



UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

DISIT
DISTRIBUTED SYSTEMS
AND INTERNET
TECHNOLOGIES LAB

Dashboard Management System

Dashboard Management System

This tool has been developed in the context
of REPLICATE H2020 project



www.replicate-project.eu



V0.4

DISIT Lab
University of Florence
Department of Information Engineering

REPLICATE has received funding from the European Union's Horizon H2020 Research and Innovation Programme under grant agreement No. 691735



Indice

Sommario

1.	Introduction	5
2.	Dashboard System Manager Features.....	7
3.	Architecture of Dashboard System	10
3.1.	Deploy Notes	11
4.	Installation notes	13
4.1.	Dashboard backend installation notes	13
4.2.	Dashboard frontend installation notes.....	13
4.3.	How to set the data ingestion process	13
4.3.1.	Set the data source	14
4.3.2.	Define a new metric	15
4.3.3.	Define a new widget.....	17
5.	Quick start of the Dashboard Management System.....	23
5.1.	Use Dashboard Management System.....	25
5.1.1.	Login	25
5.1.2.	Creating a new dashboard	26
5.1.3.	Adding your first widget in your dashboard.....	29
5.1.4.	Adding your second widget in your dashboard	31
5.1.5.	Modify your widget.....	34
5.2.	Conclusion	37
6.	Designing and editing Dashboard	39
6.1.	Create a new dashboard.....	40
7.	Designing and Editing Widgets	44
8.	How to embed a dashboard in a HTML external webpage.....	47
8.1.	Dashboard embeddability and relative whitelist.....	47
8.2.	Dashboard embedding options, preview and markup generation.....	47
9.	Available Widgets details.....	50
9.1.	Available widgets on DISIT Lab tool	50
9.2.	The Generic Widget properties.....	51
9.3.	The Widget Detailed information.....	52
9.4.	The Specific Widget properties	53
10.	Details per widget.....	54
10.1.	widgetBarContent	54
10.2.	widgetColumnContent	54
10.3.	widgetEvents	54



10.4.	widgetGaugeChart	56
10.5.	widgetRadarSeries	57
10.6.	widgetGenericContent	59
10.7.	widgetTimeTrendCompare	60
10.8.	widgetPieChart	60
10.9.	widgetPrevMeteo	63
10.10.	widgetSce.....	64
10.11.	widgetSingleContent.....	64
10.12.	widgetSmartDS.....	65
10.13.	widgetSpeedometer.....	65
10.14.	widgetTimeTrend.....	66
10.15.	widgetTrendMentions	66
10.16.	widgetStateRideAtaf	67
10.17.	widgetServiceMap	68
10.18.	widgetButton	68
10.19.	widgetProcess	69
10.20.	widgetProtezioneCivile	70
10.21.	widgetExternalContent	71
10.21.1.	Add a traffic flow map in a widget ExternalContent	72
10.21.2.	Add a WiFi map in a widget ExternalContent	72
10.21.3.	How to find the position coordinates.....	73
10.22.	widgetTable.....	74
10.23.	widgetLineSeries.....	74
10.24.	widgetCurvedLineSeries.....	74
10.25.	widgetBarSeries	75
10.26.	WidgetSelector.....	75
10.26.1.	Add an item to be shown in the Selector widget	77
10.26.2.	Show the data of a pin in another widget.....	82
11.	How to embed a dashboard in external web pages. A step by step guide.	84
12.	Introduction to the “Notificator”	91
12.1.	Architecture of the Notificator tool.....	92
12.2.	System entities and their lifecycle.....	94
12.2.1.	System entities.....	94
12.2.2.	System entities’ lifecycle	94
12.3.	Notification Management System	97
12.3.1.	Login.....	97
12.3.2.	Profiles and Notification pop-up.....	97
12.3.3.	Client applications (TBD)	98



12.3.4.	Events generators management	100
12.3.4.1.	Add a notification	101
12.3.4.2.	Edit a notification	105
12.3.4.3.	Delete a notification	109
12.3.4.4.	Set the validity period.....	109
12.3.5.	Message book.....	113
12.3.5.1.	Create a new message.....	114
12.3.5.2.	Edit a message.....	116
12.3.5.3.	Delete a message.....	118
12.3.6.	Address book	119
12.3.6.1.	Add a new recipient.....	120
12.3.6.2.	Edit a recipient	120
12.3.6.3.	Delete a recipient	121
12.3.7.	Rest book (TBD).....	121
12.3.8.	Events log.....	122
12.3.8.1.	Filter events.....	123
Annex I –	Available metrics	127
Annex II –	Examples of Dashboards	139

What's new in v. 0.4

- Completely renew User Interface of the tool
- Sec. 4.3 *How to set the data ingestion process* added

What's new in v. 0.3.7

- Improved installation notes in Sec. 4
- Updated “Generic Widget Properties” section (Sec. 11.2)
- Updated section on Button widget (Sec. 12.18)
- Updated section on Civil Protection widget (Sec. 12.20)
- Improved Sec. 12.21 on ExternalContent widget

What's new in v. 0.3.5

- New Meteo widget (Sec. 12.9)
- Improved Sec. 12.26 Widget Selector
- Possibility in some widget to select the font for the text in the generic widget properties

1. Introduction

In the development of Smart City there is a great emphasis to the set-up of what is called Smart City Control Room, SCCR. An SCCR is a substantially an area (of one or more rooms) in which all the high-level data are summarized and made accessible for the decision makers and for the city operators.

In large metropolitan cities the SCCR include large panels/monitors (even covering large walls) in which the status of the city is reported presenting the view of the city with some synthesis of data regarding: mobility, energy, social, environment, weather, public transportation, people flow, health, water, security, ICT, governmental, first aid, civil protection, police (118/112), and thus almost all the city resources. Most of the City resources are in effect network of resources deployed in the city, and thus some of them are considered critical infrastructures for the city functionality and life of city users such as: transportation, energy, security, health, water, civil protection, etc.

Typically, in medium sized cities, the daily management of city resources is performed by a set of separate city operators such as: energy operator, water providers, several telecom operators, several ICT infrastructures for the different level of govern to provide different services, civil protection, etc. This means that all of these separate city operators are autonomously managing their control room, with access at their own data to take their own and limited decisions. For example, when the energy network has a problem in an area of the city the energy is rerouted to reach the area via a different path, when the water network has a problem valves on tubes are controlled and water is provided in other means, in presence of traffic congestion the red light timing is acted to facilitate the flow and bus paths may be changed/rescheduled.

The presence of a set of independent control rooms may be positive to be reactive and solve microproblems. On the other hand, in presence of relevant disasters the distributed solution is not efficient and effective as it should do for the lack of concertation and synchronization of knowledge and actions. In addition, in some cases, cascade effect may be provoked. For example, a water flooding may provoke lack of energy and communication, the lack of garbage collection may provoke health problem, the lack of water quality may provoke health problems, etc.

To cope with this kind of problems, in many countries, for the small and medium size cities an integrated control room, a sort of situation emergency room is formally foreseeing and automatically created when critical situations are detected.

In that case, representative people and devices, control panels, of each city operators are moved and activated in a single place, even if for a temporary period. This approach reduces the communication time, and reduce the misunderstanding among operators, while does not solve all the lack of synchronization among the city operators, and the lack of integrated views for the city data. As put in evidence before, most of the critical events in the city impact on multiple infrastructures, and may provoke cascade effects. Thus, the single city operator is not in the condition to solve the problem, and in most cases, neither to identify the cause of the detected problem. This problem can be solved by activating cross information among the several control rooms of the city operators, and better to have a stable SCCR.

One identified and understood the need of having an integrated SCCR, immediately growth up the issue regarding what has to be shown on (i) the panels on walls and what on (ii) computers of the operators. These approaches are very complex to be managed since the amount of information is very at wide range and has to be easily understood by the observers of the panels on the wall and of the computers. It is not only a problem of usability is a problem of understandability, a problem of data representation, a problem of competence for the observers, the operators, the decision makers, etc.

In most cases, they have to be trained to understand the graphics representation. They have to become confident on what they see to understand in deep all the single details represented on the screen. For example, all of us we are used to understand: (i) a traffic representation observing the city map with red, orange, yellow, and green



segments on the streets; (ii) a temperature and the humidity percentage, etc., while it is more difficult to read the pollution, pollination, etc. Alarm signals in red, blinking signal, etc., may help on learning and on understanding.

From the technical point of view, the tools for rendering information on SCCR are typically called Dashboard. The Dashboards should be capable to present real-time data in several different manners with real time updates autonomously H24 7/7 days. Dashboard for control rooms should not be confused with business intelligence tools that produce graphics from the combination of data extracted from some sources (database, files, API, etc.).

The Dashboard Builder is a tool that allow to produce multiple dashboard accessing at static and real-time data from some data store and services.

This manual is particularly addressed for users that are interested to design and edit dashboards on the basis of yet defined metrics and widget. [Section 4](#) explains how to install and configure the tool, how to create new datasources, new metrics and widgets.

If you want to test a demo account in the DISITL Lab tool (<https://dashboard.km4city.org/management/index.php>) please, send an email to paolo.nesi@unifi.it to ask a username and password.

2. Dashboard System Manager Features

The following is a list of features covered by the dashboard builder. In particular, the dashboard builder is able to:

- show graphics by using several different kinds of visual paradigms, and parameters;
- be shown on control room screens H24, 7 days per week;
- enable the visualization to specific users even if they are not user of the Control Room. Visualize dashboard to public or limited in access to a number of qualified users. This would allow creating Dashboard for general public, and versions with sensitive data for the Civil Protection, police 118, etc.
- produce dashboards of different size (e.g., PC to mobile, large screen, multiscreen) with or without main header and frame. Please note that responsiveness is not so relevant if a Dashboard is designed to stay H24/7, since to be responsive from UHD to small mobile (e.g., it would be almost impossible to have the same user experience at all the resolutions). If the aim of the Dashboard consists in being a monitoring tool, it has to be firstly efficient in that activity and secondly in the interactivity and responsiveness;
- show data coming from some data lake, virtual store; thus, data taken from different **Data Sources**: SQL database, database via ODBC, noSQL database (for example MongoDB, Hbase), HDFS directly or via Phoenix, RDF stores via SPARQL, from API calls and from streaming data.
- Be composed as a set of Widgets of different size, allocated on a grid, by using a visual combination of Widgets;
- collect data and make replay of the data history from a given date and time. This feature is adopted to play back eventual critical condition and shown data;

The dashboard builder allows also to:

- set and change parameters as: size, color of the background, font of the header, color of the header, etc.;
 - create and editing Dashboard produced by one or more users;
 - edit controlling each single Dashboard by a single person;
 - compose a set of possible views (in most cases called widgets), by visual interaction, visual composition of widgets;
 - control size and zoom of the Widgets;
 - allow each single dashboard to be viewed/accessed in reading by one or more specific people;
- be cloned to derive a new dashboard from a former one and perform versioning.

Dashboard Widget is capable to show data according to different paradigms, graphics models (for showing and collecting data):

- Classic graphic Widgets:
 - tables (single and multiple columns and rows);
 - graphs (single and multiple values), trend compare (hours, week, day, 12 hours, a month, an year);
 - pies diagrams, simple and multiple;
 - kiviati: simple and multiple;
 - direct values with unit of measure: a tile with the temperature, a tile with the number of accesses in the last 10 minutes;
 - histograms with multiple values/bars, vertical and horizontal, different colors;
 - gauge: round bar with multiple colors and firing conditions;
 - tachymeter: with multiple colors and firing conditions;
 - images, photos, etc.;
- Map based Widgets:
 - One or more elements on map with their: paths, areas, position, etc.
 - Position of public busses, position, lines, stops, status, forecast, etc.;
 - Parking status, position, status and predictions;
 - Panels for Variable Messages: position, status text;

- Recharge stations, position and status of the recharge;
- Process monitoring from the DISCES scheduler [<http://www.disit.org/6746>], Distributed Smart City Engine Scheduler
- Web site monitoring status verifying a textual pattern into a Web Page file;
- Wi-Fi status in real time, and clustering of trends;
- Heat maps for the most requested/visited places;
- maps with controlled/added pin, POI (using OSM and/or Google Map or ServiceMap),
- traffic flow densities, Traffic flow reconstruction;
- real time position of busses and/or other vehicles;
- **Stream Widgets (under development based on Nodejs UI)**
 - Connecting streams to graph widgets for trends, values, etc.
- **Special Widgets:**
 - List of Tweets, list of Hashtags on Tweets, List of Citations on Tweets;
 - Weather forecast, pollution, pollination;
 - Special list of events of different kinds: traffic, civil protection, health, etc.;
 - List from Enterprise Service Bus, with event classifications;
 - First Aid data, status for several different hospitals, overview and real-time status details per hospital;
 - TV camera for monitoring the city and security, according to different protocols;
 - Smart DS, Italian flag, as metrics taken from some source or from SmartDS solution and tools;
 - Energy consumed, energy saved/stored, etc.;
 - DISCES Scheduler Status, from smart city or smart cloud infrastructure;
 - Multiple metrics on the same block;
 - Service Level Agreement for Public transportation;
- **Nested Dashboard**
 - Containing a produced Dashboard into a Widget to create a more complex dashboard with multiple view that can be activated by a single call on a large or ultra large screen.
- **External web pages as IFRAMES, third party**
 - web pages or third party tools;
- Be flexible being capable to change colors, fonts of the text, details of the graphics elements
- **Actuator Widgets:**
 - buttons, sliders, commutators,
 - text box,
 - switch on/off, percentage of light (for collecting data);
- **Provide and present for all Widgets**
 - associate with one or more **Data Sources** to extract specific **Metric**;
 - a button (i) for getting information about the specific Widget;
 - a micro button (i) for each metric on tables for providing a popup on which the user can read the description of the data listed/shown;
 - the possibility of associating to each element of the Widget a call to an URL and thus to open a web page;
 - refresh the data according to a periodic time by setting it per widget, coherently with the refresh setting/value of the corresponding metric, with different time interval with respect to the other widget of the same Dashboard. For example, a temperature is read every X seconds, while the corresponding Dashboard has been set up to request an update every Y minutes;
 - host and thus to create new widgets by programming, for example in PHP, Java, Python, may be using third party libraries on it for: TV camera, graphics, execution, stream...;
 - provide the evidence of the periodic update of data and of the fact that the information is changing. This requirement may be satisfied by providing period updated, thus reloading of the data and may be showing some animation that are remarking when the data is updated;
 - show some evidence about the data shown, value, date, thresholds, etc.; at the mouse passage
- **embed the Widget** (as the single Widget of a certain Dashboard), into a third-party web page;
- communicate with other widgets in the same Dashboard (**Cross Widget Paradigm**), for example:



- at the push of a widget button to activate a different graph in another Widget of the same dashboard;
- at the click on some list of events, to activate the rendering on map of the event plus related data;
- at the click of a button to activate multiple cross widget events. For example, to change the content of one or more widgets, to send an API REST Call, to send commands to other widgets (change TV camera, change parking under monitoring, to change the monitored sensor, etc...)
- At the click of a button/widget to open an URL on a IFRAME widget
- allow to set-up of one or more **reference firing rules** on the data rendered and visualized by the widget. The firing rule may provoke:
 - evidence in the corresponding Widget. For example, in a graph as the line for threshold, in the box with number with some red blinking, in gauge as a bar, etc.;
 - highlight and blinking the widget when the firing condition become true and off when return false;
- send notifications to several different targets by using a range of mechanism such as: email, SMS, API rest call. The notification has to be means of the **Notificator** service component.

The Metrics are specific selections/queries performed on some Data Source, and:

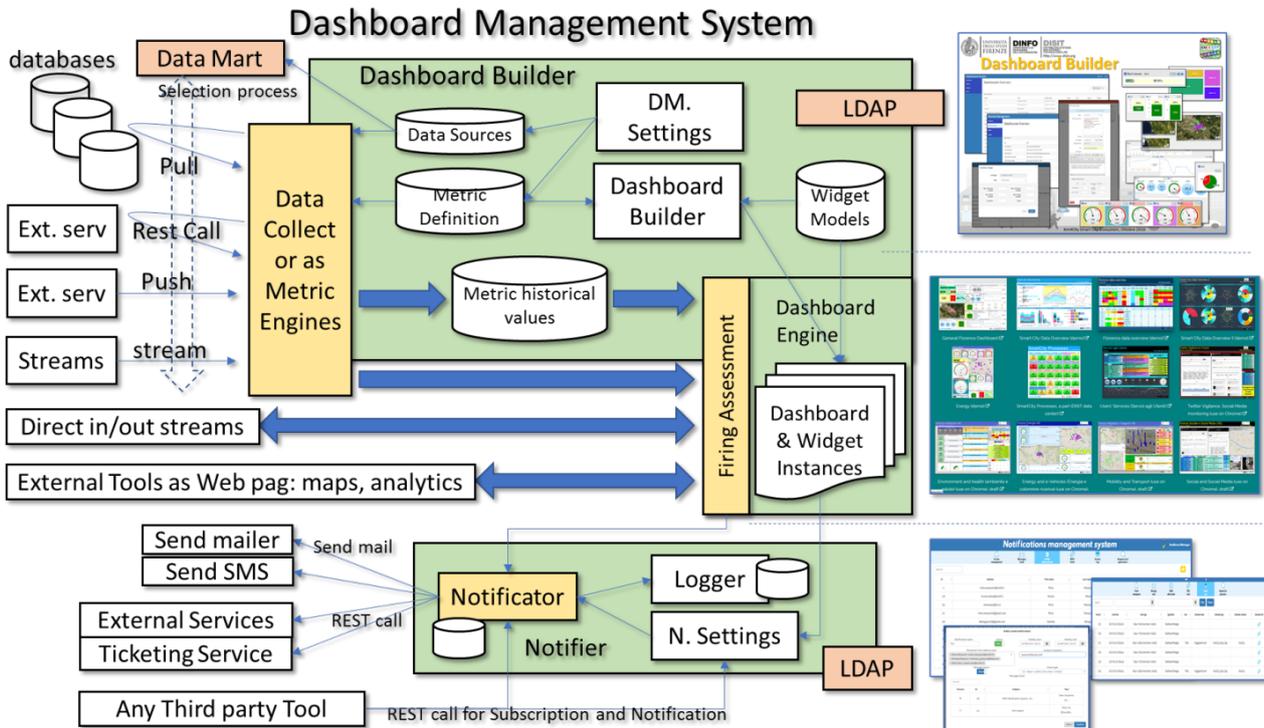
- allow to take simple as well as complex data from some Data Source: SQL database, database via ODBC, noSQL database (for example MongoDB, Hbase), HDFS directly or via Phoenix, RDF stores via SPARQL, from API calls and from streaming data.
- can be programmed to be refreshed, recollected with a given frequency.
- Has to be collected autonomously even if the SCCR Dashboard in not shown in each time and day.
- Collect data which can be saved or not on the Dashboard server and database. Data collected and saved from metrics can be simple (float, integer, string, date/time, etc.) but also complex data in JSON or other formats. As well as data coming from some data lake, virtual store;

A **Notificator** is able to:

- Be capable to receive registration for new Widget, events, values trend, by any external applications;
- Log any kind of event sent, why, when to who, etc.;
- Pass the notification to a tool for managing tickets of intervention to an external tool;

3. Architecture of Dashboard System

In the following figure, the general architecture of the Dashboard solution is presented.



According to the architecture the users can use the:

According to the above architecture the users can use the:

- **Data Mart** for browsing and drilling down into Data Sources (SQL, noSQL, RDF, API etc.). To this end we suggest using:
 - Zeppelin for data base browsing in SQL (ODBC, JDBC). It is also capable to accept script, to set up alerts, and top create simple graphic diagrams: pie, trend, histograms, etc.
 - FLINT for SPARQL interface, for example on RDF stores;
 - ServiceMap (<http://servicemap.disit.org>) for knowledge base browsing over the city data as Km4City knowledge base, which is RDF store as well;
 - LOG (<http://log.disit.org>) for browsing into any RDF store by using SPARQL and visual interface. This tool allows you to browse all the RDF stores accessible in the world which provide a public end point for SPARQL queries, from dbPedia, to Europeana, Geonames, Km4City, Camera, Senato, Getty, etc.
- **Dashboard Builder** (<http://www.disit.org/dashboardSmartCity/management/>) for:
 - Setting up **Data Sources**: IP address, protocol, user name and password for accessing at each specific the data source.
 - Setting up **Metrics** on the basis of accesses at the Data Sources. Metrics are defined in terms of queries/call which have been identified and finalized by using the DataMart set of tools according to the different kind of data stores.
 - Load new **Widget** that can be created manually by using PHP. A Widget may render/exploit the same Metric by means of different graphic models. For example, a temperature read every 5 minutes, can be visualized as current value on a thermometer, while the trend in the last 24 hour, last week, last month, etc. can be show on a graph.
 - Compose **Dashboard**, taking interested **Metrics** and associating them to one or more **Widget**. **Thus**, composing the **Dashboard** by placing the Widgets into the Dashboard frame. The Dashboards are created by using the visual interface of the **Dashboard Builder**.



- Set up firing condition/criteria (computed in real time by **Firing Assessment**) for each Metric of each Widget instance in each defined Dashboard, with the aim of computing criteria for sending notifications. For example, when Metric M is adopted in Widget W of Dashboard D, the certain criteria is saved and controlled (M,W,D,C). One or more criteria can be defined for each (M,W,D, C1....Cc), each of them may produce multiple Notifications, N: (M,W,D, C1 (N1,....., Nn),....., Cc(...)).
- **Notifier to**
 - Accept registration of all the above mentioned (M,W,D,C), to enable the reception of *Notification Requests*, that are used to send Notification according to different approaches.
 - Accept registration by Any Third-Party Tool, in addition to those of the Dashboard Builder, to send alarms about the firing of the registered condition.
 - The notifications can provoke emails, SMS, REST call, as well as the activation of maintenance ticketing system.
 - Log all the registration and the notification of *Notification Requests*.
 - Maintain and use a list of Notification recipient, that are the users which are going to receive the notifications. This list of uses is just listed as: name, surname, email, telephone (if any), role, organization.
- **LDAP for registering:** all the above tools provides an authentication process based on LDAP.

Note that: LDAP and Data Mart are realized as external third party open source tools.

In this view, there are two autonomous processes are active on the Dashboard solution:

- The **Data Collector as Metric Engines**, one or more processes dedicated to collect data from the several Data Sources. The collected data can be (i) saved into a data base of **Metric Historical Values**, or can be (ii) directly accessed from Widget/Dashboard for their visualization.
- The **Firing Assessment** engine, that on the basis of the firing condition imposed for each metric of each widget of each dashboard (active) is capable to verify the condition and in the cases of assertive value, a **Notification Request** is sent to the Notificator. Please note that the presence of the Firing Assessment allows to compute the firing condition on all the data:
 - on Metric Historical Values taken in Push from the data base of the builder,
 - on the streams directly arriving from the Data Collector;
 - from the data stream arriving directly from outside,
 - From the external tools embedded as IFRAME into widget if there is some integration.
- The **Notificator**, one or more processes that compute/verify the firing conditions (saved in their N.Settings) and produce notifications. These notifications are also logged for further analysis and security evidence. The notification process has to work even without any activation or view of **Dashboard Instances**. *Moreover, the Notificator has to provide the reference values of the firing conditions and the event of firing also the Widgets and Metrics since they have to be visualized on the Dashboard as well.*

As a conclusion, **Dashboard Instances** are available for view and activate corresponding widgets according to their Settings. The saving of data into the database of **Metric Historical Values**, allows keeping track of what has been visualized/monitored and thus enabling the replay of data logged. On the other hand, it is also possible to adopt widgets that (i) directly show/provide the data from in/out streams, respectively (for example, Civil protection alert status, etc.); (ii) directly render/visualize web page segments into the Dashboard (for example for showing social media analytics, traffic flow reconstruction tools).

3.1. Deploy Notes

The Dashboard solution can be deployed on:

- a **single VM** with all inside as well as on multiple VMs.
- A very large solution is obtained by using:
 - One or more VM the **Data Collector as Metric Engines** that could write on a common distributed database for **Metric Historical Values** which can be an SQL like or HDFS with Phoenix



- One or more VM the **Notificator** that could write on a common distributed database for **Metric Historical Values** which can be an SQL like or HDFS with Phoenix for logging events. Typically, a classic SQL can be enough.
- One VM for the **Dashboard Management System** for creating new **Dashboard**, many users on LDAP can share the same solution.
- One or more VM with **Dashboard Control Engine** even in balance on the front end, which provide access to the Dashboards.

A large range of intermediate solutions are viable as well according to your number of user accessing to the Dashboards, and to the number of computed Metrics.

4. Installation notes

Dashboard Builder connected to Km4City smart city platform, developed per REPLICATE H2020

The dashboard builder is made of two main components:

1. the front-end, a PHP web application with a MySQL DB;
2. the back-end, a java console application that is running in the back to collect data from the datasources to be put in the front-end DB

Instructions to install and configure each component can be found below.

4.1. Dashboard backend installation notes

This component is a java console application that reads the metrics definitions in the "Descriptions" table, performs the query on the data source and it inserts the data in the "Data" table.

Requirements

- java7 or above, running on linux or windows environments.

Build

- build the jar file using a java development environment (e.g. NetBeans)

Import DB tables

- create a schema "Dashboard"
- import in the MySQL DB the tables defined in the dashboard.sql file using for example "mysql -u user -p password -D Dashboard < dashboard.sql"
- the dashboard user admin/admin is predefined and it allows to login in the front-end

Run & Configure

- before starting the process for the first time add at least one datasource and one metric using the dashboard frontend tool.
- To run the process use "java -jar DashboardProcess.jar" in a folder where is present the config.properties file where are present the credentials to access to the DB.
- an example of config.properties file is available in the root folder, it should be modified at least to setup the url to the DB as well as the user and password.

Notes

- currently the process query for the metric definitions only at start, so if a new metric is added using the front-end the process need to be restarted.

4.2. Dashboard frontend installation notes

This component is a PHP web application

Requirements

- PHP 5.7 (not tested with PHP 7)
- MySQL 5.5 or above

Install

- copy the front_end folder in a web accessible directory (i.e. /var/www/html) and rename it as you like

Import DB tables

- see details in the dashboard_backend component

Configure

- edit the config.php file at least to set the host and credentials for the DB.
- change the permissions to the img folder to allow the webserver to write in this folder. (e.g. "chown -R www-data.www-data img")

4.3. How to set the data ingestion process

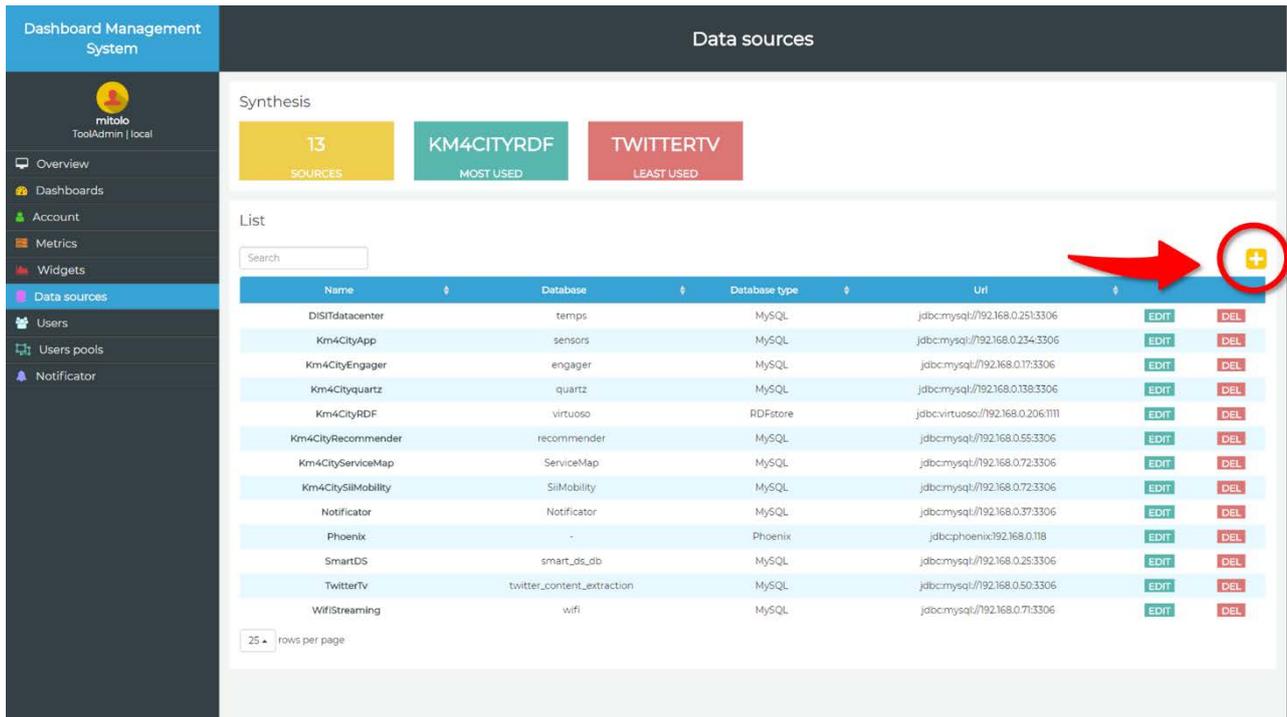
To use the dashboard data ingestion process, an external data base (not provided with the dashboard system) is needed, having some data that you want to regularly query to produce a metric to evaluate. So that, the process has to follow the following steps:

1. set the data source(s)

2. define the metric(s)
3. define specific widget, if needed
4. create the dashboard.

