



D4.1 Monitoring procedures and environment for data collection from selected pilot cases

PROJECT	CHESTER
PROJECT NO.	764042
DELIVERABLE NO.	D4.1
DOCUMENT VERSION	V2.1
DOCUMENT PREPARATION DATE	05-05-2019
RESPONSIBLE PARTNER	AIGUASOL
DISSEMINATION LEVEL	Public

Type of Deliverable		
R	Document, Report	X
DEM	Demonstrator, pilot, prototype	
DEC	Websites, patent fillings, videos, etc.	
OTHER		
ETHICS	Ethics requirements	
ORDP	Open Research Data Pilot	



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 764042.

This deliverable reflects only the author's views and neither Agency nor the Commission are responsible for any use that may be made of the information contained therein.

EC Grant Agreement	No.764042
Project Acronym	CHESTER
Project Title	Compressed Heat Energy Storage for Energy from Renewable sources
Programme	HORIZON 2020
Start Date of Project	01-04-2018
Duration	48 Months

Financial/Administrative Coordinator	
Project Coordinator Organization Name	TECNALIA
Address	Parque Tecnológico de Bizkaia C/Geldo, Edificio 700 (Spain)
Phone Numbers	+34 946 430 850
E-mail Address	eduardo.zabala@tecnalia.com
Project web-site	www.chester-project.eu

Version Management			
Filename	Requirements of individual technologies that form the CHEST system		
Author(s)	[Name]		
Reviewed by	[Name]		
Approved by	[Name]		
Revision No.	Date	Author	Modification description
V1.0	08/04/2019	A. Carrera	First version
V1.1	15/04/2019	S. Alonso and C. Pascual	Contents added and structure rearrangement
V2.0		A. Carrera	Second version
V2.1	16/05/2019	S. Alonso	Figures modified, and sections added
V3	24/05/2019	A. Sotil	Last version
V4	29/05/2019	C. Pascual and N. Fernandez	Final revision and ready for submission

Contents

1. Introduction.....	7
1.1. Purpose and Scope	7
1.2. Structure of the document	7
2. Monitoring and control	8
2.1. Ispaster case study	8
2.2. Aalborg case study.....	10
2.1. CHESTER monitoring and control platform	12
2.1.1. CHESTER monitoring and control platform architecture	12
2.1.2. Data storage system overview	14
2.1.3. Data visualization system overview	15
2.1.3.1. Security certificate.....	17
2.1.4. CHESTER Graphical User Interface (GUI).....	17
2.1.4.1. Initial page and navigation	17
2.1.4.2. Case-studies pages	18
3. Data forecasting system	22
3.1. Data Forecasting overview	22
3.2. Forecasting methods	22
3.2.1.1. S-ARIMA.....	23
3.2.1.2. Artificial Neural Network.....	24
4. Conclusions.....	26
References.....	27
Appendix 1 – Registered variables in Ispaster	28
Appendix 2 – Registered variables in Aalborg	29
Appendix 3 – Grafana’s user guide.....	31
A3.1. Queries configuration.....	31
A3.2. Graph.....	32
A3.3. Singlestat and gauge	33
A3.4. World map panel.....	34
A3.4. Heatmap.....	35

List of Figures

Figure 1: Principle scheme of DH network in Ispaster, supply and demand side.	10
Figure 2: Simplified scheme of the DH system of Aalborg.	11
Figure 3: Configuration of the monitoring and control system.....	14
Figure 4: Exporting data from Grafana as JSON or CVS.....	15
Figure 5: Grafana's login area.....	16
Figure 6: Users' permissions.....	16
Figure 7: Process to import a certificate in Firefox.	17
Figure 8: Selection of data source in Grafana	17
Figure 9: Navigation bar of CHESTER's GUI.	18
Figure 10: CHESTER's introductory dashboard.....	18
Figure 11: Overall dashboard of the Aalborg pilot – presentation of the local system	19
Figure 12: Overall dashboard of the Aalborg pilot – general figures	19
Figure 13: Overall dashboard of the Aalborg pilot – geographic distribution of the heat and electricity producers.....	20
Figure 14: Part of the climate conditions dashboard of the Aalborg pilot.....	20
Figure 15: Energy Balance dashboard of the Aalborg pilot – heat demand vs production and heat exchange with TTES	21
Figure 16: Energy Balance dashboard of the Aalborg pilot – pie charts	21
Figure 17: Energy Balance dashboard of the Aalborg pilot – detail for a determined point of time.....	21
Figure 18: Residuals graph, histogram, q-q graph and correlogram.....	23
Figure 19: 12-step-ahead prediction of the electricity price in Spain.	24
Figure 20: Representation of the ANN for heat demand prediction.....	25
Figure 21: 24-step-ahead prediction of the heat demand in Aalborg.....	26
Figure 22: Configuration menu of a query in Grafana.....	31
Figure 23: Graph of a time series generated in Grafana.	32
Figure 24: Visualization menu of a graph panel.	32
Figure 25: Time series graph with highlighted time regions.	33
Figure 26: Summary stats available in Grafana/OpenTSDB.	33
Figure 27: Summary stats of time series in Grafana presented as a) singlestat b) gauge.	34
Figure 28: Singlestats of the data presentation system of Aalborg's CHESTER.	34
Figure 29: Thresholds coloring configuration for a) singlestat b) gauge.....	34
Figure 30: Visualization menu of the Worldmap panel.....	35
Figure 31: Heatmap panel	35
Figure 32: Visualization menu of the heatmap panel.	36



List of Tables

Table 1: Variables collected from the DH network of Ispaster. 9

Table 2: Variables collected from the DH network of Aalborg..... 12

Table 3: Remote terminal unit for data collection 13

Table 4: Variables registered in Ispaster – metrics and tags 28

Table 5: Variables registered in Aalborg – metrics and tags 29

Table 6: Accepted syntax to define timestamps. 32

Glossary, abbreviations and acronyms

AIC	Akiake Information Criterion
ANN	Artificial Neural Network
API	Application Programming Interface
BIC	Bayesian Information Criterion
CHEST	Compressed Heat Energy Storage
DH	District Heating
DSO	Distribution System Operator
FTP	File Transfer Protocol
GUI	Graphical User Interface
MAE	Mean absolute error
MAPE	Mean absolute percentual error
MLE	Maximum Likelihood Estimation
MSE	Mean squared error
RTU	Remote Terminal Unit
S-ARIMA	Seasonal AutoRegressive Integrated Moving Average
SCADA	Supervisory Control And Data Acquisition
SSL	Secure Socket Layer
STS	Solar Thermal System
TLS	Transport Layer Security

1. Introduction

1.1. Purpose and Scope

The present document describes the monitoring and control structure developed for CHESTER project pilot cases selected after the description done in D2.1.

The monitoring and control system can be split into two parts: the devices and software structure located on site that reads and sends measured data and the web-based infrastructure that stores and arranges that data and displays it. The data presentation system supports the introduction of external libraries for analytical or data processing purposes. For instance, among the external libraries implemented considering the future needs of WP4 there is an external library that delivers forecasted values of some of the boundary conditions of the pilot cases, as can be the electrical market prices or the district heating network demand.

The overall structure is implemented in Ispaster and partially in Aalborg. In Aalborg, the real time monitoring system is not implemented due to confidentiality and security reasons in the Aalborg district heating plan. Instead, a virtual monitoring system will be set where several parameters of the district heating system that were measured for a period of 1 year are presented on the data presentation system with the forecasted variables.

Furthermore, this document aims to give the reader the necessary guidelines to edit the data presenting system platform.

Measured data will be used to develop load and availability profiles of electricity and thermal energy. These profiles will then be implemented in the CHEST integrated model in Task 4.2., which will allow for optimization the CHEST system in Task 4.3. based on real energy profiles. The developed monitoring platform will serve as basis for the SEMS interface to be developed in Task 4.4.

1.2. Structure of the document

This document is divided in two main sections.

Section 2 presents the designed monitoring and control system for the selected case studies. The architecture of the designed monitoring and control system is explained in section 2.1.1. More details of each component of the system are described in subsequent sections, from the data storage system to the visualization platform.

The data forecasting processes deployed for variable prediction are described in section 3.

2. Monitoring and control

The monitoring and control system consist of those processes and infrastructures to track, review and orchestrate the process and performance of the installations. These systems receive data from several sources that has to be integrated into a single platform enabling a greater knowledge of the current status of the whole installation and the potential issues within it.

This kind of systems are composed by a Supervisory Control System And Data Acquisition (SCADA), several sensors distributed in the installation and external data sources. All data has to be integrated and analysed to later display the most interesting values in a graphical user interface (GUI) that provides the user, at a glance, with an overview of the performance and status of the system.

In CHESTER project there are 7 case studies where the CHEST system could be applied. Nevertheless, for WP4, two case studies have been selected to go deeper in the CHESTER system application. The criteria followed for the selection were the potential of the case study for the CHESTER system and the availability of data. Thus, after several discussions, **Aalborg** and **Ispaster** were clear candidates for going further in the monitoring and smart control strategies that will be developed in WP4. Moreover, Aalborg and Ispaster case studies demonstrated great availability of data and both of them showed a high potential for deploying the CHESTER technology, Aalborg at large scale and Ispaster at smaller scale, which is perfect for analysing the different operational modes for the CHEST system. In the sections below a brief description of the case studies is given, and a more detailed description of both case studies can be found in D2.1.

2.1. Ispaster case study

On one hand, in the case study of Ispaster, the monitoring system deployed before CHESTER project was poor and nearly non-existent: only the most important systems, such as the boiler, the PV panels and the DH temperatures, were monitored independently without any connexion between them. This situation was perfectly aligned with WP4 purpose and enables an upgrade of the monitoring and control system providing CHESTER real time data of the case study. Thus, after analysing the electrical microgrid and the district heating network a new monitoring and control system was designed based on an existing project for the same purpose in the municipality.

Ispaster has an electrical micro-grid (besides the local DSO grid) and a small DH network to supply sustainable energy based on local renewable sources to the public buildings of the town. Local heat is produced by a biomass boiler and solar thermal collectors and electricity is produced by PV panels (

Figure 1). The surpluses of heat and electrical energy are stored by thermal buffers and electric batteries, ensuring the stability of demand and generation. Ispaster does not have a large installation, but it may provide interesting boundary conditions for analysing the potential of a CHEST system integrated into an energy system with a high share of RES.

