



D4.4. Service Layer

Lead: Engineering Ingegneria Informatica S.p.a.

Contributors: Cetaqua Andalucía

Date: February 2022

Public



Project deliverable

Project Number 1922	Project Acronym GOTHAM	Project Title Governance Tool for sustainable water resources allocation in the Mediterranean through stakeholder's collaboration. Towards a paradigm shift in groundwater management by end users.
Instrument: Research and Innovation Action		Thematic Priority: Mediterranean water co-operation
Title D4.4 Service Layer		
Contractual Delivery Date 28/02/2022		Actual Delivery Date 28/02/2022
Organisation name of lead contractor for this deliverable Engineering – Ingegneria Informatica S.p.A.		Document version V1.0
Dissemination level Public X Confidential		Deliverable Type Document, Report Demonstrator X
Authors (organisation) ENG, CET		
Reviewers (organisation) ENG, CET		
Abstract <p>The aim of this report is to document the software prototype (D4.4) developed in the context of Task 4.4 of the Gotham project. The result of the Task represents the Service Layer components of the overall GTool architecture (see D4.1). In such architecture, the Service Layer is responsible for expose, through Dashboards with graphs and maps, data and result produced by the Analytics Layer, taking inputs from T4.1. This module will also allow to define and monitor KPIs for the</p>		

evaluation of the impacts resulting from an action of a decisional process.

Keywords

Dashborad, GTool, graph, map, data visualization

Disclaimer

This document is provided with no warranties whatsoever, including any warranty of merchantability, non-infringement, fitness for any particular purpose, or any other warranty with respect to any information, result, proposal, specification or sample contained or referred to herein. Any liability, including liability for infringement of any proprietary rights, regarding the use of this document or any information contained herein is disclaimed. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by or in connection with this document. This document is subject to change without notice.

GOTHAM has been financed by the European Commission.

This document reflects only the view of the authors and the European Commission cannot be held responsible for any use which may be made of the information contained herein.

Project summary

The overarching objective of the GOTHAM project is to develop and validate a user-driven tool that enables effective groundwater governance to ultimately preserve the quantity and quality of this strategic resource in the Mediterranean basin. The GOTHAM Tool (GTool) uses an integrated methodological approach that targets optimal allocation of water resources from an environmental, social and economic perspective, including stakeholder knowledge, priorities and behaviour.

One of the main strengths of the tool is that it provides a common framework for collaboration and engagement of the different water users (mainly, agricultural communities but also municipal and industrial users), as well as other relevant stakeholders such as water producers/operators and regulator(s). The GTool will enable them to exchange information in order to reach the optimal water governance at each point in time as well as in future scenarios.

The concept of the proposed GTool targets effective groundwater governance for the improvement of the management and preservation of this essential and strategic resource. This effective groundwater management remains an important and complex challenge in the Mediterranean and elsewhere, but is essential to ensure long-term sustainable use of the resource.

In this regard, GOTHAM integrates multicriteria decision methods for stakeholder group decision making and social learning, and use socio- hydrological water balance framework as a theoretical foundation for water allocation to evaluate the dynamic balance between the societal and ecological systems in catchments. GOTHAM project presents a bottom-up decision-making approach inspired in this methodological framework.

GOTHAM project presents a scalable and user-specific tool for decentralising water resources management, using big data analysis. The proposed user-based tool leverages six analytical modules:

- The **water balance and water quality dynamics module** uses advanced investigation of the main aquifer formations and real-time monitoring (on site and distant), including preliminary analysis of the background hydrogeological and hydro-meteorological information to create a baseline.
- The **water availability and demand forecasting module** predicts different water scenarios and assess their impact on groundwater quality and quantity status using remote-sensing measurements to model agriculture water demand and assess water availability.
- The **Managed Aquifer Recharge (MAR) and aquifer remediation module** mobilises multicriteria analysis (QGIS environment), including hydrogeological, economic, and chemical (water quality) indicators as well as regulatory restrictions to evaluate the feasibility of MAR schemes.
- The **agro-economic module** simulates the effect of different economic instruments, such as water tariff structures, water markets contexts and incentives for water savings (water demand management) and assessing the economic use values and trade-offs between users in alternative resource allocation scenarios.
- The **user's engagement module** enables to fix water priorities (water boundary conditions) by water users, taking into consideration water resources to meet water demands.

- The **optimised water allocation module** calculates the optimal mix of water sources satisfying their requirements

Gtool uses data visualisation techniques to deliver the results into customisable dashboards tailored for the needs of each stakeholders.

Broad outreach activities will take place in Europe, Lebanon and Jordan, therefore contributing to GOTHAM impact maximization.

The further development and exploitation (beyond the project) of the GTool will be done by CETaqua, both on B2B and B2C approaches.

Table of content

Table of content	6
1. Introduction	9
2. Software Modules.....	10
2.1. Docker	10
2.1.1. Description.....	10
2.1.2. Technical Details	10
2.1.3. Installation	11
2.2. Airflow.....	11
2.2.1. Description.....	11
2.2.2. Technical Details	11
2.2.3. Installation	11
2.3. Dashram	11
2.3.1. Description.....	11
PCA Chart.....	13
Raster Image Chart.....	14
Kmeans Chart	16
Wilcox Chart	18
Piper Chart.....	19
Ionic 1-2-3 Charts	22
Composed Chart (Multiline, Map, Table).....	23
Mannkendall Chart.....	25
2.3.2. General information about charts	26
2.3.3. Dashboard.....	28
Hydrochemical dashboard.....	28
Groundwater potential dashboard	30
KPIs	31
2.4. APIs.....	32
2.4.1. Technical Details	32

2.4.2. Installation	33
---------------------------	----

Table of figures

Figure 1 Image which depicts project architecture.....	9
Figure 2 This image illustrates the Dashram's home page for the technical users.....	12
Figure 3 This image illustrates the Dashram's home page for the non-technical users	13
Figure 4 This image illustrates results of PCA algorithm execution.	13
Figure 5 This image explains how PCA works with data.	14
Figure 6 This image represents the raster image chart with an applied raster level.....	14
Figure 7 Raster image chart flow.....	15
Figure 8 Creation of style in the GeoServer	16
Figure 9 This image illustrates results of Kmeans algorithm execution.....	16
Figure 10 This image explains how Kmeans works with data and how it is built using dataset.....	17
Figure 11 Plot of Silhouette method is depicted in this image	18
Figure 12 This image illustrates results of Wilcox algorithm execution.....	18
Figure 13 This image explains how Wilcox chart works with data.....	19
Figure 14 Piper chart	19
Figure 15 This image explains how Piper chart works with data.	20
Figure 16 This couple of image explains how to read Piper diagram step by step.....	21
Figure 17 This image explains the final step of reading explanation of Piper diagram.	21
Figure 18 This image represents all ionic diagrams.	22
Figure 19 The left image explains the FreshWater/Sea Water guide line schema. The right image explains the schema for two guide lines.	23
Figure 20 Prediction chart. It is not a single	23
Figure 21 Composed chart pipeline	24
Figure 22 Mannkendall chart	25
Figure 23 Mannkendall chart pipeline.....	25
Figure 24 This image explains how tooltip works on the chart.....	26
Figure 25 Remove element from legend and chart	27

Figure 26 Section for creating charts	28
Figure 27 The whole “hydrochemical dashboard”. It contains Piper, PCA, KMeans, Wilcox, Ionic1-2-3 and Mannkendall charts.....	29
Figure 28 Mobile version of hydrochemical dashboard. It contains less charts than the desktop version; this is due to user-experience motivations. In addition, dashboard differs in disposition of charts themselves respectful to desktop version.	30
Figure 29 Desktop version of the groundwater potential dashbord.	31
Figure 30 KPI dashboard is shown in this image. Dashboard contains just one KPI chart but it can be improved further accordingly to users need.....	31
Figure 31 List of APIs	32

1.Introduction

The aim of this report is to document the software prototype (D4.4) developed in the context of Task 4.4 of the Gotham project. The result of the Task represents the Service Layer components of the overall GTool architecture, which is depicted below and that has been defined in Task 4.1 (D4.1).

