An Advanced Visualisation Engine with Role-Based Access Control for Building Energy Visual Analytics

Georgios Kormpakis
Decision Support Systems Laboratory, School of Electrical and Computer Engineering National Technical University of Athens
Athens, Greece
gkorbakis@epu.ntua.gr

Panagiotis Kapsalis
Decision Support Systems Laboratory, School of Electrical and Computer Engineering National Technical University of Athens
Athens, Greece
pkapsalis@epu.ntua.gr

Konstantinos Alexakis
Decision Support Systems Laboratory, School of Electrical and Computer Engineering National Technical University of Athens
Athens, Greece
kalexakis@epu.ntua.gr

Sotiris Pelekis
Decision Support Systems Laboratory, School of Electrical and Computer Engineering National Technical University of Athens
Athens, Greece
spelekis@epu.ntua.gr

Evangelos Karakolis
Decision Support Systems Laboratory, School of Electrical and Computer Engineering National Technical University of Athens
Athens, Greece
vkarakolis@epu.ntua.gr

Haris Doukas
Decision Support Systems Laboratory, School of Electrical and Computer Engineering National Technical University of Athens
Athens, Greece
h_doukas@epu.ntua.gr

***Abstract* — One of the main challenges of today's societies is the avoidance of the climate change since the climate crisis in now more evident than ever. Buildings have a large share of total energy consumption and, thus, it is obvious that actions should be taken to reduce their needs. Taking into consideration that nowadays data related to building's metrics are available in significantly higher rate than in the past, due to the advance of the related technologies, it is necessary to find ways to exploit them in order to draw useful inferences regarding their consumptions and how they can be reduced. For that reason, in this paper we present a Visualisation Engine, which offers a variety of visualisations over stored data. With the usage of the proposed Visualisation Engine, we envision to be able to conduct sufficient research over the data, to generate insightful information regarding their behaviour, and to assist the development of useful solutions towards the direction of more energy efficient buildings.**

Keywords* — *building data, energy efficiency, visualisations, visual analytics

I. INTRODUCTION

A. *Background*

Nowadays, the technologies related to the Internet of Things devices, such as smart meters and various sensors, are constantly advancing and being enriched, resulting in massive amounts of produced data [1]. At the same time, current climate change threats undoubtedly represent a domain where action is required [2] and the European Union (EU) is called to provide efficient solutions for this serious problem since Energy Efficiency has become an undoubted necessity worldwide. As almost 40% of EU energy consumption is produced from buildings [3], the building sector must adapt to the current situation and contribute to the identification and implementation of effective measures that will lead to a decarbonized future and to the achievement of the objectives set out in the Paris Agreement [4], since reducing energy consumption is one of the current major global challenges [5]. In order to increase energy efficiency, we must standardise the data collection, data processing and visualisation techniques [6]. As a result, suitable ground for research activity is currently offered.

As the digitalisation era is more apparent than ever, data availability and quality have significantly improved [7]. However, more challenges also emerge[8], since the data are now generated in an unprecedented rate assisted by cutting-edge technology. Recently, great attention has been paid to the challenges for great data, largely due to the wide spread of applications and systems used in real life, such as presentation, modelling processing and large (often unlimited) data storage.

Data management has grown in importance, since extracting the significant value out of a huge pile of raw data is of prime importance, in order to make different decisions[9].

All these massive amounts of data have to be exploited [10], leading to valuable conclusions, detailed insights and decision support for the adoption of more efficient measures and policies towards maximum energy efficiency, minimum operating costs and, thus, reduced environmental impact [11].

The overarching driving vision of visual analytics is to turn the information overload into opportunities [12]. To exploit data towards this vision, there is a necessity of being able to represent them in a well-defined, easily readable, and adjustable way. In this direction, there is a variety of tools that assist the visualisation of data while various analytical visual tools are increasingly being developed for scientific communities [13] serving data analysis tasks.

B. State-of-the-art

One of the most widely used is the Apache Superset[1]. It is a modern, enterprise-ready business intelligence (BI) web application which offers a simple and feature-rich interface that allows the users to create and share dashboards. It also allows users to focus on each graph/metric as well as to filter and organise the needed data.

Tableau [2] is also widely used. Tableau is a free data visualisation software, which supports various concepts and commands to create charts, assemble them into dashboards and extract insights and patterns existing in the available data [14].

Another commonly used approach for data visualisations is the Matplotlib[3] library, which is a comprehensive library for creating static, animated and interactive visualisations in Python. It offers the capability of creating various plots, interactive figures that can zoom, pan, and update, customisable layouts, the capability of exporting in various file formats and it can be embedded in JupyterLab and Graphical User Interfaces. It runs on all major operating systems, with binaries for Macintosh's OS X, Microsoft Windows, and the major Linux distributions [15].

All the above-mentioned tools are widely user for a variety of visualisations, including building data visualisations. Some further tools that are commonly used are Qlik[4], Datawrapper[5] and Plotly[6].

