HYDROUSA

H2020-CIRC-2-2017

Water in the context of circular economy

Full project title:

Demonstration of water loops with innovative regenerative business models for the Mediterranean region

Deliverable: D33

Relative Number in WP D5.7

Integrated monitoring and controlling platform

Due date of deliverable: 30/06/2020

Actual submission date: 06/03/2022

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 776643

This project has received funding from
the European Union's Horizon 2020
Research and Innovation Programme
under Grant Agreement No 776643

DOCUMENT INFORMATION

Deliverable	**Number**	D5.7	**Title**	Integrated monitoring and controlling platform
Work Package	**Number**	WP5	**Title**	Monitoring of the demonstration sites and ICT integration

Due date of deliverable	**Contractual**	M24	**Actual**	M45	
Version number	1.2				
Format	MS Office Word document				
Creation date	31/12/2021				
Version date	06/03/2022				
Type	☒ R	☐ DEM	☐ DEC	☐ OTHER	☐ ETHICS
Dissemination Level	☒ PU Public		☐ CO Confidential		
Rights	Copyright "HYDROUSA Consortium". During the drafting process, access is generally limited to the HYDROUSA Partners.				

Responsible authors	**Name:**	Zisis Tsiropoulos Evangelos Skoubris	**E-mail:**	tsiropoulos@agenso.gr eskoubris@agenso.gr
	Partner:	AGENSO	**Phone:**	+30 2109234473

Brief Description	This document describes the online monitoring and controlling platform developed within HYDROUSA, including information about the architecture and the functionalities of the developed platform.
Keywords	Online monitoring, controlling platform, water management, agricultural applications

Version log			
Rev. No.	**Issue Date**	**Modified by**	**Comments**
1.0	31/12/2021	Zisis Tsiropoulos Evangelos Skoubris	First draft of the deliverable
1.1	10/01/2022	Simos Malamis, Eleni Nyktari	Comments from the Coordination Team
1.2	06/03/2022	Zisis Tsiropoulos, Evangelos Skoubris	Final version of the deliverable

TABLE OF CONTENTS

This project has received funding from
the European Union's Horizon 2020
Research and Innovation Programme
under Grant Agreement No 776643

LIST OF FIGURES

EXECUTIVE SUMMARY

In order to visualize and monitor the readings from all the sensors installed for the purposes of the HYDROUSA project, the Grafana platform was chosen and installed on a dedicated server. It is an open-source platform, capable of receiving all the data from the data repository (Deliverable 5.6 "Data repository"), visualizing and presenting them to the interested stakeholders in a reasonable and intelligible way. This is realized with the use of dashboards, by combining various data through SQL queries and using fully customizable widgets (called "panels" in Grafana).

The first chapter of this report provides a brief introduction to the content and the purpose of the platform.

The second chapter provides a brief overview of the Grafana platform, its functions and the specifications of the server on which it is installed.

The third chapter explains in detail how to create a dashboard in the Grafana platform using screenshots, while the fourth and fifth chapters show the steps to create a dashboard for the demo site HYDRO 3 (as an example) and the final result.

Finally, the sixth chapter presents a solution developed to enable user control and automation.

HYDROUSA has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776643.

ABBREVIATIONS

ALv2	**Apache License Version 2**
CLA	**Contributor License Agreement**
CPU	**Central Processing Unit**
DR	**Data Repository**
IMEI	**International Mobile Equipment Identity**
MB	**Megabyte**
RAM	**Random Allocation Memory**
RPM	**Red Hat Package Manager**
SIEM	**Security information and event management**
SQL	**Structured Query Language**
UI	**User Interface**

1. INTRODUCTION

For the visualisation and monitoring of the measured values of the sensors installed at the demonstration sites of the HYDROUSA project, the Grafana platform was chosen, which is available as open source and can be installed on a dedicated server, properly configured and used by the interested stakeholders.

In the following chapters, after a brief presentation of the Grafana platform, its features and technical specifications required, the step-by-step process of creating a dashboard is given and it is applied for the creation of the HYDRO 3 pilot.

The dashboard of the HYDRO 3 pilot is presented in detail, including all the following data:

- General information about the pilot (geolocation, water quantity of the two tanks)
- Data from the weather station installed at the pilot, including temperature, humidity, wind speed, wind gust, light intensity, battery of the station, pressure and rain
- Data from the first and second tank, including water quantity and temperature for each tank
- Soil moisture

This project has received funding from
the European Union's Horizon 2020
Research and Innovation Programme
under Grant Agreement No 776643

2. ABOUT GRAFANA

Grafana is a cross-platform, open-source analysis and interactive visualisation web application. It provides charts, graphs and alerts for the web when connected to supported data sources. A licenced Grafana Enterprise version with additional features is also available as a self-hosted installation or as an account with the Grafana Labs Cloud Service. It is extensible via a plug-in system. End users can create complex monitoring dashboards with interactive query builders. Grafana is divided into a frontend and a backend, written in TypeScript and Go respectively.

As a visualisation tool, Grafana is a popular component in monitoring stacks, often used in combination with time series databases such as InfluxDB, Prometheus and Graphite, monitoring platforms such as Sensu, Icinga, Checkmk, Zabbix, Netdata and PRTG, SIEMs such as Elasticsearch and Splunk, and other data sources. Grafana's user interface was originally based on version 3 of Kibana.

Figure 2.1 An example of a Grafana dashboard

Grafana was first released in 2014 by Torkel Ödegaard as an offshoot of a project at Orbitz. It targeted time series databases such as InfluxDB, OpenTSDB, and Prometheus, but evolved to support relational databases such as MySQL, PostgreSQL and Microsoft SQL Server.

Grafana is widely used including in Wikipedia's infrastructure. Grafana has over 1,000 paying customers, including Bloomberg, JP Morgan Chase, eBay, PayPal, and Sony.

As of April 20, 2021, Grafana is licensed under an AGPLv3 license. Contributors to Grafana need to sign a CLA that gives Grafana Labs the right to relicense Grafana in the future. The CLA is based on The Apache Software Foundation Individual Contributor License Agreement.

Previously Grafana was licensed with an ALv2 license and used a CLA based on the Harmony Contributor Agreement.

2.1. Features of Grafana[1]

Below the most important features of Grafana are presented:

- Dashboard templates: These allow users to create a dashboard that suits their needs. These templates do not contain hard-coded values, so if someone has a test server and a production server, the same dashboard will work on both. Templates allow you to explore data at any level, from macro to micro. For example, you can start with an entire country, then break it down to a specific region and go as far as granularity allows. These dashboards can then be shared with everyone, from teams across the organisation to the entire community.
- Provisioning: It may be easy enough to set up a single dashboard with a few clicks, drags and drops, but some users need even more simplicity in a way that scales. That's why Grafana offers a provisioning feature that lets you automate the setup with a script. In Grafana, everything can be scripted. For example, if you want to create a new Kubernetes cluster, Grafana can automatically help you with a script that already has the right server, IP address and data sources set up and locked down. This is also a way to control many dashboards.
- Annotations: This Grafana feature allows the user to highlight graphs, which is particularly useful when you need to correlate data when something is not behaving correctly. The user can click and type on a graph to manually create your annotations, or you can pull data from any source to populate them. (An example of this is the way Wikipedia uses annotations on its public Grafana dashboard). A good application case would be to automatically create annotations at the time of releases. Then, if errors appear sometime after a new release, you can go back to your annotations and check if the errors correlate. This kind of automation is possible with Grafana HTTP API. Many of Grafana's biggest customers use it for a variety of tasks, including setting up databases and adding users. This is an alternative to provisioning for automation, and there is more that can be done with it. For example, the team at DigitalOcean have used the API to include a snapshot feature that allows them to review dashboards.
- Kiosk mode and playlists: playlists are great for "rolling coverage". The user can select the Grafana dashboards that they would like to display on a monitor or TV, and it can cycle through them throughout the day. Kiosk mode lets the user only show the UI elements that they need in view-only mode. Useful tip: The Grafana Kiosk utility handles logins, switching to kiosk mode, and opening a playlist, so if a TV the user wants to use has no keyboard they can still set it up without hassle.

[1] https://www.plesk.com/blog/various/grafana-monitoring-solution

- Custom plugins: the user can extend Grafana's functionality with plugins that offer extra tools, visualizations, and more. Popular examples include Worldmap Panel (which superimposes data on a map), Zabbix (which integrates with Zabbix metrics), and Influx Admin Panel (which enables database creation or lets one add users). These are just a few examples and there are many more. The user can write a simple code and Grafana can visualise anything that generates a timestamp. Grafana Enterprise customers can also access additional plugins that facilitate integration with Datadog, New Relic, Splunk and others.

- Alerting and alert hooks: Grafana alerts can be sent through several different notifiers, including email, PagerDuty, or Slack or texts. If these aren't enough, it's very easy to code alert hooks that create different notifiers.

- Teams and permissions: where an organization has one instance of Grafana and several teams, they usually like to have the option to enforce some dashboard segregation. It used to be the case that this wasn't possible because Grafana automatically made everyone's dashboards accessible to everyone else. The later edition of multi-tenant mode meant that users could switch organizations but couldn't share dashboards. Some judicious hacks could enable both, so Grafana created an easier route to achieving this. It's now possible to create a team of users and then assign permissions on folders, Grafana dashboards, and so on, right down to the data source level for Grafana Enterprise users.

- SQL data sources: Grafana natively supports SQL, which helps the user to graph any kind of data that might be held in an SQL database. High-end users are doing lots of interesting things with SQL data sources, including building business dashboards that can appeal to the higher hierarchy personnel of a company.

- Monitoring the user's monitoring: if users take monitoring seriously enough to want to monitor their own monitoring, Grafana features its own Prometheus HTTP endpoint that can be scraped by Prometheus, making it simple to get statistics and dashboards. Once the enterprise version is up and running the user will be able to get Google Analytics-style data access, so they can find out just how much CPU their Grafana is using through or how much time alerting takes.

- Authentication: Grafana supports LDAP and OA and other authentication styles and allows to map users to organizations. With Grafana Enterprise, it's also possible to map users to teams: so if one's organization uses their own authentication system, Grafana allows them to map teams in their in-house systems to teams in Grafana, which automatically gives team members access to their own designated Grafana dashboards.

2.2. Versions of Grafana

There are three different versions of Grafana, tailored to different categories users' needs.

The first one is the open-source self-managed version. Users can download, install, administer and maintain their own versions without any limitations.

The second version is the cloud version. It is offered as a fully managed service and is the fastest way to adopt Grafana. It includes a scalable, managed backend for metrics, logs, and traces and is managed and administered by Grafana Labs with free and paid options for individuals, teams and large enterprises. It includes a robust free tier with access to 10k metrics, 50GB logs, 50GB traces, 2 weeks data retention and 3 users.

Finally, there's the enterprise version. It is enhanced with access to Enterprise data source plugins and built-in collaboration features. There's also a self-managed option for organizations that have special requirements around data localization and privacy.

2.3. Grafana for the purposes of the HYDROUSA project

The Grafana version that was chosen for the purposes of the HYDROUSA project was the open source self-managed one.