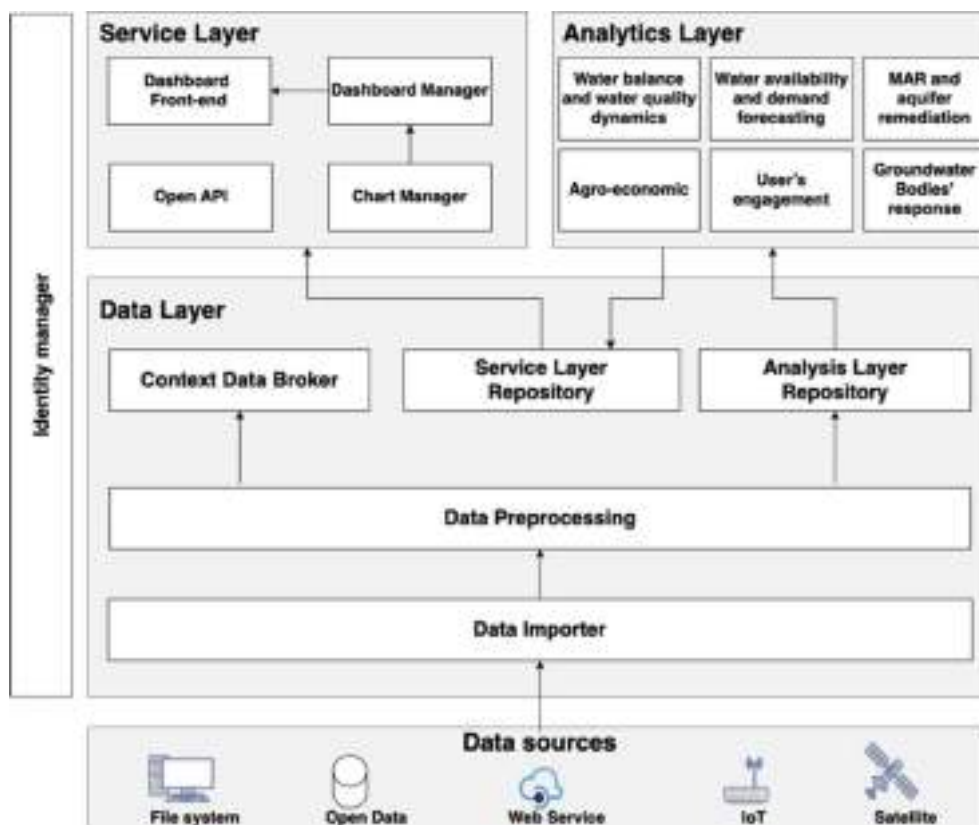


Figure 1 Image which depicts project architecture

In general terms the architecture is composed of three main layers:

- the GTool Data Layer will enable a smooth interfacing with the different and heterogeneous data sources of the water domain.
- the GTool Analytics Layer will provide multicriteria analysis (availability, demand, quality, cost) regarding water allocation scenarios and their impact on the environmental, economic and social sustainability of the water resources.
- the GTool Service Layer exposes, in different ways, data and results produced by the Analytics Layer and data imported from other sources.

The rest of the document describes in a technical way each component of the Service Layer, providing a general description, a table with some technical details and some indications for the installation.

2. Software Modules

2.1. Docker

2.1.1. Description

Docker is an opensource software platform that allows to create, test and deploy applications. It gathers software into standardized units called Containers that offer everything needed for their proper execution, including libraries, system tools, code and runtime. It is also possible to distribute and recalibrate resources for an application in any environment, always keeping the executed code under control. This is a native Linux technology but is available for other popular platforms.

Main reasons in using Docker are:

- The most important feature is the isolation capacity of the processes. Isolating processes ensures that they can be run independently but simultaneously. This independence is the goal of Containers: the ability to run multiple processes and applications separately to make the most of the existing infrastructure while maintaining the level of security that would be guaranteed by the presence of separate systems (such as with virtual machines);
- Using containers, resources can be isolated, services limited and processes started so that it is possible to have a completely private perspective of the operating system, with their own identifier, file system, and network interfaces. Multiple containers share the same kernel, but each of them can be forced to use a certain amount of resources, such as CPU, memory and I/O;
- Using Docker to create and manage containers can simplify the creation of distributed systems, allowing different applications or processes to work independently on the same physical machine or on different virtual machines. This allows a **Platform as a service (PaaS)** style of development.

The community version of Docker is released under the **Apache 2** license, while the Enterprise version is released with a proprietary license.

2.1.2. Technical Details

Type	Middleware
Programming Language	GoLang
Additional libraries	Docker Compose

2.1.3. Installation

The installation procedure depends on the operating system used and also changes depending on the version of the operating system in use. It is well described in the Docker official documentation¹.

2.2. Airflow

2.2.1. Description

The open-source workflow tool Airflow was used to pre-process the data. Airflow is a platform to programmatically create, schedule, and monitor workflows. Pipelines are created as directed acyclic graphs (DAGs) of tasks and Airflow's scheduler executes the tasks on an array of workers following specified dependencies. Pipelines are configured as code (Python), which allows them to be generated dynamically. It is also possible to simply define our own operator by extending the libraries provided by the tool. Airflow has a rich user interface that makes it easy to visualize pipelines running in production, troubleshoot when needed, and monitor progress, and thanks to its modular architecture it uses a message queue to orchestrate an arbitrary number of workers.

2.2.2. Technical Details

Type	
Programming Language	Python
Additional libraries	Pandas, Airflow

2.2.3. Installation

The installation procedure depends on the operating system used and also changes depending on the version of the operating system in use. It is well described in the Docker official documentation.

2.3. Dashram

2.3.1. Description

Dashram is an open source tool for advanced data visualization, featuring a simple and functional interface that provides the means to create and share 2D and 3D charts, map-based visualizations and dashboards and is based on the open source Apache Superset tool.

Dashram is a highly scalable and customizable tool, it uses a Postgres database to store information, but its architecture allows to add more than one SQL-speaking datastore even different from each other. This tool is also able to retrieve information from different sources, in addition to databases, including: open data portals, FIWARE Context Broker, importing csv or xls files from the filesystem or even using third party JSON Service.

In order to access and visualize the data it is necessary to login, therefore having a user and a password. Thanks to this and through the management of roles made available it is possible to create two different types of users: a technical user, who is able to access the platform to create and/or edit charts and dashboards and then accessing view all the information.

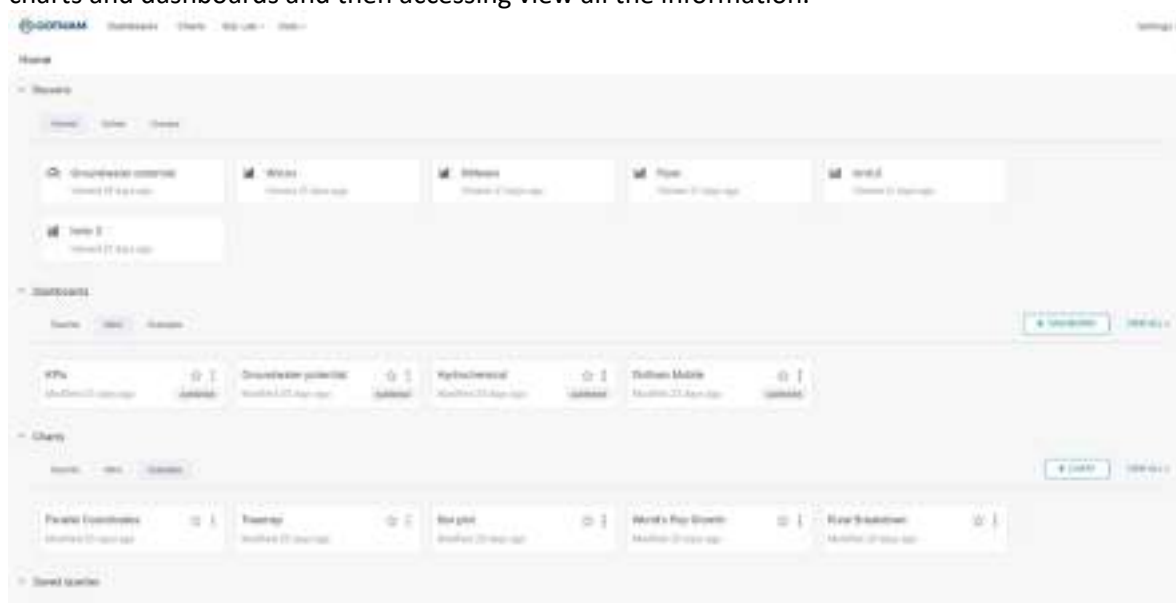


Figure 2 This image illustrates the Dashram's home page for the technical users.

On the other hand, non-technical users are just able in viewing dashboards related to their geographic location and role. In this way, each user accessing the platform can only view the dashboards that pertain to them.

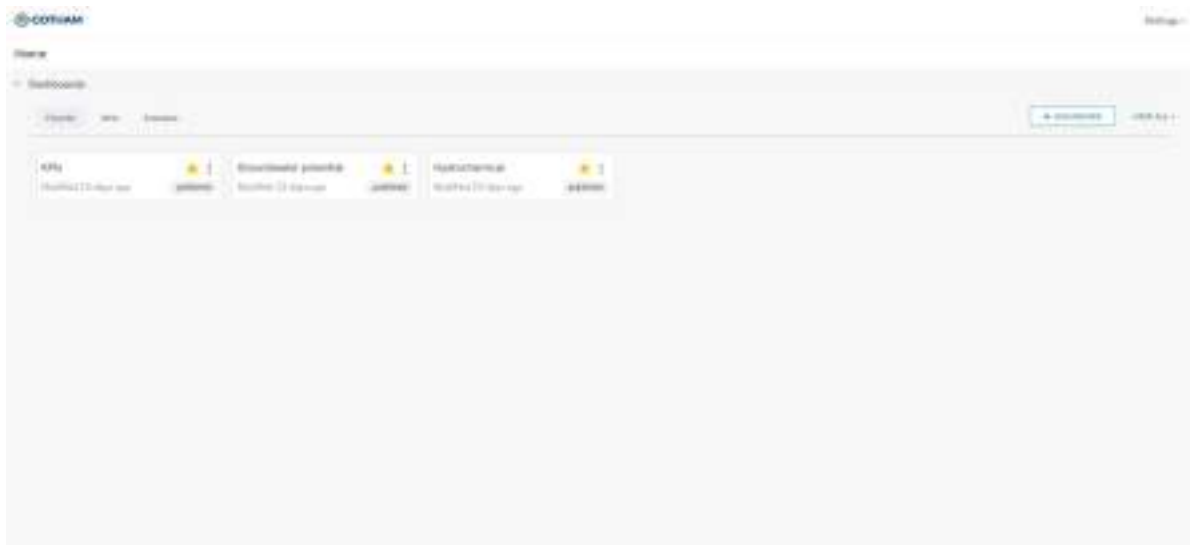


Figure 3 This image illustrates the DASHRAM's home page for the non-technical users