C. Contribution

In the present paper we propose a Visualisation Engine, implemented in the context of the MATRYCS[7] H2020 project, which visualises data in a variety of ways aiming to assist the extraction of useful conclusions and insights. These data derive from the project's large-scale pilots and include a wide variety of data that contain information about building energy, heating consumption, CO2 emissions, cooling demands, and Energy Performance Certificate (EPC) information.

Our proposed Visualisation Engine is, in essence, a dashboard that communicates with the database of the stored data and, via a user-friendly and adjustable interface, provides the capability of visualising the requested data in a variety of commonly used charts (pie charts, bar charts, gauge charts, line charts, radar charts), as well as map and plain table visualisations. Since the building data of the project are sensitive, a security mechanism was developed in the context of the project. Specifically, due to the fact that a number of pilots were involved in the project, it was necessary to assure that each potential user would only be able to only access the data of their organisation and, of course, that no unauthorised users would be able to access any data. In this respect, the aforementioned security mechanism was designed properly in order to achieve role-based access control which was, in our case, the needed approach for efficiently securing the data.

Thus, apart from a comprehensive way of visualising the available data, which includes, as mentioned, a variety of charts as well as maps and tables, the proposed Visualisation Engine also offers a significant merit which is no other than the role-based access management. In this way, motley datasets are efficiently secured, and a variety of capabilities are offered, since any connected database can simultaneously have various datasets that, through proper handling, can be restricted to non-authorised users.

D. Structure of the Document

The rest of the paper is organised as follows: in Section II we analyse the architecture of the Visualisation Engine and the technologies and libraries that have been leveraged during its development. In Section III we present the outcomes of this Visualisation Engine, and we thoroughly describe the different pages that it includes, among with some representative screenshots that demonstrate the work conducted, as well as the security mechanism with which it is integrated. In Section IV we demonstrate a use case scenario of the Visualisation Engine and, finally, in Section V we describe the conclusions that derive from this implementation, and we discuss regarding some possible extensions and future work that can be implemented.

II. ARCHITECTURE

The visualization of the available building data is of utter importance towards the effort to maximize energy efficiency via methodical and systematic monitoring and reporting. Building data are either timeseries data or metadata that describe their characteristics. There is a number of different architectures that describe the management of these data with either property databases [16] or hybrid ways that can manage timeseries data [17]. The proposed architecture is focused on visual analytics for building data. The high-level view of this architecture is presented in Fig. 1.

A. Libraries and Tools Used

The application was developed using powerful and modern tools and libraries in order to ensure a quality and robust implementation.

The Visualization Engine comprises a dashboard, which is a Single Page Application (SPA) developed using the React 8 framework. SPAs are a state-of-the-art approach to web development and offer significant advantages compared to Multi-Page Applications. Specifically, SPAs are faster than Multi-Page Applications since most of the resources (HTML9 / CSS10 / Scripts) are only loaded once throughout the lifespan of an application and from that point on they do not update the entire page but only the required content. Also, SPAs offer more

[1] https://superset.apache.org/
[2] https://www.tableau.com/
[3] https://matplotlib.org/
[4] https://www.qlik.com/us/
[5] https://www.datawrapper.de/
[6] https://plotly.com/
[7] https://matrycs.eu/
[8] https://reactjs.org/
[9] https://en.wikipedia.org/wiki/HTML
[10] https://www.w3.org/Style/CSS/Overview.en.html

efficient caching, easier debugging and, in general, a better user experience.

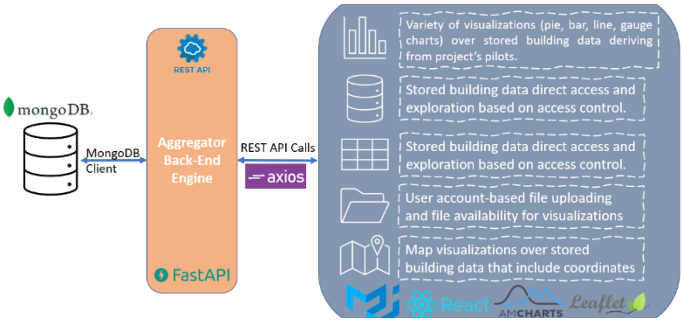

Fig. 1: High-level architecture of the proposed Visualisation Engine.

The dashboard has been designed and implemented according to latest instructions for an optimal user experience, meaning that it is simple, legible, and fully responsive, with an interface that adjusts perfectly in both desktop and mobile devices

The CSS framework used for the creation of the dashboard is the Material-UI library[11], one of most popular libraries used with React. It has been created by Google[12] and can be used as a single resource for all the styling needs, since it provides components, styles, themes, and icons.

The visualizations are developed using amCharts[13], which is a library for data visualization that includes a variety of basic and advanced charts, as well as various extendable plugins and other functionalities.