Table 1 enumerates the variables monitored and

Figure 1 shows the principle scheme of the DH network in Ispaster. A table with these variables and the respective metric and tags that identify them in the storage and presentation systems can be found in Appendix 1.

D4.1: Monitoring procedures and environment for data collection from selected pilot cases

Table 1: Variables collected from the DH network of Ispaster.

System	Monitored variable	Unit
PV system	Monthly electrical energy supplied by batteries	kWh
	Total electrical energy supplied by batteries	kWh
	Monthly electrical energy supplied to batteries	kWh
	Total electrical energy supplied to batteries	kWh
	Monthly electrical energy supplied by the PV system	kWh
	Total electrical energy supplied by the PV system	kWh
	Status of charge of the batteries	V
	Temperature of the batteries	° C
Thermal solar system	Temperature of supply	° C
	Temperature of the collectors	° C
	Efficiency of the collectors	kW
	Daily energy generated by the STS	kWh
	Total energy generated by the STS	kWh
Biomass boiler	Temperature of supply	° C
	Temperature of return	° C
Destination grid	General heat demand	kWh
	Heat consumption – City Hall	kWh
	Heat consumption – Bar	kWh
	Heat consumption – Showers	kWh
	Heat consumption – Restaurant	kWh
	Electrical energy demand	kWh

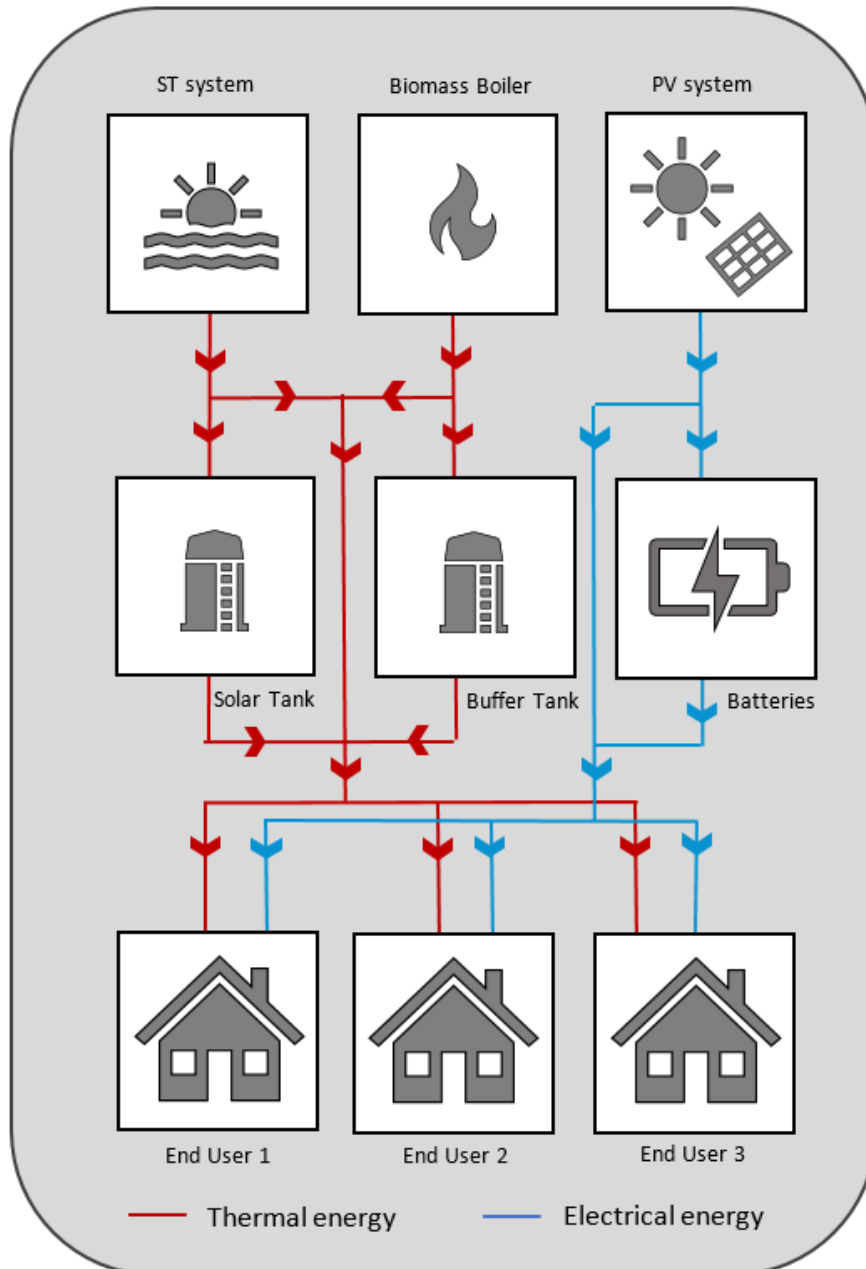


Figure 1: Principle scheme of DH network in Ispaster, supply and demand side.

2.2. Aalborg case study

On the other hand, in the case study of Aalborg, the municipally-owned DH company Aalborg Forsyning that supplies the heat to the city open to provide data about the network but for security and confidentiality issues they are not allowed to provide real time data and so they will provide historical data. Although the data is not in real time, it is of great value for CHESTER to have access to real data from a DH provider. With regards to the control system, this fact does not imply any deviation from the commitments of T4.1 and data has all parameter needed for monitoring and control purposes.

In 2016, 98 % of heated buildings within the area covered by the DH network were connected to the network, for a total number of 36,716 customers. Multiple-apartment buildings are counted as a single customer, so the number of households supplied by the DH network is higher than the number of connections. Most of the heat demand occurs in the period October-May, as there is much less space heating demand between May and September.

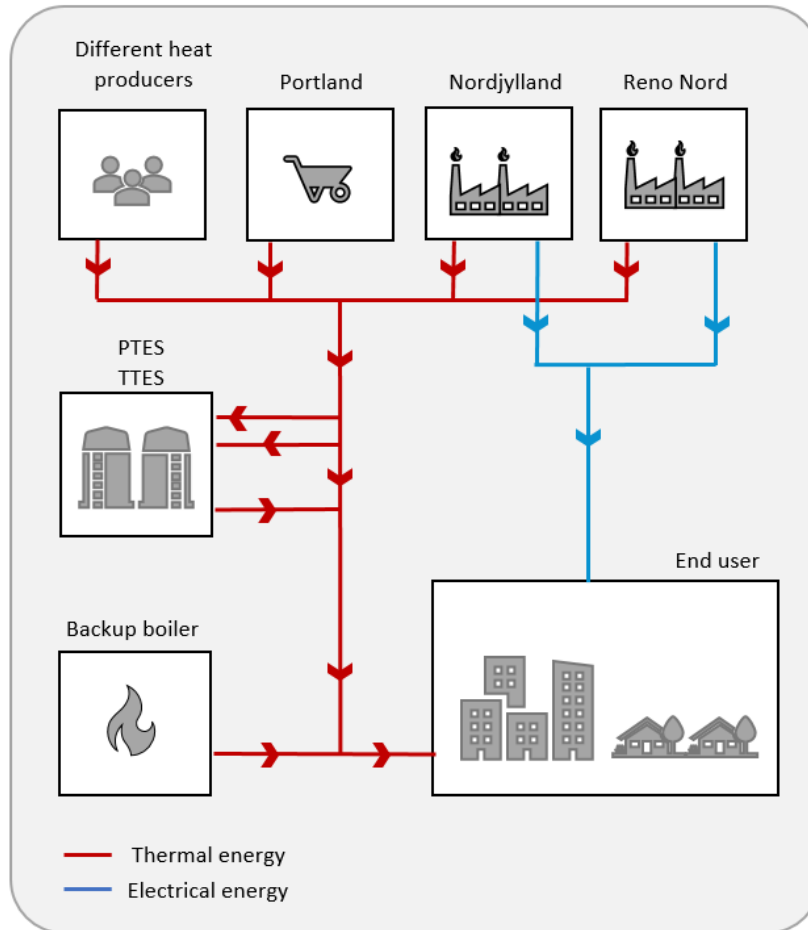


Figure 2: Simplified scheme of the DH system of Aalborg.

A large cement factory, Aalborg Portland, is located just outside the city and supplies the DH network with large amounts of industrial waste heat. There are plans of building a large long-term pit thermal energy storage, so that the excess heat which is not required in summer can be stored and used later in winter. Additionally, Aalborg has a waste incineration combined heat and power plant and a large coal-fired CHP power plant, both of which supply heat to the DH network and electricity to the electrical grid.

Table 2 presents the variables collected from this DH system and Figure 2 provides an insight on how the different actors are connected. Each variable monitored in the Aalborg DH system is identified in detail in Appendix 2.

Table 2: Variables collected from the DH network of Aalborg.

Installation	Monitored variables	Unit
Reno-Nord CHP plant (area L3, L4 and bypass)	Consumed fuel	MWh
	Electricity produced	MWh
	Heat produced	MWh
	Electricity consumed (bypass only)	MWh
Aalborg Portland (VG 1, VG 2 and additional)	Excess heat produced	MWh
	Temperature reduction of the excess heat from Aalborg Portland	MWh
Different small producers	Excess heat produced	MWh
Coal fired CHP plant of Nordjylland	Consumed fuel	MWh
	Electricity produced	MWh
	Heat produced	MWh
	Electricity consumed	MWh
Backup boiler	Heat produced	MWh
	Electricity consumed	MWh
Overall system	Heat transmitted to PTES	MWh
	Heat transmitted from PTES	MWh
	Heat lost to/from PTES	MWh
	Heat content (PTES and TTES)	MWh
	Heat demand	MWh

2.1. CHESTER monitoring and control platform

In CHESTER project two case studies have been selected for monitoring and control purposes as explained before. This section describes the monitoring platform of the CHESTER systems implemented in Aalborg and Ispaster. First, in subsection 2.1.1, the architecture of the system is described. Subsection 2.1.2, presents an overview of the data storage system, followed by the overview of the data visualization system at subsection 2.1.3. In the subsection 2.1.3.1 it is explained how to import the security certificates to access the data storage system. Finally, subsection 2.1.4 describes the GUI created for CHESTER upon the data presentation platform.

Furthermore, clear guidelines on how to use this platform from the editor's perspective can be found at Appendix 3.

2.1.1. CHESTER monitoring and control platform architecture

The CHESTER monitoring and control platform has been designed according to the case studies' needs. This section presents the architecture of the system of monitoring and control and the platforms and components selected for that purpose.