In order to visualize the results coming from the different modules of the Analytics Layer, it was necessary to create specific plugins:

PCA Chart

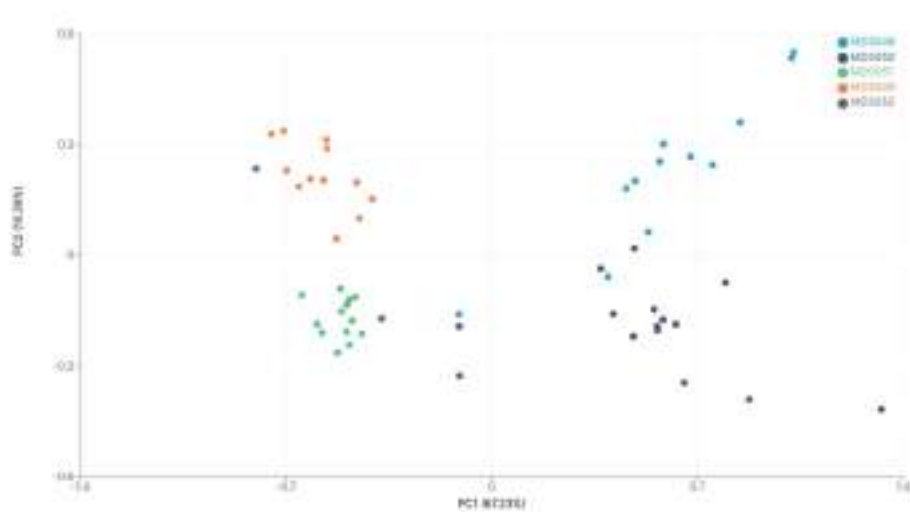


Figure 4 This image illustrates results of PCA algorithm execution.

PCA (which is the acronym of Principal Component Analysis) is a dimensionality reduction algorithm. It is used for finding out correlations amongst dataset points. Plotted points are the result of a dimensionality reduction from the n-dimensional space of the original dataset, to the 2-dimensional space of the plotted chart. PCA enables users to visualize geographically a multi-dimensional dataset (as Figure 5 depicts).

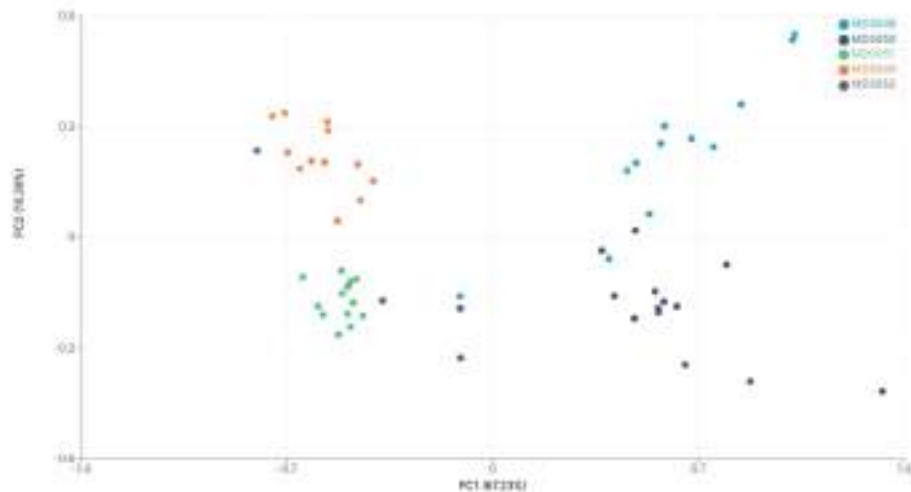


Figure 5 This image explains how PCA works with data.

PCA is a machine learning algorithm and for this reason data have been computed properly before being plotted on the chart. The processing step is preceded by a pre-processing phase which consists of transforming data in "meq", percentages and turn the whole dataset with zero-mean and variance equals to one. After pre-processing phase, the PCA algorithm has been computed and just the first two principal components (those which have got highest variance values) have been considered.

Raster Image Chart

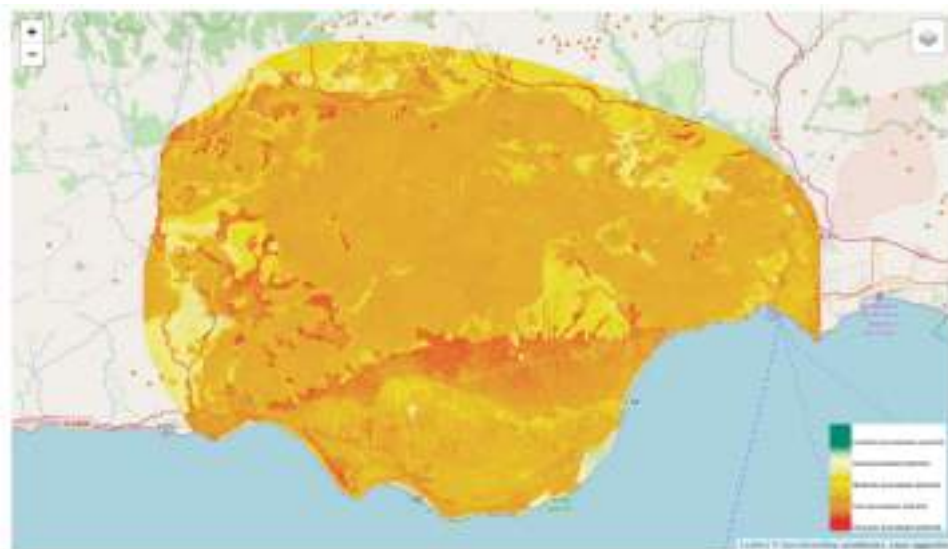


Figure 6 This image represents the raster image chart with an applied raster level.

The graph represents one or more raster geographic images on a base layer. The images are requested by an external GeoServer. This is an open source server written in Java that allows users to share, process and modify geospatial data. The library used to create the map is Leaflet js.

Leaflet is a JavaScript library used to develop interactive geographic maps. It supports most browsers and HTML and CSS standards. This plugin can project on a map any raster image. The raster can represent any geographical region and any value. Communication between Leaflet and Geoserver takes place through requests using the WMS protocol. This protocol provides an HTTP interface for requesting map images, specifying in the request which layers are required and the areas to be processed.

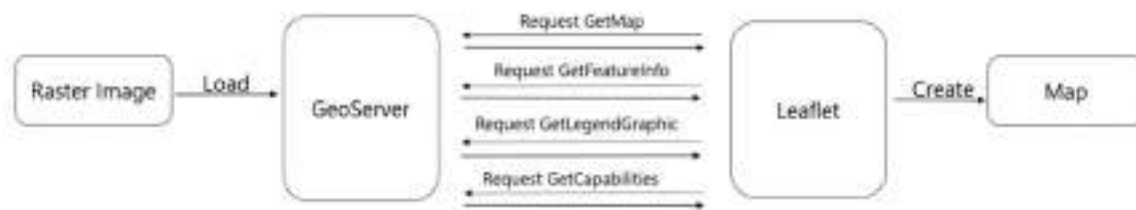


Figure 7 Raster image chart flow

In order to use this chart we assume the existence of a working GeoServer and a list of published layers. Once the images are loaded on the Server it is fundamental to define a default style to be used in the projection of the image on the plugin map. If this does not exist it is necessary to create it. The creation of the style and the legend consists in writing an xml file in which the rules for colouring pixels according to their values are defined.

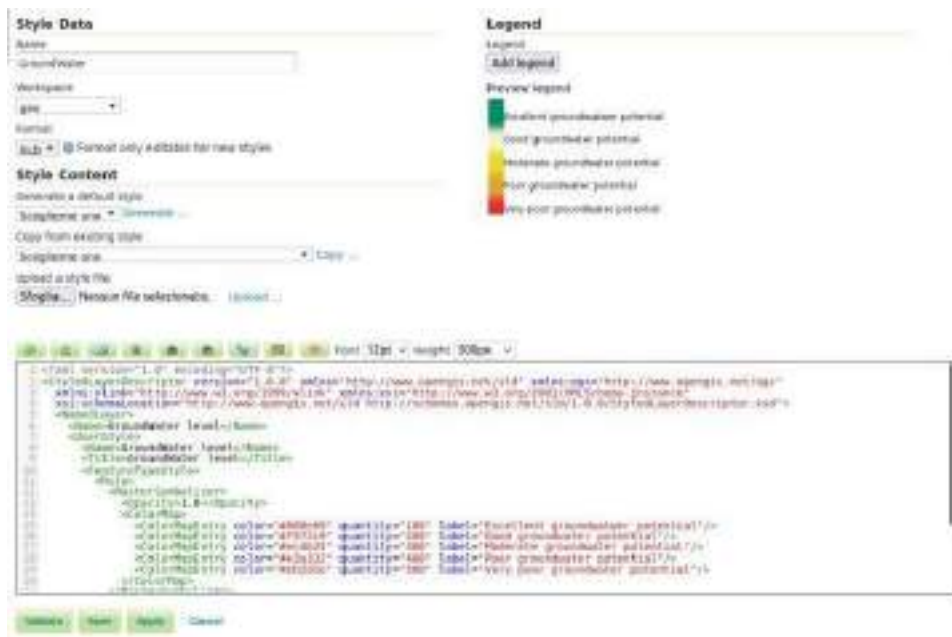


Figure 8 Creation of style in the GeoServer

In the plugin screen will appear in the upper right of the list of levels required to the GeoServer. Once selected, the result of the WMS request 'GetMap' will be loaded in the map. While to be able to focus on the loaded image, and then to know the Bbox, the information is extracted via the request 'GetCapabilities'. Returns an XML file containing the information of each level of the server.