B. *Functionalities Implemented and Execution Steps*

When using the dashboard, the user can choose between a variety of visualizations, which are displayed in a sidebar menu that appears on the left side of the screen. The users can also visualize the needed data into tables, by filling a meticulously designed HTML form. The form supports common SQL[14] commands (ORDER BY, LIMIT, DESC/ASC).

Other than forms, the user can also directly execute SQL queries to the database by writing plain SQL queries in a textbox that the platform offers. After submitting a valid query, the requested data are fetched and are displayed in fully responsive tables, offering a clean and legible view of the data.

The procedure executed upon the submission of a request is as follows. After submitting a valid form, the front-end application connects to the backend and, using the Axios[15] library, it sends a REST API call which includes the form's data as payload, in JSON[16] format. The backend retrieves the requested data from the database and proceeds with properly formatting them. Following, the back-end sends the properly formatted data to the front-end application, again in JSON format. Finally, the requested data are available to the front-end application, where they are processed and injected into the -requested by the user- visualization chart.

The mechanism responsible for sending the data to the Visualisation Engine is the Data Feed Module. It is the cloud pipeline responsible for receiving data from Building information systems, conducts aggregations and, via its collections of REST APIs, distributes the information to the application layer. The component enables Pandas[17] SQL[18], and Presto[19] for combining information from different sources.

project_no	project_activity	type_of_activity	type_of_building	co2_reduction	energy_reduction
xxxxx	Expanded clay wall insulation	renovation of the building's enclosing structures	preschool educational institution	23.193	87.854
xxxxx	Silicate brick wall insulation	renovation of the building's enclosing structures	preschool educational institution	15.712	59.514
xxxxx	Exterior door replacement of glazed doors	renovation of the building's enclosing structures	preschool educational institution	2.245	8.502
xxxxx	Wooden window replacement with double-glazed windows	renovation of the building's enclosing structures	preschool educational institution	5.985	22.672

Fig. 2: LSP data table representation.

The data deriving from each pilot contain dissimilar information, thus fields, which can lead to different results and outcomes. For clarity reasons, Fig. 2 depicts the content from one of the project's large-scale pilots' tables. As shown, this LSP's table contains information regarding the project's number, the project's stored activity, the type of the activity accomplished, the type of the building in which the refurbishment, the CO_2 emission and energy consumed reduction that was achieved, the CO_2 emission factor and the CO_2 emission reduction and energy consumed scaled values, as well as the CO_2 emission factor scaled values. Below, the plain SQL command sent to the back-end will be displayed, as well as the returned data, with a limit of eight rows. The SQL command sent is the following:

The data related to the projects' numbers have been hidden for confidentiality reasons.

C. *Security Considerations and Integration*

```
SELECT * FROM lsp_name_table LIMIT 4
```

An aspect of the Visualisation Engine that had to be considered is the security of the application. It is of utter importance to ensure that the data, that are sensitive and private, would be secured and that non-authorized users could not have access to them. In order to fulfil this requirement, another component, named Security Enabler, was implemented. The purpose was to enable role-based access control, with the vision to totally exclude non-authorised users from the entirety of the

[11] https://mui.com/
[12] https://www.google.com/
[13] https://www.amcharts.com/docs/v4/
[14] https://en.wikipedia.org/wiki/SQL
[15] https://axios-http.com/docs/intro
[16] https://www.json.org/json-en.html
[17] https://pandas.pydata.org/
[18] https://en.wikipedia.org/wiki/SQL
[19] https://prestodb.io/

Visualisation Engine's content and to restrict specific users' access to specific datasets, based on the project's pilot they are involved in. For this reason, we initially leveraged KeyCloak's[20] functionalities. KeyCloak is an open-source management system. It provides user federation, strong authentication, user management, fine-grained authorisation and several further functionalities which facilitate the work of the developers by providing well-implemented authentication ant authorisation mechanisms. Its functionality is based on OAuth221 & UMA 2 security protocols [18]. OAuth2 is a modern authorization framework that enables applications to obtain limited access on an HTTP service22. User-Managed Access (UMA 2 or UMA 2.0) is a standard that aims to strengthen data privacy based on the well-known privacy by design principles 23. Technically, UMA 2 is a party-to-party authorization protocol based on the OAuth2 authorization framework.

On top of KeyCloak, we developed a new component, named Security Enabler. Security Enabler builds up on KeyCloak's functionalities and exposes a new REST API which provides all the above-described functionalities and some further extensions, hence ensuring that users cannot access KeyCloak, which is the basic user management tool. By using this component, we managed to create various roles for the Visualisation Engine with the vision to apply role-based access control and grant access to specific datasets only to the users that should have access to them.

Specifically, three processes, which are not serial, are required for the user to be authenticated and granted with access. The first process is the Sign In process, which is realised with a REST API call that includes the username and the password, as shown in Table I.

