Ref. Ares(2025)9851802 - 14/11/2025

D2.3 – COMMUNITAS Core Platform first release and dashboard for assets visualization

COMMUNITAS

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

D2.3 – COMMUNITAS Core Platform first release and dashboard for assets visualization

Dissemination Level: PU - Public
Lead Partner: RINA-C
Due date: 31/07/2024
Actual submission date:

Published in the framework of:
Bound to accelerate the roll-out of Energy Communities and empower consumers

Authors:
Giorgio Tardito, RINA-C
Davide Martini, RINA-C
Pasquale De Lucia, RINA-C
Rolando Quaretti, RINA-C
Elena Conte, E@W
Marco Antonio Insabato, E@W
Giuseppe Mastandrea, E@W
Giuseppe Rocco Rana, E@W
Reshma Penjerla, E@W

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Revision and history chart

Version	Date	Editors	Entity	Comment
0.1	**28/06/2024**	**Giorgio Tardito**	**RINA-C**	**Initial ToC**
0.2	**18/07/2027**	**Giorgio Tardito, Davide Martini, Pasquale De Lucia, Rolando Quaretti**	**RINA-C**	**Integration of the contents related to COMMUNITAS Core Platform**
0.3	**26/07/2024**	**Elena Conte, Marco Antonio Insabato, Giuseppe Mastandrea, Giuseppe Rocco Rana, Reshma Penjerla**	**E@W**	**Integration of contents related to the visual dashboard**
0.4	**29/07/2024**	**Giorgio Tardito**	**RINA-C**	**Finalization of the contents**
0.5	**31/10/2025**	**Giorgio Tardito, Benedetta Barchi**	**RINA-C**	**Inclusion of the Project Officer comments**

Disclaimer:

The information in this document is subject to change without notice. Company or product names mentioned in this document may be trademarks or registered trademarks of their respective companies.

All rights reserved

The document is proprietary of the COMMUNITAS consortium members. No copying or distributing, in any form or by any means, is allowed without the prior written agreement of the owner of the property rights.

This document reflects only the authors' view. The European Community is not liable for any use that may be made of the information contained herein. Responsibility for the information and views expressed in the therein lies entirely with the author(s).

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Table of contents

List of figures:

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

List of tables:

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Glossary:

Acronym	Full name
CCP	COMMUNITAS Core Platform
UI	User Interface
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
API	Application programming interface
GUI	Graphical User Interface
IoT	Internet of Things
HTTPS	HyperText Transfer Protocol over Secure Socket Layer
JWT	JSON Web Token
TLS	Transport Layer Security
AMQP	Advanced Message Queuing Protocol
STOMP	Streaming Text Oriented Messaging Protocol
HTML	HyperText Markup Language
CSS	Cascading Style Sheets

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

1. Executive summary

This document is a complementary report to the first release of the COMMUNITAS Core Platform (CCP) and the Customizable Dashboard, which are the main software outputs of the COMMUNITAS project. Its main purpose is to describe the methodology, rationale and implementation choices that guided the development of the first functional version of the Core Platform, serving as a foundational milestone towards the fully operational system that will be released in subsequent project phases.

The COMMUNITAS Core Platform constitutes the technological backbone of the project, providing a unified digital environment for the integration and orchestration of data streams, services, and tools aimed at enabling citizen engagement in energy markets and fostering the creation and operation of Energy Communities. The platform is conceived as a modular and scalable system capable of interconnecting a wide range of applications developed in Work Package 3 and ensuring interoperability with pilot sites distributed across Europe.

The document begins by outlining the methodology adopted to collect and harmonise requirements from both tool developers and pilot partners. Through structured surveys and bilateral exchanges, the team gathered detailed information regarding data inputs and outputs, communication protocols, and interface specifications. This participatory and iterative approach ensured that the architecture of the platform was tailored to the functional and technical needs of the consortium while maintaining compliance with state-of-the-art standards for IoT data management and cybersecurity.

The Core Platform's architecture is built around a dual-database solution that combines the robustness of a relational MySQL database for structured data with the flexibility of a non-relational InfluxDB database for time-series data collected from sensors. This configuration allows for efficient data storage, retrieval and processing while ensuring scalability and resilience. Data ingestion and distribution are managed through secure and standardised communication protocols such as MQTTS and RabbitMQ, complemented by Telegraf for preprocessing and integration with InfluxDB. The backend, developed using Node.js and deployed via Docker, guarantees cross-platform compatibility, modularity, and ease of maintenance.

Security and user management are handled through Keycloak, an open-source identity and access management system that enables single sign-on across all the integrated tools. This solution not only enhances usability but also ensures compliance with privacy and data protection principles, crucial for handling user-generated and sensor-based data within Energy Communities.

A significant part of the document is devoted to the integration strategy adopted for the visual components of the platform. Given the heterogeneity of the tools developed under WP3, two main approaches were defined: embedding static interfaces through iFrames for tools with simple workflows, and redirecting users via secure links for dynamic tools requiring user interaction. For those tools without existing interfaces, dedicated frontends have been designed following a shared visual identity defined within Task 2.3. This approach guarantees coherence, usability, and recognisability across all tools and ensures a seamless experience within the COMMUNITAS ecosystem.

The Customizable Dashboard, developed using Dash and React frameworks, represents the main access point for end-users, allowing them to visualize data, interact with the different tools, and manage their profiles. The dashboard integrates pages dedicated to individual tools, as well as sections

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

for user authentication, contact forms, and documentation. The flexible architecture of Dash enables advanced data visualization capabilities and the possibility of future integration of cross-tool analytics without substantial modifications to the existing structure.

Overall, this first release of the COMMUNITAS Core Platform provides a solid technological basis upon which further developments will build. The current version already enables the collection of data from sensors using standard protocols and supports user registration and authentication. The next development phase will focus on extending the set of available APIs, integrating the diverse tools developed in WP3, and connecting the platform to all sensors deployed in the pilot sites.

The final release of the platform, including all new developments, complete tool integration, and validation of the Continuous Development Infrastructure, will be reported in Deliverable D2.5 “COMMUNITAS Core Platform final release & Continuous Development Infrastructure set-up, deployment and validation.” This next step will mark the transition from a prototype to a fully operational system supporting the COMMUNITAS vision of citizen-centric, data-driven, and interoperable Energy Communities.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

2. Introduction

2.2. Objectives

This deliverable has been categorised as *Other*, so this document needs to be considered as a supplement to the actual deliverable which is the first release of the COMMUNITAS Core Platform (CCP) and the Customizable Dashboard. The purpose of this document is to provide an overview of the current state of the platform describing its main components, technologies and functionalities.

2.3. Structure of the document

The document is structured in three main chapters, and a final which summarize the conclusions.

- Chapter 3 regards the methodology adopted to gather feedback and needs form the pilots and tools developers.
- Chapter 4 contains a description of the Core Platform. In this chapter is presented the status of the first version of the platform highlighting its main functionalities.
- Chapter 5 describes the work conducted regarding the Customizable Dashboard.
- Chapter 6 present the conclusion and the next steps

2.4. Relation to other tasks

This deliverable can be seen as the first step of the actual development for the COMMUNITAS Core Platform. The work described in this deliverable has been conducted in T2.3, building on the work which have been conducted for the deliverable D2.1 “COMMUNITAS Core Platform architecture and specifications”.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

3. Methodology adopted

The COMMUNITAS project aims to create a platform that can integrate and analyse data from various sensors and provide useful insights for the end-users thanks to the integration of different tools. To achieve this goal, surveys have been shared with the tool developers and the pilot partners to collect their needs and requirements. The data sources and formats that the sensors will produce have been also examined to ensure compatibility and interoperability. The COMMUNITAS Core Platform is the outcome of this collaborative and iterative process, started in T2.1 *Architecture design and specifications' identification*, which strives to deliver a robust, scalable, and interoperable solution for the COMMUNITAS project.

3.1. Information gathered from tool developers

In order to ensure that the tools developed in WP3 *Innovative tools unlocking citizens' active participation in energy markets and communities* will have access to needed data, it has been asked to all the tool providers involved in this Work Package to fill two tables to collect the preliminary information about the data they will require as input (Table 1) and that they will provide as an output (Table 2). This information was crucial to select the most suitable technology for the databases that will store and process the data.

Table 1: Tool provider input data collection form

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**

Table 2: Tool provider output data collection form

OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**

Table 3,Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 10, Table 11 and Table 12 report the answers received.

Table 3: Inputs and outputs of the Demand Response and Optimal market position tool employing NILM

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**
float	Watt / watt-hour	every 15 minutes	Outputs of NILM tool with predictions of power consumption. 1 file per appliance considered (or a JSON dictionary with all appliances)

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

float	Watt / Watt-hour	every 15 minutes	Local generation forecast
float	Watt / Watt-hour	every 15 minutes	metering data from considered houses
float	Watt / Watt-hour	every 15 minutes	Load Forecast (other than NILM)
OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**
float	Watt-hour	15 mins	flexibility forecast for implicit demand response
float/string	N/A	daily	User advisory messages (require frontend with user)

Table 4: Inputs and outputs of the Guarantees of Origin marketplace

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**
integer	kWh	daily/weekly/monthly	Whole production in a production unit during a previous period (day, week, month)
integer	kWh	daily/weekly/monthly	Whole consumption in a house during a previous period (day, week, month)
OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**
Complex data	N/A	Asyncronous	Information about a GoO(chaged/issued): [identifier, issuing date, expiry date, energy type, production date, production, status]

Table 5: Inputs and outputs of the NILM

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**
float	Watts	1 measurement per minute or higer	Total active power consumption of a household.
float	kvar	1 measurement per minute or higer	Total reactive power consumption of a household (optional).
float	Volts	1 measurement per minute or higer	Voltage measured at the central smart meter of a household (optional).
float	Amperes	1 measurement per minute or higer	Current measured at the central smart meter of a household (optional).