The data collection from Ispaster and Aalborg is made through a SCADA system (Supervisory Control And Data Acquisition). In the real time monitoring system, the selected parameters are read and collected by Remote Terminal Units (RTU), physically connected to the equipment, and sent via Modbus to the central SCADA station. Table 3 describes the type of RTU used in Ispaster.

Table 3: Remote terminal unit for data collection

Part Number	Type	Description
FLEXY10300_00MA	Flexy 103	M2M Data Gateway – 4 Ethernet ports + MPI

As aforementioned, in Aalborg the data available concerns to a 1-year period monitorization. The data obtained from the central station was transferred by FTP (File Transfer Protocol) and stored in a .csv file for posterior analysis and treatment. As mentioned before, the Ispaster system has a constant, real time, monitorization. In Ispaster the central station communicates with an API that converts and loads the received data to the data storage system, OpenTSDB [1]. OpenTSDB is a scalable Open Source data base that can collect, store, and serve data points without any loss of precision, which makes it a perfect solution for the monitoring system.

Among the existing platforms for as for the data visualization, platform Grafana boast powerful visualization capabilities for the T4.1 purposes. Grafana is an open source visualization platform that helps users visualize and understand trends within vast amounts of log data. Besides, Grafana ships with built-in user control and authentication mechanisms that allow you to restrict and control access to your dashboards, including using an external SQL or LDAP server. Grafana can work with multiple time-series data, hence is suitable for represent data stored in OpenTSDB.

As the data presentation system, Grafana [2], is always connected to OpenTSDB, functioning as a graphical interface of the database, the data is displayed in the platform in real-time.

In addition to the monitoring data, forecasted data about the electricity price in Spain and Denmark and of the heat demand of Aalborg's District Heating System are necessary. In order to maximize the efficiency of the overall system, a SEMS (Smart Energy Management System) has to be developed. This system requires forecasted data to determine the optimal strategy to adopt having in mind 24 hours ahead conditions as much as present conditions. The forecasted variables are displayed along with the measured data. These variables are obtained by blocks of Python [3] code that use as input real time data imported from websites via APIs and/or some of the measured parameters imported from the database. This process of calculation and data importation and exportation is programmed to be executed regularly and is explained in more detail in section 3. The code and software necessary to do so are contained in Docker [4], in a local server.

The configuration of the monitoring and control system is shown in Figure 3.

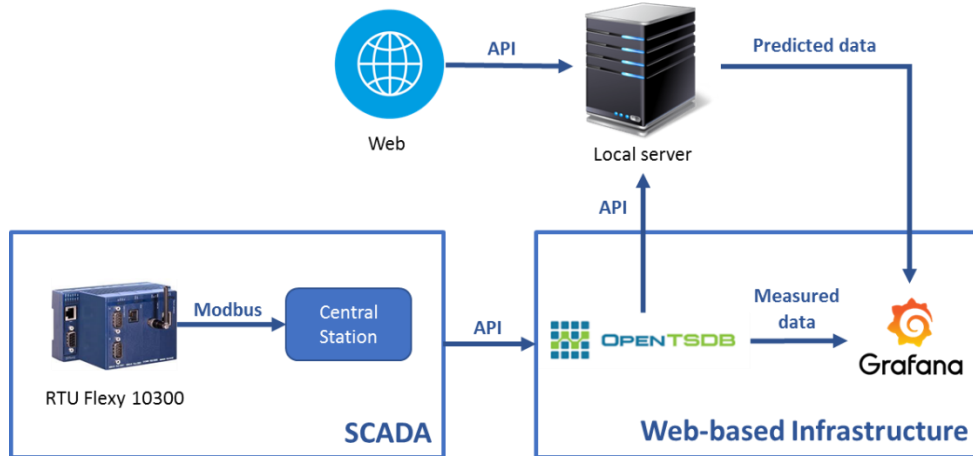


Figure 3: Configuration of the monitoring and control system.

2.1.2. Data storage system overview

The measured data of the monitoring system are transferred and stored in OpenTSDB (Open Time Series Database). OpenTSDB is a free, distributed, scalable database specifically optimized for dealing with time-stamped data. Due to the scalability of HBase, a column-oriented database management system on the top of which OpenTSDB was built, it is prepared to handle and facilitate the storage and analysis of massive amounts of data points generated from a huge number of endpoints with millisecond precision, keeping the raw data forever. [5]

To deliver the full capabilities from OpenTSDB resorts to two programming interfaces: a reading API, to receive information from onsite endpoints, and a writing API that bridges the gap between the database and GUI (graphical user interface) platforms are used, enabling an easy integration with the data presentation system.

Most of the features of the database are still available through the API, such as querying time series data, managing metadata and storing data points. Hand to hand with Grafana's features, it is possible to treat and represent the data in the platform as flexibly as in OpenTSDB interface itself. In this regard, the most relevant call of the writing API is the Query, which allows the extraction of data from the storage system. The following request parameters are mandatory:

- Start - A relative or absolute timestamp (string or integer) that indicates the start time for the query.
- Queries – At least one subquery.

For their turn, each subquery must be defined by, at least, the following fields:

- Aggregator – a string that represent the aggregation function to use,
- Metric – a string that points the name of a stored metric.

Each time series is identified by the metric designation and a tag composed by a key/value pair. This feature allows a quick and flexible treatment and display of the data. For instance, as the same metric name can store more than one time series, if there is not any tag specified when defining the subqueries, all the time series under the same metric will be aggregated according with the function chosen at the aggregator field. If it matters to represent one particular metric that shares its name with other timeseries, the tag field must be set as it will work as a filter.

Besides the tag, there are other non-mandatory fields that can be used to define the queries, such as the rate, that enables the visualization of the rate of change between points, or the down sample, that shifts the data resolution by aggregating the data points in time, as determined by the mode selected by the user.

All these parameters can be defined on Grafana when designing the panels for showing the data.

2.1.3. Data visualization system overview

Data visualization is one of the key features of a monitoring and control system. An adequate user graphical interface can help to explore big amounts of data and have at a glance an overview of the status and performance of the system.

Nowadays, there are a number of open source tools for data visualization that provide great versatility and integration with data base management tools like Data Studio, Dash or Grafana. However, some of them need to have advanced programming skills as Dash or Python.

Grafana and Data Studio are two popular open source tools that help users visualize and understand trends within vast amounts of log data. Data Studio is a part of the products that Google offers for data analytics, but it is still in beta version which means that some functionalities could change and that could have some bugs. Besides, Data Studio has some issues with data sources while merging data and combining sets of data. In this sense, Grafana can be used with a variety of different data sources although is most commonly used with OpenTSDB. This versatility in the data sources along with the simplicity in the creation and edition of the dashboards make Grafana a great choice for implementing the data visualization interface within the CHESTER project.

As already mentioned above, Grafana is an online open source platform for data visualization, monitoring and analysis in real time that allows the user to create and customize data dashboards. The panels in the dashboards can be saved externally as JSON files and the data exhibited can be exported as csv files. These options are accessible through the title of panel, as exemplified in Figure 4.

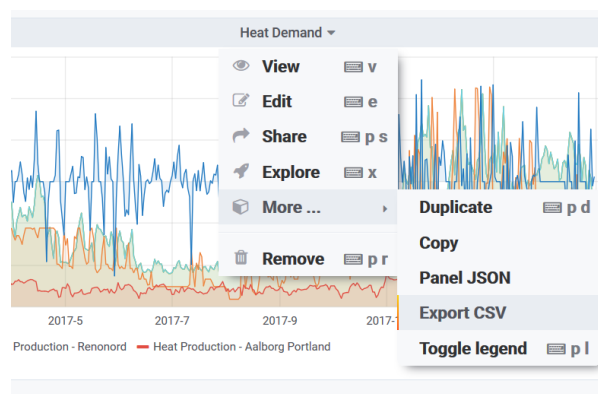


Figure 4: Exporting data from Grafana as JSON or CVS.

Two of the most remarkable perks of this type of platform are the accessibility of data and the design-oriented usability: being web-based it can be easily accessed by any device with a browser and does not require installation of software; and as a graphical user interface it enables an easy customization of the data visualization. Furthermore, it is easier and cheaper to obtain

updated versions and to find assistance, and the centralization of data facilitates its security and backing-up.

The communication protocol for transferring information used in the data presentation system is HTTPS, the secure version of HTTP that resorts to SSL/TLS protocols to encrypt the information, ensuring the security of sensitive data.

A Grafana page can be accessed by different users, under authentication with their own credentials in the login area (Figure 5).

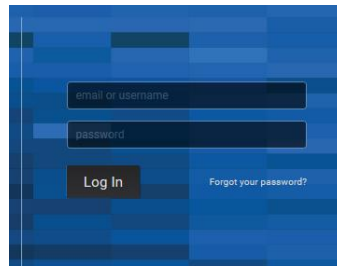


Figure 5: Grafana's login area.

The platform has a permit system that limits the level of access each according to the type of permit attributed: admin, editor or viewer (Figure 6). The admin (usually the owner of the page) has total control over the features of the platform, hence is able create or delete dashboards, panels and data sources, add and remove users and manage their permits. The editor can't create or erase data sources or users but has total clearance at the dashboard level, which means that is allowed to create, delete and edit dashboards and the panels within according with Grafana's data treatment abilities explained in the following sections (such as data aggregation by metric, definition of the data resolution or selection of the time range). Lastly, by default the viewer can only visualize the content of the dashboards, although it is possible to enable this type of user to modify panels and queries without saving them. This option allows the viewer a more extended analysis of the data while ensuring the protection of the existing visualization design.

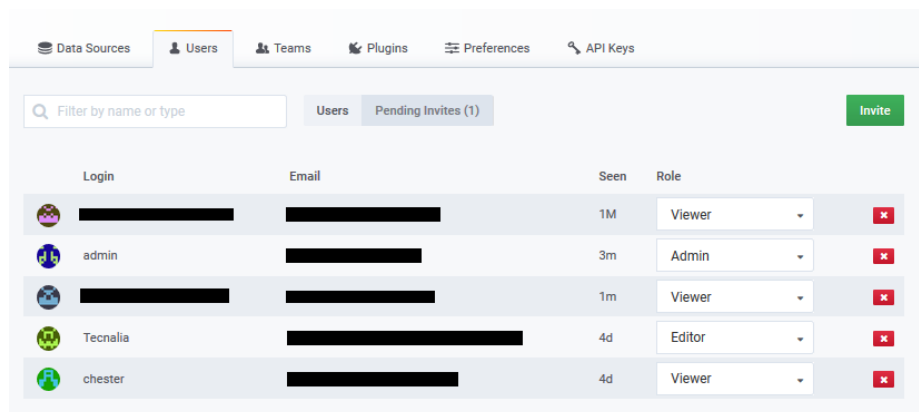


Figure 6: Users' permissions.