The requirements for the installation of Grafana are as follows:

- Supported operating systems
  - Debian / Ubuntu
  - RPM-based Linux (CentOS, Fedora, OpenSuse, RedHat)
  - macOS
  - Windows
- Minimum recommended RAM[2]: 255MB
- Minimum recommended CPU[2]: one (1)
- Supported databases
  - SQLite
  - MySQL
  - PostgresSQL
- Supported web browsers[3][4]
  - Chrome/Chromium
  - Firefox

2 Some features, such as server rendering of images, alerting and data source proxy might require more RAM or CPUs
3 Older versions of the aforementioned browsers might not be supported, therefore an update is recommended
4 JavaScript should always be enabled in the user's browser, as running Grafana without JavaScript enabled is not supported

- Safari
- Microsoft Edge
- Internet Explorer 11[5]

The chosen version of Grafana was downloaded from the designated download location and installed on a dedicated server.

The server's specifications are:

- CPU: Intel(R) Core (TM) i7-4930K CPU @ 3.40GHz
- RAM: 14.4GB DDR3 ECC
- HDD: 1x 1TB SATA II
- Operating system: Ubuntu 18.04 (64-bit)
- Monthly bandwidth: 32TB traffic, unlimited incoming

It is obvious that the server satisfies the requirements of the Grafana software.

5 Fully supported only in Grafana versions prior to v6.0

3. CREATING A DASHBOARD AT GRAFANA

This chapter presents the steps for creating a dashboard in Grafana.

Once the chosen version of Grafana is properly installed and configured, users can log in by navigating to the server's URL pointing to port 3000 (the default port where Grafana is installed).

Once the page is loaded, users are prompted to enter their credentials to log in (Figure 3.1).

The first time the user logs in, they will be asked to change their password for security reasons.

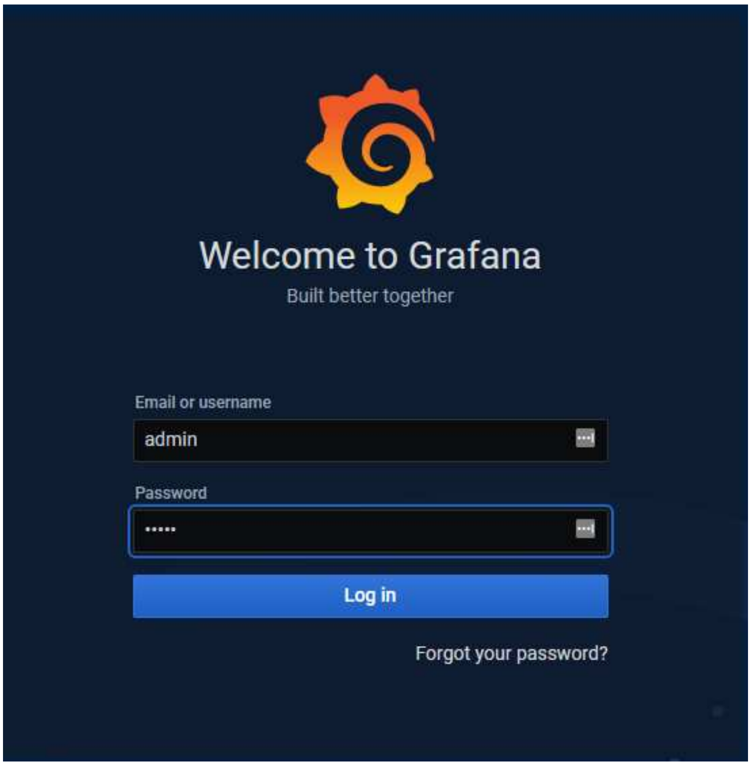

Figure 3.1 Grafana login screen

After successfully logging in with their credentials, the user is greeted with the welcome screen of Grafana (Figure 3.2), where all the available options appear. It should be noted that in the example presented in this report, the screens are of a user with administrative privileges – normal users have less choices available and the environment appears slightly different.

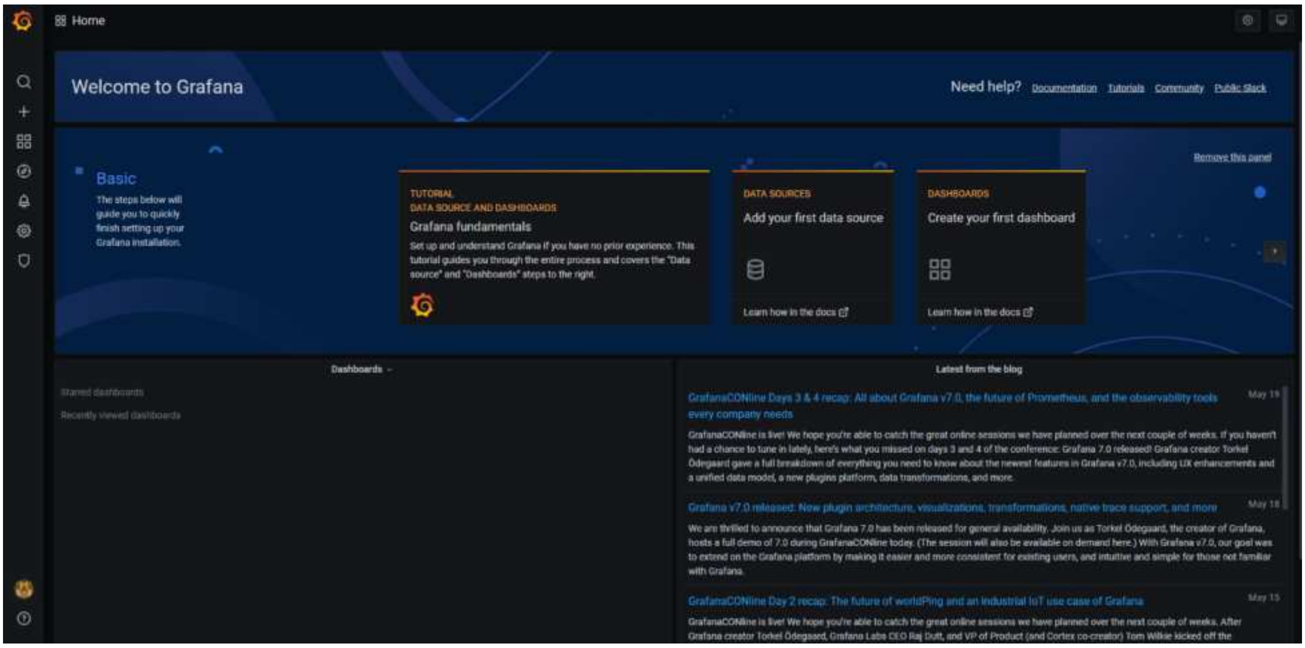

Figure 3.2 Grafana dashboard overview

The first step for the creation of a dashboard, is to connect the Grafana installation with a data source from which the data will be fetched. In this example, the data will be fetched from the data repository's database (Deliverable 5.6).

This is done by clicking on "Add your first data source" at the "DATA SOURCES" tile (Figure 3.3).

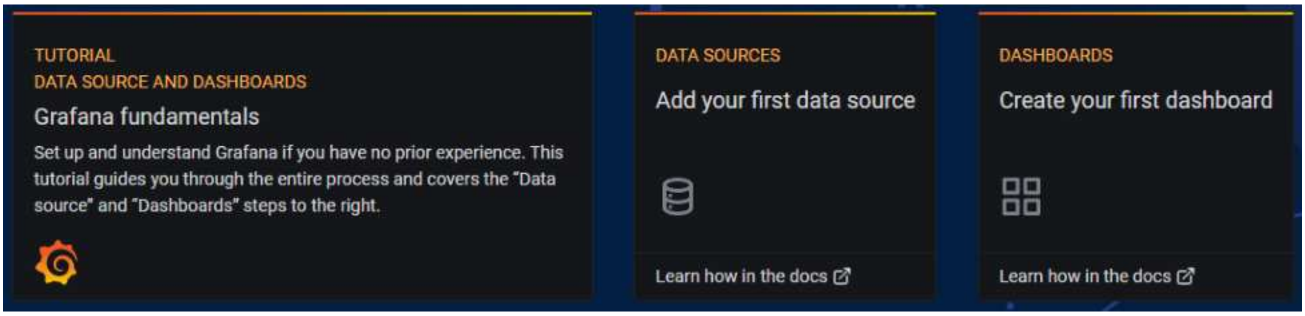

Figure 3.3 Adding a data source

The user is redirected to the "Add Data Sources" page, where they can add a data source from several available options.

In this example, the option to be chosen is the "MySQL" one, under the SQL category (Figure 3.4). The user can then proceed by clicking on the "Select" button.

