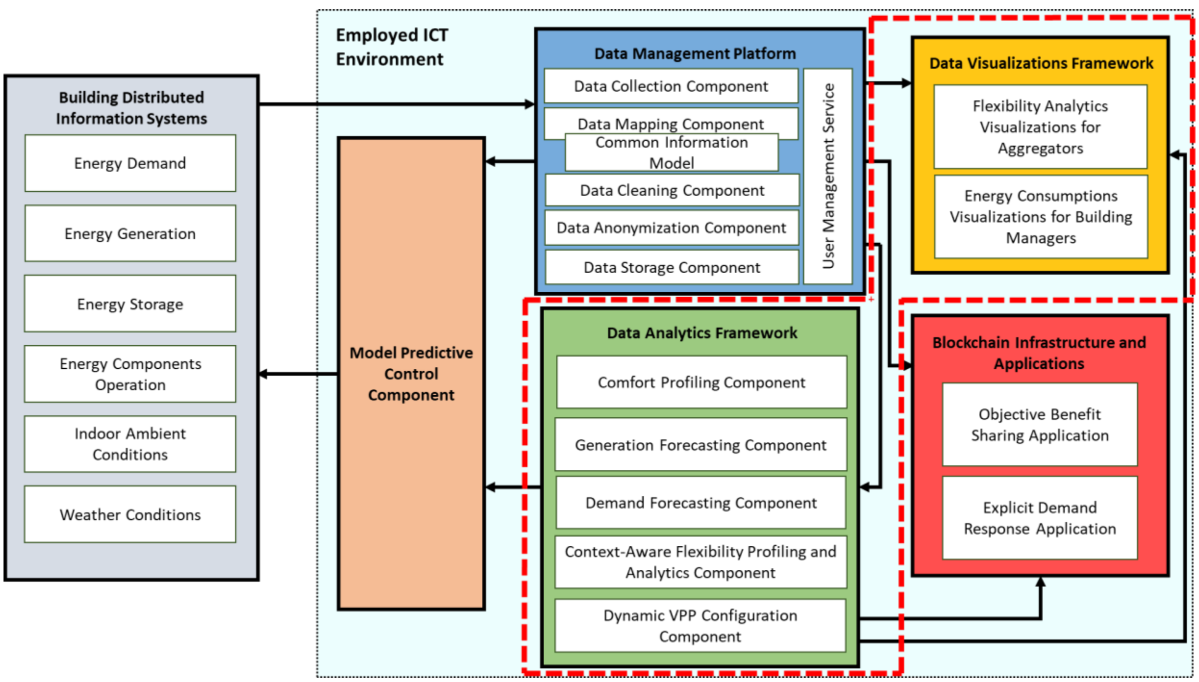

Figure 2-1: EXCESS High-level ICT Architecture (red-dotted area described in current deliverable)

The **Comfort Profiling component** enables through the analysis of data coming from sensors and actuators in the apartments of the demo sites' buildings, the definition of the comfort profiles of the building occupants.

The **Demand Forecasting component** allows the specification of short-term forecasts for the demand of devices and loads of the demo sites' building by analyzing data coming from their operation and the local weather stations near these buildings.

The **Generation Forecasting component** facilitates the specification of short-term forecasts for the generation of energy components of the demo sites' building by analyzing data coming from their operation and the local weather stations near these buildings.

The **Dynamic VPP (Virtual Power Plant) Configuration component** consolidates and analyses the flexibilities offered by the building occupants and allows the aggregators to make the necessary cluster configurations in order to trade these flexibility clusters in the local energy markets.

The **Context-Aware Flexiblity Profiling and Analytics component** enables the definition of flexibility profiles through the combination and analysis of the comfort profiles of building occupants along with the demand and generation forecasts of demo sites' buildings. These flexibility profiles are the input for the operation of the MPC components in the demo sites' buildings.

The **Flexibility Analytics Visualizations for Aggregators** allow the monitoring of flexibility clusters by the aggregators through intuitive dashboards in order to make the necessary configurations for the trading of flexibilities of building occupants to the local energy market.

The **Energy Consumptions Visualizations for Building Managers** facilitate the monitoring of the energy consumption within the demo sites' buildings by building managers through dashboards in order to understand the energy behaviour of the building occupants and discover ways towards energy savings through optimal energy consumption.

The first release of the EXCESS Data Analytics Framework (Visualization Dashboards) is deployed at: **https://dashboards.excess.s5labs.eu/**
(credentials can be provided upon request)

The various components of the EXCESS Data Analytics Framework along with the EXCESS Visualization Dashboards are described in further detail in the following sections of the deliverable.

3 Comfort Profiling component

3.1 Design and functionalities

3.1.1 State of the Art

Human comfort is typically regarded as a condition of mind that expresses satisfaction within an environment. As such, comfort is influenced by several factors, either static (e.g. gender, race) or dynamic (e.g. age, psychology, weather). Research studies have shown that feeling comfortable has a significant impact on peoples' life, their health, well-being and productivity. Moreover, the knowledge of when or in which conditions a human feels satisfied (or dissatisfied) is an important factor in determining a building's energy demand which can lead in an optimal energy control strategy for modern “smart” buildings. For these reasons, the research interest in human comfort within built environments has grown exponentially during the last decade [1].

The most recent trend investigates individual behaviours and tries to predict comfort preferences directly from data collected in their everyday environment in contrast to more generic approaches that base their findings in aggregated responses. The modelling of the individual preferences of building occupants and the identification of their comfort boundaries is called comfort profiling. Such personalized comfort models should be cost-effective, use easily obtainable data and be able to adapt as updated information is introduced [2]. Most of the comfort profiling research is focused on **thermal and visual comfort**, since these two factors can be controlled more effectively and improve energy savings [3].

Indicatively, Zhao et al. study the thermal preferences of employees in an office environment and result in a data-driven personalized and dynamic comfort model, which integrates the temporal dimension as well as the current state of the subjects in the form of voting [4]. A drawback of this approach is that it requires the participation (voting) of the occupants constantly in order to set the comfort setpoint of the installed HVAC system. In a similar fashion, Li et al. manage to successfully apply a Random Forest classifier using human physiological and behavioural data in order to decide the appropriate HVAC control strategy in a building environment [5]. Another approach produces a range of acceptable temperature values by analyzing the time-series data of the indoor operative temperature and the occupants’ feedback coming from daily questionnaires [6], while the authors in [7] extract thermal profiles from the occupants and divide them in three groups depending on the time it takes for each subject to feel discomfort.

For visual dynamic preferences, the literature has less to provide. Typically, lighting profiles are provided as part of the early design stages of a building and are related to natural daylight control strategies [8, 9, 10]. A thorough study for the modelling of non-static lighting preference profiles is provided in [11]. The authors use a clustering technique to group occupants based on their control actions and luminaire output data, as well as feedback from questionnaires and interviews. Malavazos et al., on the other hand, introduce a learning model of user preferences that employes ambient conditions and infers comfort level by the occupants’ control actions (or lack of) [12].

3.1.2 Description of functionalities

The EXCESS comfort profiling component offers a bundle of functionalities that can be used to extract thermal and visual comfort boundaries, either:

a) as personalized aggregated information that averages the subject's behaviour from a temporal perspective (i.e. building a temporal behaviour profile, per time of day, per day of the week, per month or per season), or,
b) as personalized aggregated information that averages the subject's behaviour as a distribution on the temperature and illuminance scale (i.e. building a conditions' based profile)

In order to provide these functionalities as a service, we have implemented a pipeline of various data processing steps which produces the required statistical results, such as:

- Thermal preferences (comfort temperature range) per day of week, per month, per time of day
- Visual preferences (comfort Illumination range) per day of week, per month, per time of day
- Thermal comfort distribution
- Visual comfort distribution

In more detail, the functionalities of the service are as follows:

1. <u>Input data retrieval:</u> In order for the model to run, four types of data are required:
   - historic values of indoor environmental conditions, such as temperature, humidity, illuminance
   - historic log of human actions and events, like turning on/off a HVAC system, switching on/off the lights, setting a temperature setpoint, etc.
   - temporal information for the datetime of the requested forecast (which is directly available)
   - weather forecast in order to extract the required features for the estimations to be made.
2. <u>Input data preprocessing and feature creation:</u> To bring the input data in an appropriate format, i.e. in the format required to extract useful insights, we perform a number of data manipulation steps, indicatively:
   - Using the datetime feature, we produce new temporal/calendar features (such as month, weekday, hour).
   - Using the weather forecast, we extract the required features and format them in respect to measurement types and units.
   - We apply other data pre-processing steps such as normalization on numeric features, filling null values, as needed.
3. <u>Data Analysis:</u> Once the input data have been appropriately processed, we perform their analysis in order to extract useful insights and statistics. An indicative example of the performed computations includes finding the average (also minimum, maximum, median) of the indoor temperature/illuminance/humidity across different times of the day, across seasons and days of the week. Another example involves processing the input data that indicate user actions in order to extract the implicitly provided feedback regarding their comfort.
4. <u>Provision of results through API:</u> The service offers endpoints through which the most important aspects of the analyzed data can be retrieved. For instance, one endpoint returns the aforementioned average illuminance levels during the different hours of the day and this can be filtered according to the season and/or weekday.

3.2 Technologies and tools

The component is implemented in Python and the main libraries that are used are: scikit-learn[1], pandas[2] and NumPy[3].

3.3 Software package repository

The Comfort Profiling component is closed source and no source code is available publicly. The source code and the related deployment instructions are maintained in the related private repositories and the corresponding subcomponents are containerized with Docker.

[1] https://scikit-learn.org/stable/
[2] https://pandas.pydata.org/
[3] https://numpy.org/

4 Demand Forecasting Component

4.1 Design and functionalities (HVAC)

4.1.1 State of the Art

Short-term demand forecasting has become an important tool in reducing the energy consumption of air conditioning systems. Indoor and outdoor environmental conditions, occupancy behavior, room usage, and building physical parameters are some of the factors that introduce randomness, oscillation, and low periodicity [13]. To tackle these problems, academics have investigated data-driven methods for time-series prediction that can be categorized as time-series based, machine learning, and deep learning algorithms. Some indicative approaches are briefly presented below.

Using past week's AC consumption together with weather predictions, temporal features, and PV production measurements, Manivannan et al. [14] achieved better results on predicting the AC load using the Random Forest model compared to the multilayer perceptron (MLP). In another research, the authors implemented an LMA-based (Levenberg–Marquardt Algorithm) artificial neural that produced very promising results compared to other ANNs and multiple linear regression approaches [15]. Deep Neural Networks (DNNs) have grown in popularity recently, as their ability to describe complex non-linear effects has demonstrated increased prediction accuracy and efficiency [16, 17]. Xu et al. [18] demonstrated an attention-based LSTM network outperforming other ML models using AC-generated data.

4.1.2 Description of functionalities

The EXCESS demand forecasting component offers a service that produces hourly AC demand forecasts for a 24 hour ahead horizon. To provide this service, we have implemented a deep neural network for multi-step regression that generates hourly AC energy consumption predictions for the next 24 hours. Specifically, a multi-layer perceptron (ML) with five hidden layers and a mean squared error (MSE) loss function was used. The architecture consists of a decreasing number of neurons in each layer starting from the first layer and the rectified linear activation function (ReLU) function. Also, dropout layers between each hidden layer aid in reducing overfitting caused by the large capacity of the model. The model was trained for 40 epochs on:

- 169-hour (past-week) historical values of AC energy consumption
- Temporal/calendar features (e.g weekday and hour) that correspond to the most recent datetime

The trained model is then used to generate forecasts on an hourly basis. The forecast is offered as a service and can be invoked by the provided endpoint.

In more detail, the functionalities of this service are as follows:

1. <u>Input data retrieval:</u> To run the model, two types of input are required:
    - Historical values of AC demand which are made available through the actual demand measurements
    - Temporal information for the datetime of the most recent input (which is directly available)
2. <u>Input data preprocessing and feature creation:</u> To bring the input data in an appropriate format, i.e. in the format required for the trained model to be applied on, some preprocessing steps are implemented that are applied on input data:

- Temporal/Calendar features are created from the datetime (month, weekday, day of month, hour)
- Lag features are created from the historical AC consumption values
- Numerical features are scaled

3. Model application: Once the input data have been appropriately processed, the trained model is applied to obtain the hourly AC energy consumption forecasts for the next 24 hours.
4. Provision of the forecast through API: The service offers an endpoint through which the 24 values that correspond to the most recently generated forecast can be retrieved. As the service is scheduled to be executed on an hourly basis, the forecast will be updated hourly, thus providing a 24-hour ahead sliding window of expected AC energy demand.

4.2 Design and functionalities (DHW)

4.2.1 State of the Art

Domestic hot water (DHW) heating contributes to a large percentage of the energy use in the building sector. Emerging DHW consumption patterns are often complicated and highly fluctuating. In recent research, machine learning approaches are gaining popularity in predicting the future DHW demand in residential buildings by modelling complex relations between various parameters like building location, usage purpose, occupant behavior, temporal features, etc. [19, 20].