The monitoring web platform is accessible through the URL: <https://datuma.aiguasol.coop/db/api/>, and the consortium partners receive privately the necessary user and password.

2.1.3.1. Security certificate

For security matters, to access to the data stored in OpenTSDB of CHESTER project, it is necessary that the user's browser holds the authentication certificate. To do so, the user must possess a ".p12" extension file: "generated.p12". This type of file is generally used to group a private key with its certificate.

The following figure (Figure 7) explains the steps to follow to import a new certificate in Firefox:

```
Firefox -> Options -> Privacy & Security -> Security -> Certificates -
> View Certificates...
On tab "Your Certificates" click "Import" and select "generated.p12"
file. The browser will ask for a password.
Then on tab "Authorities" find "KrakenCA" and click "Edit Trust...".
Select "This certificate can identify websites".
```

Figure 7: Process to import a certificate in Firefox.

2.1.4. CHESTER Graphical User Interface (GUI)

CHESTER User Graphical Interface (GUI) is built upon data dashboards. The dashboards are composed of individual panels that can be dragged and dropped, rearranged on the dashboard and resized to achieve the most convenient configuration.

Each panel is able to handle data from any configured Grafana storage backend. Currently the platform supports the following data sources: Graphite, Prometheus, InfluxDB, Elasticsearch, Google Stackdriver, AWS CloudWatch, Azure Monitor, Loki, MySQL, PostgreSQL, Microsoft SQL Server (MSSQL), OpenTSDB and Testdata. Figure 8 displays the default options available when adding a data source.

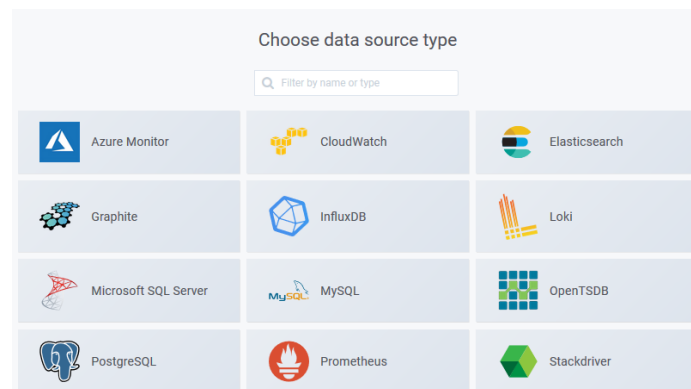


Figure 8: Selection of data source in Grafana

2.1.4.1. Initial page and navigation

CHESTER GUI is composed by 7 dashboards accessible through the navigation bar on the top of each of them (Figure 9).

D4.1: Monitoring procedures and environment for data collection from selected pilot cases



Figure 9: Navigation bar of CHESTER's GUI.

The dashboard identified in the navigation bar as “About”, depicted at Figure 10, acts as a welcome page. It displays a schematic image of the CHESTER system and a brief description of the project and its goals, along with a map pointing the location of all the case-studies of the project and highlighting Ispaster and Aalborg.

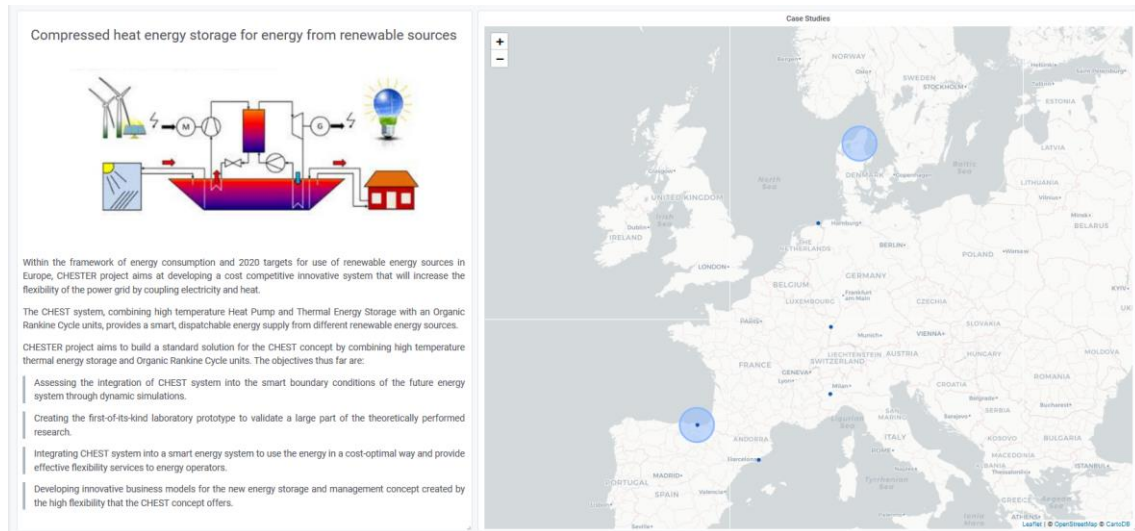


Figure 10: CHESTER's introductory dashboard.

The map panel allows to associate values to a location. Therefore, whereas the points of the other case-studies are merely indicative of their location, the larger dots that identify the case-studies of Aalborg and Ispaster represent the annual average heat demand of each DH system, as it is an indicator of the magnitude of the overall system installed in the region. The exact value can be seen by placing the mouse over the dot.

2.1.4.2. Case-studies pages

The Aalborg and Ispaster buttons of the main navigation bar open a sub-menu with three more buttons that lead to the three dashboards of the corresponding pilot case. These dashboards are structured similarly according to the category:

- the Overall dashboards open by default when a case-study is selected in the main menu. It displays generic information about the existing system of that case-study, a diagram of the installation (

Figure 11), the averages of the most significant monitored variables (

Figure 12) and a map with the distribution of the different suppliers around the cities. For the case-study of Aalborg, depicted below as example, the gauges display the daily averages of heat demand, heat and electricity production and stored heat of all the

system. It is the aggregation feature of Grafana that allows an easy stacking of variables of the same type, even when prevented from different reading sites. Through the heatmap panel, the same values are effortlessly treated by the platform that returns their histogram. This representation can be a valuable tool to analyse the behaviour of the overall system along the time.

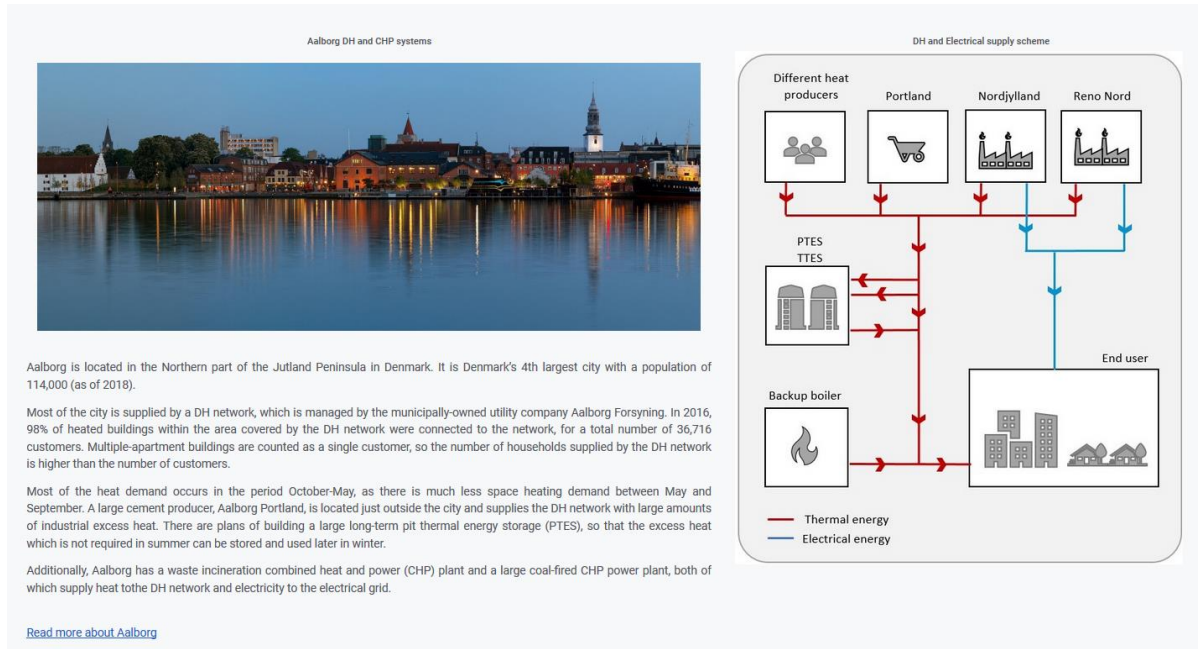


Figure 11: Overall dashboard of the Aalborg pilot – presentation of the local system



Figure 12: Overall dashboard of the Aalborg pilot – general figures

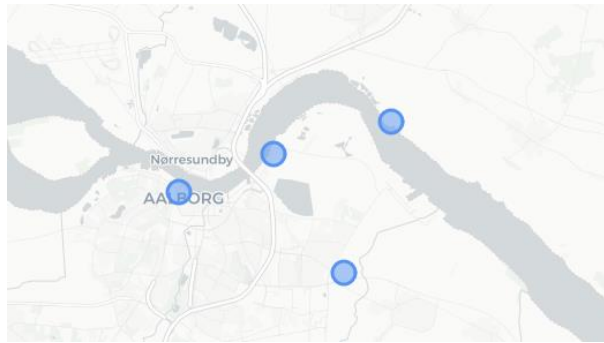


Figure 13: Overall dashboard of the Aalborg pilot – geographic distribution of the heat and electricity producers

- the Climate Conditions dashboards present the climate conditions of the respecting region based on data from the past five years, such as the minimum and maximum temperatures and wind speeds registered in the region, their daily averages along the time, the weekly total values of precipitation, the yearly averages and the distributions of cloud cover and radiation (Figure 14).