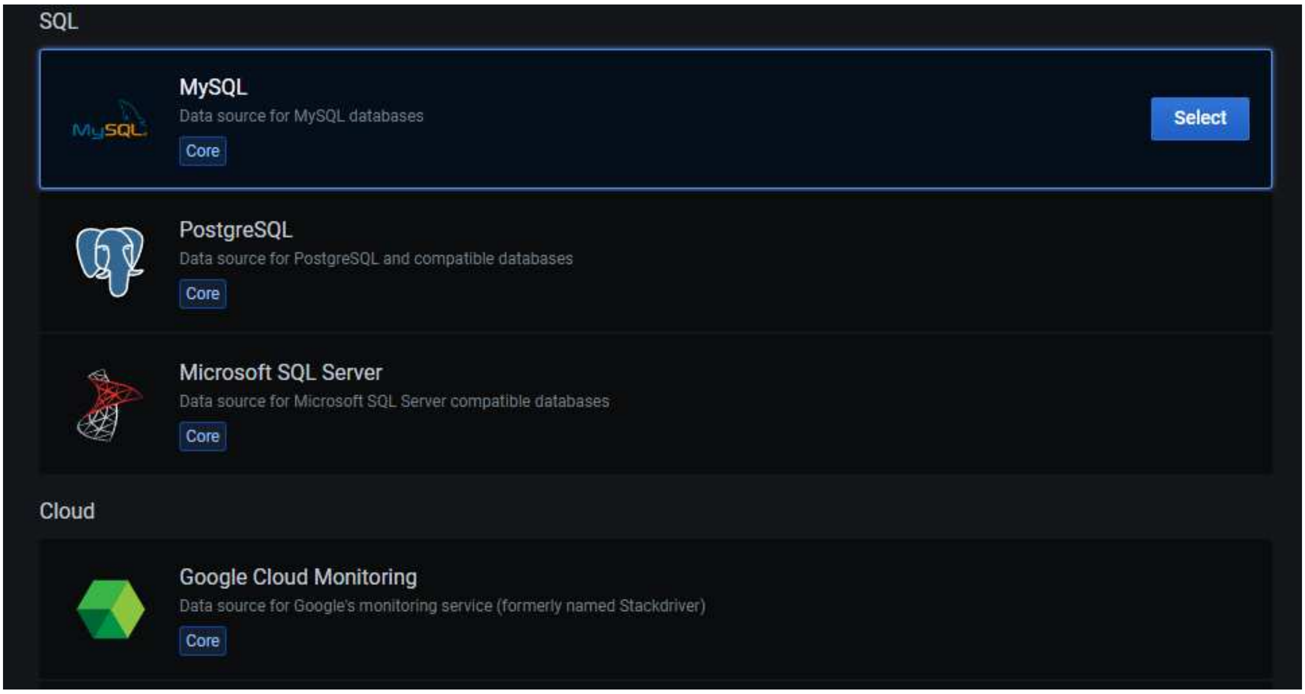

Figure 3.4 Selection of a MySQL database

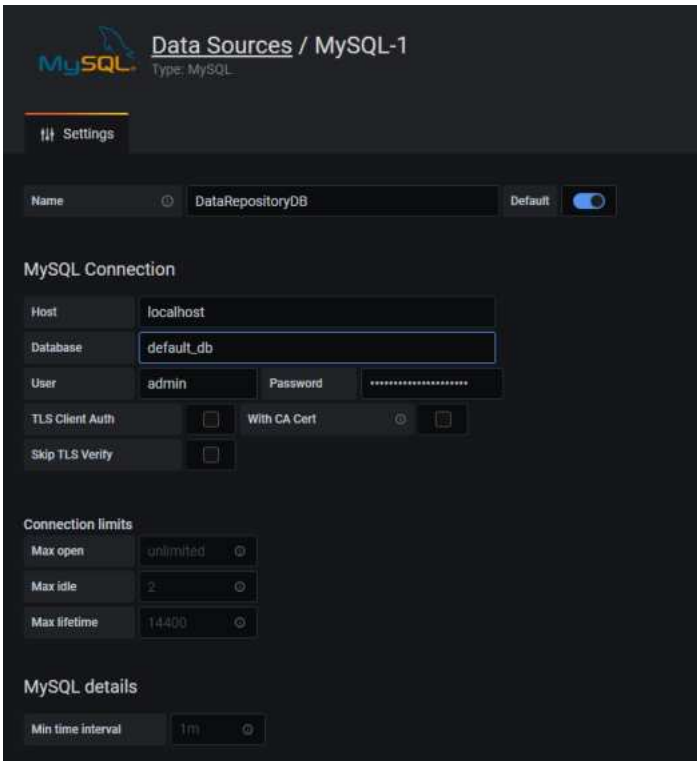

Figure 3.5 Settings for connecting to a MySQL database

If the connection parameters with MySQL (Figure 3.5) are correct and the connection is successful, the notification shown in Figure 3.6 appears.

Figure 3.6 Successful connection to database

Once the data source has been successfully added, the user can proceed to create the dashboard. On the left hand side you need to click on the plus sign (+) and then select "Dashboard" (Figure 3.7).

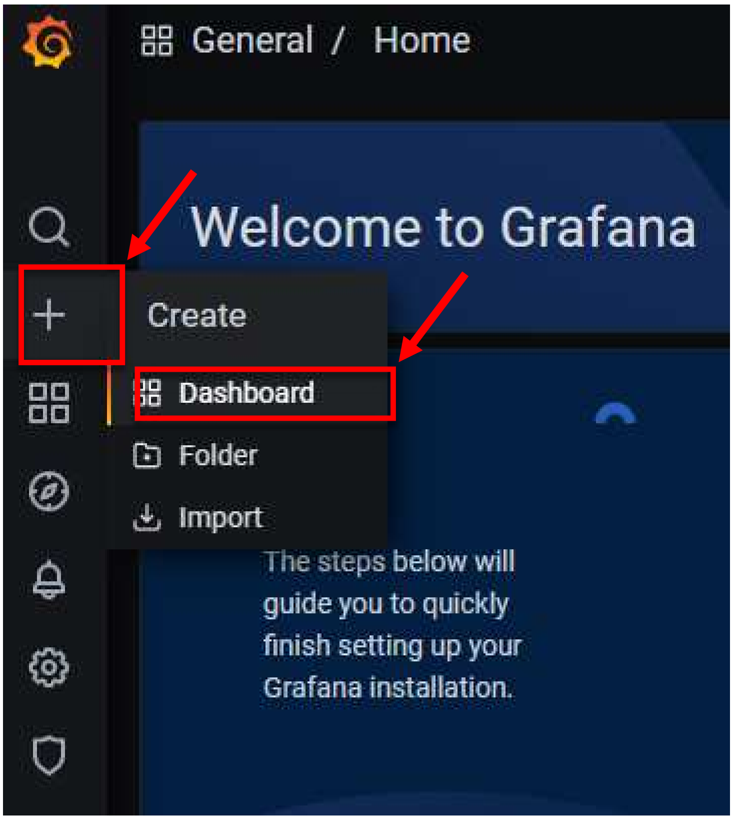

Figure 3.7 Creation of a new dashboard

Each dashboard can consist of panels and rows. To add the first panel, the user must click on the button "+ Add new panel" on the screen that appears after the dashboard has been created (Figure 3.8).

This project has received funding from
the European Union's Horizon 2020
Research and Innovation Programme
under Grant Agreement No 776643

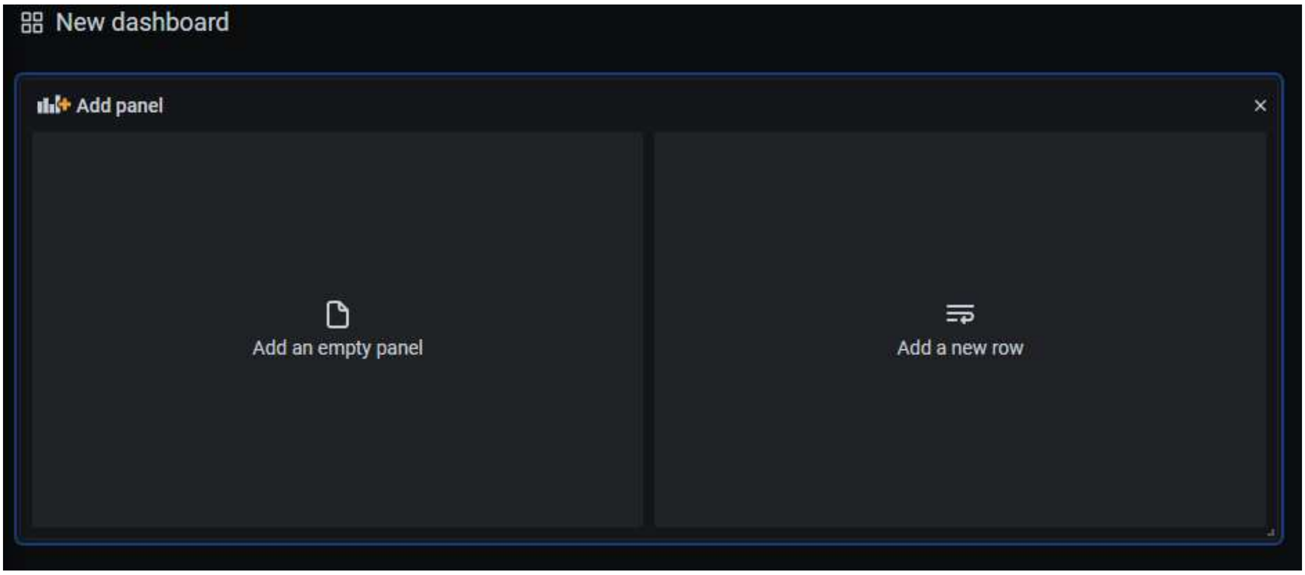

Figure 3.8 Adding a new panel to the dashboard

A new default panel will appear on the newly created dashboard, containing an empty dataset (Figure 3.9).

To change that, the user needs to change the data source and select the data repository database as the panel’s data source.

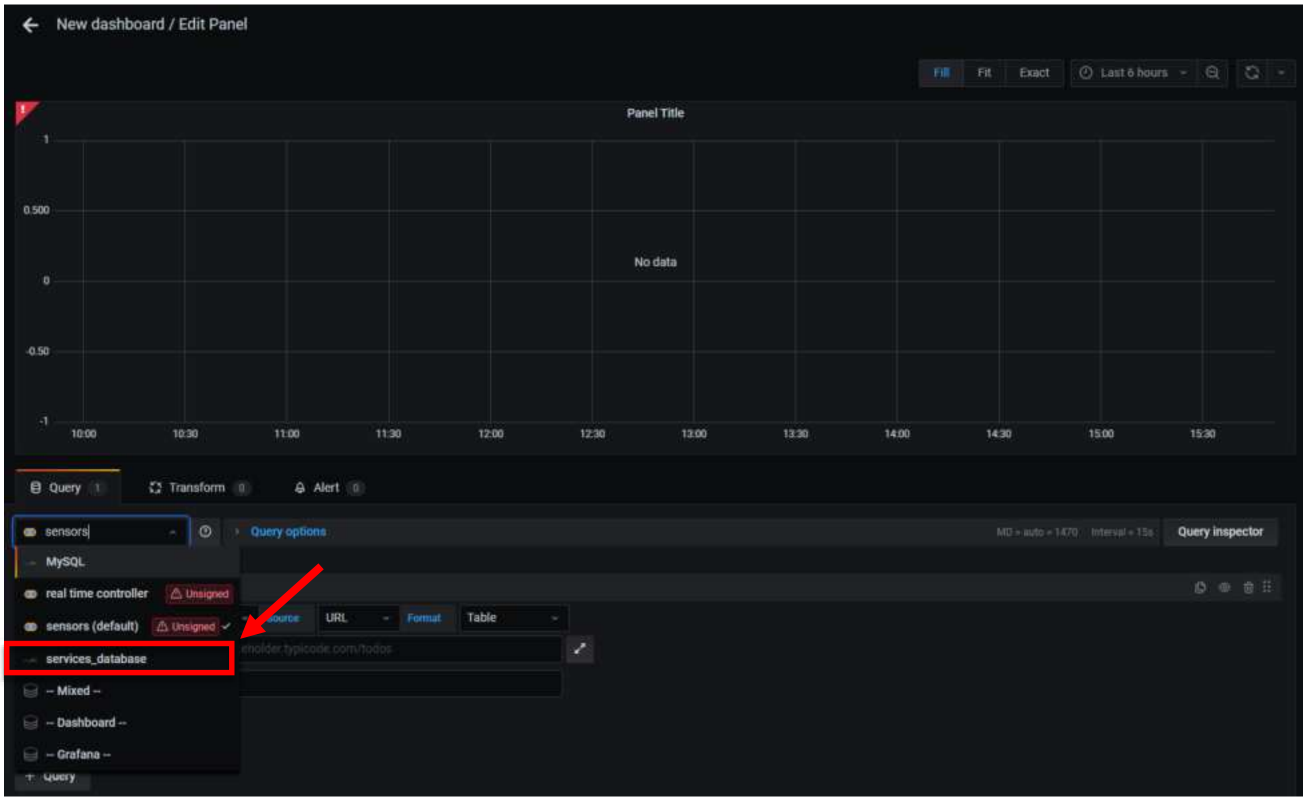

Figure 3.9 Changing the data source

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 776643

When the data source is changed, the chart will appear empty, because the query must be specified in order to fetch the data. When the query is filled (as shown at the text box below) in order to fetch the appropriate data, a graph will be generated (Figure 3.10).

```
SELECT
    timestamp AS "time",
    temperature, humidity
FROM sensor_data
WHERE
    $__unixEpochFilter(timestamp) AND
    imei = 'xxxxxxxxxxxxxxx'
ORDER BY timestamp
```

