H2020-NMBP-ST-IND-2018-2020- GA 958218

PLUG-AND-USE RENOVATION WITH ADAPTABLE LIGHTWEIGHT SYSTEMS

PLURAL

D5.2 Multi-objective and multi-criteria Decision Support Tool (MODEST)

Version: 1.0

	Name	Date
Prepared by	Ioannis Christou (INTRA), Nasos Grigoropoulos (INTRA), Christos Stravrogiannis (INTRA), Marios Logotheris (INTRA)	29/11/2022
Reviewed by WP leader	Ioannis Christou (INTRA)	29/11/2022
Reviewed and edited by	Maria Founti (Project Coordinator, NTUA)	30/11/2022

Distribution list

External		Internal	
European Commission	1x	Consortium partners	1x

Change log

Issue	Date	Pages	Remark / changes	Pages
0.1	12.07.2022	5	Initial TOC prepared	All
0.2	18.11.2022	20	First issue by INTRA	All
0.3	29.11.2022	78	WP leader review by INTRA	All
1.0	30.11.2022	78	Reviewed and edited by PC	All

To be cited as

INTRA (2022): “D5.2 – Multi-objective and multi-criteria Decision Support Tool (MODEST)” of the HORIZON 2020 project PLURAL. EC Grant Agreement No. 958218, Luxemburg, Country.

Disclaimer

The sole responsibility of this publication lies with the author. The European Union is not responsible for any use that may be made of the information contained therein.

Table of contents

List of Figures

List of Tables

List of Listings

Terms, definitions, and abbreviated terms

API	**A**pplication **P**rogramming **I**nterface
BIM	**B**uilding **I**nformation **M**odelling
CI/CD	**C**ontinuous **I**ntegration / **C**ontinuous **D**evelopment
CLI	**C**ommand **L**ine **I**nterface
CPU	**C**entral **P**rocessing **U**nit
CSV	**C**omma-**S**eparated **V**alues
DHW	**D**omestic **H**ot **W**ater
DSL	**D**omain **S**pecific **L**anguage
DST	**D**ecision Support Tool
GA	**G**rant **A**greement
GUI	**G**raphical **U**ser **I**nterface
HTTP	**H**yper**t**ext **T**ransfer **P**rotocol
HTTPS	**H**yper**t**ext **T**ransfer **P**rotocol **S**ecure
IEQ	**I**ndoor **E**nvironmental **Q**uality
IFC	**I**ndustry **F**oundation **C**lasses
IOPS	**I**nput/Output **O**perations **P**er **S**econd
IT	**I**nformation **T**echnology
JSON	**J**ava**S**cript **O**bject **N**otation

KPI	**K**ey **P**erformance **I**ndicator
LCA	**L**ife **C**ycle **A**ssessment
LCC	**L**ife **C**ycle **C**osting
MODEST	**M**ulti-**O**bjective **DE**cision **S**upport **T**ool
nZEB	**n**early **Z**ero **E**nergy **B**uilding
PMV	**P**redicted Mean Vote
PnU	**P**lug and **U**se
PPM	**P**arts **P**er **M**illion
PPS	**P**ackets **P**er **S**econd
REST	**RE**presentational **S**tate **T**ransfer
SEM	**S**mart **E**nergy **M**anagement
TCP	**T**ransmission **C**ontrol **P**rotocol
TLS	**T**ransport **L**ayer **S**ecurity
TSDB	**T**ime **S**eries **D**ata**B**ase
UI	**U**ser **I**nterface
URL	**U**niform **R**esource **L**ocator
VCS	**V**ersion **C**ontrol **S**ystem
VM	**V**irtual **M**achine
WP	**W**ork **P**ackage

Executive summary

This document accompanies PLURAL Deliverable D5.2 “Multi-Objective and multi-criteria Decision Support Tool (MODEST)”, which is of type software. The present document describes the work done in the frame of:

- Task 5.2: Development of LYSIS modules for Component long term Global environmental impact, cost for energy use and improvements to IEQ, comfort and health (LCA/LCC)
- Task 5.3: LYSIS platform: Integration, implementation, validation of data interoperability platform inter-connecting the PLURAL tools
- Task 5.4: Development of Multi-Objective and multi-criteria DEcision Support Tool (MODEST) for the selection of refurbishment solutions taking into account end user needs and requirements

all of which culminate in the delivery of the software implementing the PLURAL “MODEST” (Multi-Objective and multi-criteria DEcision Support Tool).

MODEST exchanges information with the big data repository LYSIS, which is based on INTRA’s proprietary “Streamhandler”.

Deliverable D5.2 presents the LYSIS infrastructure, supporting services and tools running on the platform and finally the platform itself, which is needed for the MODEST tool to run on top of. The architecture of the MODEST tool is then presented in detail, together with a “Users Reference Guide” that allows the users of the MODEST tool to use it for the pilot Use-Cases of the project. In the end, we present our conclusions. Additional information can be found in the 3 Appendixes of this document.

1. Introduction

Deliverable *"D5.2 – Multi-objective and multi-criteria DEcision Support Tool (MODEST)"* is a software deliverable that includes the LYSIS platform, its modules and the accompanying developed PLURAL tools including MODEST. This document is the accompanying report that describes the developed software with its supporting computing infrastructure and provides information for their usage by different stakeholders, let them be IT engineers or end users. In particular, here we present the outcomes of the following three tasks:

- T5.2 "Development of LYSIS modules for Component long term Global environmental impact, cost for energy use and improvements to IEQ, comfort and health (LCA/LCC)"
- T5.3 "LYSIS platform: Integration, implementation, validation of data interoperability platform inter-connecting the PLURAL tools"
- T5.4 "Development of Multi-Objective and multi-criteria DEcision Support Tool (MODEST) for the selection of refurbishment solutions taking into account end user needs and requirements"

The presented results are connected to some extent with the following WPs:

- WP1: The KPIs defined in T1.5 "Key Performance Indicators*"* and documented in its linked deliverable *"D1.4 – Key performance indicators, system components and performance targets"* where used as input for the calculation of the performance indicators related to monitoring data coming from the real demo sites.
- WP3: As part of T3.2 "Building and asserts data management", a cloud data architecture has already been built for supporting the management and storage of data with different granularity, origin and structure (for more details, see *"D3.2 – Building and assets data management and analytics"*). In the activities that took place in WP5 and more specifically in T5.2, T5.3 and T5.4, this cloud data architecture has been extended and enriched with new functionalities to serve the project's objectives in the best possible way. Further, there is a connection with the works taking place in T3.5. "SEM implementation, systems test adaptation and validation under theoretical conditions" where the algorithms developed in T3.3 "Development of the simulation framework" and T3.4 "Adaptation of the Smart Energy System (SEM) analytics" are being integrated in the LYSIS platform.
- WP7: Even though not clearly depicted in the GA's original pert diagram with the WP dependencies, there is a close interaction with the ongoing T7.6 "Monitoring campaign" where

technical support and tools have been provided to facilitate and ensure the uninterrupted flow of monitoring data during the pre-renovation (pre-PRURAL) phase.

The rest of this document is structured as follows (the related task(s) are mentioned in parenthesis):

- Section 2 gives an overview of the computing infrastructure that has been setup for hosting the LYSIS platform and its accompanying services and tools (set up in T5.3 and used in T5.2, T5.3, T5.4)
- Section 3 describes the Continuous Integration / Continuous Development (CI/CD) workflow that supports the software development activities and the specific set of the tools deployed (set up in T5.3 and used in T5.2, T5.3, T5.4)
- Section 4 provides details on the architecture of the LYSIS platform and the design of its various components (developed in T5.2 and T5.3)
- Section focuses on MODEST, the calculation and retrieval of KPIs and the GUI created for the visualization of the various types of information (developed in T5.4)

2. LYSIS Infrastructure

This section provides an overview of the computing infrastructure that has been set up for hosting the LYSIS platform (section 2.1) and of the security features that play a significant role in its smooth operation (section 2.2).

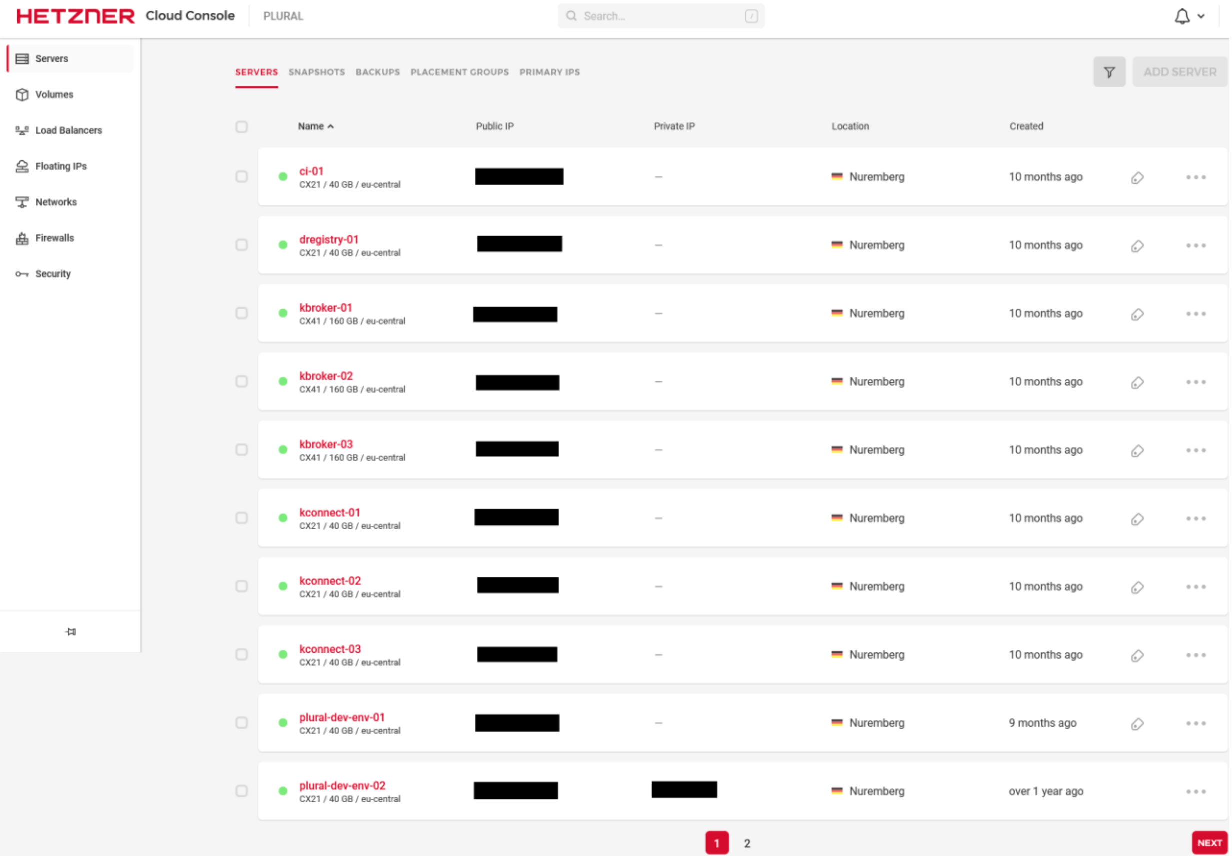
2.1 Setup

For the purposes of the project multiple virtual servers have been acquired from a public cloud provider, namely Hetzner Cloud [1] and setup accordingly. For the time being, this LYSIS infrastructure consists of a total of 17 virtual machines (VMs) where 12 of them are used to support and enhance the functionality of the StreamHandler system and the remaining 5 are used for the CI/CD tools, the development/testing/deployment of the software components, and the storage of the created binary files (containers). All VMs run the Rocky Linux [2] open-source enterprise operating system (version 8) while their hardware setup in terms of CPU, memory and disk depend on the intended functionality. Table 1 lists the VM servers and describes the functionality of the tools and/or components they host, while Figure 1 shows a screenshot of the infrastructure via Hetzner's Cloud Console UI.

TABLE 1: THE SERVERS THAT MAKE UP THE LYSIS INFRASTRUCTURE

Server name(s)	Description
ci-01	Hosts supporting CI/CD tools for the automated building, testing and deployment of software components (Jenkins) and the management and monitoring of the containerized components (Portainer).
dregistry-01	Hosts the container registry (JFrog Container Registry) for the management, storage and distribution of produced container images.
plural-dev-env-01, plural-dev-env-02	Development environments used during the software development and testing phases.
plural-prod-env	Production environment used for the deployment of successfully tested software components.
kbroker-01, kbroker-02, kbroker-03	Apache Kafka cluster, part of the StreamHandler platform.
zkeeper-01, zkeeper-02, zkeeper-03	Apache ZooKeeper quorum, part of the StreamHandler platform.
kconnect-01, kconnect-02, kconnect-03	Kafka Connect cluster, part of the StreamHandler platform used for the deployment of Connectors integrating Apache Kafka with external systems.
rproxy-01	Hosts the REST proxy for accessing Apache Kafka using a REST API.

sregistry-01	Hosts the schema registry for storing and managing the data models (schemas) of the data exchanged via the Apache Kafka.
ui-01	Hosts the UI tools for the management of the StreamHandler platform.



Name	Public IP	Private IP	Location	Created
ci-01 CX21 / 40 GB / eu-central	[redacted]	–	Nuremberg	10 months ago
dregistry-01 CX21 / 40 GB / eu-central	[redacted]	–	Nuremberg	10 months ago
kbroker-01 CX41 / 160 GB / eu-central	[redacted]	–	Nuremberg	10 months ago
kbroker-02 CX41 / 160 GB / eu-central	[redacted]	–	Nuremberg	10 months ago
kbroker-03 CX41 / 160 GB / eu-central	[redacted]	–	Nuremberg	10 months ago
kconnect-01 CX21 / 40 GB / eu-central	[redacted]	–	Nuremberg	10 months ago
kconnect-02 CX21 / 40 GB / eu-central	[redacted]	–	Nuremberg	10 months ago
kconnect-03 CX21 / 40 GB / eu-central	[redacted]	–	Nuremberg	10 months ago
plural-dev-env-01 CX21 / 40 GB / eu-central	[redacted]	–	Nuremberg	9 months ago
plural-dev-env-02 CX21 / 40 GB / eu-central	[redacted]	[redacted]	Nuremberg	over 1 year ago

FIGURE 1: SCREENSHOT OF THE LYSIS INFRASTRUCTURE IN HETZNER CLOUD

2.2 Security features

To ensure the security of the deployed software components and tools and protect all retrieved and produced data, the LYSIS infrastructure incorporates various security features in different layers. In particular, we consider the following security measures:

- Hard disk encryption of infrastructure servers. This prevents access to data by unauthorized users by ensuring that they remain encrypted and become readable only to authorized entities [3].
- Firewall protection of infrastructure severs. Using the netfilter framework [4] and setting corresponding iptables rules all incoming, outgoing and forwarded packets are dropped and only specific connections are allowed, corresponding to the project's software components or trusted external interacting components. This prevents the most common network attacks.
- Authenticated access to infrastructure servers. Access to the servers for management purposes is performed only using SSH public key authentication through cryptographic keys [5] that provide strong and encrypted verification and communication, while password-based authentication is disabled.
- Encrypted and secure communication with infrastructure servers and deployed tools. Access to the infrastructure servers used for the deployment of software components and to various supporting tools (e.g., the CI/CD tools) is secured by using HTTP over TLS (HTTPS) [6]. This provides encryption, data integrity and authentication, preventing various types of attacks.
- User authentication of services/tools. CI/CD services (e.g., Jenkins, Portainer, JFrog Container Registry) as well as the developed LYSIS/MODEST GUI are secured via user authentication, which could be unified through an open-source identity and access management solution like Keycloak [7].
- Reverse proxy configuration for services/tools. Using the Nginx web server [8] as a reverse proxy the existence and characteristics of the CI/CD services and the developed tools is hidden.

3. LYSIS Supporting Services and Tools

This section briefly describes the Continuous Integration / Continuous Development (CI/CD) workflow that supports the software development activities (section 3.1) and then gives more information about the specific set of supporting tools that have been deployed (sections 3.2-3.5).

3.1 CI/CD concept, pipeline and stack

CI/CD is a collection of software development practices that enables technical teams to build, test and release software in a more automated and reliable way [9] [10]. Some of the benefits that have made CI/CD so popular, are the shorter development cycles which lead to more frequent releases, the reduced risk of defects due to the automation introduced to the different steps of the process and the early testing, and the easier management of rollbacks in case where something goes wrong. Consequently, all these also result to higher quality of delivered software.

Figure 2 depicts an indicative CI/CD pipeline. Continuous Integration is the practice in which software developers commit code changes frequently and each change triggers an automated build-and-test sequence, followed by immediate feedback on the results. Then, once testing is successful, Continuous Development takes care of packaging and deploying the resulted software component.

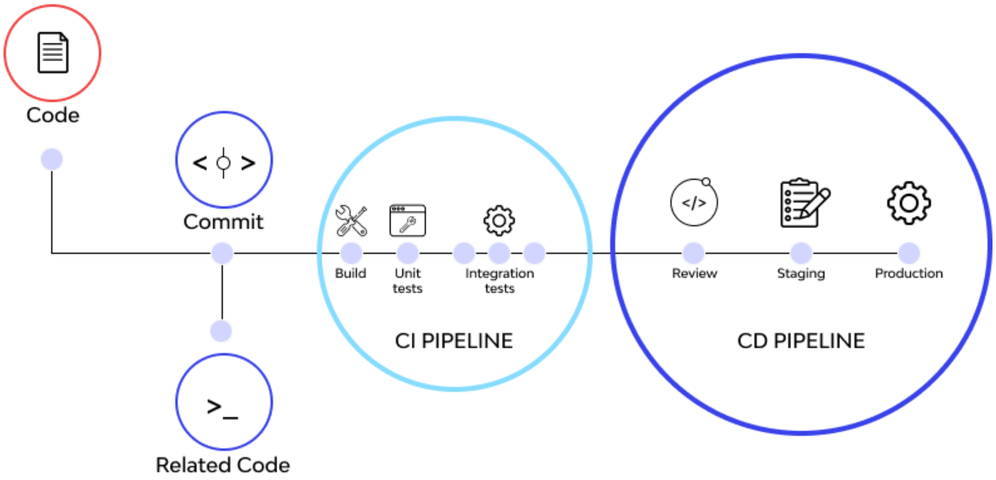


FIGURE 2: A CI/CD PIPELINE (FROM [9])

Figure 3 shows the CI/CD stack which consists of the toolset supporting the entire software lifecycle up to the release of fully tested and deployed operational components. The main tools are:

- GitHub, for source code management and versioning
- Jenkins, for automated building, testing and deployment
- Docker, for the packaging of software components into containers using the de facto standard
- JFrog Container Registry, for the management, storage and distribution of produced container images
- Portainer, for the management and monitoring of the containerized software components
- Nginx, for the efficient management of requests towards the deployed CI/CD tools

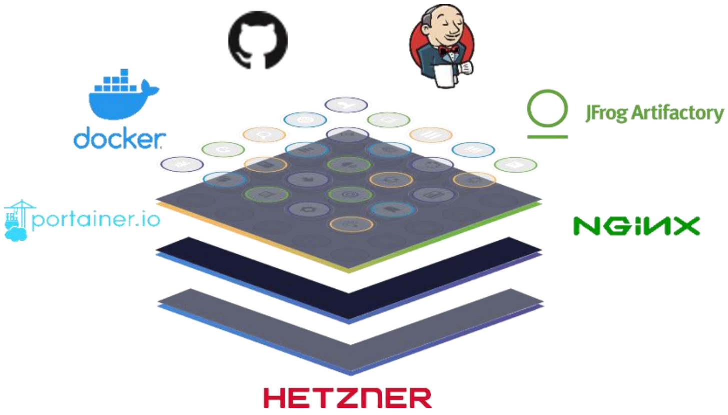


FIGURE 3: THE PLURAL CI/CD STACK

Table 2 lists the URLs of the CI/CD tools. Next, the most important ones are described in more details.