Figure 14: Part of the climate conditions dashboard of the Aalborg pilot

- the Energy Balance dashboards contain a variety of panels that allow to analyse in detail the demands, consumptions, performances and so on of the elements of the selected case-study.

For example, in the case of Aalborg, it is possible to visualize how heat production and demand vary together along the time, the hourly heat exchange of the TTES (Figure 15), the amount of heat and/or electricity provided for each supplier, the percentual

contribution of each of them and also the percentual contribution of each system concerning heat supply (Figure 16), among others.

For all the panels that do not expressly show a value, such as the representations of variables along time, the viewer can find more precise information by placing the mouse on the panel (Figure 17).

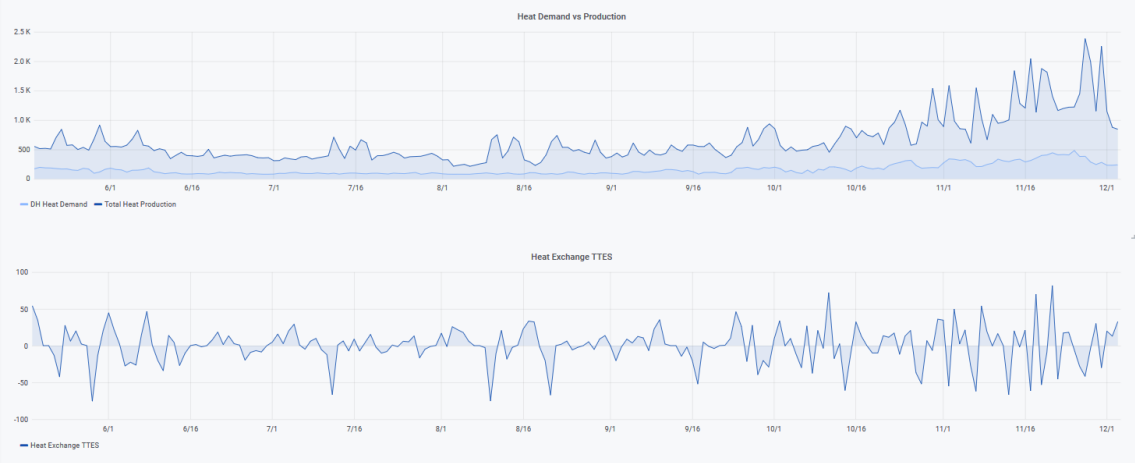


Figure 15: Energy Balance dashboard of the Aalborg pilot – heat demand vs production and heat exchange with TTES

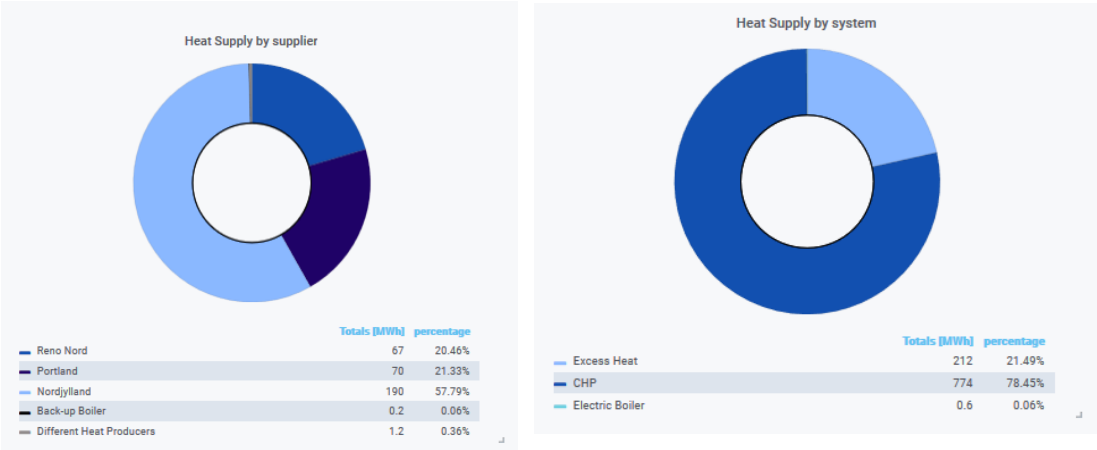


Figure 16: Energy Balance dashboard of the Aalborg pilot – pie charts

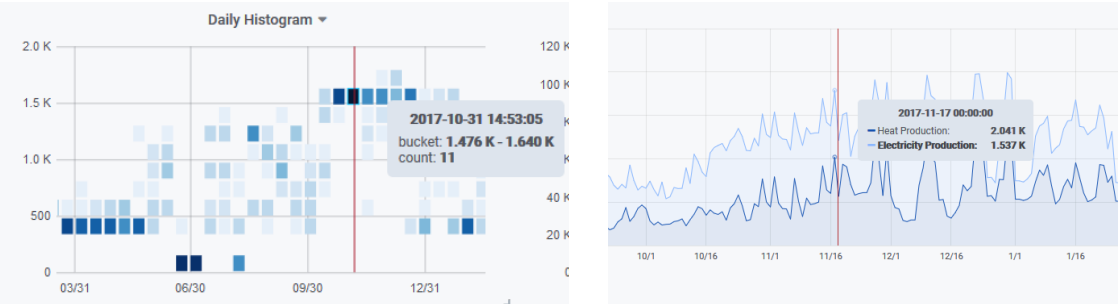


Figure 17: Energy Balance dashboard of the Aalborg pilot – detail for a determined point of time

Given the nature of the values, the panels in the Aalborg dashboards will remain static. The panels concerning to Ispaster, on the other hand, will be regularly and automatically updated, once the data source is constantly receiving new data.

3. Data forecasting system

This section explains the forecasting system built to CHESTER project and describes the forecasting methods used to obtain predictions of the electricity price in Spain and Denmark and of the heat demand of Aalborg's District Heating System. It is also explained how the temperature and relative humidity forecasts of several locations are attained.

3.1. Data Forecasting overview

Data forecasting is a process that consists on the analysis of past and present data to make future predictions. Usually part of the data is used to build the model while a smaller range of data is reserved to validate its results.

Prediction results and the models that generate them are highly dependent on historical data. In some cases, that data is in constant change thus the models must be regularly updated as well. The model for electricity price forecast is built on the basis of the hourly prices of the last year (8760 values). Every day at 8:00 pm the electricity prices for the next 24 hours are broadcasted by an electricity provider, Red Eléctrica de España. Through their API, this information is automatically fed to the code that is scheduled to re-run at that hour using the most recent 8760 values to generate the new model and calculate new predictions with 24 steps into the future (for the day after tomorrow). Both the data released by the provider and the forecast are then uploaded to OpenTSDB and read in Grafana, as explained in section 3.1.

On the other hand, the models to predict the heat demand of the DHS of Aalborg and Ispaster were developed with static data - hourly measured variables of a 1-year time span - and the same model can be applied over and over. To obtain the heat demand prediction for a certain period of time, it suffices to insert the input variables corresponding to that period. Once heat demand depends on the weather conditions, and so does the chosen predictive model, some input variables rely on weather forecasts. The heat demand prediction models are executed once a day to generate 24-steps-ahead predictions.

The weather forecast is provided by Accuweather [7]. Every hour Accuweather releases new 12-steps-ahead forecasts that are received through their API, uploaded to OpenTSDB and become visible in Grafana. The parameters received from Accuweather are the dry-bulb temperature and the relative humidity of the air.

The predictive models were built in Python with free libraries.

3.2. Forecasting methods

Several approaches can be used to obtain a prediction, depending on the characteristics of the available data and on the type of variable to predict. Two different methods were used in this project:

- S-ARIMA, to predict the electricity price in Spain and Denmark
- ANN, to predict the heat demand in Aalborg and Ispaster DHS

Broadly speaking, S-ARIMA is a time series statistics-based model that analyses the evolution of a variable over the course of time until the present moment and, based on that, predicts how that variable is going to behave afterwards. On the other side, an ANN is a black box model that learns the effect of a set of input parameters on an output variable so that for a different set of the same input parameters it can return the corresponding output.

3.2.1.1. S-ARIMA

A S-ARIMA model (Seasonal AutoRegressive Integrated Moving Average) falls within the Time Series forecast category. An S-ARIMA is composed by a stationary component and a component that expresses seasonal dependency. Each of these components is composed by an autoregressive term (AR), an integrating term (I) and a moving average term (MA). A model can be represented with the following configuration, in which p , d and q represent, respectively, the order of the terms aforementioned. The lower-case letters concern to the stationary component while the high-case letters refer to the seasonal component. S represents the period of the cycle.

$$\text{ARIMA}(p,d,q) (P,D,Q) S$$

A preliminary visualization of the time series should allow to have an educated guess on the order of the terms. A model is obtained once the order of each component is defined and the coefficients are estimated through an iterative calculus procedure.

There are Python libraries dedicated to the estimation of statistical models, such as statsmodels [8]. One of the features of statsmodels is a class that, for a given dataset, generates iteratively all the S-ARIMA models that can result from all the possible combination of components' orders within a specified range and returns the coefficients of the models and other information that enables to identify the most suitable model for the dataset, such as the AIC. The coefficients of the model are estimated by this class by applying the MLE (Maximum Likelihood Estimation) method.

Once a model is selected its suitability has to be assessed. Its residuals must have a random distribution, which means no trend shall be identified in their function. This is possible to verify through their histogram, the quantile-quantile graph and the autocorrelation and partial correlation functions of the residues, as exemplified in the image below (Figure 18).