TABLE I: JSON INPUT FOR USER SIGN-IN AND ACCESS TOKEN ACQUISITION

```
POST /user/get/token
{
  "username": "userUsername",
  "password": "userPassword"
}
```

If the credentials are correct, the backend responds with a JSON object that includes the access token and some additional fields.

The front-end application stores the access token and the username in order to adjust its content based on them. By using one of the available REST API calls, as shown in Table II, in which the username is required, the roles of this specific user can be retrieved. The request sent is shown in Table II and the response of the REST API is shown in Table III. These roles are also stored in the front-end application and the available datasets are filtered in order to only display to the users the datasets of their organisation.

TABLE II: GET REQUEST TO ACQUIRE USER'S ROLES

```
GET /user/userUsername/client/clientName/roles
```

TABLE III: REST API RESPONSE AFTER USER'S ROLES REQUEST

```
{
    "message": "User roles for client VisualizationEngine:",
    "roles": [
        "testRole1 ",
        "testRole2",
    ]
}
```

The third process, which takes place on each page's load, is the introspection of the stored access token. A REST API call has been implemented, in which the token is sent to the back-end for introspection. The payload sent to the REST API is displayed in Table IV. The introspection's reply contains information about the validity of the provided access token as well as the related user's information. In this way, since this procedure takes place before each page's rendering, it is ensured that no unauthorised or signed-out user can access the specific page. The same happens for a user with an expired access token. In the above-mentioned cases, the users are automatically navigated to the Sign In page, in order to provide again their credentials and acquire a new access token.

TABLE IV: JSON INPUT FOR THE INTROSPECTION OF THE ACCESS TOKEN

```
POST /token/introspection
{"access_token": "acquiredAccessToken"}
```

III. RESULTS

The outcome of the above-mentioned considerations and architecture description was a software application aimed for advanced analytics and named Visualisation Engine.

The ability of visualising data deriving from large scale pilots is becoming increasingly needed, taking into consideration that nowadays data are generated in a vast rate and must be exploited in order to lead to important insights that will

[20] https://www.keycloak.org/
[21] https://oauth.net/2/
[22] https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2
[23] https://wso2.com/articles/2019/3/user-managed-access-in-action-part-one/

help towards the big challenge of our days; meeting the standards to avoid climate change.

The Visualisation Engine proposed in the present paper is implemented to offer to the users the capability of visualising the requested data in various formats including a variety of common visualisation charts (e.g., pie charts, bar charts, gauge charts, radar charts and line charts), as well as the possibility of rendering interactive maps if the data include geospatial information, such as coordinates. Simple tables are also available for the data representation, along with smart query possibilities, in order to maximise the potential exploitation of the available data.

One of the main offerings of the proposed Visualisation Engine is its integration with the security mechanism, which is another component we developed for the security aspect of our implementation. During the course of this project, we followed a microservice-oriented implementation workflow which resulted in these two components. The main outcome of the integration of these two components was the fact that, unlike other visualisation tools, out application applies role-based access control. This feature is of utter importance, especially if we take into consideration that building data, or project data related to buildings (consumptions, weather data, etc.) are sensitive and shall be protected from being accessed by unauthorized parts.

A. Functionality

The first step for the usage of the Visualisation Engine is the Sign In process. For security reasons, the visualisations need to be only available for signed-in users, as the data displayed could be sensitive and/or confidential. To sign in to the platform, the users are automatically navigated to the Sign In page (Fig. 3), where they are prompted to fill in the "Username" and "Password" fields. For the implementation of the Sign In functionality, a request is sent to the back-end application, which contains the provided username and password. If the credentials are correct, the back-end's response contains an access token. This access token is stored in the front-end application and through the introspection functionality provided by the back-end application, the access rights of this particular user (e.g., which datasets they can access) are decided.

The sidebar menu consists of three main items. The first one, namely Visualisations, consists of a number of subitems, which are matched with the different charts that are available (pie chart, bar chart, gauge chart, radar chart and line chart). The second one is named "SQL query tables" and the third one is named "Maps". Additionally, a User Profile page has been implemented. This page contains the basic information of the user, as well as the user's latest queries and files uploaded. All the aforementioned pages will be thoroughly described in the following section.

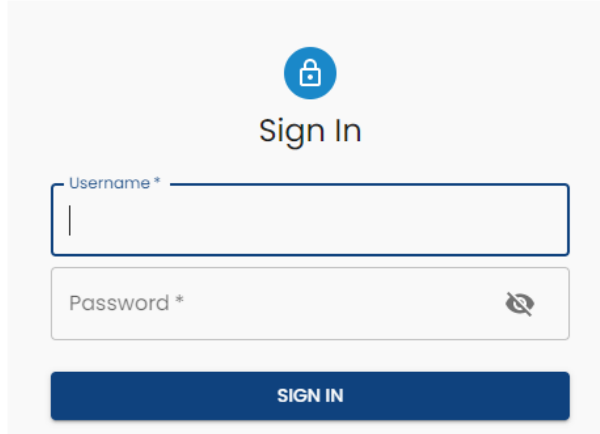

Fig. 3: Sign In page.