Several forecasting strategies have been investigated, including statistical, time-series-based, machine learning, and deep learning methods. For example, Autoregressive Integrated Moving Average (ARIMA) models trained on a window of historical data are often used for next-day hourly predictions [21, 22]. Furthermore, a support vector machine (SVM) was used to model shower habits using data from 7 residents and the predictions were utilized for the development of an energy-saving control strategy [23]. In addition, artificial neural networks (ANN) have been a popular choice among researchers lately. Trained with historical, temporal, and often meteorological data they manage to capture the uncertainty in both the overall trend, seasonality, and additional observation noises [24, 25, 26].

4.2.2 Description of functionalities

The EXCESS demand forecasting component offers a service that generates hourly DHW demand forecasts for a 24 hour ahead horizon. In order to provide this service, we have implemented a multi-step regressor model based on an artificial neural network that generates hourly hot-water energy consumption predictions for the next 24 hours. Specifically, a multi-layer perceptron (MLP) with two hidden and two dropout layers was trained on:

- 24-hour historical values of DHW consumption
- Temporal/calendar features (e.g. weekday and hour) that correspond to the datetimes for which the forecast is requested
- temperature forecast for the timesteps for which the demand forecast will be generated

The trained model is then used to generate new forecasts on an hourly basis. The forecast is offered as a service and can be invoked by the provided endpoint.

In more detail, the functionalities of this service are as follows:

1. Input data retrieval: In order for the model to run, three types of input are required:

- historic values of DHW demand which are made available through the actual demand measurements
- temporal information for the datetime of the requested forecast (which are directly available)
- weather forecast in order to extract the required temperature features for the model

2. Input data preprocessing and feature creation: To bring the input data in an appropriate format, i.e. in the format required to apply the trained model:
    - Temporal/Calendar features are created from the datetime (month, weekday, hour)
    - Lag features are created from the historical DHW consumption values
    - Temperature forecast features are created based on the retrieved weather forecast data
    - Numeric features are scaled between zero and one
3. Model application: Once the input data have been appropriately processed, the trained model is applied to obtain the hourly demand forecast for the DHW usage in the next 24 hours.
4. Provision of forecast through API: The service offers an endpoint through which the 24 values that correspond to the most recently generated forecast can be retrieved. As the service is scheduled to be executed on an hourly basis, the forecast will be updated hourly, thus providing a 24-hour ahead sliding window of expected DHW demand.

4.3 Design and functionalities (Lights)

4.3.1 State of the Art

Despite the fact that electrical lighting consumption in a building does not have the same impact in energy management strategies as the consumption of a HVAC or a water heating system, it is still of great value, especially in office environments, because it can be easily controlled and the appropriate use of daylight may lead to significant energy savings. To this end, the occupant-centric, short-term forecasting of lighting load can assist in the optimization of smart building energy management systems, while it can provide useful information for network operators and retailers to improve on energy efficiency and minimize costs [27].

Most of the data-driven forecasting methods for energy consumption, utilize statistical algorithms and machine learning models, including support vector machines (SVM), decision trees, and artificial neural networks (ANN), among others [28][29]. More recently, Shan et al. employ a novel ensemble ANN prediction model in order to address linear and nonlinear issues that characterize electricity consumption [30]. Amasyali et al., however, highlight the lack of studies for lighting energy consumption prediction, although it is considered essential for a building's energy efficiency [31], while Runger and Zmeureanu in their thorough study of the field conclude the same thing; lighting forecasting remains a relatively untouched, yet essential part of energy load control and management [32].

In the few dedicated works that are available, office buildings are dominant. Liu and Chen utilize the hourly lightining energy consumption, the solar radiation instensity and the number of people in the office, to feed a support vector regressor and predict next hour's load [33]. The proposed method claims to be better than a simple RBF neural network and demonstrates better generalization capabilities. An SVM-based approach for commercial buildings is also presented in [34], where the daily lighting consumption is estimated based on two features: the daily average sky cover and the day type (i.e. weekday, weekend, holiday, etc). In the rest of the related studies, the task is partially covered by an overall energy consumption model such as the work in [35] and [36]. These approaches, however, fail to capture the impact of lights usage in consumption trends and saving opportunities.

4.3.2 Description of functionalities

The EXCESS demand forecasting component offers a service that is able to make predictions of lighting devices consumption for each of the next 24 hours.

The underlying regressor model that implements this forecast is a trained deep feed-forward multi-output neural network. As far as the network architecture is concerned, the NN consists of 5 dense layers followed by a ReLU activation, with the layer's size decreasing as the depth increases. Due to the network's depth, dropout layers were added between each layer to provide robustness against overfitting. The model was trained for 40 epochs on a training set, every entry of which refers to a specific datetime and consists of the energy consumption measurements of lighting-devices consumption for the duration of the previous week of the corresponding datetime combined with some temporal features providing information about the position of the corresponding datetime on the time grid.

The trained model is then used to generate forecasts on an hourly basis. The forecast is offered as a service and can be invoked by the provided endpoint.

The functionalities offered by the lights demand forecasting service are the following:

1. Input data retrieval: The model needs to be provided with information about
   - the current datetime from which temporal information will be extracted
   - the latest energy consumption measurements for lights usage, i.e. the actual measurements
2. Input data preprocessing - feature creation: To bring the input data in an appropriate format, i.e. in the format required for the trained model to be applied on, some preprocessing steps are implemented that are applied on input data. In this stage, the service:
   - Produces the lag features from the historical consumption data,
   - Produces the temporal features from the datetime feature and
   - Scales the newly created features into the [0,1] interval.
3. Model application: The preprocessed data are fed into the trained model. The model then produces 24 output values that correspond to the forecasts for the next 24 hours.
4. Provision of the forecast through an API: The service offers an endpoint through which the 24 values that correspond to the most recently generated forecast can be retrieved. As the service is scheduled to be executed on an hourly basis, the forecast will be updated hourly, thus providing a 24-hour ahead sliding window of expected lighting-devices demand.

4.4 Technologies and Tools

The Demand Forecasting component is implemented in Python and the main libraries and frameworks that are used are: scikit-learn, TensorFlow[4], Keras[5], pandas and NumPy.

[4] https://www.tensorflow.org/
[5] https://keras.io/

4.5 Software package repository

The Demand Forecasting component is closed source and no source code is available publicly. The source code and the related deployment instructions are maintained in the related private repositories and the corresponding subcomponents are containerized with Docker.

5 Generation Forecasting Component

5.1 Design and functionalities

The objective is to develop a location-independent system that integrates the forecast layer into a MPC (Model Predictive Control) component as flexibly as possible. In the following chapters, the general basics of different approaches in the forecasting system are explained and illustrated using the Austrian demo site as an example.

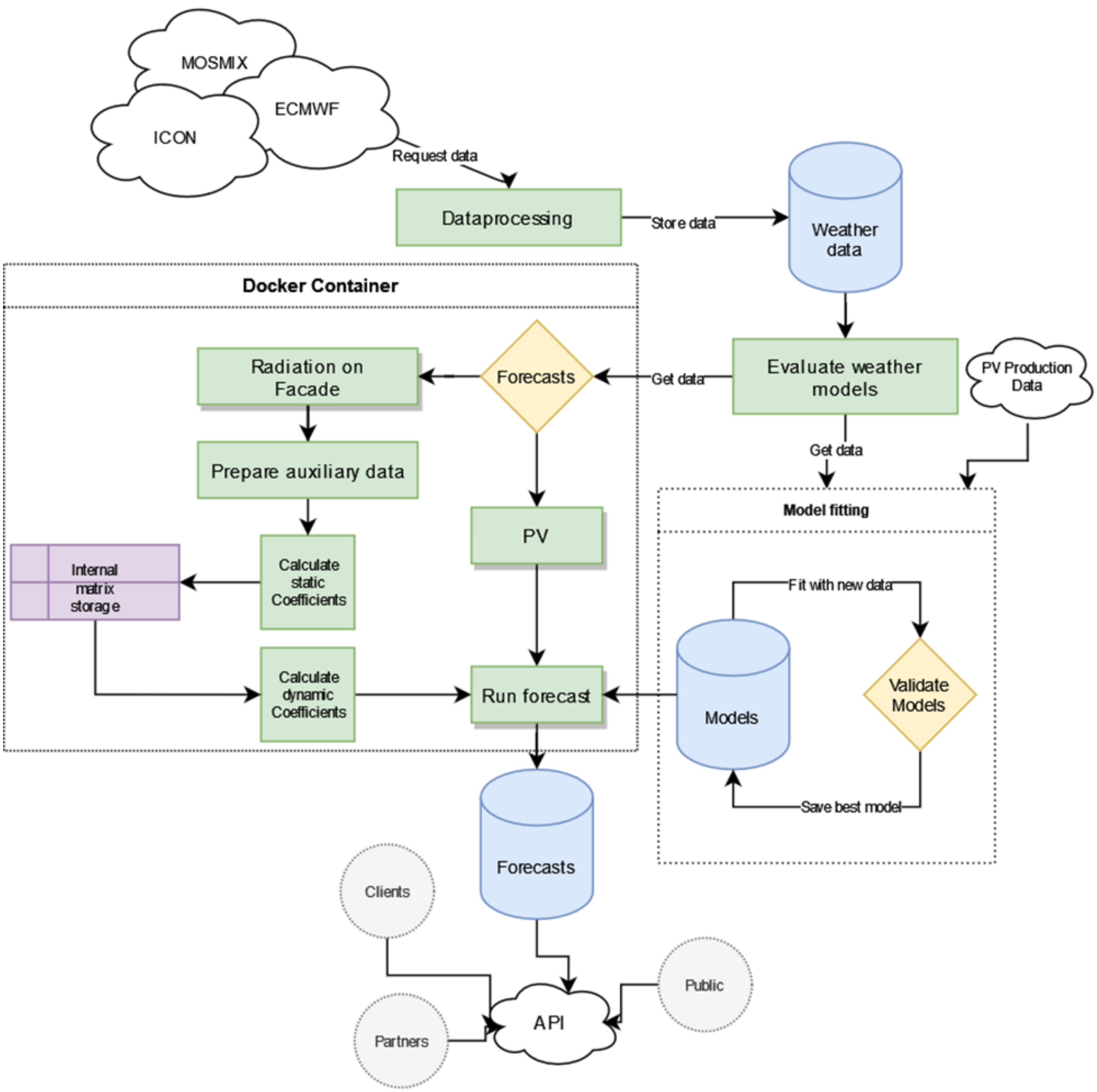

Figure 5-1: Simplified flowchart of the forecast infrastructure. Source: JR-LIFE

Since the basis for all MPC levels is the forecast layer, it is necessary to create these forecasts as precisely as possible. The Forecast Layer consists of weather, load, PV and radiation. The weather module is responsible for forecasting air temperature, cloudiness and radiation. In order to ensure a high quality of the weather forecasts, various established weather models (ICON, GFS, ECMWF, MOSMIX) are compared and statistically processed. Energy profiles for commercial and domestic buildings are estimated for the load forecast. The PV Forecast will use AI and statistical methods to predict the consumption and production of PV energy. Since the solar radiation on the building at a certain position of the sun is a relevant factor for the heating of the facade and the interior, the shading or solar radiation on the building is calculated on a daily and seasonal basis using a digital surface model with a resolution of 1m². This radiation calculation is also based on the weather forecasts and weather variables such as the global horizontal irradiance (GHI). The result is a 2-day forecast of solar radiation in hourly resolution, with the first hours being calculated in 15-minute resolution. These forecasts are then integrated into the other layers or the MPC.

Finally, a flexible ecosystem with open-source applications should be created, which communicates internally and externally via standardized interfaces. The code quality, including documentation and API integration, should help to understand the system itself and make data exchange as easy as possible.