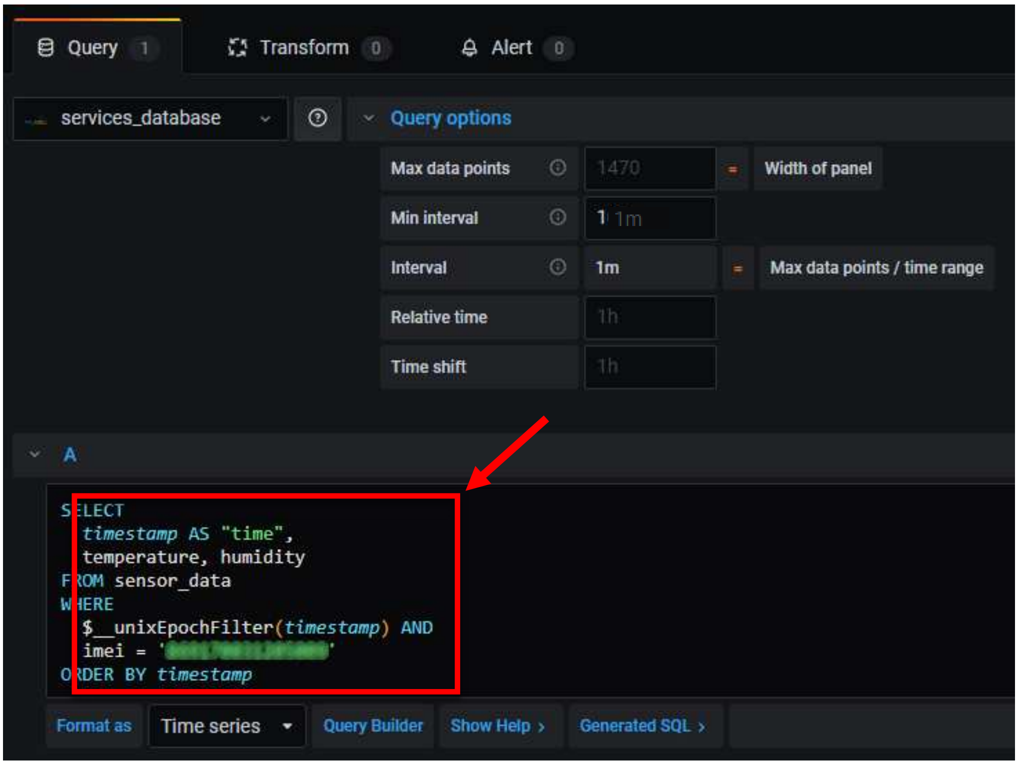

Figure 3.10 Grafana query example

After the query is executed, the panel looks as in Figure 3.11.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 776643

Figure 3.11 Time series panel

Additionally, the user can change several options of the graph, such as the display, the series overrides, the aces, the legend, the thresholds, the time regions, the links and the repeat options (Figure 3.12).

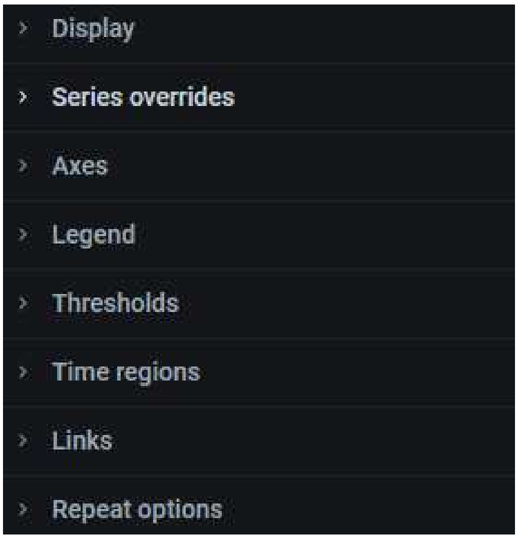

Figure 3.12 Graph options

As mentioned before, a dashboard can be comprised of several rows and panels. Each of these panels can be configured separately, something that increases the dashboard's flexibility.

The list of panels types that the user can choose from are presented in the Figure 3.13 below.

Figure 3.13 Visualization types

4. CREATING THE DASHBOARD FOR HYDRO 3 PILOT

In this chapter, the creation of the Grafana dashboard for one of the HYDRO sites (HYDRO 3) will be explained as an example.

The HYDRO 3 Demonstration site is an innovative sub-surface rainwater harvesting system, located in a remote area in the island of Mykonos, Greece.

The demonstration site consists of the following components for monitoring and controlling its operation:

- Two water tanks which are used for storing the water harvested from the rainwater collection system
- Two water level (distance) sensors for measuring the water quantity in each one of the two tanks
- Two temperature sensors for measuring water temperature in each tank
- One weather station for monitoring environmental parameters of HYDRO 3 demonstration site (e.g. precipitation, light intensity etc.)
- Nine soil moisture sensors for monitoring the moisture in each one of the nine plots of the oregano field located on the same area.
- Nine electrovalves for automating irrigation at each plot separately
- One relay for tuning on the pump when irrigation in one of the plots is needed.

The data transmitted from the pilot, can be summarized as follows:

- Location (Coordinates)
- Water quantity (2 tanks)
- Water Temperature (2 tanks)
- Air Temperature
- Air Humidity
- Wind speed
- Wind gust
- Wind Direction
- Light intensity
- Atmospheric Pressure
- Precipitation
- Soil moisture (9 Plots)
- Status of the pumps (opened/closed)

In order to develop the HYDRO 3 demonstration site dashboard in Grafana, a choice was made to divide the aforementioned data in five (5) distinctive categories:

- HYDRO 3 Pilot (General information)
- Weather data
- 1st Tank
- 2nd Tank
- Soil moisture

For the Grafana implementation, this means that the dashboard structure contains five (5) rows.

Each of these rows, contain panels – this is the name of the widgets that Grafana uses to visualize data measurements or data sets.

Based on the available data, their type and the aforementioned categorization, the rows structured as follows:

- **Row 1: HYDRO 3 Pilot**
  - Location (Worldmap panel)
  - Tank 1 – Water quantity (Gauge panel)
  - Tank 2 – Water quantity (Gauge panel)

- **Row 2: Weather data**
  - Temperature (Gauge panel)
  - Humidity (Gauge panel)
  - Wind speed (Gauge panel)
  - Wind gust (Gauge panel)
  - Light intensity (Gauge panel)
  - Weather station battery (Gauge panel)
  - Pressure (Stat panel)
  - Rain (today) (Stat panel)
  - Rain (month) (Stat panel)
  - Rain (year) (Stat panel)
  - Temperature – Humidity (Graph panel)
  - Wind (Graph panel)
  - Light intensity (Graph panel)

- **Row 3: 1st Tank**
  - Water quantity – 1st Tank (Gauge panel)
  - Temperature – 1st Tank (Gauge panel)
  - Water quantity – temperature – 1st Tank (Graph panel)
- **Row 4: 2nd Tank**
  - Water quantity – 2nd Tank (Gauge panel)
  - Temperature – 2nd Tank (Gauge panel)
  - Water quantity – temperature – 2nd Tank (Graph panel)
- **Row 5: Soil moisture**
  - Soil moisture (Gauge panel) – one for each sensor
  - Soil moisture (Graph panel) – one for each sensor

Summarizing, the Grafana dashboard comprised of 5 rows containing 40 panels in total.

As inferred from the structure above, the unique types of panels that was used are four – the worldmap panel, the gauge panel, the stat panel and the graph panel.

In the sections below, the creation of a panel from the four different panel categories will be explained, using the procedure explained in the previous chapter.

4.1. Worldmap panel example (HYDRO 3 location)

In order to receive the location data (latitude and longitude) from the GPS receiver of the nodes at the HYDRO 3 demonstration site, a properly structured query was needed, having as input the node's IMEI which is the unique identifier used for each node.

The query is as follows:

```
SELECT
  lat,
  lng
FROM imei_coordinates
WHERE
  imei = "106490826305373"
```

After entering the query to the designated field, the visualization type panel (Worldmap panel) was set.

The panel was modified by using the options on the right side of the screen (Figure 4.1).

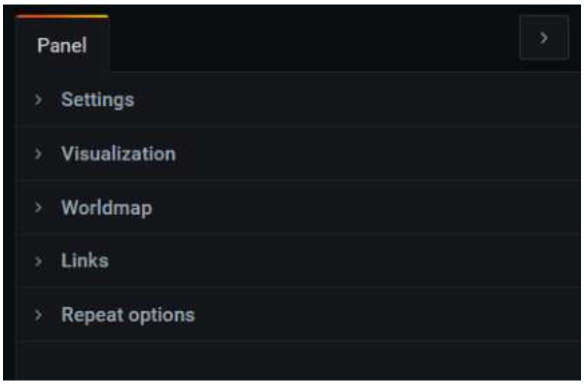

Figure 4.1 Options overview for the worldmap panel

By expanding “Settings”, a title can be given to the panel (Figure 4.2).

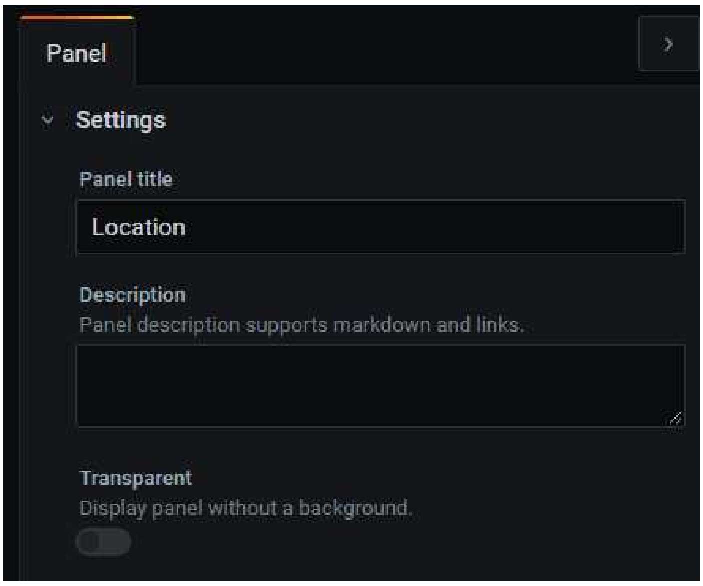

Figure 4.2 Settings for the worldmap panel

After giving title to the panel, the worldmap options was set, containing info such as the center of the map, the initial zoom, the maximum and minimum size of the circle, the location of the data, etc. (Figure 4.3).