The legend, previously created together with the style assumed by the image, is the result of a WMS request: 'GetLegendGraphic'. Finally there is another feature. it is possible to return the value assumed by a particular selected pixel when clicking on the map by adding this function in the graph creation phase. This is done through the 'GetFeatureInfo' request.

Kmeans Chart

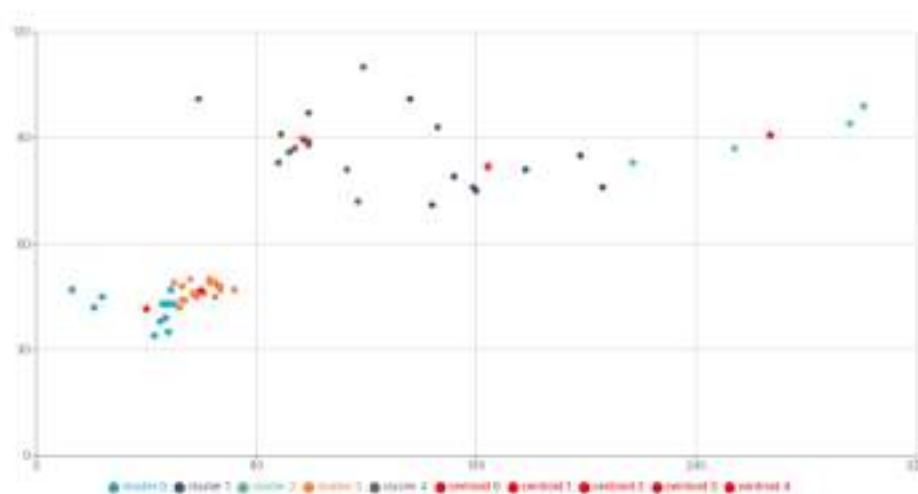


Figure 9 This image illustrates results of Kmeans algorithm execution.

KMeans is a clustering algorithm. Each cluster is identified by a different color and by a centroid, which is the "center" of the whole cluster. KMeans plots a pair of variables. Result is useful to highlight some relations among data (as Figure 10 depicts).

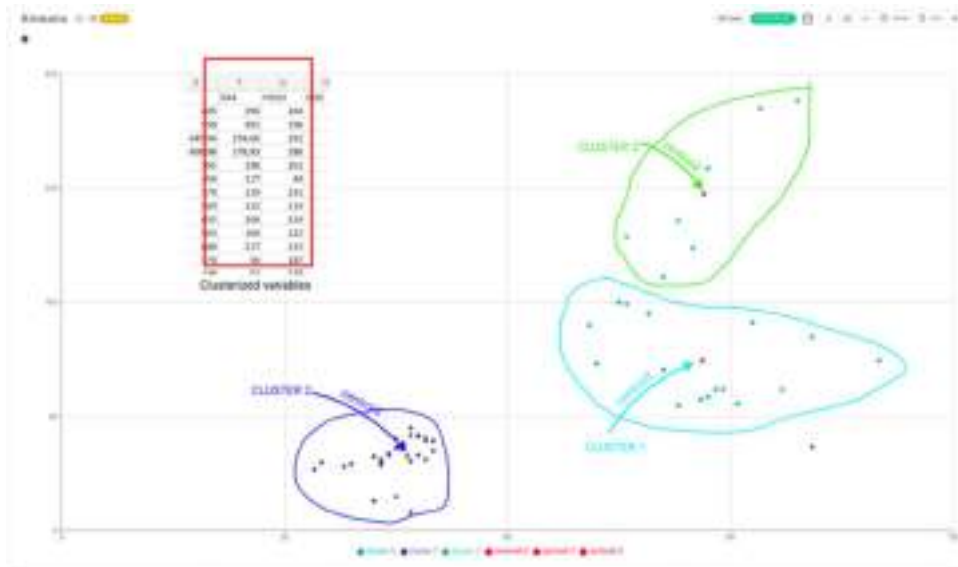


Figure 10 This image explains how Kmeans works with data and how it is built using dataset.

KMeans is performed using a proper library which extracts clusters from a dataset made up by two variables. KMeans needs an hyperparameter: "k". This parameter identifies the number of clusters that the data scientist expects. This is a drawback of the algorithm because users should set this parameter and correlations between variables could change dramatically accordingly to the parameter K. To solve the problem a fitting method has been used: the Silhouette method. It allows to choose dinamically the "k" parameter using the Silhouette score (Figure 11) shows a plot of Silhoutte method which is an alternative to elbow method). As has been said for other plugins, before performing the algorithm, a pre-processing aimed to transform values in "meq" and then in percentages has been performed.

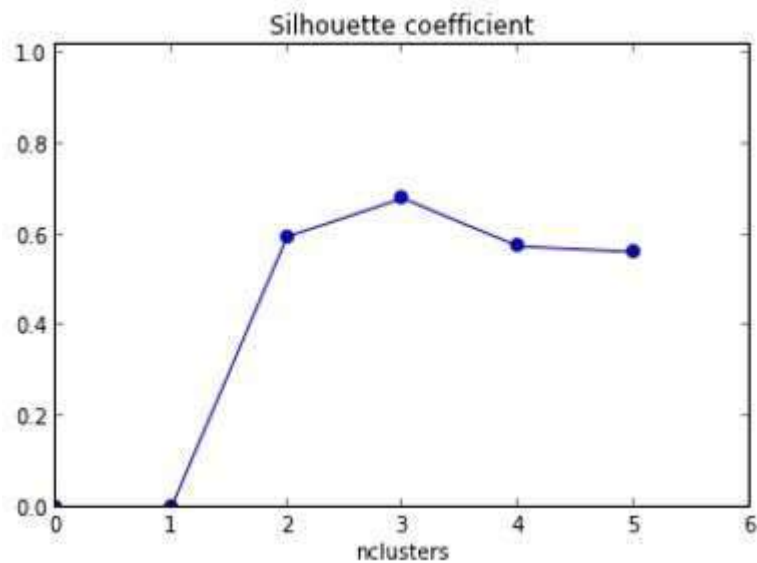


Figure 11 Plot of Silhouette method is depicted in this image

Wilcox Chart

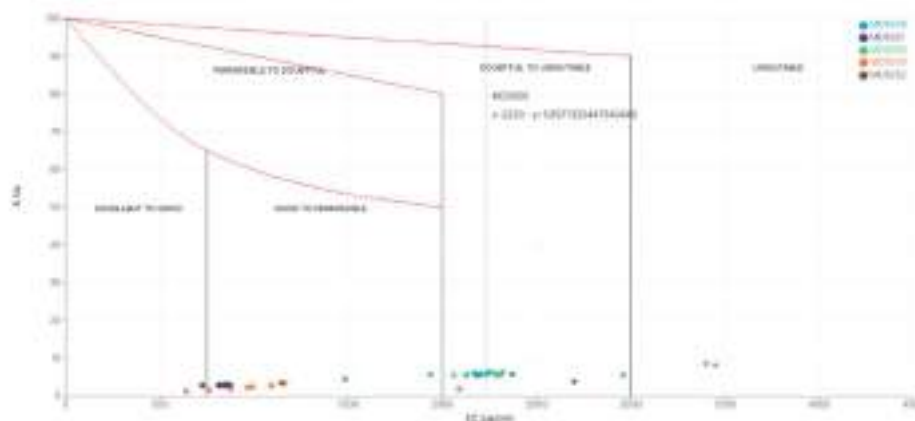


Figure 12 This image illustrates results of Wilcox algorithm execution.

Wilcox diagram is a graphical representation of chemistry of water samples for irrigation quality assessment. Points are plotted directly on chart using as x-coordinate the electrical conductivity (provided by the dataset) expressed in $\mu\text{S}/\text{cm}$ and as y-coordinate the percentage of sodium. Columns which contain have been highlighted (Figure 12 shows a complete execution of Wilcox diagram) . Points are plotted into the diagram area and they fall in sub-areas which are labelled properly. Accordingly to these labels, a classification about water quality could be performed (Figure 13) depicts an example of WilCoX diagram).