B. Pages

1) Visualizations

In order to create a visualisation, the uses must fill in a form which contains various fields, including the table of which the data need to be visualised and the aggregation metric that will be applied (SUM, AVG (average), COUNT, MIN (minimum), MAX (maximum)). Upon the selection of the table, a request is sent to the back-end, in order to retrieve the columns of this specific table. The corresponding form fields are automatically updated, and the users must choose the column of the table that they need to visualise. A "LIMIT" field is also provided, in case that the number of rows has to be limited and, finally, a checkbox with the label "3D chart" is provided. Based on the value of this checkbox, the visualisations are displayed in diverse ways. Fig. 4, Fig. 5, Fig. 6, Fig. 7, and Fig. 8 illustrate one instance of each chart available.

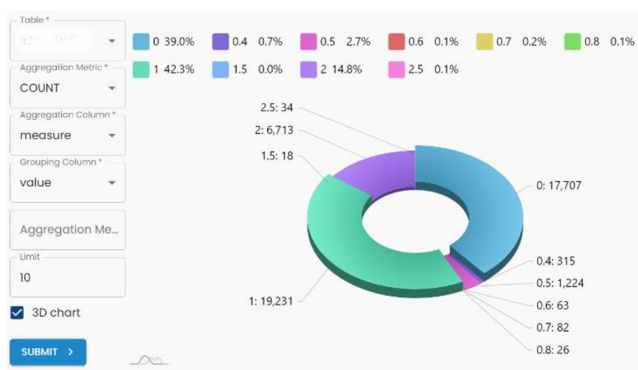

Fig. 4: Pie Chart example.

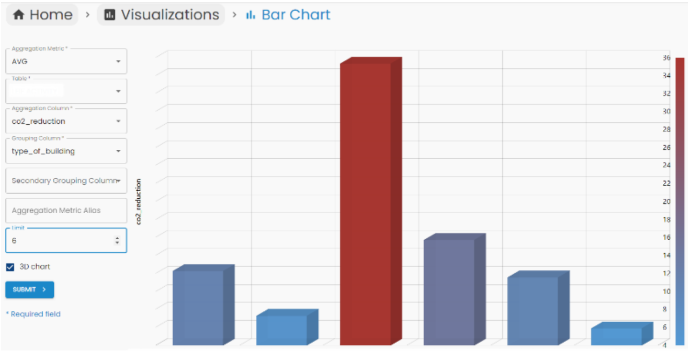

Fig. 5: Bar Chart example.

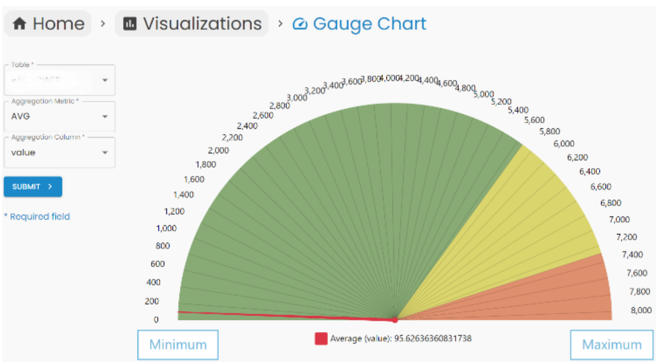

Fig. 6: Gauge Chart example.

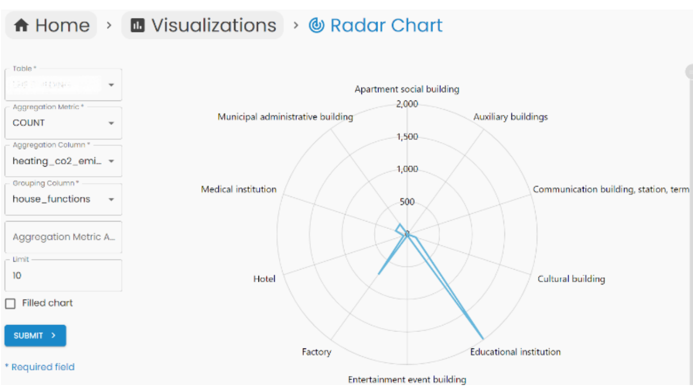

Fig. 7: Radar Chart example.

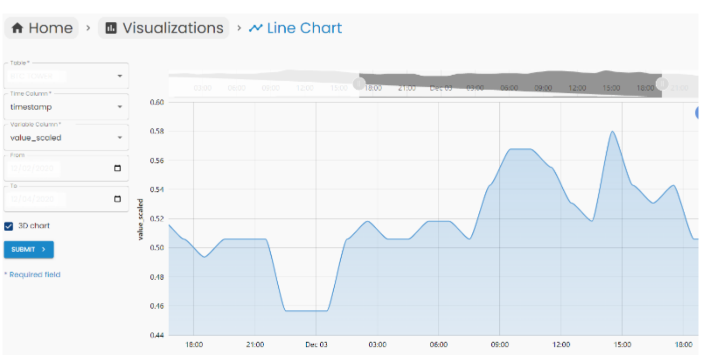

Fig. 8: Line Chart example.