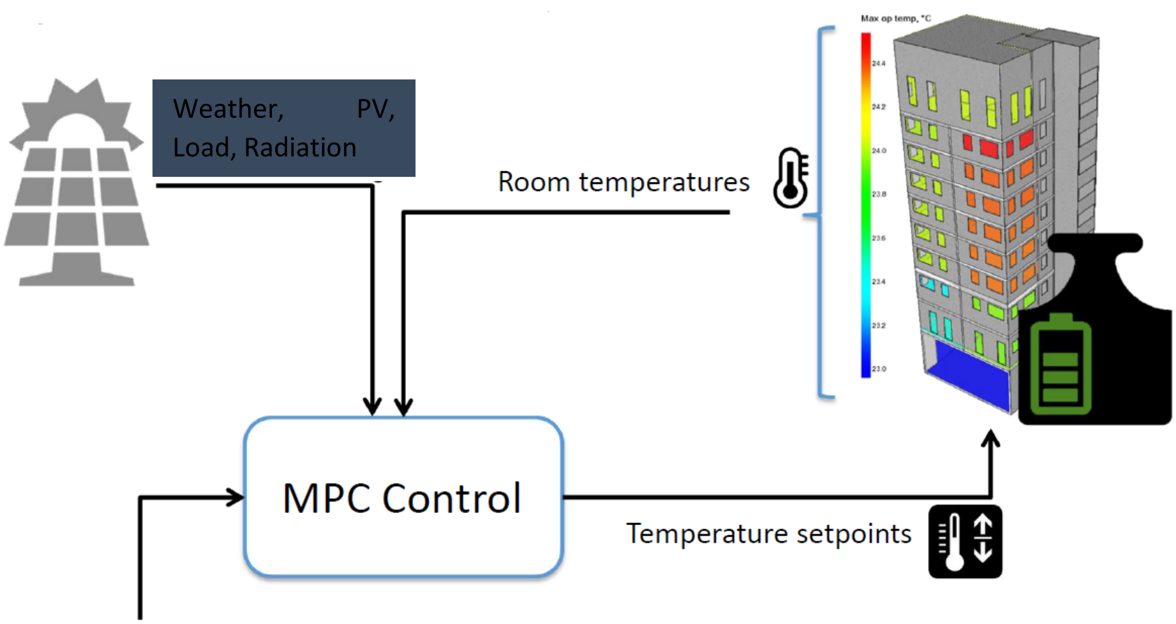

Figure 5-2: Schematic representation of the MPC. Adapted from AEE Intec. Source: JR-LIFE.

5.1.1 Weather Forecast

Precise weather forecasts are essential for MPC calculations. A basic distinction can be made between two types of weather forecasts. On the one hand, there are weather forecasts, mostly prepared by weather services, and, on the other hand, direct raw data from numerical weather prediction models (NWP models). Depending on the application, both types offer certain advantages and disadvantages. If the forecasts are processed by reputable weather services, the forecasts are usually of high quality but also usually associated with costs. "Half-edited" weather forecasts from weather apps are often insufficiently prepared. These are often more of raw data from NWPs, which are only extracted from the location. Another disadvantage of these predictions is availability. As statistical methods are mostly used for processing, historical station data is required. These are also not distributed over a large area, but are usually only available in sufficient quality in cities or larger localities. In the past, the raw data of NWPs was often not freely accessible, which often made it necessary to use weather services. Currently, however, the data from highly established models are freely accessible. The standardization of the data exchange format in recent years was also an important step in order to be able to use this raw data efficiently. The data are usually in GRIB format, but more often in netCDF format. Since weather forecasts are the basis of PV production and other energy related predictions, it is important not to rely on one forecast provider but to evaluate different models and forecasts. The quality and quantity of forecast providers depends on the region of the demo site and is therefore handled differently by the project partners. Because of the unstable air stratification in summer, a radiation prognosis in the Alpine region is much more difficult to accomplish than in more stable regions such as the Mediterranean region. And that's why different models and parameterizations can be better or worse.

Austrian Demo Site

The alpine region has always been a challenge for general and specific weather forecasts. The strong variability of the seasons and the complex topography make weather forecasting a great challenge all year round. Especially in summer, when convection is the main cause of cloud formation, good regional models are of great importance.

The following models are currently available for Central Europe: GFS, WRF, COSMO, ECMWF (MET NO), ICON-D2 and MOSMIX, the latter not being a direct NWP model, but rather data from COSMO and ICON-D2 linked and with a MOS (model output statistics) improved. In addition, it must be noted that the WRF data is not directly available, only the source code of the software. The model is nested with GFS and has to be calculated on its own server. In order to find the most suitable weather model for the project, the aforementioned models were statistically examined and compared.

It turns out that overall, the GFS 0.25 performs worst of all models, especially as far as the GHI is concerned. If R^2 is very good for the temperature at 0.86, the forecast quality for the radiation is less good at 0.60. ICON-D2 and MET NO are better here with 0.65 and 0.73 respectively. MOSMIX currently seems to be the best model with an R^2 of 0.78 for radiation. The mean absolute error is also the lowest at 46.15. MET NO seems with an R^2 of 0.97 to be the best predictor of temperature. All models except MET NO underestimate the radiation on average. The temperature is also systematically underestimated by all models. MET NO shows the strongest underestimation of the temperature. This applies especially to the lows in the morning. The inner-city heat island effect could play a role here, which the models insufficiently recognize. The weather station "University of Graz" is a relatively "warm" station that is strongly influenced by the heat island effect. This may be the reason for the systematic underestimation of the models. Since the EXCESS demo site “tagger area” is more likely to be assigned to the outskirts of the city, the models could provide a more realistic picture here and the bias could be reduced. If the results do not change significantly in the future, MET NO will be used for the temperature forecast and MOSMIX for the radiation forecast.

5.1.2 PV Forecast

An important part of the calculations of the MPC is the optimal allocation of different (electrical) power sources. A focus in this project is the integration of power produced by PV plants on the site. Therefore the MPC needs a forecast of the power produced by PV plants as input. Methods for estimating power produced by PV plants depend on the available data. In this project, we will concentrate on methods that rely on (numerical weather prediction) NWPs as main underlying data for the power forecast. Generally, methods that use NWPs show high ability skill in forecasting power produced by PV plants, and work best for lead times longer than a few hours. For short lead times using online data, can significantly improve the forecast quality, but at this stage of the project it is not planned to integrate this type of models in the algorithm. This means that in this project the task of the PV-Forecasts module is to translate the weather forecasts (radiation and temperature) into forecast of the produced power. For this task, one can broadly distinguish between two different modelling approaches. The first approach can be described as using physical models, meaning that a physical model calculates the power output from the physical properties of the PV plant, such as orientation; shading; efficiency at different light and temperature conditions; characteristics of the used inverter; etc.. The advantage of this method is that one can apply this type of model also without the need of actual measured power production. The disadvantage is that misspecification in the model can lead to a bias in the predicted forecasts, especially if some components of the PV plant change over time (e.g. efficiency of panels or inverters) this can lead to problems when the models are applied over a longer period of time. The second approach is to use black box models that relate the input (weather variables) to the output (produced power), without the aspiration to explain the physical processes involved. Examples of methods that use this approach are statistical methods like regression, or methods from machine learning (ML) like neural networks, support vector machines and random forests. The advantage of this approach is that one does not need the details of the PV plant. The disadvantage is that one needs data for the calibration of the methods. Which means forecasting of the plant can only start after enough data for the calibration was produced. Beside these two main modelling approaches, one can also use so called hybrid models (sometimes also called grey box models). Hybrid models are models, that in cooperate aspects of physical models and black box models, like only using a physical model for some parts of the system and use black models for the rest, or use statistical models that incorporate some of the key physical aspects of the system.

In this project we use a variety of black box respectively hybrid models, from which the best model is chosen during the calibration phase of the modelling. For the computations, we use python. We use the libraries pvlib for physical modelling of the plant and sklearn for ML algorithm. For the statistical oriented models, we use R with the library mgcv and the rpy2 package as interface.

We use a physical model from pvlib that calculates the power output of a physical plant. At the moment we only incorporate the kWp and the orientation of the considered plant and standard values for the rest of the system. Note that especially shading is not incorporated in this model now but might be added later. Further we can use statistical respectively ML models to account for shading. The purpose of the physical model is twofold; first, it can be used as a backup model when a new plant is installed. The second purpose is to use the forecast of the physical model as an input for statistical models respectively models from ML.

From ML algorithm we use random forests as only representative. The reason is that random forests have (for the considered purpose) a similar performance like support vector machines or neural networks, but a smaller computational burden. As input for the random forest, we use azimuth and zenith angle and either direct and indirect radiation, or the forecast of the physical model. While the first model is a classical black box model, the second is a grey box model, where a black box corrects shading and bias.

From the statistical Model we use General Additive Model (GAM) that uses as input the azimuth and zenith angle and the similar to the ML model the purpose of the GAM is to correct for bias and shading. Beside this model, we use also several models that use azimuth and zenith angle, either direct or indirect radiation and temperature as input. These models are again from the GAM class but different model formulas are used, that represent the structure of the PV-system.

5.1.3 Radiation Forecast

To estimate the energy balance in a building, in addition to the exchange of temperature between inside and outside, the direct radiation on the facade is a relevant factor. On the one hand, radiation energy leads to an increase in the temperature of the building substance itself and on the other hand to an increase in the room temperature if radiant energy penetrates the interior through windows. The latter factor plays a much greater role in modern insulated buildings. Even if windows have good insulating properties, there is a certain greenhouse effect, which can have a significant influence on the interior temperature with large window areas. Therefore, this is an important parameter for control units of the buildings MPC, which must be predicted for a certain period of time. Since some controlled units sometimes react slower (e.g. heat pumps), the radiation on the facade must be forecast for a period of up to one day in advance. The forecast variables diffuse horizontal irradiance (DHI), direct normal irradiance (DNI) and global horizontal irradiance (GHI) depend on the position of the sun and the season, the atmosphere, especially clouds and the water vapor content of the air and surrounding objects, especially shadowing and reflection affected. In order to be able to map all these factors accurately and to create a reliable forecast, a system is set up that brings together different existing components and derives precise radiation forecasts with a resolution of less than 1m based on weather forecasts and a digital surface model. This makes it possible to determine the energy in W/m² for each point on a surface at a certain time of day.

The basic structure of the software is written in Python, in particular with the library pvlib and pyrano. Other important components are EnergyPlus and Radiance. Compared to the usual, often commercial software, all components are open source and modern, while frequently used interfaces and exchange formats are used. In addition, instead of 3D models with often proprietary file formats, a surface model is used, which was generated from a LiDAR point cloud and is available in a common raster format (GeoTIFF). All file formats are in plain text and thus ensure maximum transparency. Surface models are currently freely available in many countries and therefore have a clear advantage over 3D building models, since often only the target building and not the surrounding buildings are included in the model.

The simulation of the radiation modeling is based on the "Daylight Coefficient" method (Tragenza and Waters 1983). This method is often used to simulate indoor lighting, but with some modification it can also be used to calculate outdoor solar radiation (Bognar, Loonen, and Hensen 2021) [37]. Since ray tracing is very complex for each time step, this method avoids this and shows good performance even with high temporal resolution. The method is referred to as a 2-phase method by Bognar, Loonen, and Hensen (2021). 2 phases because on the one hand the daylight coefficient vector is calculated for each time step and on the other hand the luminance of the sky is calculated for each sky segment.

$$radiation\left(\frac{W}{m^2}\right) = DaylightCoeff * SkyLuminance \quad (1)$$

Daylight coefficients are calculated numerically using DAYSIM. The radiation is discretized based on partial segments of the sky (α) with a certain luminance (E_α). Each sub-segment is created by the multiplication of daylight coefficients (DC_α), luminance (L_α) and angular size (S_α) of each α^{th} sky segment. The sum of the radiation of all sky segments (ΔE_α) finally represents the incoming radiation at the viewer.

$$\Delta E_\alpha = DC_\alpha * L_\alpha * \Delta S_\alpha \tag{2}$$

This model was modified by Bognar, Loonen, and Hensen (2021) [37] to use LiDAR point clouds as input data. LiDAR (Light Detection and Ranging) is an optical remote sensing technique that uses laser light to scan the Earth's surface densely, obtaining highly accurate X, Y, and Z values. The point clouds determined in this way are post-processed after the LiDAR data acquisition and integrated into a highly accurate georeferenced XYZ coordinate system by analyzing the laser time period, laser scanning angle, GPS position and INS information. Remote sensing products such as digital surface models (DSM) are then created from these point clouds. The big advantage compared to pure 3D models is the mostly open data format (TIFF), the easy availability and integration. The disadvantages are individual inaccuracies in the 3D objects and the need for manual adaptation when modifying existing structures. In the point cloud-based method, the daylight coefficient (DC_α) of the 2-phase method is calculated using 3 factors: 1) calculation of the flux-transfer coefficient of an empty scene, 2) calculation of the shading degree of each sky segment, 3) calculation of the diffuse reflected radiation of the neighboring objects. The radiation for each sensor point (E_{SP}) can be calculated as follows:

$$E_{SP} = \sum_{\alpha=0}^{n} \left(E_{cos,\alpha} * (1 - CR_\alpha) + E_{r,\alpha}\right) * L_\alpha \tag{3}$$

n represents the number of sky segments. To ensure acceptably precise calculation, 2307 sky segments were generated. The sky segments can be thought of as looking at the zenith from a point on the surface.

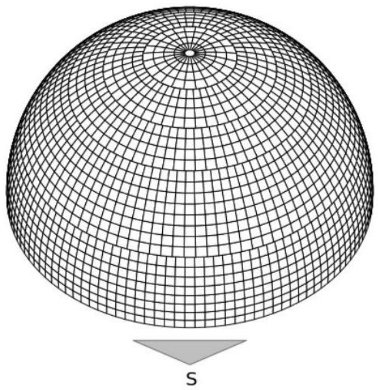

Figure 5-3: Visualization of the 2307 sky segments. Source: JR-LIFE.