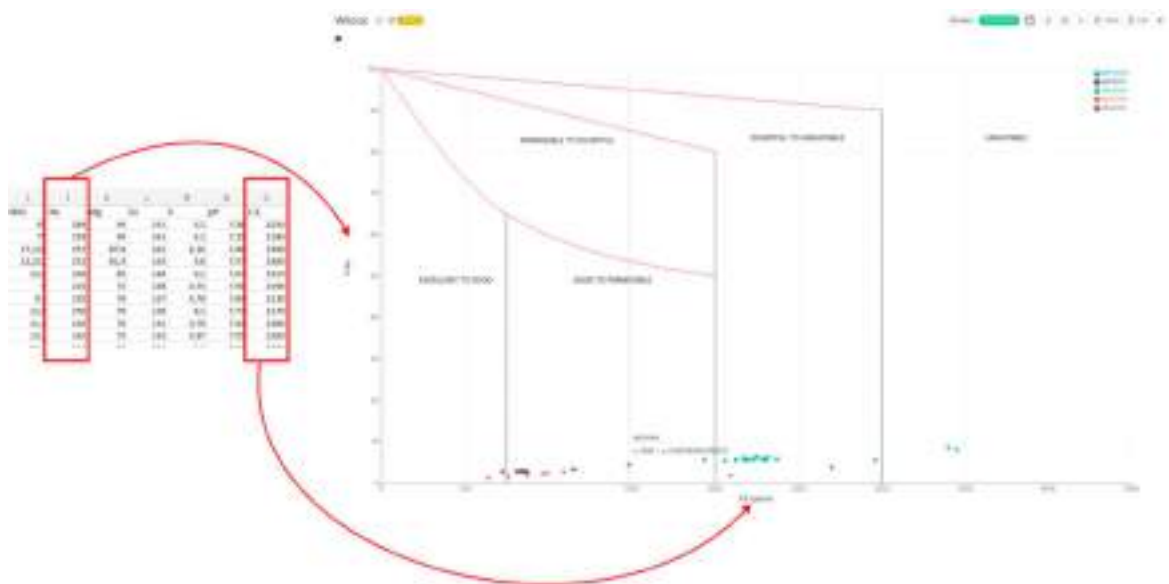


Figure 13 This image explains how Wilcox chart works with data.

Wilcox diagram is just an arrangement of data points, there is no logic under the hood. On the other hand there is a little of data pre-processing for transforming dataset quantities to "meq" and percentages (from the chemical point of view).

Piper Chart

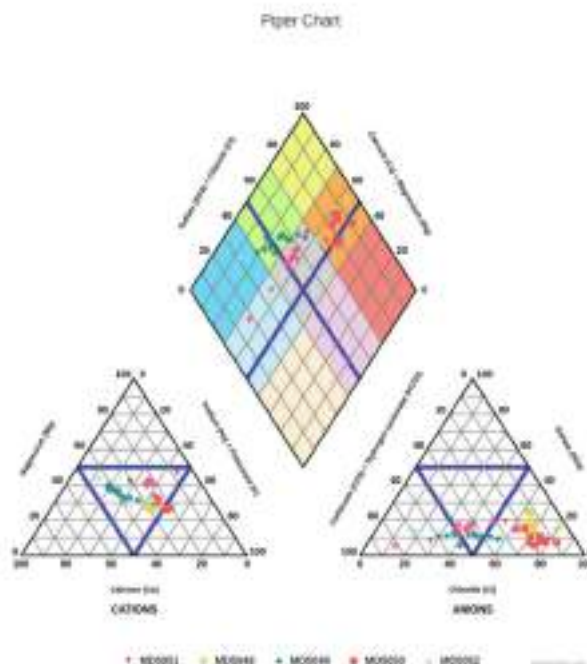


Figure 14 Piper chart

This diagram is used to represent chemistry of water samples. On the left triangle (cations triangle) Magnesium, Calcium and Sodium + Potassium value are plotted. On the right triangle (anions triangle) Sulfate, Chloride and Carbbonate + Hydogen-Carbonate values are plotted. Diamond is used for components classifications accordingly to coloured areas. Based on dataset values data have been plotted on cations and anions triangles. They are identified by a tern of coordinates, each of them represents a percentage as Figure (15) shows.

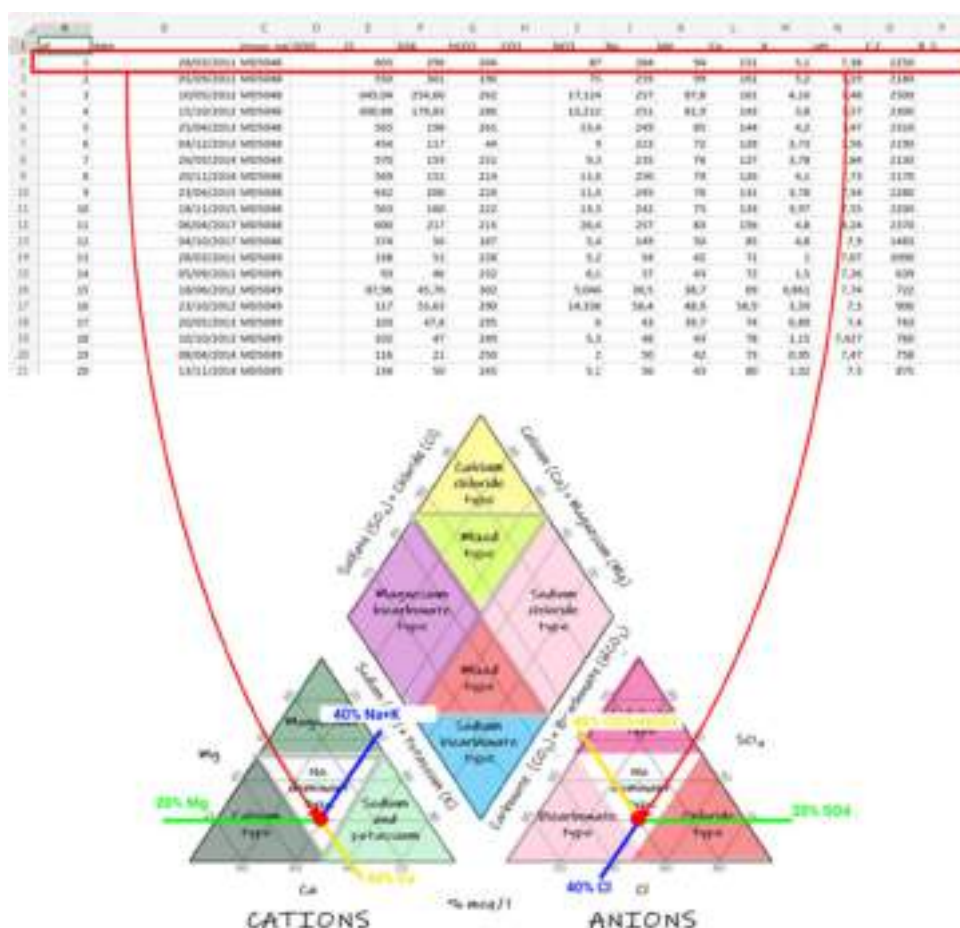


Figure 15 This image explains how Piper chart works with data.

Taking into account left triangle, drawing a line parallels to the "magnesium" axis which passes though a point plotted on Cations' triangle (Line "A" on Figure 16). Taking into account right triangle, drawing a line parallels to the "sulfate" axis which passes though a point plotted on Anions' triangle (Line "B" on the Figure 16);

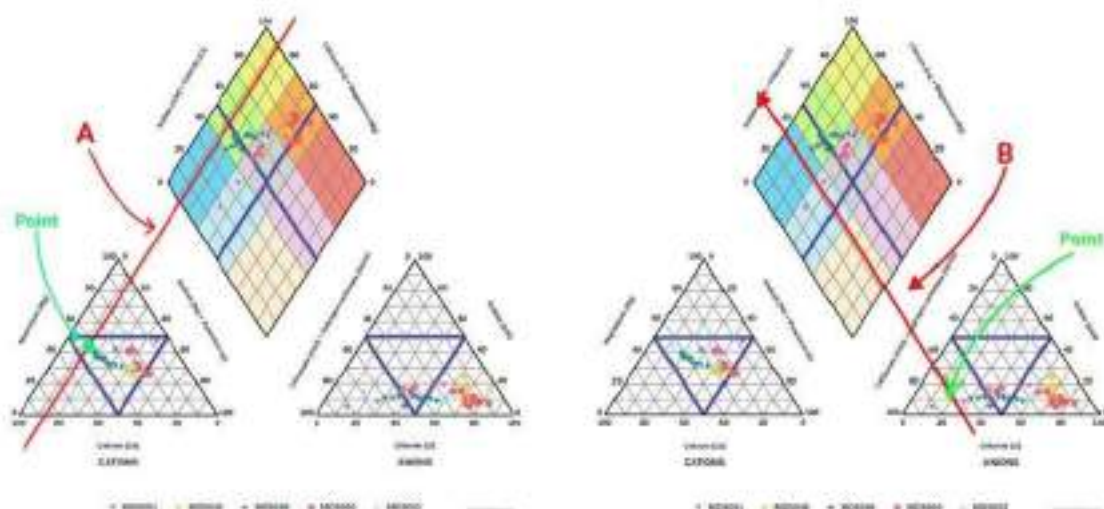


Figure 16 This couple of image explains how to read Piper diagram step by step

the end, to obtain resulting point on the diamond (rhombus in the middle), just intersect line A and line B. Final result depicted in Figure 17.