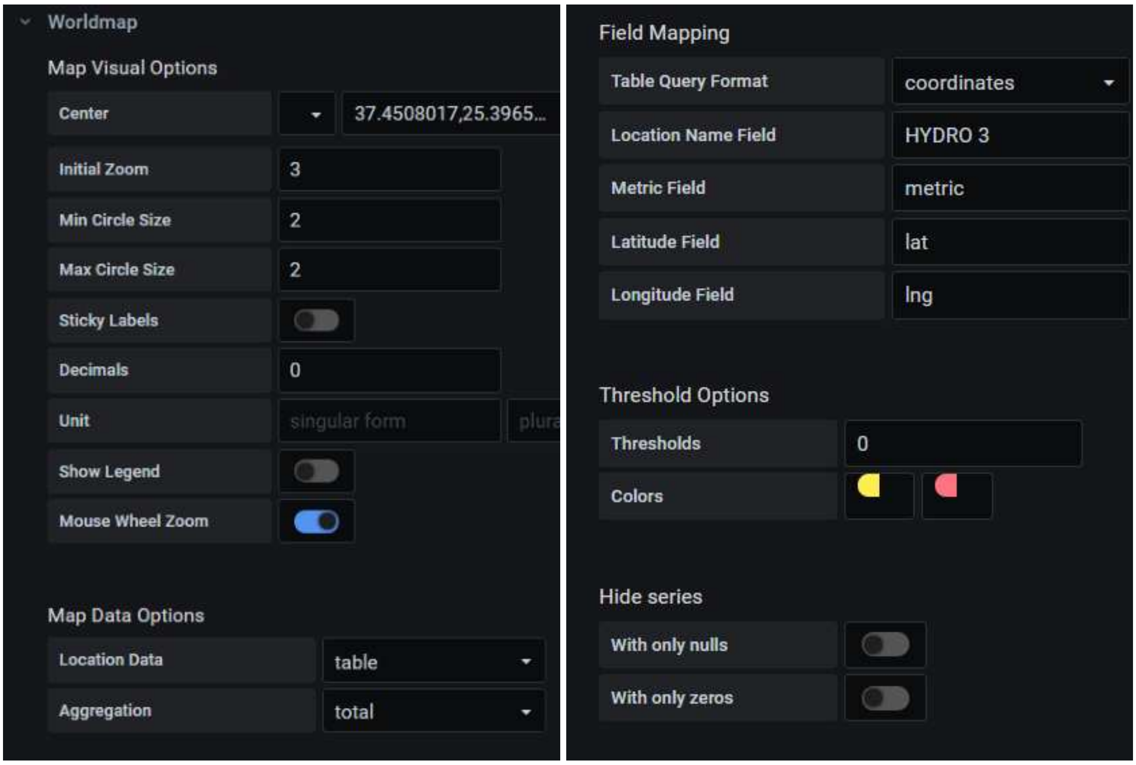

Figure 4.3 Additional worldmap options

The final form of the worldmap panel for HYDRO 3 is presented in Figure 4.4.

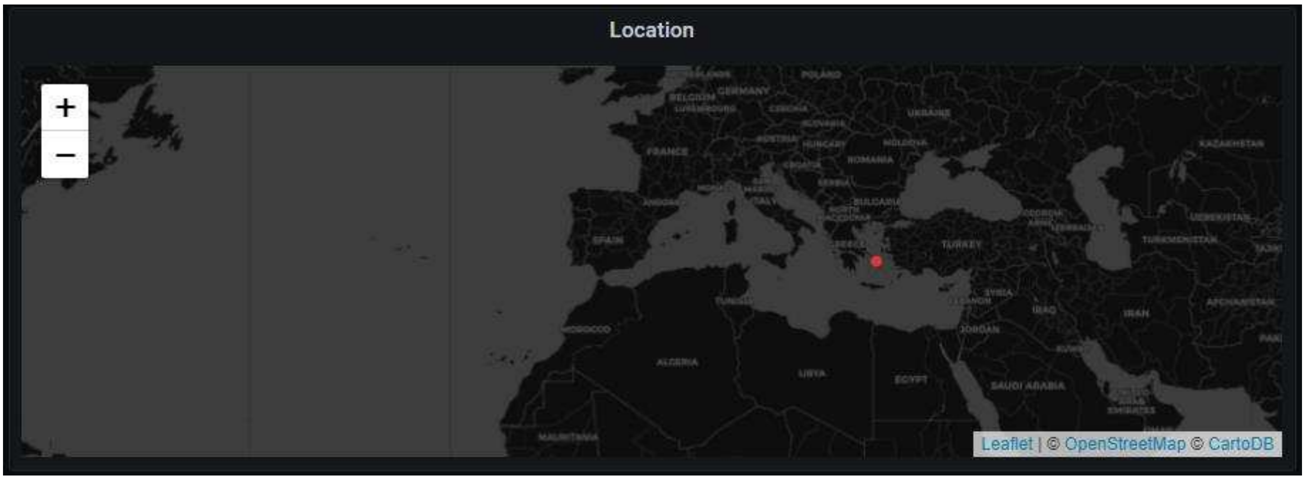

Figure 4.4 Final worldmap panel

4.2. Gauge panel example (Soil moisture)

In order to receive the soil moisture data from the soil moisture sensors at the HYDRO 3 Pilot site, a properly structured query was needed, having as input the node's IMEI in which the sensors have been installed.

The query was as follows:

```
SELECT
  soil_moisture
FROM sensor_data
WHERE
  $__timeFilter(created_at)
  AND imei = "106490826305373"
ORDER BY created_at
```

After entering the query to the designated field, the visualization type panel (Gauge Panel) was set.

The panel was modified by using the options on the right side of the screen (Figure 4.5).

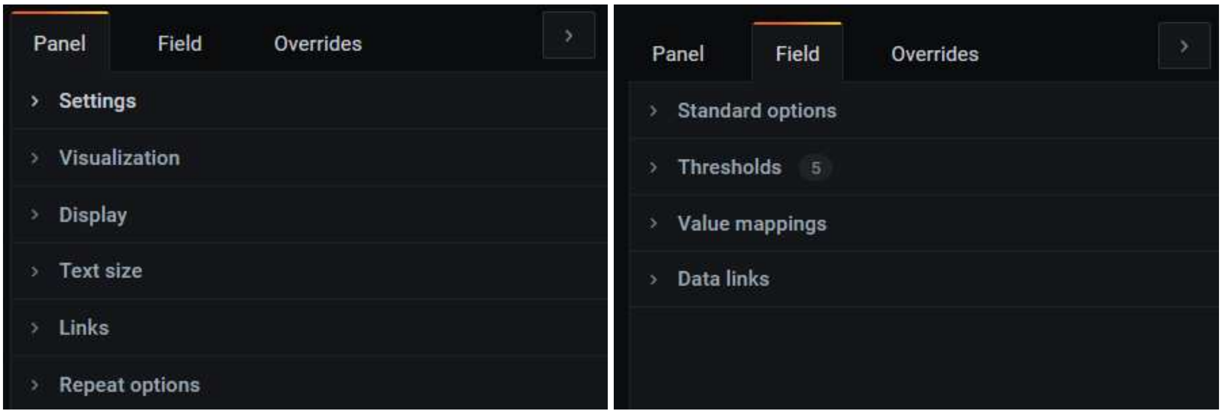

Figure 4.5 Options overview for the gauge panel

By expanding "Settings", a title was given to the panel (section 4.1).

After giving title to the panel, the gauge panel options for display was set (Figure 4.6). In the example of HYDRO 3 Pilot dashboard, the options set to calculate the latest value (instead of showing the series of values), as well as showing the threshold markers.

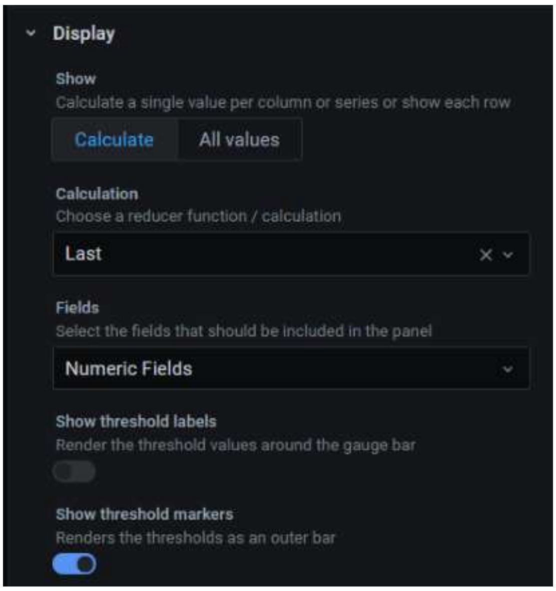

Figure 4.6 Additional gauge panel options for display

In the standard options menu, the units for the panel was set to Percent (0-100), having a minimum and maximum value of 0 and 100 respectively, with 1 decimal. The panel's color scheme was defined from the thresholds menu, where colors for four different thresholds have been set (at 5%, 10%, 20% and 25%), as well as the color of the base-zero value (Figure 4.7).

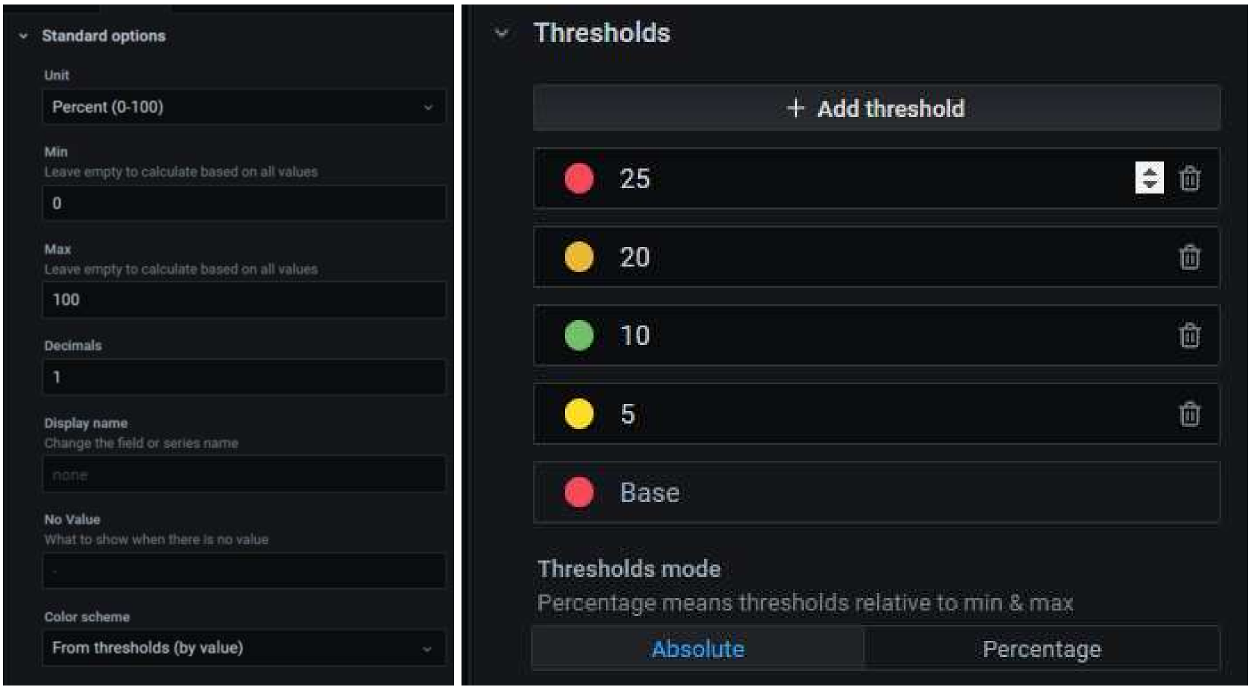

Figure 4.7 Additional gauge panel options

The final form of the gauge panel for soil moisture is presented in Figure 4.8.