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

float	Watts	1 measurement per minute or higer	Active power consumption of the individual appliances monitored through smart plugs. (This is not an input perse but, these values are needed to evaluate and validate the NILM AI model predictions)
OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**
float	Watts	Same frequency as input data	Predictions of active power consumption of the individual appliances to be monitored.

Table 6: Inputs and outputs of the P2P platform & fungible and non-fungible token solution

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**
float	kW or kWh	15 minute, 30 minute, hourly	Total power/energy consumption of a household.
float	kW or kWh	15 minute, 30 minute, hourly	Total power/energy production of a household (Solar PV etc).
float	kW or kWh	15 minute, 30 minute, hourly	Total power/energy net export of a household (Solar PV etc).
float	kW or kWh	15 minute, 30 minute, hourly	Active consumption of the individual appliances monitored through smart plugs. (This is not an input perse but, these values are needed to evaluate and validate the NILM AI model predictions)
float	kW or kWh	15 minute, 30 minute, hourly	Any consumption reduction from automated load shedding of individual appliances.
tuple	$/kWh, kW, time	On demand (although not necessarily real time).	Provision of bidding data, including price willing to pay, energy required, timing of energy required.
tuple	$/kWh, kW, time	On demand (although not necessarily real time).	Provision of offer data, including price willing to pay, energy required, timing of energy required.
OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**
tuple	$/kWh, kW, time	On demand (although not necessarily real time).	Cleared bidding data, including price, energy, timing of energy.
tuple	$/kWh, kW, time	On demand (although not necessarily real time).	Cleared supply data, including price, energy, timing of energy.
float	$/kWh	15 minute, 30 minute, hourly	market clearing price.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Table 7: Inputs and outputs of the Knowledge base and Energy Community management tool, Investment advisor for household- and community-level sustainable investments, Non-energy services: health & wellbeing and sustainable finance

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**
Static	Excel file (String)	Once	Excel file with repositories of base line information (data base with solutions and assumptions for investments plans calculation)
Static	String	Once	Contents to be displayed in the platform (news and other tips for energy efficiency improvement). Can be updated as needed
Dynamic	kWh		Energy consumed per household (per appliance if NILM available)
Dynamic	Geographical	When using	(Still a possibility only) Connection with Maps to estimate roof area
Dynamic	String	Before each simulation	Users surveys' answers
OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**
Dynamic	String	After performing simulation	Investment plan
Dynamic	String	After performing simulation	Updates performance KPIs to be displayed in the dashboard

Table 8: Inputs and outputs of the Community-scale energy management: real-time optimisation of DERs (OptiMEMS)

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**
float	kWh	15 minutes	Forecasted PV production
float	kWh	15 minutes	Forecasted load consumption
float	euro	15 minutes	Forecasted electricity pricing
N/A	N/A	daily	Optimization mode
OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**
float	kWh	15 minutes	Inverter set points

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Table 9: Inputs and outputs of the P2P NFT and Fungible Tokens

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**
integer	monetary unit	Asynchronous	Depending on the user request certain monetary units are minted, burnt or transferred
OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**
boolean	N/A	Asynchronous	Acceptance or rejection of the user request

Table 10: Inputs and outputs of the Real-time multi-energy state estimation for Energy Communities for DER optimization - MultiFASE

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**
Electrical Network			
Network topology	-	once	Nodes and Lines
Nominal voltages	V	once	
Connectivity	-	once	Y or Delta
Line Impedances	Ω	once	
Lines Nominal Power	W	once	
Line Nominal Voltage	P	once	
Transformers	-		Type (Y or Delta / StepUp or Step Down)
Transformer Nominal voltages per side	V	once	
Transformer per unit impedance	pu	once	
Meters location	-	once	
Meter Types	-	once	Voltage or P-Q
Meters standard error	%	once	
Active Power consumption and production per prosumer/asset per phase	W	15min or Hourly	Realtime, Historic data or Forecast
Reactive Power consumption and or production per prosumer/asset per phase	var	15min or Hourly	Realtime, Historic data or Forecast
Inline or Node measurements	V or W	15 min or Hourly	Voltage or P-Q
District Heating Network			
Network topology	-	once	Nodes, Valves and Pipes
Pipe Length	km	once	
Inner Pipe Diameter	m	once	
Pipe Roughness	mm	once	

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Loss coefficient		once	Localized pressure loss coefficient which might account for e.g. bends
Heat transfer coefficient	W/(m^2K)	once	
Inner diameter for Loads (Short Pipe Theory)	m	once	
Loss coefficient for Loads (Short Pipe Theory)		once	Localized pressure loss coefficient
Inner diameter for Valves			
Loss coefficient for Valves			Localized pressure loss coefficient
Design Values	-	once	Pressure, Mass Flow, Temperature
Meters location	-	once	
Meter Types	-	once	Pressure, Mass Flow, Temperature
Meters standard error	%	once	
Working Medium	-	once	In case of other than water
Pumps/Circulators	-	once	Position and technical characteristics
Heat exchangers	-	once	Position and technical characteristics
Asset Heat Production (Temperature, Pressure, Flow)	W, K, m3/hr and bar	15min or Hourly	Realtime, Historic data or Forecast
Heat Demand per consumer (Temperature, Pressure, Flow)	W, K, m3/hr and bar	15min or Hourly	Realtime, Historic data or Forecast
Inline or Node measurements (Heat Temperature, Pressure, Flow)	W, K, m3/hr and bar	15min or Hourly	Realtime, Historic data or Forecast
OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**
Electrical Network			
Voltage per node per phase	V	15min or Hourly	
Power flow per line and phase	W	15min or Hourly	
Power losses	W	15min or Hourly	
Messaging to Optimems per meter in case of error	-	15min or Hourly	
District Heating Network			
Pressure per node	bar	15min or Hourly	
Flow per line	m3/hr	15min or Hourly	
Temperature per node or line	K	15min or Hourly	
Heat Flow per node/line	W	15min or Hourly	
Heat Losses	W	15min or Hourly	

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Table 11: Inputs and outputs of the Uniform Smart community Evaluation tool - USE

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**
KPI: Name	-	Once	The name of the KPI.
KPI: Aggregation levels	-	Once	One or more of: City, building, PEB, PED.
KPI: Aggregation level values	-	Once	The value achieved for each aggregation level. Units depend on the functional unit (FU) of the KPI.
KPI: Dimension	-	Once	One of: Economic, Social, Environmental, Governance, Mobility, Propagation, Social. The dimension that the KPI is applied.
KPI: Dimension weight per aggregation level	ratio	Once	The weight of the KPI for pairs of Aggregation Level – Dimension. Defined by experts.
KPI: Sectors	-	Once	One or more of: Quality of life and prosperity, Climate change and mitigation, Resource efficiency, Smart and reliable infrastructure. The sectors that the KPI applies.
KPI: Sector weight per aggregation level	ratio	Once	The weight of the KPI for pairs of Aggregation Level – Sector. Defined by experts.
KPI: Margins of success	KPI FU or %	Once	The margins that define if the KPI's value has: Failed, Achieved, Overachieved or Excelled.
PSI: Name	-	Once	The name of the PSI.
PSI: Call impact	-	Once	The related call impact to the PSI.
PSI: Value	-	Once	The value achieved by the PSI.
PSI: Margins of success	-	Once	The margins that define if the PSI's value has: Failed, Achieved, Overachieved or Excelled.
OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**
Project Performance Index (PPI)	Likert (1-5)	Once	Harmonic mean of the PSIs scores. The greater the better.
Sustainability Impact Index (SII)	Likert (1-5)	Once	The aggregated value of the KPIs scores in their related dimension. The greater the better.
Sustainable Performance Index (SPI)	Likert (1-5)	Once	The arithmetic mean of the KPIs scores in their related sectors. The greater the better.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Table 12: Inputs and outputs of the Multisector static and real time lifecycle environmental and costing assessment (LCA/LCC) service for Energy Communities - VERIFY

INPUT DATA			
Type	**Unit**	**Frequency**	**Description**
Building			
Total Floor surface	m2	Once	Building's total floor surface in m2.
Total Floor height	m	Once	Building's total floor height in m.
Total External wall surface	m2	Once	Building's total external wall surface in m2.
External wall material	-	Once	E.g. concrete, brick, cinderblock.
Target Temperature – summer	°C	Once	The indoor target temperature to be achieved during cooling period.
Target Temperature – winter	°C	Once	The indoor target temperature to be achieved during heating period.
Number of floors	-	Once	The number of floors of the building.
Heating/Cooling components: Type	-	Once	E.g. natural gas boiler, air-to-air heat pump
Heating/Cooling components: Thermal rating	kW	Once	Thermal rating of the component in kW.
Heating/Cooling components: Usage	%	Once	The percentage of the component's contribution in thermal energy production.
Heating/Cooling components: Property	-	Once	Owned or central.
Heating/Cooling components: Equivalent users	-	Once	The number of users that share a central component.
Heating/Cooling components: Priority	-	Once	The order in which the heating/cooling components are used.
Heating/Cooling components: Purpose	-	Once	Heating, cooling, both.
Heating/Cooling components: Heating/Cooling efficiency	ratio	Once	Efficiency factor of heating components. For heat pumps / ACs, the COP_h/EER_h value should be provided (real, example range [0-5]). For boilers, the thermal efficiency (range [0-1]) should be provided.
Ventilation components: Type	-	Once	E.g. natural, mixed
Ventilation components: Average air changes per hour	ACH	Once	Number of average changes per hour.
Ventilation components: Power rating	kW	Once	The power rating of the ventilation (unless natural).