Apart from charts, the users can also visualise data into tables. This capability is offered through the "SQL Query tables" page of the Visualisation Engine. In this page, the user can either submit a plain SQL Query (Fig. 9), or fill a form (Fig. 10) which includes all the needed fields and following sent to the back-end application in order to retrieve the needed data. The retrieved data are then displayed into responsive and adjustable tables. A screenshot of an example query is provided in Fig. 11.

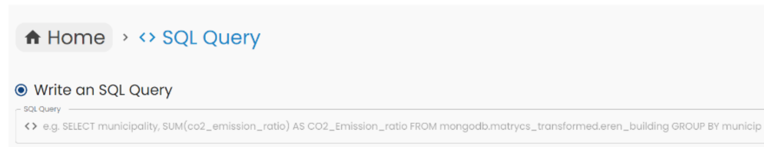

Fig. 9: Text field for submitting common SQL commands.

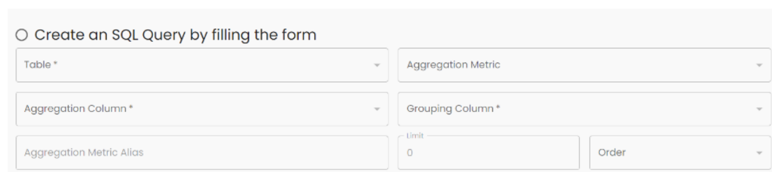

Fig. 10: Form for submitting queries

municipality	CO2_Emission_Rating_S
VALLADOLID	24989
SALAMANCA	15286
BURGOS	13837
LEÓN	12396
PALENCIA	6710
ÁVILA	4574
ZAMORA	4562
SEGOVIA	3533
PONFERRADA	3457
MIRANDA DE EBRO	3032

Fig. 11: Table containing retrieved data.

2) Maps

In case the data include coordinate information, they can be visualised into maps (**Error! Reference source not found.**). These maps are created by using the Leaflet.js library and can be zoomed-in and out. Each entity is represented as a pin on the map and upon clicking on each pin, the requested information is displayed. Fig. 12 illustrates an example of a map visualization.

3) User Profile Page

The User Profile Page contains several basic information about the user, provided during the user's registration to the platform. It also contains a list of the roles assigned to the particular user, as well as a list of the last queries that she / he has submitted to the pages that provide visualisations. Additionally, this page provides the user with the capability of uploading files to the Visualisation Engine via Comma Delimited Value (CSV) files. To upload a file, the user must specify the columns and the type of value (INTEGER, FLOAT, DOUBLE, DATE, STRING, BOOLEAN, BINARY) they contain, as well as to specify a name for the specific file. Fig. 13 depicts the above-described functionality and the created component for this procedure. After uploading the file, the back-end processes, stores, and relates it with the user in question. Following, the file is available for downloading and it is also available for all the potential visualisations. A snapshot of the User Profile page, in a testing environment, is presented in Fig. 14.

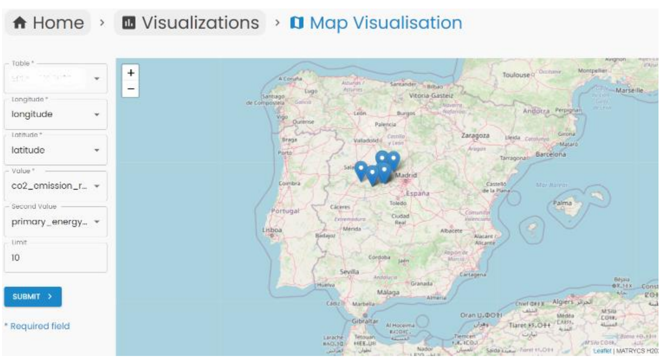

Fig. 12: Map visualisation example

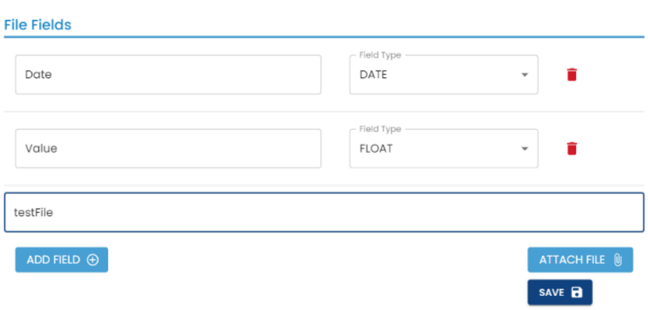

Fig. 13: Upload file form.