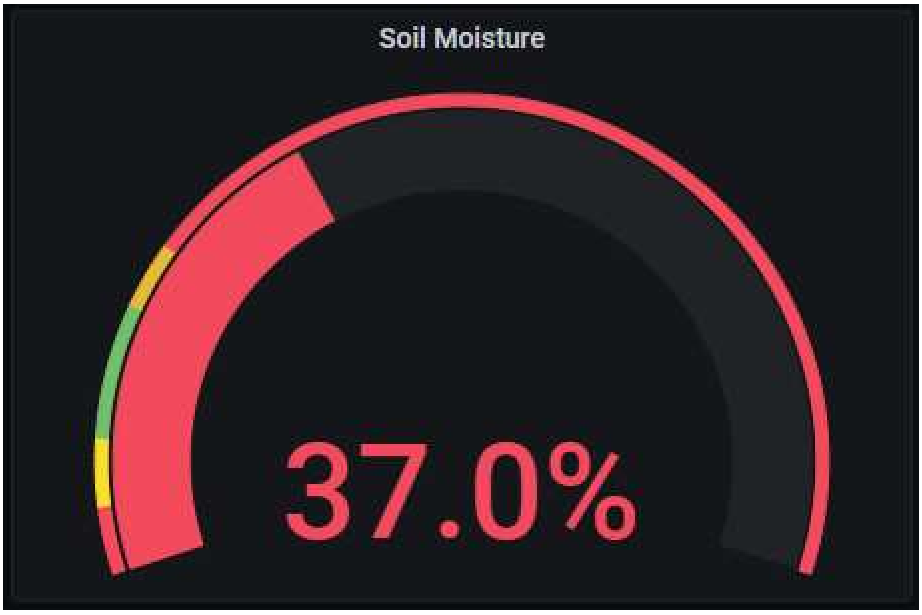

Figure 4.8 Final gauge panel

This project has received funding from
the European Union's Horizon 2020
Research and Innovation Programme
under Grant Agreement No 776643

4.3. Graph panel example (Wind)

In order to receive the wind data (wind speed and wind gust) from the weather station at the HYDRO 3 demonstration site, a properly structured query was needed, having as input the node's IMEI, in which the weather station has installed.

The query was as follows:

```
SELECT
  timestamp AS "time",
  wind_speed,
  gust_speed
FROM sensor_data
WHERE
  $__unixEpochFilter(timestamp) AND
  imei = "106490826305373"
ORDER BY timestamp
```

After entering the query to the designated field, the visualization type panel (Graph Panel) was set.

The panel was modified by using the options on the right side of the screen (Figure 4.9).

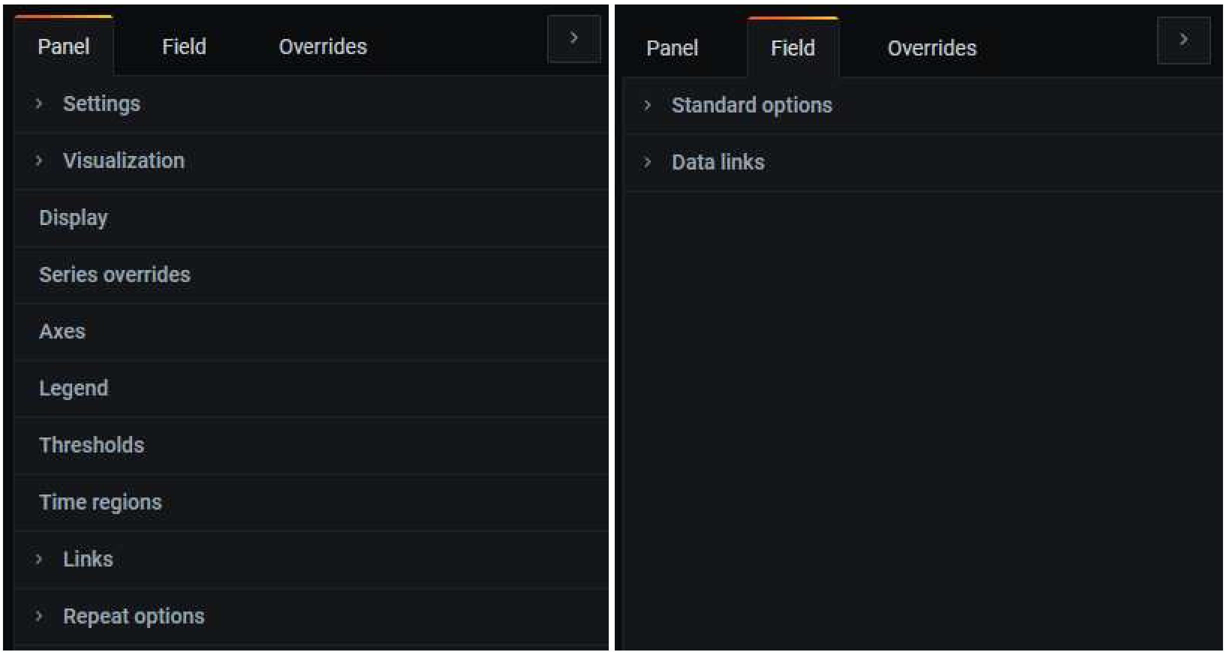

Figure 4.9 Options overview for the graph panel

By expanding "Settings", a title was given to the panel (section 4.1). After giving title to the panel, the graph panel options for display, axes and legend were set (Figure 4.10), as apart from the graph itself, the minimum, maximum, average and current values for both wind gust and wind speed should appear at the graph. Moreover, the units in axis Y were set to meters/second (m/s), while axis X shows time (in 24h format).

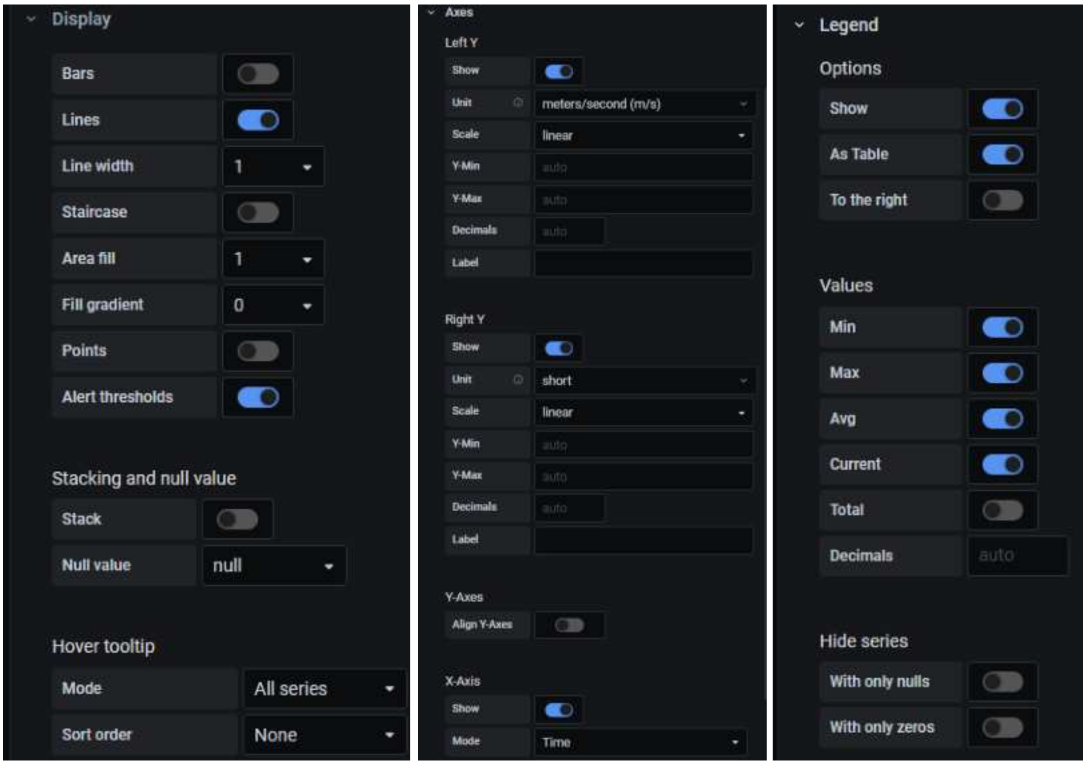

Figure 4.10 Additional graph options for display, axes and legend

Additionally, the units for the minimum, maximum, average and current values for both wind gust and wind speed were set at the standard options of field section (Figure 4.11) as meters/second (m/s).

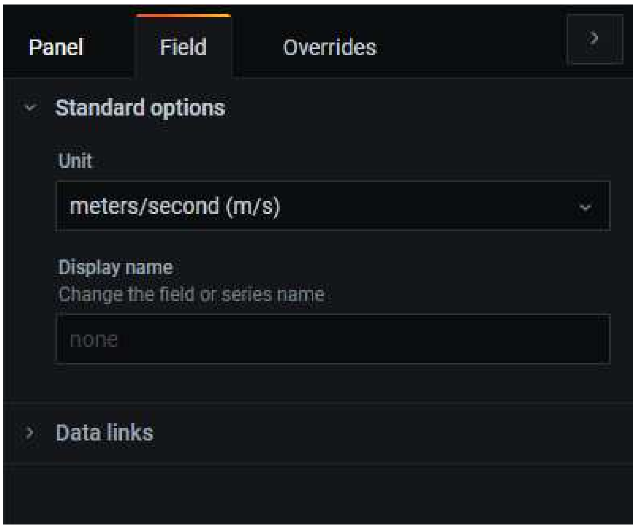

Figure 4.11 Additional graph options field

The final form of the graph panel for wind is presented in Figure 4.12.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 776643

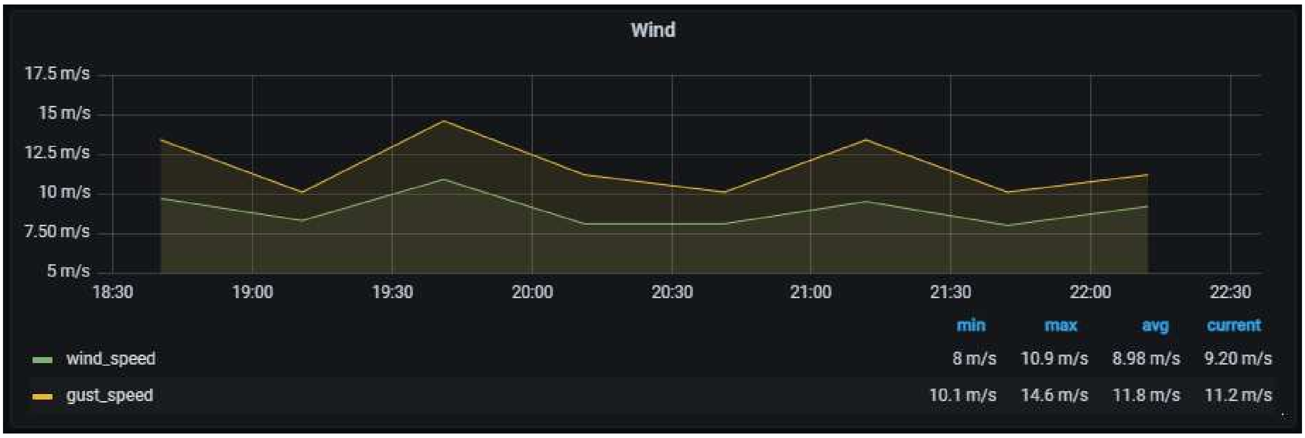

Figure 4.12 Final graph panel

4.4. Stat panel example (Atmospheric Pressure)

In order to receive the pressure data from the weather station at the HYDRO 3 Pilot site, a properly structured query was needed, having as input the node's IMEI, in which the weather station has installed.

The query was as follows:

```
SELECT
  pressure
FROM sensor_data
WHERE
  $__timeFilter(created_at)
  AND imei = "106490826305373"
ORDER BY created_at
```

After entering the query to the designated field, the visualization type panel needs was set.

The panel then was modified by using the options on the right side of the screen (Figure 4.13).