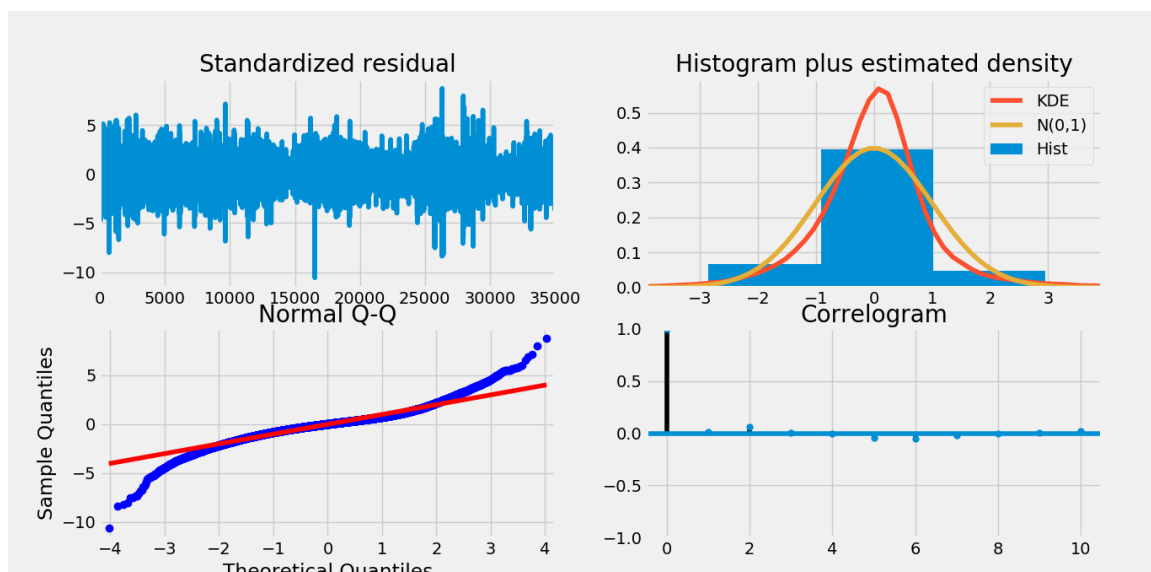


Figure 18: Residuals graph, histogram, q-q graph and correlogram.

As this analysis is somewhat subjective, it is possible to objectively compare models by implementing a criterion of information of the model. There are several, but the most commonly used are AIC (Akiake Information Criterion) and BIC (Bayesian Information Criterion). These were the criteria used to compare the models used in CHESTER project.

For last, the model can be validated through direct contrast of the predicted values with real values. When building predictive models, it is common to use part of the data to obtain the model and a little sample to calculate the error and validate the results. The errors calculated to verify the accuracy of the predictions were the MAE (mean absolute error), MAPE (mean absolute percentual error) and MSE (mean squared error), though more attention was given to the first two.

Figure 19 presents a 12-step-ahead prediction results contrasted with observed values of the electricity price in Spain. The grey area corresponds to the confidence interval.

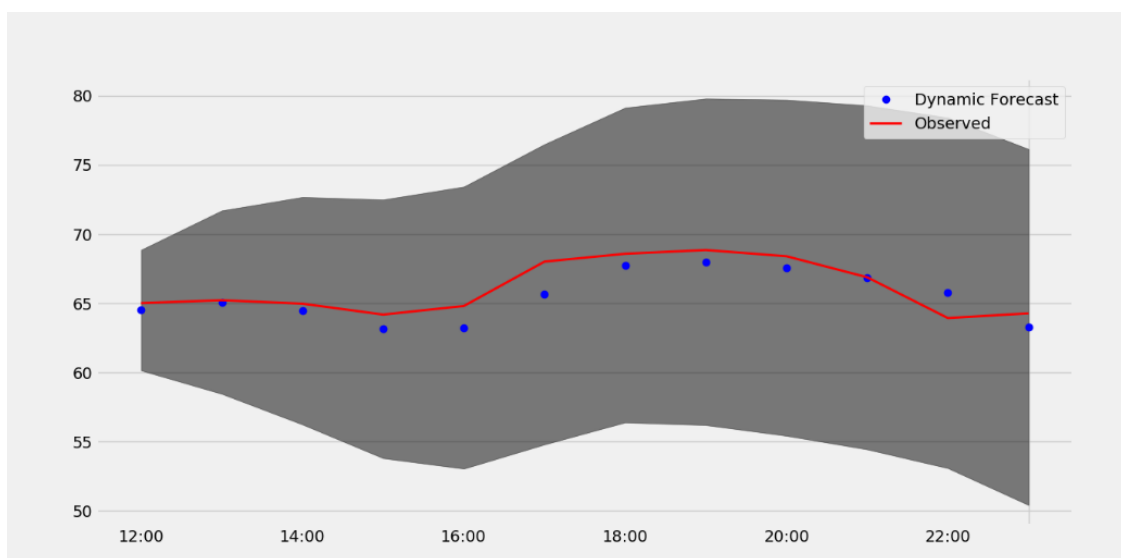


Figure 19: 12-step-ahead prediction of the electricity price in Spain.

3.2.1.2. Artificial Neural Network

ANN (Artificial Neural Networks) are computational systems, usually employed in machine learning, that learn through successive training how a set of input variables altogether impact an output variable. In a broad sense, what an ANN does is to detect patterns in data.

There are several different types of neural networks according with their complexity, information flow, learning processes... Nonetheless all of them are composed by an input layer, at least one hidden layer and an output layer. The number of neurons of the input and output layers corresponds, respectively, to the number of input and output parameters.

The ANN used to predict the heat demand of the DH of Aalborg is a multi-layer perceptron network. Each neuron of the input layer is connected to each neuron of the subsequent layer and so on until the output layer and all of them have associated weight and bias factors. The learning technique employed in this model is called backpropagation. It consists on the adjustment of the weights and biases toward the reduction of the loss function result that is obtained with the prediction result and the real values. That adjustment is accomplished through the derivatives of the loss function with respect to the weights and biases.

Finding a good model is an iterative process. Different amount and types of inputs, number of layers, number of neurons in each layer, activation functions, types of loss function, optimizers and other features have to be tested. In order to find the input variables of the model it is necessary to perceive what physically affects the variable to predict. It is rather straightforward that the lower the temperatures, the higher the heat demanded from a DHS will be, however there are other factors that can strongly dictate the heat demand and are not so easily identified.

The input variables used to build this model were the outdoor air temperature, the relative humidity, the day of the year, the hour of the day, the amount of heat demanded the previous day at the same hour, and the type of week day (defined as follows: 0 – labour day, 1 – Friday or day before a holiday, 2 – weekends and holidays) (Figure 20).

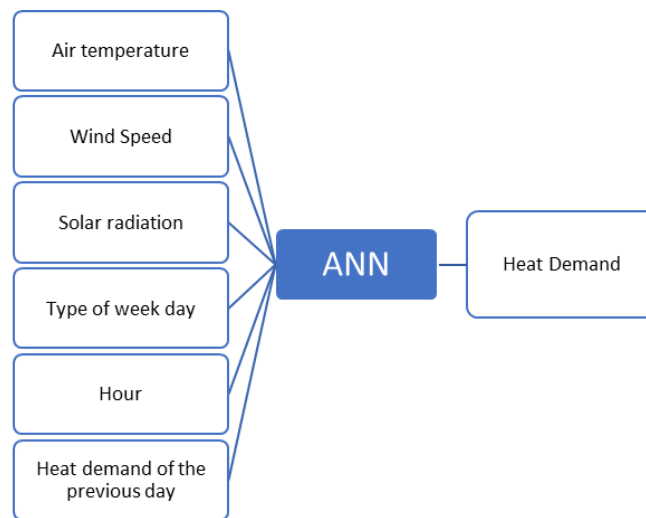


Figure 20: Representation of the ANN for heat demand prediction.

The model to predict the heat demand in Aalborg's DH system was built resorting to Keras [6], a deep learning python library. The better model found to predict the heat demand in Aalborg's DH system is defined as follows.

- Number of hidden layers: 4
- Number of neurons on each hidden layer: 450
- Loss function: Mean Squared Error
- Activation Function: Sigmoid
- Optimizer: Adam - an "algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments." [7]
- Weights initializer: Truncated Normal - random values with a normal distribution, discarding values more than two standard deviations from the mean.
- Bias initializer: 0.1
- Batch size: 24

The model is trained with part of the data and tested with a smaller sample. Its accuracy is assessed through the calculation of the MAE, the MAPE and the R-square. Figure 21 shows the result of a 24-step-ahead prediction.

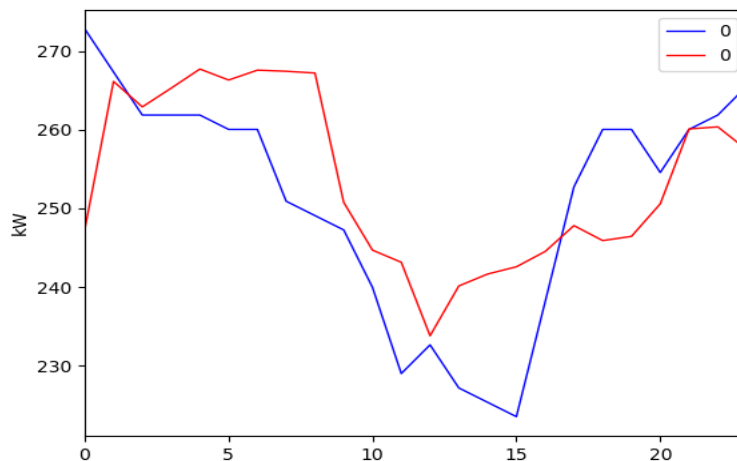


Figure 21: 24-step-ahead prediction of the heat demand in Aalborg.

4. Conclusions

This deliverable describes the work done in the scope of the T4.1 concerning the monitoring and control structure developed for the CHESTER project. After the selection of the two case studies of Aalborg and Ispaster considering their potential for deploying a CHESTER system and the availability of data, both real time and historical, the monitoring and control system has been designed and partially developed. Further work will be done to ensure that the data collection and evaluation works as expected. Moreover, the developed monitoring platform will serve as basis for the SEMS interface to be developed in subsequent tasks of WP4.

In Ispaster case study, real time data of the DH network and the microgrid will be collected after a new monitoring and communications network is deployed in the municipality, while in Aalborg use case, the monitoring platform will be feed from DH historical data of past years due to security and confidentiality issues. Both case studies will use the same monitoring platform that is developed leveraging a free and scalable data base (OpenTSDB) and an open source platform for data visualization (Grafana). Besides, for control purposes, the electricity price of Spain and Denmark and the heat demand of Aalborg's District Heating System will be forecasted using neuronal network.

In this deliverable it is furnished the necessary information to offer an insight on the functionalities and operation of the overall structure and, in particular, to provide the basic knowledge necessary to manipulate the data exhibition in the data presentation platform.

The monitoring and control system will consist on a practical tool in the optimization of the CHESTER system allowing to save substantial amounts of time and effort in what concerns to data collection and analysis.