TABLE 2: URLS OF THE PLURAL CI/CD TOOLS

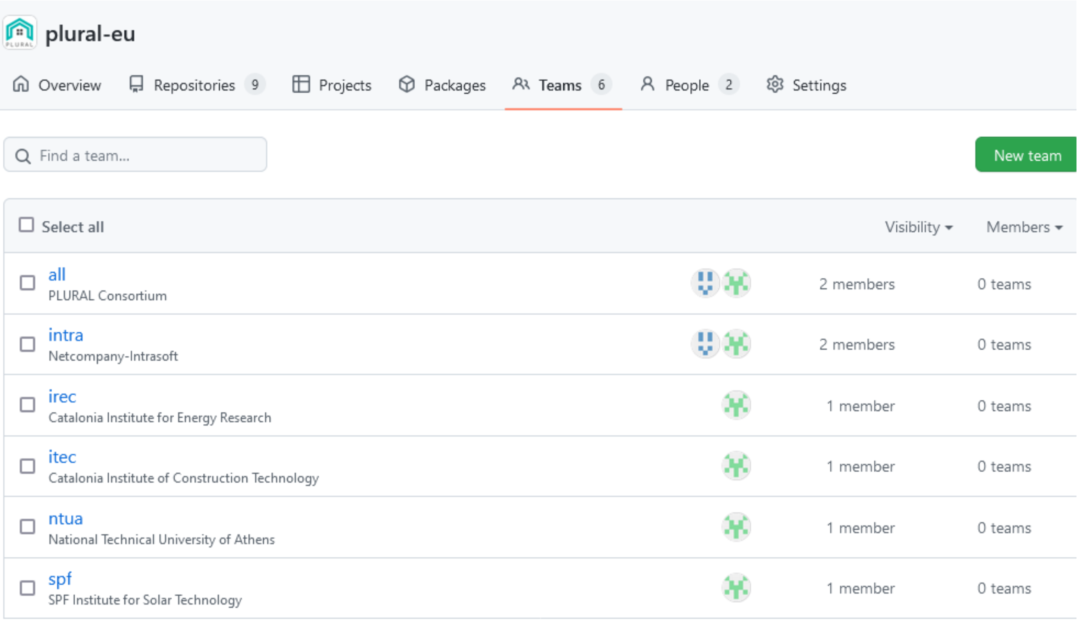
CI/CD tool	URL
GitHub	https://github.com/plural-eu/
Jenkins	https://jenkins.plural.rid-intrasoft.eu/
JFrog Artifactory	https://registry.plural.rid-intrasoft.eu/
portainer.io	https://portainer.plural.rid-intrasoft.eu/

3.2 GitHub

Version control, also called source control, is the practice of monitoring and dealing with modifications to software code [11]. Version control systems (VCS) are software tools that assist developers to record and handle modifications to source code over time. VCS provides a trail for each change to the code in a unique kind of database. If a mistake is made, developers can examine earlier versions of the code to revert the specific problematic modules to a previous state, thus minimizing the overall disruption.

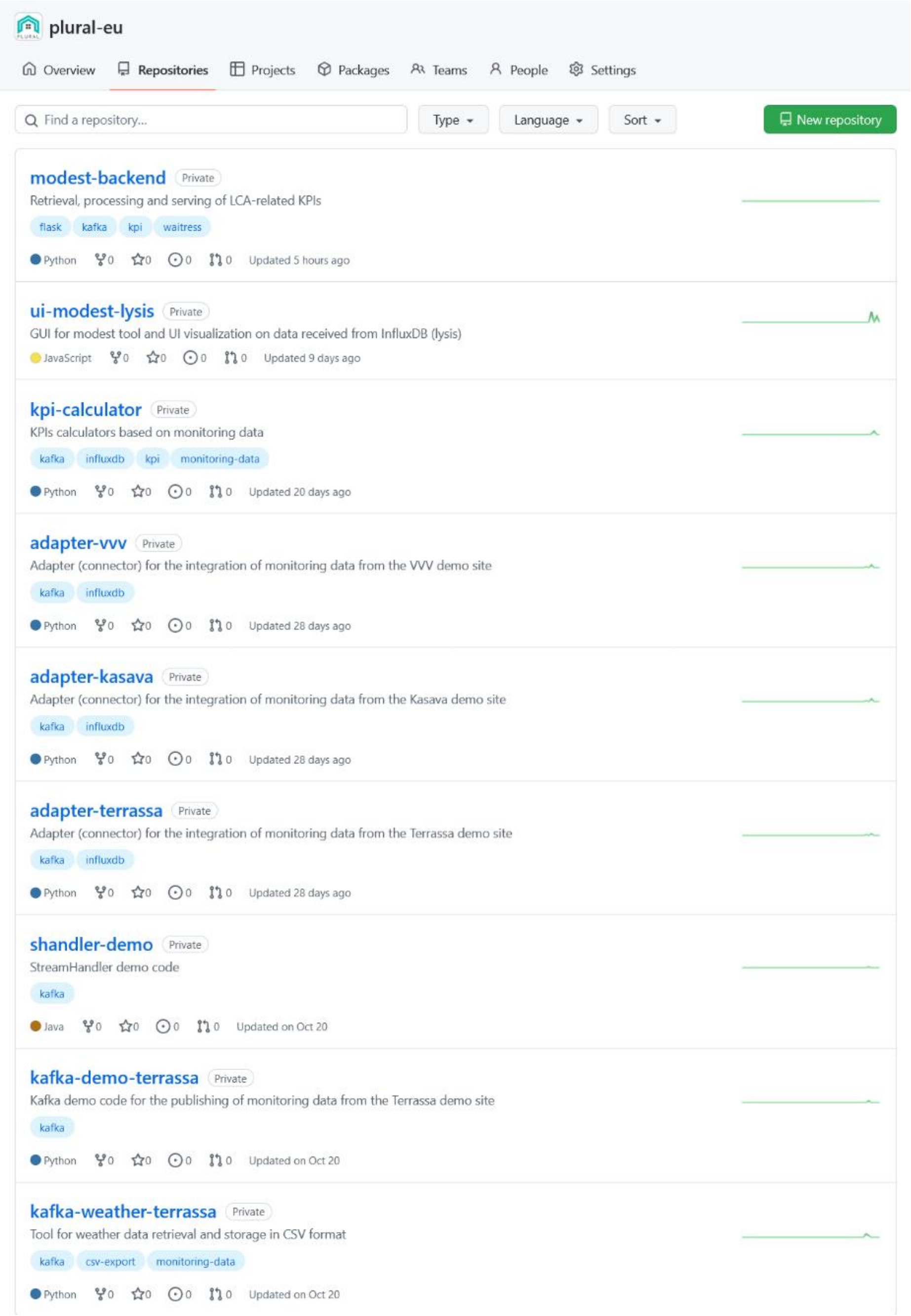
GitHub [12] is used as the VCS in the PLURAL CI/CD platform. It provides a distributed Git-based version control that comes with an extended set of features for the development procedure, which apart from version control include issue tracking, code reviews, wiki, etc. In addition, it provides extensive branching capabilities. Typically, there is a main branch in a repository from which each developer working on a specific feature, or a bug fix creates an additional diverging feature or bugfix branch. Once developers complete their source code changes, they can merge back into the main branch.

For the PLURAL project, a new organization named ***plural-eu*** has been created, under which there are several teams for various project partners, as shown in Figure 4. A partial list of the existing code repositories is depicted in Figure 5. These are currently accessible only to authorized users of the consortium at https://github.com/plural-eu/.



☐ Select all	Visibility ▾	Members ▾
☐ all PLURAL Consortium	2 members	0 teams
☐ intra Netcompany-Intrasoft	2 members	0 teams
☐ irec Catalonia Institute for Energy Research	1 member	0 teams
☐ itec Catalonia Institute of Construction Technology	1 member	0 teams
☐ ntua National Technical University of Athens	1 member	0 teams
☐ spf SPF Institute for Solar Technology	1 member	0 teams

FIGURE 4: THE PLURAL GITHUB ORGANIZATION TEAMS



plural-eu

modest-backend Private
Retrieval, processing and serving of LCA-related KPIs
flask kafka kpi waitress
Python 0 0 0 0 Updated 5 hours ago

ui-modest-lysis Private
GUI for modest tool and UI visualization on data received from InfluxDB (lysis)
JavaScript 0 0 0 0 Updated 9 days ago

kpi-calculator Private
KPIs calculators based on monitoring data
kafka influxdb kpi monitoring-data
Python 0 0 0 0 Updated 20 days ago

adapter-vvv Private
Adapter (connector) for the integration of monitoring data from the VVV demo site
kafka influxdb
Python 0 0 0 0 Updated 28 days ago

adapter-kasava Private
Adapter (connector) for the integration of monitoring data from the Kasava demo site
kafka influxdb
Python 0 0 0 0 Updated 28 days ago

adapter-terrassa Private
Adapter (connector) for the integration of monitoring data from the Terrassa demo site
kafka influxdb
Python 0 0 0 0 Updated 28 days ago

shandler-demo Private
StreamHandler demo code
kafka
Java 0 0 0 0 Updated on Oct 20

kafka-demo-terrassa Private
Kafka demo code for the publishing of monitoring data from the Terrassa demo site
kafka
Python 0 0 0 0 Updated on Oct 20

kafka-weather-terrassa Private
Tool for weather data retrieval and storage in CSV format
kafka csv-export monitoring-data
Python 0 0 0 0 Updated on Oct 20

FIGURE 5: THE PLURAL GITHUB REPOSITORIES

3.3 Jenkins

Jenkins [13] is an automation server that serves as the main facilitator of the CI/CD process, which typically includes the steps listed below:

- Developers make a local clone or copy of an existing, remote source code repository and perform updates

- Changes are pushed to the centralized code repository

- The Jenkins server is automatically notified of all incoming changes

- The server initiates the following list of actions

  1. Pulls the most recent code version

  2. Builds the software and check for any existing issues

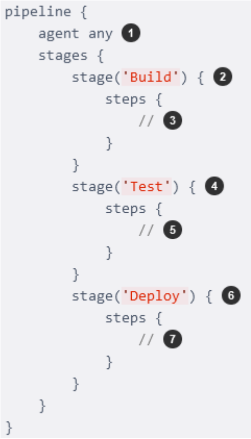
  3. Runs unit and integration checks and report any potential issues

  4. Releases the artifact to be deployed for testing

  5. Assign configuration tags to each newly-constructed the constructed software version

Jenkins uses Pipelines [14] to generate an ordered sequence of events/tasks/actions related to building, testing, packaging, deploying and storing software. Jenkins Pipeline provides an extensible toolset for modeling user-defined delivery pipelines "as code" via the Pipeline domain-specific language (DSL) syntax [15]. A Pipeline can be defined either through the web UI or with a text file called Jenkinsfile [16]. The Jenkinsfile approach is generally considered the best practice and forms the foundation of "Pipeline-as-code" by treating the CI/CD process as part of the application since it is being checked in to source control and versioned like the source code.

A Jenkinsfile can be in declarative of scripted format with the former being the preferred one. Figure 6 shows an indicative structure of a declarative pipeline taken from [14] where the following actions take place at each numbered step:

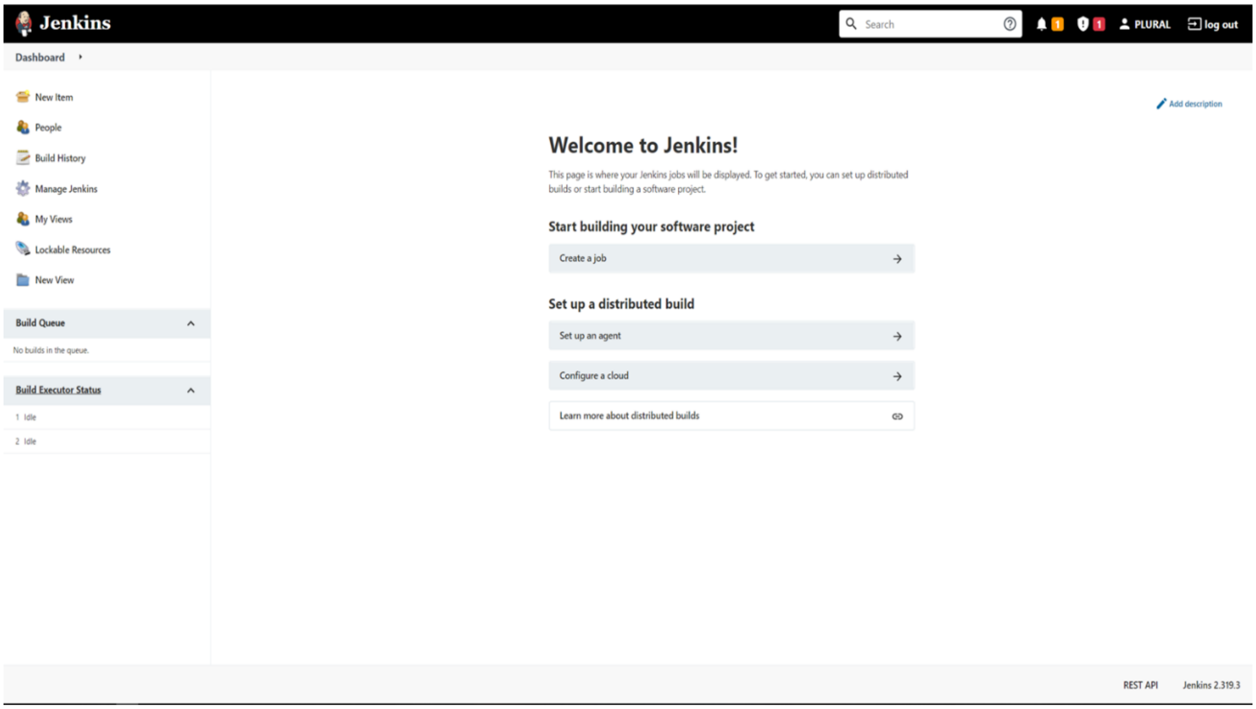


```
pipeline {
    agent any ❶
    stages {
        stage('Build') { ❷
            steps {
                // ❸
            }
        }
        stage('Test') { ❹
            steps {
                // ❺
            }
        }
        stage('Deploy') { ❻
            steps {
                // ❼
            }
        }
    }
}
```

1. Execute this Pipeline or any of its stages, on any agent
2. Defines the "Build" stage
3. Perform some steps related to the "Build" stage
4. Defines the "Test" stage
5. Perform some steps related to the "Test" stage
6. Defines the "Deploy" stage
7. Perform some steps related to the "Deploy" stage

FIGURE 6: JENKINSFILE USING THE DECLARATIVE PIPELINE SYNTAX

Repositories under the **plural-eu** GitHub organization can be associated with corresponding Jenkins Pipelines and that way changes in source codes in the Git repositories can trigger a corresponding Pipeline. The web UI of the PLURAL project’s Jenkins deployment for managing and monitoring the execution status of the defined pipelines is available at: https://jenkins.plural.rid-intrasoft.eu/.



No builds in the queue.

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

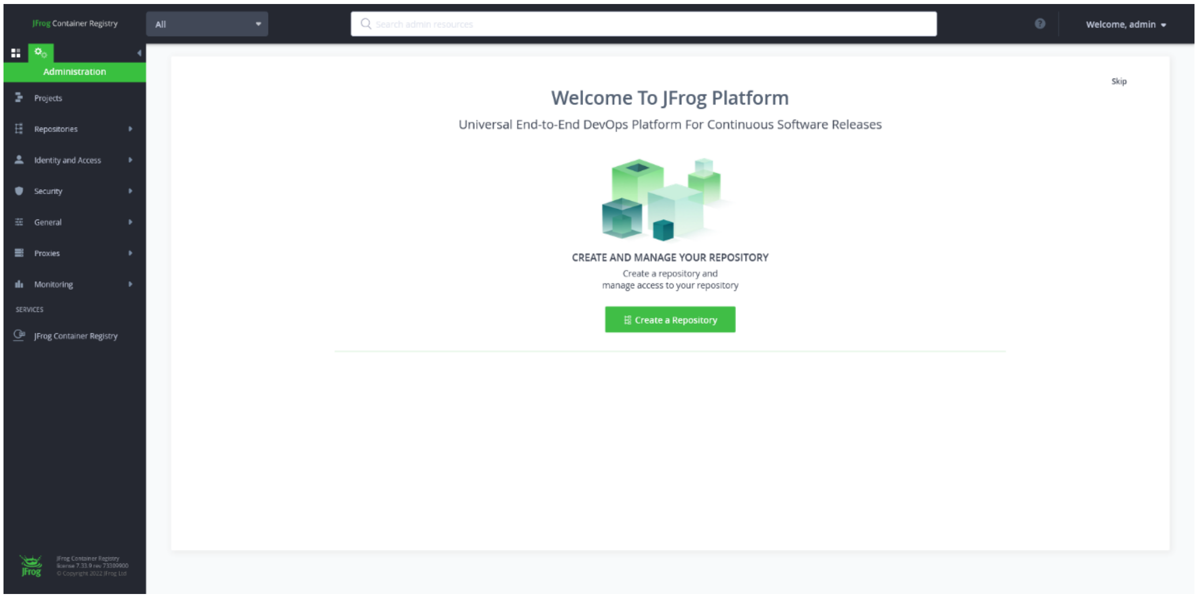
Start building your software project

Set up a distributed build

FIGURE 7: THE PLURAL JENKINS DASHBOARD

3.4 JFrog Container Registry

For the purposes of the project and to accommodate the protection of proprietary code packaged in Docker container image format, which may include sensitive data, a private Docker registry has been provisioned using the JFrog Container Registry solution [17]. Developers can push and/or pull private container images of software components for the PLURAL project by accessing the registry located at: https://registry.plural.rid-intrasoft.eu/. In addition, the registry offers a GUI for management purposes shown in Figure 8.