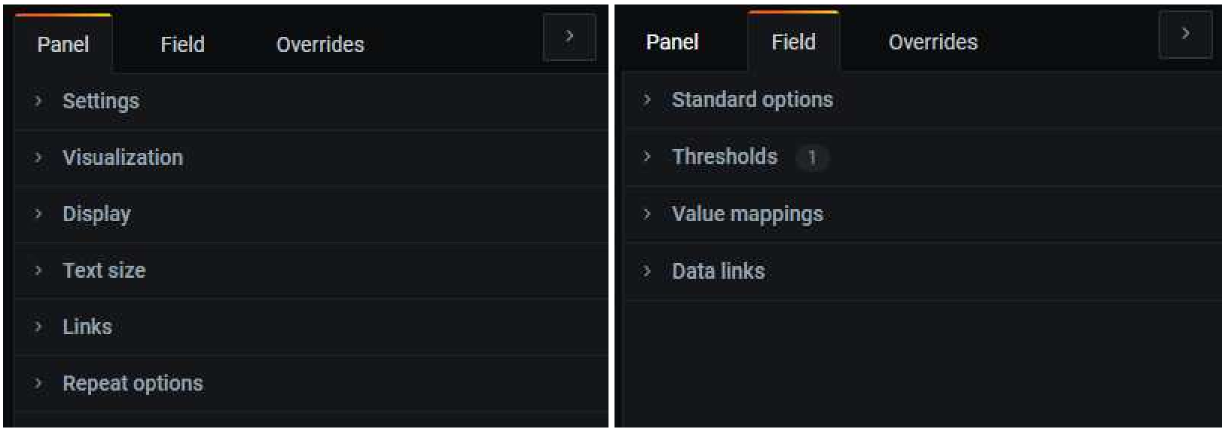

Figure 4.13 Options overview for the stat panel

By expanding "Settings", a title was given to the panel. After giving title to the panel, the stat panel options for display and standard options were set (Figure 4.14). In the example of HYDRO 3 Pilot dashboard, the options set was to calculate the latest value (instead of showing the series of values), setting the displayed color and setting the unit to Hectopascals.

This project has received funding from
the European Union's Horizon 2020
Research and Innovation Programme
under Grant Agreement No 776643

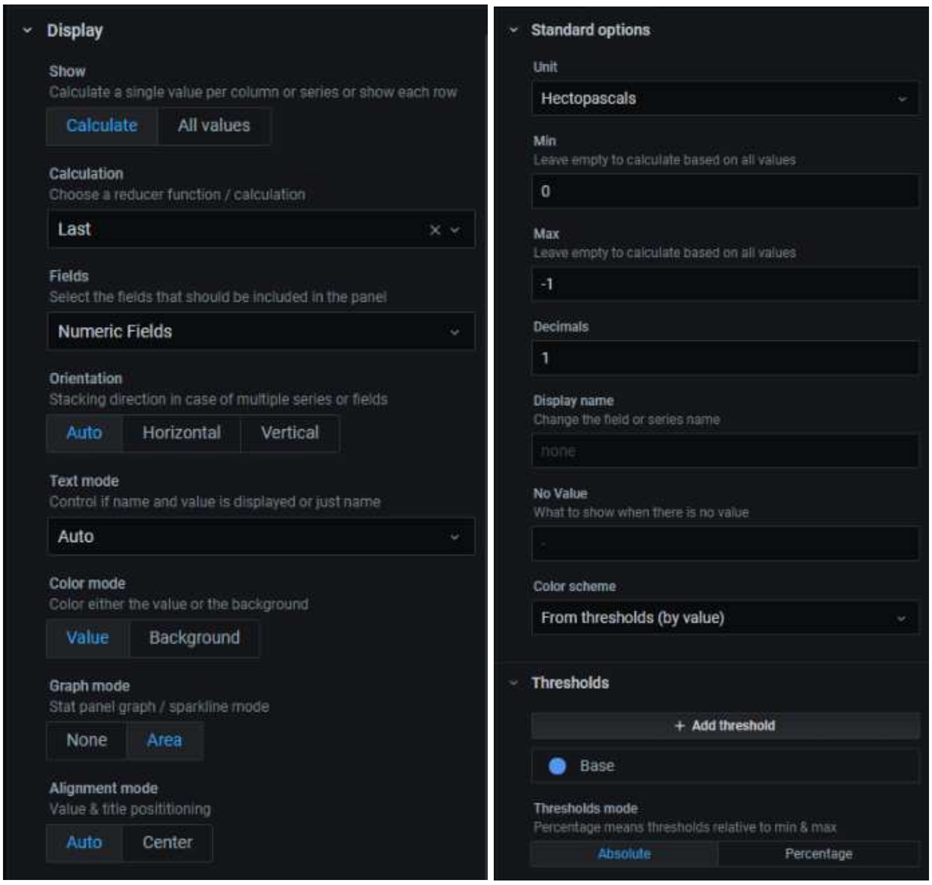

Figure 4.14 Additional stat panel options

The final form of the stat panel for pressure is presented in Figure 4.15.

Figure 4.15 Final stat panel

This project has received funding from
the European Union's Horizon 2020
Research and Innovation Programme
under Grant Agreement No 776643

5. PRESENTATION OF THE DASHBOARD FOR HYDRO 3 PILOT

In the following Figures (Figure 5.1 to Figure 5.6), the final dashboard created for the HYDRO 3 demonstration site is presented.

Figure 5.1 The rows of the HYDRO 3 Pilot dashboard

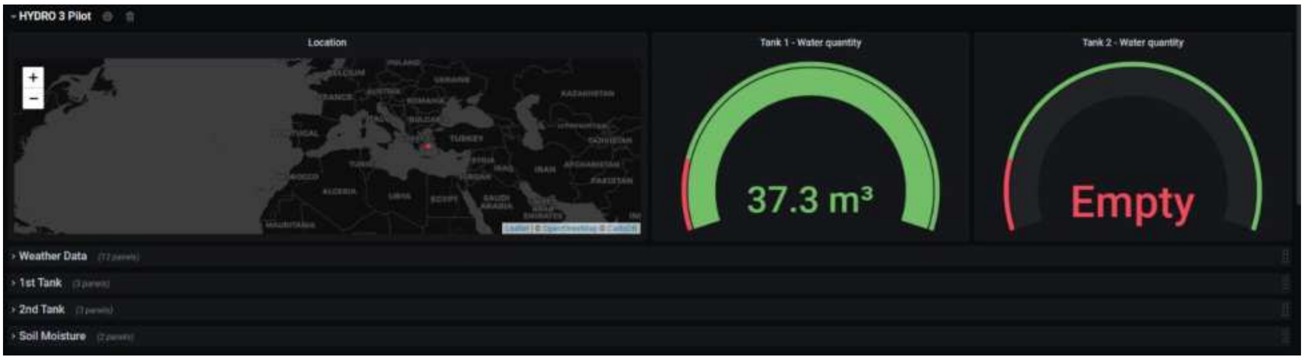

Figure 5.2 General information about the HYDRO 3 Pilot

Figure 5.3 Weather data at the HYDRO 3 Pilot

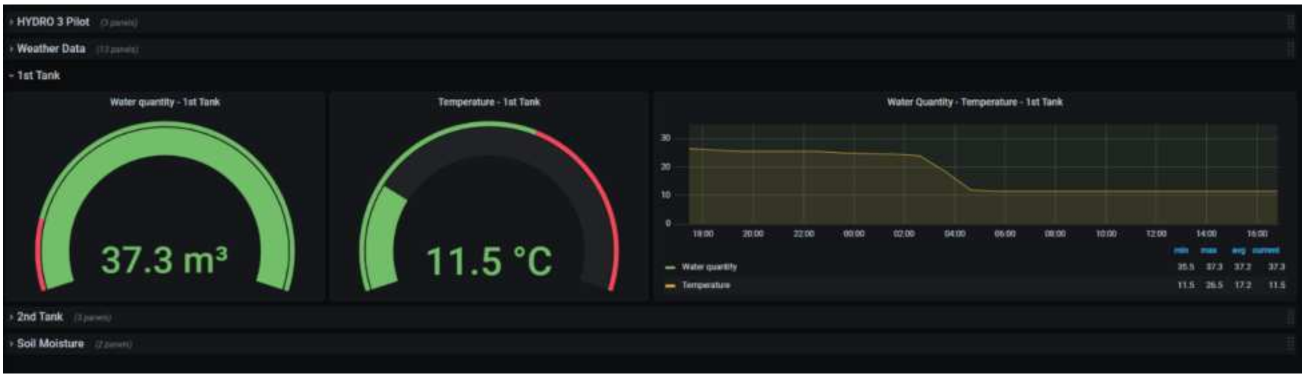

Figure 5.4 Data of the 1st tank at the HYDRO 3 Pilot

Figure 5.5 Data of the 2nd tank at the HYDRO 3 Pilot

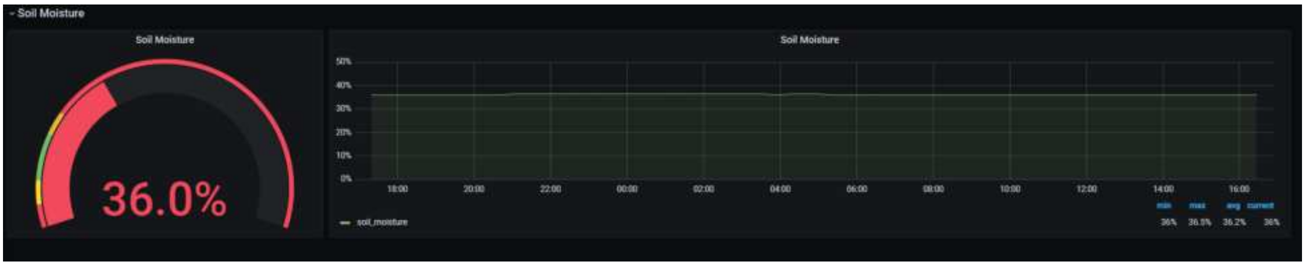

Figure 5.6 Data from the 1st soil moisture sensor at the HYDRO 3 Pilot

For each panel, users can change their size or placement, so that the final layout suits their needs. In Figure 5.7, there's an example of changing the size and placement of the two panels in the soil moisture row. The changing of placement can be done by clicking on the panel and dragging it around to the desired space, whereas the panel's size can be changed by dragging the arrow of the panel's lower right corner to the desired size (Figure 5.8).

This project has received funding from
the European Union's Horizon 2020
Research and Innovation Programme
under Grant Agreement No 776643

Figure 5.7 Changing size and placement of panels

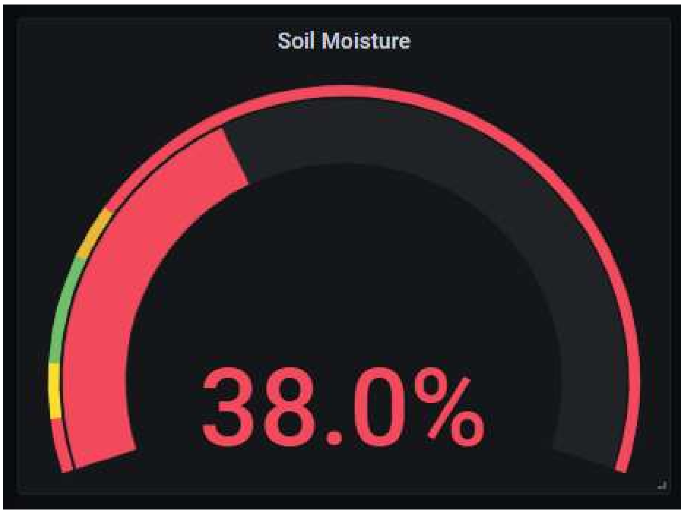

Figure 5.8 Arrow for changing the panel's size

At the top right corner of the dashboard, the user can select the time rage (Figure 5.9) in which the data of the sensors will be projected at the graph panels of the selected dashboard.