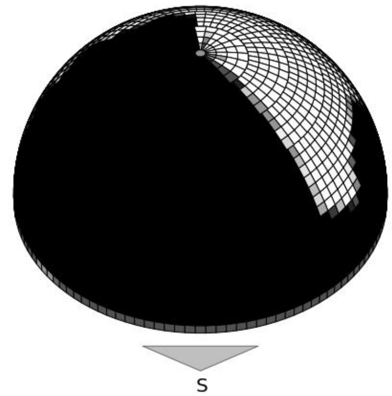

Figure 5-4: Display of the cover ratio for a sensor point on the east facade. The shading in the southern area is caused by the stairwell. Shadowing by a building can be seen in the east. Source: JR-LIFE.

$E_{cos,\alpha}$ represents the flux-transfer coefficient for an empty scene with no topography or shading objects. CR_{α} is the degree of shading for a specific sensor point. Completely shaded sky segments have the value 1 and non-shaded segments have the value 0. Since the shading can only affect a part of a segment, there are also values between 0 and 1. With 50% shading of the segment it would therefore have the value 0.5. The larger the segments, the more imprecise the calculation becomes. Therefore, care must be taken to calculate with at least 577 segments. As already mentioned, 2307 sky segments are simulated for the forecasts. $E_{r,\alpha}$ represents the reflection of neighboring objects as a flux-transfer coefficient. The luminance for a certain time of year and day is expressed in terms of L_{α}. Figure 5-5 illustrates the situation of building 10 in the Austrian demo site, based on a sunny and cloudy day.

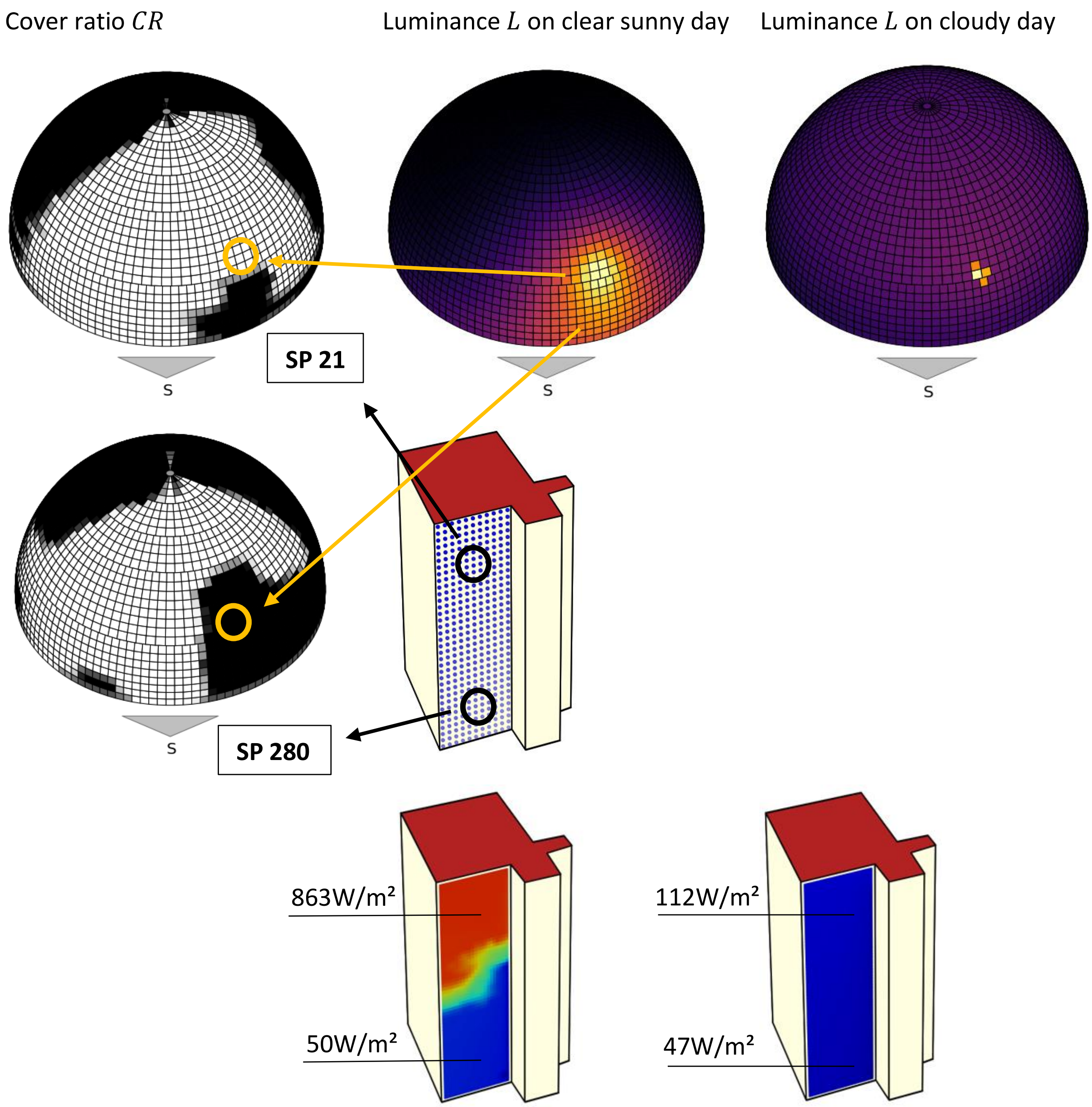

Figure 5-5: Coverage and luminance illustrated by sky segments. Source: JR-LIFE

The above figure presents the south side of building 10 and a sunny and cloudy day on 1994-02-21 and 1994-02-13 at 10:00 UTC, respectively. The yellow arrows from the Luminance Skymatrix to the Cover Ratio Skymatrix show you the angle at which the sun stands on a bright sunny day in relation to the shading objects. For the sensor point (SP) 280 you can see that this is shaded with 50W/m² at 10:00, whereby sensor point 21, in the higher area of the building, is exposed to direct solar radiation with 863W/m².

Austrian Demo Site

The Tagger area with building 10 in Puchstraße in Graz is a building complex of several former old factory buildings. Building 10 stands northwest of the building complex and is surrounded by taller buildings to the south and east. When the sun is low, these buildings cause certain shading of the southern and eastern facades, especially from winter to spring. Since building 10 is to be converted into a student residence and windows are being installed in the facade, it is of particular interest to estimate the radiation on these windows.

In the simulation, the coefficients previously calculated in preprocessing are multiplied by the luminosity L. External programs are mostly used for this. In the first step, a sky matrix is created with *gendaymtx*. The basis for this is the Reinhart Patch division. As already mentioned, 2307 sky segments are created. *Gendaymtx* works with *.wea* input files, a program-internal weather data format. Since this format is hardly used in common software, a file converter was implemented, which converts Python *pandas.DataFrames* into *.wea* files. L is calculated by *gendaymtx* for each time step. Finally, with *dctimestep*, the coefficients are multiplied by the matrix created by *gendaymtx* and post-processed with rmtxop.

MOSMIX and ICON-D2 data from the DWD open data portal are used as weather input data, whereby these are available for the variables global horizontal irradiance (GHI), direct normal irradiance (DNI) and diffuse horizontal irradiance (DHI) in 15-minute time steps. For ICON-D2 raw data it should be mentioned that the radiation data does not represent the average of the time step as usual, but the average since the start of the run. The result of the simulation is the radiation in W/m² for each sensor point, whereby the result is linearly interpolated on the area to get a better visualization. Compared to preprocessing, the runtime of the simulation is only marginal.

Python is a highly flexible programming language that is widely used in particular in the natural sciences but also in engineering. More and more open-source solutions are offered by software vendors and universities, like pyrano and pvlib with which a large part of the radiation simulation was implemented. The documentation of both libraries is excellent and using them with certain programming skills is relatively easy. It has also been shown that the use of a digital surface model enables the radiation to be calculated precisely. In addition to the point-based method, Pyrano also offers a method using a pure 3D model. This is useful if no DSM but a 3D model is available. The accuracy can be described as good in a first visual analysis. The calculations are also performant in terms of runtime, although unfortunately multiprocessing is not supported for the Pyrano library. It became clear that the number of sensor points and the number of sky segments play an important role in performance. On the one hand, the spatial accuracy decreases with a low number of segments and sensor points, and on the other hand, there are significant performance losses if the resolution is too high. A sensor density of 1sp/m² and a number of sky segments of 2307 is therefore ideal for most applications. However, the need for higher computing resources only arises when calculating the static coefficients, which only has to be carried out once. The dynamic component shows a high performance. This is especially important for on-demand predictions. The system was tested under Windows10 and under Linux with the operating system Ubuntu 20, whereby the use of Linux is preferable.

5.2 Technologies and Tools

A list of the technologies used for the development of the Generation Forecasting component is following:

- Programming languages: Python 3.8.5, R 4.1.2
- Environment: Docker container with Linux Ubuntu20
- Monitoring System: icinga2
- Database: postgreSQL
- Database Client: pgModeler
- API: Gunicorn as HTTP server and flask as middleware, nginx as proxy
- General important python libraries used
  logging (Logging system), subprocess (calling external processes), multiprocessing (parallelization), sqlalchemy or psycopg2 (database interaction), xarray (Handling netCDF files and multidimensional datasets), pandas (Handling data tables), numpy, dynaconf (Config handler), scipy
- Weather forecasts
  Main Python libraries: pvlib, dwdGribExtractor, wetterdienst
- Radiaton forecast
  3D Models: EnergyPlus
  3D Model Visualization and GUI: SketchUp 2017 with Euclide extension
  Geographic information system to handle georeferenced data like digital surface models: QGIS 3.22.1
  Main Python libraries: pvlib, pyrano, geomeppy, eppy, mpl_toolkits, osgeo (Geographical files handler)
  External command line programs: radiance-online
- PV forecast
  Main R libraries: solaR, mgcv, suncalc, lubridate, matrixStats
  Main Python libraries: pvlib, rpy2 (R <-> Python interface), sklearn (ml models), dill (object loader)

5.3 Software package repository

In order to exchange data such as PV, load and radiation forecasts among the various project partners in a standardized manner, a suitable interface was implemented that corresponds to all current standards of an API (Application Programming Interface). The API is based on Representational State Transfer (REST), a paradigm of the software architecture of distributed systems, especially for web services. This ensures that data can be exchanged via standardized queries (GET, POST). This means that every project partner has a flexible way of accessing the data generated by JOANNEUM RESEARCH. It does not matter which interface or which programming language is used. The main communication takes place via the internet and most programming languages support libraries which simplify the queries. The paradigm requires that all information an application needs to restore the page state be included in the request. The URI identifies the resource, while the HTTP header can contain information such as the type of access (GET, POST), return format or authentication. To ensure a high level of security, the JR API only uses GET requests and HTTPS encryption. In addition, queries can only be carried out by authenticated users who have to generate a token using their email address and a password. The token is valid for 60 minutes. When this time expires, the token must be requested again. The credentials must be requested from JR and cannot be modified. Via the API, all data required by other project partners is made available in the JSON format.

The whole System is implemented in Python and uses *Gunicorn*[6] as the webserver with the *Flask*[7] web framework. Gunicorn is a Python Web Server Gateway Interface (WSGI) HTTP server. It is a pre-fork worker model, ported from Ruby's Unicorn project. The Gunicorn server is broadly compatible with a number of web frameworks, simply implemented, light on server resources and fairly fast. The API can be called with the domain https://api.jr-excess.at and is illustrated in the following chapter with a Python example. More detailed documentation can be found at https://api.jr-excess.at.