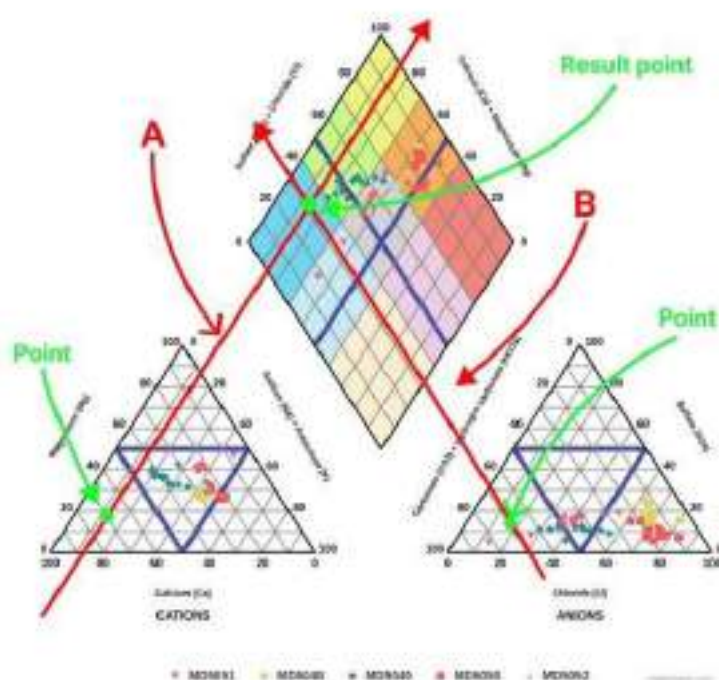


Figure 17 This image explains the final step of reading explanation of Piper diagram.

From the technical point of view, piper diagram development follows same steps explained before. However, deepening some technical points, main steps of Piper diagram development are:

- Converting all values in "meq" (dividing values by their molecular mass);

- Converting all values in percentage following the formula: $\frac{e_i}{\sum e_i} \times 100$ where " e_i " is the i-th chemical element;
- After that, a changing of coordinate system is performed. In fact, using references given by triangles axis, all points have been transposed to a cartesian coordinate system;
- In the end, intersection between two lines is computed. Javascript is not so able to solve equations, for this reason the resulting system is presented in explicit form and parameters used for the computation are: coordinates of two points and angular coefficients of two axis (magnesium and sulfate);
- The result of these steps is the point on the diamond;

Ionic 1-2-3 Charts

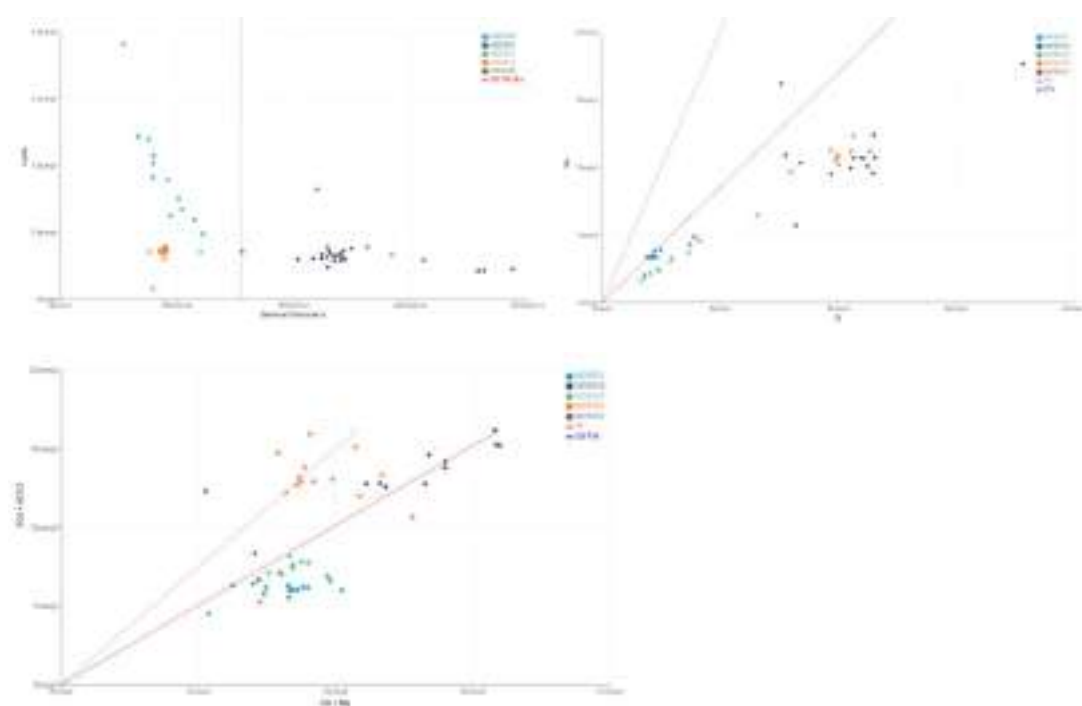


Figure 18 This image represents all ionic diagrams.

Ionic 1/2/3 diagrams are a set of diagrams which enable users to visualize data accordingly to different units of measure and with respect to different guide lines. Ionic 1 plots Electrical Conductivity with respect to the ration between Ca and Na. For this chart a FreshWater/Sea Water guide line has been plotted using the schema depicted on Figure 19. Ionic 2 plots Cl on x-axis and Na on y-axis. In this case there are two guide lines: 1:1 which represents the main bisector and R2 which is computer following the schema depicted in figure 19. Similarly to other charts, Ionic 3 plots on x-axis Ca + Mg and on the y-axis SO₄ + HCO₃. Also in this case there are two guide lines which are 1:1 and SeaWather/FreshWater line which have got same meanings of the previous explained before.

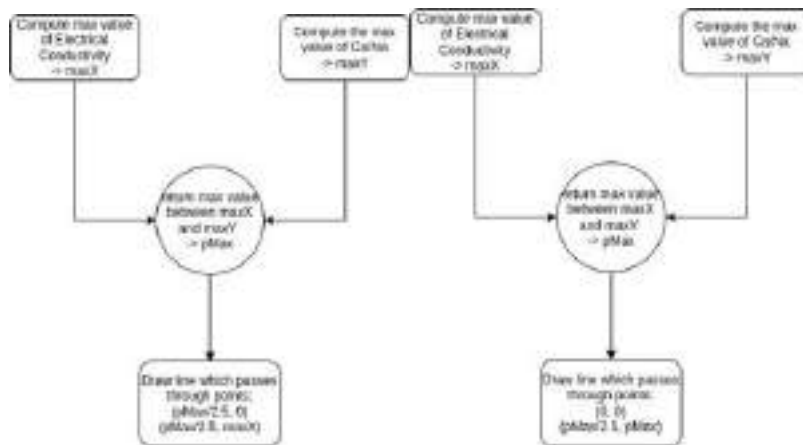
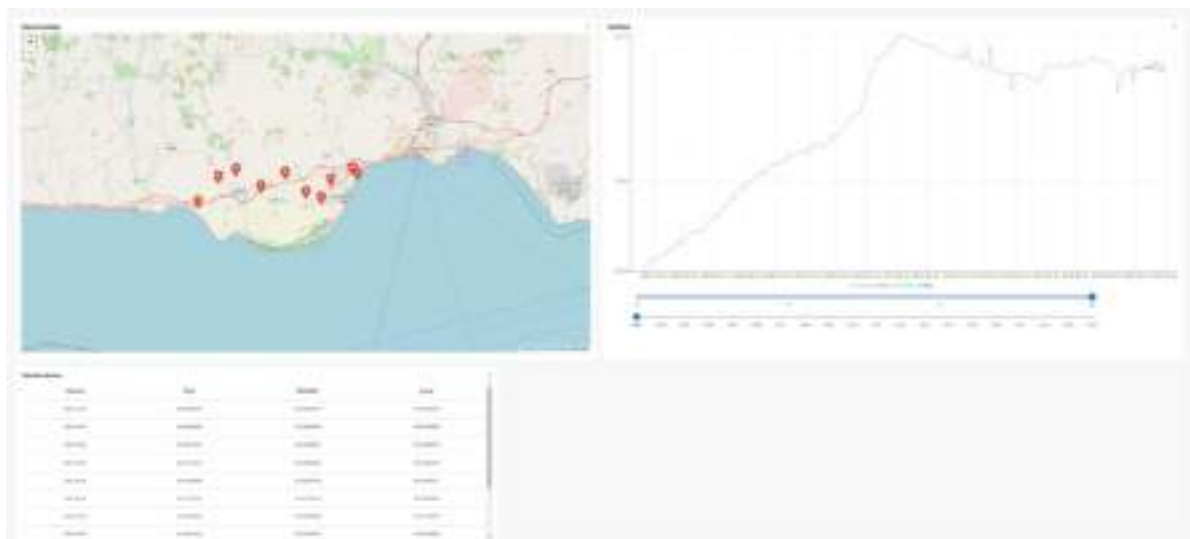


Figure 19 The left image explains the FreshWater/Sea Water guide line schema. The right image explains the schema for two guide lines.

Composed Chart (Multiline, Map, Table)



The chart consists of three components: a map, which displays geo-referenced information, a line-chart, which displays historical information and forecasts, and finally a table, which displays all forecast data.

To obtain the data in the correct format for chart creation, the dataset (a .csv file) is analysed through an Airflow pipeline. This is able to reorganize the position of the rows and columns in the file and make the date format used homogeneous



Figure 21 Composed chart pipeline

The pipeline consists of a task (scriptCetaqua) that exclusively reorganizes the file to make it usable by the composed chart script.

When the final .csv file is obtained, it is inserted into the platform and made available. The graph imports the data through the map and it is able to change the format of the geographic coordinates from UTM (WGS84) to ESPG:4326, projecting points onto the base layer. Each point entered is associated with all the information specific to that area, so when the user clicks on it, it will be sent to the Multiline and Table charts that will display the results.

The Multiline chart provides a view of both historical and predicted data. The user may choose to view a different number of predicted data by editing the first slider provided in the chart itself (by default, all forecasting data are visible). This slider will send the command not only to the Multiline but also to the Table, which instead displays in tabular form only the predicted information.