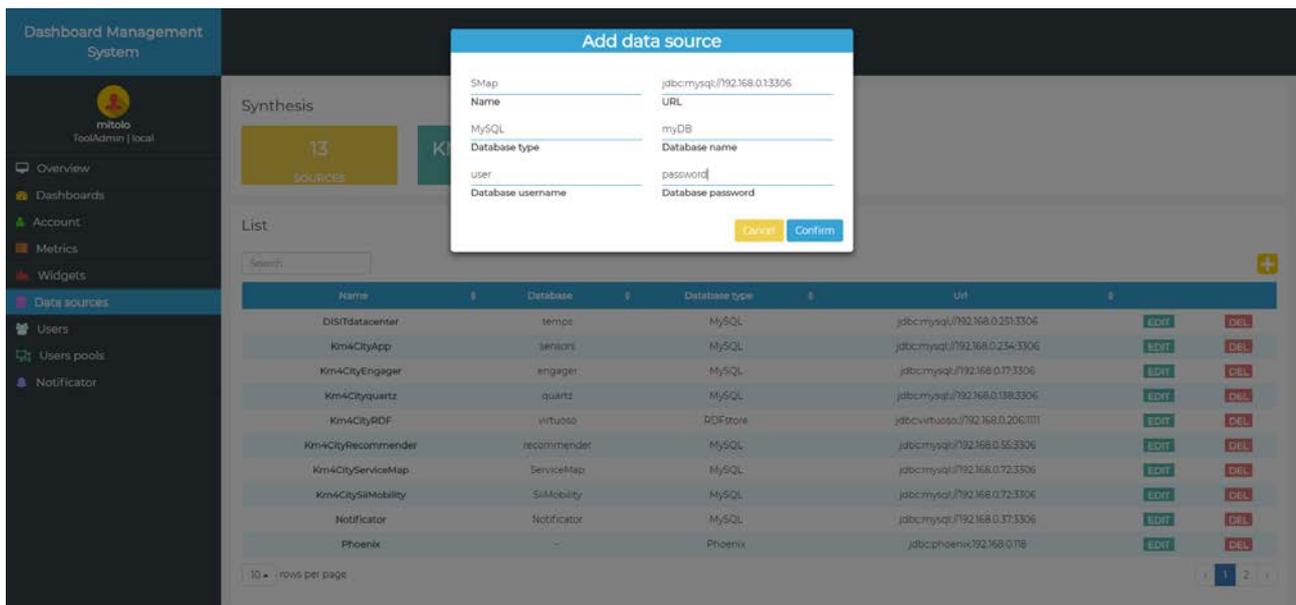
4.3.1. Set the data source

Assuming to have a mysql DB at 192.168.0.1 with database named "myDB" accessible with user "user" and password "password", first of all to define in the dashboard system this "datasource" is needed. This can be done using the front-end clicking the "Data sources" item on the left menu column. Click the  button on the top right to set a new datasource



Name	Database	Database type	Url		
DISITdatacenter	temps	MySQL	jdbc:mysql://192.168.0.251:3306	EDIT	DEL
Km4CityApp	sensors	MySQL	jdbc:mysql://192.168.0.234:3306	EDIT	DEL
Km4CityEngager	engager	MySQL	jdbc:mysql://192.168.0.17:3306	EDIT	DEL
Km4Cityquartz	quartz	MySQL	jdbc:mysql://192.168.0.138:3306	EDIT	DEL
Km4CityRDF	virtuoso	RDFstore	jdbc:virtuoso://192.168.0.206:1111	EDIT	DEL
Km4CityRecommender	recommender	MySQL	jdbc:mysql://192.168.0.55:3306	EDIT	DEL
Km4CityServiceMap	ServiceMap	MySQL	jdbc:mysql://192.168.0.72:3306	EDIT	DEL
Km4CitySiIMobility	SiIMobility	MySQL	jdbc:mysql://192.168.0.72:3306	EDIT	DEL
Notificator	Notificator	MySQL	jdbc:mysql://192.168.0.37:3306	EDIT	DEL
Phoenix	-	Phoenix	jdbc:phoenix:192.168.0.118	EDIT	DEL
SmartDS	smart_ds_db	MySQL	jdbc:mysql://192.168.0.25:3306	EDIT	DEL
TwitterTv	twitter_content_extraction	MySQL	jdbc:mysql://192.168.0.50:3306	EDIT	DEL
WifiStreaming	wifi	MySQL	jdbc:mysql://192.168.0.71:3306	EDIT	DEL

A new pop-up will be opened to be filled as shown in the following picture.



Name	Database	Database type	Url		
DISITdatacenter	temps	MySQL	jdbc:mysql://192.168.0.251:3306	EDIT	DEL
Km4CityApp	sensors	MySQL	jdbc:mysql://192.168.0.234:3306	EDIT	DEL
Km4CityEngager	engager	MySQL	jdbc:mysql://192.168.0.17:3306	EDIT	DEL
Km4Cityquartz	quartz	MySQL	jdbc:mysql://192.168.0.138:3306	EDIT	DEL
Km4CityRDF	virtuoso	RDFstore	jdbc:virtuoso://192.168.0.206:1111	EDIT	DEL
Km4CityRecommender	recommender	MySQL	jdbc:mysql://192.168.0.55:3306	EDIT	DEL
Km4CityServiceMap	ServiceMap	MySQL	jdbc:mysql://192.168.0.72:3306	EDIT	DEL
Km4CitySiIMobility	SiIMobility	MySQL	jdbc:mysql://192.168.0.72:3306	EDIT	DEL
Notificator	Notificator	MySQL	jdbc:mysql://192.168.0.37:3306	EDIT	DEL
Phoenix	-	Phoenix	jdbc:phoenix:192.168.0.118	EDIT	DEL

Click Confirm to save the new data source. A confirmation message will be shown.



If the data filled in the form are wrong, an error message will be shown.



Click the  button again to open the form, check and correct the data.

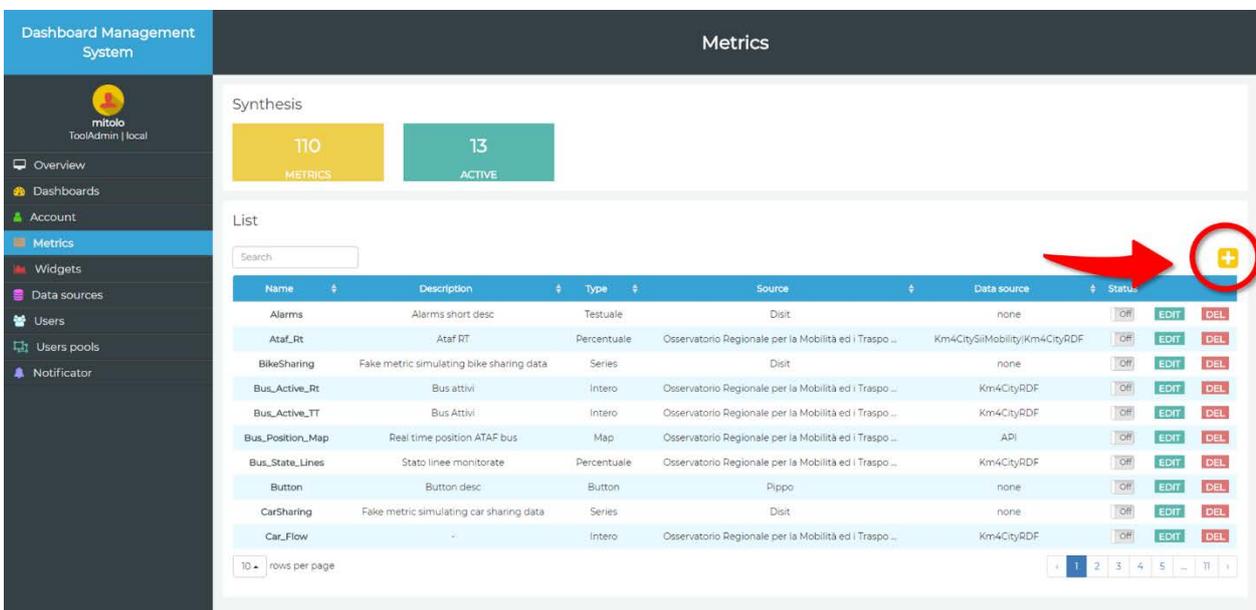
The following are the fields to be filled:

These parameters are used by the data ingestion process that import the data and are necessary to connect to the data source of every active metrics.

- *Name*: unique identifier of the data source
- *URL*: the complete URL of the server to call. This has to include eventual parameters (e.g. protocols, port, specific parameters, etc.)
- *Database type*: e.g. MySQL, POSTGRES, SQLite, etc.
- *Database name*: name of instance in the schema
- *Database username*
- *Database password*

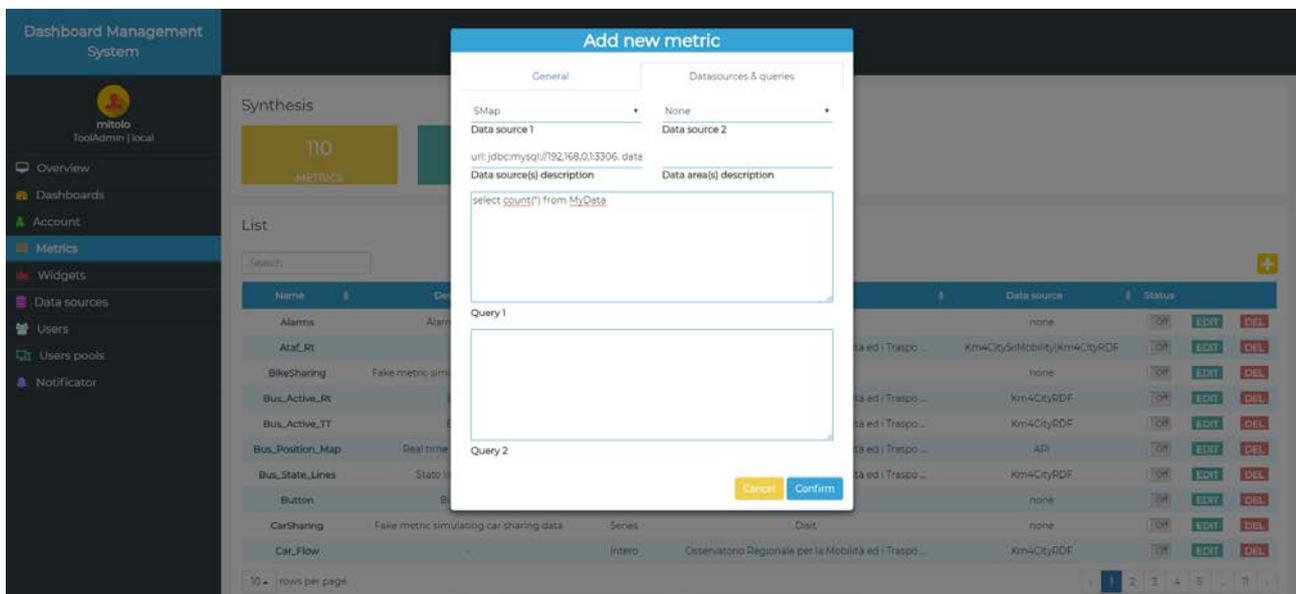
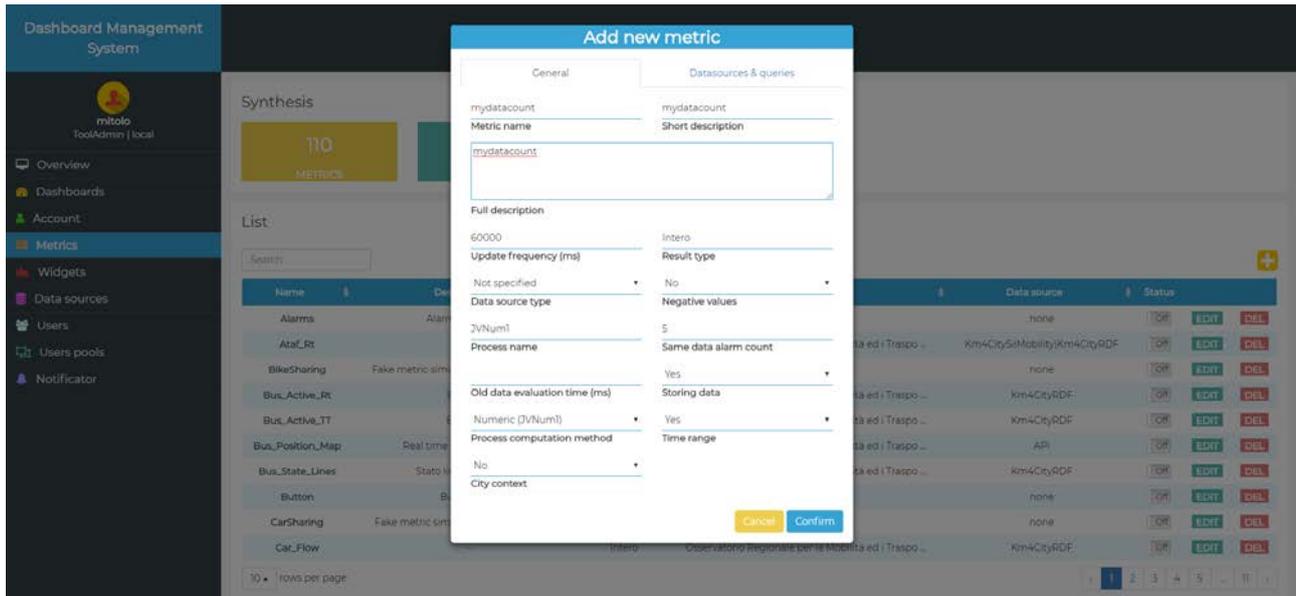
4.3.2. Define a new metric

After that, it is necessary to add a new metric. Click the Metrics items in the left menu column.



Click the  button on the top right to add a new metric.

Assuming to have a table "MyData" in the myDB database where almost every 10 minutes a new row is added with some data, to define a metric with the current row count, the following are the two tabs to be completed to define a metric.



Click Confirm to save the new data source. A confirmation message will be shown.

The following are the fields to be filled:

General tab:

- *Metric Name*: unique identifier of the metric
- *Short description*: brief description (not mandatory)
- *Full description*: not mandatory
- *Update frequency (ms)*: interval in ms that represents each time the new data is downloaded for that metric. It is the update interval.
- *Result type*: type of data produced by the metric (the following are the valid possible values: 'Testuale', 'Percentuale', 'Series', 'Intero', 'Map', 'Button', 'Float', 'Actuator', 'Separator', 'isAlive')
- *Data source type*: type of data source
- *Negative values*: set to YES if the metric generates negative numeric values.
- *Process name*: Name of the process that elaborates the metric. It is a free field.
- *Same data alarm count*: number of consecutive times with constant data beyond which an alert is triggered for constant data too many times

- *Old data evaluation time*: the time after which the most recent data acquired for this metric is considered too old and therefore a too old data alert is triggered
- *Storing data*: Set to YES if the metric stores the data.
- *Process computation method*: method that execute the data import in the process.
- *Time range*:
- *City context*: Set to YES if the metrics has a parameter referred to a municipality (for example the meteo widget allows to set the municipality).

Datasources & queries Tab

- *Data source 1 and 2*: indicates one or two of the data source available. Pointing to a DB that resides on a PC.
- *Data source description*: optional
- *Data area description*: informal description of the relevance area of data sources
- *Queries 1 and 2*: The queries that are executed on the datasource to find the most updated data of the metric.

Once the metric has been created, it is possible to start the java process, and, if all is ok, it will begin to populate the "dashboard.Data" table. So that it is possible to define a new dashboard with a widget showing the acquired value, as explained in the following sections of this manual.

In case of problems in editing the metric, it is possible to modify the Descriptions table directly from the db (the UI for adding and editing a metric will be modified in the next version). Please. remember to restart the java process after every metric update.

In this version there are some predefined metrics that are placeholders (e.g. Button, clock, separator) they are not used for data ingestion and allow using some widgets that are not directly related with external data. Using these predefined "metrics" you can define a dashboard even without a running java process.

Please note that the relation of a metric with a widget type is defined in the Widgets table (that is preloaded), the widgetType column define which types of metrics the widget can render (e.g Intero, Float, Testuale, Series, Percentuale, etc.) moreover the unique_metric columns allows to bind a widget to a specific metric.

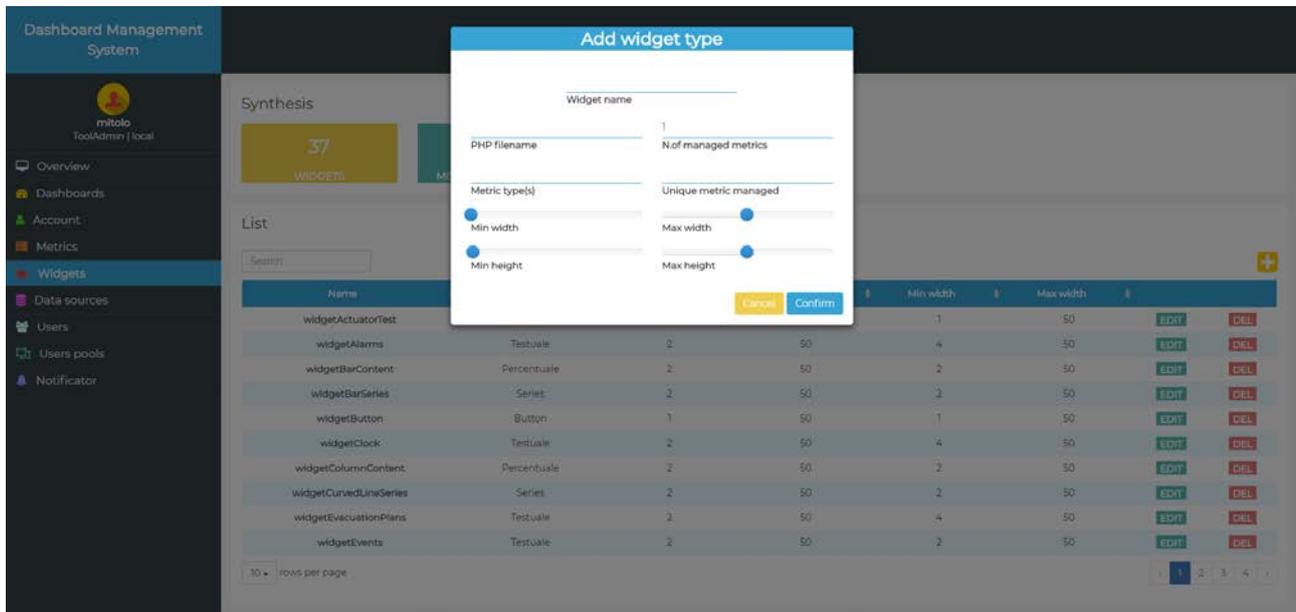
4.3.3. Define a new widget

If needed it is possible to manually define a new widget, for example to represent particular and specific metrics. Click the Widgets items in the left menu column.

The screenshot shows the 'Widgets' management interface. At the top, there are three summary cards: '37 WIDGETS', 'BUTTON MOST INSTANCES - 135', and 'SMARTDS LEAST INSTANCES - 1'. Below these is a table with columns: Name, Data type(s), Min height, Max height, Min width, and Max width. Each row includes 'EDIT' and 'DEL' buttons. A red arrow points to a '+' button in the top right corner of the table area, which is used to add new widgets.

Name	Data type(s)	Min height	Max height	Min width	Max width	EDIT	DEL
widgetActuatorTest	Actuator	2	50	1	50	EDIT	DEL
widgetAlarms	Testuale	2	50	4	50	EDIT	DEL
widgetBarContent	Percentuale	2	50	2	50	EDIT	DEL
widgetBarSeries	Series	2	50	2	50	EDIT	DEL
widgetButton	Button	1	50	1	50	EDIT	DEL
widgetClock	Testuale	2	50	4	50	EDIT	DEL
widgetColumnContent	Percentuale	2	50	2	50	EDIT	DEL
widgetCurvedLineSeries	Series	2	50	2	50	EDIT	DEL
widgetEvacuationPlans	Testuale	2	50	4	50	EDIT	DEL
widgetEvents	Testuale	2	50	2	50	EDIT	DEL

Click the  button on the top right to add a new widget.



The following are the parameters to be set up to define a new widget:

- Widget name:
- PHP filename: the PHP file name that contains the PHP code of the widget. The PHP files has to be in /widgets folder.
- N.of managed metrics: number of metrics that the widget can show (please note that typically a widget will show just one metric but some widgets are present showing 2 or 3 metrics).
- Metric type(s): one or more type of metrics separated by a pipe “|” char. For example “Intero | Percentuale | Float”
- Unique metric managed: in case a widget is specific for a particular metric, specify here the metric name (for example, this is the case of the Meteo and SmartDS widgets).
- Size limits in terms of number of cells within which the widget can be instantiated

A template of the widget PHP code is available to facilitate the creation of new widgets.

```
<?php
/* Dashboard Builder.
   Copyright (C) 2017 DISIT Lab https://www.disit.org - University of Florence

   This program is free software; you can redistribute it and/or
   modify it under the terms of the GNU General Public License
   as published by the Free Software Foundation; either version 2
   of the License, or (at your option) any later version.
   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
   GNU General Public License for more details.
   You should have received a copy of the GNU General Public License
   along with this program; if not, write to the Free Software
   Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301,
   USA. */
```

```
include('../config.php');
header("Cache-Control: private, max-age=$cacheControlMaxAge");
?>

<script type='text/javascript'>
  var colors = {
    GREEN: '#008800',
    ORANGE: '#FF9933',
    LOW_YELLOW: '#ffffcc',
    RED: '#FF0000'
  };
  var parameters = {};

  $(document).ready(function (<?= $_GET['name'] ?>(firstLoad,
metricNameFromDriver, widgetTitleFromDriver, widgetHeaderColorFromDriver,
widgetHeaderFontColorFromDriver, fromGisExternalContent,
fromGisExternalContentServiceUri, fromGisExternalContentField,
fromGisExternalContentRange, /*randomSingleGeoJsonIndex,*/ fromGisMarker,
fromGisMapRef)
  {
    <?php
      $titlePatterns = array();
      $titlePatterns[0] = '/_/';
      $titlePatterns[1] = '/\//';
      $replacements = array();
      $replacements[0] = ' ';
      $replacements[1] = '&apos;';
      $title = $_GET['title'];
    ?>

    var hostFile = "<?= $_GET['hostFile'] ?>";
    var widgetName = "<?= $_GET['name'] ?>";
    var divContainer = $("<?= $_GET['name'] ?>_content");
    var widgetContentColor = "<?= $_GET['color'] ?>";
    var widgetHeaderColor = "<?= $_GET['frame_color'] ?>";
    var widgetHeaderFontColor = "<?= $_GET['headerFontColor'] ?>";
    var nome_wid = "<?= $_GET['name'] ?>_div";
    var linkElement = $("<?= $_GET['name'] ?>_link_w");
    var color = '<?= $_GET['color'] ?>';
    var fontSize = "<?= $_GET['fontSize'] ?>";
    var fontColor = "<?= $_GET['fontColor'] ?>";
    var timeToReload = <?= $_GET['freq'] ?>;
    var widgetPropertiesString, widgetProperties, thresholdObject, infoJson,
styleParameters, metricType, metricData, pattern, totValues, shownValues,
descriptions, udm, threshold, thresholdEval, stopsArray, delta, deltaPerc,
seriesObj, dataObj, pieObj, legendLength, rangeMin, rangeMax, widgetParameters =
null;

    var metricId = "<?= $_GET['metric'] ?>";
    var elToEmpty = $("<?= $_GET['name'] ?>_chartContainer");
    var url = "<?= $_GET['link_w'] ?>";
    var barColors = new Array();
    var embedWidget = <?= $_GET['embedWidget'] ?>;
    var embedWidgetPolicy = '<?= $_GET['embedWidgetPolicy'] ?>';
    var headerHeight = 25;

    if(url === "null")
  {
```

```
        url = null;
    }

    if((embedWidget === true) && (embedWidgetPolicy === 'auto'))
    {
        var showHeader = false;
    }
    else
    {
        var showHeader = true;
    }

    //Specific functions definition

    //End functions definition

    setWidgetLayout(hostFile, widgetName, widgetContentColor,
widgetHeaderColor, widgetHeaderFontColor, showHeader, headerHeight);
    if(firstLoad === false)
    {
        showWidgetContent(widgetName);
    }
    else
    {
        setupLoadingPanel(widgetName, widgetContentColor, firstLoad);
    }
    addLink(widgetName, url, linkElement, divContainer);
    $("#<?=$_GET['name'] ?>_titleDiv").html("<?=$_GET['name'] ?>_titleDiv");
preg_replace($titlePatterns, $replacements, $title) ?>";
    widgetProperties = getWidgetProperties(widgetName);

    if((widgetProperties !== null) && (widgetProperties !== 'undefined'))
    {
        //Start of the eventual ad-hoc code based on widget properties
        styleParameters = getStyleParameters();
        widgetParameters = widgetProperties.param.parameters;
        manageInfoButtonVisibility(widgetProperties.param.infoMessage_w,
$("#<?=$_GET['name'] ?>_header"));

        if(widgetParameters !== null)
        {
            // Code to insert in case you want to use parameters of the
widget
        }
        else
        {
            // Code to insert in case you want to use parameters of the
widget but they are not present
        }

        //End of the ad-hoc code based on widget properties
        metricData = getMetricData(metricId);
        if(metricData !== null)
        {
            if(metricData.data[0] !== 'undefined')
            {
                if(metricData.data.length > 0)
```

```
{
    // Start of the ad hoc code based on metric data
    var metricType =
metricData.data[0].commit.author.metricType;

    if(firstLoad !== false)
    {
        showWidgetContent(widgetName);
    }
    else
    {
        elToEmpty.empty();
    }
    //End of the ad hoc code based on metric data metrica
}
else
{
    showWidgetContent(widgetName);
    $('#<?=$_GET['name'] ?>_noDataAlert').show();
}
}
else
{
    showWidgetContent(widgetName);
    $('#<?=$_GET['name'] ?>_noDataAlert').show();
}
}
else
{
    showWidgetContent(widgetName);
    $('#<?=$_GET['name'] ?>_noDataAlert').show();
}
}
else
{
    console.log("Error in the widget properties loading");
}
startCountdown(widgetName, timeToReload, <?=$_GET['name'] ?>,
metricNameFromDriver, widgetTitleFromDriver, widgetHeaderColorFromDriver,
widgetHeaderFontColorFromDriver, fromGisExternalContent,
fromGisExternalContentServiceUri, fromGisExternalContentField,
fromGisExternalContentRange, fromGisMarker, fromGisMapRef);
});
</script>

<div class="widget" id="<?=$_GET['name'] ?>_div">
    <div class='ui-widget-content'>
        <div id='<?=$_GET['name'] ?>_header' class="widgetHeader">
            <div id="<?=$_GET['name'] ?>_infoButtonDiv"
class="infoButtonContainer">
                <a id ="info_modal" href="#" class="info_source"><i
id="source_<?=$_GET['name'] ?>" class="source_button fa fa-info-circle"
style="font-size: 22px"></i></a>
            </div>
                <div id="<?=$_GET['name'] ?>_titleDiv" class="titleDiv"></div>
                <div id="<?=$_GET['name'] ?>_buttonsDiv" class="buttonsContainer">
```

```
<div class="singleBtnContainer"><a class="icon-cfg-widget"
href="#"><span class="glyphicon glyphicon-cog glyphicon-modify-widget" aria-
hidden="true"></span></a></div>
  <div class="singleBtnContainer"><a class="icon-remove-
widget" href="#"><span class="glyphicon glyphicon-remove glyphicon-modify-
widget" aria-hidden="true"></span></a></div>
</div>
  <div id="<?=$_GET['name'] ?>_countdownContainerDiv"
class="countdownContainer">
  <div id="<?=$_GET['name'] ?>_countdownDiv"
class="countdown"></div>
</div>
</div>

<div id="<?=$_GET['name'] ?>_loading" class="loadingDiv">
  <div class="loadingTextDiv">
    <p>Loading data, please wait</p>
  </div>
  <div class="loadingIconDiv">
    <i class='fa fa-spinner fa-spin'></i>
  </div>
</div>

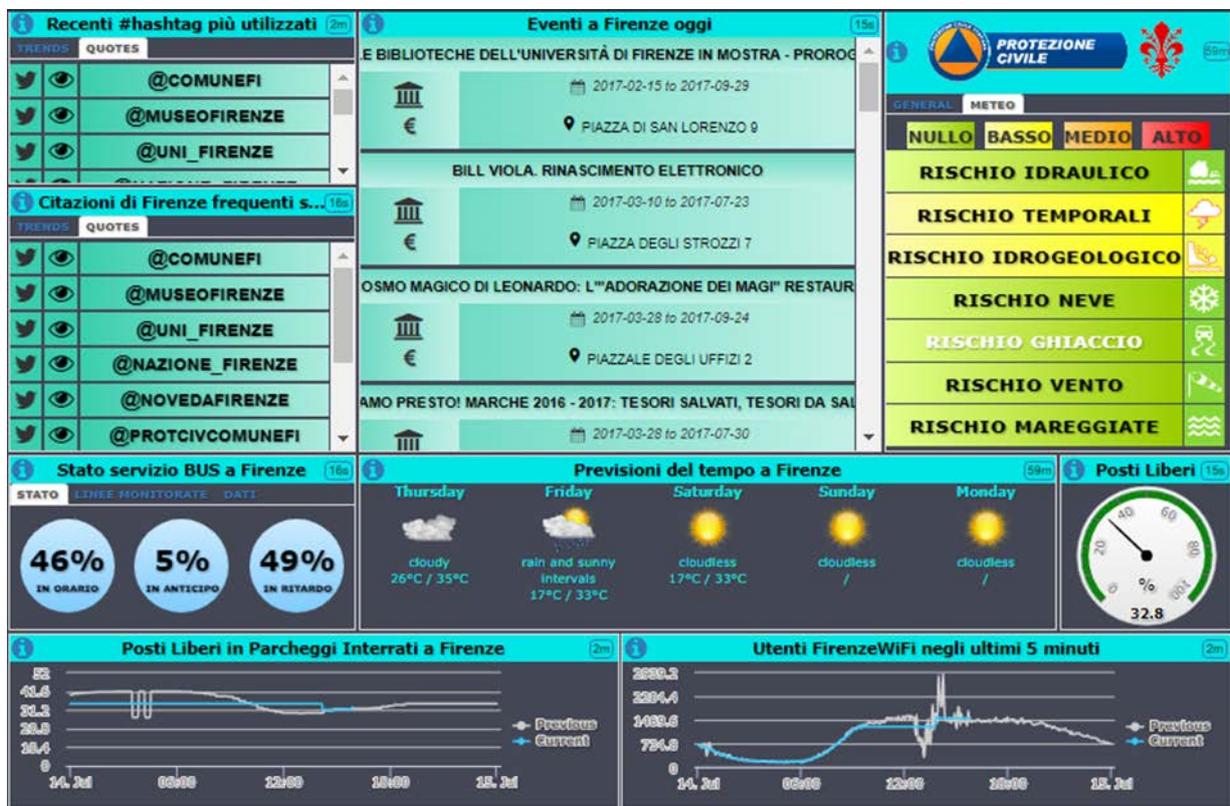
<div id="<?=$_GET['name'] ?>_content" class="content">
  <p id="<?=$_GET['name'] ?>_noDataAlert" style='text-align: center;
font-size: 18px; display:none'>No data available</p>
  <div id="<?=$_GET['name'] ?>_chartContainer"
class="chartContainer"></div>
</div>
</div>
</div>
```

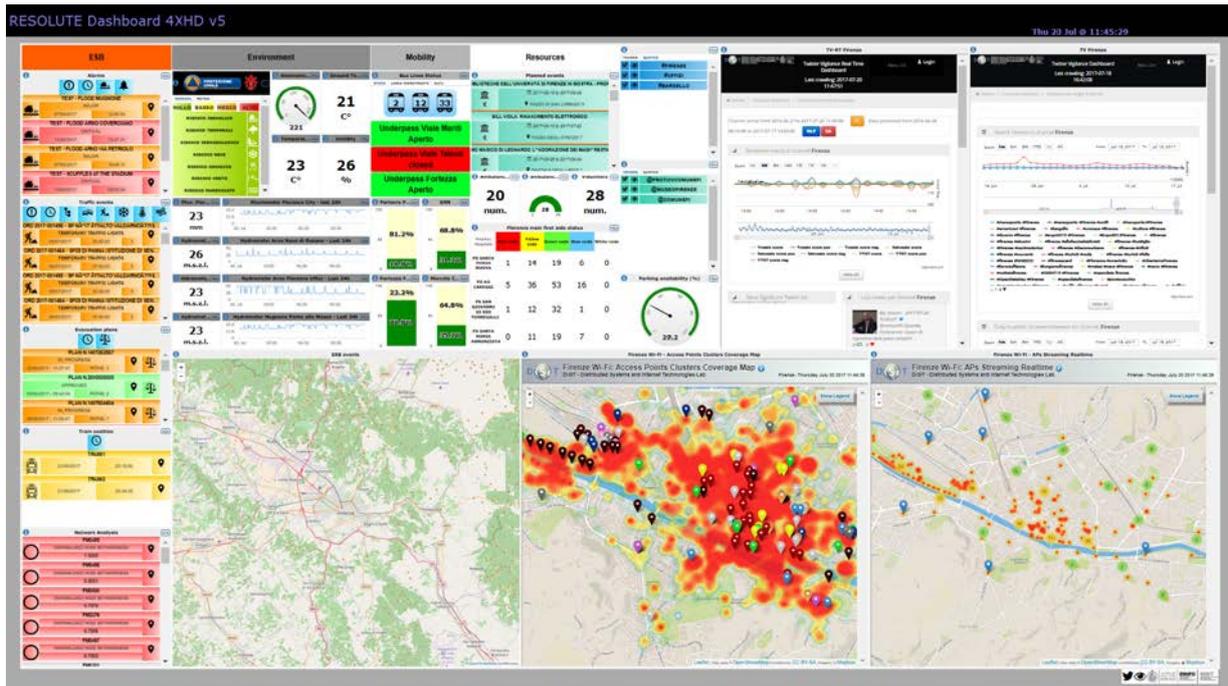
5. Quick start of the Dashboard Management System

This guide is a step-by-step pathway aiming to help users to easily setup their own dashboards, using the Dashboard Management System tool.