Welcome To JFrog Platform

Universal End-to-End DevOps Platform For Continuous Software Releases

CREATE AND MANAGE YOUR REPOSITORY

Create a repository and
manage access to your repository

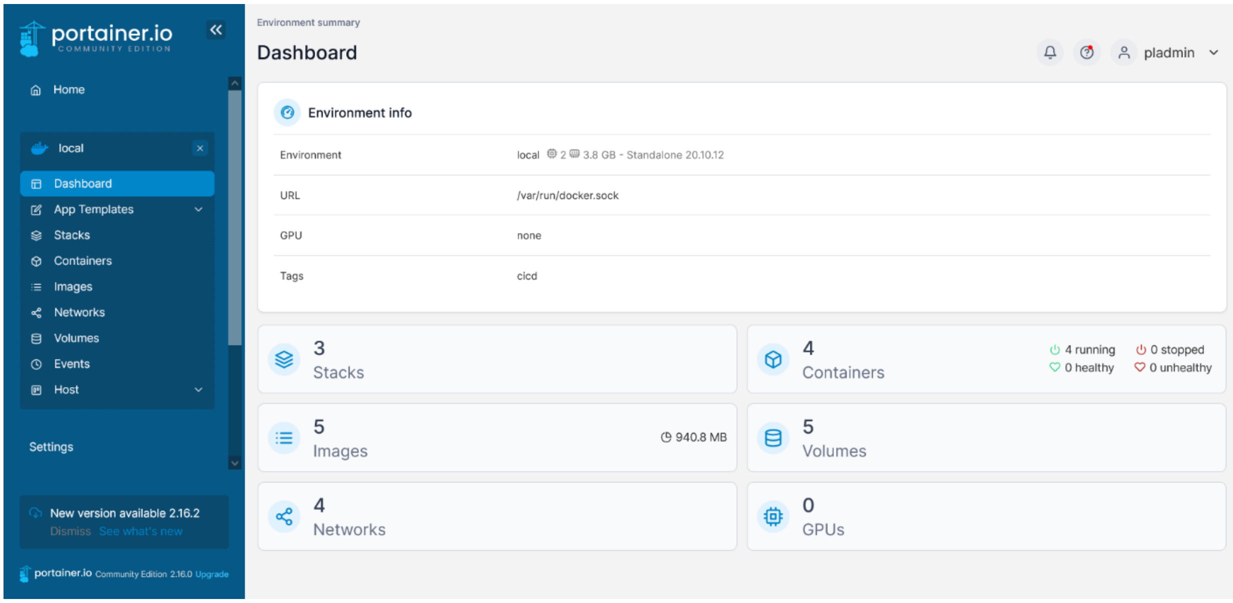
FIGURE 8: THE PLURAL JFROG CONTAINER REGISTRY DASHBOARD

JFrog offers support for both local and remote repositories and imposes no constraint regarding the limit on the number of Docker Registries. A remote repository provides a proxy caching mechanism for a remote URL that manages each repository. Artifacts in remote repositories can be retrieved and investigated, removed from the repository cache, but new artifacts cannot be deployed on it. To deploy artifacts, one should use a local repository as an alternative. Local and remote repositories can be combined into a virtual repository under a common URL for artifact exploration and declaration.

3.5 Portainer

Portainer [18] is a free software that has as main function to manage Docker hosts by offering a convenient UI tool for visualization. It can manage different Docker environments and offers administrators various monitoring and management options for all Docker resources (containers, images, volumes, networks, etc.). For PLURAL, Portainer is containerized and deployed on the Hetzner Cloud infrastructure, being accessible at: https://portainer.plural.rid-intrasoft.eu/.

On the home page, Portainer lists all Docker hosts/servers in the corresponding environment. The dashboard also provides a quick overview of the status of each Docker host. Monitoring and management options are also provided for each container on each Docker host, as shown in Figure 9.



Environment summary

Dashboard

Environment info

Environment	local 2 3.8 GB - Standalone 20.10.12
URL	/var/run/docker.sock
GPU	none
Tags	cicd

3 Stacks

4 Containers — 4 running, 0 stopped, 0 healthy, 0 unhealthy

5 Images — 940.8 MB

5 Volumes

4 Networks

0 GPUs

FIGURE 9: THE PLURAL PORTAINER DASHBOARD

4. LYSIS Platform

This section provides an overview of the LYSIS platform architecture and the data flows (section 4.1) followed by more detailed descriptions of the internals (sections 4.2- 4.6). Finally, it presents some of the user interfaces used for the monitoring and management of the platform (section 4.7).

4.1 Architecture overview

In terms of provided functionality, the overall architecture of the LYSIS platform is divided in three high-level layers:

- BIM layer
- Monitoring data layer
- Simulation data layer

The **BIM layer** provides the infrastructure for receiving, storing and making accessible to other components (a) Building Information Modelling (BIM) files for the buildings of the real and virtual demo sites in International Foundation Classes (IFC) format and (b) the extracted parameters of interest in JSON format. The **monitoring data layer** includes the infrastructure, tools and software components for the collection, storage, analysis, processing and visualization of the sensor measurements coming from the real demo sites as well as the calculation and visualization of the relevant KPIs. Finally, the **simulation data layer** includes the infrastructure for interconnecting simulation tools and the software components for the incorporation, processing and visualization of relevant KPIs. Figure 10 presents an overview of the architecture, showing for each layer the data flow between abstract components representing different types of functionalities.

At the **BIM layer** the *BIM Producers* insert in a *Persistent Messaging and Storage System* the IFC files for the various demo sites. From there, the IFC files can be retrieved by *BIM Consumers* from whom they can be used as-is. A special type of consumers is the *BIM Parsers* that extract the BIM parameters of interest and make them available in the *Persistent Messaging and Storage System* which in term can be used by *BIM Parameter Consumers* to extract and use specific fields in a variety of calculations.

At the **monitoring data layer**, the *Data Collector(s)* of each demo site concentrate measurements from various *Data Sources* including energy, IEQ and weather sensors. The data are then inserted in the *Persistent Messaging and Storage System* either directly or through *Data Adapters* that take also care of storing them in a *Time Series Database* (TSDB). From these storage systems data can be retrieved by *Data Processors* for performing data cleaning, harmonization and calculation of related key performance indicators (KPIs). Also, *Data Visualizers* can be used for visualizing the processed data, the calculated KPIs or even raw data directly from the *Time Series Database*.

Finally, at the **simulation data layer**, the *Simulated Data Producers*, which are typically simulation tools create among others LCA/LCC and energy data for specific PnU configurations, that are inserted either directly or through *Simulated Data Adapters* in the *Persistent Messaging and Storage System.* From there they are used by the *Simulated Data Processors* for performing some extra processing operations if necessary and are then forwarded to *Data Visualizers* for making them available to the end users.

For the *Persistent Messaging and Storage System* LYSIS utilizes INTRA's StreamHandler asset, which is based on Apache Kafka [19], while for the *Time Series Database* it makes use of InfluxDB [20]. Before diving into some more technical details for each layer, in the following section we discuss these two systems (marked green in Figure 10) that used across all of them.

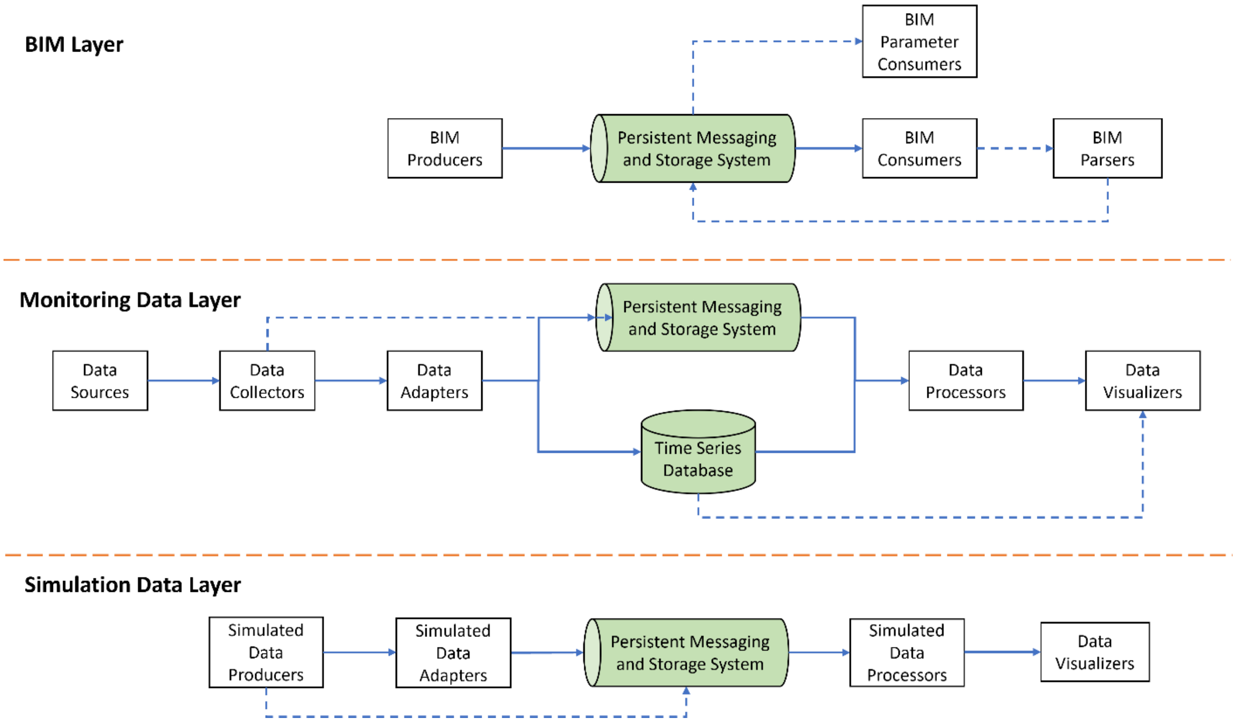


FIGURE 10: LYSIS ARCHITECTURE OVERVIEW

4.2 Messaging and storage systems

In the LYSIS platform, StreamHandler serves as the *Persistent Storage and Messaging System* and InfluxDB as the *Time Series Database*. Figure 11 shows the LYSIS messaging and storage systems architecture where for each module the respective server VM hosting it (from Table 1) is marked with red color.

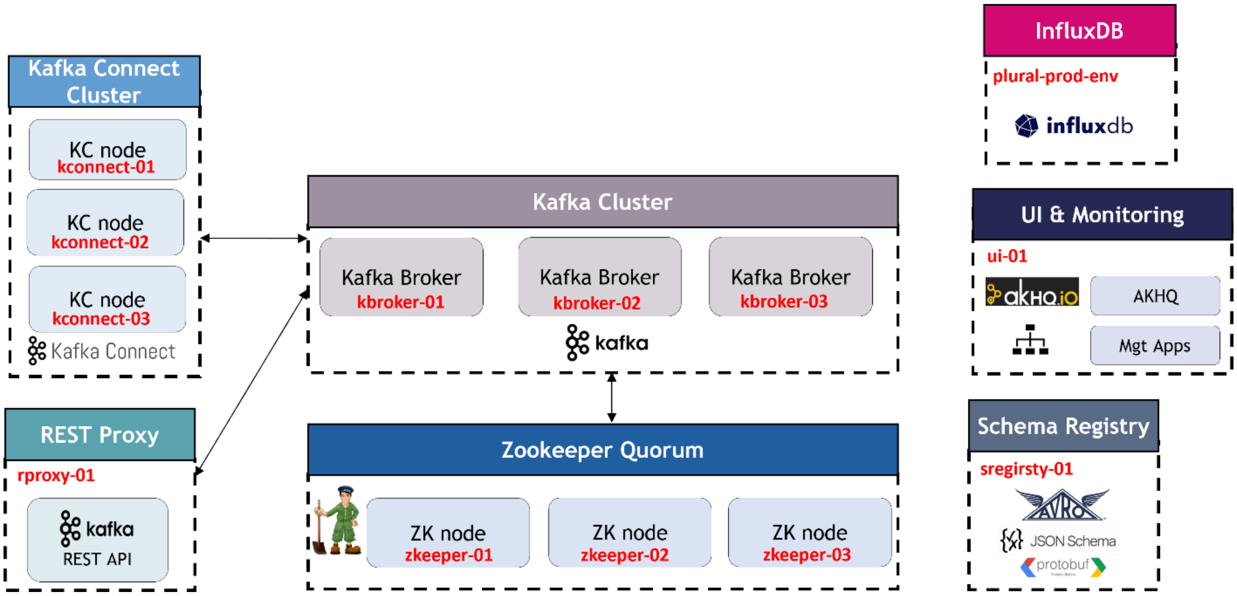


FIGURE 11: LYSIS MESSAGING AND STORAGE SYSTEMS ARCHITECTURE

StreamHandler itself is built around Apache Kafka [19], which is a distributed event-streaming platform following a topic-based publish/subscribe messaging pattern. The way StreamHandler is deployed, it forms a highly scalable, elastic, fault-tolerant and secure cluster and offers the storage of data streams in a durable and fault-tolerant manner, acting as a distributed event log. Among others, StreamHandler comes bundled with a Kafka Connect Cluster and a REST Proxy for enabling the interoperability with a variety of data sources and sinks following different communication patterns (e.g., real-time, batch mode, request-reply). For more information about the various StreamHandler modules and their interconnection please refer to deliverable *"D3.2 – Building and assets data management and analytics"*, section 5. In the context of the LYSIS platform, StreamHandler is used both for facilitating the asynchronous, loosely coupled interaction between the different components by defining respective Apache Kafka topics and configuring them accordingly, and for offering the long-term storage of data for post-processing activities.

InfluxDB [20] is a high-performance data store optimized for the storage and retrieval of time series data in fields like sensor data, real-time analytics, etc. Internally, it consists of measurements where each entry has a timestamp, a collection of field key-value pairs for storing the various measured quantities, and optionally a collection of tag key-value pairs for storing accompanying metadata. In addition, InfluxDB comes with the Flux functional data scripting language [21] for querying, processing, writing and analyzing the data, while it offers a management UI for creating dashboards specified by Flux scripts. In the context of the LYSIS platform InfluxDB, is used for the storage and visualization of monitoring data from the real demo sites, as well as the calculation and visualization of the related KPIs.

4.3 BIM layer

Figure 12 depicts the exact dataflow at the BIM layer. In particular, two topics have been created (a) *bim.ifc.file* and (b) *bim.ifc.params*. BIM Producers publish IFC files at the former topic which are received by a BIM Consumer component running at ITeC's own servers. Each IFC file is internally communicated to the BIM Parser component which obtains the specified BIM parameters and publishes them to the latter topic, thus making them available to any interested party for consumption. For more information regarding the internal functionality of the BIM Parser and the message formats, please refer to deliverable *"D5.1 - Modules for DST – Information requirements for BIM components and databases".*

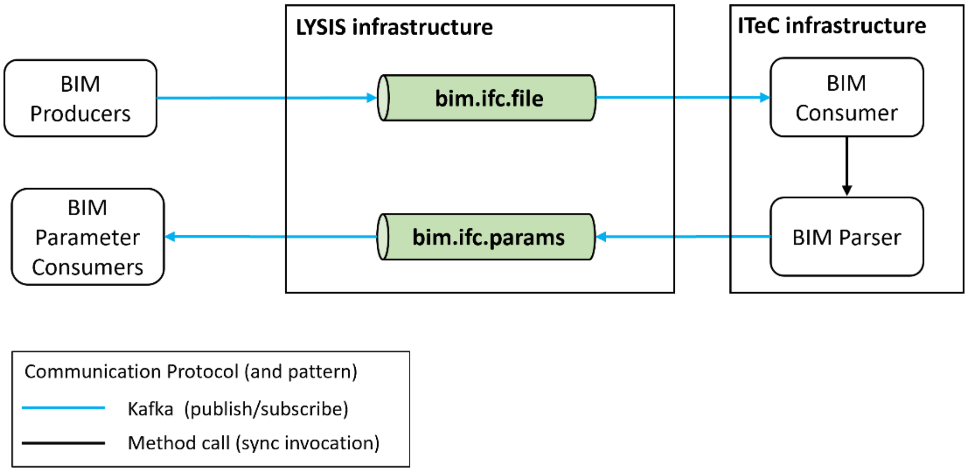


FIGURE 12: BIM LAYER DATAFLOW

It is worth noting that to facilitate the transmission, storage and retrieval of both IFC files and BIM parameters whose size typically is in the order of tens of MB, special configuration options need to be set both for the topics and at the publisher/subscriber sides listed in Table 3-Table 5 (see also [22]). Finally, in case a timeout is set at the Kafka publisher or consumer side, this should be large enough to allow the completion of the transmission or reception operation.

TABLE 3: KAFKA TOPIC CONFIGURATION FOR HANDLING LARGE RECORDS (MESSAGES)

Configuration	Value
max.message.bytes	104857600
segment.bytes	1073741824

TABLE 4: KAFKA PRODUCER CONFIGURATION FOR SENDING LARGE RECORDS (MESSAGES)

Configuration	Spring Kafka constant	Value
max.request.size	ProducerConfig.MAX_REQUEST_SIZE_CONFIG	52428800
buffer.memory	ProducerConfig.BUFFER_MEMORY_CONFIG	52428800
compression.type	ProducerConfig.COMPRESSION_TYPE_CONFIG	snappy

TABLE 5: KAFKA CONSUMER CONFIGURATION FOR RECEIVING LARGE RECORDS (MESSAGES)

Configuration	Spring Kafka constant	Value
fetch.max.bytes	ConsumerConfig.FETCH_MAX_BYTES_CONFIG	52428800
max.partition.fetch.bytes	ConsumerConfig.MAX_PARTITION_FETCH_BYTES_CONFIG	52428800
compression.type	ProducerConfig.COMPRESSION_TYPE_CONFIG	snappy

4.4 Monitoring data layer

At the monitoring data layer, slightly different approaches have been followed for the three real demo sites. This is mainly due to the different combination of sensors, networking technologies and protocols used in each deployment. However, these variations demonstrate the modularity of the LYSIS platform since the integration of monitoring data from all demo sites is seamless. The following sections present the established data flows for each site.

4.4.1 VVV demo site

Figure 13 depicts the data flow for the VVV demo site. The VVV Adapter mainly consists of the following modules:

- EnergyCollector: Reads periodically new energy measurements from the HAM Systems platform either using the HTTP API [23] or the hamapi Python package [24] and integrates them into the LYSIS platform by sending the data to the *meas.energy.vvv* Kafka topic and storing them in the *energy_vvv* InfluxDB measurement.
- IeqCollector (for each monitoring zone/room): Reads periodically new IEQ measurements from NTUA's InfluxDB and integrates them into the LYSIS platform by sending the data to the *meas.ieq.vvv* Kafka topic and storing them in the *ieq_vvv* InfluxDB measurement.
- WeatherCollector: Reads periodically new weather measurements from the AWEKAS [25] HTTP API and integrates them into the LYSIS platform by sending the data to the *meas.weather.vvv* Kafka topic and storing them in the *weather_vvv* InfluxDB measurement.