References

- [1] «OpenTSDB,» [En línea]. Available: <http://opentsdb.net/>.
- [2] “Grafana,” [Online]. Available: <https://grafana.com/plugins/grafana-worldmap-panel>. [Accessed April 2019].
- [3] «Python,» [En línea]. Available: <https://www.python.org/>.
- [4] «Docker,» [En línea]. Available: <https://www.docker.com/>.
- [5] I. InfluxData, “InfluxDB - Time series data storage.,” [Online]. Available: <https://www.influxdata.com/time-series-database/>. [Accessed March 2019].
- [6] «Keras: The Python Deep Learning library,» [En línea]. Available: <https://keras.io/>.
- [7] J. B. Diederik Kingma, “Cornell University,” Cornell University, 22 12 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980v8>. [Accessed 28 March 2019].
- [8] “Grafana Labs,” [Online]. Available: <http://docs.grafana.org/>. [Accessed March 2019].
- [9] «Accuweather,» [En línea]. Available: <https://www.accuweather.com>.
- [10] «StatsModels' Documentation,» [En línea]. Available: <https://www.statsmodels.org/stable/index.html>.

Appendix 1 – Registered variables in Ispaster

Table 4 presents all the variables registered in Ispaster and available in the presentation data system, along with the metric and tags that identifies them.

The variables and have associated three more tags: lat, long (respectively, the latitude and longitude of the corresponding installation) and location (“ispaster” for all of them). The elements in the table are featured as they are in the cloud infrastructure: one single word, all lower cases.

Table 4: Variables registered in Ispaster – metrics and tags

Metric's designation	Tags (Value/Key)		Description
	System	Installation	
energy.out.monthly	pv	battery	Monthly energy supplied by the batteries
energy.out.total			Total energy supplied by the batteries
energy.in.monthly			Monthly energy received by the batteries
energy.in.total			Total energy received by the batteries
energy.monthly		pv	Monthly energy generated by the PV system
energy.total			Total energy generated by the PV system
energy.load.monthly		inverter	Monthly load of the PV system
energy.load.total			Total load of the PV system
voltage		battery	Status of charge of the batteries
temperature			Temperature of the batteries
temperature.sup	sts	solarcollector	Temperature of supply of the STS
temperature			Temperature of the collector of the STS
efficiency			Efficiency of the collector
energy.daily			Daily energy generated by the STS
energy.total	boilerroom	boiler	Total energy generated by the STS
temperature.sup			Supply temperature of the boiler
temperature.ret	hx	boilerroom	Return temperature to the boiler
energy.heating			Total heat demand (measured at the boiler room)
energy.heating			Heat consumption measured at the city hall
energy.heating			Heat consumption measured at the bar
energy.heating			Heat consumption measured at the showers
energy.heating			Heat consumption measured at the restaurant
energy.electrical	smartmeter	boilerroom	Total electricity demand (measured at the boiler room)
energy.electrical		cityhall	Electricity consumption measured at the city hall
energy.electrical		herrikotaberno	Electricity consumption measured at the bar
energy.electrical		frontonduchas	Electricity consumption measured at the showers
energy.electrical		areitzbi	Electricity consumption measured at the restaurant

Appendix 2 – Registered variables in Aalborg

Table 5 presents all the variables registered in Aalborg and available in the presentation data system, along with the metric and tags that identifies them.

All these variables are presented in MWh and have associated three more tags: lat, long (respectively, the latitude and longitude of the corresponding installation) and location ("aalborg" for all of them). The elements in the table are featured as they are in the cloud infrastructure: one single word, all lower cases.

Table 5: Variables registered in Aalborg – metrics and tags

Metric's designation	Tags (Value/Key)		Description
	System	Installation	
fuel	chp	renonordl4	Fuel used to generate heat and electricity to area L4 of the Reno-Nord CHP plant
electricityproduction			Electricity produced at area L4 of the Reno-Nord CHP plant
heatproduction			Heat produced at area L4 of the Reno-Nord CHP plant
fuel	chp	renonordl3	Fuel used to generate heat and electricity to area L3 of the Reno-Nord CHP plant
electricityproduction			Electricity produced by area L3 of the Reno-Nord CHP plant
heatproduction			Heat produced by area L3 of the Reno-Nord CHP plant
electricityconsumption	chp	renonordbypass	Electricity consumed by the bypass of the Reno-Nord CHP plant
heatproduction		renonordbypass	Heat produced by the bypass of the Reno-Nord CHP plant
heatproduction	excessheat	aalborgportlandvg1	Excess heat produced by Portland's factory – group 1
		aalborgportlandvg2	Excess heat produced by Portland's factory – group 2
		aalborgportlandadditional	Additional excess heat produced in Portland's factory
		differentheatproducers	Excess heat produced by different small producers
fuel	chp	nordjyllandsvaerketcoalfiredplantminlastcondensation	Fuel used to generate electricity through the condensation turbine at the coal fired CHP plant of Nordjylland, minimum load
electricityproduction			Electricity produced by the condensation turbine at the coal fired CHP plant of Nordjylland, minimum load
electricityconsumption	chp	nordjyllandsvaerketcoalfiredplantminlastcounterpressure	Electricity consumed in the operation of the load counter-pressure turbine at the coal fired CHP plant of Nordjylland, minimum load

Metric's designation	Tags (Value/Key)		Description
	System	Installation	
heatproduction			Heat produced to the operation of the counter-pressure turbine at the coal fired CHP plant of Nordjylland, minimum load
fuel			Fuel used by the Benson steam generator - backpressure turbine, at the coal fired CHP plant of Nordjylland,
electricityproduction	chp	nordjyllandsvaerketcoalfiredplantbensontbackpressure	Electricity produced due to the benson steam generator - backpressure turbine, at the coal fired CHP plant of Nordjylland
heatproduction			Heat produced by the benson steam generator- backpressure turbine at the coal fired CHP plant of Nordjylland
fuel	chp	nordjyllandsvaerketcoalfiredplantfullloadbackpressure	Fuel used to generate electricity through the back pressure turbine at the coal fired CHP plant of Nordjylland, at full load
electricityproduction			Electricity produced by the back pressure turbine at the coal fired CHP plant of Nordjylland, at full load

Appendix 3 – Grafana's user guide

First of all, in order to create a dashboard, it is necessary to add a data source to the platform in the general configuration menu. The dashboard itself is created in the configuration menu as well.

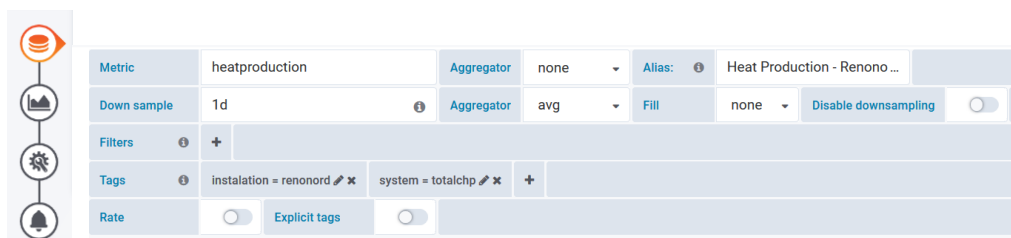
Once the dashboard is created, it is possible to design a panel. The panel is the basic visualization building block in Grafana. Each panel provides a query editor, dependent on the data source selected in the panel, that enables the manipulation of the data according to the capabilities of the data source.

Grafana has seven predefined types of panels: graph, singlestat, table, heatmap, alert list, dashboard list and text. Nevertheless, it is possible to extend the variety of panels by downloading panel plug-ins. The type of panel is defined in the visualization menu. In general terms, this menu allows to design the appearance of the panel, define axes, limits, labels, units, markers, numeric representation, and other. The options available depend on the type of panel selected [8].

The following subsections explain how to create and edit the panels that were used the most for this project.

A3.1. Queries configuration

Figure 22 presents the configuration menu of a query. The method of definition of queries is identical regardless the type of the panel.



Metric	heatproduction	Aggregator	none	Alias:	Heat Production - Renono ...
Down sample	1d	Aggregator	avg	Fill	none
Filters	+				
Tags	installation = renonord ✕ system = totalchp ✕ +				
Rate	<input type="checkbox"/>	Explicit tags	<input type="checkbox"/>		
Disable downsampling <input type="checkbox"/>					

Figure 22: Configuration menu of a query in Grafana.

The timeseries have to be configured in Grafana according with the OpenTSDB specifications already mentioned in subsection 2.1.2.

The metric used in the example is *heatproduction*. This designation stores the measurement of different heat productions, according with the system and the installation (the keys) that generates the heat. The tags specify a particular time series by attributing values to the tag-keys.

As the first aggregator is defined as *none*, if no tags were defined, all the timeseries under the metric *heatproduction* would be shown separately in the graph. If, for example, only the pair *installation = renonord* was defined and the metric aggregator was set as *sum*, the graph would display the sum of the values of the heat produced by all the systems in Reno Nord at each point of time.

As aforementioned, the down sample defines the resolution of the data. According to Figure 22, it is set as 1d (one day), with the avr (average) aggregator. Thus, the values presented in the graph (Figure 23) are the daily averages of the heat production in the CHP system in the Reno Nord installation.

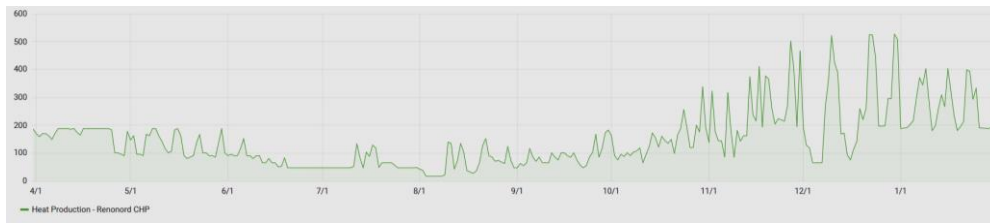


Figure 23: Graph of a time series generated in Grafana.

It is mandatory to follow the OpenTSDB syntax. The down sample have to be defined with a number followed by a letter that represents a timestamp. The sample options available are show in Table 6.

Table 6: Accepted syntax to define timestamps.

Syntax	Timestamp
s	Seconds
m	Minutes
d	Days
w	Weeks
y	Years

A3.2. Graph

The graph is the main panel in the platform since it is the most convenient graphical design to display time series. It is possible to represent more than one query in a Graph.

The time series represented in a graph can be defined as lines, bars or points. The visualization menu allows to stack values, to establish how to deal with null values and to control axes, legends and labels as presented in Figure 24.