The second slider allows to users in choosing the period of historical data. Moving the slider backward and forward the initial year for the historical period will be modified accordingly to what is indicated by the slider itself.

[illegible]

This graph shows predictions of hydrochemical data based on the Mann-Kendall methodology.

The calculation of the predictions in this chart, was performed using an Airflow pipeline. The pipeline must be run to obtain the dataset containing the predicted values.

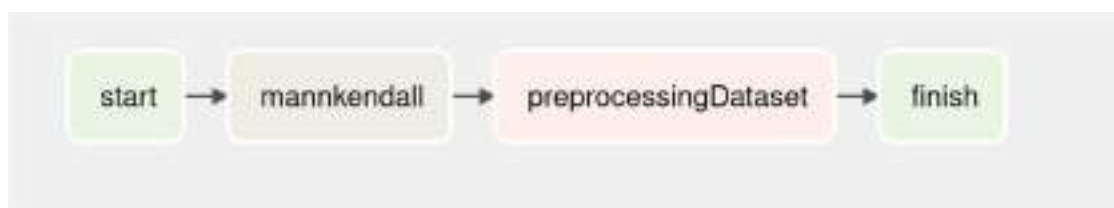


Figure 23 Mann-Kendall chart pipeline

The pipeline consists of an initial task (Mann-Kendall) that aims to open the .csv file containing all the information about the measurements with the values for each variable and a date that characterizes each measurement. Only after that the algorithm is run that allows to calculate the predicted values on that data and save them in another .csv file. Before it can be used and then imported into the platform for creating the charts, the next task (preprocessingDataset) performs a pre-processing of the data obtained to make it of the format used.

2.3.2. General information about charts

Graphs are interactive. This means that users can perform certain actions on it to gather information or increase the meaning of the plotted data. By hovering the mouse over a point, graphs automatically highlight the selected point by placing relevant information within a tooltip.

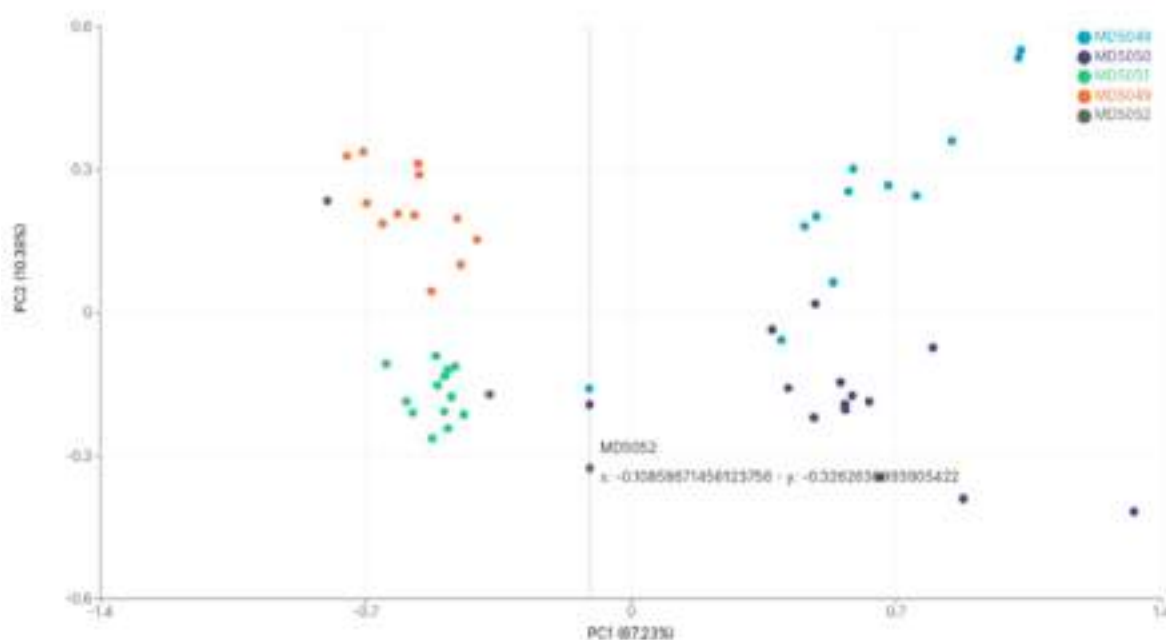


Figure 24 This image explains how tooltip works on the chart

In addition, by clicking on legend items, users would be able to hide/show data points or groups. In this case the points are 'MD5050'.

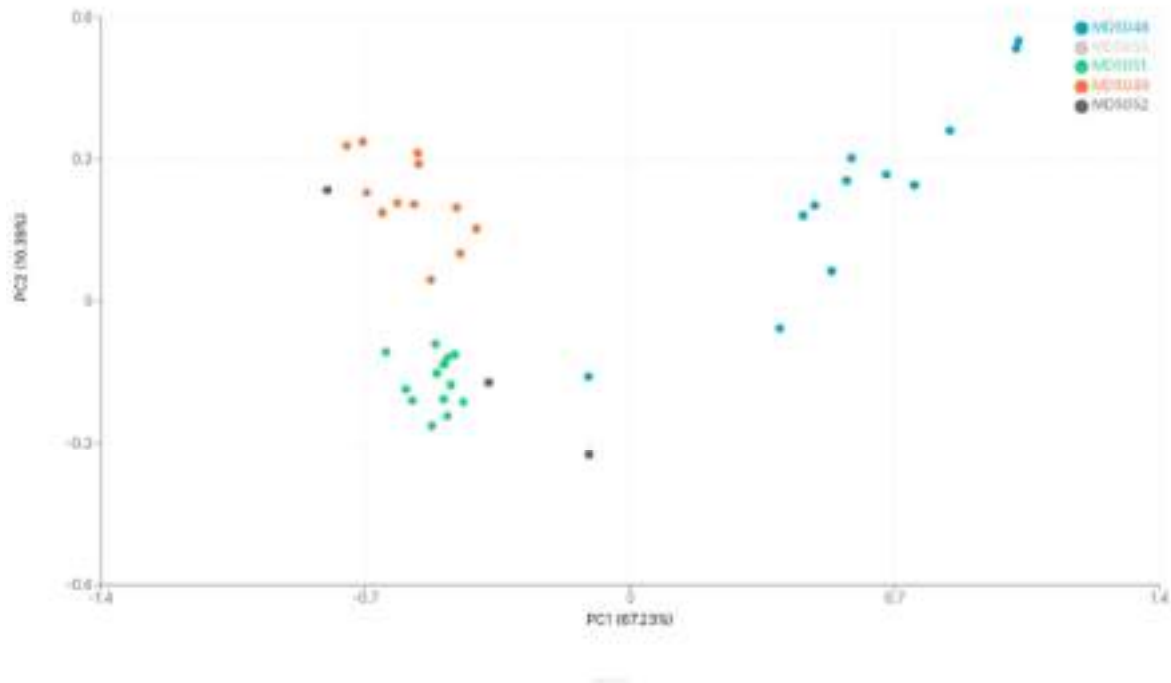


Figure 25 Remove element from legend and chart

When technical users create the chart, they can dynamically choose not only the information needed to create the chart, but also the colour palette using Dashram's built-in left panel, and to improve the explainability of the charts themselves, an info-box has been added to the top of the charts, again with customizable content.

In the chart creation section, after selecting the dataset, there are several information of the same on the left column of the screen, such as the name of all the columns. The user will then choose the type of query to be made and all the various customizations, then executing the query will render the chart.

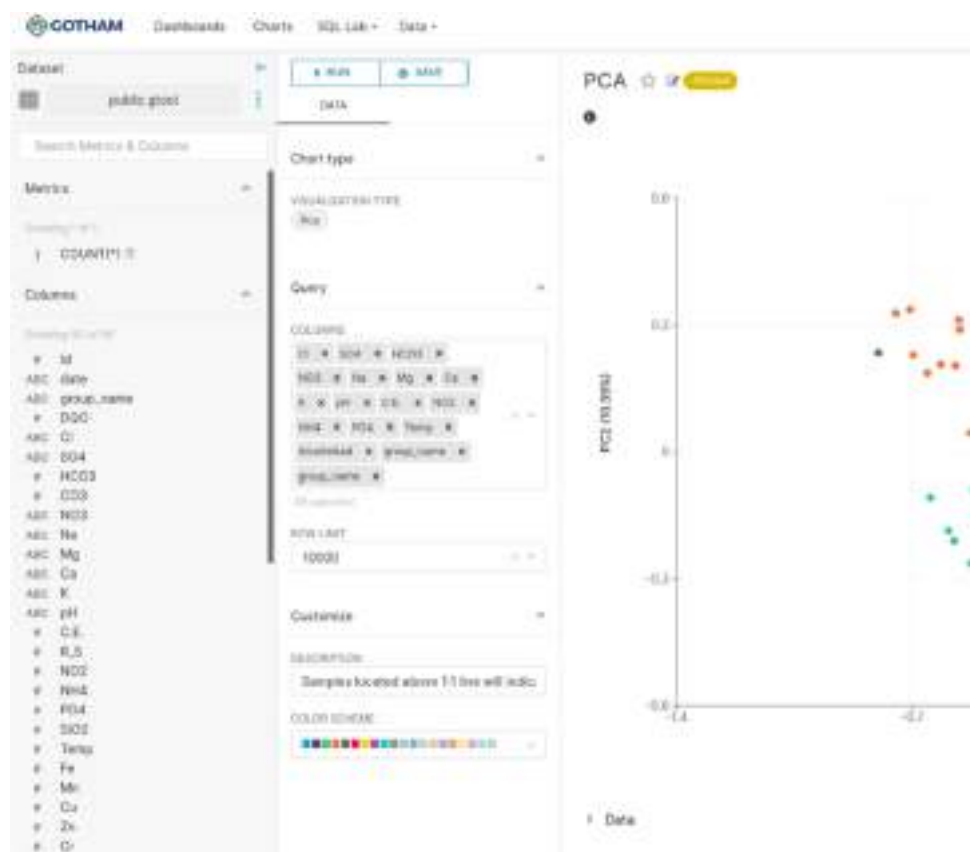


Figure 26 Section for creating charts

2.3.3. Dashboard

In the creation of dashboards, as in the creation of charts, it is possible to distinguish between technical and non-technical users. Dashboards can be created and assigned to a single type of non-technical user with a certain role, and in this way only users with that role can view it.