The raw monitoring data from the InfluxDB are directly visualized through dashboards. In addition, they are used by KPI Calculator Flux and Python scripts for the computation of relevant KPIs which are also presented in graphs for visual inspection. More information about the JSON Schemas used for the Kafka records of each topic, as well as the fields and tags of the corresponding InfluxDB measurements are listed in Appendix A (section 7). Further, the *PLURAL_D5_2_ANNEX_VVV_Adapter_Doc.pdf* provides a more in-depth documentation of the various modules and scripts comprising the VVV Adapter.

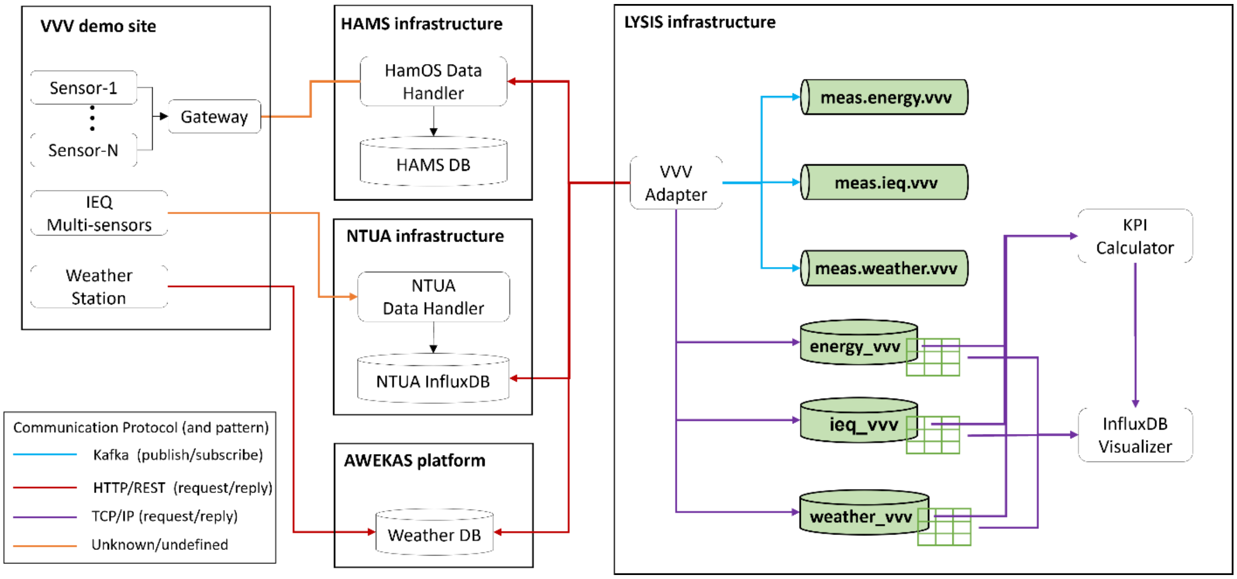


FIGURE 13: MONITORING DATA LAYER DATAFLOW OF VVV SITE

4.4.2 Kasava demo site

Figure 14 depicts the data flow for the Kasava demo site. The Kasava Adapter mainly consists of the following modules:

- EnergyCollectors for the building and for each apartment
- IeqCollectors for each monitoring zone/room)
- WeatherCollector for the whole site

Each collector read periodically new energy, IEQ and weather measurements using the UCEEB REST API [26] (login is required to view it) and integrate them into the LYSIS platform by sending the data to the respective Kafka topics (*meas.energy.kasava*, *meas.ieq.kasava*, *meas.weather.kasava*) and storing them in the InfluxDB measurements (*energy_kasava*, *ieq_kasava*, *weather_kasava*).

The raw monitoring data from the InfluxDB are directly visualized through dashboards. In addition, they are used by KPI Calculator Flux and Python scripts for the computation of relevant KPIs which are also presented in graphs for visual inspection. More information about the JSON Schemas used for the Kafka records of each topic, as well as the fields and tags of the corresponding InfluxDB measurements are listed in Appendix B (section 8). Further, the *PLURAL_D5_2_ANNEX_Kasava_Adapter_Doc.pdf* provides a more in-depth documentation of the various modules and scripts comprising the Kasava Adapter.

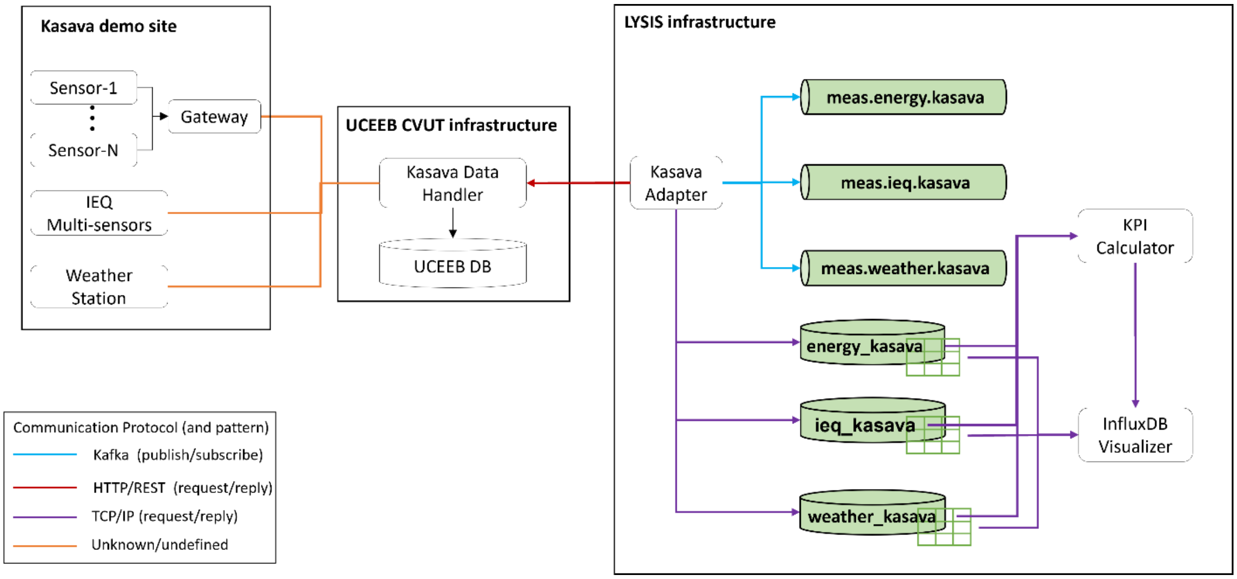


FIGURE 14: MONITORING DATA LAYER DATAFLOW OF KASAVA SITE

4.4.3 Terrassa demo site

Figure 15 depicts the data flow for the Terrassa demo site. The Terrassa Adapter reads continuously new energy, IEQ and weather records from the respective Kafka topics (*meas.energy.terrassa*, *meas.ieq.terrassa*, *meas.weather.terrassa*) published by the RDmes Terrassa Data Handler, and stores them in InfluxDB measurements (*energy_terrassa*, *ieq_terrassa*, *weather_terrassa*).

As it is the case for the other two demo sites, monitoring data from the InfluxDB are directly visualized through dashboards and are used by KPI Calculator scripts for the computation of relevant KPIs and their visual representation. More information about the JSON Schemas used for the Kafka records of each topic, as well as the fields and tags of the corresponding InfluxDB measurements are listed in Appendix C (section 9). Further, the *PLURAL_D5_2_ANNEX_Terrassa_Adapter_Doc.pdf* provides a more in-depth documentation of the various modules and scripts comprising the Terrassa Adapter.

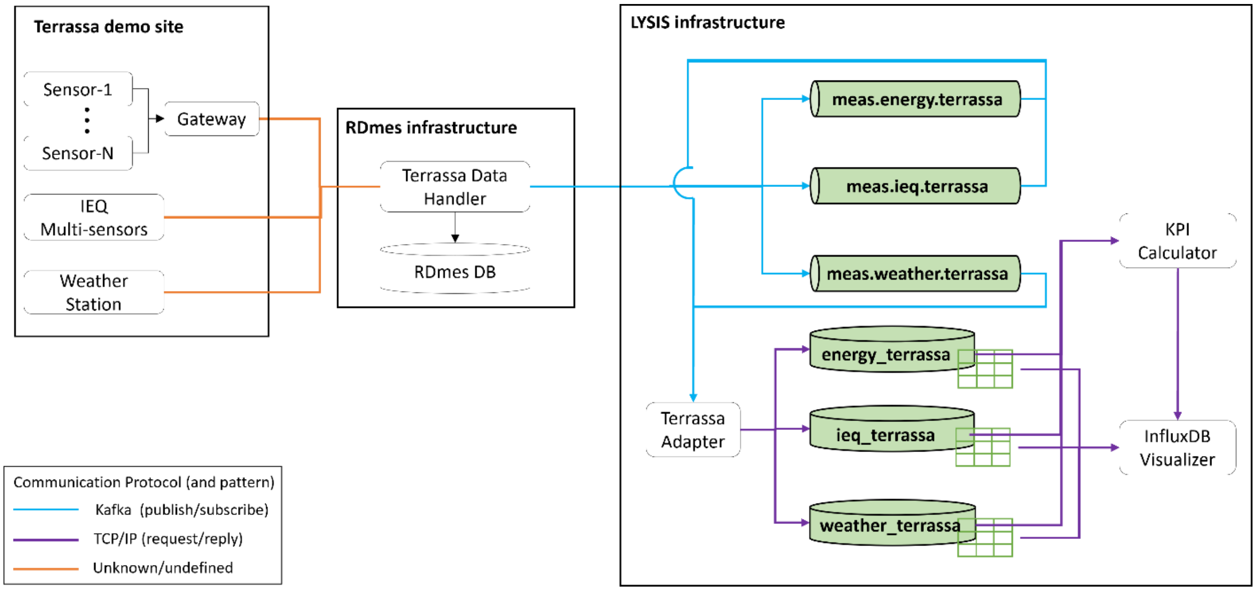


FIGURE 15: MONITORING DATA LAYER DATAFLOW OF TERRASSA SITE

4.5 Simulation data layer

Figure 16 depicts the data flow at the simulation data layer where the KPIs calculated using the various simulation tools are inserted in the LYSIS platform and are made available to the end users. In particular, the TCQ tool running on ITeC's infrastructure calculates LCA/LCC-related KPIs for the various PnU alternatives which are stored in a spreadsheet following a pre-specified format. In the following, this is transferred in the LYSIS infrastructure where the LCA KPI Importer publishes the received entries in the *kpi.lca.tcq* Kafka topic. In a similar manner, data coming from TRNSYS simulations including technical (mostly energy related) KPIs, comfort KPIs as well as economic KPIs are stored in spreadsheet file of a specific template and communicated to the LYSIS infrastructure where the TRNSYS KPI Importer integrates them in the platform through the *kpi.trnsys* Kafka topic. KPIs coming from both of these topics are collected by the MODEST Backend which combines them and ultimately makes them accessible via an HTTP/REST API. Finally, the MODEST Frontend acts according to end user's requirements and criteria, performing the corresponding requests, retrieving and visualizing the respective data.

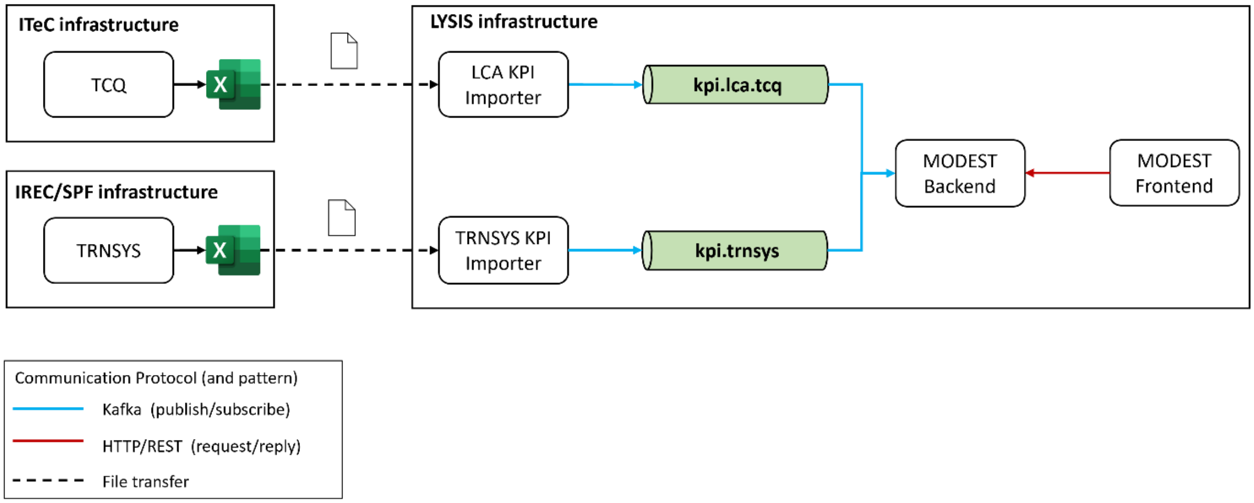


FIGURE 16: SIMULATION DATA LAYER DATAFLOW

4.6 Source code repositories

The source code of the developed components is stored in GitHub at private repositories under the **plural-eu** organization. For most of the core components comprising the LYSIS platform, currently, only the INTRA team has authorization access and can update them. In addition, some helper components have been developed for functionality demonstration purposes and shared with other members of the consortium. Table 6 below lists the current GitHub repositories.

TABLE 6: PLURAL LYSIS GITHUB REPOSITORIES

Component	Repository	Description
VVV Adapter	https://github.com/plural-eu/adapter-vvv	Adapter (connector) for the integration of monitoring data from the VVV demo site
Kasava Adapter	https://github.com/plural-eu/adapter-kasava	Adapter (connector) for the integration of monitoring data from the Kasava demo site
Terrassa Adapter	https://github.com/plural-eu/adapter-terrassa	Adapter (connector) for the integration of monitoring data from the Terrassa demo site
Demo Terrassa Data Handler	https://github.com/plural-eu/kafka-demo-terrassa	Kafka demo code for the publishing of monitoring data from the Terrassa demo site
Weather Data Processor	https://github.com/plural-eu/kafka-weather-terrassa	Tool for weather data retrieval and storage in CSV format
KPI Calculators/ Importers	https://github.com/plural-eu/kpi-calculator	KPIs calculators based on monitoring data and KPI importers based on simulation data
MODEST Backend	https://github.com/plural-eu/modest-backend	Retrieval, processing and serving of LCA and TRNSYS-related KPIs
MODEST Frontend	https://github.com/plural-eu/ui-modest-lysis	GUI for modest tool and UI visualization on data received from InfluxDB (lysis)

4.7 Management UIs

The main monitoring tools used for the administration, management and continuous checking of the platform's operational status are the Hetzner Cloud Console, AKHQ and the InfluxDB UI described in the following.

4.7.1 Hetzner Cloud Console

The Hetzner Cloud Console is the administrator interface of Hetzner Cloud that hosts the server VMs of the LYSIS infrastructure. It provides the ability to manage the server's networking functionality, take backups and snapshots of a specific server or add a disk volume in case the storage requirements are increased. In addition, as shown in Figure 17, it has an UI for monitoring the CPU usage, disk throughput, disk input/output operations per second (IOPS), the input/output network traffic and the respective packets per second (PPS). The graphs can present monitoring information for a selectable period of time that can be the last month, week, day, hour or even depict live information updated in real time. Finally, the UI provides also a VNC console giving the ability to login directly to a server to perform debugging or administrative actions. Note that for the purposes of the project, all these controls are used in conjunction with traditional Sysadmin tools.