Libraries

```
import requests
```

Vars

```
host = 'https://api.jr-excess.at'
```

Authentication

```
headers = { "x-access-token": token }
req = requests.get(f'{host}/login', auth=('<email>', '<password>'))
token = req.json()["token"]
```

Radiation Forecast

Get surfaces (wall_id, …)

```
requests.get(f'{host}/radiation/walls', headers = headers)
```

Get sensors (sensor_id, …)

```
params = {
    "wall_id": 2
}
requests.get(f'{host}/radiation/sensors', data = params, headers =
headers)
```

Radiation forecast for specific sensor IDs

```
params = {
    'sensor_ids': [24, 25, 26],
    'time_start': '2021-02-13 09:00:00',
    'time_end': '2021-02-13 14:00:00'
}
requests.get(f'{host}/radiation/forecasts', data = params, headers =
headers)
```

PV Forecast

Get forecast runs

[6] https://gunicorn.org/
[7] https://flask.palletsprojects.com/en/2.0.x/

```
requests.get(f'{host}/pv/runs', headers = headers)
```

The query can be limited in time. This means that only runs for a certain period of time are displayed with {'later_than': 'last'} being the last run.

```
params = {'later_than': '2022-01-03 00:00:00'}
requests.get(f'{host}/pv/runs', data = params, headers = headers)
```

The forecast can now be queried via the run ID. Example for the prediction of the last run:

```
x = requests.get(f'{host}/pv/runs', data = params, headers = headers)
js = x.json()
latest_run_id = dict(js[0])["id"]

params = {
    'runid': latest_run_id
}

requests.get(f'{host}/pv/forecasts', data = params, headers = headers)
```

6 Context-Aware Flexibility Profiling and Analytics Component

6.1 Design and functionalities (HVAC)

6.1.1 State of the Art

Flexibility is defined as the modification of the energy consumption patterns in reaction to an external signal [38]. The estimation of the flexibility that can be offered to the grid is useful as it can be used to balance the electricity grid in a cost-efficient way, avoiding significant investments in new power plants and transmission lines. Demand response programs aim to achieve grid balancing by exploiting the flexibility that consumers can offer. Residential buildings are responsible for a significant amount of energy demand [39] and air conditioners in particular are responsible for an average of 45% of domestic electricity consumption [40]. Thus, the estimation of the AC flexibility is important for planning more targeted and efficient demand response programs.

To quantify the flexibility that consumers can offer, most approaches focus on the formulation of one or more models that aim to describe the flexibility of appliances, the thermal characteristics, and the energy needs of the building under study. Chen, Y., et.al., (2019) [41] formulated models to estimate the potential flexibility that different resources can offer (e.g., thermal mass, appliances, HVAC system, water tank). Alic, O., & Filik, Ü.B. (2020) [39] built a dynamic model that expresses the correlation among the power consumption, temperature, and time of ACs activations. The scope was to estimate the AC energy reduction, by selecting lower desired indoor temperatures during high pricing periods. The authors of (Che, Y., et.al., 2019) [42] proposed an electric model to describe the operation of an AC and estimate its consumption. They suggested an approach to reduce the energy demand and at the same time maintain the indoor temperature stable, without compromising resident's comfort.

6.1.2 Description of functionalities

The EXCESS context-aware flexibility profiling and analytics component offers a service that estimates the AC flexibility for a specific duration and AC settings (desired indoor temperatures).

The service builds upon the services offered by the thermal comfort profiling component and the HVAC demand forecasting component, from which it receives information along the following axes: (a) the thermal conditions at which the user in the given context feels most comfortable and the boundaries (in terms of temperature) inside which he retains an adequate thermal comfort level even if not the optimal one and (b) the HVAC demand forecasts.

The component further implements and makes use of 2 models: (a) a thermal model, implemented using a linear regressor, that computes the change in indoor temperature depending on the AC status and the the environmental conditions and (b) an AC state predictor, implemented using a random forest classifier, that based on indoor temperature changes and the difference of indoor temperature to the desired one predicts the AC activation states.

Leveraging the above models, the AC consumption for a specific duration and desired temperature is estimated based on the AC activation states. The consumption difference between 2 unique temperatures (which are extracted from the thermal comfort profiles) equals the flexibility. Both models are trained on features related to AC historical consumption data, environmental and weather data.

Following, the trained models are employed to estimate the flexibility based on new input data.

In more detail, in order to provide the flexibility forecasts, the component implements the functionalities presented below:

1. Input data retrieval: In order for the model to run, the following types of input are required:
   - Historical AC consumption data
   - Historical indoor environmental data
   - Weather forecasts

2. Input data preprocessing and feature creation. To bring the input data in an appropriate format, i.e. in the format required for the trained models to be applied, we perform the following steps as part of an automated data preprocessing pipeline:
   - Resample the dataset (change data resolution) if needed in order to have the same frequency as the one used during model training
   - Select the features related to the task and create new ones as needed
   - Normalize features, where applicable

3. Model application: Once the input data have been appropriately processed, the trained models are applied as follows:
   a. Information from the thermal comfort profiles is leveraged to define the temperatures that are of interest in the specific case.
   b. The input features are fed to the thermal model which outputs an estimation of the indoor temperature change for the next time interval.
   c. Following, the state predictor estimates the next AC activation state based on the thermal model output and the desired indoor temperature.
   d. The AC consumption is computed using the predicted AC activation states (which are computed by applying the first two steps as many times as needed to get the results for the desired time interval).
   e. The flexibility is calculated as the difference of the AC consumption estimated for 2 different desired indoor temperatures

4. Provision of results through API: The service offers an endpoint through which the flexibility forecasts per time step for the specified time interval can be retrieved.

6.2 Technologies and Tools

The component is implemented in Python and the main libraries that are used are: scikit-learn, pandas and NumPy.

6.3 Software package repository

The Context-Aware Flexibility Profiling and Analytics component is closed source and no source code is available publicly. The source code and the related deployment instructions are maintained in the related private repositories and the corresponding subcomponents are containerized with Docker.

7 Dynamic VPP Configuration Component

7.1 Design and functionalities

VITO develops a Dynamic VPP component that aggregates consumption and generation of multiple buildings and VPP shared assets (like shared generation or storage). This component manages a collection of buildings and shared assets, meaning that it creates aggregated consumption/injection plans that can be communicated with grid or market stakeholders. It as well characterizes the collective flexibility as well as the flexibility of each individual building and shared asset. In a first instance, this flexibility characterization is used to create an optimal consumption/injection plan by the optimal activation of flexibility in each of the buildings and shared assets. The 'remaining' flexibility is then available to ensure that the VPP as a whole sticks to the agreed/committed consumption/injection plan. Deviations that occur e.g. because of incorrect forecasts, will be self-corrected (auto-balanced) to not cause or be subjected to imbalance penalties.

As extended functionality, an additional Flex Trading step can be added between the consumption/injection plan optimization, and the self-correction/auto-balancing. Thus, after the determination of the optimal VPP consumption/injection plan (and the disaggregation there-of in optimal building consumption/injection plans), flex services can be offered with the 'remaining flexibility', and if offers are accepted, plans can be adopted accordingly and communicated to relevant grid and market stakeholders, and the 'remaining' flexibility will be used for VPP self-correction (auto-balancing) of deviations in relation to the committed flex offering and the communicated injection/consumption plans.

In contrast to (most traditional) consumption/injection plan forecast creation approaches that leverage AI and ML (like RL), our Dynamic VPP component employs a Model-Predictive approach. This means that although ML/AI is used to create models and forecastst, the optimal plans themselves are created by means of a Model Predictive Optimisation that uses these models and forecasts. For small VPPs this can be done by solving an optimization problem for the collection of all buildings and all assets. But it is also possible to solve the optimization problem for each individual building and shared asset, and then aggregate these plans. The provided solution can be used in a hierarchical manner, and therefore scaled well for large VPPs as well. An important advantage of this approach is that it does not require a – lengthy – training period, and does not need training data other than for the model and forecast creation. Also, in contrast with traditional ML based approaches that assume a fixed set of buildings and VPP assets in the training and operational phase, our hierarchical model-predicitive approach can deal with a time-varying collection of buildings and shared assets, and can deal with time-varying preference these may have. That's why it is called a DYNAMIC VPP (aka Dynamic Coalition Manager (FHP[8])). Hence buildings may decide at any time (only constrained by the 'closure times' of committing flex service activation and consumption/injection plans to enter or leave the VPP, and changes its contribution/commitment (e.g. offered flex) to the VPP.

The above described model-predicitive optimization framework that is used in the Dynamic VPP, is used as well in a Flexibility Analysis tool, called ABEPEM (Active Building Energy Performance Modelling tool: AmBIENCe[9]), to analyse the impact of a proposed composition or proposed change to the VPP e.g. adding or changing buildings characteristics, adding or changing building assets, adding or changing shared

[8] http://fhp-h2020.eu/

[9] http://ambience-project.eu

assets. Besides the impact of these VPP design changes (that impact the models that are used in the model-predicitive optimization), also the impact of different contextual scenarios (like climate or wheather conditions and profiles, tariff structures and price profiles, user preferences and setpoints, etc.) can be analysed. This can be used to compare multiple considered VPP design options, taking into account multiple contextual scenarios, and analyze the sensitivity of the forecasted VPP performance to scenario changes.

The VPP performance is forecasted by means of a model-predictive optimization that also will be used for the VPP during its operation. Hence it implicitly models the impact of an energy management system that takes optimal flex activation decisions based on models, preferences and user setpoints, and forecasts. And in the VPP analysis and design phase, this model predictive optimization is fed by scenarios for contextual information (e.g. related to climate/weather profiles, price profiles, etc.) that mimic forecasts in the operational phase.

Using this approach, the impact of specific VPP design decisions and selected scenarios on the energy consumption/injection, but also carbon emissions and energy costs can be analysed to decide on an optimal VPP composition and design. Specifically for the carbon emission quantification, this approach can take into account grid carbon intensity scenarios that account for assumed reductions over time – as the energy mix becomes greener year after year – but also (evolution of) intra-day carbon variations that can be optimally leveraged by the active control of flexibility.

Specifically, the energy cost profile output of such a VPP analysis for a specific considered VPP composition and design can be combined with investment and other operational costs associated with the considered design to perform a financial and bankability analysis.

As an example, this VPP configuration analysis will be used in the Belgian EXCESS pilot to analyse the impact of adding rooftop PV and PV-T on the carbon emissions and energy consumption cost of the (virtual) community of 20 social housing apartments and shared facilities. Also, the specific impact of smart coordinared control of the collection of electric booster heaters in the apartment substations will be analyzed.

7.2 Technologies and Tools

The Dynamic VPP Configuration and Analysis component is leveraging model-predictive optimization technology that determines an optimal consumption plan (NOT control commands) for controllable assets (associated with buildings or shared to the VPP) [43]. The actual optimisation engine is integrated in a modular and flexible optimisation framework that makes it possible to describe the specific VPP composition in a JSON format. This JSON description is then automatically parsed and combined with other relevant information like the optimization objective to generate the mathematical formulation that feeds the optimisation core algorithm.

The JSON format description supports the definition of multi-energy multi-collector configuration. I.e. it can combine multiple energy carriers (electricity, heat, gas) as well as multiple collectors for a specific energy carriers (e.g a collector per electricity phase, or a collector for space heating water and a collector for sanitary hot water production). Assets can be connected to/between collectors. This can be non-controllable consumption assets, controllable consumption assets (like smart white-goods, or building space heating or sanitary hot water buffers or EVs), generation assets like rooftop PV, conversion assets like heat-pumps (electricity to heat) or CHPs (gas to heat and electricity). And buffer assets (batteries, space heating buffers).

Depending on the asset, it can have associated with it scenarios (e.g. non-controllable consumption, solar irradiation, outdoor T, etc), behaviour/state model (e.g. dynamic thermal behaviour of a building), and constraints related to state (e.g. max capacity of a battery) or usages (e.g. max charging power of a battery, max power of a heat-pump). Also collectors can have constraints associated to them (e.g. grid connection capacity). On each of the connections between assets and collectors, and of the connection of collectors to the external world\grid, costs or prices can be allocated, for example, related to offtake from or injection to the main grid, or heat produced by the CHP, or electricity produced by the PV (e.g. they may be leased and therefore the produced electricity is not free), etc. All of this is taken into consideration for doing the model-predictive optimisation that quantifies optimal the consumption/injection (kWh profile), energy cost (€) and emissions (g CO_2) of each building/asset as well as the VPP as a whole.