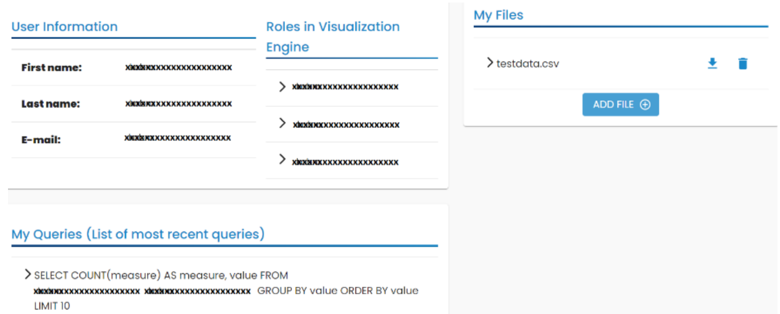

Fig. 14: User Profile Page.

IV. USE CASE SCENARIO DEMONSTRATION

For clarity reasons and in order to thoroughly demonstrate the proposed Visualisation Engine's added value, this section describes how a potential user, related to a specific large-scale pilot of the project, could take advantage of the offered functionalities.As described, not all the datasets are available to every user. On the contrary, the users, based on the role(s) they are assigned to, can have access and, thus, visualize / process the data of specific datasets.

In the context of MATRYCS project, a handful of pilots are involved providing various data. One of them, provides data related to actual performance of investments in terms of energy savings. Assuming that a user needs to visualise these data and extract conclusions about the achieved CO_2 reductions that were achieved due to the realisation of the aforementioned investments. Using the "SQL query tables" page, the user may, for instance, group the sum of the CO2 reduction, absolute numbers, grouped by the types of buildings the investments were made to (Fig. 15).

type_of_building	CO2_Reduction
residential buildings (households)	55.55
sports building and construction	946.0380000000004
cultural institution building	1700.3970239999996
college, higher education institution	1817.8030000000003
municipal administration building	2683.918999999999
medical institution	3297.9169999999995
hotel and other short-stay accommodation	4798.413000000002
office building, warehouse, etc. (companies)	5010.842999999998
preschool educational institution	6251.957999999996
school	27546.198000000022

Fig. 15: Table including CO2 Emission Reduction for each building type investments were made to in the context of a specific large-scale pilot.

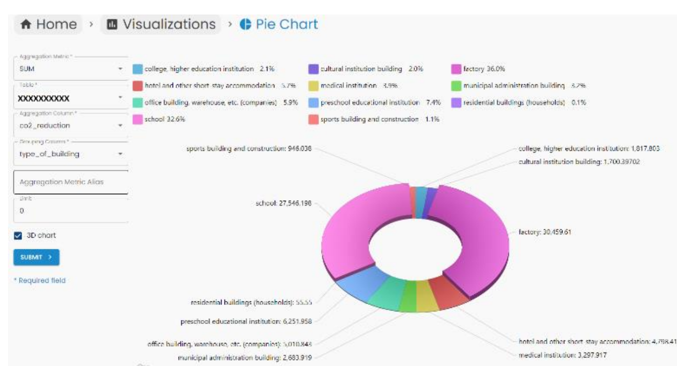

Fig. 16: Pie Chart including CO2 Emission Reduction for each building type investments made to in the context of a specific large-scale pilot.

Following, the user can also turn the same query into a Pie Chart (Fig. 16), in order to see the percentage distribution among the same investments.

Additionally, the user can export useful insights by co-relating CO2 Emission reductions with energy consumption reductions, a metric that is also offered by this specific pilot. This co-relation can either by presented in the form of tables (**Error! Reference source not found.**) or via the charts that are offered.

In this way, a wide range of insights can be exported for each and every one of the available datasets. The users can visualise the required data in plenty of ways and, thus, extract useful information from them, as well as display them in a more readable way. On top of this, the project's data are secured, via the full integration that has been implemented with the Security mechanism described in the previous sections. This is one of the most important offerings of the proposed Visualisation Engine since it ensures data privacy and requires no further configurations before setting up for usage.

V. Conclusions And Future Work

This paper introduced a Visualisation Engine. The proposed application offers a wide variety of visualisation capabilities to the user, and it also implements role-based access control over the stored data, by being fully integrated with the security framework also developed in the context of the same project. With this work, we envision to assist towards the direction of creating useful charts and visualisations that will be able to lead to useful and meaningful conclusions and insights of buildings' functional needs and energy consumption metrics. The followed development approach was modular and microservice-oriented, meaning that all the developed components, including the Visualisation Engine, can be easily adjusted to visualise any properly formatted data.

As future work, we have identified several features that can be added to the above-described application. The provided functionalities will be extended with even more aggregate methods available and further query capabilities (e.g., JOIN queries) in the next versions of our application. Further visualisation capabilities will be added, meaning more chart options. Additionally, we are planning to also offer JOIN query capabilities, to provide even more detailed and complex data exploration capabilities than can lead to more in-depth insights.