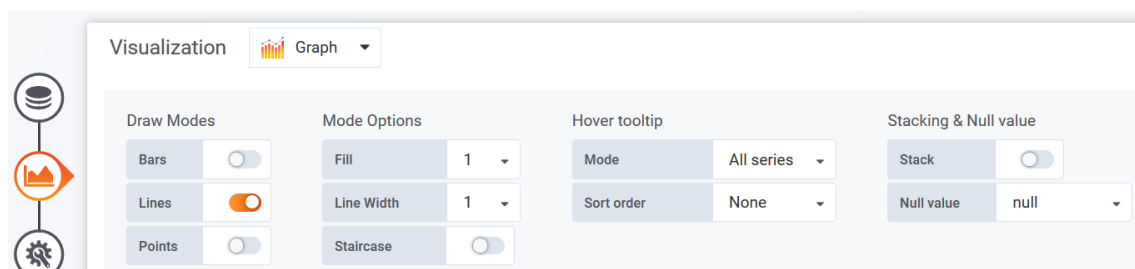


Figure 24: Visualization menu of a graph panel.

The Y-axes can be hidden, scaled and labelled and have an associated unit. By default, the X-axis represents time, but it can also be set to group the information by series or to display the values as a histogram.

In this panel it is also possible to highlight thresholds or time regions, as exemplified in Figure 25, in which the weekends are represented by the grey bars.

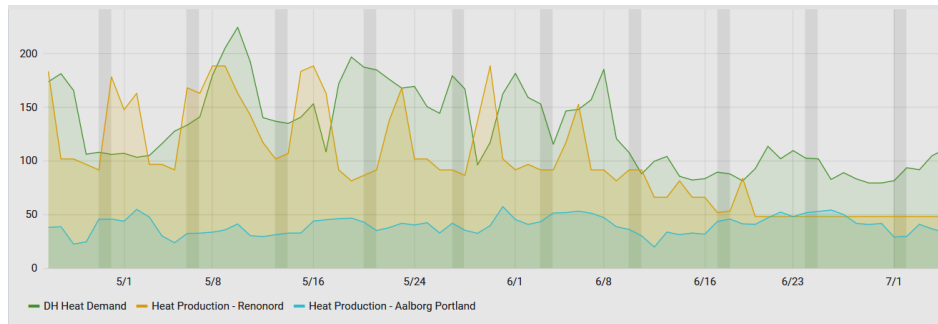


Figure 25: Time series graph with highlighted time regions.

At last, the graph is the only panel that allows the creation of alerts. The alerts can be defined to periodically evaluate a set of conditions and send a notification. All the defining parameters are set by the user in the alert configuration menu. The recipient of the notification is selected from a dropdown list in the same menu, but the addresses in that list have to be set in the general settings menu of Grafana.

Besides the customized alerts, Grafana graphs provides “No Data & Error Handling” options.

A3.3. Singlestat and gauge

These two panels are similar: their purpose is to exhibit a main summary stat of a time series and both of them allow to specify a colour scheme according with thresholds and value mappings. Figure 26 shows the list of summary stats Grafana allows.

- **min** - The smallest value in the series
- **max** - The largest value in the series
- **avg** - The average of all the non-null values in the series
- **current** - The last value in the series. If the series ends on null the previous value will be used.
- **total** - The sum of all the non-null values in the series
- **first** - The first value in the series
- **delta** - The total incremental increase (of a counter) in the series. An attempt is made to account for counter resets, but this will only be accurate for single instance metrics. Used to show total counter increase in time series.
- **diff** - The difference between 'current' (last value) and 'first'.
- **range** - The difference between 'min' and 'max'. Useful to show the range of change for a gauge.

Figure 26: Summary stats available in Grafana/OpenTSDB.

Notwithstanding, while the singlestat is one of the default panels of Grafana, gauge has to be added as a plug-in. Moreover, the first one allows to visualize the summary stat against the evolution of the series on the background (Figure 27 a)), and the latter is more appropriate to show the place a variable takes within a context (Figure 27 b)).

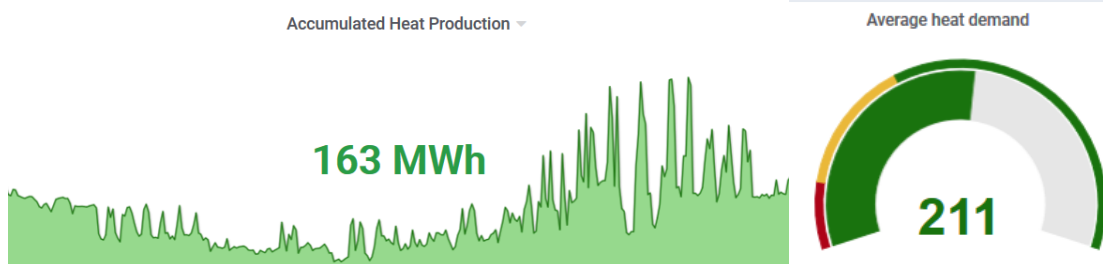


Figure 27: Summary stats of time series in Grafana presented as a) singlestat b) gauge.

To set the thresholds colour representation it is merely necessary to define the threshold and the colours that represent each range (Figure 28). In the singlestat the colour can be attributed to the value, the background of the panel (Figure 29), or to a prefix or a postfix. For the gauge panel, the colour is attributed to the value and the gauge itself. In green it can be seen the accumulated heat production in Aalborg, in red the heat provided by the PTES, in orange the heat storage and also in green the heat losses.



Figure 28: Singlestats of the data presentation system of Aalborg's CHESTER.

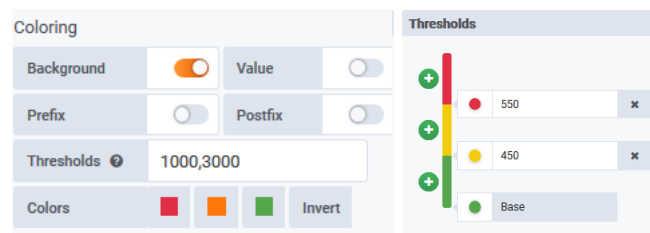


Figure 29: Thresholds coloring configuration for a) singlestat b) gauge.

At last, Value Mapping expresses the meaning of a determined value or range of values. If the value mapping is active and the value of the gauge or the singlestat falls within the specified range, the panel will display the text defined instead of the value.

A3.4. World map panel

The maps shown beforehand were created with the Worldmap Panel plug-in. To configure the map, it is necessary to provide a latitude and longitude and a query with a link to a location. If data of the query is a time series, the metric name and a key from a list of locations must match. The list of locations can be supplied through json files or endpoints that include or return a list of locations and the corresponding coordinates [2]. The plug-in includes the location files of countries with 2 and 3 letter codes and the states of the USA. Each query must return a single value.

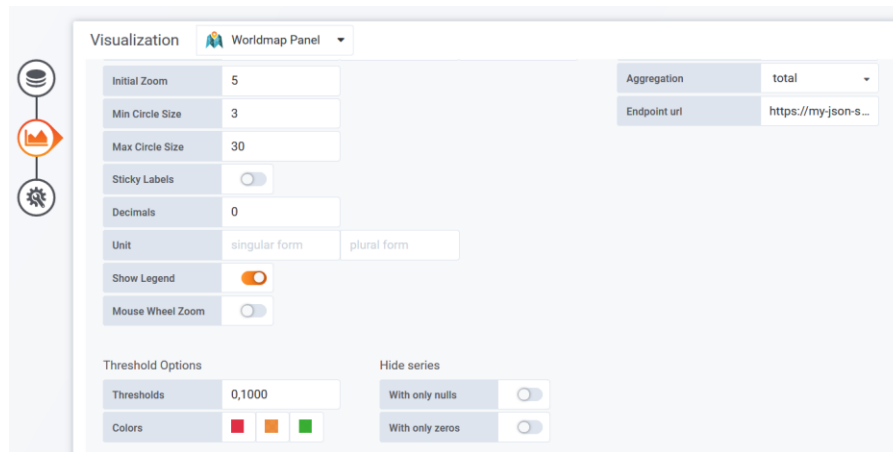


Figure 30: Visualization menu of the Worldmap panel.

The lateral menu allows to navigate between the options necessary to define a panel (left side of Figure 30). The first icon leads to the configuration of the queries, the second to the visualization options, the gear represents the page to set the general options of the panel and the bell take the user to the alerts definitions (only available for panels of the type Graph).

A3.4. Heatmap

Heatmap is the name of the panel used to create the histograms-over-time shown in subsection 2.1.4. This panel arranges the values of the time series in buckets and uses a colour scheme to disclose the number of data points that fall within that bucket, displaying as much histograms as timesteps inside the panel.

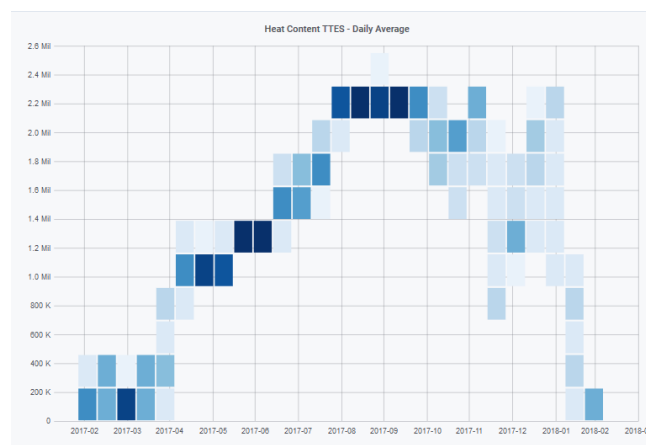


Figure 31: Heatmap panel.

The number and size of timesteps (X axis) and the size and number of the buckets (Y axis) can be customized in the visualization menu. For example, the settings of the menu presented by Figure 32 define that the values of the heatmap will be distributed in 10 buckets and that timestep is 1 week (1w), which means that the panel depicts one histogram per week.

Moreover, the information can be displayed in the format of a regular time series (such as presented in Figure 31) or in the format of a bucket time series in which each series returned from the data source represents a bucket.

Besides the usual edition options, the scale can be set as linear or logarithmic with base 2, 10, 32 or 1024.

Y Axis

Unit	short
Scale	linear
Y-Min	auto
Y-Max	auto
Decimals	0

Buckets

Y Axis	Buckets	10	Size	auto
X Axis	Buckets	auto	Size	1w

Data format

Format	Time series
--------	-------------

Figure 32: Visualization menu of the heatmap panel.