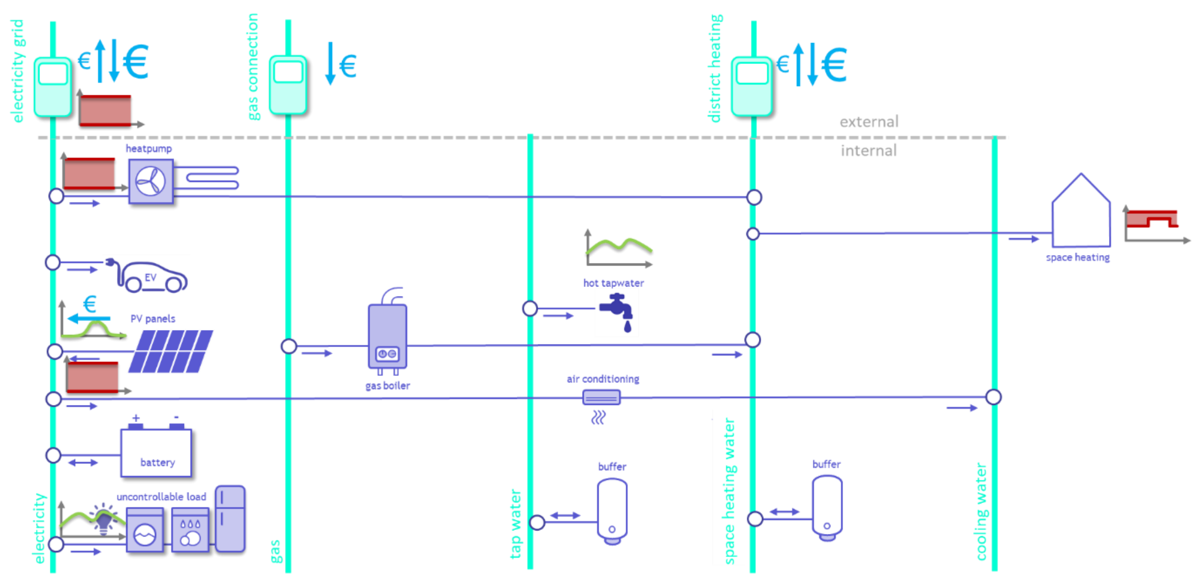

Figure 7-1: Configuration example of a multi-energy multi-collector configuration for the model-predictive optimisation framework

This configuration functionality as well as the model-predictive multi-energy optimisation framework is not only available in the VPP analysis component, but is also deployed in the VPP operational planning and control engine. It has been developed and used initially for Building Energy Management Systems, but due to its modularity and flexibility, can be deployed for clusters as buildings and VPPs as well. To further improve its scalability, extra functionality is being added to create and use more abstract flexibility representations (called Flex Graphs) and associated Flex Graph aggregation and Aggregated Optimal Plan disaggregation functionality.

Additionally to this multi-energy optimisation framework and associated multi-energy multi-collector configuration tool, a Grey-Box model creation flow and tool is available to create building thermal dynamic models that are used by the model-predictive optimisation framework [44]. The developed flow creates such Grey-Box models from 'measurements' that can be either actual physical measurements from buildings, but that can well be virtual measurements taken from Digital Twin models of buildings, which is very useful when no actual physical measurements are available. Either because no measurements were/are done, or because they can not be done because the building is not yet available (e.g. considered new-built, or considered renovation and one wants to account for and analyse the impact of the proposed renovation). A flow and methodology is available in the latter case to automatically create simple shoe-box Digital Twin models (one zone per floor) in Modelica from a standardized CSV description containing geometric information and thermal characteriscs information from used materials for floors, walls and windows. This flow will be further improved (in other projects) to enable the creation of more sophisticated Modelic models starting from Digital Building BIM information based on the IFC or gbXML standard formats.

7.3 Software package repository

The described technologies and tools are IP protected: they build on back-ground IP developed in many past projects, which are enriched and improved to accommodate EXCESS specific requirements and learnings.

The described functionality is available as a set of independent tools that run on VITO servers, and they will be used for the analysis of the Belgian Cordium pilot. The creation of web-service UI/Front-end is considered, so that it can be more easily used by non-experts to enter a considered VPP configuration (as a collection of Positive Energy Buildings) as well as associated building models and scenarios they would want to analyse. After providing this information to the UI/Front-end, the model-predictive optimization would be started to produce results in terms of energy consumption, energy cost and energy consumption related emissions.

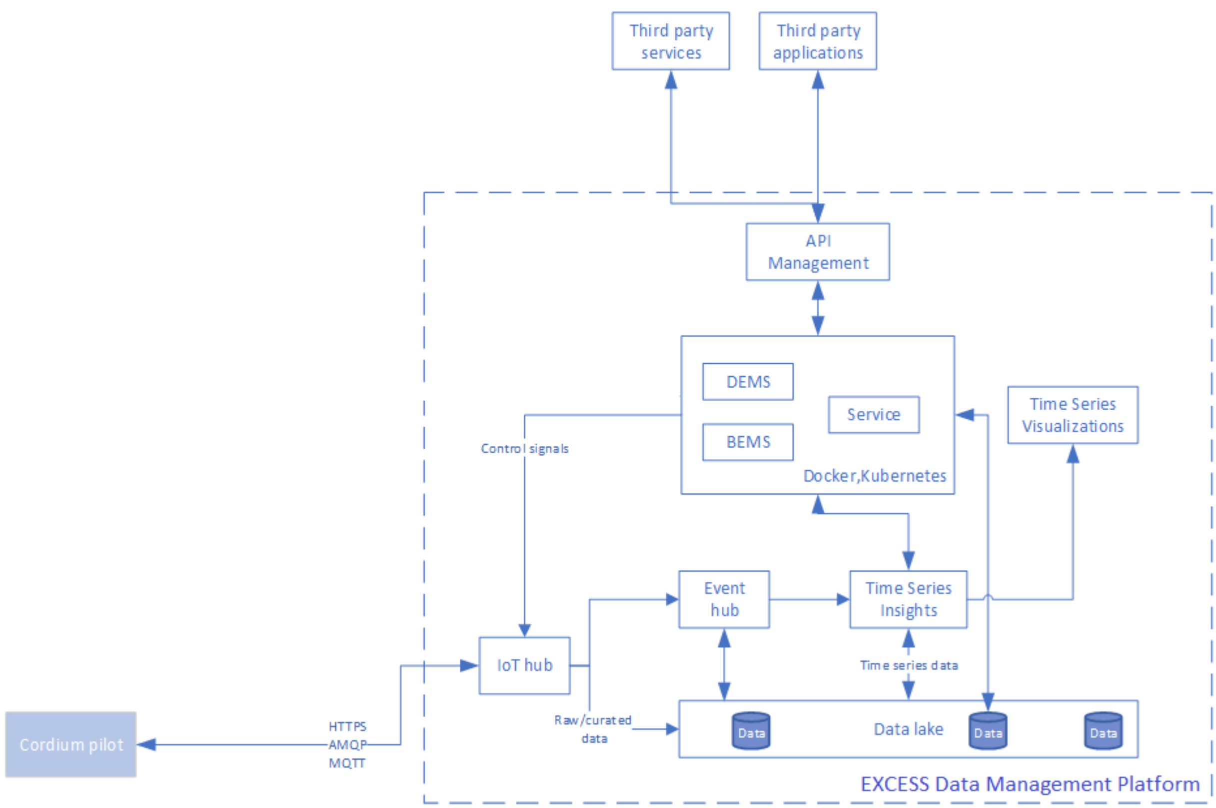

Figure 7-2: High-level Dynamic VPP Configuration component architecture

The VPP operational planning and control software (aka DEMS or DCM) is based on the same configuration functionality and optimisation framework that runs as well on a VITO server, and will be used for the optimal smart control of the electric booster heaters in the apartment substations. To feed the forecasters and models, and regulary update/recalibrate status information that is relevant for the optimization process, it will connect to the apartments through the EXCESS Data Management Platform that collects the necessary data from a locally installed PLC system. A high-level overview of this architecture is presented in the above Figure 7-2. The interactions between the dynamic VPP configuration component and related tools and components is given in the figure below.

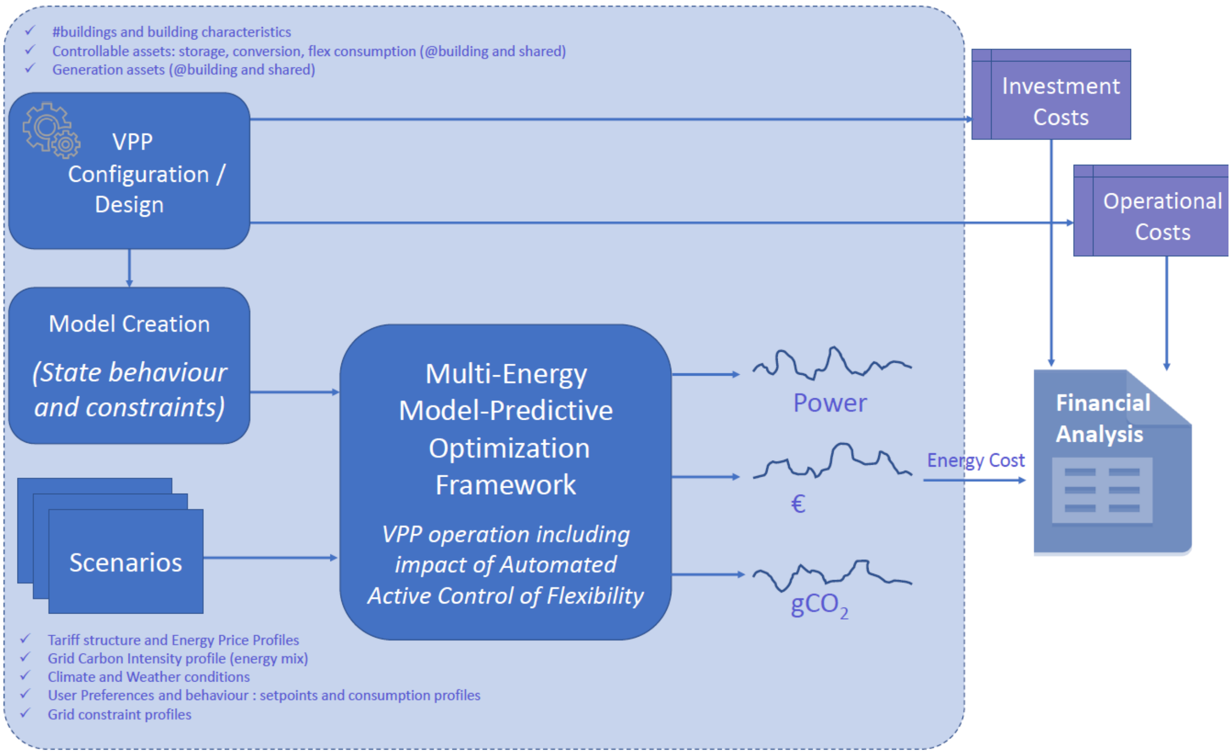

Figure 7-3: Interactions between the different components and tools

8 Flexibility Analytics Visualizations for Aggregators

8.1 Design and functionalities

The Flexibility Analytics Visualizations component enables the aggregators to monitor continuously the flexibilities that are becoming available by the building occupants in the demo site building by offering intuitive dashboards that show the historical details but also short-term flexibility forecasts. The aggregators can select specific periods of time where they can view the past flexibilities of building occupants (in an anonymized way respecting their privacy) and understand their energy behaviour. Moreover, the additional flexibility forecasts give the opportunity to the aggregators to realize which building occupants can become available with flexibilities during the next 24 hours in order to participate in VPP clusters that can be traded in the local energy markets. The Flexibility Analytics Visualizations component comprises a valuable tool for the operations of the aggregators and the potential subsequent monetary gains of building occupants through flexibility training.

A navigation to the Flexibility Analytics Visualizations component is presented in section 10.

8.2 Technologies and Tools

The Flexibility Analytics Visualizations component is written in Python. For the user interface implementation, Vue.js[10] has been exploited, while for the backend implementation Django framework[11] has been used.

8.3 Software package repository

The Flexibility Analytics Visualizations component is closed source and no source code is available publicly. The source code and the related deployment instructions are maintained in the related private repositories and the corresponding subcomponents are containerized with Docker.

[10] https://vuejs.org/
[11] https://www.django-rest-framework.org/

9 Energy Consumptions Visualizations for Building Managers and Occupants

9.1 Design and functionalities

The Energy Consumptions Visualizations component allows the building managers to monitor the energy consumption within their demo sites' buildings through self-descriptive dashboards and thus, assist them in understanding the energy behaviour of the building occupants (in an anonymized way respecting their privacy). The building managers can select specific time periods to view the energy consumption of specific apartments or devices in their demo sites' buildings or also the overall building, while they can also see short-term forecasts of such energy consumptions. The Energy Consumptions Visualizations component provides also the visual and comfort profiles of apartments and various measurements for the indoor temperature, humidity and illuminance of the buildings' apartments. Through the use of these dashboards the building managers and sunbsequently the building occupants may realize how the energy is consumed in their building and eventually move towards energy saving behaviours.

A navigation to the Energy Consumptions Visualizations component is presented in section 10.

9.2 Technologies and Tools

The Energy Consumptions Visualizations component is written in Python. For the user interface implementation, Vue.js has been exploited, while for the backend implementation Django framework has been used.

9.3 Software package repository

The Energy Consumptions Visualizations component is closed source and no source code is available publicly. The source code and the related deployment instructions are maintained in the related private repositories and the corresponding subcomponents are containerized with Docker.

10 Navigation to the EXCESS Data Analytics Framework

Within this section, the navigation to the EXCESS Data Analytics Framework is presented through descriptive screenshots. As only the Flexibilty Analytics Visualizations component and the Energy Consumptions Visualizations component have a user interface, screenshots are following only for these components of the EXCESS Data Analytics Framework.

10.1 Login

In this page, the user is prompted to enter his credentials (email address and password) in order to enter the dashboards. Depending on the type of the user (building manager or aggregator) the corresponding dashboards are displayed after the successful login.