NOTE: The examples and metrics represented in this quick start are referred to the tool <https://dashboard.km4city.org/management/index.php> of the DISIT Lab. If you install the tool from scratch, you will find a limited number of widgets and you have to set data sources and metrics to have the tool working (see Sec. 4 for details). If you need a demo account to access the DISIT Lab tool, please send an email to paolo.nesi@unifi.it

Thanks to this manual and the Dashboard Management System tool you will be able to create from simple to very complex dashboards for control rooms or informative dashboards for users like the following, in a very simple process.





- What is the **Dashboard Management System**?
 - It is a web application accessible through different web browsers (Mozilla Firefox and Google Chrome).
 - It can show data taken from different Data Sources: SQL database, database via ODBC, noSQL database (for example MongoDB, Hbase), HDFS directly or via Phoenix, RDF stores via SPARQL, from API calls and from streaming data.

- What is it for?
 - It is able to present real-time data in several different manners with real-time updates autonomously H24 7/7 days (typically for control rooms), in particular:
 - The Dashboard Builder allows to:
 - control size and zoom of the Widget of the Dashboard;
 - set and change parameters as: size, color of the background, font of the header, color of the header, etc.
 - compose a set of possible views (in most cases called widgets), by visual interaction, visual composition of widgets;
 - create and editing Dashboard produced by one or more users;
 - edit controlling each single Dashboard by a single person;
 - allow each single dashboard to be viewed/accessed in reading by one or more specific people;
 - be cloned to derive a new dashboard from a former one and perform versioning;
 - Dashboard Widget allows to:
 - show data according different paradigms and graphic models.

- refresh the data according to a periodic time by setting it per widget, coherently with the refresh setting/value of the corresponding metric providing the evidence of the periodic update of data and of the fact that the information is changing.
 - be open to host and thus to create new widgets by programming, for example in PHP, Java, Python, may be using third party libraries on it for: TVcamera, graphics, execution, stream,etc;
- The Metrics are specific selections/queries performed on some Data Source, and:
 - allow to take simple as well as complex data from some Data Source.
 - can be programmed to be refreshed, recollected with a given frequency.

5.1. Use Dashboard Management System

5.1.1. Login

This is the first web page displayed:

- Type** your *Username* and *Password* (**you have to be registered**. To access the tool managed by DISIT Lab (<http://dashboard.km4city.org/dashboardSmartCity/management/>) ask for a username and password by sending an email to paolo.nesi@unifi.it).
- Click** “*Login*” to access to the tool.



5.1.2. Creating a new dashboard

- Click the “Dashboards” menu item (1) on the left column.
- Click the “+” button (2) on the top right of the page.

The screenshot shows the Dashboard Management System interface. On the left, a sidebar menu has the 'Dashboards' item highlighted with a red circle and the number '1'. The main area displays a 'Synthesis' section with three widgets: '68 DASHBOARDS' (yellow), '68 ACTIVE' (green), and '67 PUBLIC' (red). Below this is a 'List' section with a search bar and a table of dashboards. The table has columns for Title, Creator, and Status. A red circle and the number '2' are placed over a '+' button in the top right corner of the table.

Title	Creator	Status
aaaa	marazzini	On EDIT VIEW
aaaaCloned	marazzini	On EDIT VIEW
Console	marazzini	On EDIT VIEW
Convention Bureau - Meteo for integration	FlorenceConventionBureau	On EDIT VIEW
Convention Bureau - Mobility for integration	FlorenceConventionBureau	On EDIT VIEW
Cross widget interaction 1	marazzini	On EDIT VIEW
Cross widget interaction 2	marazzini	On EDIT VIEW

- A new pop-up will be shown with different tabs, that has to be used to define the structure and the basic layout of the dashboard.
As shown in the image below, you will see some empty fields and some fields with default values. As first step you have to define the width of the dashboard (expressed in cells). The fields *Pixel width* and *Width (%) on your screen* will be filled automatically).

The screenshot shows the Dashboard Management System interface with a 'Create dashboard' pop-up dialog box open. The dialog has tabs for 'Measures', 'Header', 'Body', 'Visibility', and 'Embeddability'. The 'Measures' tab is active, showing fields for 'Width (cells)' (790 px), 'Pixel width', 'Width (%) on your screen' (41 %), and 'Width (cells)'. There are 'Cancel' and 'Confirm' buttons at the bottom of the dialog. The background shows the same dashboard list as the previous screenshot.

Fill the empty fields without editing the default values as you can see in the next image.

Measures Tab.

- 1) **Width:** type the Width of your dashboard (in number of cells). The number is automatically converted in the equivalent values in number of pixels and percentage of screen width (based on the user screen resolution).

The screenshot shows the 'Create dashboard' dialog box with the 'Measures' tab selected. A slider for 'Width (cells)' is set to 10. Below the slider, the 'Pixel width' is 790 px and the 'Width (%) on your screen' is 41%. There are 'Cancel' and 'Confirm' buttons at the bottom right.

Header Tab:

- 2) Type the Title.
- 3) Type the Subtitle.
- 4) Set the color of the header background
- 5) Set if to show the header or not (only in view mode)
- 6) Set the header font size
- 7) Set the header font color
- 8) Choose a file in your PC to load a Dashboard logo. It will be shown on the top right of your dashboard
- 9) Set the link of the logo. By clicking the logo the link will be opened in a new window.

The screenshot shows the 'Create dashboard' dialog box with the 'Header' tab selected. Fields include: Title, Subtitle, Header color (rgba(83,103,206,1)), Show header (Yes), Header font size, Header font color (rgba(255,255,255,1)), Header logo (File button), and Header logo link. There are 'Cancel' and 'Confirm' buttons at the bottom right.

Body Tab:

- 10) Widget background color

- 11) Color of the external frame
- 12) Set if the widgets have borders or not
- 13) Set the widgets borders color

Create dashboard

Measures Header **Body** Visibility Embeddability

rgba(255,255,255,1) **(10)** Widgets area color

rgba(255,255,255,1) **(11)** External frame color

(12) Yes Widgets borders

(13) rgba(221,221,221,1) Widgets borders color

Cancel Confirm

Visibility Tab:

- 14) Define the dashboard visibility by specifying who will be authorized to view the dashboard.

Create dashboard

Measures Header Body **Visibility** Embeddability

Dashboard author only **(14)**

Dashboard author only
Author and selected users
Everybody (public)

Cancel Confirm

Embeddability Tab:

- 15) Set the external URLs authorized to embed the dashboard. This functionality allow the dashboard to be embedded in an external web page or not.
- 16) Click *Confirm* to save the settings.

Create dashboard

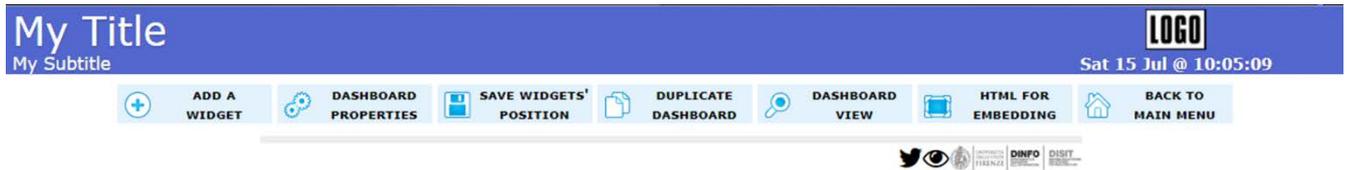
Measures Header Body Visibility **Embeddability**

(15) | [input] [check] [x]

Empty

Cancel **(16)** Confirm

You will get a similar result.

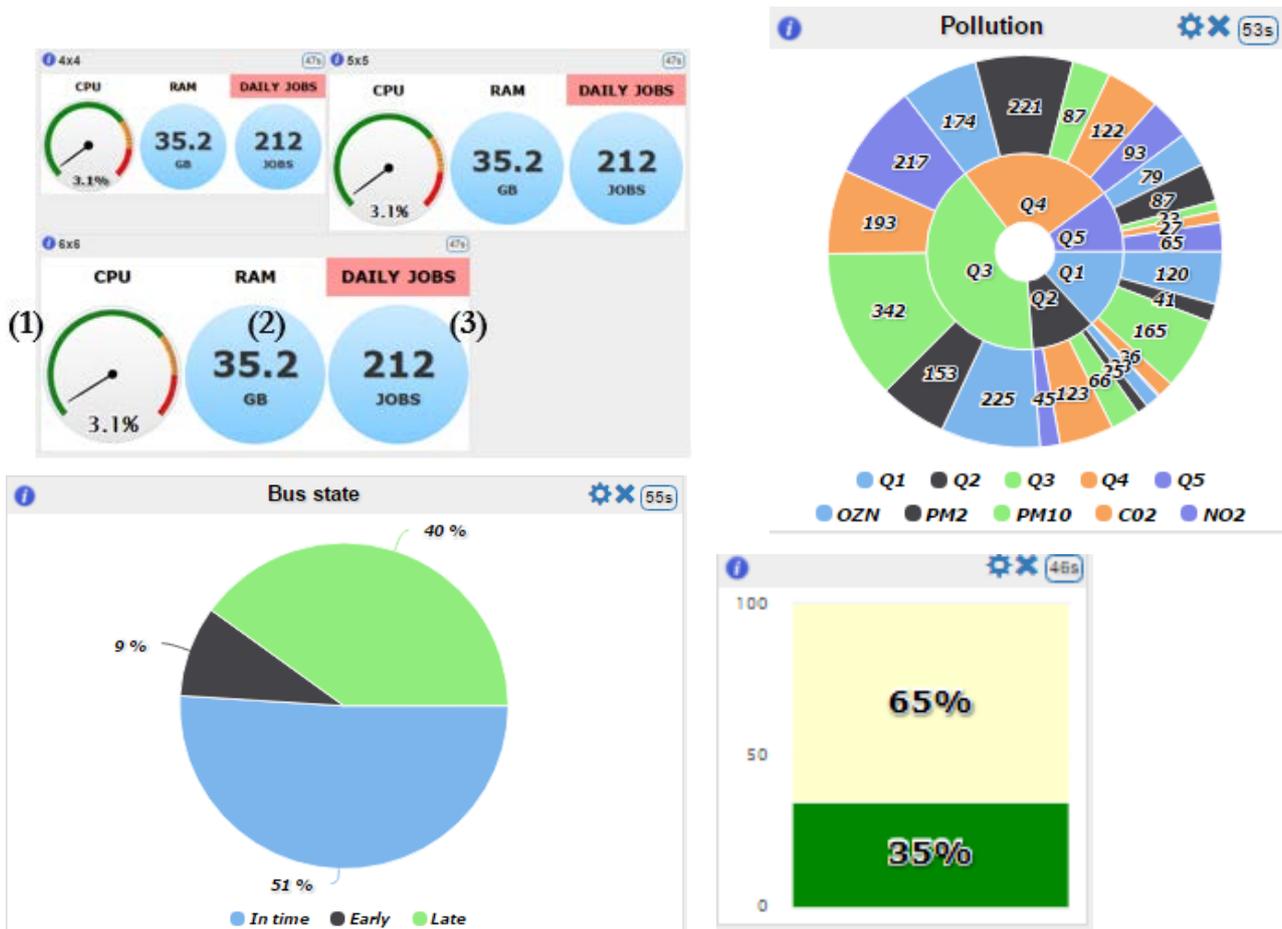


Now you can populate it with widgets!

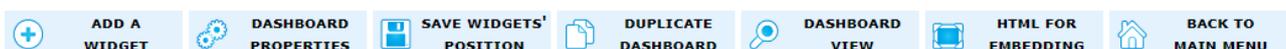
5.1.3. Adding your first widget in your dashboard

You have created your first dashboard, now add a widget!

In this section and in the next one you will learn step-by-step how to populate your dashboard with widgets like these:

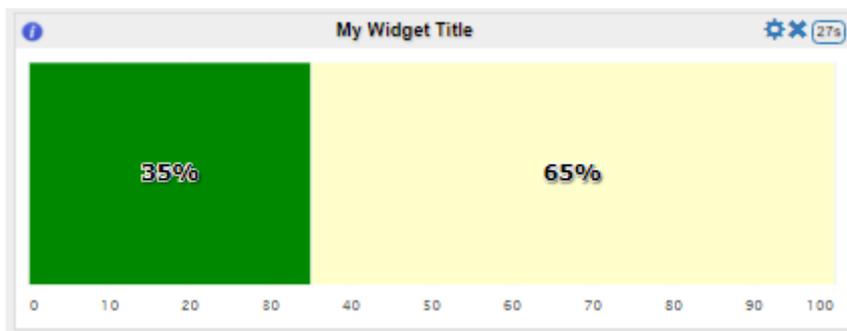


a) Click the "ADD A WIDGET" button.



- Click the little triangle in the *Width* field to open the drop-down menu and **choose** the value 8.
- Click the “*Create*” button to save the widget properties.

You will get a similar result.

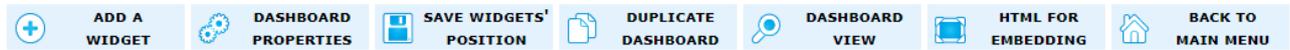


You have created your first widget.

5.1.4. Adding your second widget in your dashboard

After creating your first widget, you are ready to create your second one:

a) **Click** the “*ADD A WIDGET*” button.



b) **Fill** the empty fields without editing the default values following these steps.

We help you to add your second widget, a Pie Chart with the percentage of early, in time and late available bus lines:

1. **Click** the little triangle in the *Metric* field to open the drop-down menu and **choose** *Bus_State_Lines*.
2. **Click** the little triangle in the *Widget type* field to open the drop-down menu and **choose** *widgetPieChart* (*Specific widget properties* fields are filled automatically).
3. **Click** “*Add*” button to add the selected metric to the selected widget.
4. **Type** the widget *Title* (the *Content font size* and the *Content font color* fields are filled automatically).
5. **Click** the little triangle in the *Height* field to open the drop-down menu and **choose** the value 8.
6. **Click** the little triangle in the *Width* field to open the drop-down menu and **choose** the value 8.
7. **Click** the “*Create*” button to save the widget properties.

Add new widget to dashboard

Metric and widget choice

Metric: Bus_State_Lines (1) Add (3)

Metric description: Description: Percentuale di ritardi/anticipi sulle linee disponibili.

Context: [empty]

Widget type: widgetPieC max 1 metrics (2)

Selected metrics: Bus_State_Lines

Widget link: [empty]

Generic widget properties

Title: My Second W (4) Background color: #ffffff

Content font size: [empty] Content font color: [empty]

Header color: #cccccc Header text color: #000000

Period: [empty] Refresh rate (s): 60

Height: 8 (5) Width: 8 (6)

Units of measure: [empty]

Specific widget properties

Legend font size: 10 Legend font color: #000000

Data labels distance: 15 Data labels font size: 10

Data labels font color: #000000 Inner radius (%): 0

Start angle (°): 0 End angle (°): 360

Diagram center Y(%): 50 Slices colors: Autom

Set thresholds: No

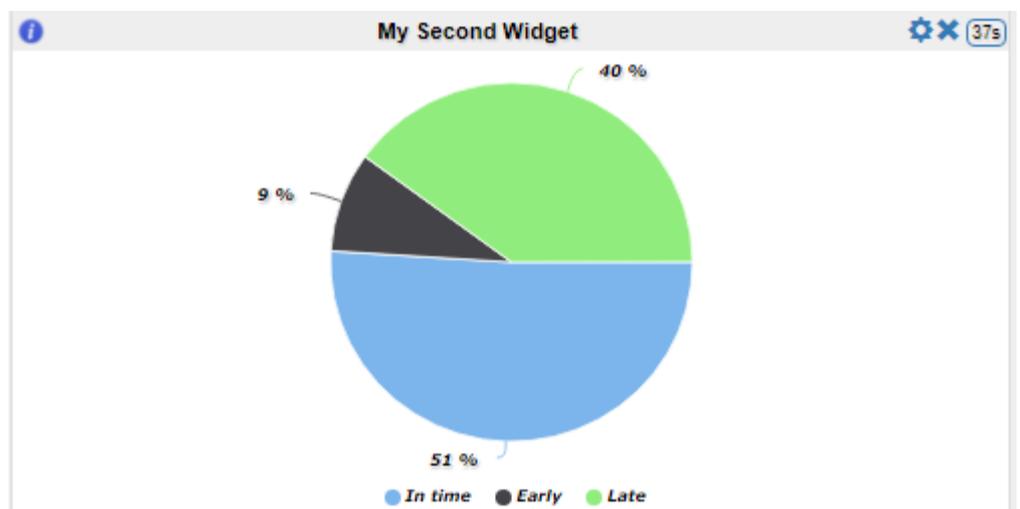
Widget detailed informations

[Rich text editor toolbar with icons for bold, italic, link, unlink, text color, background color, bulleted list, numbered list, indent, outdent, undo, redo, source, help]

(7)

Close Reset Create

You will get a similar result.



5.1.5. Modify your widget

In this chapter you will learn step-by-step how to modify your widget properties!

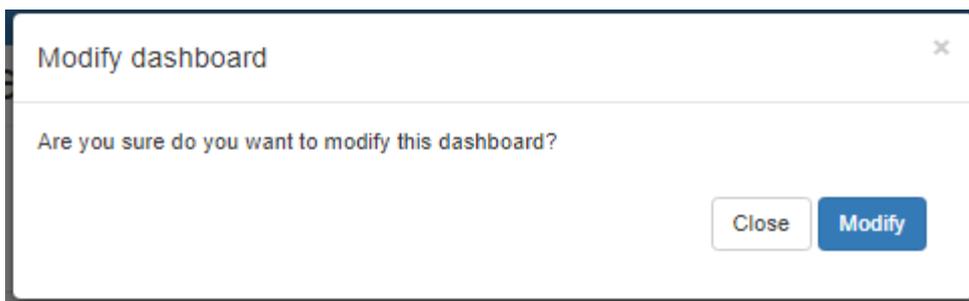
You have added two widgets to your dashboard but, if you wanted to modify one of them (the first one for example) what should you do?

We are in the homepage (the page displayed after the login).

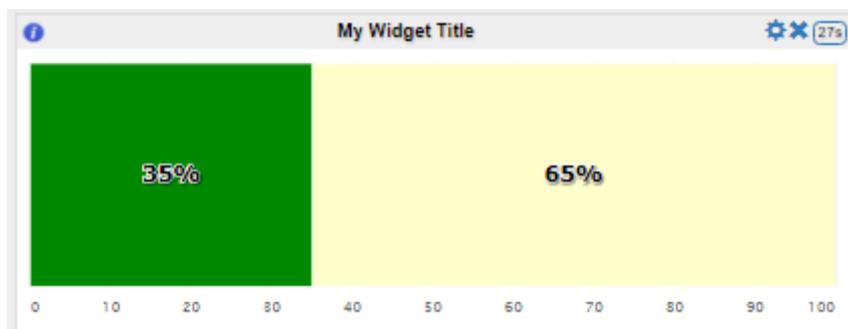
- a) **Click** the *Modify* button of your dashboard.

Name	Title	Creation Date	Status	User	Modify	Preview
My Dashboard	My Title	2017-07-15 10:42:41	<input checked="" type="checkbox"/>	Filippo		

- b) **Click** *Modify* to confirm the choice.



- c) Your dashboard and its widgets are shown.
Click to open the pop-up with widget properties.



- d) You can see the widget properties chosen when you have created this widget.

In particular:

Metric and widget choice

Metric: Ataf_Rt Add

Metric description: Description: Percentuale dei segnali in tempo reale che vengono da ATAF.

Context: [Empty field]

Widget type: widgetBarC max 1 metrics

Selected metrics: [Empty list]

Widget link: [Empty field]

Generic widget properties

Title: My Widget Ti Background color: #ffffff

Content font size: 16 Content font color: #0000

Header color: #eeee Header text color: #0000

Period: [Empty dropdown] Refresh rate (s): 60

Height: 6 Width: 8

Units of measure: [Empty field]

Specific widget properties

Widget detailed informations

[Rich text editor toolbar and source field]

Close Reset Create

Modify some fields with the following steps:

1. **Modify** the *Title*.
2. **Click** the square in *Background color* field and **choose** a different color.
3. **Modify** the *Refresh rate* value to 30 s.
4. **Click** the little triangle in the *Height* field to open the drop-down menu and **choose** the value 8.
5. **Click** the little triangle in the *Width* field to open the drop-down menu and **choose** the value 11.
6. **Click** the “*Modify*” button to save the modified widget properties.

Modify widget

Metric and widget choice

Metric:

Widget name:

Widget type: max 1 metrics

Context:

Widget link:

Metric description: Name: Ataf_Rt. Description: Percentuale dei segnali in tempo reale che...

Generic widget properties

Title:

Background color:

Content font size:

Content font color:

Header color:

Header text color:

Period:

Refresh rate (s):

Height:

Width:

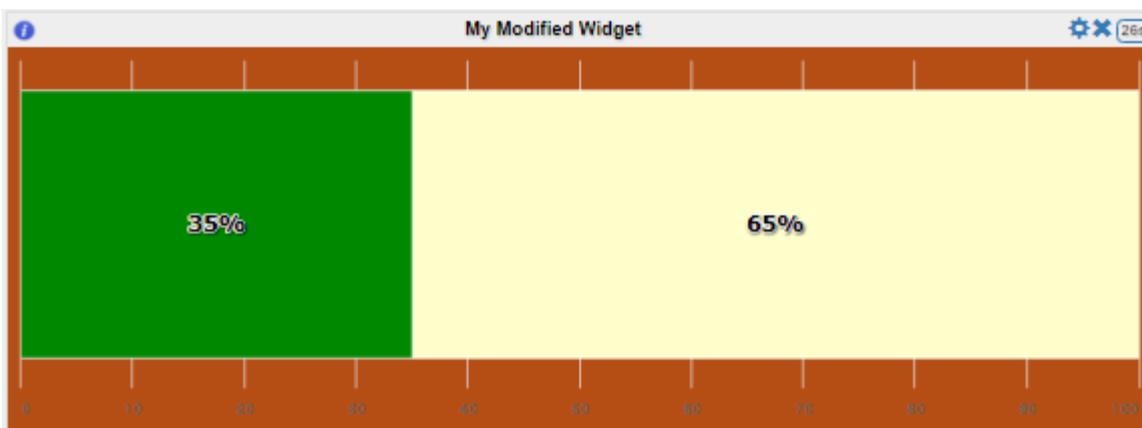
Specific widget properties

Widget detailed informations

1 3 4 5 6

Close Modify

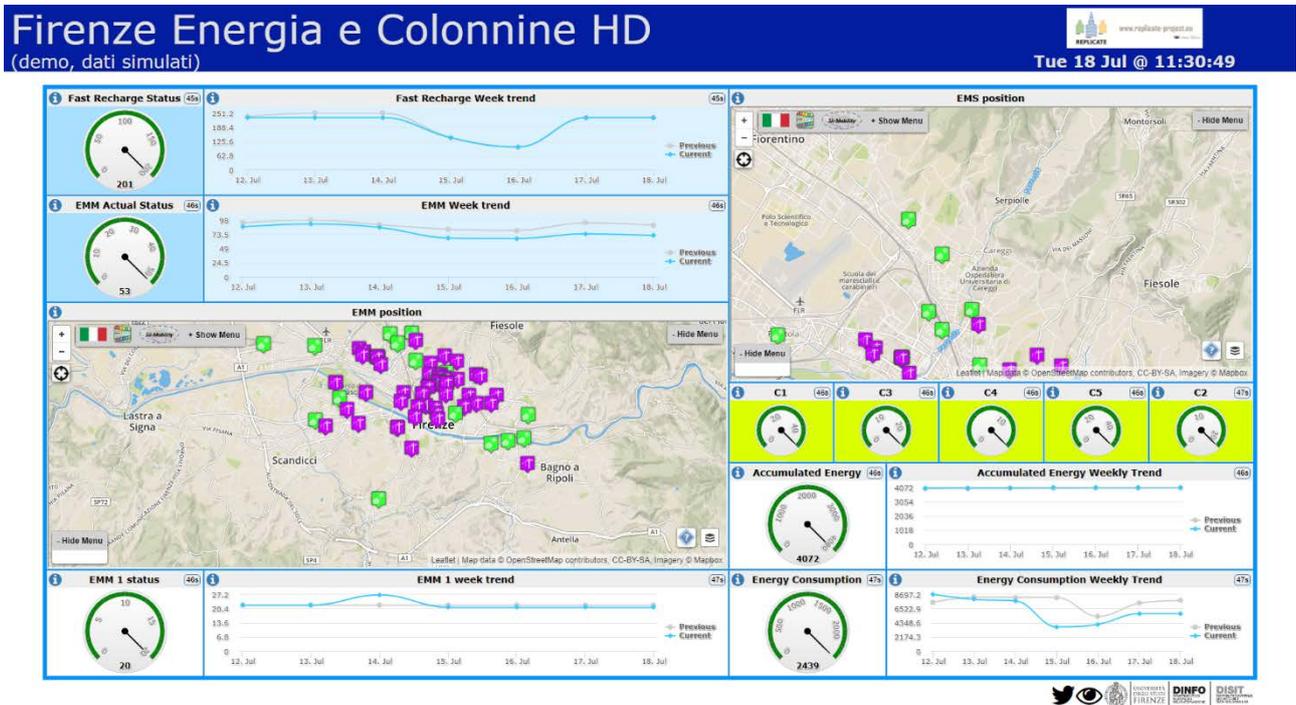
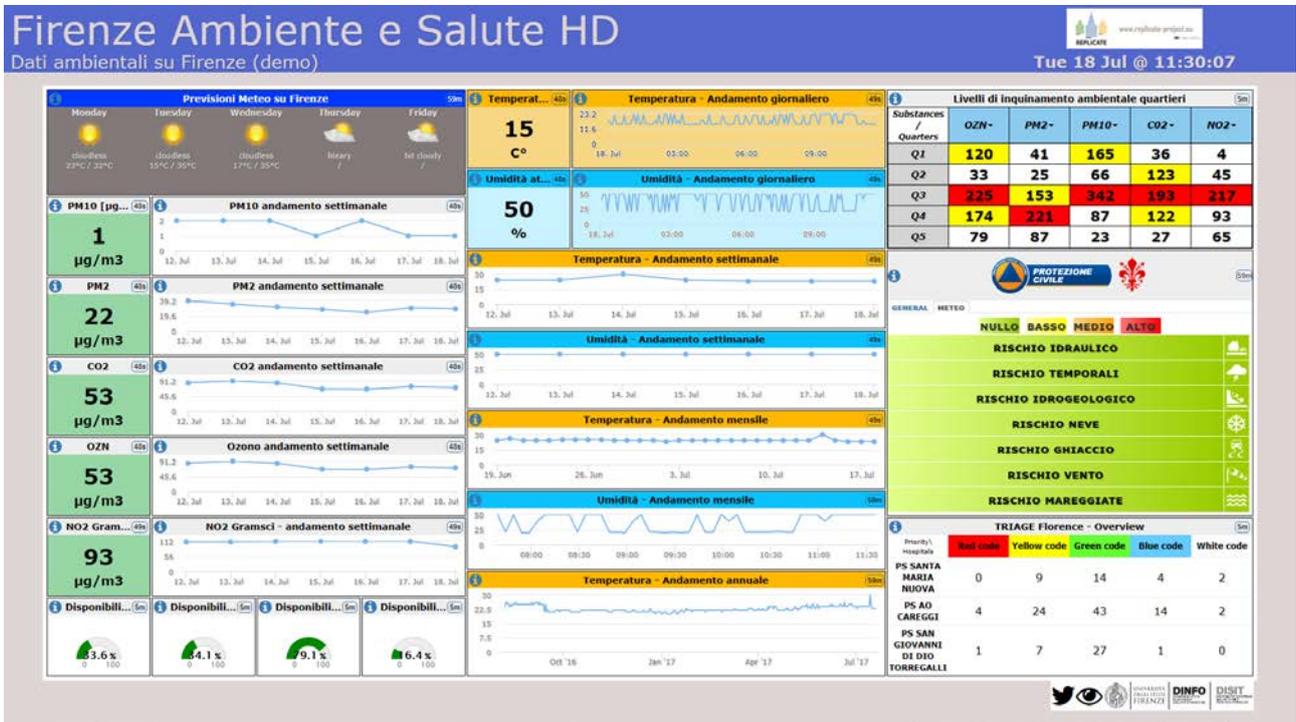
You will get a similar result.