The user, after the login phase, finds a set of dashboards that can be consulted. Thanks to role management, it's possible to access to the platform to view only a specific list of available dashboards.

After logging in, the user can open a dashboard and interact with all the charts entered within, filter results or view details.

There are three different types of dashboards depending on modules and sub-modules that GTool needs.

Hydrochemical dashboard

Hydrochemical dashboard contains all charts (presented in previous modules) which analyse water

from a chemical point of view. In particular, in this dashboard, could be found: Piper, KMeans, PCA, Wilcox, Ionic 1, Ionic 2, Ionic 3 and Mannkendall diagrams.

Thanks to the flexibility given to charts in development phase, they can be arranged in any order and size.

Figure 27 shows the whole hydrochemical architecture.

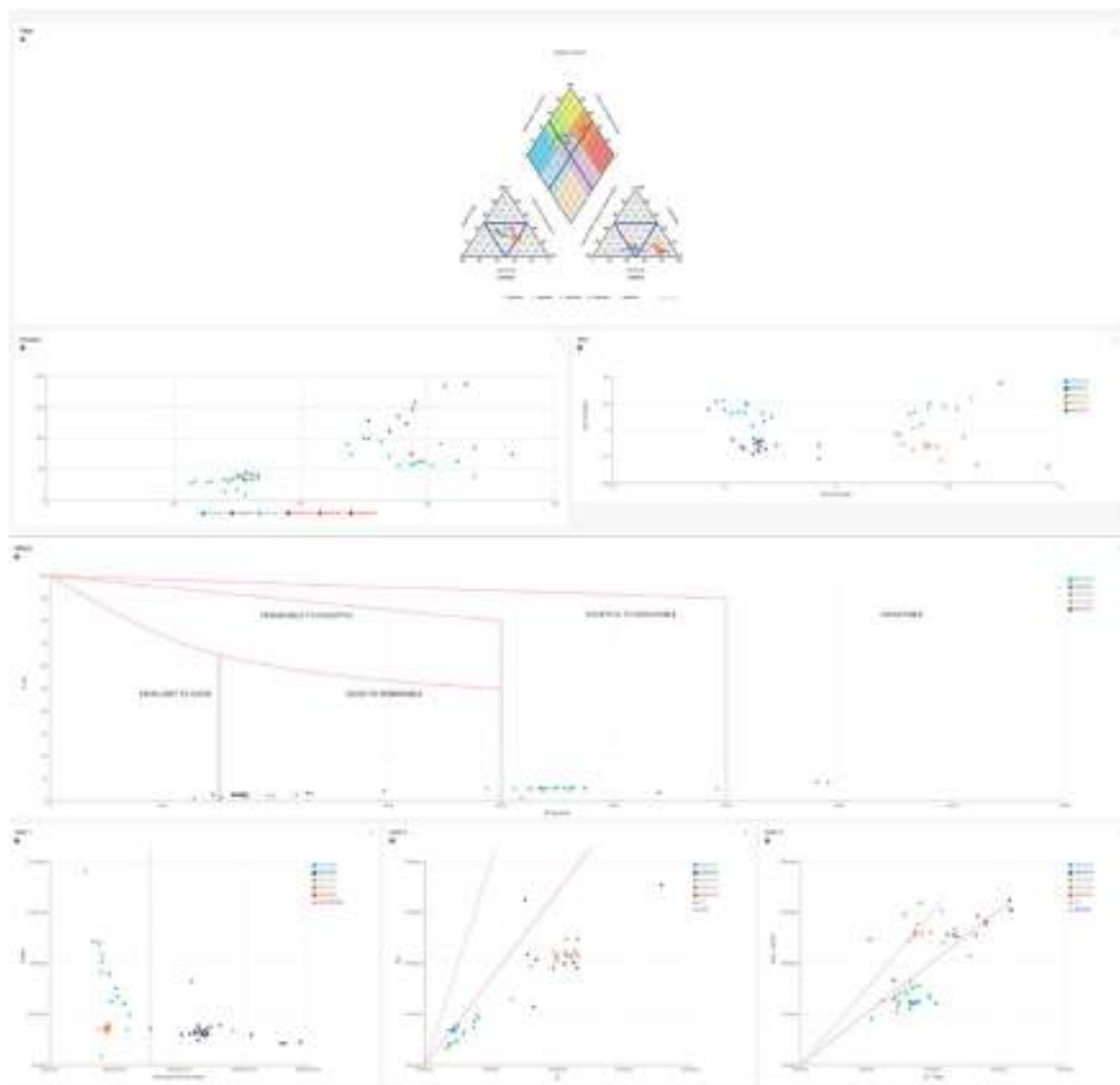


Figure 27 The whole “hydrochemical dashboard”. It contains Piper, PCA, K-Means, Wilcox, Ionic1-2-3 and Mann-Kendall charts.

There is also a mobile version of the “hydrochemical dashboard” which changes mainly in charts disposition. Figure 28 shows the mobile version of the dashboard.

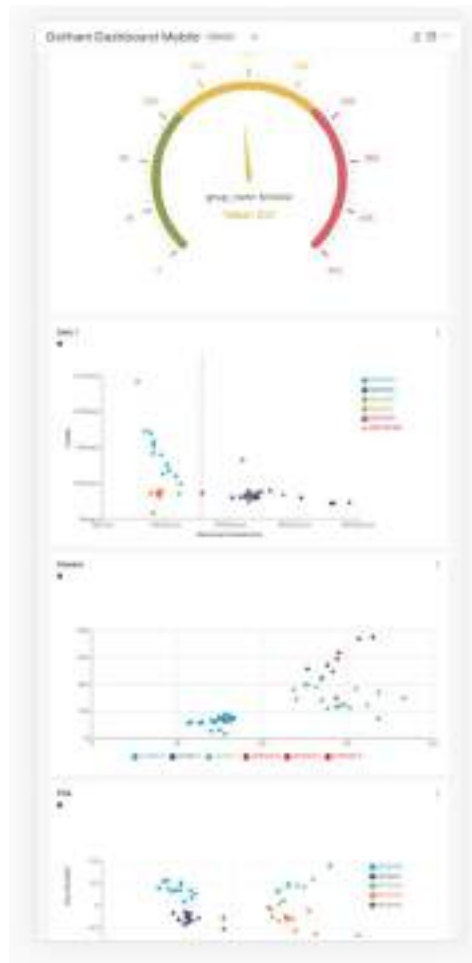


Figure 28 Mobile version of hydrochemical dashboard. It contains less charts than the desktop version; this is due to user-experience motivations. In addition, dashboard differs in disposition of charts themselves respectful to desktop version.

Groundwater potential dashboard

This dashboard is simpler than the hydrochemical one. It contains information about the ground from the water usability perspective. In particular, this analysis, is performed by the raster image chart. For this reason the dashboard, designed just for a desktop version (for usability motivations), contains just one chart. Figure 29 shows groundwater potential dashboard.



Figure 29 Desktop version of the groundwater potential dashboard.

KPIs

The last dashboard groups KPI metrics.

GTool enable technical users to create KPIs for monitoring trends and manage exceeding threshold situations. Different parameters from different modules created can be taken into account and put into a single dashboard to get a complete view.

An example is shown in Figure 30.



Figure 30 KPI dashboard is shown in this image. Dashboard contains just one KPI chart but it can be improved further according to users need.

2.4. APIs

In addition to a set of API made available to retrieve the information obtained from the Data Layer in D4.2., APIs were created to make available the information obtained from the GTool.



Figure 31 List of APIs

The API created for the GTool was designed to return information regarding the forecast values for two modules: 'Groundwater level forecasting' and 'Streamflow & stored water forecasting'.

These modules consist of a set of historical information and forecast values: 'low rainfall', 'middle rainfall' and 'high rainfall'. Each of them refers to a specific area whose coordinates are defined with the UTM format. To request the information, the user must specify which area to query by entering latitude and longitude and define for how many months to display these values.

The predicted values range from a minimum of 1 to a maximum of 12 months; if the user enters a coordinate that is not present or a number of predictions requested that is outside the range, an error message will be returned.

2.4.1. Technical Details

Type	
Programming Language	TypeScript, JavaScript, Python
Additional libraries	Leaflet, Rechart, Pandas, PubSub

2.4.2. Installation

The installation procedure is simplified by the use of Docker technology. In particular, to be able to deploy the entire frontend environment, simply run the docker-compose up command in the folder that contains the docker-compose.yml file.