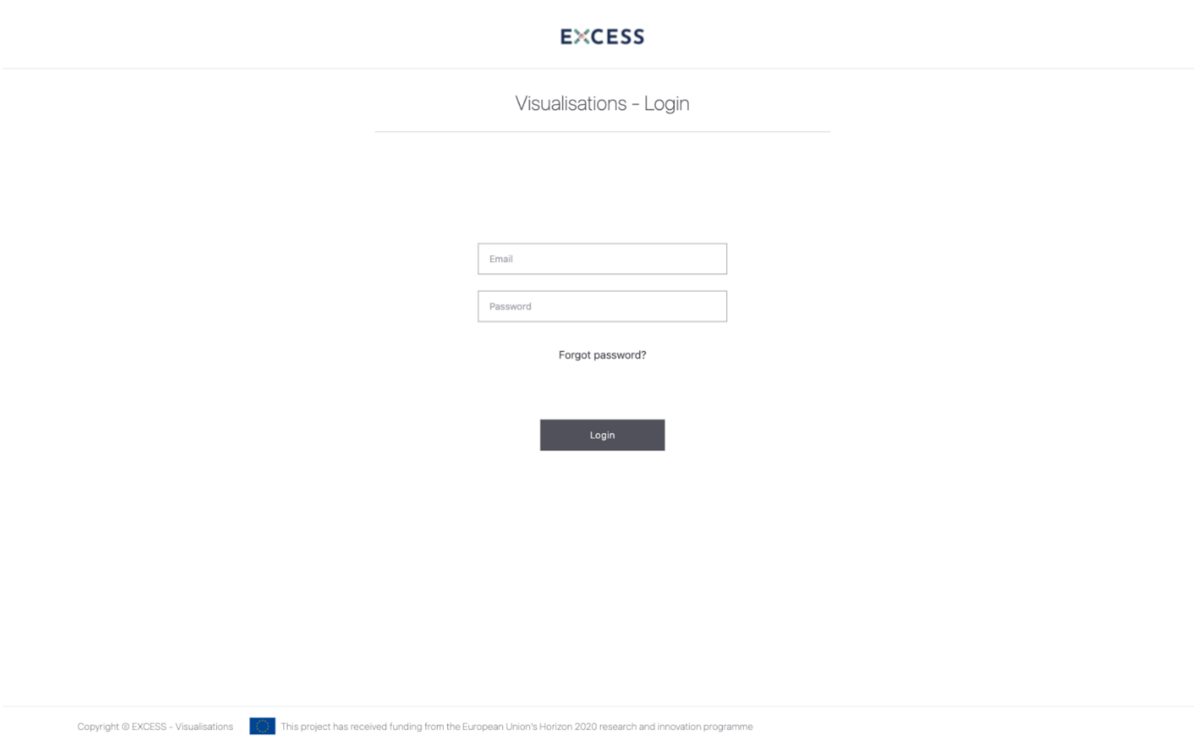

Figure 10-1: Login page

10.2 Energy Consumptions Visualizations - Menu

The Energy Consumptions Visualizations offer the opportunity to the building manager to navigate to the different visualization categories, namely the main dashboard with energy consumptions, the comfort visualizations and the sensor measurements.

Figure 10-2: Energy Consumptions Visuaizations Menu

10.3 Energy Consumptions Visualizations – Main Dashboard

In this page, the building manager may view the energy consumption of the building or a specific apartment or device. By selecting a certain time period, s/he can change the historical details that s/he desires to view. In addition, a 24-hr energy consumption forecast is displayed.

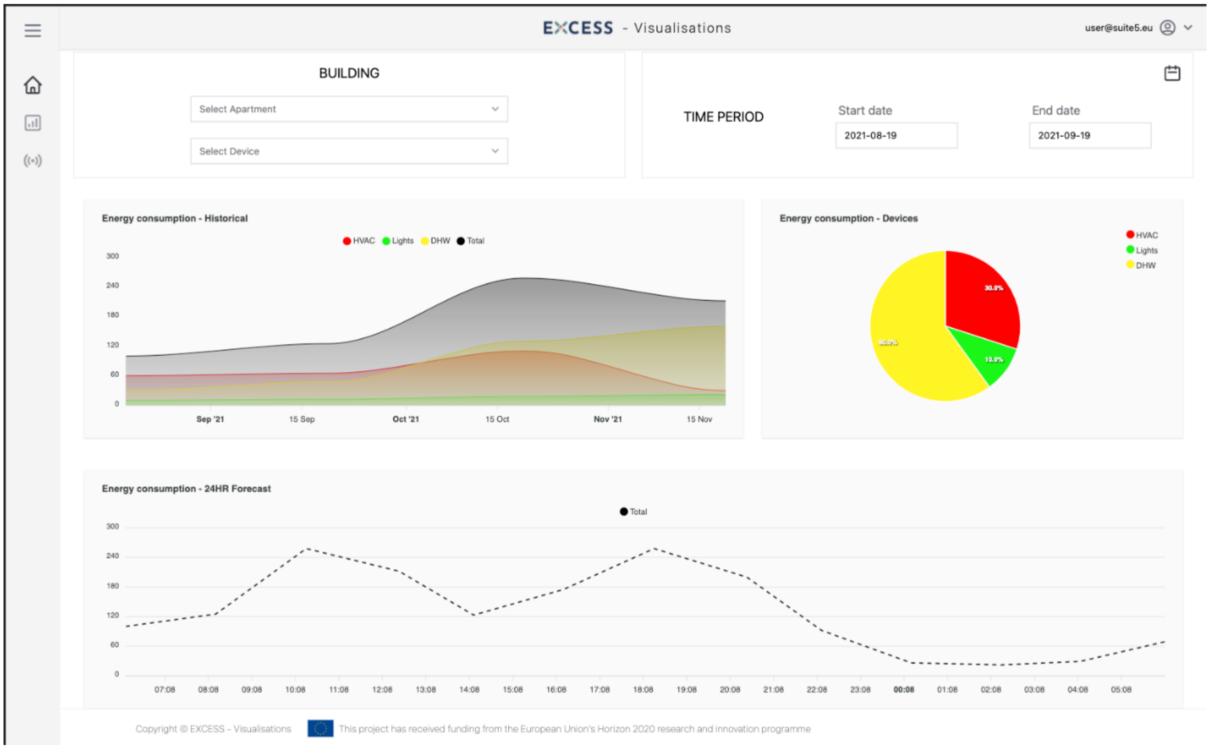

Figure 10-3: Energy Consumptions Visuaizations Main Dashboard

10.4 Energy Consumptions Visualizations - Comfort

In this page, the building manager can view the thermal and visual comfort profiles for every apartment of the building in the corresponding heatmaps. The time period can be changed to view certain historical information.

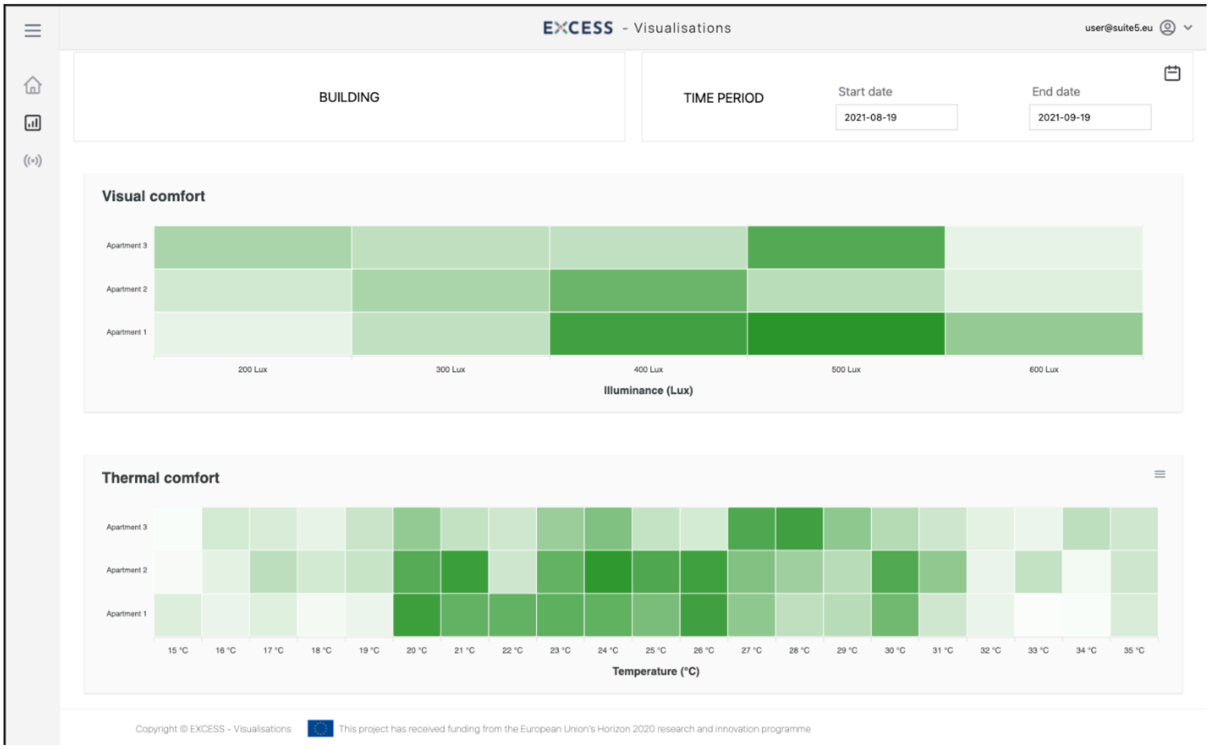

Figure 10-4: Energy Consumptions Visuaizations Comfort Dashboards

10.5 Energy Consumptions Visualizations – Sensor measurements

In this page, the building manager can view the different indoor measurements for an apartment regarding temperature, humidity and illuminance by selecting a specific time period.

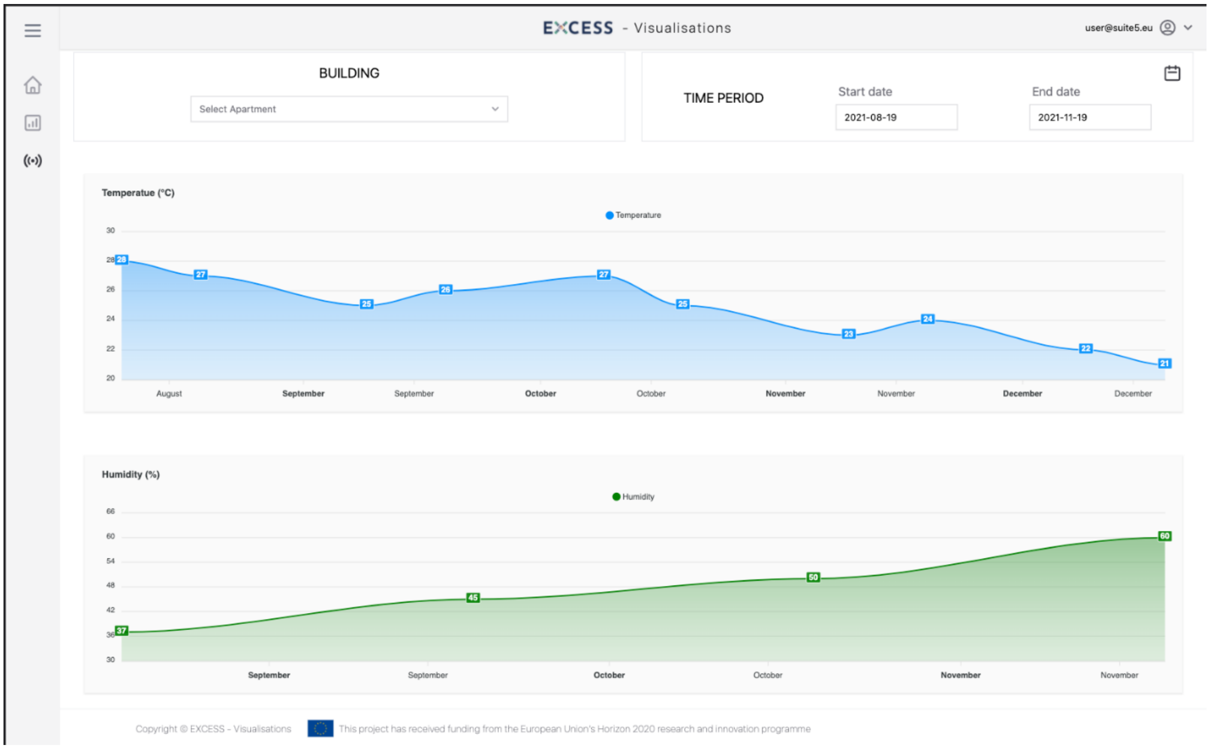

Figure 10-5: Energy Consumptions Visualizations Sensor Measurements (1)