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Insulation components: Surface	m2		The total surface of the insulation component in m2.
Insulation components: Material	-	Once	E.g. glass wool, stone wool, polyurethane
Insulation components: Thickness	mm	Once	The thickness of the insulation layer in mm.
Insulation components: Orientation	-	Once	E.g. North, south, all walls
Insulation components: Thermal conductivity	W/mK	Once	Thermal conductivity of the insulation.
Glazing components: Opening surface	m2	Once	The total surface of the glazing component.
Glazing components: Number of layers	-	Once	The number of glass layers in the component.
Glazing components: Layer thickness	mm	Once	The thickness of a glass layer of the glazing in mm.
Glazing components: Glass type	-	Once	E.g. normal glass, thermochromic
Glazing components: Frame material	-	Once	E.g. wood, pvc, aluminium
Glazing components: Frame coverage	%	Once	The percentage of frame that covers the glazing.
Glazing components: Orientation	-	Once	E.g. North, south, all walls
DHW (Heat storage): Tank material	-	Once	E.g. steel, iron, plastic
DHW (Heat storage): Tank capacity	ltr	Once	The volume of water that the tank can hold in litres.
DHW (Heat storage): Source	-	Once	E.g. natural gas boiler, electric heater
RES systems: Type	-	Once	E.g. photovoltaic ground polycrystalline, wind turbine geared
RES systems: Installed power	kW	Once	The installed power of the RES system in kW.
RES systems: Efficiency	%	Once	The percentage of renewable resource that is converted into electrical energy.
Grid			
Energy production components: Type	-	Once	E.g. solar park, power plants
Energy production components: Installed power	kW	Once	The power of the energy production component in kW.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Energy production components: Number	-	Once	Number of same type components.
Energy storage components: Type	-	Once	E.g. flywheel, bess
Energy storage components: Capacity	kWh	Once	The amount of the electrical energy the battery can store in kWh.
Energy storage components: Number	-	Once	Number of same type components.
EV charging stations: Type	-	Once	E.g. AC, DC
EV charging stations: Installed power	kW	Once	The installed power of the charging station in kW.
EV charging stations: Annual consumption	kWh	Once	The annual consumption of the charging station in kWh.
EV charging stations: Number	-	Once	Number of same type components.
Timeseries			
Timeseries: Temperature Indoor	°C/ K	Hourly/Daily	The indoor temperature of the building. Hourly is preferred, but daily suffices if not available.
Timeseries: Temperature Outdoor	°C/ K	Hourly/Daily	The outdoor temperature of the building. Hourly is preferred, but daily suffices if not available.
Timeseries: Heating infrastructure consumption	kWh, litre	Hourly/Daily	The fuel or electricity consumption for heating. If possible separate for each heating component. Hourly is preferred, but daily suffices if not available.
Timeseries: Cooling infrastructure consumption	kWh	Hourly/Daily	The electricity consumed for cooling. If possible separate for each cooling component. Hourly is preferred, but daily suffices if not available.
Timeseries: Electrical energy consumed by appliances and lights.	kWh	Hourly/Daily	The electrical energy consumed, excluding thermoelectric consumption, in kWh. Hourly is preferred, but daily suffices if not available.
Timeseries: Electrical energy produced by RES	kWh	Hourly/Daily	The electrical energy produced by RES systems in kWh. If possible separate for each RES system. Hourly is preferred, but daily suffices if not available.
Timeseries: Electrical energy produced by power plants	kWh	Hourly/Daily	The electrical energy produced by power plants in kWh. If possible separate for each plant. Hourly is preferred, but daily suffices if not available.
OUTPUT DATA			
Type	**Unit**	**Frequency**	**Description**

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Electrical Production	kWh	Hourly/Daily/ Monthly/Yearly	Total electrical energy generated by RES. Frequency depends on the frequency of the input data.
Electrical Consumption	kWh	Hourly/Daily/ Monthly/Yearly	Total electrical energy consumed. Frequency depends on the frequency of the input data.
Self Consumption	kWh	Hourly/Daily/ Monthly/Yearly	The amount of energy consumed from the energy generated by RES. Frequency depends on the frequency of the input data.
Use-phase CO_2 Emissions	kgCO2-eq	Hourly/Daily/ Monthly/Yearly	Total CO2 emissions that emerge from the use-phase of the installed components. Frequency depends on the frequency of the input data.
Use-phase Primary Energy Demand	kWh	Hourly/Daily/ Monthly/Yearly	Total primary energy demand that emerge from the use-phase of the installed components. Frequency depends on the frequency of the input data.
Total Infrastructure Embodied CO_2	kgCO2-eq	Once	The CO_2 emissions produced in order to install the new components.
Total Infrastructure Embodied Primary Energy	kWh	Once	The primary energy demand in order to install the new components.
Lifetime CO_2 Emissions	kgCO2-eq	Once	Total project CO2 emissions during the project's lifetime.
Lifetime Primary Energy Demand	kWh	Once	Total project primary energy demand during the project's lifetime.

3.2. Information gathered from pilots

To ensure that it will be possible to collect the data provided by the sensors, it was needed to understand the type of sensors that the pilots are expecting to deploy in their sites. Therefore, it was requested to provide some preliminary information (Table 13) about the type and characteristics of the sensors they used or planned to use. However, the responses received were scarce since some pilots had not started the deployment of sensors at the time. Nevertheless, based on the available information and the feedback from the other pilots, it has been decided to adopt MQTT as a communication protocol, since it is a common standard supported by all the sensors that were deployed or intended to be deployed.

Table 13: Sensor information collection form

What is its identifier? (e.g. TEMP001)
Which type of sensor is it? (e.g. thermometer)
Which kind of measurement does it perform? (e.g. temperature)
Which unit of measurement is used to express the physics quantity? (e.g. °C, ...)
Is it connected to a controller (e.g. NI cRIO, SIMATIC PLC, Arduino, Raspberry ...) or does it come with an on-board processor?

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

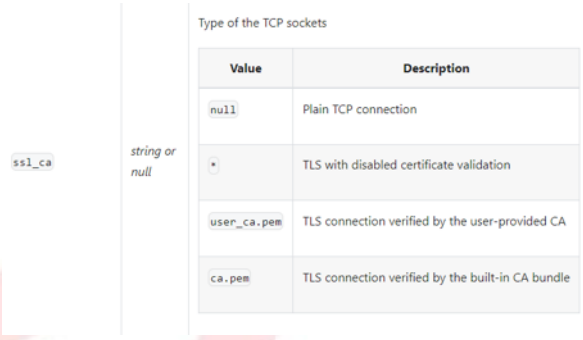
Which protocols of communication does it support? (e.g. MQTT, Line Protocol, …)
If already implemented, can you provide the URL, the host and the gateway of the broker?
If applicable, which MQTT Quality of Service (QoS) policy is used?
Which data format is used? (e.g. JSON, CSV, InfluxDB Line Protocol)
If applicable, which certificates are used?
If applicable, can you provide username and password?
Can you provide an example of message?

Table 14, Table 15, Figure 1 and Figure 2 report the answers received.

Table 14: Sensor information for the NILM

SENSORS	
What is its identifier? (e.g. TEMP001)	shellypro4pm-f008d1d8b8b8 (this is from a shellypro4pm but with shelly plugs should be similar)
Which type of sensor is it? (e.g. thermometer)	electrical sub-meter
Which kind of measurement does it perform? (e.g. temperature)	activePower, energyImported
Which unit of measurement is used to express the physics quantity? (e.g. °C, ...)	Watts, watts*hour
Is it connected to a controller (e.g. NI cRIO, SIMATIC PLC, Arduino, Raspberry ...) or does it come with an on-board processor?	Not needed. In a few years project we implemented some shelly devices that needed a Raspberry as a controller with OpenHub to gather all the data. However, new shelly devices have Wi-Fi connection and through the mobile app the MQTT connection can be configured directly.
Which protocols of communication does it support? (e.g. MQTT, Line Protocol, …)	MQTT
If already implemented, can you provide the URL, the host and the gateway of the broker?	Not yet implemented
If applicable, which MQTT Quality of Service (QoS) policy is used?	With shelly devices QoS 1 is used: guarantees that a message is delivered at least one time to the receiver
Which data format is used? (e.g. JSON, CSV, InfluxDB Line Protocol)	JSON
If applicable, which certificates are used?	This can be configured. See right image (Figure 1). All this information is gathered from https://shelly-api-docs.shelly.cloud/gen2/ComponentsAndServices/Mqtt/
If applicable, can you provide username and password?	Not yet implemented
Can you provide an example of message?	Should be something like below image. (Figure 2)

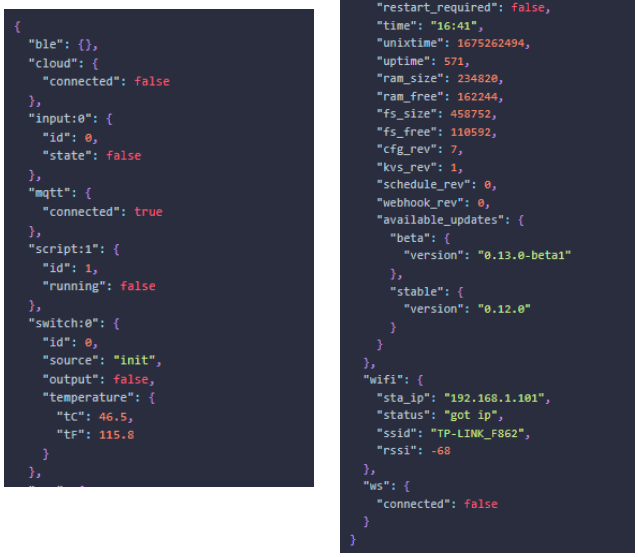
This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**



ssl_ca — string or null — Type of the TCP sockets

Value	Description
null	Plain TCP connection
*	TLS with disabled certificate validation
user_ca.pem	TLS connection verified by the user-provided CA
ca.pem	TLS connection verified by the built-in CA bundle