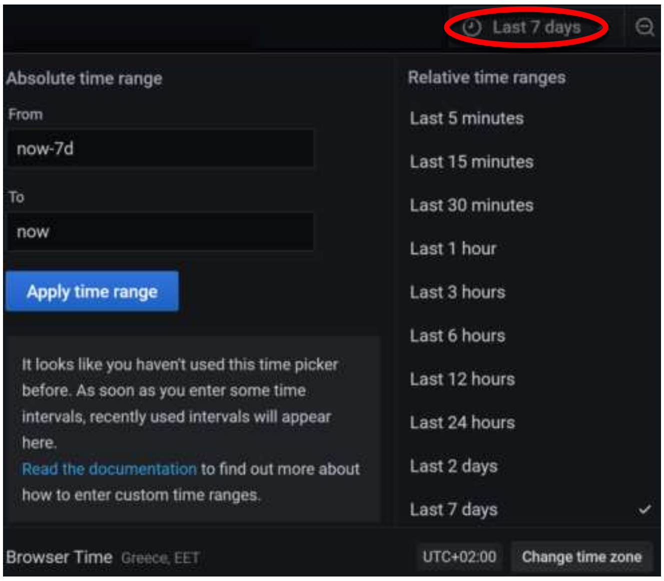

Figure 5.9 Selection of the time range of the dashboard charts

This project has received funding from
the European Union's Horizon 2020
Research and Innovation Programme
under Grant Agreement No 776643

In addition, the user can view and export the data projected at the panels in CSV format (Figure 5.10) by clicking at the panel title menu and then by selecting the option “Inspect”.

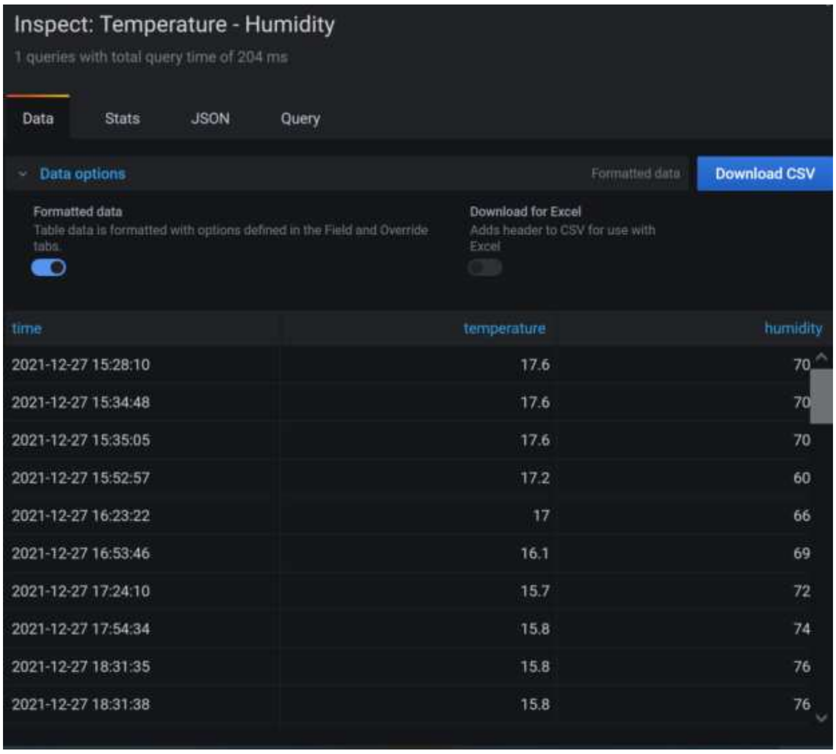

Figure 5.10 Data export

Finally, the user can add custom alerts by creating alerts at any dashboard graph panel (Figure 5.11). An example is given in Figure 5.12 where an alert was created for soil moisture value under 10%.

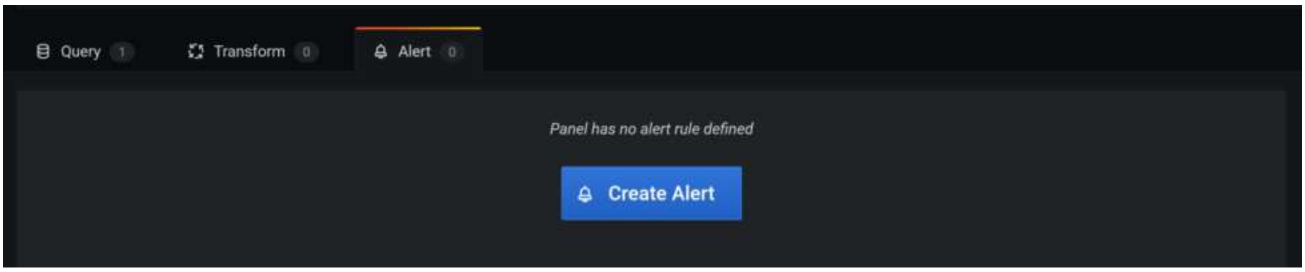

Figure 5.11 Create alert

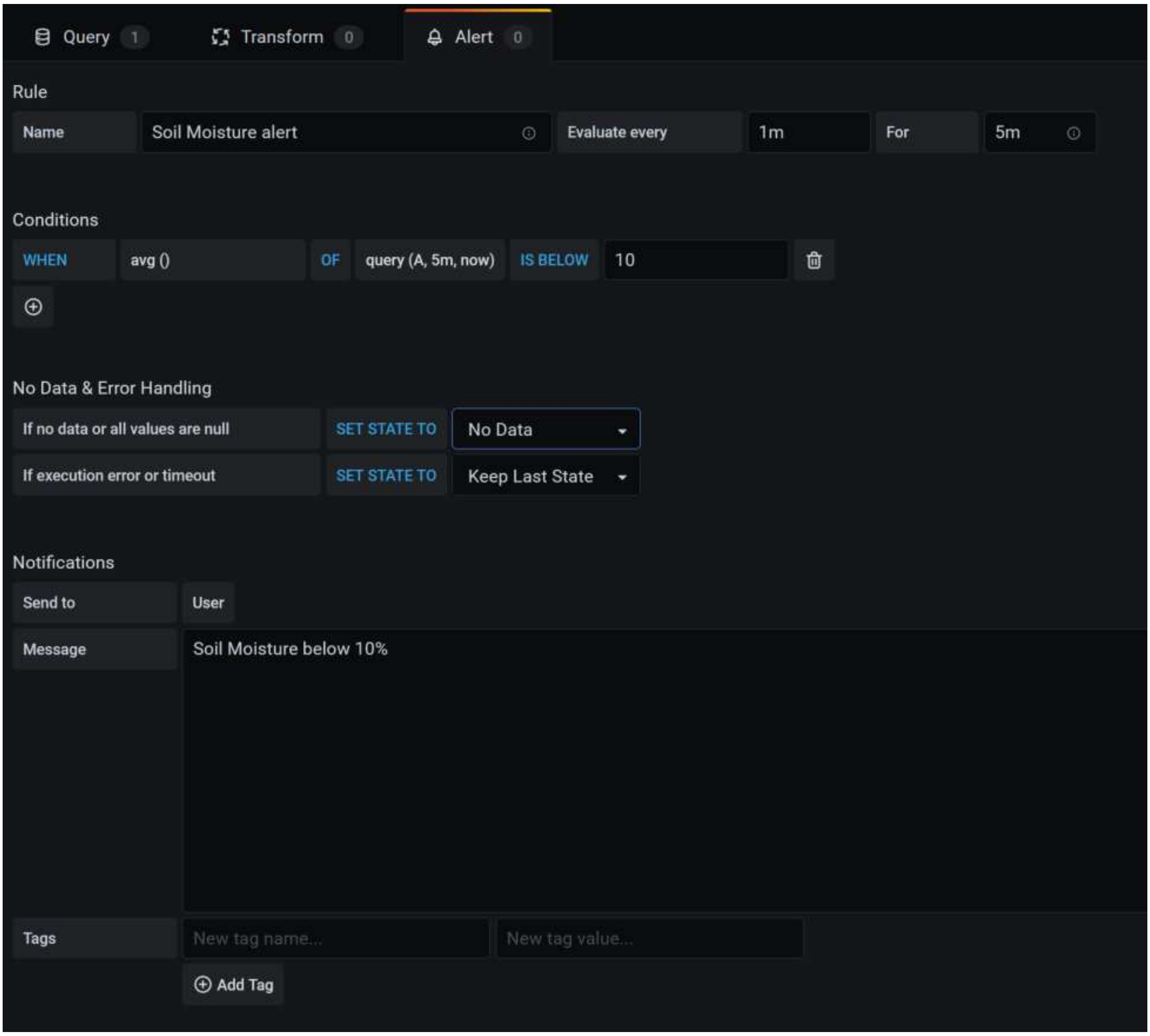

Figure 5.12 Alert rules

6. NEXT STEPS

Since Grafana is a platform originally developed for the visualisation of data, it does not directly support the control of devices of any kind. To enable the system to control ICT systems developed for the needs of the HYDROUSA project, a plugin called "Button Panel" was used.
This button helps in integrating Grafana with any kind of HTTP/REST API. To make the mentioned plugin compatible with HYDROUSA DR (see contribution D5.6), a middleware was developed in the PHP language. As a result, users can control the relays and electrovalves installed at HYDROUSA demonstration sites through the Grafana platform (Figure 6.1).

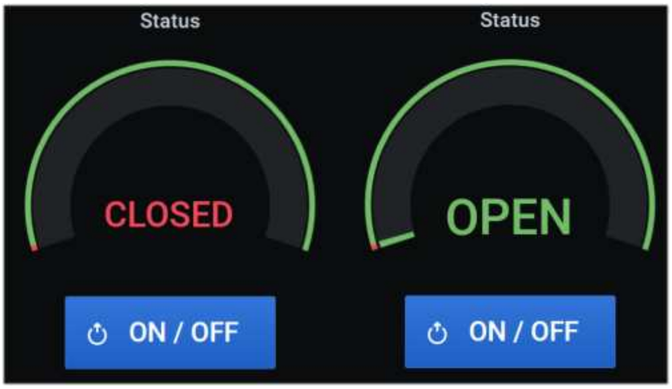

Figure 6.1 Actuator Status (Open / Closed)

At the next period, the controlling functionalities for each HYDRO site will be optimized based on the operators' and users' feedback, and the results for each HYDRO site will be presented at deliverable 5.8.

This project has received funding from
the European Union's Horizon 2020
Research and Innovation Programme
under Grant Agreement No 776643

7. CONCLUSIONS

In the previous chapters, an overview of the Grafana platform was presented, along with a detailed step-by-step process of how to create a dashboard.
These instructions were applied for the creation of a dashboard for the HYDRO 3 demonstration site, which was presented in this report.
The same procedure was followed to create dashboards for all the HYDROUSA project demonstration sites.
Finally, since Grafana is developed for the visualization of data and doesn't directly support the controlling of any device, a description of the solution to enable controlling of relays and electrovalves was given.