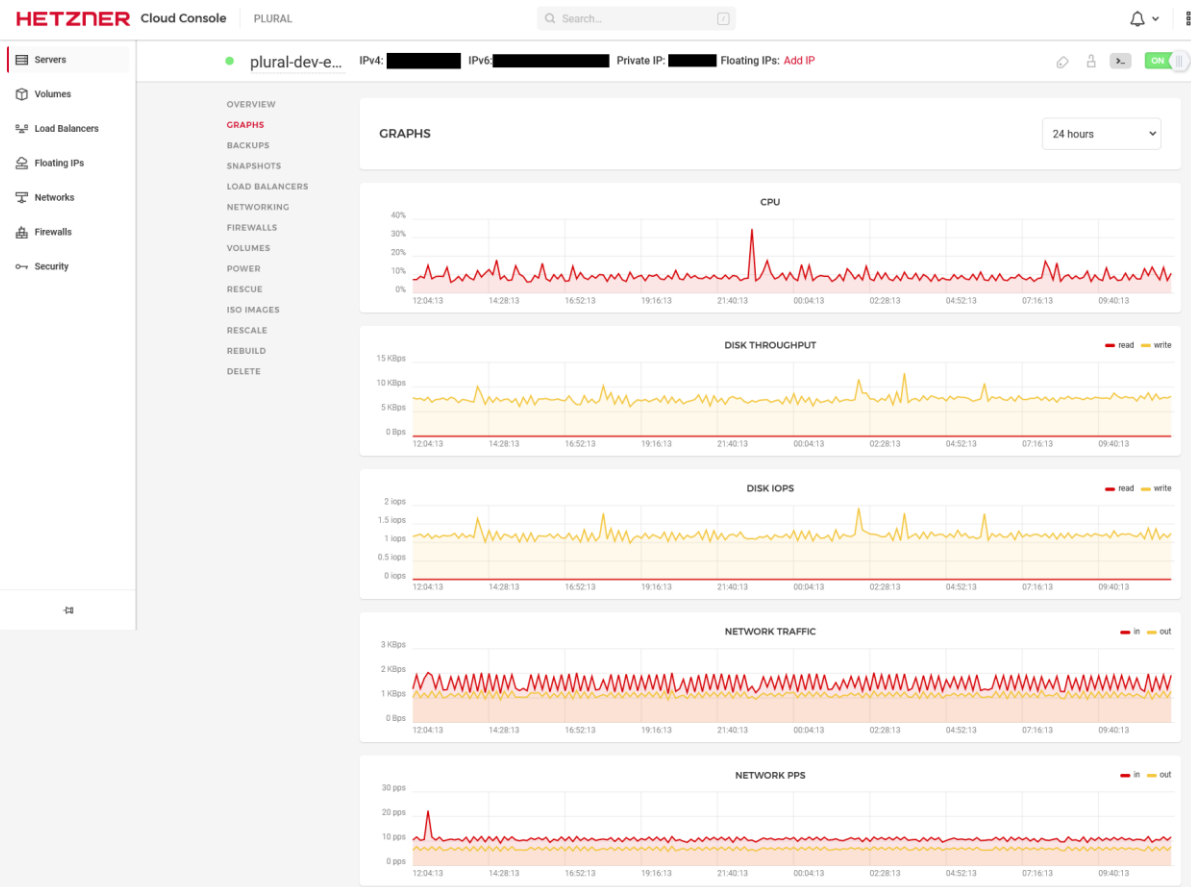
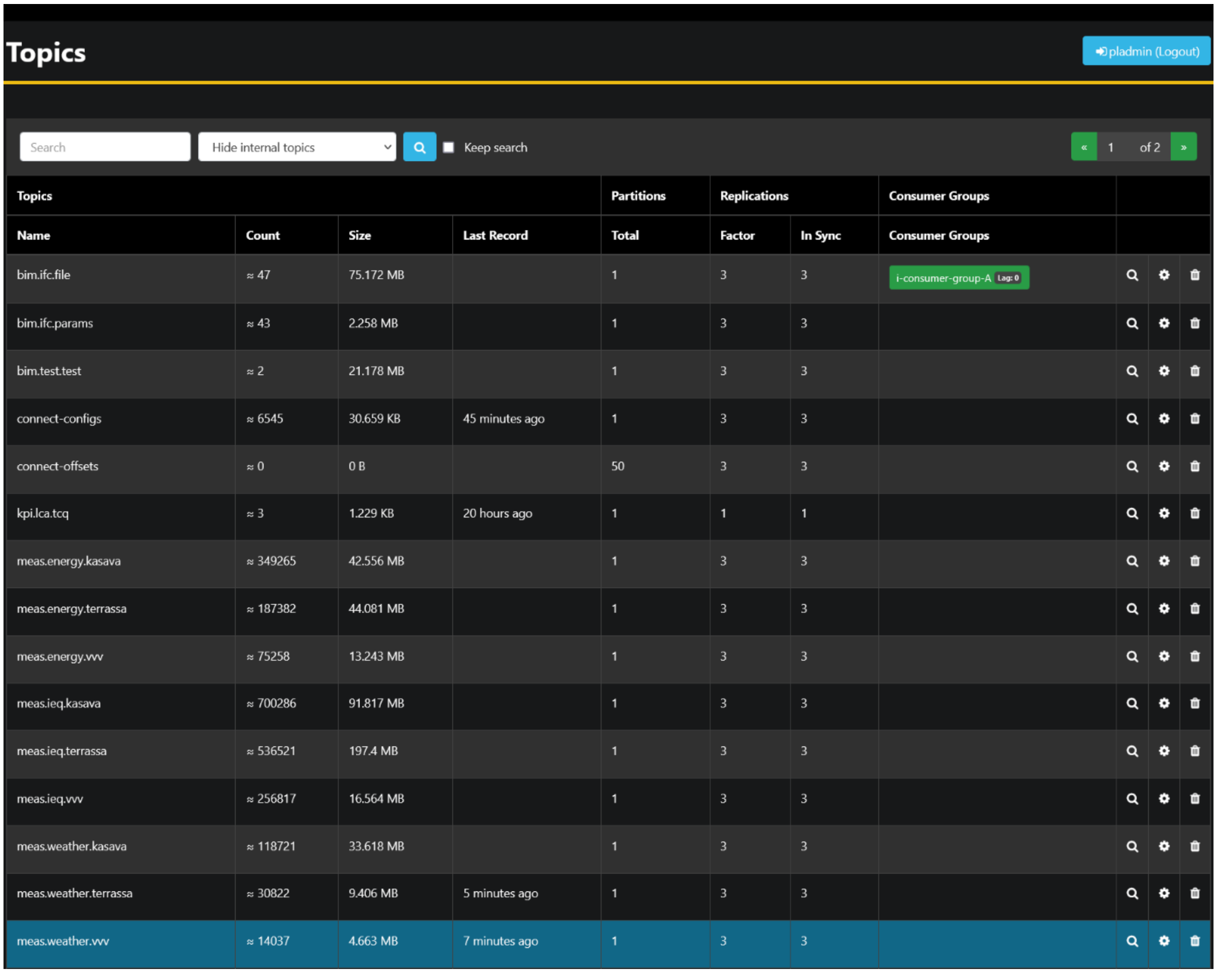


FIGURE 17: HETZNER CLOUD CONSOLE GRAPHS

4.7.2 AKHQ

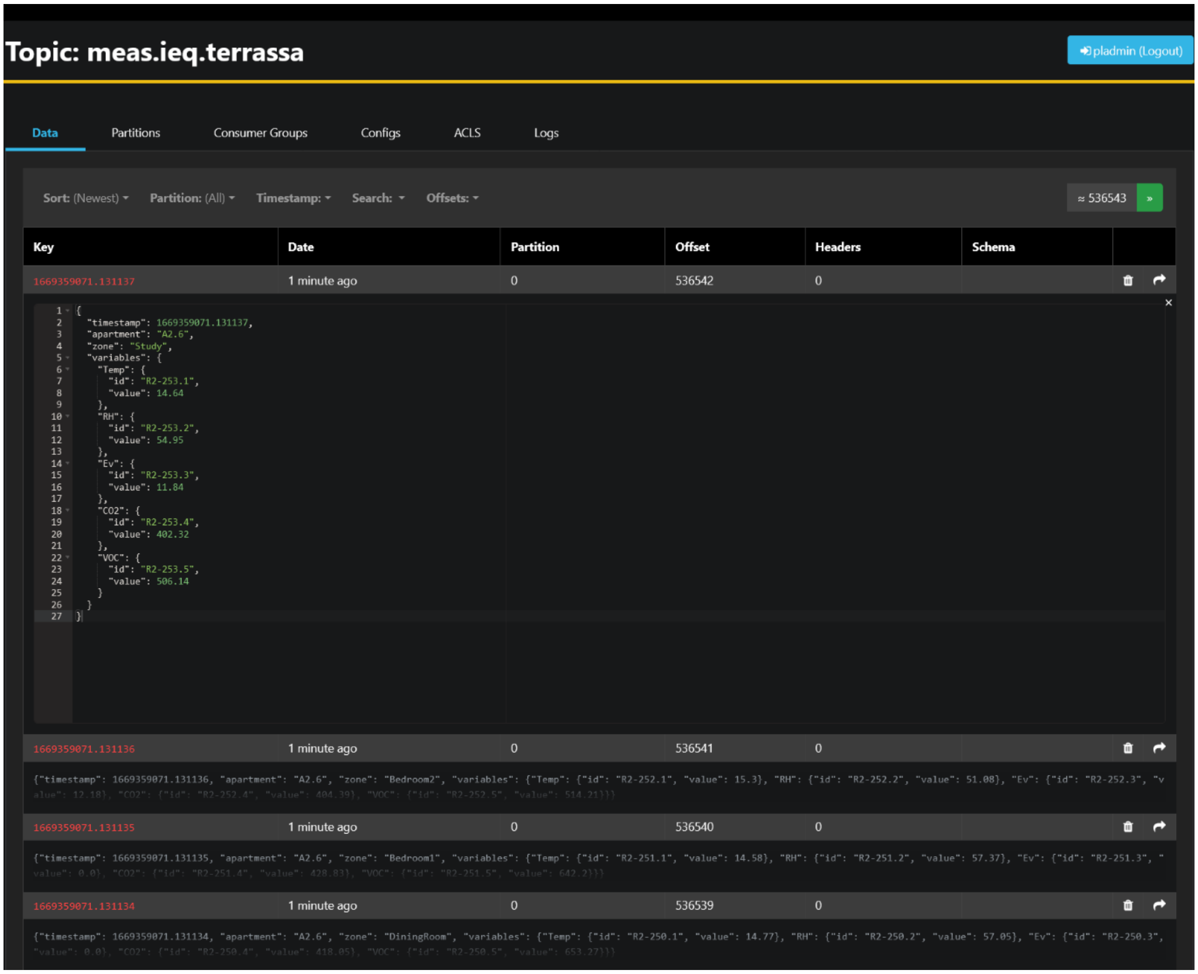
AKHQ [27] is used as part of the StreamHandler's UI functionality. It provides a Kafka GUI for the Apache Kafka deployment to manage topics, topics data, consumer groups and synchronization status between the Kafka brokers. Figure 18 shows the AKHQ homepage where all the available topics are listed and for each one of them there are various information including the total number of records (i.e., messages) received, the respective occupied disk space, the required replication factor (i.e., the number of Kafka brokers to store and serve the topic records) and the number of synchronized Kafka brokers (i.e., have already received the latest topic record).



Topics				Partitions	Replications		Consumer Groups			
Name	Count	Size	Last Record	Total	Factor	In Sync	Consumer Groups			
bim.ifc.file	≈ 47	75.172 MB		1	3	3	i-consumer-group-A Lag: 0	🔍	⚙	🗑
bim.ifc.params	≈ 43	2.258 MB		1	3	3		🔍	⚙	🗑
bim.test.test	≈ 2	21.178 MB		1	3	3		🔍	⚙	🗑
connect-configs	≈ 6545	30.659 KB	45 minutes ago	1	3	3		🔍	⚙	🗑
connect-offsets	≈ 0	0 B		50	3	3		🔍	⚙	🗑
kpi.lca.tcq	≈ 3	1.229 KB	20 hours ago	1	1	1		🔍	⚙	🗑
meas.energy.kasava	≈ 349265	42.556 MB		1	3	3		🔍	⚙	🗑
meas.energy.terrassa	≈ 187382	44.081 MB		1	3	3		🔍	⚙	🗑
meas.energy.vvv	≈ 75258	13.243 MB		1	3	3		🔍	⚙	🗑
meas.ieq.kasava	≈ 700286	91.817 MB		1	3	3		🔍	⚙	🗑
meas.ieq.terrassa	≈ 536521	197.4 MB		1	3	3		🔍	⚙	🗑
meas.ieq.vvv	≈ 256817	16.564 MB		1	3	3		🔍	⚙	🗑
meas.weather.kasava	≈ 118721	33.618 MB		1	3	3		🔍	⚙	🗑
meas.weather.terrassa	≈ 30822	9.406 MB	5 minutes ago	1	3	3		🔍	⚙	🗑
meas.weather.vvv	≈ 14037	4.663 MB	7 minutes ago	1	3	3		🔍	⚙	🗑

FIGURE 18: AKHQ HOMEPAGE WITH THE LIST OF PLURAL TOPICS

More fine-grained monitoring features are provided by selecting a topic. As shown in Figure 19, the list of received records can be viewed and searched according to specific criteria, the status of consumer groups can be reviewed and the topic’s configuration like the retention policy, the maximum allowed record size and the compression algorithm, can be updated according to the current requirements.



Topic: meas.ieq.terrassa

Key	Date	Partition	Offset	Headers	Schema
1669359071.131137	1 minute ago	0	536542	0	

```
{
  "timestamp": 1669359071.131137,
  "apartment": "A2.6",
  "zone": "Study",
  "variables": {
    "Temp": {
      "id": "R2-253.1",
      "value": 14.64
    },
    "RH": {
      "id": "R2-253.2",
      "value": 54.95
    },
    "Ev": {
      "id": "R2-253.3",
      "value": 11.84
    },
    "CO2": {
      "id": "R2-253.4",
      "value": 402.32
    },
    "VOC": {
      "id": "R2-253.5",
      "value": 506.14
    }
  }
}
```

Key	Date	Partition	Offset	Headers	Schema
1669359071.131136	1 minute ago	0	536541	0	

```
{"timestamp": 1669359071.131136, "apartment": "A2.6", "zone": "Bedroom2", "variables": {"Temp": {"id": "R2-252.1", "value": 15.3}, "RH": {"id": "R2-252.2", "value": 51.08}, "Ev": {"id": "R2-252.3", "v
alue": 12.18}, "CO2": {"id": "R2-252.4", "value": 404.39}, "VOC": {"id": "R2-252.5", "value": 514.21}}}
```

Key	Date	Partition	Offset	Headers	Schema
1669359071.131135	1 minute ago	0	536540	0	

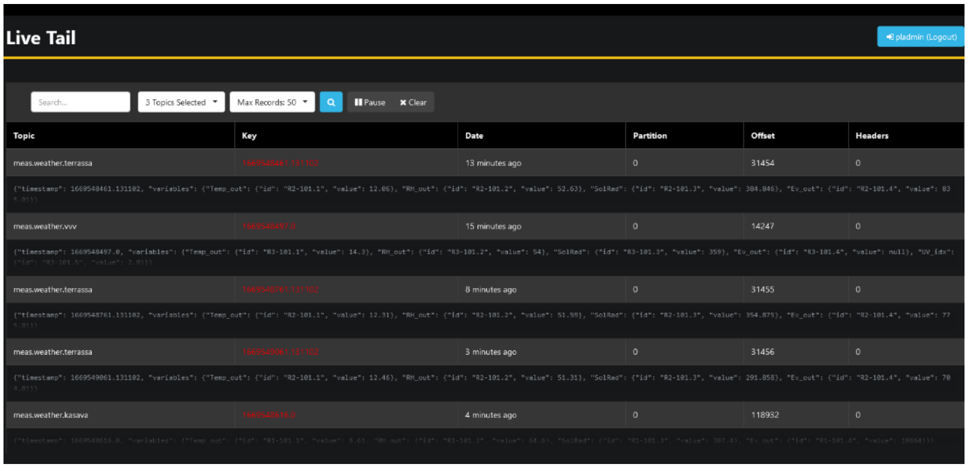
```
{"timestamp": 1669359071.131135, "apartment": "A2.6", "zone": "Bedroom1", "variables": {"Temp": {"id": "R2-251.1", "value": 14.58}, "RH": {"id": "R2-251.2", "value": 57.37}, "Ev": {"id": "R2-251.3", "
value": 0.0}, "CO2": {"id": "R2-251.4", "value": 428.83}, "VOC": {"id": "R2-251.5", "value": 642.2}}}
```

Key	Date	Partition	Offset	Headers	Schema
1669359071.131134	1 minute ago	0	536539	0	

```
{"timestamp": 1669359071.131134, "apartment": "A2.6", "zone": "DiningRoom", "variables": {"Temp": {"id": "R2-250.1", "value": 14.77}, "RH": {"id": "R2-250.2", "value": 57.05}, "Ev": {"id": "R2-250.3",
"value": 0.0}, "CO2": {"id": "R2-250.4", "value": 418.05}, "VOC": {"id": "R2-250.5", "value": 653.27}}}
```

FIGURE 19: AKHQ TOPIC RECORDS INSPECTION (IEQ MEASUREMENTS FOR TERRASSA)

Further, the so-called *Live Tail* functionality provides the ability to observe in real-time the reception of records for a selected set of topics. For instance, Figure 20 shows the live reception of messages in the weather-related topics for all three real demo sites. Note that for the purposes of the project, all these controls are used in conjunction with Kafka’s command line tools [28] [29] and custom, in-house scripts and tools developed for the management and administration tasks.



Live Tail

pladmin (Logout)

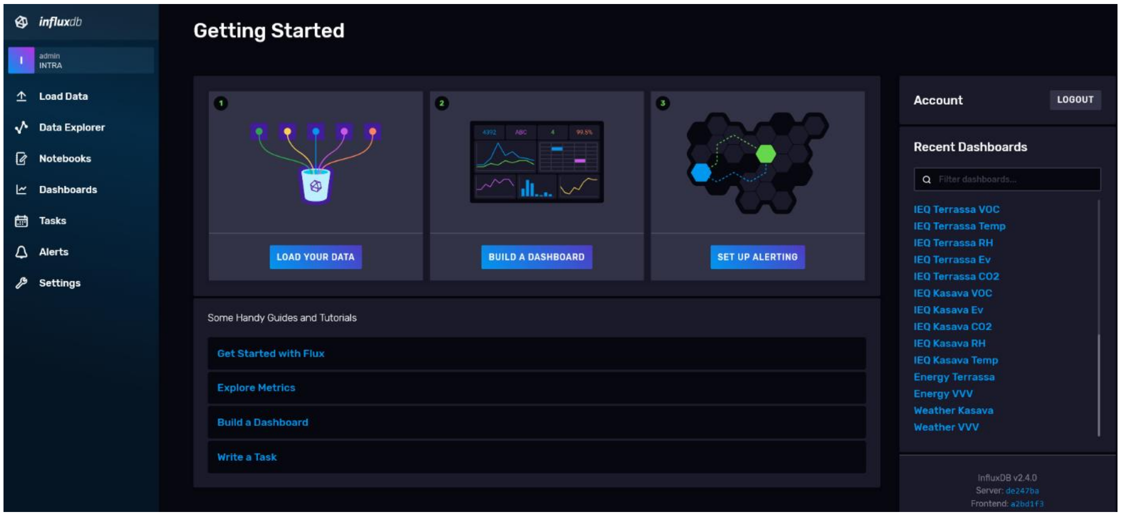
Search... | 3 Topics Selected | Max Records: 50 | Pause | Clear

Topic	Key	Date	Partition	Offset	Headers
meas.weather.terrassa	1669548461.131102	13 minutes ago	0	31454	0
{"timestamp": 1669548461.131102, "variables": {"Temp_out": {"id": "R2-101.1", "value": 12.06}, "RH_out": {"id": "R2-101.2", "value": 52.63}, "SolRad": {"id": "R2-101.3", "value": 384.846}, "Ev_out": {"id": "R2-101.4", "value": 83 5.0}}}					
meas.weather.vvv	1669548497.0	15 minutes ago	0	14247	0
{"timestamp": 1669548497.0, "variables": {"Temp_out": {"id": "R3-101.1", "value": 14.3}, "RH_out": {"id": "R3-101.2", "value": 54}, "SolRad": {"id": "R3-101.3", "value": 359}, "Ev_out": {"id": "R3-101.4", "value": null}, "UV_idx": {"id": "R3-101.5", "value": 2.8}}}					
meas.weather.terrassa	1669548761.131102	8 minutes ago	0	31455	0
{"timestamp": 1669548761.131102, "variables": {"Temp_out": {"id": "R2-101.1", "value": 12.31}, "RH_out": {"id": "R2-101.2", "value": 51.59}, "SolRad": {"id": "R2-101.3", "value": 354.875}, "Ev_out": {"id": "R2-101.4", "value": 77 5.0}}}					
meas.weather.terrassa	1669549061.131102	3 minutes ago	0	31456	0
{"timestamp": 1669549061.131102, "variables": {"Temp_out": {"id": "R2-101.1", "value": 12.46}, "RH_out": {"id": "R2-101.2", "value": 51.31}, "SolRad": {"id": "R2-101.3", "value": 291.858}, "Ev_out": {"id": "R2-101.4", "value": 70 4.0}}}					
meas.weather.kasava	1669548616.0	4 minutes ago	0	118932	0
{"timestamp": 1669548616.0, "variables": {"Temp_out": {"id": "R1-101.1", "value": 8.6}, "RH_out": {"id": "R1-101.2", "value": 64.6}, "SolRad": {"id": "R1-101.3", "value": 307.4}, "Ev_out": {"id": "R1-101.4", "value": 19864}}}					

FIGURE 20: AKHQ LIVE TAIL OF SELECTED TOPICS (WEATHER-RELATED)

4.7.3 InfluxDB UI

The InfluxDB UI [30] is used for monitoring the various sensor measurements coming from the real demo sites, which are stored in the InfluxDB. As shown in Figure 21, it provides a web admin interface that gives the ability to generate API tokens for providing access to the organization's buckets, to create tasks for performing periodic checks on the received data and to create alerts for sending notifications in case specific conditions apply. It also provides tools for building custom dashboards to visualize the data through queries.