You have modified your first widget!

5.2. Conclusion

You can handle every kind of static and real-time data quickly and easily, even reaching complex results.

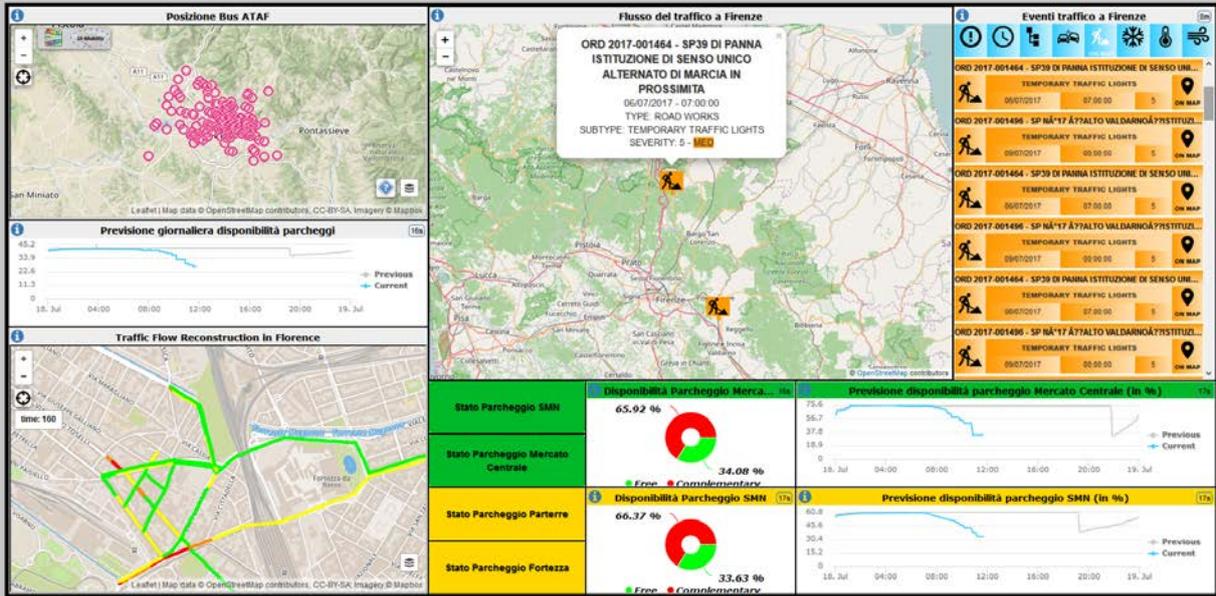




Firenze Mobilità e Trasporti HD

(demo, dati non completi)

Tue 18 Jul @ 11:38:41



Smart City Data Overview

Sperimentale

Tue 18 Jul @ 11:35:02



6. Designing and editing Dashboard

To access the tool it is necessary to enter Username and Password. The following examples and images are based on the tool <https://dashboard.km4city.org/management/index.php> of the DISIT Lab. If you install the tool from scratch, you will see that many metrics and widget are not working. Before to have the tool working properly you have to add your data sources and build metrics properly (see Sec. 4).

If you need a demo account to access the DISIT Lab tool, please send an email to paolo.nesi@unifi.it.

To access the tool it is necessary to enter Username and Password.



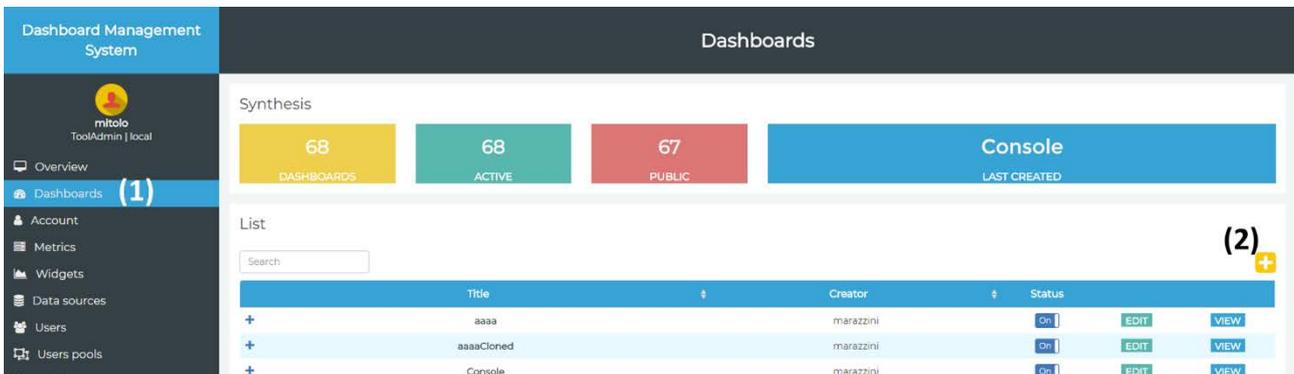
After this, the page will show the list of the whole dashboards that has been created and that are accessible for the user.

The page shows in the right column the menu with the following items:

- **Dashboard Builder**, that allows to create a new dashboard or to edit a yet available dashboard or to see a preview.
- **Metrics management**: in this page, it is possible to create and modify metrics.
- **Widgets management**: in this page, it is possible to create or edit widgets. Please note that a specific section that explain the available widgets is in this user manual;
- **Data sources management**, to define or edit the data sources used by the metrics;
- **Users management**, to create new users to provide them the access to the tool;
- **Users pools management**, to define and manage group of users;
- **Notificator**, the tool that allow to define and manage rules and thresholds to send notifications to users in case of unexpected values.

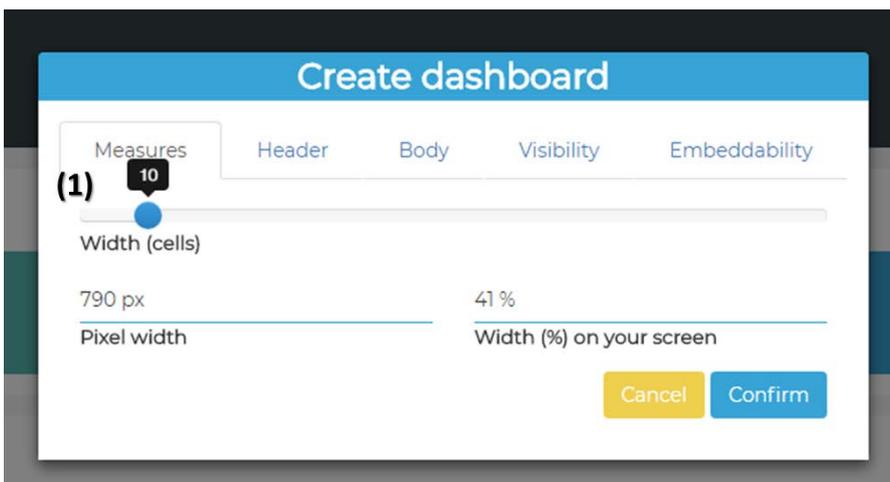
6.1. Create a new dashboard

To create a new dashboard, click the “Dashboard management” menu item (1) on the left column and click the  on (2) on the top right of the page.



Title	Creator	Status
aaaa	marazzini	On EDIT VIEW
aaaaCloned	marazzini	On EDIT VIEW
Console	marazzini	On EDIT VIEW

A new pop up will be shown with a number of tabs to be filled in to be used to define the structure and the basic layout of the dashboard, in particular:



Width (cells)

790 px 41 %

Pixel width Width (%) on your screen

Cancel Confirm

Tab **Measures**:

- (1) Width: in terms of number of cells, that is the unit of measures of the dashboard. The number of cells is automatically converted in pixel width and percentage of width of the monitor occupied and shown in the two corresponding fields below.
 - a. The dimension expressed in pixel (automatically calculated)

- b. The percentage of monitor width occupied by the dashboard (automatically calculated on the basis of the user's monitor width)

Create dashboard

Measures Header Body Visibility Embeddability

(2) Title

(3) Subtitle

rgba(83,103,206,1) (4) Header color

(5) Show header

(6) Header font size

rgba(255,255,255,1) (7) Header font color

(8) Header logo

(9) Header logo link

Cancel Confirm

Tab **Header**:

- (2) Title: the title of the dashboard that will be shown on the top left of the dashboard page;
- (3) Subtitle (optional), that is shown below the Title;
- (4) Header color: the color of the background of the header;
- (5) Show header (Yes/No). With this slide you can decide to show the header of the dashboard or not (valid only in view mode. In edit mode the header is always visible);
- (6) Header font size (in pixel);
- (7) Header font color;
- (8) Header logo: an image in any format and size. It will be resized automatically and adapted to the available header space. The dashboard logo is show on the topo right;
- (9) Header logo link: to be opened if the user clicks the logo;

Create dashboard

Measures Header Body Visibility Embeddability

rgba(255,255,255,1) (10) Widgets area color

rgba(255,255,255,1) (11) External frame color

(12) Widgets borders

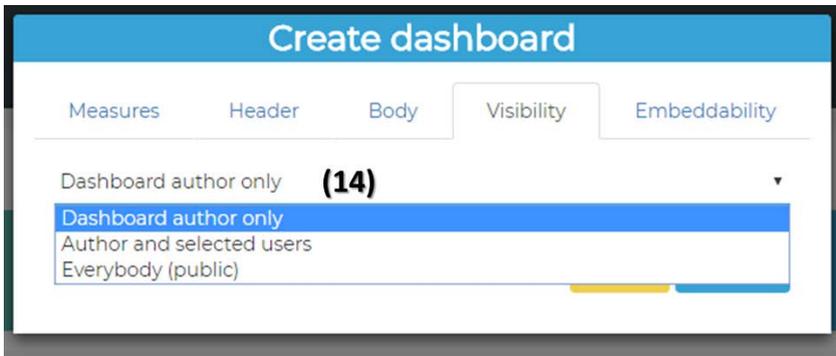
(13) Widgets borders color

Cancel Confirm

Tab **Body**:

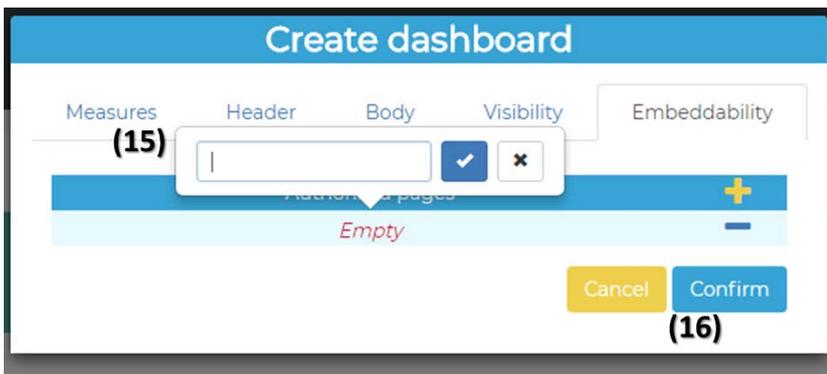
- (10) Widget area color: The background color of the dashboard;

- (11) External frame color
- (12) Widget borders (Yes, No)
- (13) Widget borders color



Tab **Visibility**:

- (14) Dashboard visibility: select from a drop-down menu who can view the dashboard. It can be a public dashboard visible to all that have access through the direct link (“Everybody (public)”), a dashboard visible only to the author or a dashboard accessible to a group of selected users.

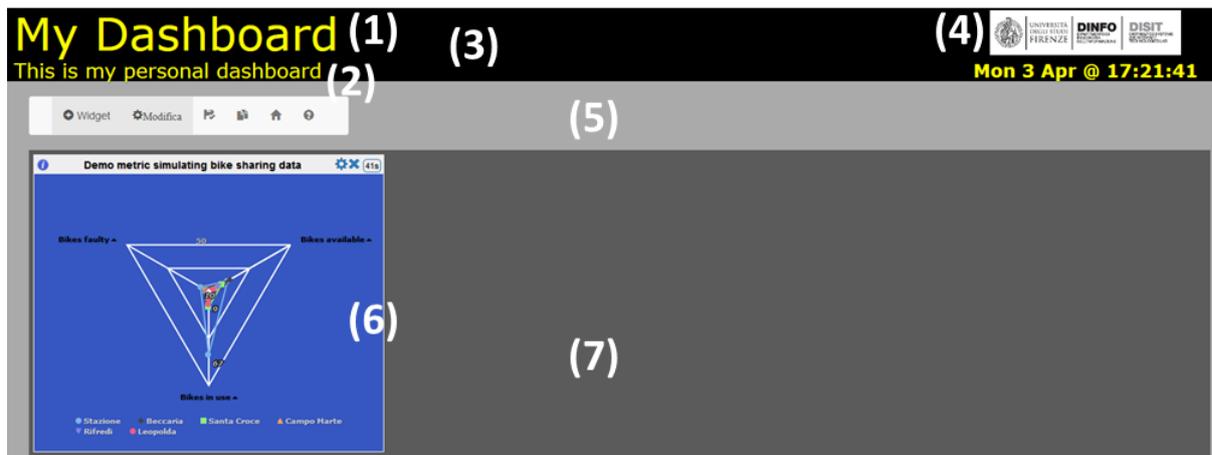


Tab **Embeddability**:

- (15) To authorize or not the possibility to embed the dashboard in external web pages, it is needed to specify the URLs of the webpage that hosts the dashboard, otherwise the widgets will not be visible. It is sufficient to indicate the main URL and the dashboard will be visible also in all the subpages of the same link.

By clicking the “Create” (16) button the dashboard will be saved and will be possible to populate it with widgets.

The following image show what each field of the form is representing with respect the previous list of settings. The widget inside is just an example.



The following image maps the setting defined during the dashboard creation and the dashboard

- (1) Dashboard title, with header font size and color
- (2) Dashboard subtitle
- (3) Header color
- (4) Header logo
- (5) External frame color
- (6) Widget border color
- (7) Background color

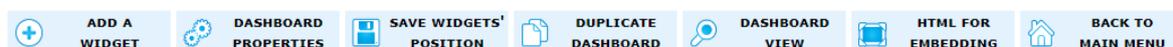
7. Designing and Editing Widgets

To add a new widget in a dashboard you have to enter in the edit mode. Click the edit button in the list of available dashboards (Menu: *Dashboards*) to add a new widget in an existing dashboard, or create a new dashboard.

Title	Creator	Status	EDIT	VIEW
Stato parcheggi	marazzini	On	EDIT	VIEW
Temperatures - Room of machine 166	disit	On	EDIT	VIEW
Test	marazzini	On	EDIT	VIEW
Test 2	marazzini	On	EDIT	VIEW
Tourism info	marazzini	On	EDIT	VIEW
Tourism info	FlorenceConventionBureau	On	EDIT	VIEW
Tourism info	marazzini	On	EDIT	VIEW
Tourism info	FlorenceConventionBureau	On	EDIT	VIEW
Tuscany weather dashboard 1	marazzini	On	EDIT	VIEW
Tuscany weather dashboard 2	marazzini	On	EDIT	VIEW

As shown in the above image, a number of buttons will be shown on the top of the page (from left to right):

- **Add Widget**, click here to add a widget in your dashboard;
- **Dashboard Properties**, click here to modify the dashboard settings (see previous section);
- **Save Widgets Position**, click here to save the current position of the widgets in the dashboard. In edit mode, since you can move your widgets on the screen to find a better position, you should click here to save the status when it is satisfactory for you;
- **Duplicate dashboard**, click here to duplicate your dashboard. You will have to specify a new name for your new duplicated dashboard;
- **Dashboard view**, click here to see your dashboard in view mode. For a better experience we suggest visualizing your dashboard by the browser in full screen mode;
- **HTML for embedding**, click here to see the HTML code automatically generated to be copied in external webpages to embed the dashboard (as iFrame). See [Sec. 8](#);
- **Back to main menu**, click here to exit to the edit mode without saving the current status. You will be redirect to the page with the list of available dashboards.



To add a widget, click the Widget button with the **ADD A WIDGET** button. A new pop-up form will be open that will allow you to select the widget you prefer and to set the corresponding parameters.

As first step, you have to select the metric in the dropdown menu on top of the page (1). Once selected the correct metric, click the *Add* button (2). You can add more different metrics and widgets to compose areas with different widgets inside (if the number of added widgets exceed the page size, the widgets will be moved in the next line).

In the *Widget detailed information* on the bottom right, it is possible to add a info description that will be shown when the user clicks on the info button of the widget when published (3). According to the metric selected, you will be able to select the *Widget type* (4). In the *Widget type* dropdown menu a list of widgets will be shown coherent with the metric selected. If needed you can add a link to the widget that will be open if the user clicks on the widget itself (5).

The *Generic widget properties* on the top right are common for all widgets (6) whereas, once selected the widget, the *Specific widget properties* will be shown on the bottom left of the page (7). These properties are different for the various widgets. As example, the next image maps the fields available in the Generic widget properties form and the widget areas (in this example the widget is a *Speedometer*).

In particular, in the **Generic widget properties** box it is possible to specify:

- (1) The title of the widget that will be shown in the header bar;
- (2) The color of the widget background;

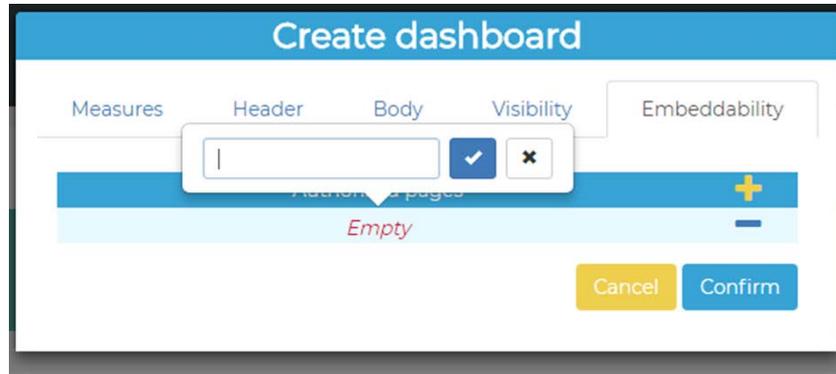


- (3) The font size and color of the content text shown in the widget;
- (4) The color of the header bar and of the text in the header;
- (5) The period dropdown menu is available for specific widget types that represent values showed in a period of time (also the last period can be compared with a previous period of time by using the TimeTrendCompare widget). The period showed in the widget can cover 4 hours, 12 hours, Daily, Weekly, Monthly, Annually. These values have to be stored and available to be showed correctly;
- (6) Refresh rate represent the period of time the value in the widget has to be refreshed. As default, every 60 seconds the widget is refreshed but this refresh rate value has to be set according to the metric since it is not useful to update and refresh every 60 secs a widget representing a value updated for example 1 time per day;
- (7) Height and width are expressed in columns;
- (8) The unit of measure of the value to be shown in the widget.

8. How to embed a dashboard in a HTML external webpage

8.1. Dashboard embeddability and relative whitelist

A dashboard can be allowed or not to be hosted in external Web pages via the *Visibility* Tab available in both *Add dashboard* and *Edit dashboard* menus.



- *Authorized pages*: here you have to specify the web pages where the embedded dashboard has to be visualized by adding the specific URL. You can add new pages as you want to the list by clicking the plus sign, then clicking on the *Empty* word you have to put the precise URL of the page you want to allow to embed your dashboard in the line editor that has appeared, then press the *blue flag* button to confirm.

E.g., put <https://www.disit.org> if you want this page to be allowed to embed your dashboard.

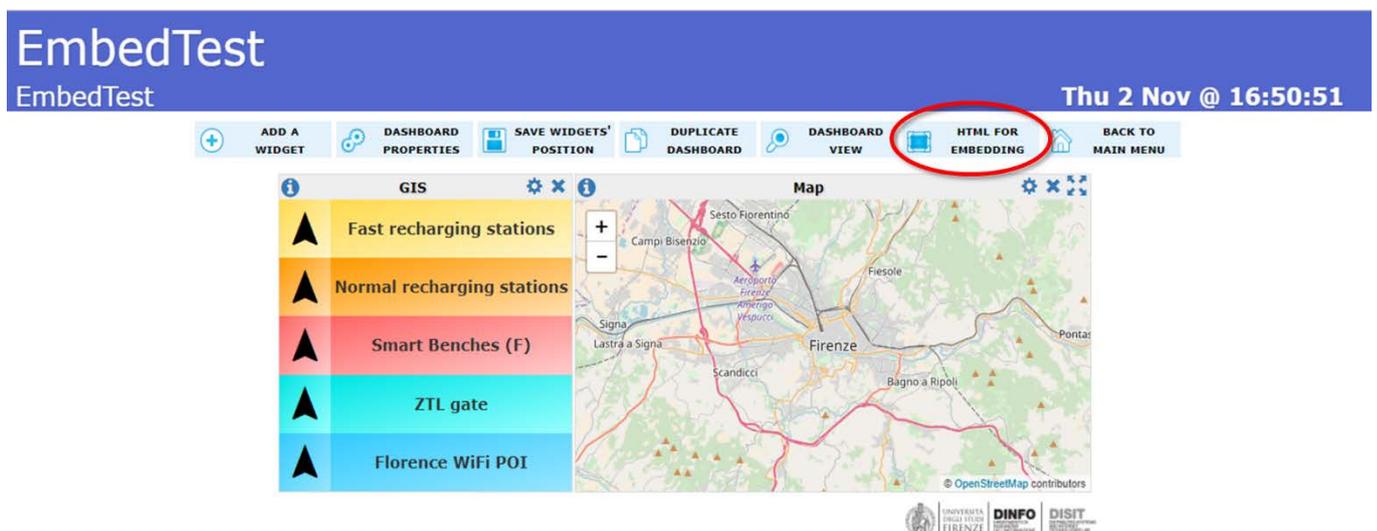
If you want to change a page URL in the white list, click on the URL, then edit the address in the line editor.

In order to remove a page from the list, simply press the minus sign next to the page to be deleted.

By indicating an URL you authorize the dashboard to be embedded also in any sub-URL of any sub level of the specified URL.

8.2. Dashboard embedding options, preview and markup generation

A dashboard can be embedded in an external Web page in an iframe object, setting its *src* attribute with a specific URL. This URL can be retrieved in every moment of dashboard lifecycle from the *HTML for embedding* button in the *Edit dashboard* window:



By clicking this button, the *Embed this dashboard* window will open:

This window is composed by three main areas:

Iframe markup generation - In this area you can valorize some parameters in order to customize the emedding mode and get the generated markup via cut&paste.

- *Embed mode* - There are two embed modes available:
 - *Automatic* - This mode will remove dashboard header and widgets's header, moreover it will perform an autofit of the dashboard to the iframe area. Autofit is implemented by a CSS scaleX and scaleY application to the document body, so it suits very well iframes with dimensions not too much different from dashboard ones. A "*auto refit in progress, please wait!*" message will appear while autofit is being performed. Dashboard footer will always be shown with **KM4City** logo when dashboard is embedded.
 - *Manual* - If you select this mode you can manually choose whether to show or hide dashboard header when embedded (this option is indipendent from the homologous one available in the dashboard management menu, with the latter applying only to classic stand-alone dashboard view) and to apply or not autofit. You can choose to show or hide the header of each widget via the *add widget* or *edit widget* menu. Dashboard footer will always be shown with **KM4City** logo when dashboard is embedded.
- *Iframe width (px)* - Width expressed in pixels for the host iframe.
- *Iframe height (px)* - Height expressed in pixels for the host iframe.
- *Code to embed this dashboard* - This HTML markup is generated in real time during parameters adjustment.
- **Save iframe markup** - In this area you can save the generated markup in two different ways:
 - *Save in txt file*: click this button and the system will open a popup window to save in your hard drive a .txt file containing the generated markup.



- *Save in HTML file*: click this button and the system will open a popup window to save in your hard drive a basic but complete HTML page containing the generated iframe.
- **Iframe preview** - This area will show you the generated iframe object in real time during parameters adjustment.

More information regarding how to embed a dashboard in external webpages is available in Section 13.

9. Available Widgets details

The widget is a graphical elaboration of data (Php File). It is useful to represent some *Metrics* in a different and graphic user-friendly way.

9.1. Available widgets on DISIT Lab tool

The following tables shows the list of all the widgets available in the Dashboard Management tool managed by DISIT Lab, with information about minimum/maximum number of column and rows and the type of graphical representation. Please note that some of the following widgets are specific for the DISIT Lab installation.

Widget name	min columns	max columns	min rows	max rows	Type
widgetAlarms	2	50	4	50	Textual
widgetBarContent	2	50	2	50	Percentage
widgetBarSeries	2	50	2	50	Series
widgetButton	1	50	1	50	Button
widgetClock	2	50	4	50	Textual
widgetColumnContent	2	50	2	50	Percentage
widgetCurvedLineSeries	2	50	2	50	Series
widgetEvacuationPlans	2	59	4	50	Textual
widgetEvents	2	50	2	50	Textual
widgetExternalContent	2	50	2	50	Textual (URL)
widgetFirstAid	2	50	2	50	Textual
widgetGaugeChart	2	50	2	50	Percentage, Integer
widgetGenericContent	2	50	2	50	Percentage, Integer, Textual
widgetLineSeries	2	50	2	50	Series
widgetNetworkAnalysis	2	50	4	50	Textual
widgetPieChart	2	50	2	50	Percentage or Series
widgetPrevMeteo	2	50	2	50	Textual
widgetProcess	2	50	2	50	Textual
widgetProtezioneCivile	2	50	2	50	Textual
widgetRadarSeries	2	50	2	50	Series (table of values)
widgetResources	2	50	4	50	Textual
widgetSce	2	50	2	50	SCE
widgetSelector	2	50	4	50	Textual
widgetSeparator	1	50	1	50	Separator
widgetServiceMap	2	50	2	50	Map
widgetSingleContent	2	50	2	50	Percentage, Integer, Float

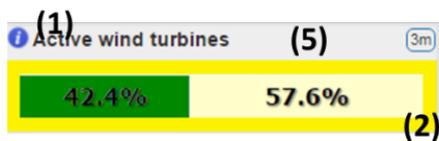
Widget name	min columns	max columns	min rows	max rows	Type
widgetSmartDS	2	50	2	50	Percentage
widgetSpeedometer	2	50	2	50	Percentage, Integer, Float
widgetStateRideAtaf	2	50	2	50	Percentage
widgetTable	2	50	2	50	Series
widgetTimeTrend	2	50	2	50	Percentage, Integer, Float
widgetTimeTrendCompare	2	50	2	50	Percentage, Integer, Float
widgetTrafficEvents	2	50	4	50	Textual
widgetTrendMentions	2	50	2	50	Textual

Some of the above listed widgets are specific for only one metric and cannot be used for others (this means that a metric can be represented in the dashboard by using only one specific widget). For example:

- *widgetFirstAid* can be selected only for the metric *FirstAid*;
- *widgetEvents* can be selected only for the metric *List_Events_FI_Day*;
- *widgetPrevMeteo* can be selected only for the metric *Prev_Meteo*;
- *widgetTrendMentions* can be selected only for the metric *MentionsTrends_FI_Day*;
- *widgetProcess* can be selected only for the metric *Process*;
- *widgetProtezioneCivile* can be selected only for the metric *ProtezioneCivile*;
- *widgetExternalContent* can be selected only for the metric *ExternalContent*.
- *widgetTrafficEvents* can be selected only for the metric *TrafficEvents*
- *widgetResources* can be selected only for the metric *Resources*
- *widgetNetworkAnalysis* can be selected only for the metric *NetworkAnalysis*
- *widgetClock* can be selected only for the metric *Clock*
- *widgetSeparator* can be selected only for the metric *Separator*
- *widgetEvacuationPlans* can be selected only for the metric *EvacuationPlans*
- *widgetAlarms* can be selected only for the metric *Alarms*
- *widgetSelector* can be selected only for the metric *Selector*

9.2. The Generic Widget properties

The Generic Widget properties area allows to define the following settings:



- (1) Title: title of the widget
- (2) Background color: background color of the widget
- (3) Content font size: font size of the values showed in the widget
- (4) Content font color: color of the font in the widget
- (5) Header color: color of the header bar

In view mode, the icon  on the top left of the widget is hidden if the Widget detailed information remains empty.