Figure 1: NILM sensors certificates




```
{
  "ble": {},
  "cloud": {
    "connected": false
  },
  "input:0": {
    "id": 0,
    "state": false
  },
  "mqtt": {
    "connected": true
  },
  "script:1": {
    "id": 1,
    "running": false
  },
  "switch:0": {
    "id": 0,
    "source": "init",
    "output": false,
    "temperature": {
      "tC": 46.5,
      "tF": 115.8
    }
  },
  "sys": {
    "mac": "A8032ABE54DC",
    "restart_required": false,
    "time": "16:41",
    "unixtime": 1675262494,
    "uptime": 571,
    "ram_size": 234820,
    "ram_free": 162244,
    "fs_size": 458752,
    "fs_free": 110592,
    "cfg_rev": 7,
    "kvs_rev": 1,
    "schedule_rev": 0,
    "webhook_rev": 0,
    "available_updates": {
      "beta": {
        "version": "0.13.0-beta1"
      },
      "stable": {
        "version": "0.12.0"
      }
    }
  },
  "wifi": {
    "sta_ip": "192.168.1.101",
    "status": "got ip",
    "ssid": "TP-LINK_F862",
    "rssi": -68
  },
  "ws": {
    "connected": false
  }
}
```


Figure 2: NILM Sensor message example

Table 15: Sensor information related to UC5

SENSORS	
What is its identifier? (e.g. TEMP001)	N/A
Which type of sensor is it? (e.g. thermometer)	Smart Meter
Which kind of measurement does it perform? (e.g. temperature)	Electric measurements (energy, power, voltage, current, frequency etc.)
Which unit of measurement is used to express the physics quantity? (e.g. °C, ...)	Energy -> Wh, Power -> W, Voltage -> V, Current -> A, Frequency -> Hz
Is it connected to a controller (e.g. NI cRIO, SIMATIC PLC, Arduino, Raspberry ...) or does it come with an on-board processor?	It will be connected with a controller (still discussing upon this matter)
Which protocols of communication does it support? (e.g. MQTT, Line Protocol, …)	N/A
If already implemented, can you provide the URL, the host and the gateway of the broker?	not implemented
If applicable, which MQTT Quality of Service (QoS) policy is used?	
Which data format is used? (e.g. JSON, CSV, InfluxDB Line Protocol)	JSON
If applicable, which certificates are used?	
If applicable, can you provide username and password?	
Can you provide an example of message?	sensors are not yet installed in the pilot sites, so we will answer this at a later stage

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

3.3. Gathering information from tools developers for GUI integration

The initial plan for the platform was to have a unified interface that would integrate all the tools developed within the project. Additionally, a single authentication and user control mechanism was planned to ensure authorized access to the entire platform, protecting the privacy and consistency of the information stored and visualized.

However, since most of the tools were based on existing frameworks that needed to be adapted and tailored to fit COMMUNITAS' requirements and use cases, creating a second interface for tools that already had one seemed unnecessary. Therefore, in this phase, the priority is to integrate the existing interfaces of the tools, and to address any possible improvements to the graphical interfaces to achieve a smooth integration into the COMMUNITAS platform in the following development sprints.

After defining the activities related to the integration of the interfaces of the tools already equipped with them, the first action was to start defining common scenarios or pathways that could interconnect the different single tools, in terms of input-output connections.

Once collected the available information corresponding to the tools (Table 3, Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 10, Table 11 and Table 12), the analysis results in a partial understanding of the complexity of the platform in its entirety. It was noted that each single tool had a specific input/output schema and was not really interconnected with the others. Therefore, in order to integrate the different isolated tools into the Core Platform, an extra effort was put to define a set of Scenarios that could group a subset of tools to serve a specific goal. Therefore, an additional level of detail was needed to clearly define the missing points and to identify common paths for the definition of the Scenarios. This second collection of information has been carried out via a second template (Figure 3), which was already pre-filled with preliminary information gathered during the design phase implemented in Task T2.1.

UC	Title	Input	Output	Is API available? If yes, provide documentation or instructions, like Swagger, for utilizing the APIs	If no API, when will it be available? Outline the expected message structures	Is a frontend available? If yes, provide access details and instructions, screenshots, or mockups.	Frontend not available? Provide mockup if expected to developed from your side and in case of not, please provide for information on what you need from the front-end and whether you can describe the expected operation of the front-end, possibly through mock-ups (e.g. using Figma)	If the frontend is unavailable, Provide a mockup for the expected frontend development or detail your frontend requirements and functionalities for the development in the context of Communitas platform	Do applications need to use the Communitas central relational database? Specify data types, tables, and user auth management to align with RINA, including preferred protocols.	Any other information you think should be considered for the Communitas frontend development?
3	Energy Community Management Tool	Smart meters data from households and production assets	File with allocation of energy sharing coefficients per member to community manager and members							
4	Investment Advisor Tool (IAT)	Local RES generation (Generation metering system) Energy consumption (smart meters) User preferences - comfort (user surveys - community input) User preferences - ambitions	Investment advices to web-app elaboration							
5	OptiMEMS	Forecasted PV production (from RES metering equipment) Forecasted load consumption (from house smart meters) Forecasted electricity pricing	Inverter set points							
6	MultiFASE	voltage magnitude and angle at each electrical node, temperature/specific enthalpy at each thermal node mass flow rate at each thermofluid node stream direction at each thermofluid node, pressure at each thermo-fluid node	voltage magnitude and angle at each electrical node, temperature/specific enthalpy at each thermal node mass flow rate at each thermofluid node stream direction at each thermofluid node, pressure at each thermo-fluid node							
7	NILM Demand Response and Optimal Market Position	Total active power consumption of a household (house smart meters or Communitas db). Total reactive power consumption of a household (optional). Voltage measured at the central smart meter of a household (optional). Current measured at the central smart meter of a household (optional). Active power consumption of the individual appliances monitored through smart	Predictions of active power consumption of the individual appliances to be monitored Flexibility forecast sent to any client							

Figure 3: Second template about information on APIs and frontend

The template lists all the Use Cases defined in the project, and asks for information about the interface, a possible API to access the tool, and if the tool needs to retrieve and/or store data to the central database.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

After collecting the preliminary information in the first phase, it was found that it would be necessary to wait for the end of the first sprint of tool development in order to obtain more precise information needed to define the scenarios that would ensure the interconnection and data exchange of the components through the Communitas Core Platform. It was therefore decided to postpone the activity of integrating the tools by implementing scenarios that would allow the integrated use of the tools in the customisable common graphical interface of the Communitas Core Platform to subsequent development phases.

Therefore, as previously mentioned, the activities in this phase were primarily focused on the integration of the tools and their already available graphical interfaces, assessing the possibility of developing new ones for the tools not provided.

By analysing the results collected from the consultation of the developers, an extra effort was necessary to clarify the undefined features of the tools and to understand the differences and peculiarities of the single platforms. Therefore, in order to accommodate for the different tools' needs and to optimize effort, a hybrid twofold integration strategy was defined, depending on the features of the single tool as follows:

- Tools that already provide an interface:
  - iFrames if the interface is static, the tool does not ask the user for additional input and the output produced is immediate and simple and can be visualized in a single page,
  - Redirect buttons if the interface is dynamic, the tool needs additional input added by the user and the output is more complex,
- Tools that do not provide an interface (i.e. are developed from scratch): an ad-hoc interface will be developed, following the requirements defined by the reference technical developers about the type of input required and the output to be visualized.

So, after defined the abovementioned approach for the first integration activities, it was necessary to organize bilateral meetings with the technical developers of each single tool, in order to clearly present the methodology and commonly agree on the strategy to be followed to integrate each tool, depending on its proper features and characteristics. The results of the meetings are summarized in the Table 16

Table 16: List of COMMUNITAS' tools and related Integration strategy

COMMUNITAS' tool	Integration strategy	Responsible Partner	Comments
Knowledge Base	iFrames	EDP	The tools already provide an interface. The flow is rather simple, the user can interact with the tool in a single window integrated in the dashboard.
ECPT		RINA	
MultiFASE/OptiMEMS		CERTH	
USE	Redirect/buttons	CERTH	

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

VERIFY		CERTH	The tools already provide an interface. The flow is complex and some inputs are required from the user. Therefore, a redirect is more appropriate.
GoO		ETRA	
ECMT/gamification		EDP	
Investment Advisor tool		EDP	
Non-energy services		EDP	
Demand Response and NILM	Dedicated frontend	UNINOVA	The ad-hoc frontend will collect user input, store it in the central database and retrieve outputs from the central database for visualization.
P2P NFT and Fungible tokens	TBD	CERTH	It already has a frontend but, since it will be integrated with the Demand Response tool, some more work is needed.
P2P energy market	TBD	UNL	TBD

Given the fact that most of the tools already provide an interface, the definition of a common style to be applied to the different interfaces was needed to ensure visual consistency across the different platforms. Proper style sheet has been defined and is under constant update through alignment activities of the interfaces, conducted and to be conducted with the developer partners in the context of T2.3. This style sheet states the colour schemes, themes, fonts and sizes to be applied by the different developer teams and it follows the visual identity of the project.

After the collection of this information from all the technical partners, it was possible to create a mock-up of the visual interface of the platform, to include a proposal for the visualization of the different types of integration methodologies for the tools. The mock-up represents at high level how the information will be visualised on the Communitas Core Platform. It has been used for the actual implementation of the Communitas Core Platform Dashboard, but some pages might vary due to technical or implementation requirements that may emerge over the course of the project.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

4. COMMUNITAS Core platform

The COMMUNITAS Core Platform is the tool which will allow to empower the engagement of citizens in the energy market, and it will foster the development Energy Communities. This will be achieved thanks to the development of different tools that will be made available through the CCP.