Getting Started

Some Handy Guides and Tutorials

- Get Started with Flux
- Explore Metrics
- Build a Dashboard
- Write a Task

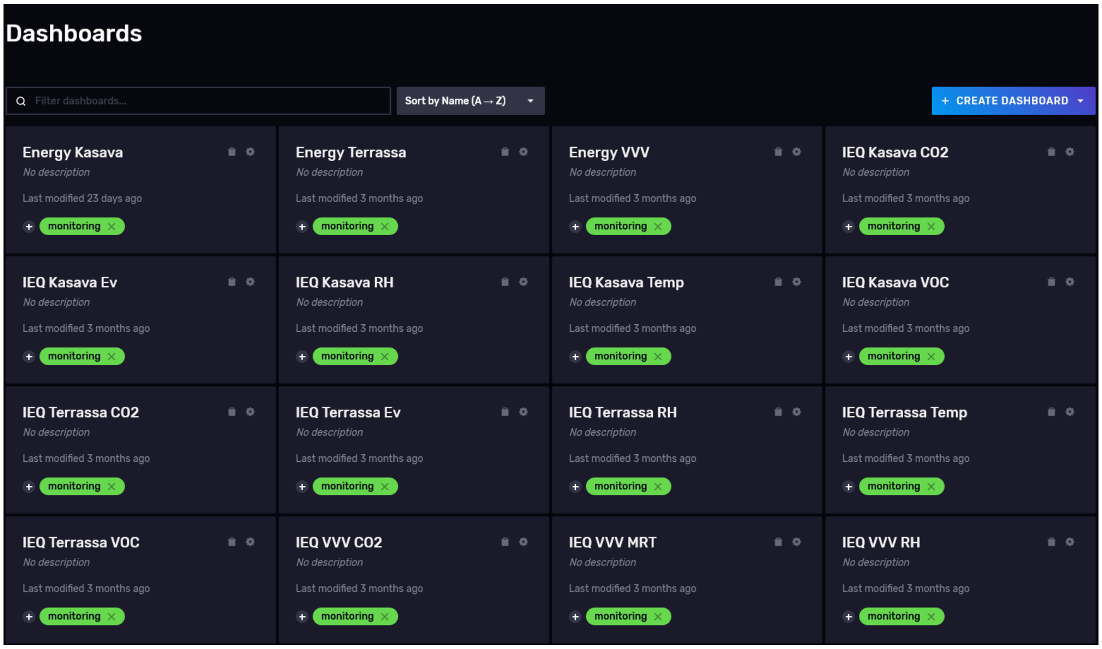
Recent Dashboards

- IEQ Terrassa VOC
- IEQ Terrassa Temp
- IEQ Terrassa RH
- IEQ Terrassa Ev
- IEQ Terrassa CO2
- IEQ Kasava VOC
- IEQ Kasava Ev
- IEQ Kasava CO2
- IEQ Kasava RH
- IEQ Kasava Temp
- Energy Terrassa
- Energy VVV
- Weather Kasava
- Weather VVV

InfluxDB v2.4.0
Server: de247ba
Frontend: a2bd1f3

FIGURE 21: INFLUXDB UI HOMEPAGE

Figure 22 shows the a (non-complete) list of the dashboards created for the monitoring and visualization for the data coming from all the demo sites, while Figure 23 presents as an example the weather data received from Terrassa during the last month. Note that as it was the case for the previous two management UIs, the above controls are used in conjunction with the influx command line interface (CLI) [31] and custom-built tools.



Dashboards

Energy Kasava
No description
Last modified 23 days ago
monitoring

Energy Terrassa
No description
Last modified 3 months ago
monitoring

Energy VVV
No description
Last modified 3 months ago
monitoring

IEQ Kasava CO2
No description
Last modified 3 months ago
monitoring

IEQ Kasava Ev
No description
Last modified 3 months ago
monitoring

IEQ Kasava RH
No description
Last modified 3 months ago
monitoring

IEQ Kasava Temp
No description
Last modified 3 months ago
monitoring

IEQ Kasava VOC
No description
Last modified 3 months ago
monitoring

IEQ Terrassa CO2
No description
Last modified 3 months ago
monitoring

IEQ Terrassa Ev
No description
Last modified 3 months ago
monitoring

IEQ Terrassa RH
No description
Last modified 3 months ago
monitoring

IEQ Terrassa Temp
No description
Last modified 3 months ago
monitoring

IEQ Terrassa VOC
No description
Last modified 3 months ago
monitoring

IEQ VVV CO2
No description
Last modified 3 months ago
monitoring

IEQ VVV MRT
No description
Last modified 3 months ago
monitoring

IEQ VVV RH
No description
Last modified 3 months ago
monitoring

FIGURE 22: INFLUXDB DASHBOARDS LIST FOR VISUALIZATION OF MONITORING DATA

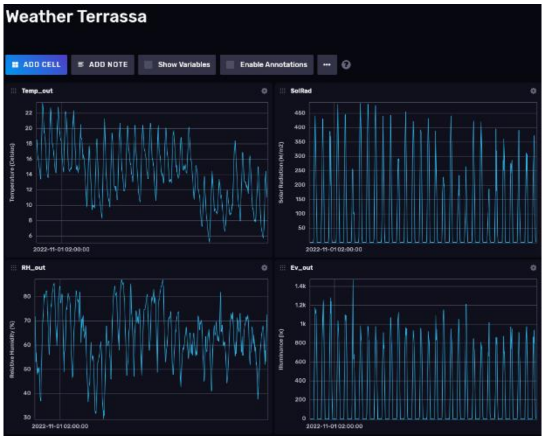


FIGURE 23: INFLUXDB DASHBOARD VISUALIZING WEATHER DATA (FOR TERRASSA)

5. MODEST and Supporting Components

This section focuses on the calculation/retrieval of KPIs (section 5.1) and the GUI created for the visualization of the various types of information (section 5.2). Note, that for the latter a continuous improvement process is followed in order to incorporate input from the relevant stakeholders that will be the end users of the system and ultimately present the data in the most intuitive and user-friendly way.

5.1 KPIs

In terms of origin, the provided KPIs can be of two kinds:

- **Monitoring data related**: These involve energy and IEQ specific KPIs that are calculated based on the various monitoring measurements coming from the sensors installed in the three real demo sites. Their purpose is to assess the status of the buildings in their original, pre-renovation state (also noted as pre-PLURAL phase) and to deduce the actual gains due to the overall upgrades and the deployed PnU kits after the renovation (also noted as post-PLURAL phase).
- **Simulation data related**: These involve LCA/LCC, energy, IEQ and economic KPIs calculated by the simulation tools for a variety of PnU alternatives. Their purpose is to evaluate different combinations of materials and their potential effects in the renovation activities of a building and thus to assist in the decision-making process.

Table 7 lists the monitoring related KPIs calculated at the current phase of the project, which belongs to the pre-PLURAL stage. Their calculation is performed by retrieving the measurements from the InfluxDB and using either Flux scripts through the InfluxDB UI for their direct visualization or Python scripts that store the computed values in new measurements that are then visualized through dashboards in the InfluxDB UI. Table 8 lists the KPIs calculated by ITeC's TCQ tool and through TRNSYS simulations for the PnU alternatives of each demo site. For a more detailed description regarding their actual calculation please refer to *"D1.4 – Key performance indicators, system components and performance targets"*.

TABLE 7: LIST OF KPIS BASED ON MONITORING DATA

KPI	Unit(s) or period
Primary Energy Consumption	kWh/m^{2}yr and kWh/m^{2}mo
Non-renewable Primary Energy	kWh/m^{2}yr and kWh/m^{2}mo
Renewable Energy Ratio (RER)	Yearly and monthly
Thermal Energy Consumption	kWh/m^{2}yr and kWh/m^{2}mo

Electric Consumption for mechanical Ventilation	kWh/m^{2}yr and kWh/m^{2}mo
Electric Consumption for DHW	kWh/m^{2}yr and kWh/m^{2}mo
Solar Energy for DHW	kWh/m^{2}yr and kWh/m^{2}mo
CO2 Index (Air Quality)	Yearly and monthly
VOC	ppm
Thermal Comfort- Degree Hours Exceed	Yearly and monthly
Predicted Mean Vote (PMV)	Yearly and monthly
Visual Comport – Daylight Factor	Yearly and monthly

TABLE 8: LIST OF KPIS BASED ON SIMULATION DATA (PARTIAL)

KPI	Unit(s) or period
nZEB Compliance	-
Primary Energy Demand/Consumption (Ped/Pec)	kWh/m^{2}yr
Non-renewable Primary Energy	kWh/m^{2}yr
Renewable Energy Ratio (RER)	Yearly and monthly
Thermal Energy Demand	kWh/m^{2}yr
Thermal Energy Consumption	kWh/m^{2}yr
Electric Consumption for mechanical Ventilation	kWh/m^{2}yr
Electric Consumption for DHW	kWh/m^{2}yr
Solar Energy for DHW	kWh/m^{2}yr
Thermal Transmittance/Resistance of a Wall system	W/m^{2}K
Thermal Transmittance of windows	W/m^{2}K
CO2 Index (Air Quality)	ppm
Predicted Mean Vote	Yearly
Total Annual Cost (TAC)	Euros/m^{2}yr
Global Warming Potential	kgCO2 eq/m^2
Depletion of abiotic resources – elements	kg Sb-eq/m^2
Depletion of abiotic resources – fossil fuel	MJ-eq/m^2
Specific CO2 emissions	g/ m^2

5.2 LYSIS-MODEST UI

The LYSIS-MODEST UI tool is designed to work in conjunction with the core functionality of the LYSIS platform, the LYSIS InfluxDB and the MODEST Backend. Figure 24 presents the sequence diagram which illustrates the sequence of interactions between the various LYSIS-MODEST UI sub-components and external components or entities. It offers two basic user interfaces, the LYSIS UI and the MODEST UI which are further discussed in the following.

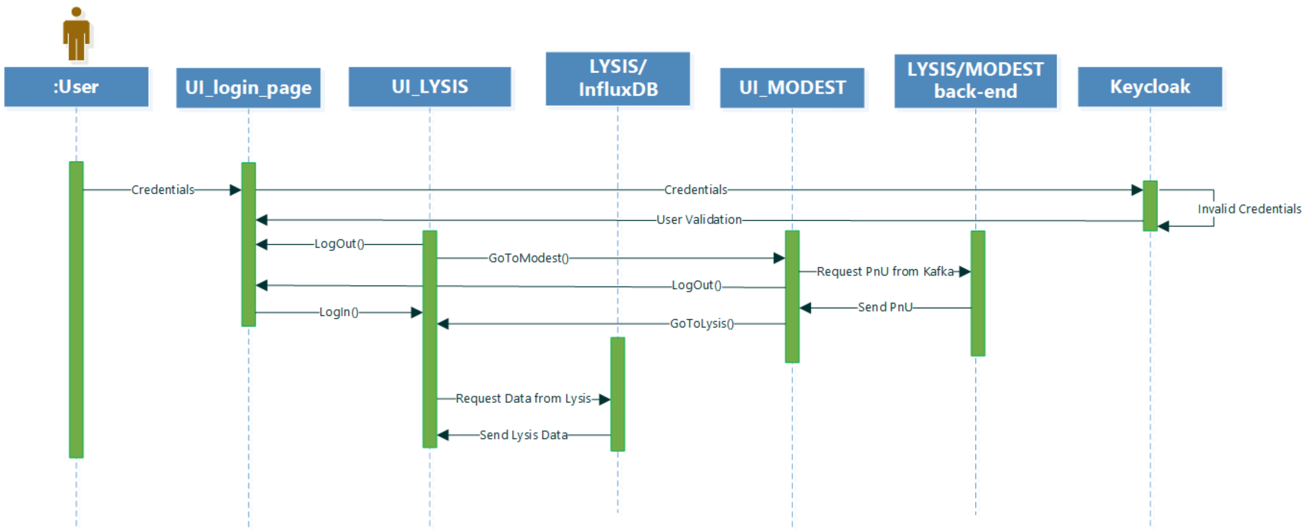
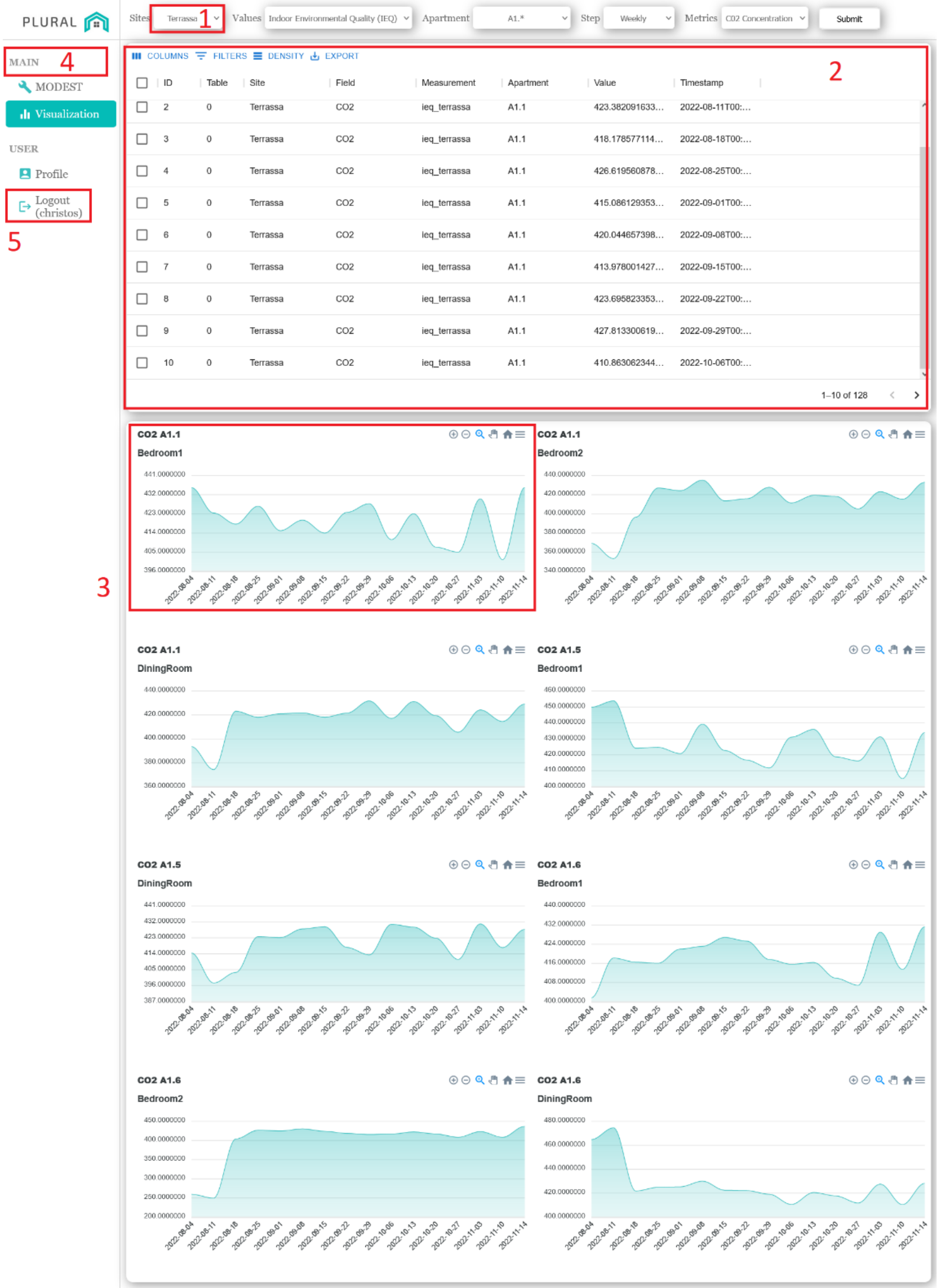


FIGURE 24: LYSIS-MODEST UI SEQUENCE DIAGRAM

5.2.1 LYSIS UI

The main purpose of the LYSIS UI is to offer to the end users, in this case the consortium or any external entity that may require access to the data, a seamless way to visualize, inspect as well as save and keep track of any requested information regarding the pilot sites in all three locations without accessing directly the complex and sensitive information residing in the InfluxDB.

Through the LYSIS UI tool, the user can choose specific fields using dropdown menus and view detailed information regarding the data of each site. Internally, InfluxDB queries are executed, retrieve the required information, and visualize it in chart as well as in table format. In addition, the user can download the data in CSV format and store them on their local machine for later inspection and analysis. Finally, the tool supports integration with Keycloak for the authentication/authorization of the users, ensuring this way that unauthorized end users will not have access to the information.



Sites Terrassa | Values Indoor Environmental Quality (IEQ) | Apartment A1.* | Step Weekly | Metrics C02 Concentration

□	ID	Table	Site	Field	Measurement	Apartment	Value	Timestamp
□	2	0	Terrassa	CO2	ieq_terrassa	A1.1	423.382091633...	2022-08-11T00:...
□	3	0	Terrassa	CO2	ieq_terrassa	A1.1	418.178577114...	2022-08-18T00:...
□	4	0	Terrassa	CO2	ieq_terrassa	A1.1	426.619560878...	2022-08-25T00:...
□	5	0	Terrassa	CO2	ieq_terrassa	A1.1	415.086129353...	2022-09-01T00:...
□	6	0	Terrassa	CO2	ieq_terrassa	A1.1	420.044657398...	2022-09-08T00:...
□	7	0	Terrassa	CO2	ieq_terrassa	A1.1	413.978001427...	2022-09-15T00:...
□	8	0	Terrassa	CO2	ieq_terrassa	A1.1	423.695823353...	2022-09-22T00:...
□	9	0	Terrassa	CO2	ieq_terrassa	A1.1	427.813300619...	2022-09-29T00:...
□	10	0	Terrassa	CO2	ieq_terrassa	A1.1	410.863062344...	2022-10-06T00:...

1–10 of 128

FIGURE 25: LYSIS UI VISUALIZING MONITORING DATA

Figure 25 displays the appearance and functionality of the UI implementation marking the various controls and visualization areas:

1. **Dropdown menus** from where the user can choose:
    a. **Site:** the demo site for which they are interested in retrieving information
    b. **Values** (e.g., Energy Consumption, Indoor Environmental Quality, etc.)
    c. **Apartment:** the specific area for which they are interested to (available only with “Values” equal to “Indoor Environmental Quality”)
    d. **Step**: the time window of the retrieved data (e.g., 15 minutes, Hourly, Daily, Weekly, Monthly)
    e. **Metric** which is directly linked to the Value attribute
2. **Table** where the results are presented. The table implemented here provides the users with the ability to sort, filter or even exclude specified columns and finally export in a CSV format all the results or even specific rows.
3. **Charts** that act as means of visualization of the raw data. The charts are automatically generated and display the change of metrics through time as well as info regarding the area they correspond to.
4. **Sidebar** displaying the available options to the user.
5. **Login/Logout function** implemented with Keycloak.