9.4. The Specific Widget properties

The **Specific widget properties** area allows to activate the Notificator tool in case values reached by the widget exceed the defined threshold.

Specific widget properties

Register to Notificator (1)

 No
 Yes
 No

Set thresholds (2)

 Yes

Notify events	Thr operator	Thr 1	Thr 2	Range color	Short description	
(3) <input type="checkbox"/>	< (4)	100 (5)	(6)	rgb (7) ■	Data center (8) temperature (°C)	(9) ✖ (10)
<input type="checkbox"/>	<	32		rgb ■	Empty	✖
<input type="checkbox"/>	<	31	49	rgb ■	Empty	✖

- (1) Set YES to activate the Notificator (see [Sec. 12](#) for additional information about the Notificator and how it works);
- (2) Set YES to set the thresholds. This allows to show the threshold colors in the widget. A new table will be shown below with the following fields:
- (3) add a tick to activate the notification for a specific event;
- (4) select from the dropdown list the operator(s);
- (5) set the threshold 1 value;
- (6) set the threshold 2 value, if needed
- (7) set the range color. It will be shown in the widget.
- (8) Write a short description, if useful
- (9) click the plus button  to add a new threshold, or the minus button to delete a threshold.

10. Details per widget

In this section, the look and feel of each widget with the associated specific parameters and properties are explained.

Every widget can represent values (typically integer numbers, float numbers, percentage, text content, series of data from a table) and can have its Specific Widget Properties.

10.1. widgetBarContent

The widget BarContent shows a horizontal green bar representing a percentage. Generic properties are explained in [section 9.2](#).

Please note that the following generic properties are not available for this widget: period, unit of measure, unit of measure position.

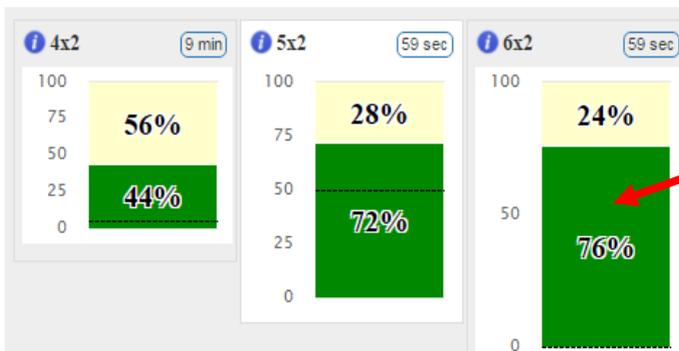


The green bar shows the percentage of the selected metric. The complementary value is automatically calculated.

For this widget, there are Specific Widget Properties available that allows to set threshold to activate notifications when certain conditions are met ([Sec. 9.4](#)). See the Section related to the Notificator for additional information ([Sec. 12](#)).

10.2. widgetColumnContent

The widget ColumnContent shows a vertical green bar representing a percentage.



The green bar shows the percentage of the selected metric. The complementary value is automatically calculated.

For this widget there are Specific Widget Properties available. See [Sec. 9.4](#) for additional information. Generic properties are explained in [section 9.2](#).

Please note that the following generic properties are not available for this widget: period, unit of measure, unit of measure position.

10.3. widgetEvents

The Events widget is specific for the metric *List_Events_FI_Day* and shows the list of public events available in Florence in the day.

The widget shows the list of events divided in two columns. In the first column on the left, for every event, there are two icons specifying the type of the event and if the event is free or not. The icons specifying the type of the event are categorized for the following typologies of events:

- Other event

- Special openings
- Estate Fiorentina
- Local fairs, markets
- Film festival
- Exhibition
- Classic music, opera and ballet
- Rock music, jazz, pop, contemporary
- News
- Readings, conferencies, congress
- Readings, book club thing
- Sport
- Theater
- Popular tradition
- Walking

Below the type of event icon, the second one provides information about the event's cost:

-  Paid event
-  Free event

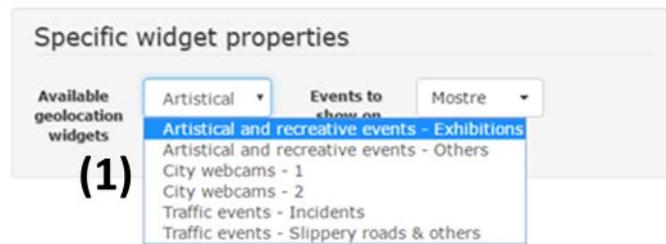


The following information are shown in the event widget:

- (1) Event title
- (2) Event date
- (3) Event location
- (4) Show location on map via cross widget
- (5) Event type icon
- (6) Event cost (paid or free)

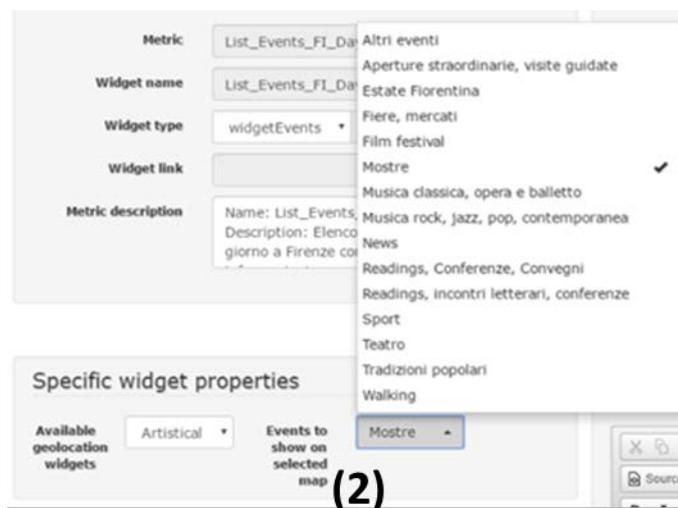
For this widget, there are **Specific Widget Properties** settings available that allows to select the destination widget where to show the location of a clicked event (via the cross-widget functionality) in a map. The following

image show the specific properties available for the *Events* widget and the list of all the available widget in the dashboard where to show the map.



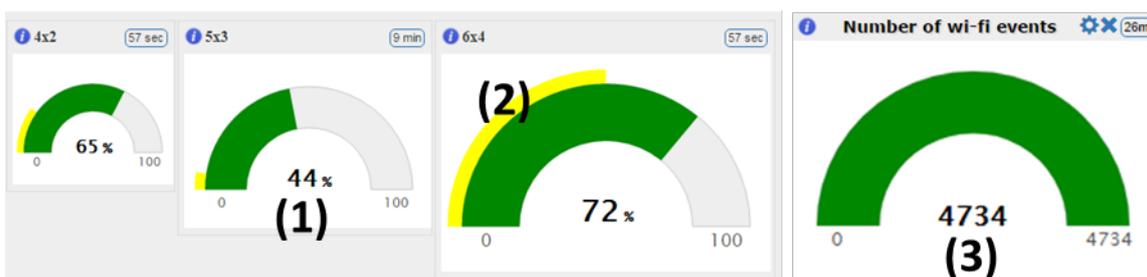
The “Available geolocation widget” drop-down menu (1), lists the titles of the available widgets the can potentially show the position of the event in a map. Substantially, the drop-down menu shows the list of titles of *ExternalContent* widgets available in the dashboard. The user has to select the widget that will show the position of the event. The interaction is activated when the user clicks on the “ON MAP” text after the address of the event.

As shown in the following image, in the *Event show on the selected map* drop down menu (2) it is possible to select the typology of event to be shown in the selected widget. Multiple selections are admitted. This means that different types of events, can be shown in different and independent widgets and maps.



10.4. widgetGaugeChart

The widget GaugeChart shows a semicircle representing a percentage (1) or an integer value (3). In case of an integer value, the semicircle is entirely colored.



The yellow arc in the widget represents a threshold value **(2)** that can be useful to set up notifications in case of unexpected values. Please note that actually this threshold has to be manually set in the metric definition.

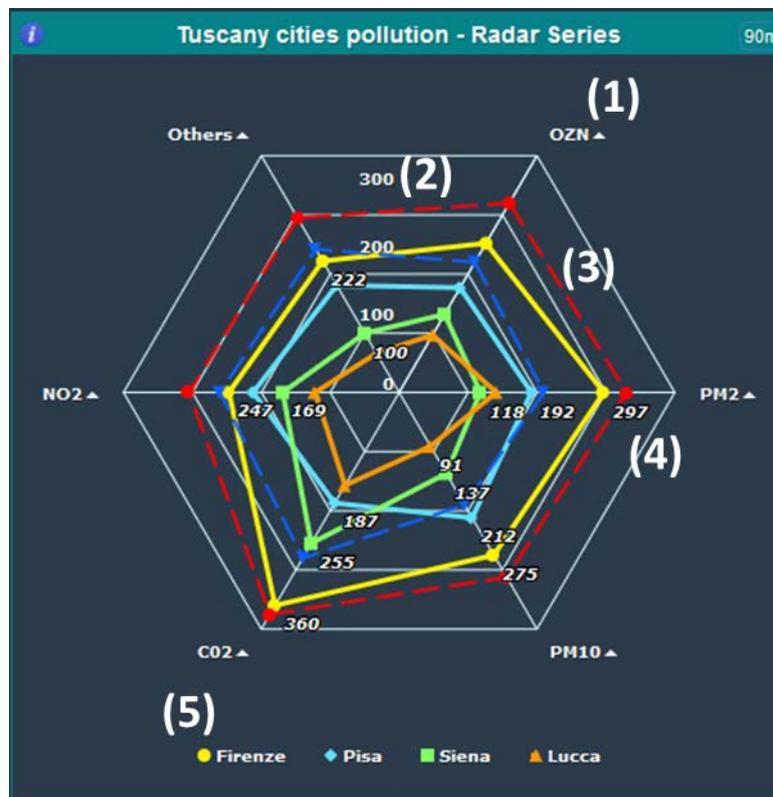
Refer to the manual on how to define metrics for more information.

Refer to section related to the notification tool for more information about the notification settings.

For this widget there are Specific Widget Properties available. See [Sec. 9.4](#) for additional information. Generic properties are explained in [section 9.2](#).

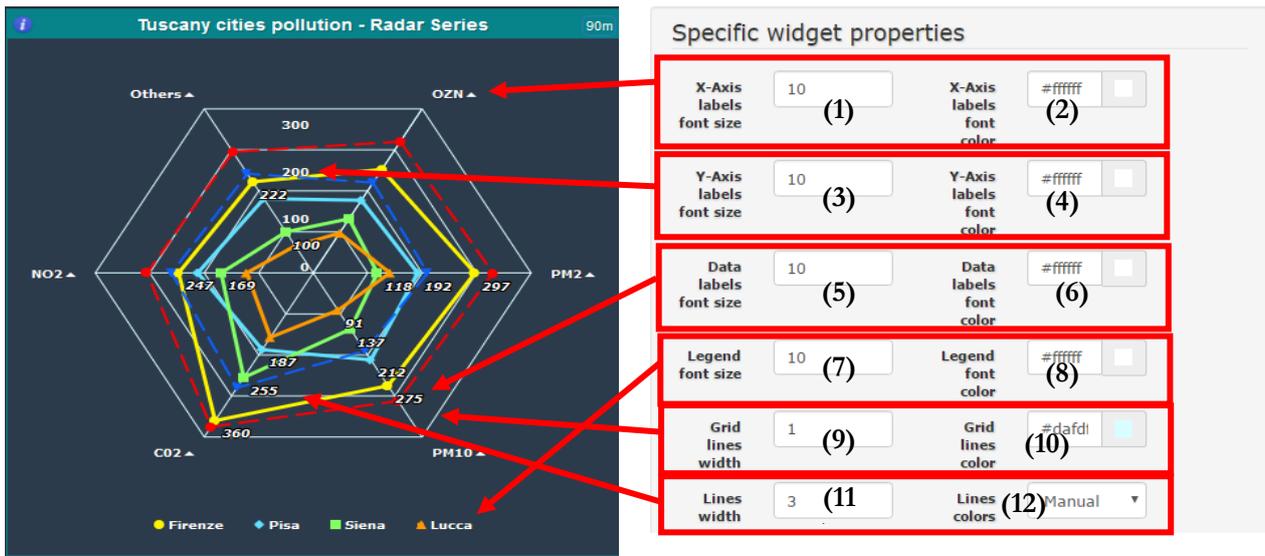
10.5. widgetRadarSeries

The RadarSeries widget allows to compare series of data in a radar chart representation. The following image shows an example of RadarSeries widget clarifying how the chart has to be interpreted.



- (1) The names in the angles of the geometric figure represent an observed quantity.
- (2) Every line (from the centre to the the angle) represents an axis but **note**: the centre is the origin (smaller values are closer to the centre).
- (3) You can also see the observed values.
- (4) In some cases you can see one or more dashed lines. The dashed lines represent the thresholds and you can add them in the *Specific Widget Properties*.
- (5) The legend associates a different color to a different context in which the quantities are observed.

In the Specific Widget Properties box, you can change the X-Axis labels properties: font size **(1)** and color **(2)**. The X-Axis labels are the labels in the angles of the geometric figure representing an observed quantity (e.g. CO2).



The Y-Axis labels are the labels of some reference values in the figure (eg. 0, 100, 200, etc.) and you can change the font size **(3)** and color **(4)**. The Data are the observed values quantity and it is possible to change the font size **(5)** and color **(6)**.

It is possible to change the Legend font size **(7)** and its color **(8)**.

The grid helps to give you some reference points; you can choose its lines width **(9)** and its lines color **(10)** and the lines width **(11)** and colors **(12)** of the lines referred to the values.

Lines width	3	Lines colors	Manual
Series	Color		
Firenze	#fef400	<input type="checkbox"/>	
Pisa	#60e2ff	<input type="checkbox"/>	
Siena	#82ff63	<input type="checkbox"/>	
Lucca	#ff9900	<input type="checkbox"/>	

The colors of the lines representing metric values in the radar chart, can be set automatically or manually, as shown in the following image. By selecting the manual color setting, the box will show values names represented in the selected metrics with the possibility to define one specific color for each of them.

An important feature of this widget is the possibility to define thresholds that will be shown as dashed lines in the radar chart. In the following, how to set thresholds in the Radar widget is explained.

The screenshot shows a configuration form for thresholds. At the top, there is a 'Set thresholds' dropdown menu set to 'Yes' (1) and a 'Thresholds lines width' input field set to '2' (2). Below this is a table with columns for 'Color', 'Short description', 'OZN limit', 'PM2 limit', 'PM10 limit', 'CO2 limit', 'NO2 limit', and 'Others limit'. The first row has a color of '#0e60ff' (4), a short description of 'OK' (5), and limits of 220, 210, 190, 280, 260, and 242 (6). The second row has a color of '#ff0000' (4), a short description of 'Warning' (5), and limits of 320, 330, 312, 375, 305, and 295 (6). A '+' button (3) is to the left of the table, and a '-' button (7) is below it.

To set threshold values Switch the *Set thresholds* drop down menu from *No* to *Yes* (1), then:

- Set the *Thresholds lines width* (2).
- Press the button **+** (3).
- Choose the color of the dashed line (4).
- Write a short description (You can leave default value *Empty*) (5).
- Choose the limit for every quantity in the table (You can leave default 0) (6).
- Remember to press *Modify* to save the new widget properties.
- To remove the thresholds, press the button **-** (7).

10.6. widgetGenericContent

By using this widget it is possible to show up to three values from different metrics.



It's composed of two parts:

- Metric's name (left side).
- Metric's observed value (right side).

To use this widget, you have to follow the following steps:

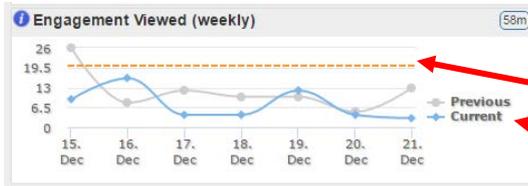
1. Click the button to add a new widget in the dashboard;
2. In the metric field, select on numeric metric (1) and press Add (2);
3. In the Widget type drop down menu select widgetGenericContent (3);
4. Select another metric in the metric drop down menu and press Add;
5. You can have up to three different metrics represented in this widget.
6. Set the Generic widget properties as you prefer;

For this widget there are not Specific Widget Properties.

The dialog box shows the 'Metric and widget choice' section. The 'Metric' dropdown is set to 'Ataf_Rt' (1) and has an 'Add' button (2). The 'Metric description' field contains the text: 'Description: Percentuale dei segnali in tempo reale che vengono da ATAF.' The 'Widget type' dropdown is set to 'widgetBarCor' (3) with a note 'max 1 metrics'. The 'Selected metrics' list includes 'widgetBarContent', 'widgetColumnContent', 'widgetGaugeChart', 'widgetGenericContent' (highlighted), 'widgetTimeTrendCompare', 'widgetPieChart', 'widgetSingleContent', 'widgetSpeedometer', and 'widgetTimeTrend'. The 'Widget link' field is empty.

10.7. widgetTimeTrendCompare

This widget is useful to show the historical values evolution of a metric during a period of time in comparison with a previous period (in the same time slot). The historical values of the previous period are represented with a light grey line.



- **Current trend**
- **Previous trend**
- **Trend thresholds**
- **Legend** (show the **current trend's** color and the **previous trend's** color).

Generic widget properties

Title	<input type="text"/>	Background color	<input type="text" value="#ffffff"/>
Content font size	<input type="text" value="11"/>	Content font color	<input type="text" value="#666666"/>
Header color	<input type="text" value="#eeeeee"/>	Header text color	<input type="text" value="#000000"/>
Period	<input type="text" value="4 Hours"/>	Refresh rate (s)	<input type="text" value="60"/>
Height	<input type="text" value="12 Hours"/>	Width	<input type="text" value="2"/>
U/M	<input type="text" value="Weekly"/>	U/M position	<input type="text"/>

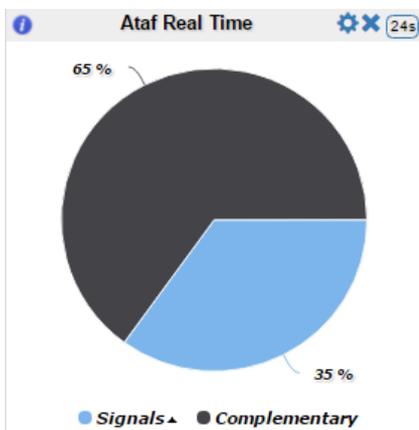
Please note that the trend threshold has to be set manually during the definition of the metric.

In the Generic properties it is possible to set the period of time to be show in the widget (4 hours, 12 hours, daily, weekly, monthly, annually) as shown in the image on the left.

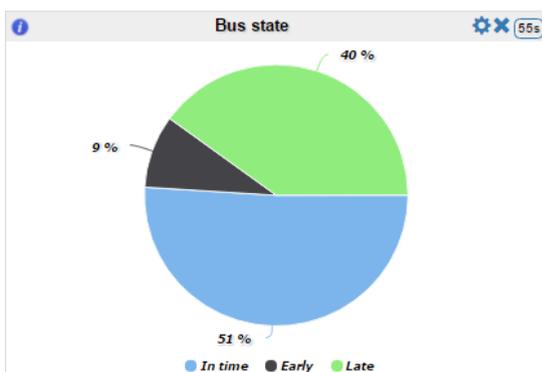
For this widget there are Specific Widget Properties available. See [Sec. 9.4](#) for additional information. Generic properties are explained in [section 9.2](#).

10.8. widgetPieChart

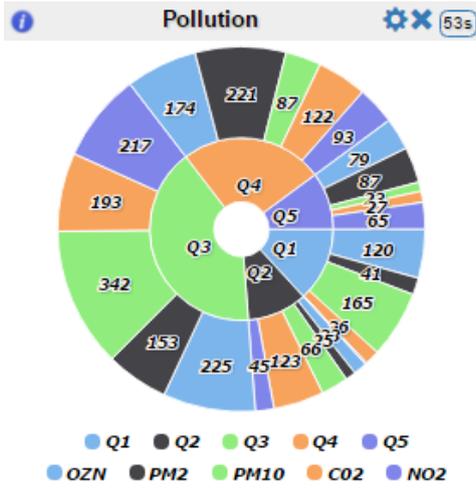
This widget can have different representations, according to the type and the number of observed data available in the selected metric to be represented.



In the case of a single value, this as to be a percentage and the Pie Chart will show the value and will automatically calculate the complementary value.



The Pie Chart widget can represent three or more values (a series)³. The legend associates a different color to a different observed value.



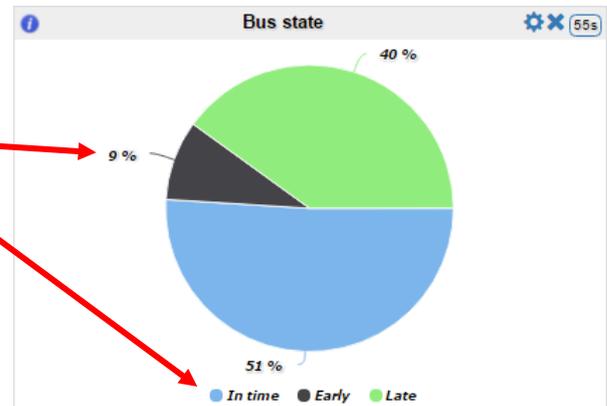
The Pie Chart widget can also represent more complex data. The image on the left shows an example of the complexity that can be represented. In this case the environmental values of different city districts are represented (e.g. the observed values of pollution for different quarters in the city).

The internal circular sectors represent the quarters while the external circular sectors represent the pollution values for each of them. In fact the two radius bordering a quarter, are also bordering the observed values of the pollution substances in the quarter.

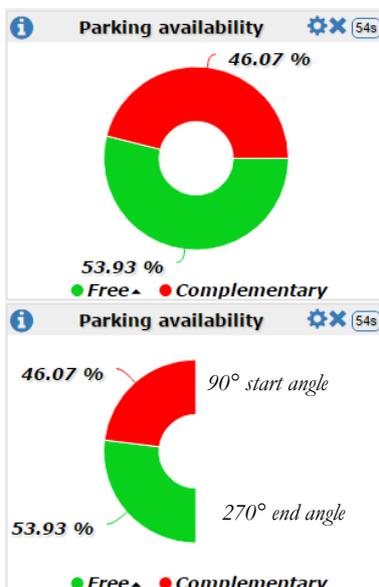
This widget has *Specific Widget Properties* that allow to define specific properties.

Specific widget properties

Legend font size: 10	Legend font color: #0000
Data labels distance: 15	Data labels font size: 10
Data labels font color: #0000	Inner radius (%): 0
Start angle (°): 0	End angle (°): 360
Diagram center Y(%): 50	Slices colors: Automati
Set thresholds: No	



- **Legend font size and Legend font color:** You can change legend font size and color.
- **Data labels distance, Data labels font size and Data labels font color:** You can change the data labels distance from the Pie Chart, font size and text color.



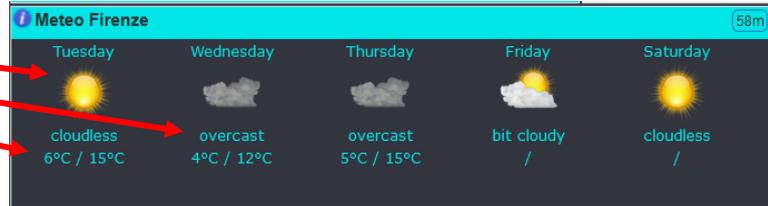
Inner radius: allow to define the radius of an empty circle inside the Pie Chart. In the image on the left an Inner radius of 40 has been defined.

Start angle (°) and End angle (°): allow to modify You can modify these two options to obtain a different circular sector. In the image on the left a Start angle of 90° and an End angle of 270° has been defined starting from the same widget represented above.

10.9. widgetPrevMeteo

This widget shows the weather of the current day and the forecast for the next four days:

- image
- a short description
- a temperature range (min/max).



The weather forecast service is provided by the [LaMMA](http://www.lamma.rete.toscana.it) consortium of the Tuscany Region (<http://www.lamma.rete.toscana.it>).

The widget visualizes the meteo forecast in different way to be specified in the *Specific Widget Properties*.

Metric and widget choice

- Metric: Previ_Meteo (1)
- Metric description: Rilevi meteo su un comune. Metric Typology: Testuale.
- City (autosuggestion) (2): Type city name's first letters
- Widget type: widgetPrevMe (max 1 metrics)
- Selected metrics: Previ_Meteo
- Widget link: none

Generic widget properties

- Title: [empty]
- Background color: rgba(255, 255, 255, 1)
- Content font size: [empty]
- Content font color: rgba(0, 0, 0, 1)
- Header color: #eeeeee
- Header text color: #000000
- Period: [empty]
- Refresh rate (s): 60
- Height: 2
- Width: 2
- U/M: [empty]
- U/M position: [empty]
- Show header: Yes
- Font type (autosuggestion): Auto

Specific widget properties

- Widget mode: Horizontale (3)
- Language: English (4)
- Today box dimension: [empty] (5)
- Background mode: Color (6)
- Weather symbols: Images (7)

Widget detailed informations

[Rich text editor toolbar with Source, Bold, Italic, Underline, Link, Unlink, Text color, Background color, Styles, Format, and Help icons]

To add the widget PevMeteo in a dashboard, click the Add Widget button and follow the following steps:

- Select the Previ_Meteo metric in the Metric drop-down menu (1);
- Write the name of the city to be monitored (autosuggestion functionality available in the field) (2);
- In the Specific widget properties, select the widget orientation in the drop-down menu (among: Horizontal, Vertical, Vertical Compact, Today only) (3)
- Select the language of the text (Italian or English) (4);
- Define the dimension in percentage of the Today box with respect the area of the widget for the next days forecast (5)
- Define the Background mode (single color or automatic) (6). The images below show examples of automatic background that are shown on the basis of the real time meteo situation.
- Select the weather symbols to be shown in the widget (images or icon).



One-day meteo



Horizontal meteo forecast



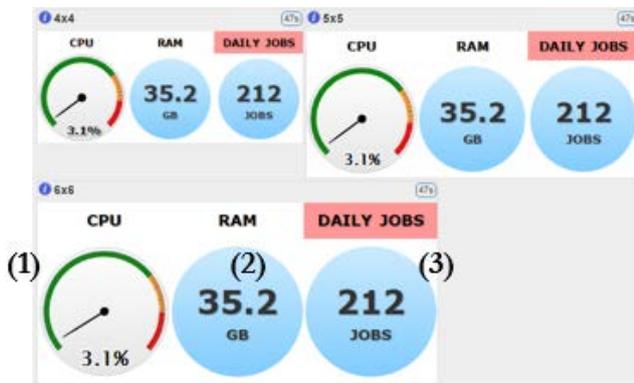
VerticalCompact meteo forecast (with icons)



Vertical meteo forecast

10.10. widgetSce

This widget shows information related to the Km4City data center.



- **CPU (1):** The percentage of CPU utilization.
- **RAM (2):** The amount of RAM utilization, expressed in GigaByte.
- **DAILY JOBS (3):** The number of daily executed jobs.

For this widget there are not yet Specific Widget Properties.

10.11. widgetSingleContent

This widget shows a single observed value. The *Generic widget properties* allow to define some settings like the background color, the font size and color, the refresh rate, the width and the height and the Unit of Measure (U/M) that can be shown next to the value or below it.

Generic widget properties

Title	<input type="text" value="Humidity"/>	Background color	<input type="text" value="#71645e"/>
Content font size	<input type="text" value="24"/>	Content font color	<input type="text" value="#ffffff"/>
Header color	<input type="text" value="#e0e0e0"/>	Header text color	<input type="text" value="#000000"/>
Period	<input type="text" value=""/>	Refresh rate (s)	<input type="text" value="60"/>
Height	<input type="text" value="2"/>	Width	<input type="text" value="4"/>
U/M	<input type="text" value="%"/>	U/M position	<input type="text" value="Below val"/>



Same widget with the *Unit of measure position* below the value



For this widget there are *Specific Widget Properties* available. See [Sec. 9.4](#) for additional information. Generic properties are explained in [section 9.2](#).

10.12. widgetSmartDS

This is a widget is specific for the SmartDS_Process metric and is showing an example of a decision process generated by the Smart Decision Support tool (<http://smartds.disit.org/dss/>). Click the following link for additional information regarding SmartDS tool: www.disit.org/6711.