The first phase of the development of the platform has resulted in the provision of the core functionalities that are essential for enabling user login, collecting data from sensors, and storing of the data. The current state of these functionalities does not have to be considered as final and they could be revised and improved in the subsequent sprints.

In the following months, the platform will be integrated with the tools that are being developed in WP3 and the number of sensors that are connected to the platform will increase as well.

4.2. Architecture

The development of the COMMUNITAS Core platform started form the structural view described in D2.1. (Figure 4). In this representation the platform and tools are presented in all of their main functional components giving special attention to their dependencies and needs of external components that are required in order to perform the expected functionalities.

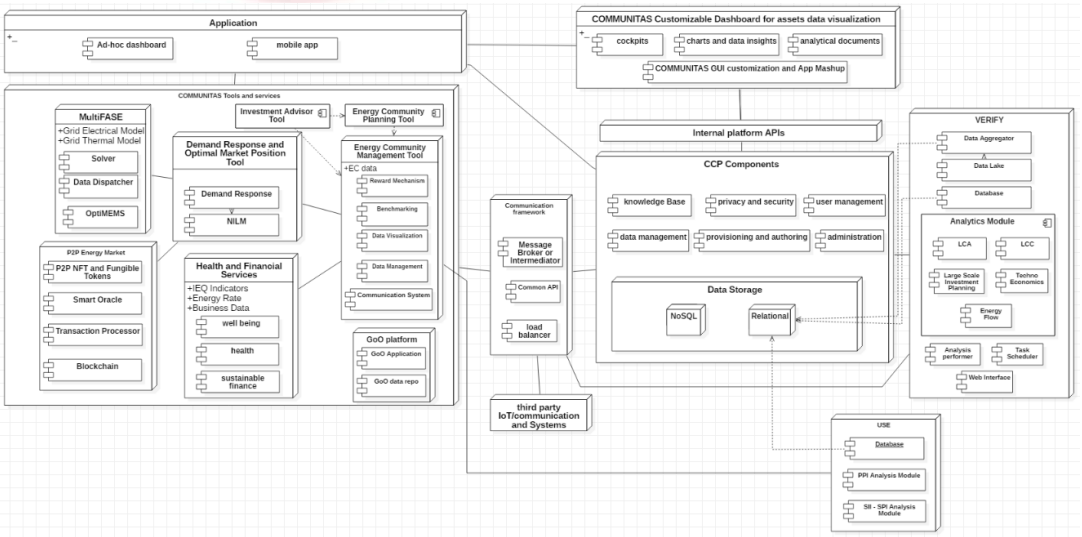


Figure 4: COMMUNITAS Platform Structural view [1]

The main objective of this phase of the project was to develop and implement the core functionalities of the CCP that would enable the effective and efficient management of data from the pilot sites. The platform has a common backend that serves as the main controller of the platform ensuring seamless coordination among its diverse components, a data collection process that can receive data from various types of sensors, and two databases that keeps and organizes the data in a organized and secure way.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

The diagram in Figure 5 shows the architectural design of the CCP, which show also the technologies adopted and the communication protocols that are used to ensure a smooth and secure data transfer among the different components of the platform.

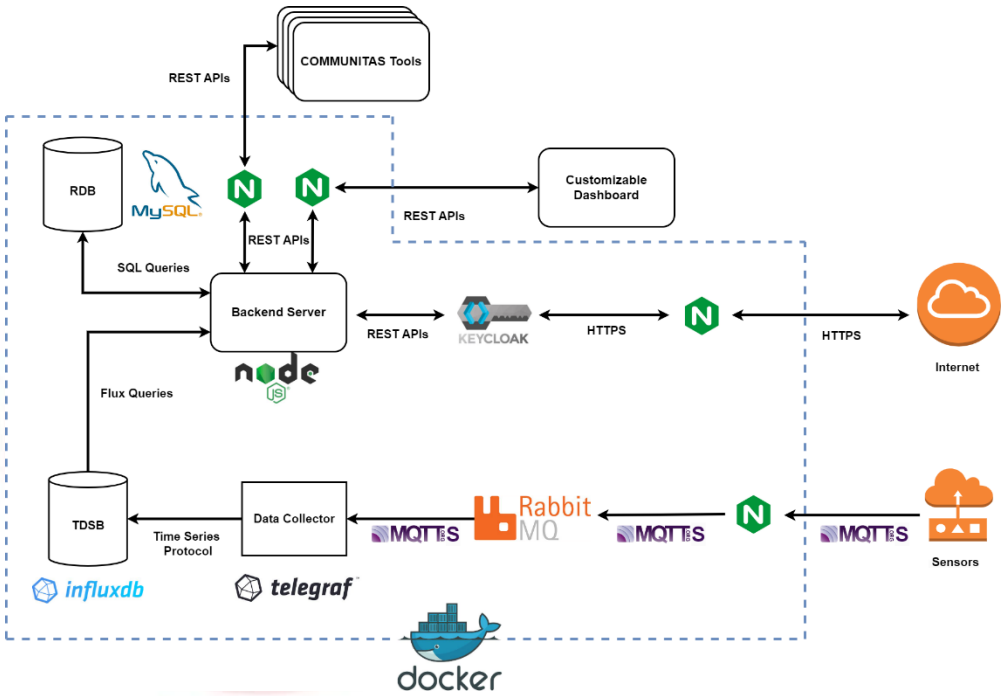


Figure 5: COMMUNITAS Core Platform Architecture

4.3. Databases and Backend

To develop such platform different technologies are going to be adopted. In order to manage the diverse type of data that are going to be needed for the operation of both the tools, it has been chosen to develop both a relational database, using MySQL, and a non-relational database, using InfluxDB.

A relational database is a type of database that stores data in structured tables, where each table consists of rows and columns. Each row represents a record or an entity, and each column represents an attribute or a property of that entity. In order to add, modify and retrieve the data contained in such tables a relational database uses a specific query language called SQL (Structured Query Language). The main benefit of adopting this solution is the ease of creating links among data stored in different tables, uncovering the relation among the data. This solution allows for efficient data organization and avoids duplication, as well as facilitating backup operations. For all of these reasons relation databases are usually adopted when there is need to store structured information, like in this case. [2]

The chosen solution for the relational database is MySQL, which is an open-source software that is widely used in both academic and industrial settings (Figure 6). MySQL has gained a reputation as one of the most popular and reliable database systems, as it offers high performance, scalability, and security. Moreover, it has an extensive documentation that facilitates its use and maintenance. Given

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

its maturity, it is regularly updated and patched to prevent security breaches and bugs. Hence, MySQL is an appropriate option for the relational database that this project requires.

421 systems in ranking, July 2024

Rank Jul 2024	Rank Jun 2024	Rank Jul 2023	DBMS	Database Model	Score Jul 2024	Score Jun 2024	Score Jul 2023
1.	1.	1.	Oracle	Relational, Multi-model	1240.37	-3.72	-15.64
2.	2.	2.	MySQL	Relational, Multi-model	1039.46	-21.89	-110.89
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	807.65	-13.91	-113.95
4.	4.	4.	PostgreSQL	Relational, Multi-model	638.91	+2.66	+21.08
5.	5.	5.	MongoDB	Document, Multi-model	429.83	+8.75	-5.67
6.	6.	6.	Redis	Key-value, Multi-model	156.77	+0.82	-7.00
7.	↑ 8.	↑ 11.	Snowflake	Relational	136.53	+6.17	+18.84
8.	↓ 7.	8.	Elasticsearch	Search engine, Multi-model	130.82	-2.01	-8.77
9.	9.	↓ 7.	IBM Db2	Relational, Multi-model	124.40	-1.50	-15.41
10.	10.	10.	SQLite	Relational	109.95	-1.46	-20.25

Figure 6: Database engine rankings [3]

A non-relational database is a type of database that does not store data in predefined tables, but rather in flexible and dynamic structures, such as documents, graphs, or key-value pairs. This allows for handling large amounts of unstructured and complex data. Non-relational databases are also faster than relational databases, since they do not have to perform complex operations to join data from different tables. Moreover, they can scale, meaning that they can distribute data across multiple servers and handle increasing workloads. Non-relational databases have been widely used for applications that deal with diverse and heterogeneous data sources, such as social media, web analytics, or IoT. [4], [5]

The selected technology for the non-relational database in this project is InfluxDB. (Figure 7) This is an open-source solution that is specifically designed for managing time series data. InfluxDB can scale horizontally to handle large volumes of data across multiple servers. It has also been optimized for storing and querying time-stamped data, making it an ideal choice for the project's needs. [6] Therefore, a non-relational database can offer more flexibility and performance than a relational database in this specific scenario.

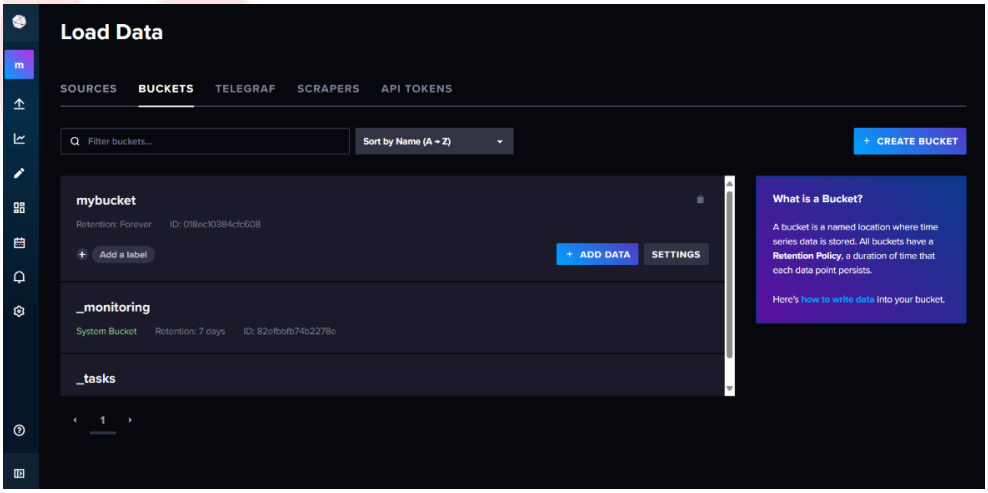


Figure 7: InfluxDB UI

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

The reason for adopting this dual database solution is that the non-relational database will store the data generated by the sensors, which are expected to produce a lot of data at high frequency and granularity, while the relational database will be used to store data generated by the tools connected to the platform.