Figure 26 provides a close look at the generated charts where the marked fields indicate:

1. The area of the site, as well as specific apartment.
2. Zoom-in zoom-out functionality.

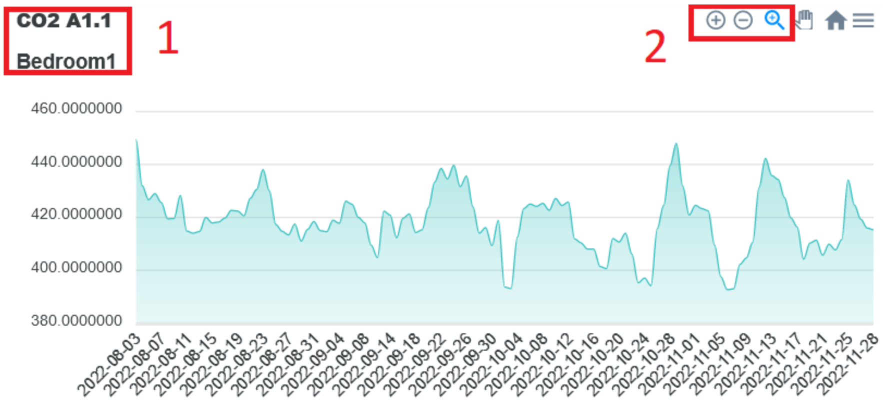
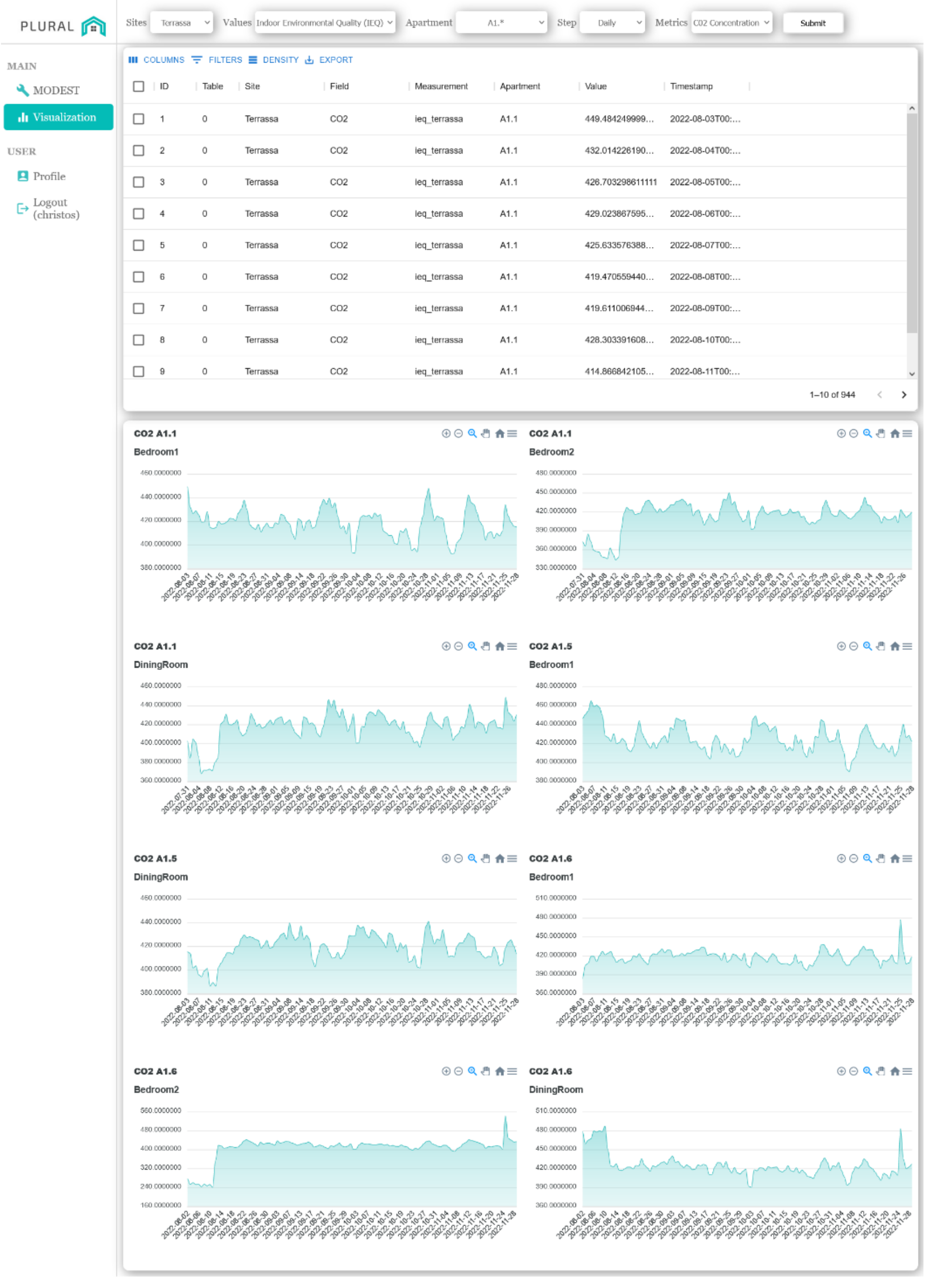


FIGURE 26: LYSIS UI CHARTS CLOSE-UP

Finally, Figure 27 depicts the retrieved monitoring data for the Terrassa demo site, averaged on a "Daily" time step.



Sites: Terrassa | Values: Indoor Environmental Quality (IEQ) | Apartment: A1.* | Step: Daily | Metrics: C02 Concentration

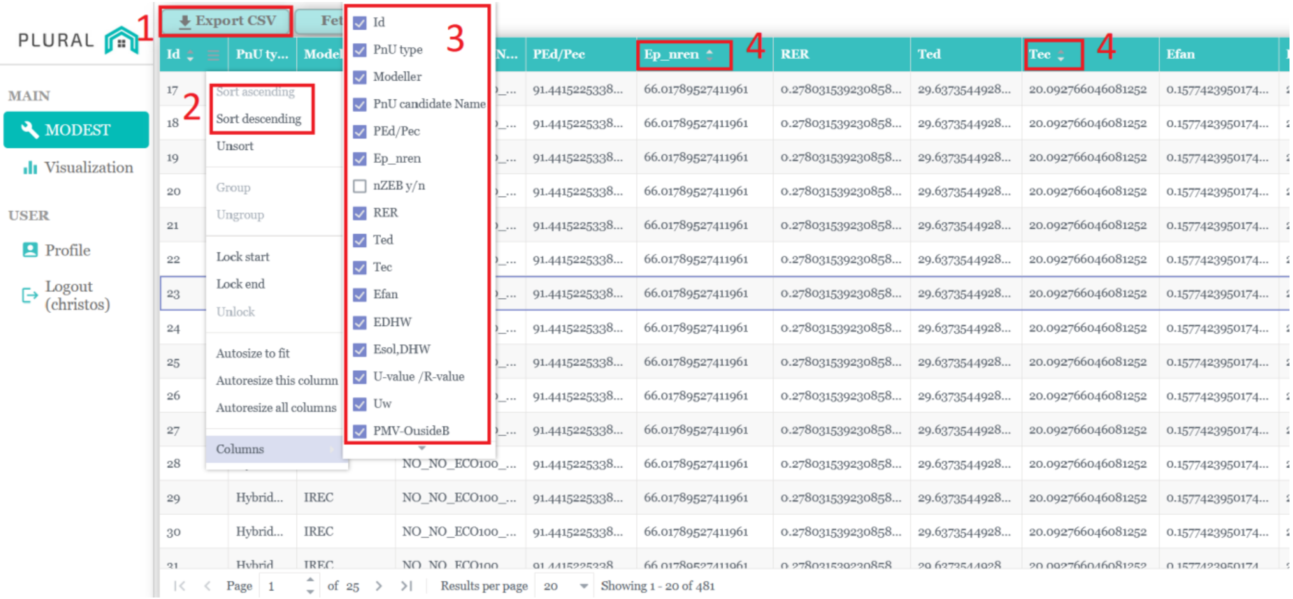
☐	ID	Table	Site	Field	Measurement	Apartment	Value	Timestamp
☐	1	0	Terrassa	CO2	ieq_terrassa	A1.1	449.484249999...	2022-08-03T00:...
☐	2	0	Terrassa	CO2	ieq_terrassa	A1.1	432.014226190...	2022-08-04T00:...
☐	3	0	Terrassa	CO2	ieq_terrassa	A1.1	426.703298611111	2022-08-05T00:...
☐	4	0	Terrassa	CO2	ieq_terrassa	A1.1	429.023867595...	2022-08-06T00:...
☐	5	0	Terrassa	CO2	ieq_terrassa	A1.1	425.633576388...	2022-08-07T00:...
☐	6	0	Terrassa	CO2	ieq_terrassa	A1.1	419.470559440...	2022-08-08T00:...
☐	7	0	Terrassa	CO2	ieq_terrassa	A1.1	419.611006944...	2022-08-09T00:...
☐	8	0	Terrassa	CO2	ieq_terrassa	A1.1	428.303391608...	2022-08-10T00:...
☐	9	0	Terrassa	CO2	ieq_terrassa	A1.1	414.866842105...	2022-08-11T00:...

1–10 of 944

FIGURE 27: LYSIS UI VISUALIZATION ON DAILY TIME STEP (TERRASSA)

5.2.2 MODEST UI

The MODEST UI tool is offered through the same web interface thus providing a unified user experience. The MODEST tool retrieves KPI related data for the different PnU alternatives and feeds the table with this information. Furthermore, it offers the ability to the end users to sort the above-mentioned data by different and multiple columns both descending and ascending order at the same time. Moreover, the users can choose to view only specific columns on which they are interested in. This functionality enables the end user to find their preferred combination better and faster regarding the required KPI values from the plethora of results presented to them. Finally, the results can be downloaded locally for further analysis.



Id	PnU ty...	Model	N...	PEd/Pec	Ep_nren	RER	Ted	Tec	Efan
17			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
18			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
19			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
20			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
21			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
22			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
23			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
24			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
25			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
26			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
27			_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
28			NO_NO_ECO100_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
29	Hybrid...	IREC	NO_NO_ECO100_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
30	Hybrid...	IREC	NO_NO_ECO100_...	91.4415225338...	66.01789527411961	0.278031539230858...	29.6373544928...	20.092766046081252	0.1577423950174...
31	Hybrid	IREC	NO_NO_ECO100	91.4415225338	66.01789527411961	0.278031539230858	29.6373544928	20.092766046081252	0.1577423950174

Page 1 of 25 Results per page 20 Showing 1 - 20 of 481

FIGURE 28: MODEST UI

Figure 28 displays the appearance and functionality of the MODEST UI implementation marking the various controls and visualization areas:

1. Ability to **export the data** populating the table to their local machine for further analysis
2. Ability to **sort by ascending or descending order multiple columns**
3. Ability to **display only specified-by-the-user columns** for faster and optimized querying of the results

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958218

4. The user can click to the column they want to sort, and an arrow will be displayed suggesting the manner of sorting (ascending/descending order). By clicking again, the user can change the order or "remove" the sorting filter for the particular column.
    a. The "order" of which the sorting takes place (e.g., sort first by Id and then by RER) is based on the order that the user sorts the columns i.e., by clicking first on the ID and then the RER columns, the results will be sorted based primarily on ID and secondly on RER.
    b. By moving the mouse over a column, the user is presented with a menu icon which they can click and be presented with further functionalities (e.g., inclusion/exclusion of specific columns).

5.3 Sorting and Selecting PnU Alternative Configurations

This section focuses in the MODEST's algorithmic description. The MODEST tool implements a straight-forward algorithm for sorting and selecting data to show, that runs in **O(n)** time in the number of PnU alternative configurations that are maintained in LYSIS for any particular site. On the backend, a REST Controller loads all data pertaining to PnU configuration values and related computed KPIs that are stored in appropriate Kafka topics in the memory of the backend application; as soon as the front-end requests these data, they are sent to the front-end application (usually a web page rendered on the user's browser) which then renders them for the user to see. The user may then select any number (currently up to 2) of KPI columns as sort keys in any order they wish, and the rows are sorted accordingly using classical **O(nlogn)** sorting in-memory algorithms, without any need for extra data traffic between the backend and the frontend, reducing communication delays to the absolute bare minimum required.

It is worth noting that as more data (results from PnU configuration simulations) are coming to the Kafka topics, the MODEST backend instantly consumes these new data and makes them available to the frontend applications on their next request, without any delays due to synchronization/notification points.

Filtering functionalities (displaying only those rows for which certain KPIs achieve values within certain thresholds) is another O(n) operation (since no indices are used to reduce the operation to essentially constant-time operation) that will be available in the next release of the software, scheduled for the next weeks.

6. Conclusions

Developing the MODEST tool has been an exercise in distributed systems development, since MODEST gathers information from a highly efficient distributed message queue, namely StreamHandler's Kafka platform, and makes available selected information to multiple concurrent clients accessing it via the web. Its design was simplified once the decisions about where the computations about KPI values should be made were finalized. As such, MODEST/LYSIS acts as a generic platform for deep building renovation projects with nZEB targets using the three Plug-and-Use technologies developed in the project. Extensive simulations via tools such TRNSYS and Pytrnsys have shown that "calculating" the main energetic properties of an entire building is possible and can be done within acceptable time-frames for such projects. In this regard, it is worth noting that INTRA has purchased a High-Performance single computer with 2 CPUs each having 128 cores, and a total of more than 1TB RAM; the machine can be used in the near future to run several complex transys simulations concurrently, should the need arise to create and evaluate more alternative PnU configurations, in the context of the current pilots, or for future ones. Having an initial database of results from representative buildings throughout Europe and an accompanying tool to help designer teams/manufacturers/building owners decide what is best for a particular building from any of several available perspectives such as energy, economic, or LCA/LCC aspects is a new and useful capability offered by the PLURAL project.

7. Appendix A: Mapping of VVV Monitoring Data to the LYSIS Platform

7.1 Apache Kafka data organization

Table 9 lists the topics used for the publishing and storage of the various types of monitoring data for the VVV demo site at Athens, Greece, while Listing 1-Listing 3 present the JSON Schemas used for the respective records.

TABLE 9: TOPICS OF VVV MONITORING DATA

Topic	Description
meas.energy.vvv	Energy-related data (JSON Schema: Listing 1)
meas.ieq.vvv	IEQ-related data (JSON Schema: Listing 2)
meas.weather.vvv	Weather-related data (JSON Schema: Listing 3)

LISTING 1: JSON SCHEMA OF VVV ENERGY-RELATED DATA

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "timestamp": {
      "type": "number",
      "description": "Measurement's Unix timestamp in secs (float)"
    },
    "apartment": {
      "type": "string",
      "description": "Apartment identifier or 'building'"
    },
    "phase": {
      "type": "string",
      "enum": ["PRE-PLURAL", "POST-PLURAL"],
      "description": "Monitoring phase based on the renovation state"
    },
    "variables": {
      "type": "object",
      "properties": {
        "THEc": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "value": {
              "type": "number",
              "description": "Thermal Heating Consumption in kWh (float)"
            }
          },
          "required": [
            "id",
            "value"
          ]
        },
        "Eelec": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "value": {
              "type": "number",
              "description": "Final Electric Energy (float)"
            }
          },
          "required": [
            "id",
            "value"
          ]
        },
        "Edhw_e": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "value": {
              "type": "number",
              "description": "Electric Energy for DHW in kWh (float)"
            }
          },
          "required": [
            "id",
            "value"
          ]
        }
      }
    }
  },
  "required": [
    "timestamp",
    "apartment",
    "phase",
    "variables"
  ]
}
```

LISTING 2: JSON SCHEMA OF VVV IEQ-RELATED DATA

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "number",
        "description": "Measurement's Unix timestamp in secs (float)"
      },
      "apartment": {
        "type": "string",
        "description": "Apartment identifier"
      },
      "zone": {
        "type": "string",
        "description": "Thermal zone identifier"
      },
      "variables": {
        "type": "object",
        "properties": {
          "Temp": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Temperature in Celsius degrees (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "RH": {
            "type": "object",
            "properties": {
              "id": {
```

```
            "type": "string"
          },
          "value": {
            "type": "number",
            "description": "Relative humidity % (float)"
          }
        },
        "required": [
          "id",
          "value"
        ]
      },
      "Ev": {
        "type": "object",
        "properties": {
          "id": {
            "type": "string"
          },
          "value": {
            "type": "number",
            "description": "Ambient light in Lux (float)"
          }
        },
        "required": [
          "id",
          "value"
        ]
      },
      "CO2": {
        "type": "object",
        "properties": {
          "id": {
            "type": "string"
          },
          "value": {
            "type": "number",
            "description": "CO2 concentration in ppm (float)"
          }
        },
        "required": [
          "id",
          "value"
        ]
      },
      "VOC": {
        "type": "object",
        "properties": {
          "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "VOC concentration (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          }
        }
      }
    },
    "required": [
      "timestamp",
      "apartment",
      "zone",
      "variables"
    ]
  }
```