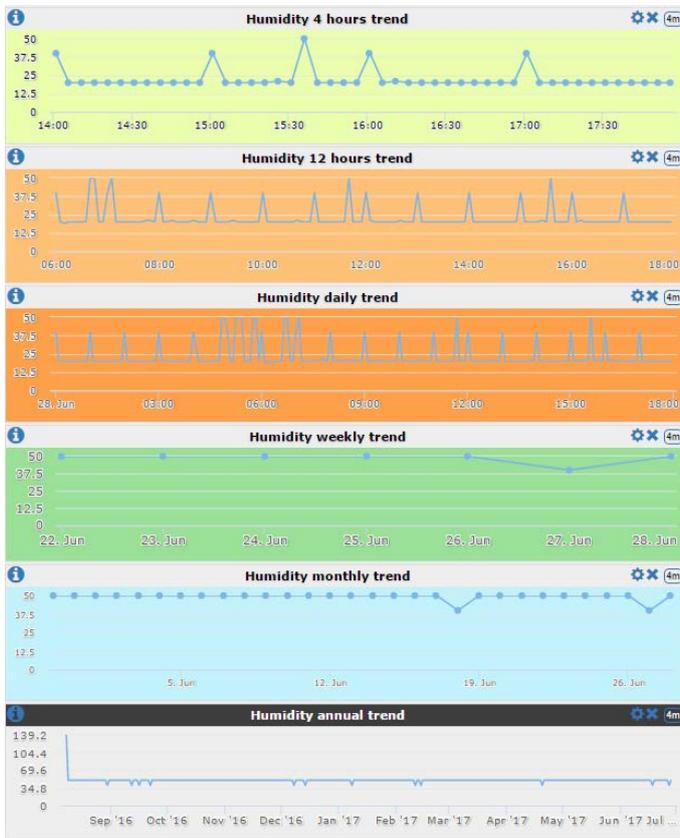
10.13. widgetSpeedometer



This widget shows a *Metric* value (Percentage, Integer, Float) represented in a speedometer.

For this widget there are *Specific Widget Properties* available. See [Sec. 9.4](#) for additional information. Generic properties are explained in [section 9.2](#).

10.14. widgetTimeTrend



This widget is useful to show the historical values evolution of a metric during a period of time.

In the Generic properties it is possible to set the period of time to be show in the widget (4 hours, 12 hours, daily, weekly, monthly, annually).

As shown in the image on the left, it is possible to specify different background color and font sizes.

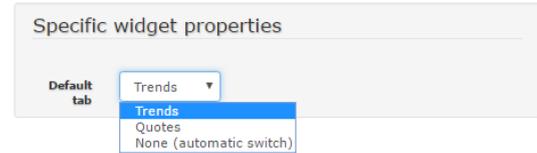
For this widget there are *Specific Widget Properties* available. See [Sec. 9.4](#) for additional information. Generic properties are explained in [section 9.2](#).

10.15. widgetTrendMentions

This widget shows the Twitter trend of a particular Twitter Vigilance channel (<http://www.disit.org/rttv/>). Actually, this widget is set for a specific channel of the Twitter vigilance tool referred to Florence. For additional information regarding the Twitter Vigilance tool, you can visit the following link: www.disit.org/drupal/6793

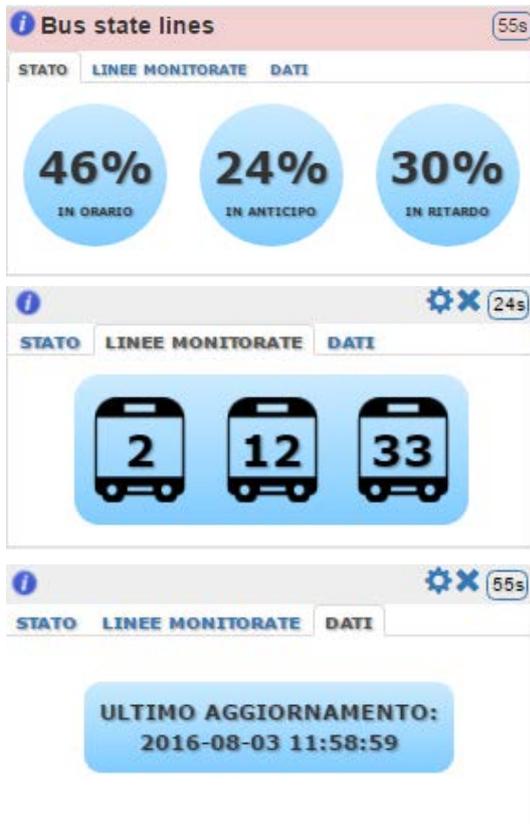
This widget shows two tabs named *Trends* (the most used hashtag in the monitored Twitter Vigilance channel) and *Quotes* (the most cited users in the monitored Twitter Vigilance channel).

In the Specific Widget Properties it is possible to set if the widget has to show by default the Trends tab, the Quotes tab or automatically slide the two tabs.



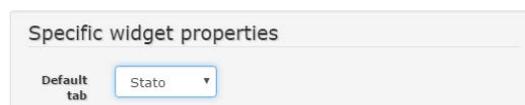
10.16. widgetStateRideAtaf

This widget shows the bus state lines through a tab having three different configurations :

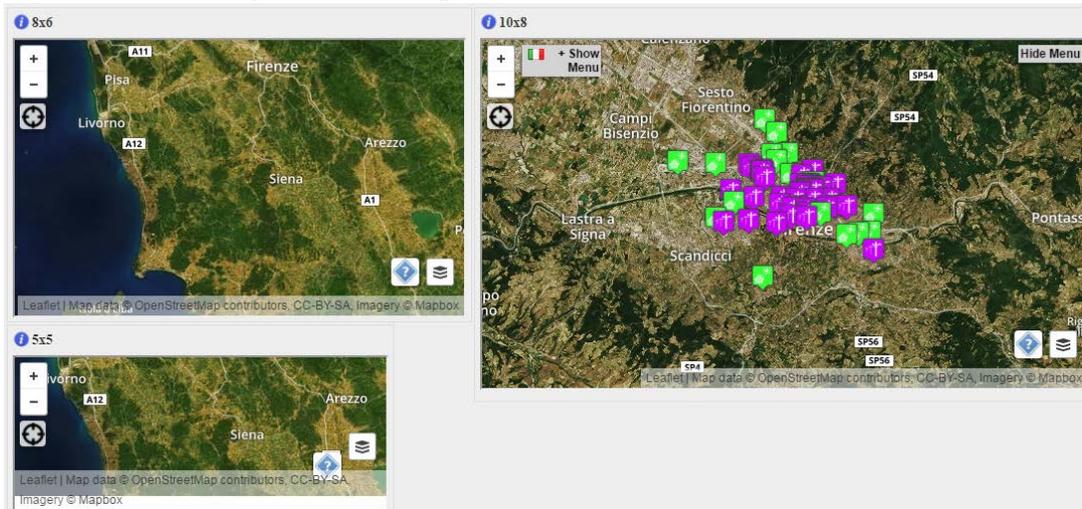


- **Bus lines State:** This tab configuration shows the percentage of bus in time, early bus and late bus.
- **Monitored Lines:** This tab configuration shows the observed bus lines.
- **Data:** This tab configuration shows the time of the last update.

In the *Specific Widget Properties*, it is possible to modify the *Default tab's* configuration through the drop down menu by selecting the different options available: Bus state, Lines, Data, None.



10.17. widgetServiceMap



10.18. widgetButton

The Button widget can:

- 1) open an external link in a new webpage;
- 2) activate a metric in a widget in the same dashboard.

The widget button can have a plain color or an image as background, and the border can be rounded by setting the broder radius.

The following are the available settings:

1. Select the *Button* metric from the drop-down menu and press the *Add* button
2. Add here the link to be opened by clicking the button.
3. Button border rounding in percentage
4. Select yes if you want to add an image as background of the button
5. Upload the background image
6. Resize the image width
7. Resize the image height
8. Select in the list the destination widget that has to be updated when the button is clicked
9. Select the metric to be shown in the previously selected widget
10. Show or Hide the text (that is the title (12)) in the button
11. Here is possible to set the widget where to show the external link. The destination widget has to be External Content. All widget ExternalContent (see [Sec. 12.21](#)) available in the dashboard will be shown in this list. If no ExternalContent widgets are available in the dashboard, this field is set to “none”.
12. Title of the button (mandatory)
13. Set *Yes* to show the widget header; *No* to hide the widget header.

Metric and widget choice

Metric: **(1)**

Metric description: Description: .
Metric Typology: Button.
Data Area: null.

Widget type: widgetButton | max 0 metrics

Selected metrics: Button

Widget link: none **(2)**

Generic widget properties

Button text: Button **(12)** | Button color: rgba(255, 0, 0, 0.5)

Content font size: 14 | Content font color: rgba(0, 0, 0, 1)

Header color: #fff | Header text color: #000

Period: No | Refresh rate (s):

Height: 2 | Width: 6

U/M: | U/M position:

Show header: Yes **(13)** | Font type (autosuggestion): Times New Roman

Specific widget properties

Border radius: **(3)** | Button image: Yes **(4)**

Image file: **(5)**

Image width: **(6)** | Image height: **(7)**

Available widgets for metric change: **(8)** | Available metrics: Humidity or Data center temperature widget **(9)**

Show button text: Yes **(10)** | Available widgets to show link content: None **(11)**

Widget detailed informations

(Rich text editor interface)

The following are examples of buttons activating the cross widget interaction (by pressing a button, another widget in the same dashboard is updated with the requested information).



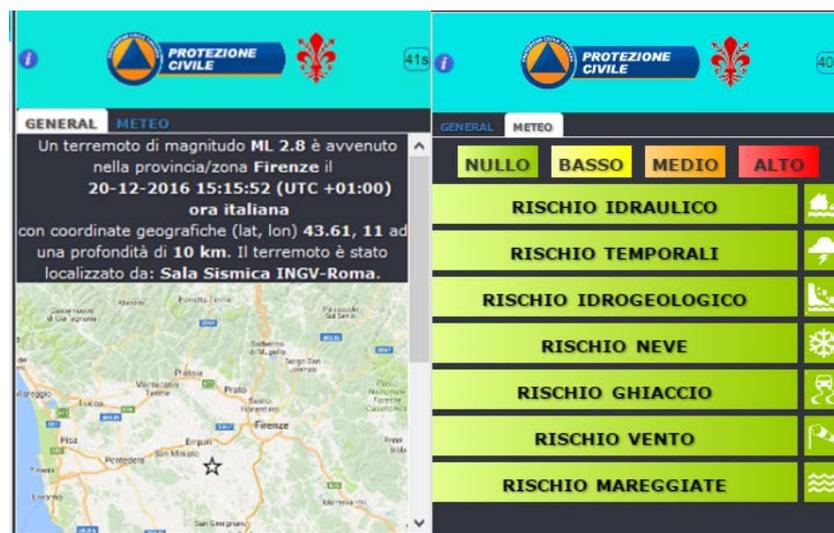
10.19. widgetProcess

This widget is specific for the monitoring of the processes managed in the DISIT cloud infrastructure.



10.20. widgetProtezioneCivile

This widget reports the Civil Protection alerts in real time. It is composed by two tabs, General and Meteo, that provides different information. The General tab provides information on alters coming from the Civil Protection. The Meteo tab provides information related to specific risks and the respective grade of severity (expressed by the following colors: green (no risk), yellow (low risk), orange (medium risk), red (high risk)).



To add this setting proceed as follow:

1. Add a new widget in the dashboard;
2. In the Metric field select the *ProtezioneCivile* metric and press the *Add* button (1). The *Widget type* field will be automatically set;
3. In the Specific widget properties tab, the “Default tab” drop down menu allows to select wich information has to be shown by default in the widget (the General tab, the Meteo tab or an automatic switch of the both tabs) (2);
4. Define the font size of the Meteo tab (3);
5. Define the font size of the General tab (4);
6. Define the background color (5);
7. Define the widget dimensions (6);
8. Click the Confirm button (7).

Metric and widget choice

Metric: ProtezioneCivile (1)

Metric description: Description: Dati provenienti dalla Protezione civile. Metric Typology: Testuale.

Widget type: widgetProtezi max 1 metrics

Selected metrics: ProtezioneCivile

Widget link: none

Generic widget properties

Title: Protezione Ci Background color: rgba(1# [color])

Content font size: Content font color: Header color: #eeeeee Header text color:

Period: Refresh rate (s): 60

Height: 8 (6) Width: 4

U/M: U/M position: Font type (autosuggestion): Auto

Show header: No

(7)

Specific widget properties

Default tab (2): None (au) Meteo (3): 10

General tab font size (4): None (automatic switch) al (5): rgba(0,0,0)

Widget detailed informations

Rich text editor toolbar with Source, Bold, Italic, Underline, Text color, Background color, Bulleted list, Numbered list, Indent, Outdent, Undo, Redo, Styles, Format, and Help icons.

(7)

10.21. widgetExternalContent

The external content widget allows to show in a widget, content that is provided by external tools. Typically, this content can be: 1) a link to an external web page, 2) a map that shows the geolocated information selected by other widgets pointing the Map, or 3) *Selector target* that specifically shown Point of Interest in a map listed in the Selector widget (see [Sec. 12.26](#)).

- 1) Select the metric “*ExternalContent*” and click the *Add* button (1);
- 2) Define the Widget Mode (2) (Web Link, Map, Selector target). This selection affects the other *Specific widget properties* fields below;
- 3) Select *Web Link* (2) to view an external website inside the widget;
- 4) Add the Web URL in the *Widget link* (3);
- 5) Set if enable/disable the possibility to view the widget in full screen in a new tab (4);
- 6) Set if enable/disable the possibility to view the widget in full screen in a pop up (5);
- 7) Set if enable/disable the zoom functionality in edit mode (see below for further explanations) (6);
- 8) If the zoom factor has been modified, this field the zoom percentage is shown (7);
- 9) Zoom control position in the edit mode of the widget (8);

Metric and widget choice

Metric: ExternalContent (1) Add

Metric description: Description: Visualizzazione di contenuti provenienti da siti esterni.

Widget type: widgetExtern. max 1 metrics

Selected metrics: ExternalContent

Widget link: <http://www.disit.org/siimobilitytra> (2)

Generic widget properties

Title: External cont (9) Background color: #eeeeee

Content font size: Content font color:

Header color: #eeeeee Header text color: #000000

Period: Refresh rate (s):

Height: 14 (10) Width: 10

U/M: U/M position:

Show header: Yes (11) Font type (autosuggestion): Auto

Specific widget properties

Widget mode: Web link (3)

Enable fullscreen in new tab: Yes (4) Enable fullscreen in a popup: Yes (5)

Zoom controls visibility: Always vi (6) Zoom factor (%): 100 (7)

Zoom controls position: Top left (8)

Widget detailed informations

Source: [Rich text editor toolbar]

10.21.1. Add a traffic flow map in a widget ExternalContent

In the external widget link, it is possible to specify a parametrized URL to the Km4City traffic flow reconstruction map. Parameters have to be specified as follow.

- *map* parameter can be: streets, satellite or grayscale;
- *position* is expressed in longitude and latitude;
- *zoom* value has to be no more than 18.

The following is an example of parametrized URL.

<http://www.disit.org/siimobilitytrafficflow2/?map=streets&position=43.7722,11.2541&zoom=15>

10.21.2. Add a WiFi map in a widget ExternalContent

In the external widget link, it is possible to specify a parametrized URL to the Km4City WiFi map. Parameters have to be specified as follow.

<http://wifimap.km4city.org/wifi-firenze/ap/ap-clusters.php?title=Firenze+Wi-Fi%3A+Access+Points+Clusters+Coverage+Map%26nbsp%3B>

- radius=[valore]
- maxOpacity=[valore]
- scaleRadius=[true,false]
- useLocalExtrema=[true,false]
- APs=[true,false]
- APsSaturday=[true,false]
- APsSunday=[true,false]
- heatmap=[true,false]

- legend=hidden
- zoom: from 1 (entire world) to 22
- position=long, lat as

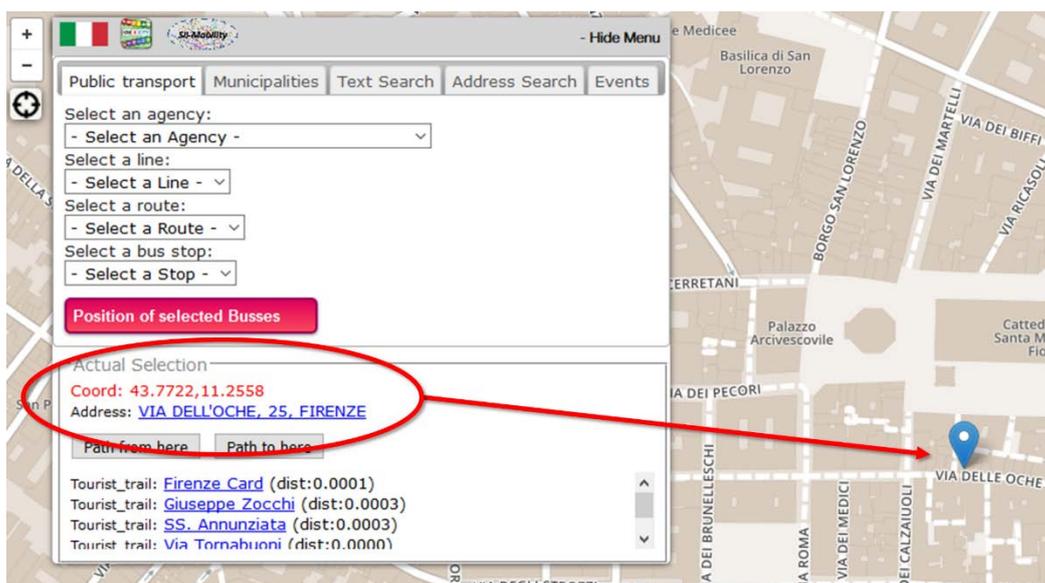
<http://wifimap.km4city.org/wifi-firenze/ap/streaming-realtime/streaming-realtime.php?title=Firenze+Wi-Fi%3A+APs+Streaming+Realtime%26nbsp%3B>

- radius=[valore]
- maxOpacity=[valore]
- heatmap=[true,false]
- streamingAPs=[true,false]
- timeRange=[5,10,15,30,60,180,360,540,720,1440,4320,10080,44640]
- distinctPeople=[true,false]
- localExtrema=[true,false]
- clusteredMarkers=[true,false]
- replySpeed=[0,10,30,60,300,900,1800,3600,7200]
- legend=hidden
- zoom: from 1 (entire world) to 22
- position=long, lat

10.21.3. How to find the position coordinates

The above mentioned parametrized links can contain a position (long, lat) to center the map in the specified point. To recover these coordinates, follow the following steps:

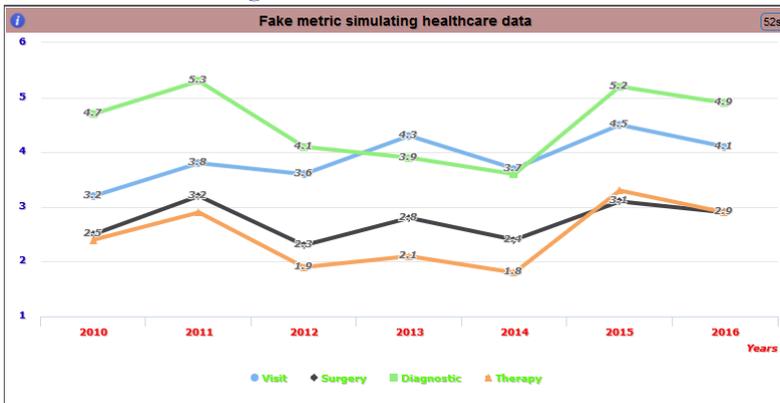
- 10) Open the service map tool (<http://servicemap.km4city.org/>);
- 11) Center the map in the point of your interest, zoom in/zoom out according to your needs
- 12) Click on the map to add a pin in a specific point that will be the center of the map view
- 13) The menu on the top left will show an area named “Actual Selection” which shows the coordinates of the clicked point and the recovered address
- 14) Copy and paste the coordinates in the parametrized link, after the “*position=*” (for example: ...&position=43.7722,11.2558&...)



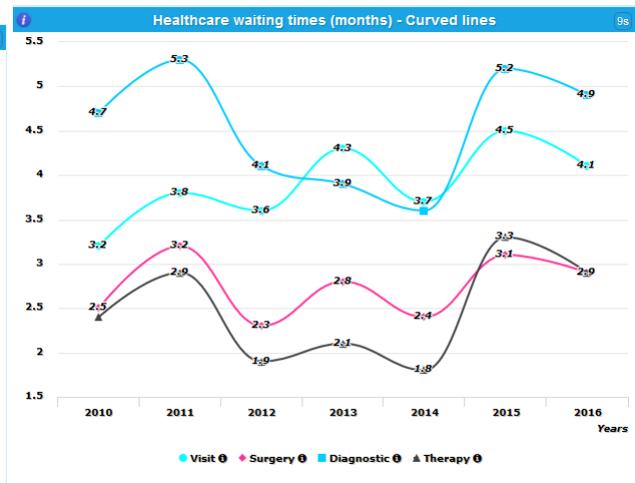
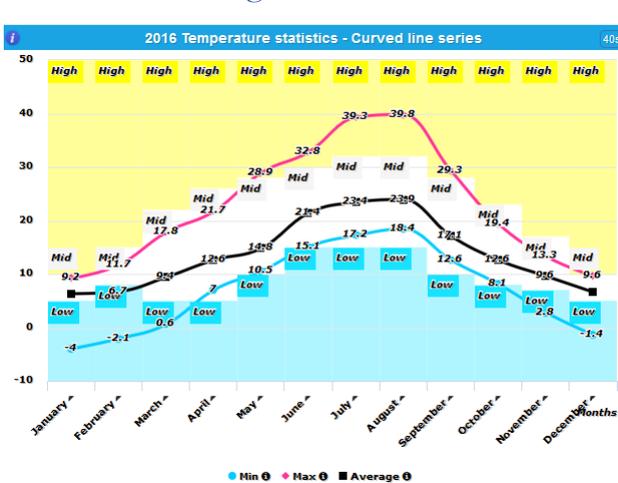
10.22. widgetTable

Data / Station	Wind speed (km/h) ▾	Direction	Temperature (°C) ▾	Humidity (%) ▾	Rain today (mm) ▾	Pressure (mbar)
Sesto Fiorentino	50	N	7	32	0	922
Livorno	65	NE	5.5	37	0	876
Grosseto	78	E	4	22	0	1022
Vada	42	S	6	0	34	895
Follonica	102	N	7.2	23	0	913
Giglio	97	O	3	19	0	957

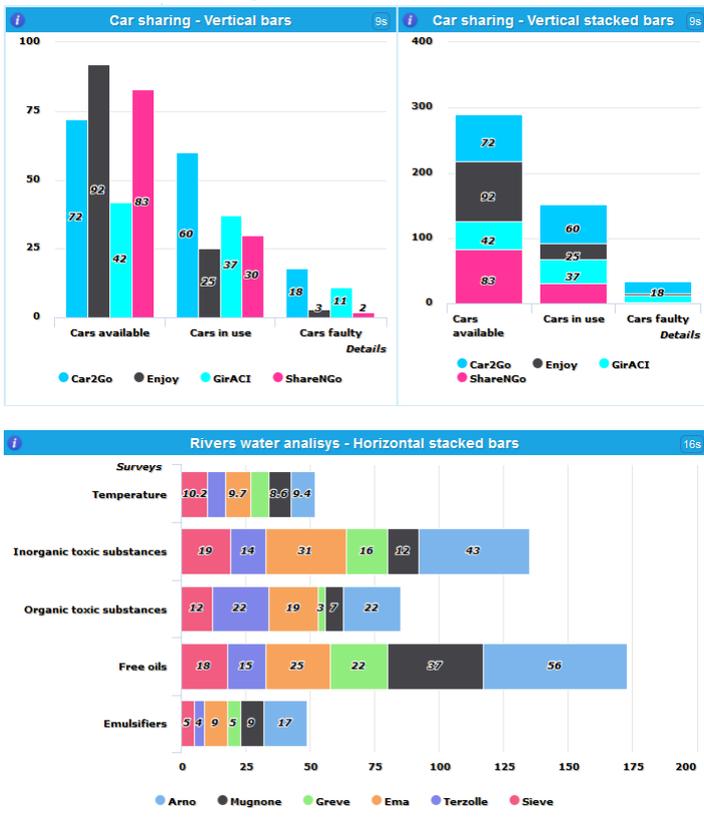
10.23. widgetLineSeries



10.24. widgetCurvedLineSeries



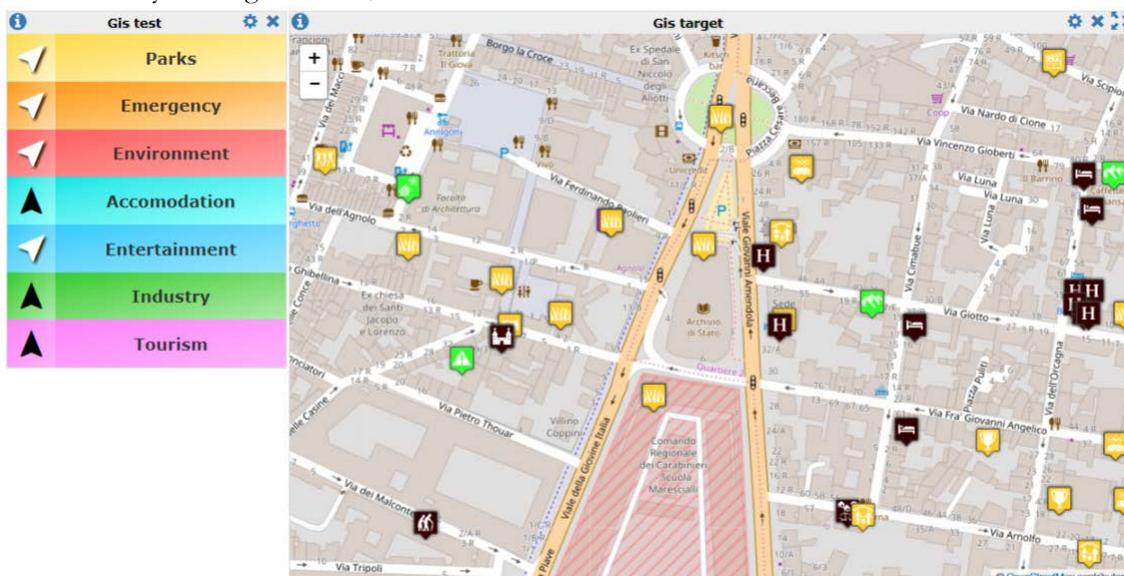
10.25. widgetBarSeries



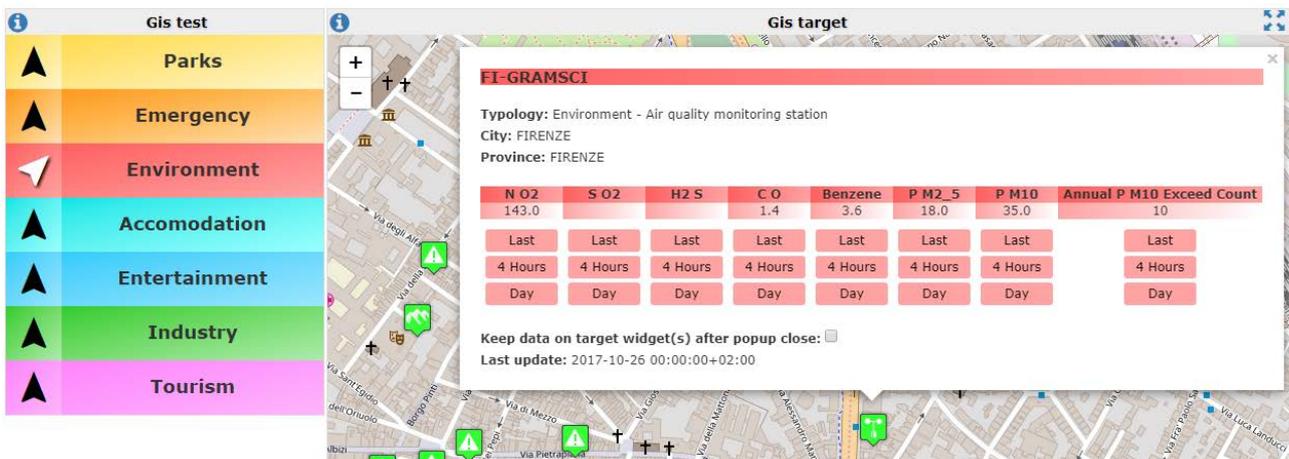
For this widget there are Specific Widget Properties available. See [Sec. 9.4](#) for additional information. Generic properties are explained in [section 9.2](#).

10.26. WidgetSelector

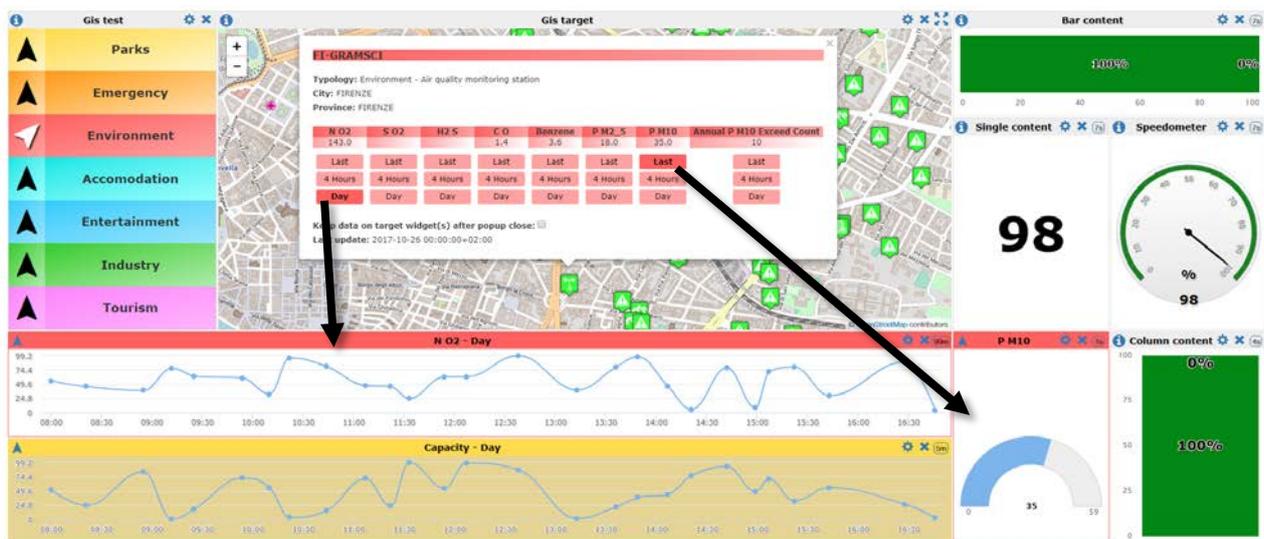
The widgetSelector is a widget that allows to show on a map the results of different queries, singularly or collectively, by clicking the service of interest (POI) on the widget. In the example below, it is possible to see a list of Point of Interest in the “Gis test” widget (Parks, Emergency, Environment, Accommodation, Entertainment, Industry, Tourism). By clicking the black arrow on the left of each POI, the services of interest are shown on the map “Gis target” in the form of geolocated pins. More than one POI can be activated or deactivated by clicking the black/white arrow.