Acknowledgment

This work has been funded from the European Union's Horizon 2020 research and innovation program under the MATRYCS project 'Modular Big Data Applications for Holistic Energy Services in Buildings', grant agreement No 101000158.

References

[1] V. Marinakis, "Big Data for Energy Management and Energy-Efficient Buildings," *Energies*, vol. 13, no. 7, Art. no. 7, Jan. 2020, doi: 10.3390/en13071555.

[2] G. Hernández-Moral *et al.*, "Big Data Value Chain: Multiple Perspectives for the Built Environment," *Energies*, vol. 14, no. 15, Art. no. 15, Jan. 2021, doi: 10.3390/en14154624.

[3] H. Doukas and A. Nikas, "Decision support models in climate policy," *Eur. J. Oper. Res.*, vol. 280, no. 1, pp. 1–24, Jan. 2020, doi: 10.1016/j.ejor.2019.01.017.

[4] C. A. Horowitz, "Paris Agreement," *Int. Leg. Mater.*, vol. 55, no. 4, pp. 740–755, Aug. 2016, doi: 10.1017/S0020782900004253.

[5] E. Costanza, S. D. Ramchurn, and N. R. Jennings, "Understanding domestic energy consumption through interactive visualisation: a field study," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, New York, NY, USA, Jun. 2012, pp. 216–225. doi: 10.1145/2370216.2370251.

[6] K. Vikhorev, R. Greenough, and N. Brown, "An advanced energy management framework to promote energy awareness," *J. Clean. Prod.*, vol. 43, pp. 103–112, Mar. 2013, doi: 10.1016/j.jclepro.2012.12.012.

[7] E. Sarmas, E. Spiliotis, V. Marinakis, T. Koutselis, and H. Doukas, "A meta-learning classification model for supporting decisions on energy efficiency investments," *Energy Build.*, vol. 258, p. 111836, Mar. 2022, doi: 10.1016/j.enbuild.2022.111836.

[8] T. A. Mohammed, A. Ghareeb, H. Al-bayaty, and S. Aljawarneh, "Big data challenges and achievements: applications on smart cities and energy sector," in *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems - DATA '19*, Dubai, United Arab Emirates, 2019, pp. 1–5. doi: 10.1145/3368691.3368717.

[9] N. Koseleva and G. Ropaite, "Big Data in Building Energy Efficiency: Understanding of Big Data and Main Challenges," *Procedia Eng.*, vol. 172, pp. 544–549, 2017, doi: 10.1016/j.proeng.2017.02.064.

[10] C. Burnay, F. Dargam, and P. Zarate, "Special issue: Data visualization for decision-making: an important issue," *Oper. Res.*, vol. 19, no. 4, pp. 853–855, Dec. 2019, doi: 10.1007/s12351-019-00530-z.

[11] D. Lee, G. Cha, and S. Park, "A study on data visualization of embedded sensors for building energy monitoring using BIM," *Int. J. Precis. Eng. Manuf.*, vol. 17, no. 6, pp. 807–814, Jun. 2016, doi: 10.1007/s12541-016-0099-4.

[12] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, "Visual Analytics: Definition, Process, and Challenges," in *Information Visualization: Human-Centered Issues and Perspectives*, A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, Eds. Berlin, Heidelberg: Springer, 2008, pp. 154–175. doi: 10.1007/978-3-540-70956-5_7.

[13] S. Grainger, F. Mao, and W. Buytaert, "Environmental data visualisation for non-scientific contexts: Literature review and design framework," *Environ. Model. Softw.*, vol. 85, pp. 299–318, Nov. 2016, doi: 10.1016/j.envsoft.2016.09.004.

[14] S. Batt, T. Grealis, O. Harmon, and P. Tomolonis, "Learning Tableau: A data visualization tool," *J. Econ. Educ.*, vol. 51, no. 3–4, pp. 317–328, Sep. 2020, doi: 10.1080/00220485.2020.1804503.

[15] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Comput. Sci. Eng.*, vol. 9, no. 03, pp. 90–95, May 2007, doi: 10.1109/MCSE.2007.55.

[16] P. Kapsalis, G. Kormpakis, K. Alexakis, and D. Askounis, "Leveraging Graph Analytics for Energy Efficiency Certificates," *Energies*, vol. 15, no. 4, Art. no. 4, Jan. 2022, doi: 10.3390/en15041500.

[17] M. Pau, P. Kapsalis, Z. Pan, G. Korbakis, D. Pellegrino, and A. Monti, "MATRYCS—A Big Data Architecture for Advanced Services in the Building Domain," *Energies*, vol. 15, no. 7, Art. no. 7, Jan. 2022, doi: 10.3390/en15072568.

[18] E. Maler, M. Machulak, and J. Richer, "User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization," Jan. 07, 2018. https://docs.kantarainitiative.org/uma/wg/rec-oauth-uma-grant-2.0.html (accessed Apr. 09, 2022).