Listing 3: JSON Schema of VVV weather-related data

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "number",
        "description": "Measurement's Unix timestamp in secs (float)"
      },
      "variables": {
        "type": "object",
        "properties": {
          "Temp_out": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Outdoor temperature in Celsius degrees
(float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "RH_out": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Relative humidity % (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "SolRad": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Solar radiation in W/m^2 (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "UV_idx": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "UV index(float)"
              }
          },
          "required": [
            "id",
            "value"
          ]
        }
      }
    }
  },
  "required": [
    "timestamp",
    "variables"
  ]
}
```

7.2 InfluxDB data organization

Table 10 lists the measurements used for the storage of the various types of monitoring data for the VVV demo site at Athens, Greece, while Table 11-Table 16 present the fields and tags for the respective measurement points.

TABLE 10: MEASUREMENTS OF VVV MONITORING DATA

Measurement	Description
energy_vvv	Energy-related data (fields: Table 11, tags: Table 12)
ieq_vvv	IEQ-related data (fields: Table 13, tags: Table 14)
weather_vvv	Weather-related data (fields: Table 15, tags: Table 16)

TABLE 11: FIELDS OF VVV ENERGY MEASUREMENTS

Field	Description
Eelec	Final Electric Energy export/deployed on site
THEc	Thermal Heating Electric Consumption
Edhw_e	Energy to meet Domestic Hot Water demand

TABLE 12: TAGS OF VVV ENERGY MEASUREMENTS

Tag	Description	Value(s)
site	Demo site	VVV
apartment	Monitoring apartment	A1, A2

phase	Monitoring phase with respect to renovation status	PRE-PLURAL, POST-PLURAL

TABLE 13: FIELDS OF VVV IEQ MEASUREMENTS

Field	Description
Temp	Temperature (Celsius degrees)
RH	Relative humidity (%)
Ev	Ambient light/Illuminance (lx)
CO2	CO2 concentration (ppm)
VOC	VOC concentration (ppm)
MRT	Mean Radiant Temperature (Celsius degrees) WS Wind
WS	Wind speed (m/s)

TABLE 14: TAGS OF VVV IEQ MEASUREMENTS

Tag	Description	Value(s)
site	Demo site	VVV
apartment: A1, zone	Monitoring zones per apartment	LivingRoom, Kitchen, Bedroom1, Bedroom2, Bedroom3
apartment: A2, zone	Monitoring zones per apartment	LivingRoom, Kitchen, Bedroom1, Bedroom2

TABLE 15: FIELDS OF VVV WEATHER MEASUREMENTS

Field	Description
Temp_out	Temperature (Celsius degrees)
RH_out	Relative humidity (%)
SolRad	Solar Radiation (W/m2)
UV_idx	UV index

TABLE 16: TAGS OF VVV WEATHER MEASUREMENTS

Tag	Description	Value(s)
site	Demo site	VVV

8. Appendix B: Mapping of Kasava Monitoring Data to the LYSIS Platform

8.1 Apache Kafka data organization

Table 17 lists the topics used for the publishing and storage of the various types of monitoring data for the Kasava demo site in Czech Republic, while Listing 4-Listing 6 present the JSON Schemas used for the respective records.

TABLE 17: TOPICS OF KASAVA MONITORING DATA

Topic	Description
meas.energy.kasava	Energy-related data (JSON Schema: Listing 4)
meas.ieq.kasava	IEQ-related data (JSON Schema: Listing 5)
meas.weather.kasava	Weather-related data (JSON Schema: Listing 6)

LISTING 4: JSON SCHEMA OF KASAVA ENERGY-RELATED DATA

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "timestamp": {
      "type": "number",
      "description": "Measurement's Unix timestamp in secs (float)"
    },
    "apartment": {
      "type": "string",
      "const": "Building",
      "description": "Apartment identifier"
    },
    "phase": {
      "type": "string",
      "const": "PRE-PLURAL",
      "description": "Monitoring phase based on the renovation state"
    },
    "variables": {
      "type": "object",
      "properties": {
        "E_comm": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "value": {
              "type": "number",
              "description": "Electric Energy from common zones in kWh
(float)"
            }
          },
          "required": [
            "id",
            "value"
          ]
      }
      }
    }
  },
  "required": [
    "timestamp",
    "apartment",
    "phase",
    "variables"
  ]
}
```

LISTING 5: JSON SCHEMA OF KASAVA IEQ-RELATED DATA

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "number",
        "description": "Measurement's Unix timestamp in secs (float)"
      },
      "apartment": {
        "type": "string",
        "enum": ["GroundFloor", "FirstFloor"],
        "description": "Apartment identifier"
      },
      "zone": {
        "type": "string",
        "enum": ["CommonRoom", "Bedroom1", "Bedroom2", "Bedroom", "Room"],
        "description": "Thermal zone identifier"
      },
      "variables": {
        "type": "object",
        "properties": {
          "Temp": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Temperature in Celsius degrees (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "RH": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Relative humidity % (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "Ev": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Ambient light in Lux (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "CO2": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "CO2 concentration in ppm (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "VOC": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "VOC concentration (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          }
        }
      }
    },
    "required": [
      "timestamp",
      "apartment",
      "zone",
      "variables"
    ]
  }
```

LISTING 6: JSON SCHEMA OF KASAVA WEATHER-RELATED DATA

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "number",
        "description": "Measurement's Unix timestamp in secs (float)"
      },
      "variables": {
        "type": "object",
        "properties": {
          "Temp_out": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Outdoor temperature in Celsius degrees
(float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "RH_out": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Relative humidity % (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "SolRad": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
            "value": {
              "type": "number",
              "description": "Solar radiation in W/m^2 (float)"
            }
          },
          "required": [
            "id",
            "value"
          ]
        },
        "Ev_out": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "value": {
              "type": "number",
              "description": "Brightness in Lux (float)"
            }
          },
          "required": [
            "id",
            "value"
          ]
        }
      }
    }
  },
  "required": [
    "timestamp",
    "variables"
  ]
}
```

8.2 InfluxDB data organization

Table 18 lists the measurements used for the storage of the various types of monitoring data for the Kasava demo site in Czech Republic, while Table 19-Table 24 present the fields and tags for the respective measurement points.

TABLE 18: MEASUREMENTS OF KASAVA MONITORING DATA

Measurement	Description
energy_kasava	Energy-related data (fields: Table 19, tags: Table 20)

ieq_kasava	IEQ-related data (fields: Table 21, tags: Table 22)
weather_kasava	Weather-related data (fields: Table 23, tags: Table 24)

TABLE 19: FIELDS OF KASAVA ENERGY MEASUREMENTS

Field	Description
Eelec	Final Electric Energy export/deployed on site
THGc	Thermal Heating Gas consumption
THGd	Thermal Heating Gas demand
Edhw_e	Energy to meet Domestic Hot Water demand
E_comm	Energy from Common zones (for the whole Building)

TABLE 20: TAGS OF KASAVA ENERGY MEASUREMENTS

Tag	Description	Value(s)
site	Demo site	Kasava
apartment	Monitoring apartment	GroundFloor, FirstFloor, Building
phase	Monitoring phase with respect to renovation status	PRE-PLURAL, POST-PLURAL

TABLE 21: FIELDS OF KASAVA IEQ MEASUREMENTS

Field	Description
Temp	Temperature (Celsius degrees)
RH	Relative humidity (%)
Ev	Ambient light/Illuminance (lx)
CO2	CO2 concentration (ppm)
VOC	VOC concentration (ppm)

TABLE 22: TAGS OF KASAVA IEQ MEASUREMENTS

Tag	Description	Value(s)
site	Demo site	Kasava
apartment: GroundFloor, zone	Monitoring zones per apartment	CommonRoom, Bedroom1, Bedroom2

apartment: FirstFloor, zone	Monitoring zones per apartment	CommonRoom, Room, Bedroom

TABLE 23: FIELDS OF KASAVA WEATHER MEASUREMENTS

Field	Description
Temp_out	Temperature (Celsius degrees)
RH_out	Relative humidity (%)
Ev_out	Illuminance (lx)
SolRad	Solar Radiation (W/m2)

TABLE 24: TAGS OF KASAVA WEATHER MEASUREMENTS

Tag	Description	Value(s)
site	Demo site	Kasava

9. Appendix C: Mapping of Terrassa Monitoring Data to the LYSIS Platform

9.1 Apache Kafka data organization

Table 25 lists the topics used for the publishing and storage of the various types of monitoring data for the Terrassa demo site in Spain, while Listing 7-Listing 10 present the JSON Schemas used for the respective records.

TABLE 25: TOPICS OF TERRASSA MONITORING DATA

Topic	Description
meas.energy.terrassa	Energy-related data (JSON Schema: Listing 7 for apartments, Listing 8 for the whole building)
meas.ieq.terrassa	IEQ-related data (JSON Schema: Listing 9)
meas.weather.terrassa	Weather-related data (JSON Schema: Listing 10)

LISTING 7: JSON SCHEMA OF TERRASSA ENERGY-RELATED DATA

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "number",
        "description": "Measurement's Unix timestamp in secs (float)"
      },
      "apartment": {
        "type": "string",
        "description": "Apartment identifier"
      },
      "phase": {
        "type": "string",
        "enum": ["PRE-PLURAL", "POST-PLURAL"],
        "description": "Monitoring phase based on the renovation state"
      },
      "variables": {
        "type": "object",
        "properties": {
          "THc": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Thermal Heating Consumption in kWh
(float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "Eelec": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Final Electric Energy (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "Edhw_st": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Solar Thermal Energy consumed for DHW in
kWh (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "Efan": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Electric consumption for ventilation in
kWh (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "Edhw_e": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Electric Energy for DHW in kWh (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          }
        }
      }
    },
    "required": [
      "timestamp",
      "apartment",
      "phase",
      "variables"
    ]
  }
```

Listing 8: JSON Schema of Terrassa energy-related data (building)

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "number",
        "description": "Measurement's Unix timestamp in secs (float)"
      },
      "apartment": {
        "type": "string",
        "description": "Always equal to 'building'"
      },
      "phase": {
        "type": "string",
        "enum": ["PRE-PLURAL", "POST-PLURAL"],
        "description": "Monitoring phase based on the renovation state"
      },
      "variables": {
        "type": "object",
        "properties": {
          "THc": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Thermal Heating Consumption in kWh
(float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "Eelec": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Final Electric Energy (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "Edhw_st": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Solar Thermal Energy consumed for DHW in
kWh (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "Efan": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Electric consumption for ventilation in
kWh (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "Edhw_e": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Electric Energy for DHW in kWh (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          }
        }
      }
    },
    "required": [
      "timestamp",
      "apartment",
      "phase",
      "variables"
    ]
  }
```

LISTING 9: JSON SCHEMA OF TERRASSA IEQ-RELATED DATA

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
      "timestamp": {
        "type": "number",
        "description": "Measurement's Unix timestamp in secs (float)"
      },
      "apartment": {
        "type": "string",
        "description": "Apartment identifier"
      },
      "zone": {
        "type": "string",
        "description": "Thermal zone identifier"
      },
      "variables": {
        "type": "object",
        "properties": {
          "Temp": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Temperature in Celsius degrees (float)"
              }
            },
              "required": [
                "id",
                "value"
              ]
            },
            "RH": {
              "type": "object",
              "properties": {
                "id": {
                  "type": "string"
                },
                "value": {
                  "type": "number",
                  "description": "Relative humidity % (float)"
                }
              },
              "required": [
                "id",
                "value"
              ]
            },
            "Ev": {
              "type": "object",
              "properties": {
                "id": {
                  "type": "string"
                },
                "value": {
                  "type": "number",
                  "description": "Ambient light in Lux (float)"
                }
              },
              "required": [
                "id",
                "value"
              ]
            },
            "CO2": {
              "type": "object",
              "properties": {
                "id": {
                  "type": "string"
                },
                "value": {
                  "type": "number",
                  "description": "CO2 concentration in ppm (float)"
                }
              },
          "required": [
            "id",
            "value"
          ]
        },
        "VOC": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "value": {
              "type": "number",
              "description": "VOC concentration (float)"
            }
          },
          "required": [
            "id",
            "value"
          ]
        }
      }
    }
  },
  "required": [
    "timestamp",
    "apartment",
    "zone",
    "variables"
  ]
}
```

LISTING 10: JSON SCHEMA OF TERRASSA WEATHER-RELATED DATA

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "timestamp": {
      "type": "number",
      "description": "Measurement's Unix timestamp in secs (float)"
    },
    "variables": {
      "type": "object",
      "properties": {
        "Temp_out": {
          "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Outdoor temperature in Celsius degrees
(float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "RH_out": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Relative humidity % (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "SolRad": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Solar radiation in W/m^2 (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          },
          "Ev_out": {
            "type": "object",
            "properties": {
              "id": {
                "type": "string"
              },
              "value": {
                "type": "number",
                "description": "Brightness in Lux (float)"
              }
            },
            "required": [
              "id",
              "value"
            ]
          }
        }
      }
    },
    "required": [
      "timestamp",
      "variables"
    ]
  }
```

9.2 InfluxDB data organization

Table 26 lists the measurements used for the storage of the various types of monitoring data for the Terrassa demo site in Spain, while Table 27-Table 32 present the fields and tags for the respective measurement points.

TABLE 26: MEASUREMENTS OF TERRASSA MONITORING DATA

Measurement	Description
energy_terrassa	Energy-related data (fields: Table 27, tags: Table 28)
ieq_terrassa	IEQ-related data (fields: Table 29, tags: Table 30)
weather_terrassa	Weather-related data (fields: Table 31, tags: Table 32)

TABLE 27: FIELDS OF TERRASSA ENERGY MEASUREMENTS

Field	Description
Eelec	Final Electric Energy export/deployed on site
THc	Thermal Heating Consumption

Efan	Electric consumption for mech. ventilation
Edhw_e	Energy to meet Domestic Hot Water demand
Edhw_st	Solar Thermal Energy consumed for Domestic Hot Water

TABLE 28: TAGS OF TERRASSA ENERGY MEASUREMENTS

Tag	Description	Value(s)
site	Demo site	Terrassa
apartment	Monitoring apartment	A1.1, A1.5, A1.6, A2.3, A2.5, A2.6, Building
phase	Monitoring phase with respect to renovation status	PRE-PLURAL, POST-PLURAL

TABLE 29: FIELDS OF TERRASSA IEQ MEASUREMENTS

Field	Description
Temp	Temperature (Celsius degrees)
RH	Relative humidity (%)
Ev	Ambient light/Illuminance (Lx)
CO2	CO2 concentration (ppm)
VOC	VOC concentration (ppm)

TABLE 30: TAGS OF TERRASSA IEQ MEASUREMENTS

Tag	Description	Value(s)
site	Demo site	Terrassa
apartment: A1.1, zone	Monitoring zones per apartment	DiningRoom, Bedroom1, Bedroom2
apartment: A1.5, zone	Monitoring zones per apartment	DiningRoom, Bedroom1
apartment: A1.6, zone	Monitoring zones per apartment	DiningRoom, Bedroom1, Bedroom2
apartment: A2.3, zone	Monitoring zones per apartment	DiningRoom, Bedroom1, Bedroom2, Study
apartment: A2.5, zone	Monitoring zones per apartment	DiningRoom, Bedroom1, Bedroom2
apartment: A2.6, zone	Monitoring zones per apartment	DiningRoom, Bedroom1, Bedroom2, Study

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958218

TABLE 31: FIELDS OF TERRASSA WEATHER MEASUREMENTS

Field	Description
Temp_out	Temperature (Celsius degrees)
RH_out	Relative humidity (%)
Ev_out	Illuminance (lx)
SolRad	Solar Radiation (W/m2)

TABLE 32: TAGS OF TERRASSA WEATHER MEASUREMENTS

Tag	Description	Value(s)
site	Demo site	Terrassa

10. References

[1] Hetzner, "Truly thrifty cloud hosting," [Online]. Available: https://www.hetzner.com/cloud. [Accessed 2022].

[2] Rocky Linux, "Enterprize Linux, the community way.," [Online]. Available: https://rockylinux.org/. [Accessed 2022].

[3] M. Rouse, "What is hard-drive encryption?," TechTarget, [Online]. Available: https://www.techtarget.com/searchenterprisedesktop/definition/hard-drive-encryption. [Accessed 2022].

[4] netfilter.org, "Firewalling, NAT, and packer mangling for linux," [Online]. Available: https://www.netfilter.org/. [Accessed 2022].

[5] SECTIGO, "What is an SSH Key? Generation, Authentication, Key Pair Info & More," 24 September 2020. [Online]. Available: https://sectigo.com/resource-library/what-is-an-ssh-key. [Accessed 2022].

[6] Cloudflare, "What is HTTPS?," [Online]. Available: https://www.cloudflare.com/learning/ssl/what-is-https/. [Accessed 2022].

[7] Keycloak, "Open Source Identity and Access Management," [Online]. Available: https://www.keycloak.org/. [Accessed 2022].

[8] Nginx, "Advanced Load Balancer, Web Server & Reverse Proxy," [Online]. Available: https://www.nginx.com/. [Accessed 2022].

[9] Wallarm, "What is CI/CD concept? How can it work?," [Online]. Available: https://www.wallarm.com/what/what-is-ci-cd-concept-how-can-it-work. [Accessed 2022].

[10] Atlassian, "What is continuous integration?," [Online]. Available: https://www.atlassian.com/continuous-delivery/continuous-integration. [Accessed 2022].

[11] Atlassian, "What is version control?," [Online]. Available: https://www.atlassian.com/git/tutorials/what-is-version-control. [Accessed 2022].

[12] GitHub Inc., "Let's build from here," [Online]. Available: https://github.com. [Accessed 2022].

[13] Jenkins, "An open source automation server," [Online]. Available: https://www.jenkins.io/. [Accessed 2022].

[14] Jenkins, "User Handbook - Pipeline," [Online]. Available: https://www.jenkins.io/doc/book/pipeline/. [Accessed 2022].

[15] Jenkins, "User Handbook - Pipeline Syntax," [Online]. Available: https://www.jenkins.io/doc/book/pipeline/syntax/. [Accessed 2022].

[16] Jenkins, "User Handbook - Using a Jenkinsfile," [Online]. Available: https://www.jenkins.io/doc/book/pipeline/jenkinsfile/. [Accessed 2022].

[17] JFrog, "Container Registy. The world's most advanced, powerful, hybrid Docker and Helm registy," [Online]. Available: https://jfrog.com/container-registry/. [Accessed 2022].

[18] Portainer.io, "Powerful container management for platform teams, DevSecOps, Devs," [Online]. Available: https://www.portainer.io/. [Accessed 2022].

[19] Apache Kafka, "Open-source distributed event streaming platform," [Online]. Available: https://kafka.apache.org/. [Accessed 2022].

[20] InfluxData, "InfluxDB Time Series Data Platform," [Online]. Available: https://www.influxdata.com/. [Accessed 2022].

[21] InfluxData, "Flux Open Source Query Language," [Online]. Available: https://www.influxdata.com/products/flux/. [Accessed 2022].

[22] Baeldung, "Send Large Messages With Kafka," 9 November 2022. [Online]. Available: https://www.baeldung.com/java-kafka-send-large-message. [Accessed 2022].

[23] HAM Systems, "Programmatic access to devices," [Online]. Available: https://hamsystems.tawk.help/en-us/article/api. [Accessed 2022].

[24] HAM Systems, "HAM Systems API for Python," [Online]. Available: https://pypi.org/project/hamapi/. [Accessed 2022].

[25] AWEKAS, "Automatic Weather Map System," [Online]. Available: https://www.awekas.at/wp/?lang=en. [Accessed 2022].

[26] UCCEB - CVUT, "API v1 Documentation," [Online]. Available: https://data.uceeb.cz/doc/api/v1. [Accessed 2022].

[27] AKHQ.io, "Manage & view data inside your Apache Kafka cluster," [Online]. Available: https://akhq.io/. [Accessed 2022].

[28] Apache Kafka, "Apache Kafka Qucikstart," [Online]. Available: https://kafka.apache.org/documentation/#quickstart_download. [Accessed 2022].

[29] Conduktor, "Kafka CLI Tutorials," [Online]. Available: https://www.conduktor.io/kafka/kafka-cli-tutorial. [Accessed 2022].

[30] influxdata, "Visualize data with the InfluxDB UI," [Online]. Available: https://docs.influxdata.com/influxdb/v2.5/visualize-data/. [Accessed 2022].

[31] influxdata, "influx - InfluxDB command line interface," [Online]. Available: https://docs.influxdata.com/influxdb/v2.5/reference/cli/influx/. [Accessed 2022].