The backend of the COMMUNITAS Core Platform was developed using Node.js, a technology based on JavaScript that is widely used for web applications. Node.js is fast, reliable, open-source, scalable, and cross-platform. It uses an event-driven programming model, which means that it can execute multiple functions asynchronously, increasing its efficiency and speed. It also has a low CPU usage compared to other technologies despite its capability of executing functions concurrently. Node.js has a large and active community of developers, which provides many resources and support online. It also has a high compatibility with other programming languages, making it easy to integrate with different frameworks and libraries. [7]

The backend created for the CCP has the aim to integrate all the necessary elements components for its proper operations. At the current state the backed provide the APIs for database operations and user authentication. Going forward, it will also offer APIs to facilitate connection with the platform's frontend and for the connection with all the tools that are developed in this project.

In order to increase the protection of the platform it has been chose to adopt NGINX to manage the connection to the other components of the platform, such as the standalone tools and the front-end, and also to manage the connection to the external world, internet and sensors. NGINX is an open-source HTTP and reverse proxy engine; it has been chosen to adopt this solution in order to ensure that all the communication with external components will be secured using a HTTPS protocol. [8]

For the deployment of the CCP, the chosen technology is Docker, a popular and widely used tool for creating and managing containers. A container is an isolated and lightweight environment that can run an application with all its dependencies, without affecting the host system or other containers. It has several benefits over other solutions, such as virtual machines, the time needed to create a containerized solution it is very fast, and it is also very quick to launch it. It provides a consistent and predictable environment for the application to run, regardless of the underlying infrastructure. This technology is very efficient since more containers can share the same kernel. In addition, containers can be easily moved across different machines, clouds, and servers. By using Docker for the CCP (Figure 8), the development, testing, and deployment processes is simplified and standardized, facilitating the deployment and performance of the platform. [9]

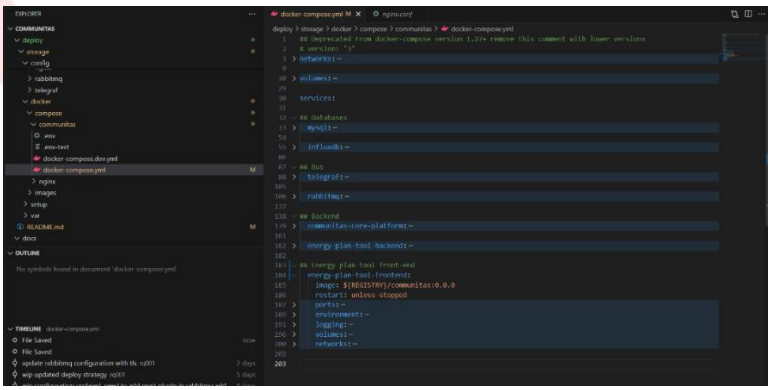


Figure 8: Docker implementation

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

4.4. Authentication

In order to ensure the security of the COMMUNITAS Core Platform an authentication service is needed. The CCP will need to be able to manage various users connect to it and several technologies could be used to tackle this challenge. On the of the most known and widely adopted identity and access management technology is Keycloack.

Keycloack is an open-source solution with a vast community of users, documentation and support. It adopts common standards and provide support for OpenID Connect, OAuth 2.0 and SAML 2.0. It provides support for social identity, meaning that gives the possibility to login using username and password used for other platform, namely: Google, Facebook, Twitter, Github, LinkedIn, Microsoft and StackOverflow. [10]

Keyclaock provides a centralized access management and authentication solution with an easy-to-use UI. (Figure 8) It has single sign-on functionality meaning that the users are able to perform a single login and have access to all the applications present on the platform, if they are allowed to. This is also ensured thanks to the integration with OpenID which used JSON Web Token (JWT) to guaranteeing access to only the applications the user is authorized to access. [11]

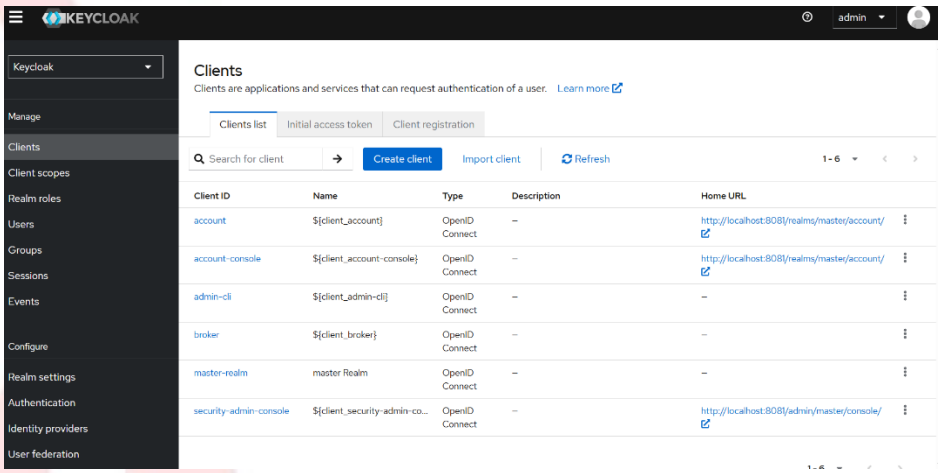


Figure 9: Keycloack clients management

4.5. Data gathering

To collect the data generated by the sensors that technologies that have been implemented are: MQTTS, Rabbit MQ, and Telegraf.

Message Queue Telemetry Transport, or MQTT, is a widely adopted communication standard for sensors, which have gained its reputation thanks to the minimal bandwidth it requires. This protocol it is composed by two elements, a client, which will act as publisher or subscriber, and a server/broker which will act as a middleman. The client in this application will register itself on a broker for a specific topic and then, it will receive the messages received by the broker on the topic on which it is subscribed. This method Is highly efficient and especially suitable from application with low bandwidth, like in scenarios where data is generated by low-powered sensors. [12]

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

For this application it has been chosen to adopt the version of MQTT with enhanced security features, MQTTS. This specific rendition secures the communication among the sensors and the repost of the platform using Transport Layer Security (TLS). This solution will implement a cypher system on top of the TCP/IP stack securing the data transfer at the transport layer. In this way the entirety of the MQTT conversation is encrypted. [13]

Having a constant stream of data written directly on the database will decrease the overall performance of the platform. So, in order to overcome this challenge a technology has been developed called Telegraf. This open-source software, it has been specifically developed to work with databases created using InfluxDB. In addition, it allows also the collection of metrics directly from some IoT sensors or MQTT brokers and it can pre-process and aggregate the data received. [14]

Since the platform will have to collect data generated by sensors deployed in pilots through Europe, there is the needed to implement a technology capable to distribute such amount of raw data. Since one of the pilots had already tested their sensors with RabbitMQ it was chosen to adopt this technology also in a centralized way inside the CCP. (Figure 10) RabbitMQ is a message broker and queuing server that can be used in diverse scenarios and can be adapted to work with many protocols like Advanced Message Queuing Protocol (AMQP), Streaming Text Oriented Messaging Protocol (STOMP), and MQTT. This broker comes with its own UI simplifying the management of permissions, exchange and queues. It is a cross-platform systems meaning it could work on Linux, Windows and Mac OS X and it supports various programming languages. [15], [16]

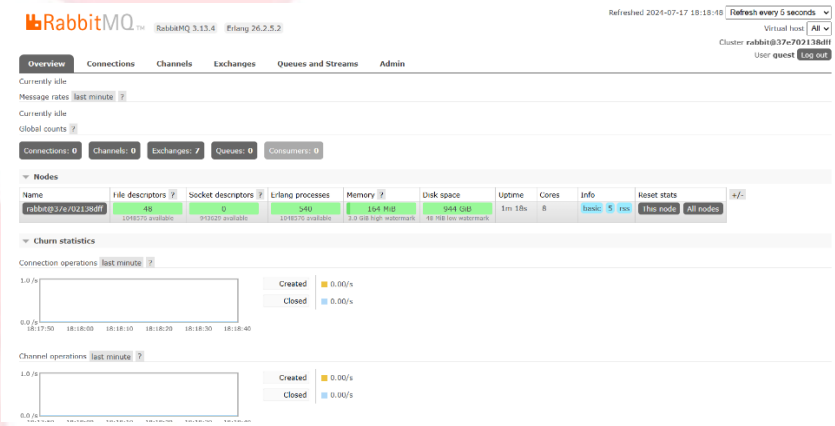


Figure 10: RabbitMQ UI

The COMMUNITAS Core Platform uses a configuration where the data from the sensors that each pilot has installed is collected via RabbitMQ. Telegraf acts as a client that receives the data from RabbitMQ and forwards it to InfluxDB for storage.