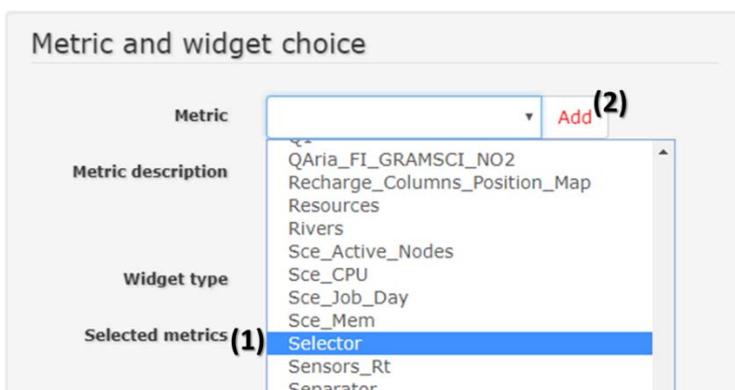
By clicking a pin of interest on the map, the information about the specific service are shown in a pop up. If the pin is related to a real-time service, all the data will be shown. In the example below, it is possible to see the real-time data of an air quality monitoring station in Florence.



The data of the pin shown in the popup, can be also represented in another widget in the same dashboard as single value (by clicking the “Last” button in the popup) or as a temporal trend (by clicking the temporal representation of interest, “4 Hours”, “Day”, “Week”, etc.). The image below show in two different external widgets the last values related to the PM10 and the Day trend of the NO2.



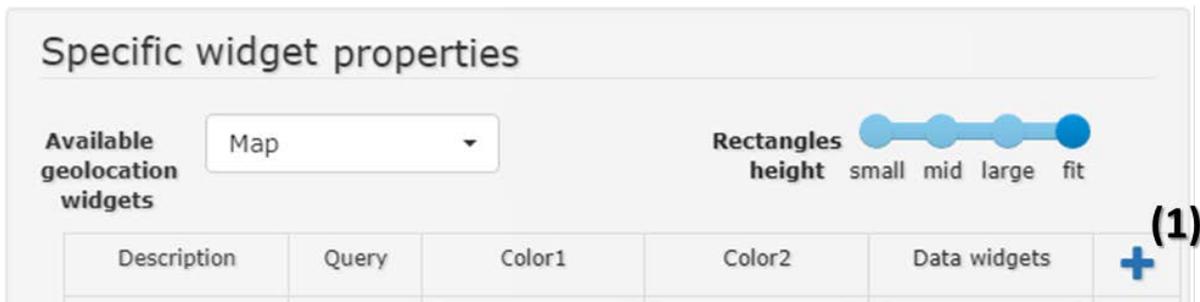
To add a WidgetSelector in a dashboard, click the Add Widget button on the dashboard and select the metric Selector in the list (1). Click the Add button (2) to confirm the selection of the correct metric.



In the Specific Widget Properties you have to:

- Select in which widget in the same dashboard the points of interest will be shown by selecting it in the “*Available geolocation widget*” drop-down menu (1).

Select the high of the single item rectangle in the list. It can be **small** (25 pixel height), **mid** (50px), **large** (75px), **fit** (the item height is automatically calculated to fit the widget height). You can see the differences in the examples below.



The items to be listed in the Selector widget, can be added by clicking the **[+]** button in the specific widget properties (1).

For each item added it is necessary to specify:

- **Description:** the name of the item that will be shown in the widget. In the example above, the descriptions are: Parks, Emergency, Environment, etc.);
- **Query:** the link to a queryID on the service map has to be copied here. More information are available below on how to obtain the query link to be copied here (see Section XXXX);
- **Color1** and **Color2**, are the two colors (expressed in hex) that can be defined by the user to create the graduated shading of the background of each item in the list. The system provides pre-defined colors that can be modified by the users. If **Color1** and **Color2** are the same, the background will be a solid color;
- **Data widget:** here you can select one or more widget(s) that will show the data available in the pop-up of the pin on the map. The example below clarifies this;
- The **[minus]** at the end of each line, delete the item in the list.

10.26.1. Add an item to be shown in the Selector widget

This is a step by step guide on how to add an item in the list to be shown in the Selector widget and to be correctly shown in a map.

- Create a new external content widget with the following settings:
 - o Widget mode: Gis target (1);
 - o Move and zoom in/out the map with the mouse or insert manually the Start lat, Start log and Start zoom values to define the initial view of the map to be shown in the widget (2).

- Define if the widget with the map can be viewed in full screen or not (3). If Yes, an icon will be shown in the right corner of the widget header to put the map view in full screen;
- Add a title (4);
- Specify the dimension of the widget (5);
- Save.

Metric and widget choice

Metric

Widget name

Widget type

Metric description

Generic widget properties

Title **Background color**

Content font size **Content font color**

Header color **Header text color**

Period **Refresh rate (s)**

Height **Width**

U/M **U/M position**

Show header

Specific widget properties

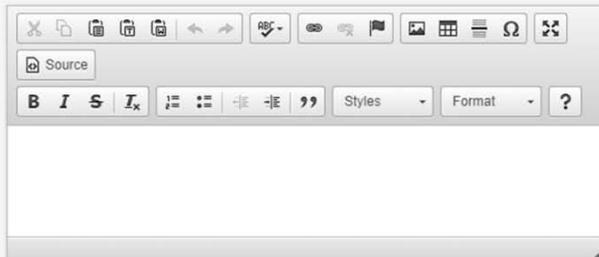
Widget mode

Start lat **Start lng** **Start zoom**

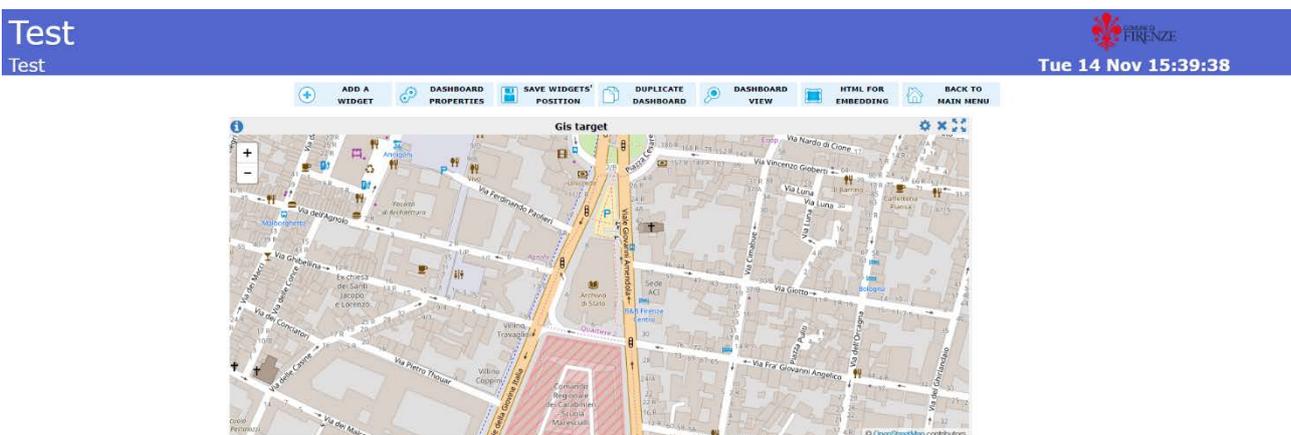


Enable fullscreen in new tab **Enable fullscreen in a popup**

Widget detailed informations

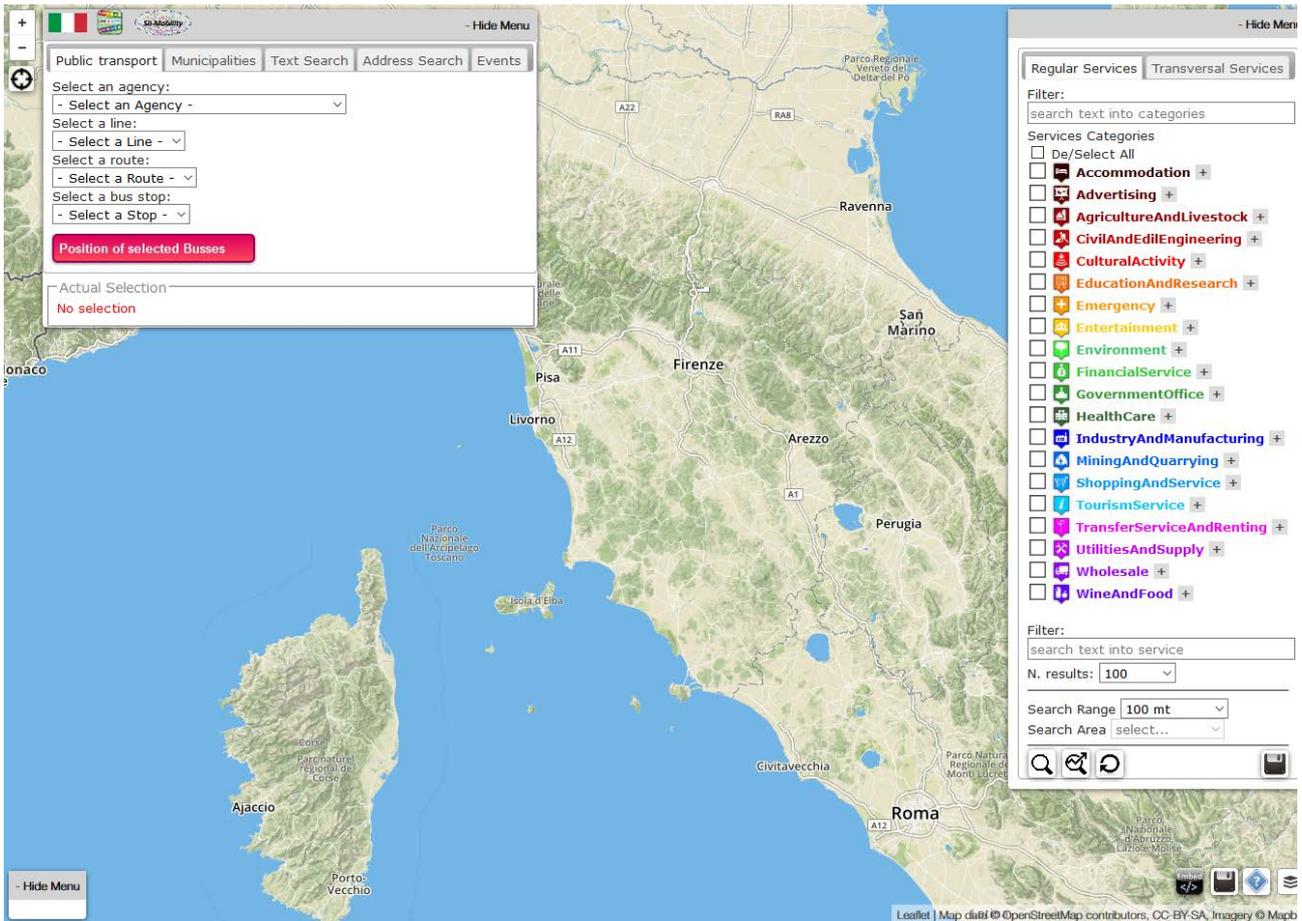


The results will be something like this:

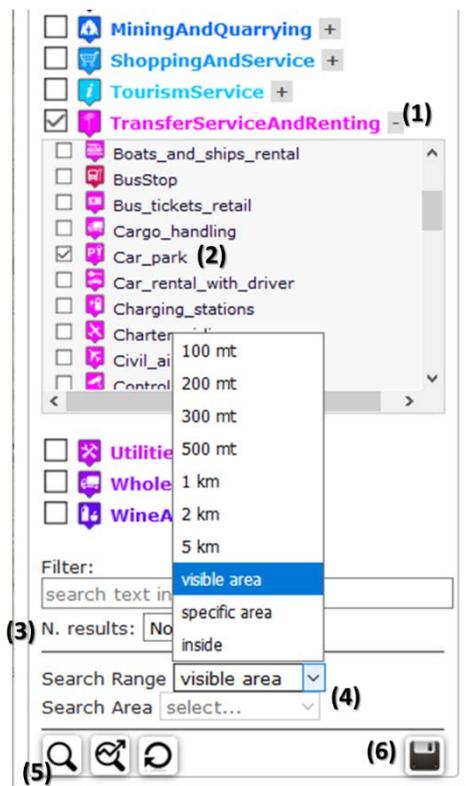


Now you have to define which items have to be shown in the Map and you need to recover the query for each of them.

Open the service map tool at: <http://servicemap.km4city.org>

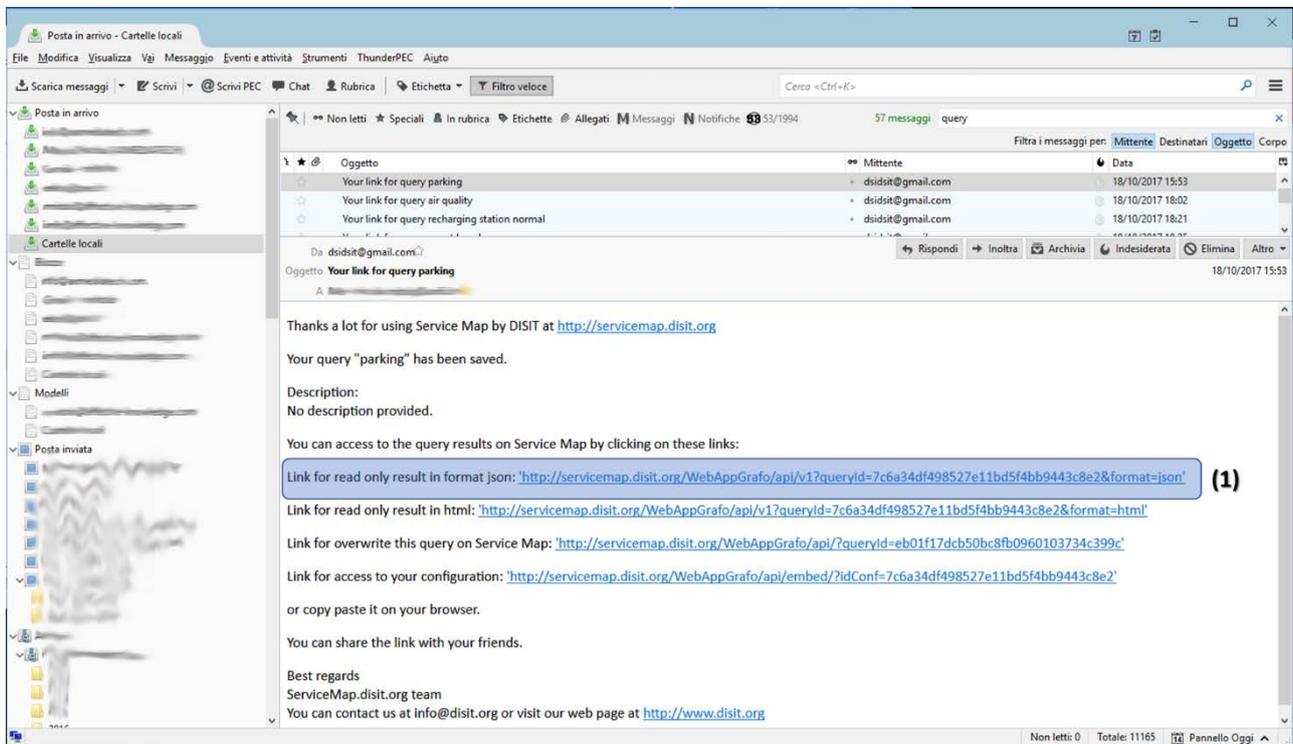


Make a geographical query by defining the services of interest. For example, to see all the parking in Florence:

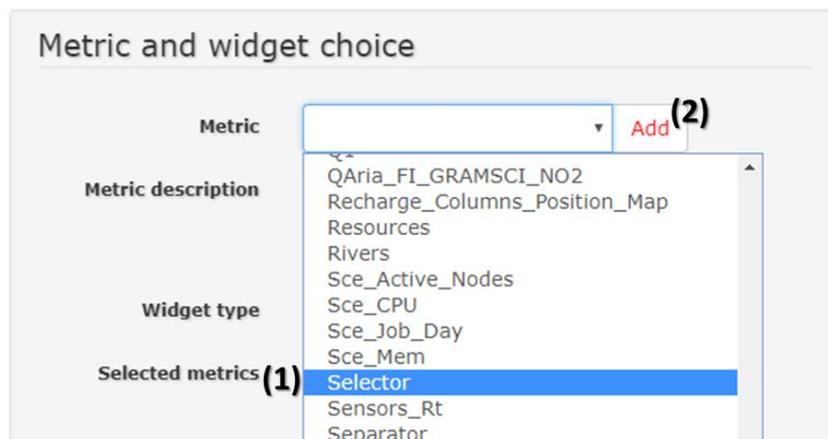


- Zoom the map and center the map view to Florence downtown
- Open the **[+]** of the category *TransferServiceAndRenting* (1) and select the *Car_park* item (2);
- Set the number of results to **[No limit]** (3);
- Set the **Search Range** to **[visible area]** (4);
- Click the lens (5) to start the search;
- All the parking will be shown on the map as geolocated pins.
- When the view is satisfactory, click on the disk (6). A pop will be shown with a form to be filled. Insert your email address where receive the query link, refine the subject of the email (not mandatory), insert a description (not mandatory) Click the **[Send]** button.
- Check your email. You will receive the email with a number of links (see the image below).

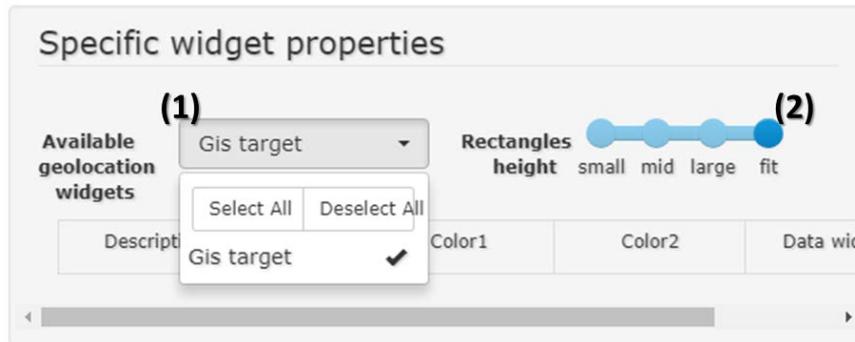
- Copy the first link (i.e. “*read only result in format json*”) without the quotation marks. The link will have the form:
`http://servicemap.disit.org/WebAppGrafo/api/v1/?queryId=7c6a34df498527e11bd5f4bb9443c8e2&format=json`
- Be sure that a / is present in the link after the “*.../v1*” and before the “*?queryId...*” (see the text of the link above highlighted in grey).



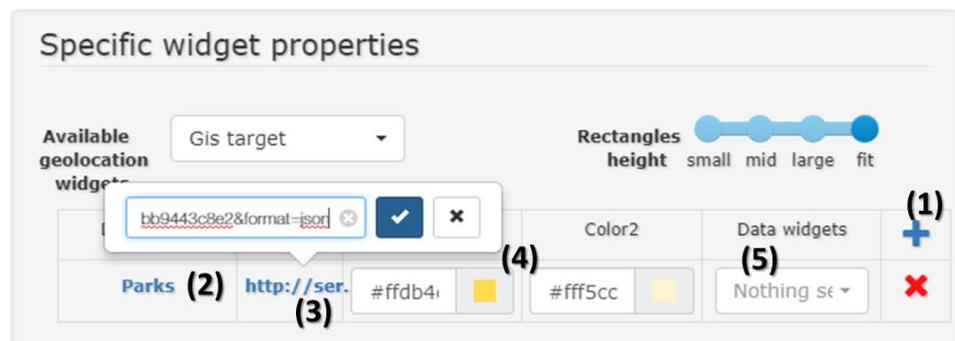
- Add a *Widget.Selector* in a dashboard: click the **[Add Widget]** button on the dashboard and select the metric **Selector** in the list (1). Click the Add button (2) to confirm the selection of the correct metric.



- In the *Specific widget properties*, open the *Available geolocation widgets* drop-down menu and select the target widget where to show the results of the queries (1). In the example only the **Gis target** widget will be available in the list. Click on it.
- Set the *Rectangle height slider* to “*fit*” (2);



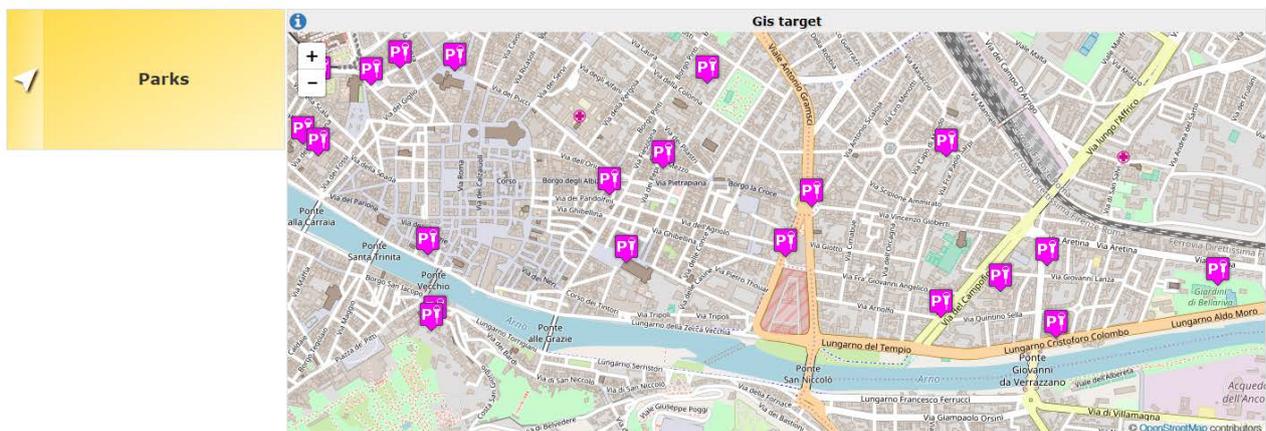
- Click the plus button to add a line (1);
- Click on the description empty field and write “Parks” and confirm by clicking the confirmation button (2);
- Click on the Query empty field and paste the link of the service map query received by email and copied before (3);
- Define Color1 and Color 2 as you prefer or leave them unchanged to accept the suggested colors (4);
- The **Data widget** drop down menu (5) can be used to specify a widget in the dashboard that will be used to visualize the data of the pin selected on the map. This allow to show in another specific widget the values or the trend of the pin selected in any preferred widget type (as a speedometer, as a bar content, a temporal trend, etc.).



- Add title and dimension of the widget in the **Generic widget properties** and Save the widget.
- The final result will be something similar to the image below. By clicking the black arrow the parking will be shown in the map “Gis target”.

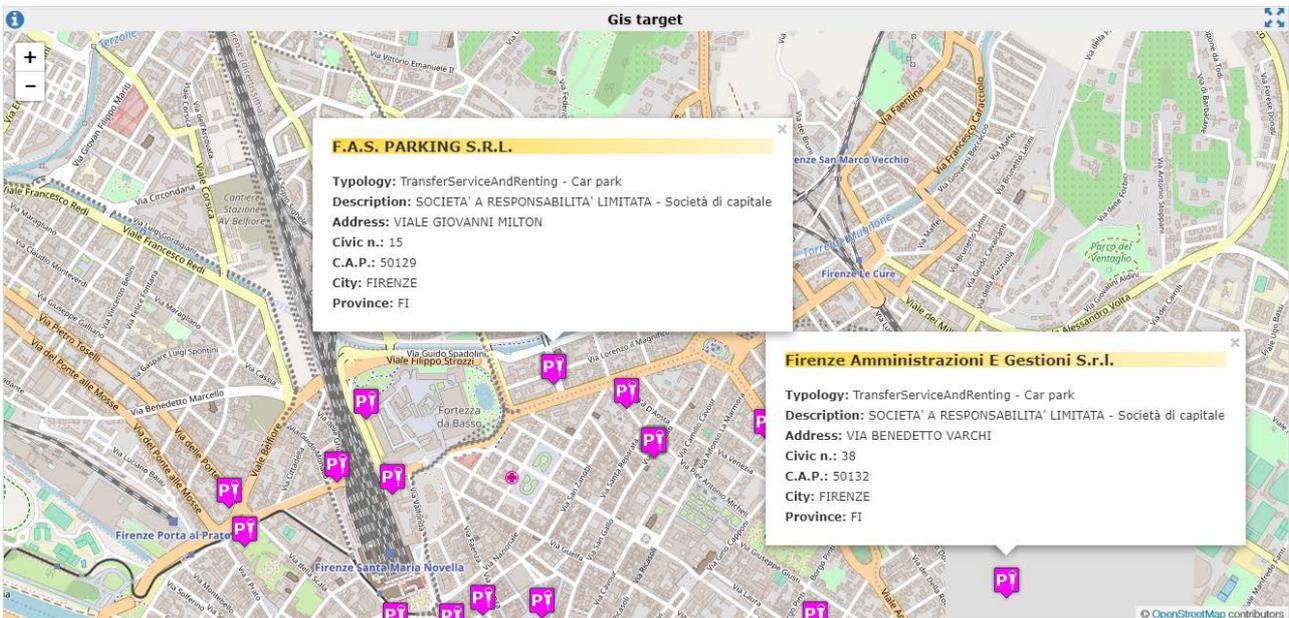
Test

Test

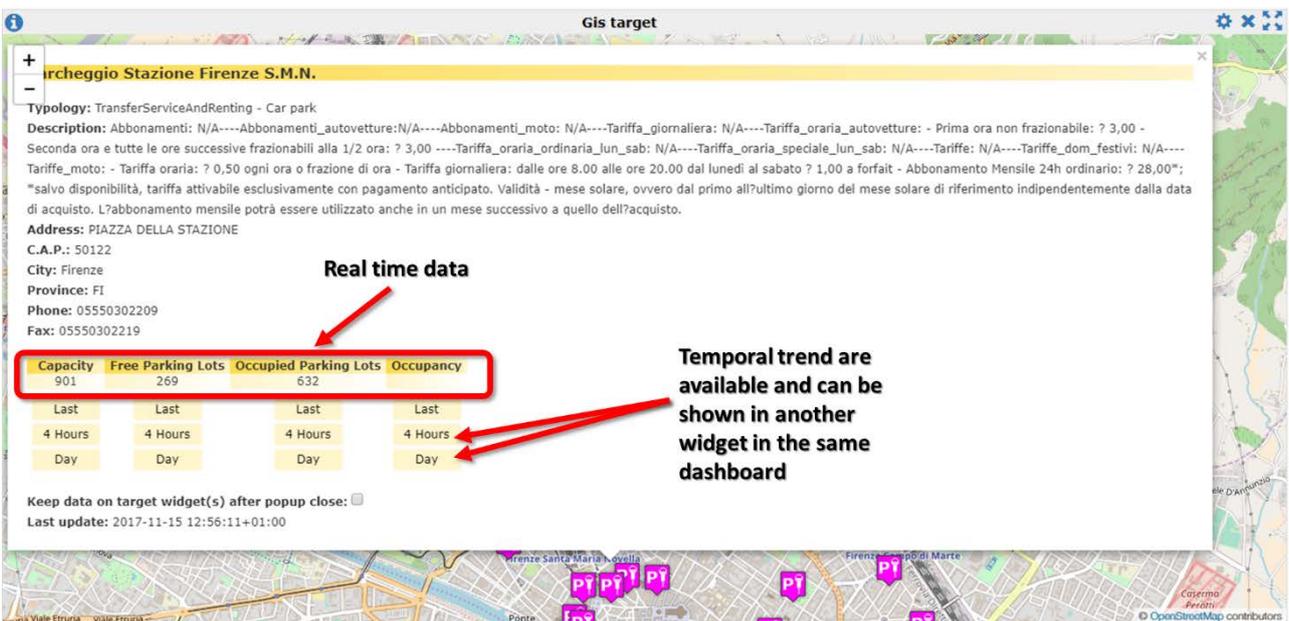


10.26.2. Show the data of a pin in another widget

By clicking a pin on the map, all information about the selected service are shown in a pop-up associated to the pin, as shown in the following image.



If the selected pin has real time data available, these data are shown in the pop-up, as in the following image.



Real time data shown in the pop-up can be visualised also in other widgets on the same dashboard. Real time data can be visualised in any widget that manage single values. Temporal trend can be visualised in any widget that manage temporal values. The destination widget(s) to visualize the pin data has(have) to be selected in the *Specific Widget Properties* tab, in the *Data widgets* column [(1) in the following image]. By clicking the drop down menu in the *Data widgets* column, the list of all the available widgets in the dashboard will be shown. One or more target widgets can be selected from the list.