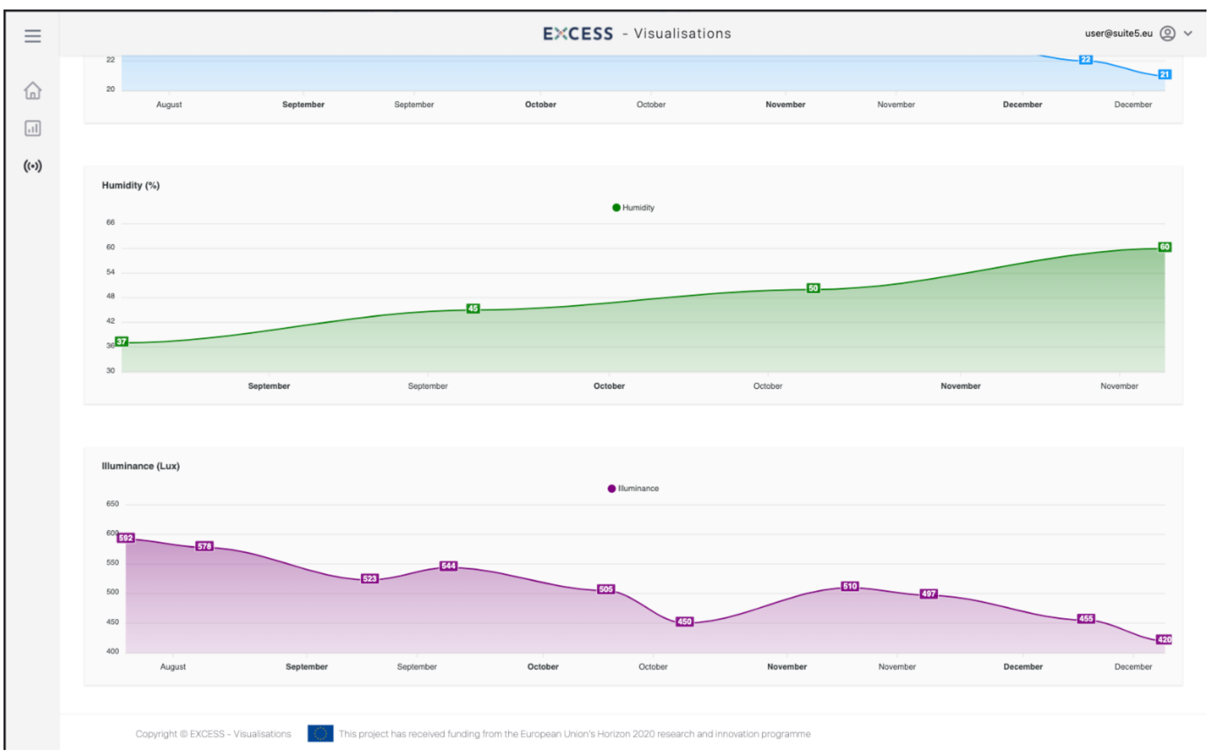

Figure 10-6: Energy Consumptions Visualizations Sensor Measurements (2)

10.6 Flexibility Analytics Visualizations Dashboard

In this page, the aggregator can monitor the historical flexibilities of the building or a specific apartment or device, by selecting a desired time period. In addition, a 24-hr flexibility forecast is displayed.

Figure 10-7: Flexibility Analytics Visualizations Dashboard

11 Conclusions

This deliverable has documented the activities of the Task 3.3 “Core ICT platform services” and Task 3.4 “Flexibility analysis and forecasting component” that have driven the design and implementation of the EXCESS Data Analytics Framework. Its different components have been presented, namely the Comfort Profiling component, the Demand Forecasting component, the Generation Forecasting component, the Dynamic VPP Configuration component, the Context-Aware Flexibility Profiling and Analytics component, the Flexibility Analytics Visualizations and the Energy Consumptions Visualizations. The state-of-the-art on which these components have been designed and developed has been described and their functionalities have been defined along with information about their exploited technologies and software code. A navigation guide is also presented with the self-descriptive schreenshots of the first release of EXCESS Visualization Dashboards.

The deliverable D3.3 has presented the first release of the EXCESS Data Analytics Framework. In M42 of the project, an updated version of the deliverable will be documented based on the feeback coming from the initial operation of the demo sites’ buildings and including also any enhancements and additional functionalities (such as Lights Flexibilities) in the final release of the EXCESS Data Analytics Framework.

References

[1] Antoniadou, P., & Papadopoulos, A. M. (2017). Occupants' thermal comfort: State of the art and the prospects of personalized assessment in office buildings. Energy and Buildings, 153, 136-149.

[2] Kim, J., Schiavon, S., & Brager, G. (2018). Personal comfort models–A new paradigm in thermal comfort for occupant-centric environmental control. Building and Environment, 132, 114-124.

[3] Godithi, S. B., Sachdeva, E., Garg, V., Brown, R., Kohler, C., & Rawal, R. (2019). A review of advances for thermal and visual comfort controls in personal environmental control (PEC) systems. Intelligent Buildings International, 11(2), 75-104.

[4] Zhao, Q., Zhao, Y., Wang, F., Wang, J., Jiang, Y. and Zhang, F., 2014. A data-driven method to describe the personalized dynamic thermal comfort in ordinary office environment: From model to application. Building and Environment, 72, pp.309-318

[5] Li, D., Menassa, C. C., & Kamat, V. R. (2017). Personalized human comfort in indoor building environments under diverse conditioning modes. Building and Environment, 126, 304-317.

[6] Ming, R., Yu, W., Zhao, X., Liu, Y., Li, B., Essah, E., & Yao, R. (2020). Assessing energy saving potentials of office buildings based on adaptive thermal comfort using a tracking-based method. Energy and Buildings, 208, 109611.

[7] Cosma, A.C. and Simha, R., 2019. Machine learning method for real-time non-invasive prediction of individual thermal preference in transient conditions. Building and Environment, 148, pp.372-383

[8] Shen, E., Hu, J., & Patel, M. (2014). Energy and visual comfort analysis of lighting and daylight control strategies. Building and Environment, 78, 155-170

[9] Ma, G., & Pan, X. (2021). Research on a Visual Comfort Model Based on Individual Preference in China through Machine Learning Algorithm. Sustainability, 13(14), 7602.

[10] Ngarambe, J., Irakoze, A., Yun, G. Y., & Kim, G. (2020). Comparative performance of machine learning algorithms in the prediction of indoor daylight illuminances. Sustainability, 12(11), 4471.

[11] Despenic, M., Chraibi, S., Lashina, T., & Rosemann, A. (2017). Lighting preference profiles of users in an open office environment. Building and Environment, 116, 89-107.

[12] Malavazos, C., Papanikolaou, A., Tsatsakis, K., & Hatzoplaki, E. (2015, May). Combined visual comfort and energy efficiency through true personalization of automated lighting control. In 2015 International Conference on Smart Cities and Green ICT Systems (SMARTGREENS) (pp. 1-7). IEEE.

[13] MA, Weiwu, et al. Modeling of district load forecasting for distributed energy system. Applied Energy, 2017, 204: 181-205.

[14] MANIVANNAN, Manoj; NAJAFI, Behzad; RINALDI, Fabio. Machine learning-based short-term prediction of air-conditioning load through smart meter analytics. Energies, 2017, 10.11: 1905.

[15] WASEEM, Muhammad; LIN, Zhenzhi; YANG, Li. Data-driven load forecasting of air conditioners for demand response using levenberg–marquardt algorithm-based ANN. Big Data and Cognitive Computing, 2019, 3.3: 36.

[16] BASHIR, Mohammed Bakri; ALOTAIBI, Abdullah Alhumaidi. Smart buildings Cooling and Heating Load Forecasting Models. IJCSNS, 2020, 20.12: 79.

[17] BEGHI, Alessandro, et al. Load forecasting for the efficient energy management of HVAC systems. In: 2010 IEEE International Conference on Sustainable Energy Technologies (ICSET). IEEE, 2010. p. 1-6.

[18] XU, Yang, et al. Potential Analysis of the Attention-Based LSTM Model in Ultra-Short-Term Forecasting of Building HVAC Energy Consumption. Frontiers in Energy Research, 2021, 463.

[19] MALTAIS, Louis-Gabriel; GOSSELIN, Louis. Predictability analysis of domestic hot water consumption with neural networks: From single units to large residential buildings. Energy, 2021, 229: 120658.

[20] FERRANTELLI, Andrea, et al. Analytical modelling and prediction formulas for domestic hot water consumption in residential Finnish apartments. Energy and Buildings, 2017, 143: 53-60.

[21] GELAŽANSKAS, Linas; GAMAGE, Kelum AA. Forecasting hot water consumption in residential houses. Energies, 2015, 8.11: 12702-12717.

[22] LOMET, Aurore; SUARD, Frédéric; CHÈZE, David. Statistical modeling for real domestic hot water consumption forecasting. Energy Procedia, 2015, 70: 379-387

[23] CAO, Sheng, et al. Predictive control based on occupant behavior prediction for domestic hot water system using data mining algorithm. Energy Science & Engineering, 2019, 7.4: 1214-1232

[24] GELAŽANSKAS, Linas; GAMAGE, Kelum AA. Forecasting hot water consumption in dwellings using artificial neural networks. In: 2015 IEEE 5th International Conference on Power Engineering, Energy and Electrical Drives (POWERENG). IEEE, 2015. p. 410-415.

[25] DELORME-COSTIL, Alexandra; BEZIAN, Jean-Jacques. Forecasting domestic hot water demand in residential house using artificial neural networks. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2017. p. 467-472.

[26] SHEN, Gulai, et al. A data-driven electric water heater scheduling and control system. Energy and Buildings, 2021, 242: 110924.

[27] López, M., Valero, S., Sans, C., & Senabre, C. (2021). Use of Available Daylight to Improve Short-Term Load Forecasting Accuracy. Energies, 14(1), 95

[28] Amasyali, K., & El-Gohary, N. M. (2018). A review of data-driven building energy consumption prediction studies. Renewable and Sustainable Energy Reviews, 81, 1192-1205.

[29] Ahmad, T., Chen, H., Guo, Y., & Wang, J. (2018). A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review. Energy and Buildings, 165, 301-320.

[30] Shan, S., Cao, B., & Wu, Z. (2019). Forecasting the short-term electricity consumption of building using a novel ensemble model. IEEE Access, 7, 88093-88106.

[31] Amasyali, K., & El-Gohary, N. M. (2018). A review of data-driven building energy consumption prediction studies. Renewable and Sustainable Energy Reviews, 81, 1192-1205.

[32] Runge, J., & Zmeureanu, R. (2019). Forecasting energy use in buildings using artificial neural networks: A review. Energies, 12(17), 3254.

[33] Liu, D., & Chen, Q. (2013, June). Prediction of building lighting energy consumption based on support vector regression. In 2013 9th Asian Control Conference (ASCC) (pp. 1-5). IEEE.

[34] Amasyali, K., & El-Gohary, N. (2016). Building lighting energy consumption prediction for supporting energy data analytics. Procedia Engineering, 145, 511-517.

[35] Moral-Carcedo, J., & Pérez-García, J. (2019). Time of day effects of temperature and daylight on short term electricity load. Energy, 174, 169-183

[36] López, M., Valero, S., Sans, C., & Senabre, C. (2021). Use of Available Daylight to Improve Short-Term Load Forecasting Accuracy. Energies, 14(1), 95

[37] Bognar, A., Loonen, R., and J Hensen, J.. (2021). 'Pyrano - A Python Package for LiDAR-Basedsolar Irradiance Simulations'. In Building Simulation 2021 (Vol. 17). International Building Performance Simulation Association (IBPSA).

[38] "Flexibility and Aggregation Requirements for their interaction in the market". Available at: http://www.eurelectric.org/media/115877/tf_bal-agr_report_final_je_as-2014-030-0026-01-e.pdf

[39] Alic, O., & Filik, Ü. B. (2020). Consumer flexibility driven energy management for air conditioning systems in a building community. IET Generation, Transmission & Distribution, 14(15), 3052-3062.

[40] Lork, Clement, et al. "How many watts: A data driven approach to aggregated residential air-conditioning load forecasting." 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, 2017.

[41] Chen, Y., Chen, Z., Xu, P., Li, W., Sha, H., Yang, Z., ... & Hu, C. (2019). Quantification of electricity flexibility in demand response: Office building case study. Energy, 188, 116054.

[42] Che, Y., Yang, J., Zhou, Y., Zhao, Y., He, W., & Wu, J. (2019). Demand response from the control of aggregated inverter air conditioners. IEEE Access, 7, 88163-88173.

[43] "Reinforced model predictive control (RL-MPC) for building energy management", Journal on Applied Energy, 2022, DOI: 10.1016/j.apenergy.2021.118346

[44] "Identification of multi-zone grey-box building models for use in model predictive control", Journal of Building Performance Simulation, 2020, DOI: 10.1080/19401493.2020.1770861