4.6. REST APIs Documentation

The platform requires some specific APIs to interact with the databases that store the data collected by the sensors. These APIs allow the platform to perform different operations on the data, such as writing, reading, deleting, or updating. (Figure 11)

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

The current APIS available on the platform are the following:

- For the InfluxDB database:
  - influx_read: this API will allow to retrieve data stored in the database.
  - influx_write: this API will allow to add or remove data present in the database.
- For the MySQL database:
  - mysql_delete: this API will allow to delete rows in a specific database table according to the Id selected.
  - mysql_insert: this API will allow to write data in a table already present in the database.
  - mysql_put: this API will perform an INSERT or an UPDATE query depending on if it will find an already existing corresponding row in the selected table or not.
  - mysql_update: this API will perform an update of all the column and all the rows indicated the request, if they exist.
- For Keycloack:
  - login: this API handle the login process of the users using the authorization token. The backend service redirects the login request to Keycloak that checks the token and the user rights.
  - logout: this API handle the logout process of the users.

```
// Mysql API
app.use('/delete', require('./routes/mysql_delete'));
app.use('/get', require('./routes/mysql_get'));
app.use('/insert', require('./routes/mysql_insert'));
app.use('/put', require('./routes/mysql_put'));
app.use('/update', require('./routes/mysql_update'));

// Auth API
app.use('/login', require('./auth/login'));
app.use('/logout', require('./auth/logout'));

if(config.isMainServer())
{
    // Use routes
    // Influx API
    app.use('/influx_read', require('./routes/influx_read'));
    app.use('/influx_write', require('./routes/influx_write'));
}
```

Figure 11: REST APIs

5. Customizable Dashboard

5.1. Implementation

The Dashboard has been developed as a web application, using Python and JavaScript as programming languages. In particular, the main framework used for the implementation of the pages and graphs of the dashboard is Dash [17]. Dash is a Python library built upon Plotly [18] and React [19], and allows the development of Machine Learning, Data science and visualization applications with easily customizable interfaces. The framework has been selected for its ease of use, reusability, scalability and high customization options, also thanks to the possibility to integrate other functionalities via other JavaScript libraries or frameworks. The frontend capabilities of the framework rely on the possibility to render components on a particular DOM element via the React DOM library. In particular,

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Dash is able to abstract typical graphical elements (i.e. divs, html tags, graphs, input fields, sliders etc.) and manage their interconnections in python.

It is also possible to integrate other frontend elements using HTML and CSS as additional programming languages. For example, the `dash-html-components` library has been used to style the views and the tabs.

On the other hand, Dash is optimal for the development of data visualization graphs, via its dedicated component called `dash-core-components`. One of its components in particular, `Graph`, embeds a wide variety of visualization graphs that can be adapted to the specific application scenario. Some examples are line charts, bar charts, pie charts and histograms. All the graphs developed using these components are interactive thanks to the use of another JavaScript graphing library called plotly.js [20], with the purpose to offer additional functionalities and display more detailed graphical analysis on the incoming data.

Dash provides enough flexibility to be able to develop both simple forms and complex graphs and visualization features. Thanks to this flexibility, it will be possible to enrich the graphical user interface of the CCP by adding more advanced and cross-tools graphs, if and provided that proper data becomes available in the central database, without having to dramatically modify the architecture already defined and adopted for the development of the dashboard, or to migrate the whole system to a different development framework, which may cause delays and additional complexity.

5.2. Graphical interface

Figure 12 presents an example of the proposed mock-up for the CCP dashboard:

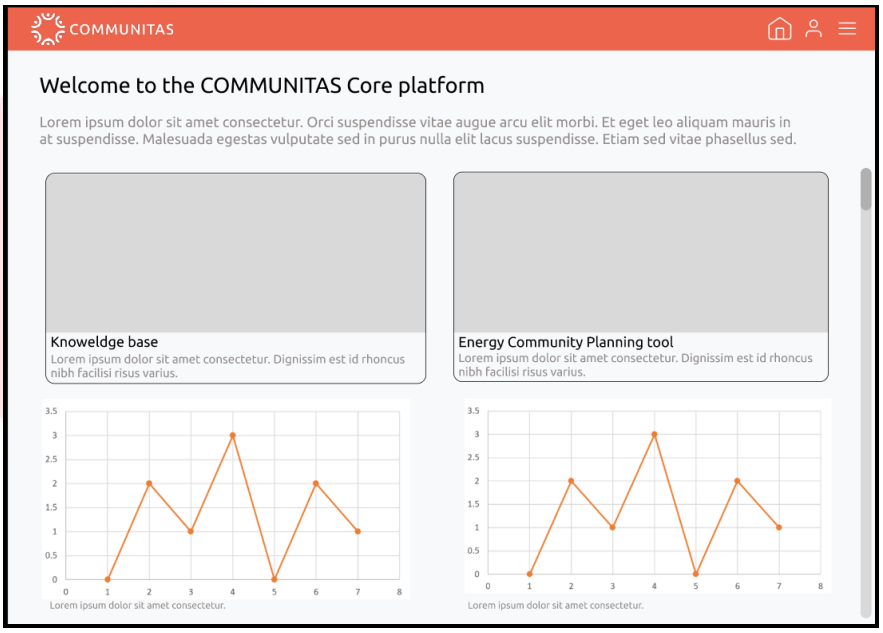


Figure 12: Main dashboard of the Communitas Core Platform, with iFrames

In particular, Figure 12 presents the case applied to integrate the tools via iFrames, as for example the Knowledge Base and the Energy Community Planning tool.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

Following the integration strategy, the remaining tools are integrated via a dedicated page in which there is a brief description of the tool and its functionalities, with a link to redirect the user to the tool itself, as showed in the mock-up in Figure 13:

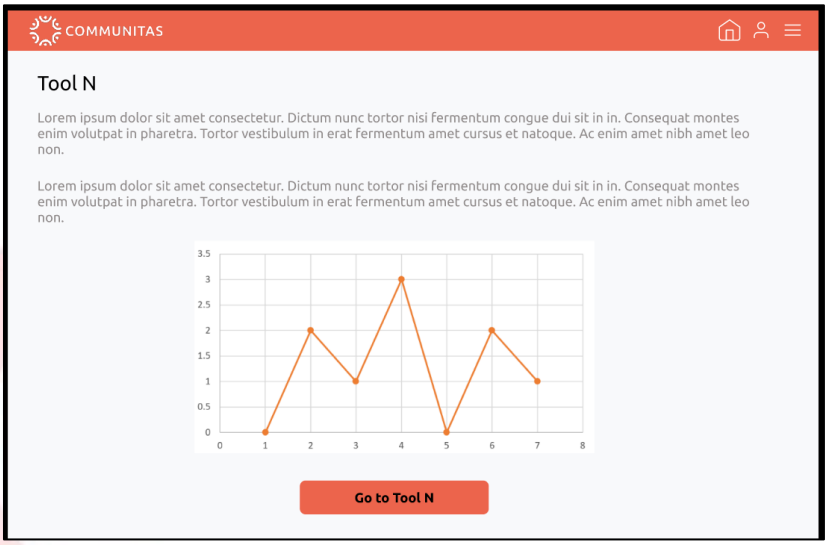


Figure 13: Page dedicated to tools integrated via Redirect

Finally, for the case of Demand response and NILM tool, for which a dedicated frontend has to be developed, a mock-up of the interface has been designed and proposed (Figure 14), including the fields required by the tool, in agreement with the responsible software developers.

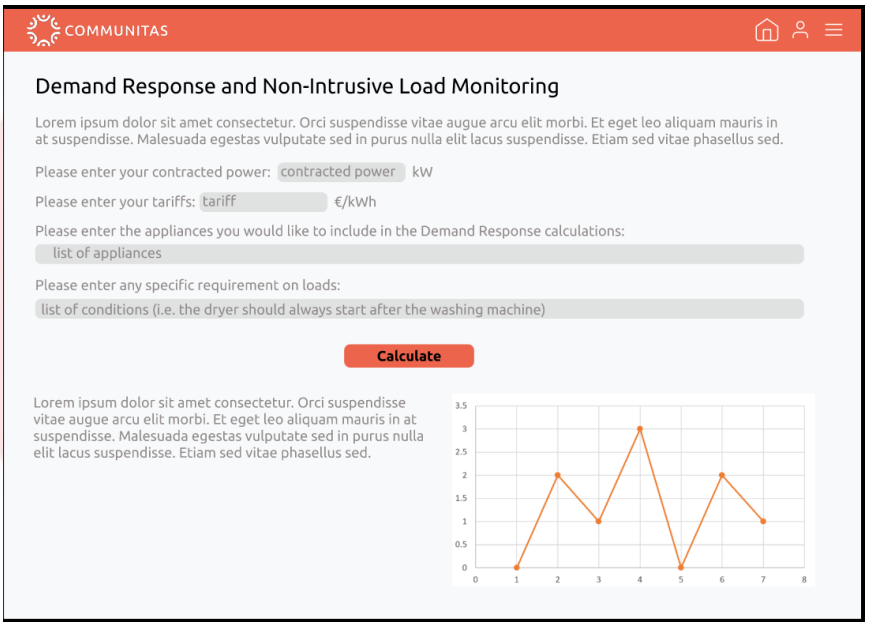


Figure 14: Demand Response and Non-Intrusive Load Monitoring tool page

On top of the pages related to the single tools and the iFrames to embed the static views, a first login page is essential to ensure a unique authentication to the whole platform (Figure 15), using the CCP authentication component.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

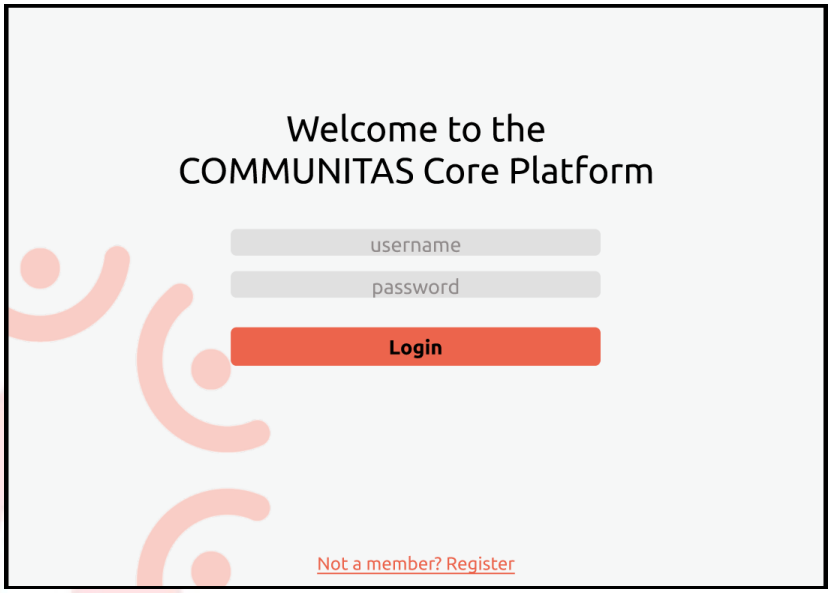


Figure 15: Login page of the main Dashboard

Indeed, the centralized login ensures that the user has to login only once to the platform, and then its access is guaranteed to all the single tools integrated on the platform via a token-based sharing mechanism. The effective authentication of existing users and registration of new ones are managed by the backend of the Core platform, as described in Section 4.4.

Additional Contacts (Figure 16) and Documentation (Figure 17) pages will be available to guide the users in the navigation of the platform and to ask for support in case of need.

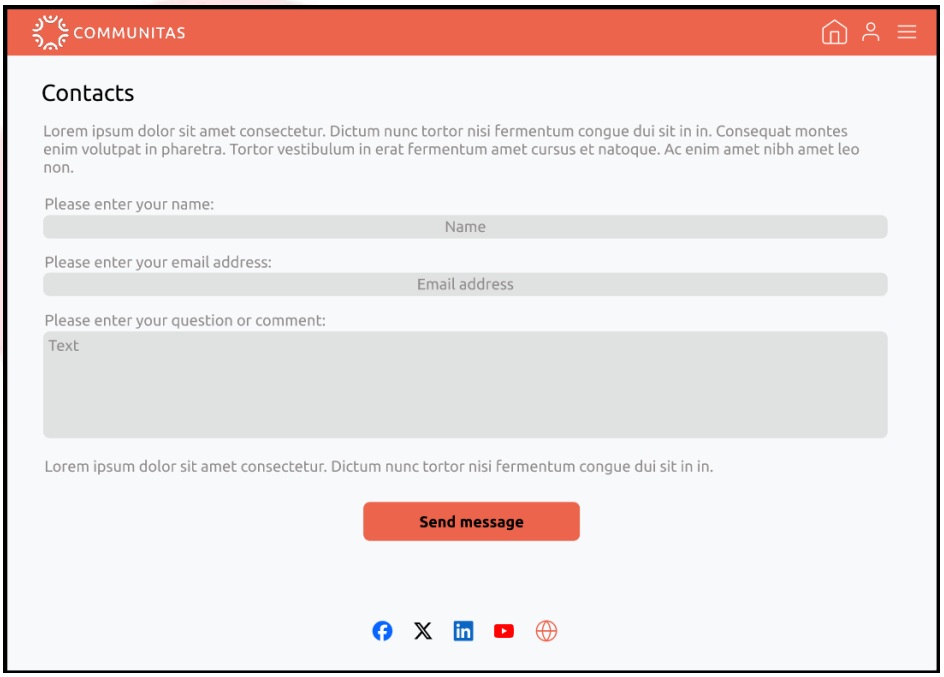


Figure 16: Contacts form detail

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

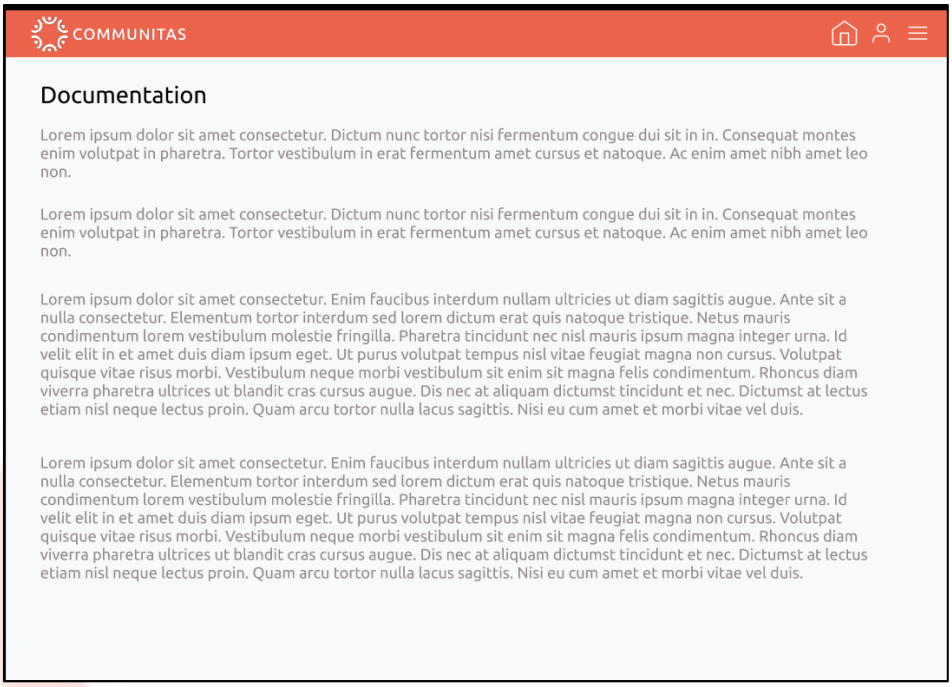


Figure 17: Documentation page detail

6. Conclusion

This deliverable titled “COMMUNITAS Core Platform first release and dashboard for assets visualization” focuses on providing a description of the basic functionalities of the platform. In this first phase the overall UI has been identified, together with the strategies for the integration of the interface of the already existing tools. The current state of the platform allows the collection of data from sensors using standard protocols and it also allows the registration of users on the platform.

In the following months specific APIs will be created in order to integrate the diverse tools that are being developed in WP3, moreover the UI of the tools will be updated in order to ensure a common identity among the diverse tools. The platform will be also connected to all the sensors that are going to be deployed in the pilots considered in the projects.

The final version of the platform, including the new developments and the complete integration of all components, will be reported in Deliverable D2.5 “COMMUNITAS Core Platform final release & Continuous Development Infrastructure set-up, deployment and validation.”

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

7. References

[1] D2.1 - COMMUNITAS Core Platform architecture and specifications

[2] What is a relational database? [Online] Link: https://www.ibm.com/topics/relational-databases [Accessed: 8 July 2024]

[3] DB-Engines Ranking [Online] Link: https://db-engines.com/en/ranking [Accessed: 9 July 2024]

[4] Thakur, Nimesh & Gupta, Nishi. (2021). Relational and Non Relational Databases: A Review. Journal of University of Shanghai for Science and Technology. 23. 117-121. 10.51201/JUSST/21/08341.

[5] Moniruzzaman, A B M & Hossain, Syed. (2013). NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison. Int J Database Theor Appl. 6.

[6] Influxdata [Online] Link: https://www.influxdata.com/ [Accessed: 9 July 2024]

[7] Basumatary, B., & Agnihotri, N. (2022). Benefits and Challenges of Using NodeJS. International Journal of Innovative Research in Computer Science & Technology. https://doi.org/10.55524/ijircst.2022.10.3.13

[8] NGINX [Online] Link: https://nginx.org/ [Accessed: 17 July 2024]

[9] Bashari Rad, Babak & Bhatti, Harrison & Ahmadi, Mohammad. (2017). An Introduction to Docker and Analysis of its Performance. IJCSNS International Journal of Computer Science and Network Security. 173. 8.

[10] Keycloak [Online] Link: https://www.keycloak.org/ [Accessed: 12 July 2024]

[11] D.N. Divyabharathi and Nagaraj G. Cholli, A Review on Identity and Access Management Server (KeyCloak), Page No.: 17-22 Volume: 14, Issue 2, 2020 ISSN: 1990-7958 International Journal of Electrical and Power Engineering

[12] Enache B., Banica C., Bogdan A.. Performance Analysis of MQTT Over Websocket for IoT Applications. The Scientific Bulletin of Electrical Engineering Faculty. 2023;23(1): 46-49. https://doi.org/10.2478/sbeef-2023-0008

[13] M. Ali Al-Muqarm, Abbas & Alkhafajee, Ahmed & Alwan, Ali & Alardawy, Zaid. (2022). Security and Performance Analysis of MQTT Protocol with TLS in IoT Networks. 10.1109/IICETA51758.2021.9717495.

[14] Telegraf [Online] Link: https://www.influxdata.com/time-series-platform/telegraf/ [Accessed: 12 July 2024]

[15] Madhu M P, Sunanda Dixit, Distributing Messages Using Rabbitmq with Advanced Message Exchanges International Journal of Research Studies in Computer Science and Engineering 2019, 6(2): 24-28.

[16] Alvaro Videla and Jason J.W. Williams, RabbitMQ in Action Distributed messaging for everyone, Manning 2012 ISBN 9781935182979

[17] Dash [Online] Link: https://dash.plotly.com/ [Accessed: 24 July 2024]

[18] Plotly [Online] Link: https://plotly.com/ [Accessed: 24 July 2024]

[19] React [Online] Link: https://react.dev/ [Accessed: 24 July 2024]

[20] Plotly.js [Online] Link: https://plotly.com/javascript/ [Accessed: 24 July 2024]

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No. 10196508**