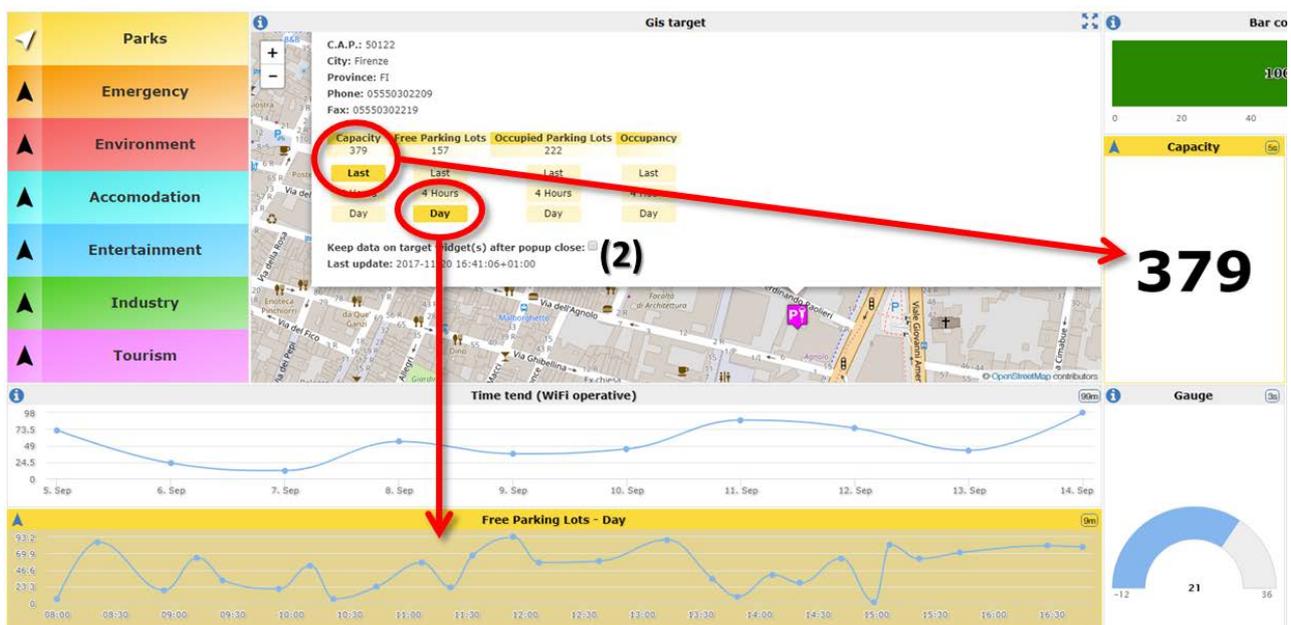
Once opened the pin with the data of the point of interest, the “Last” button in the pop-up will show the last available value in the defined destination widget (that can be any available widget on the dashboard that shows a

single value, like: single content, speedometer, bar content, etc.). Otherwise, by clicking one of the button for the temporal trend of interest, the trend will be shown in the defined destination widget (temporal trend widget).

Specific widget properties

Available geolocation widgets: Rectangles height: small mid large fit

Query	Color1	Color2	Data widgets
D://ser...	#ffdb41	#fff5cc	Single con <input type="button" value="X"/>
D://ser...	#ff9900	#ffe0b0	Select All Deselect All
D://ser...	#ff6060	#ffa3a3	Bar content
D://ser...	#00e600	#99ffff	Single content <input checked="" type="checkbox"/>
D://ser...	#33ccff	#99e6ff	Speedometer
D://ser...	#33ccff	#99e6ff	Time tend (WiFi operative)
D://ser...	#33ccff	#99e6ff	Gauge
D://ser...	#33ccff	#99e6ff	Column content
D://ser...	#33ccff	#99e6ff	Parking trends <input checked="" type="checkbox"/>
D://ser...	#fd85ff	#ffb5ff	Single con <input type="button" value="X"/>



The box “Keep data on target Widget(s) after popup close” means that the last clicked value(s) will remain in the target widget(s) also after the popup closure.

11. How to embed a dashboard in external web pages. A step by step guide.

This section explains the steps needed to embed in an external web pages, widgets and dashboards created with the Dashboard Builder tool (<http://dashboard.km4city.org/>) through the embedding feature.

In this guide will explain:

- 1) An example of a web page hosting one or more widgets created with the Dashboard Builder tool;
- 2) All the steps needed to achieve the result;
- 3) Possible applicable changes and available settings.

An example of dashboard created with the Dashboard Builder tool, hosted within a side column of the page is available in the page: <http://www.sii-mobility.org/demo/index.html>. The dashboard shown contains two distinct widgets:

- The Selector widget, which shows a list of points of interest (bus stops, bicycle paths, hotels);
- The Map widget that shows the geolocated pins on the map, based on selected points of interest.

These are just two sample widgets, but any widget available in the dashboard builder can be placed in a dashboard to be hosted on an external web page. The list and description of all available widgets are available in **Section 11** of this manual.

The screenshot shows a web page titled "DEMO 1" with a navigation bar containing "Home", "MENU 1", "MENU 2", "MENU 3", and "MENU 4". The main content area is divided into two columns. The left column contains a "Selector Widget" with the following text: "By DISIT Lab", "Questa pagina dimostrativa mostra alcune delle potenzialità offerte dal tool Dashboard Builder (<http://dashboard.km4city.org/>) di Km4City.", "Il tool Dashboard Builder permette di creare dei widget da includere all'interno di pagine web esterne tramite opportuna progettazione del widget stesso e autorizzazione alla visualizzazione all'interno della pagina specifica. I widget possono essere modificati accedendo al tool Dashboard Builder che permette di editare tutte le caratteristiche del widget, di crearne di nuovi, e di autorizzare le pagine esterne che possono visualizzare il widget stesso.", "In questo pagina di esempio, la colonna di sinistra ospita un widget esterno creato con il tool Dashboard Builder che mostra sulla mappa sottostante i punti di interesse selezionati dalle tre possibili scelte: le fermate dell'autobus, le piste ciclabili e gli hotel.", "Cliccando sulla freccia della voce di interesse, i punti di interesse vengono visualizzati sulla mappa come pin geolocalizzati." The right column contains a "MAP Widget" with a blue header "Clicca sulle frecce per visualizzare i punti di interesse nella mappa" and a list of three categories: "Bus stops", "Cycle paths", and "Hotels", each with a red arrow icon. Below the list is a map of Florence, Italy, showing geolocated pins. The map is titled "Map" and includes a search bar, zoom controls, and a "Powered by www.km4city.org" footer.

In order to host on external pages widgets created with the Dashboard Builder tool, follow the following steps:

A) Create the widget

To create a widget, access the Dashboard Builder tool by entering your user name and password.

Unregistered users who want to receive username and password to access the tool can send the request by email to info@unifi.it

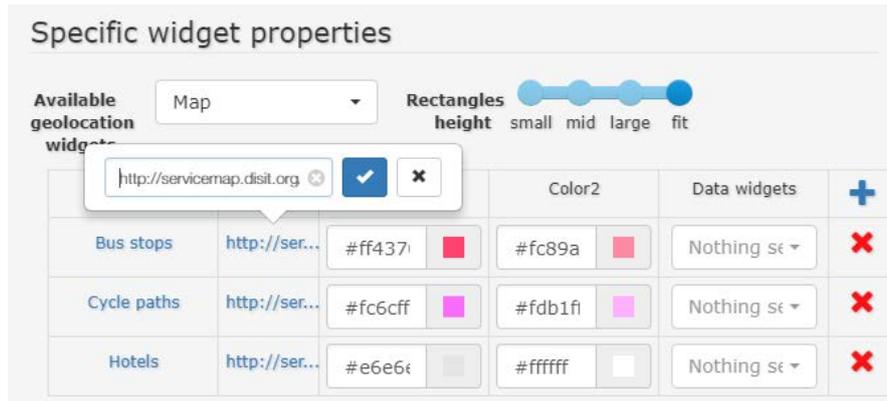
- Create a new dashboard by clicking on the **[Dashboard +]** button and set all the necessary parameters (colors, dimensions, etc.).
- The first widget to be created is that of the map:
 - o click **[Add a Widget]**;
 - o in the **Metric** drop-down menu, select **External Content** (1);
 - o Press the **Add button** (2) to confirm the selected metric;

- In the **Specific Widget Properties**, open the **Widget Mode** drop down menu and select GIS Target;
- In the **Generic widget properties**, put the title (4) and size (5) and (6);
- Press **[Confirm]** to save the widget.

The screenshot shows the 'Add new widget to dashboard' interface with four panels:

- Metric and widget choice:** Shows a 'Metric' dropdown menu with 'ExternalContent' selected (1) and an 'Add' button (2). Below it is a 'Metric description' text area and a 'Widget type' dropdown menu with 'widgetExtern' selected and 'max 1 metrics'.
- Generic widget properties:** Shows fields for 'Title' (4) set to 'Gis', 'Background color' (#eeeeee), 'Content font size', 'Header color' (#eeeeee), 'Header text color' (#000000), 'Period', 'Refresh rate (s)', 'Height' (5) set to '8', 'Width' (6) set to '4', 'U/M', and 'U/M position'. A 'Show header' dropdown is set to 'Yes'.
- Specific widget properties:** Shows a 'Widget mode' dropdown menu with 'GIS target' selected (3). Below it are 'Start lat' and 'Start lng' (11.25569) and 'Start zoom' (11) fields. A map of Florence is shown with various markers. At the bottom, there are 'Enable fullscreen in new' dropdowns set to 'No' and 'Yes'.
- Widget detailed informations:** Shows a toolbar with various icons and a 'Source' field.

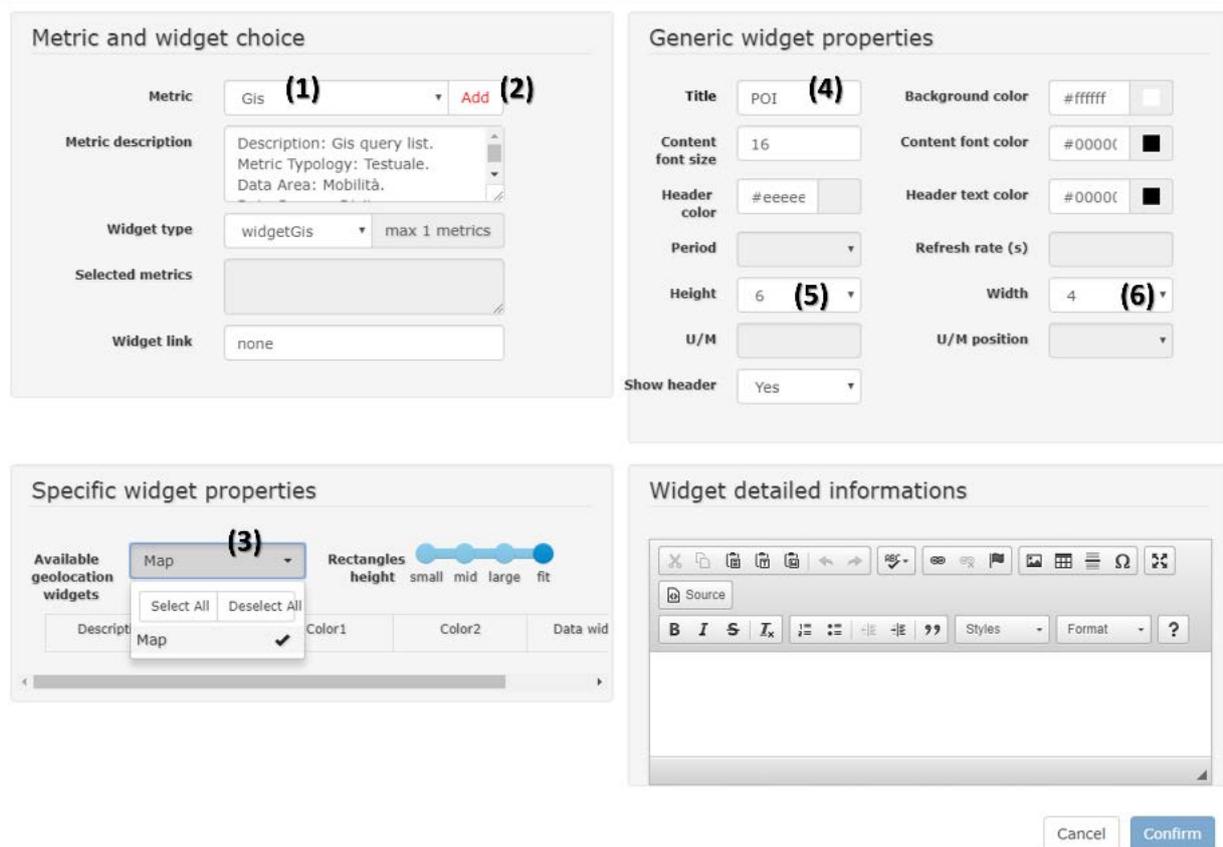
- The second Widget is the **Selector** that shows the list of points of interest to be shown on the map just created.
- click **[Add a Widget]**;
- in the **Metric** drop-down menu, select **Selector** (1);
- Press the **[Add]** button (2) to confirm the selected metric;
- In the **Specific Widget Properties**, open the **Available Geolocation Widgets** drop down menu and select the title of the previously created map widget (**Map** in the example) (3).
- Add the services you want to see in the widget by specifying the corresponding queries (see the following image as example).



Further details on how to add the services to be shown in the widget, are available in this manual in *Section 12.26*.

- In the **Generic widget properties**, add the title of the widget (4) and specify the dimensions (5) and (6);
- Press [**Confirm**] to save the widget.

Add new widget to dashboard



- Once the dashboard is completed, you must verify that it can be embedded on external pages. Open the [**Dashboard Properties**] menu at the top of the page and scroll to the **Iframe embeddability** settings box.

Iframe embeddability

Allow embed **(1)**

Authorized pages **+** **(2)**

- <http://www.effective-knowledge.c...>
- <http://www.effective-knowledge.c...>
- <http://www.sii-mobility.org/demo...>
- www.sii-mobility.org/demo/index-...

Dashboard visibility

People who can see this dashboard

- The *Allow embed* setting has to be set to Yes (1)
- Add all the URLs corresponding to the authorized web pages to be authorized. Web pages not listed here will not be able to show the dashboard.
- Confirm by clicking the *[Modify]* button.



- Select the *HTML for embedding* item in the top menu. A new window will be open allowing you to view the dashboard hosted on a simulated web page preview. By modifying the default settings shown on the left, you can see immediately the effect of the changes. Once you have the best view, you can copy the generated HTML code and paste it into your web page.

Embed this dashboard

- Select the embed mode between *Automatic* or *Manual* (1)
- In *Manual* mode, select (2) to set whether the widget should show the header or not;
- Select (3) to define whether the widget should be resized automatically (*Autofit*) based on the size of the frame that hosts it (3), or not.
- Set the size of the frame that will host the widget (4)
- Try the various settings by looking at the result in the *iframe preview* window (7) until you will reach the desired results;
- You can see the generated code in the box (5) *Code to embed this dashboard*; You can also save the generated code as a text file (txt) or as a HTML code (6).
- Copy the generated HTML code and paste it into the HTML code of the web page that will host the dashboard. Publish online the web page hosting the dashboard.

Please note that to correctly display the dashboard on external web pages, the exact link of the host page must be listed in the list of pages authorized to contain the dashboard, in *Iframe embeddability* box of the *[Dashboard Properties]*.

The following are other example of dashboards embedded in external web pages.



http://www.disit.dinfo.unifi.it

qualsiasi tipo 🔍 deep search

HOME ABOUT RESEARCH INNOVATION EDUCATION AND COURSES HOWTO EVENTS

Log in/Create account 🇮🇹

EMBEDDED DASHBOARD - TOURISM INFO NEW

▲ Bus stops	▲ Cultural activities
▲ Cycle paths	▲ Entertainment
▲ Hotels	▲ Restaurants

Map

Powered by www.km4city.org

CONTENUTI

- In primo piano
- Most Viewed (last 500)
- Most Viewed All (last 500)
- Ultimi caricati
- Più votati

CLASSIFICAZIONE

Lista dei termini

- application fields (3425)
- content kind (962)
- models and systems (3014)
- project kind (809)
- research topics (7453)
- standard (501)

www.disit.org/7100



http://www.disit.dinfo.unifi.it

qualsiasi tipo 🔍 deep search

HOME ABOUT RESEARCH INNOVATION EDUCATION AND COURSES HOWTO EVENTS

Log in/Create account 🇮🇹

DISIT LAB OVERVIEW



DISIT lab and research group is active since 1994. It is one of the most active ICT labs of the University of Florence, metropolitan Tuscany area. DISIT successfully developed a relevant number of International and National research, development and innovation projects. DISIT provides an infrastructure for cloud and distributed computing.

DISIT has coordinated a number of large EC projects, in many others has covered the role of partner, and also coordinating scientific and technical WP and performing activities of dissemination and assessment. DISIT has received a relevant number of awards and is directly involved into top level international conferences, advisory boards, and committees.

DISIT research areas: big data, artificial intelligence, natural language processing, distributed systems, formal models, metrics definition and assessment:

- Smart City integrated solutions: <http://www.disit.org/km4city>
- Data Mining and understanding: OD ingestion, quality improvement, data fusion, reconciliation: <http://www.disit.org/6056>
- Open Data: OD, LOD, RDF stores visual tools, link discovering, enrichment, such as: <http://log.disit.org>



CONTENUTI

- In primo piano
- Most Viewed (last 500)
- Most Viewed All (last 500)
- Ultimi caricati
- Più votati

CLASSIFICAZIONE

Lista dei termini

- application fields (3425)
- content kind (962)
- models and systems (3014)
- project kind (809)
- research topics (7453)
- standard (501)

Mar 14 Nov Powered by LaMMA

Firenze

Nuvoloso 9°C / 12°C

Mer 15 Nov	Gio 16 Nov	Ven 17 Nov	Sab 18 Nov
7°C / 12°C	6°C / 15°C	Temp N/A	Temp N/A

Coperto Poco nuvoloso Poco nuvoloso Poco nuvoloso

Powered by www.km4city.org

DEMO 2

Home

MENU 1

MENU 2

MENU 3

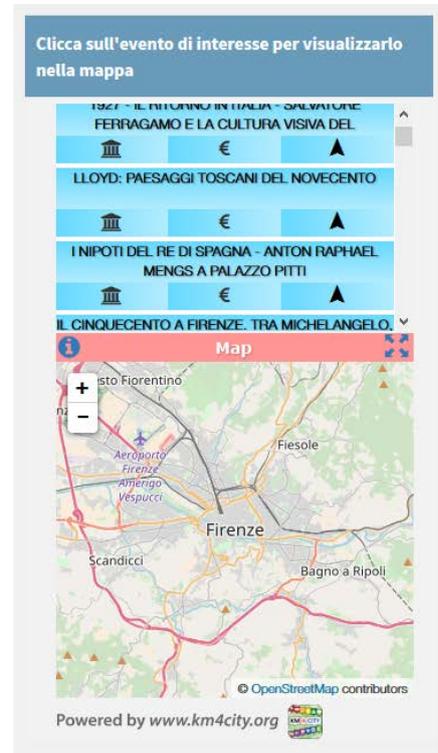
MENU 4

Pagina dimostrativa

By DISIT Lab

Questa pagina dimostrativa mostra alcune delle potenzialità offerte dal tool Dashboard Builder (<http://dashboard.km4city.org/>) di Km4City.

Il tool Dashboard Builder permette di creare dei widget da includere all'interno di pagine web esterne tramite opportuna progettazione del widget stesso e autorizzazione alla visualizzazione all'interno della pagina specifica. I widget possono essere modificati accedendo al tool Dashboard Builder che permette di editare tutte le caratteristiche del widget, di crearne di nuovi, e di autorizzare le pagine esterne che possono visualizzare il widget stesso.



<http://www.sii-mobility.org/demo/index-2.html>

12. Introduction to the “Notificator”

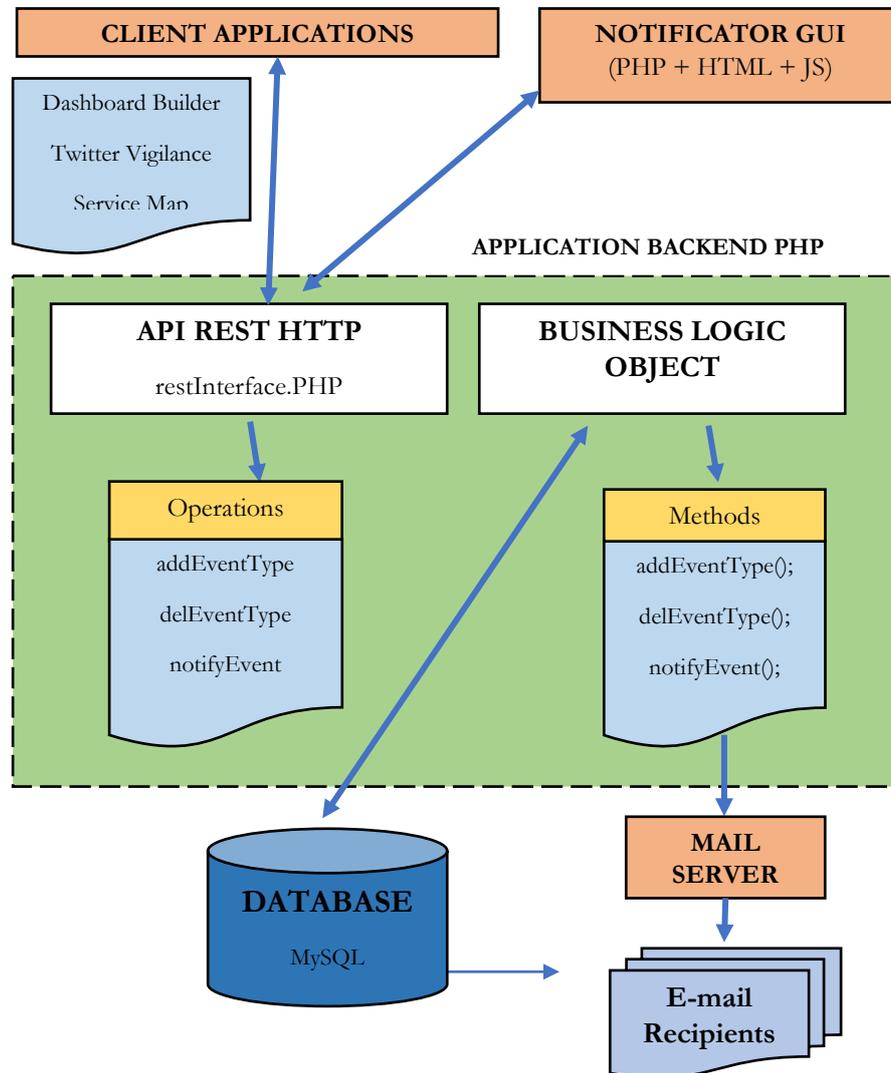
The DISIT Notification Management System, called “Notificator” in this document, is a tool designed to offer client applications the possibility to notify events of interest to users calling some REST APIs and without adding specific functionalities.

The Notificator provides some useful features:

- An *events generators management system* tool that allows to associate event generators, event types, messages, and recipients: such association is called **notification**.
When an event of a certain type occurs, an instance of each message associated with such event is sent to the recipients mentioned above.
(12.3.4 Events generators management)
- A *message book* to define and manage messages: you can add, edit or delete messages.
(12.3.5 Message book)
- An *address book* to add and manage users and their e-mail addresses: you can add, edit or delete addresses.
(12.3.6 Address book)
- Rest book (TO BE DONE).
(12.3.7 Rest book (TBD))
- An *events log* to consult past events details with several search filters.
(12.3.8 Events log)
- A GUI to add, edit or delete *client applications* to decide how to manage their notifications in the Notificator.
(12.3.3 Client applications (TBD))

12.1. Architecture of the Notificator tool

In the following figure, the architecture of the Notificator solution is presented.



At the top of the image there are the client applications and the Notificator GUI; they represent the bridge to gain access for use of the Notificator tool.

Client applications:

The client must set up own application to communicate via HTTP calls with the *Notificator Backend PHP Module* and, so, to use the Notificator tool properly.

Notificator GUI:

The graphic user interface is, obviously, accessible from the client to manage notifications.

The client applications and the Notificator graphic user interface communicate with the *Notificator Backend PHP Module*, a module composed by two PHP files:

- **API Rest HTTP** is a PHP interface (`restInterface.PHP`) which exposes different “operations” and communicates via HTTP calls.
- **Business Logic Object** contains the logic (`restController.PHP`) of the Notificator.



It implements the methods characterising this tool (e.g. the sending e-mails through the **Mail Server**).

Furthermore, it can manage the Database.

Database:

A MySQL Database to store interesting data about notifications.

12.2. System entities and their lifecycle

In this chapter, you will learn to know the system entities and their lifecycle.

These entities compose the basis of the notification and their lifecycle help you to realize how the Notificator responds to your actions.

This knowledge is very important to understand clearly how to manage notifications, so this reading is highly recommended.

12.2.1. System entities

Event

An event is, generally, an episode characterized by certain conditions.

There are several *event types* and an event is created by an *event generator*.

An event can be, for example, an out-of-range value detected.

Event Type

An event type shows the reasons of the occurrence of an event: in other words, it describes the nature of an event.

An event type can be, for example, the violation a certain threshold rule or the lack of data for a certain metric on a certain period.

Event Generator

An event generator is an application entity (e.g. a metric or a widget) that fires events of a certain type.

An event generator can be, for example, a widget in a dashboard.

12.2.2. System entities' lifecycle

Before introducing entities' lifecycle, it is necessary to explain interactions between the Notificator and the client applications.

Actions in client applications	Consequences in the <i>Notificator</i>
The user creates an event generator without threshold rules and without registering it to the Notificator. e.g. The user creates a widget in the client application Dashboard Manager without threshold rules and without registering to the Notificator.	Nothing.
The user creates an event generator with threshold rules without registering it to the Notificator.	Nothing.

<p>e.g. The user creates a widget in the client application Dashboard Manager involving the city temperature with a threshold rule $10^{\circ}\text{C} < \text{Temperature} < 30^{\circ}\text{C}$ and without registering it to the Notificator.</p>	
<p>The user creates an event generator without threshold rules, registering it to the Notificator.</p> <p>e.g. The user creates a widget in the client application Dashboard Manager without threshold rules registering it to the Notificator.</p>	<p>An event generator is created without event types associated to it.</p> <p>You can't activate notifications without event types.</p> <p>e.g. The widget is created: you can see it in the generator list events generators management window but you can't add a notification related to the widget without event types.</p>
<p>The user creates an event generator with threshold rules registering it to the Notificator.</p> <p>e.g. The user creates a widget in the client application Dashboard Manager involving the city temperature with a threshold rule $10^{\circ}\text{C} < \text{temperature} < 30^{\circ}\text{C}$ and registering it to the Notificator.</p>	<p>An event generator is created with an event type for each threshold rule.</p> <p>e.g. The widget is created: you can see it in the generator list events generators management window and associate a notification to its threshold rule.</p>
<p>The user registers an event generator to the Notificator, previously created with threshold rules without registering it.</p> <p>e.g. The user registers a widget involving the city temperature to the Notificator; it was previously created in the client application Dashboard Manager with a threshold rule $10^{\circ}\text{C} < \text{Temperature} < 30^{\circ}\text{C}$ without being registered to the Notificator.</p>	<p>An event generator is created with an event type for each threshold rule.</p> <p>Previously it wasn't visible in the generator list events generator management window.</p> <p>e.g. The widget and its threshold rule are visible after the registration so you can manage related notifications.</p>
<p>The user adds a threshold rule to an event generator.</p> <p>e.g. The user adds a threshold rule ($\text{Temperature} < 50^{\circ}\text{C}$) to a widget previously created in the client application Dashboard Manager involving the city temperature.</p>	<p>You can see the new event type: 1) in the "event type" menu in the "add notification" pop-up window; 2) in the "event type" menu in the "edit notification" pop-up window and 3) in the notification list related to that event generator and to the event type added. You can manage notifications related to it.</p> <p>e.g. The widget previously created let you manage events and notifications about its new threshold rule ($\text{Temperature} < 50^{\circ}\text{C}$).</p>

<p>The user edits a threshold rule of an event generator.</p> <p>e.g. The user edits a threshold rule from $10^{\circ}\text{C} < \text{Temperature} < 30^{\circ}\text{C}$ to $\text{Temperature} < 40^{\circ}\text{C}$ of a widget previously created in the client application Dashboard Manager involving the city temperature.</p>	<p>Each notification related to the old event type gets updated to the new event type.</p> <p>You can see the new event type instead of the old event type: 1) in the “event type” menu in the “add notification” pop window; 2) in the “event type” menu in the “edit notification” pop-up window and 3) in the notification list related to that event generator and to the event type edited.</p> <p>e.g. The widget previously created let you manage events and notifications about its new threshold rule ($\text{Temperature} < 40^{\circ}\text{C}$).</p> <p>The old threshold is not visible because it is deactivated.</p>
<p>The user edits the event generator’s title.</p> <p>e.g. The user renames a widget previously created in the client application Dashboard Manager from “Widget Temperature” to “My City Temperature”.</p>	<p>The corresponding event generator’s name is automatically updated.</p> <p>e.g. The widget is visible with its new name “My City Temperature”.</p>
<p>The user deletes an event generator.</p> <p>e.g. The user deletes a widget previously created in the client application Dashboard Manager.</p>	<p>The corresponding event generator is deactivated: the related notifications are deactivated but the related past events are kept in the events log window.</p> <p>e.g. The widget and its notifications are deactivated and not visible.</p> <p>In the events log window, the related past events are kept.</p>
<p>The user deletes a threshold rule of an event generator.</p> <p>e.g. The user deletes the threshold rule $10^{\circ}\text{C} < \text{Temperature} < 30^{\circ}\text{C}$ of a widget previously created in the client application Dashboard Manager.</p>	<p>The corresponding event type is deactivated: the related notifications are deactivated but the related past events are kept in the events log window.</p> <p>You can’t see the event type in the events generators management window.</p> <p>e.g. The threshold rule is deactivated, and you can’t see it in events generators management window.</p> <p>The related past events are kept in the events log window.</p>

12.3. Notification Management System

12.3.1. Login

The Notification Management System tool does not allow to register new users since it uses accounts created and available in the client applications.

e.g. If you are registered to the client application Dashboard Manager with:

Username: Mario.Rossi

Password: mariorossi2017

You have to use these user name and password to login in the Notificator tool.

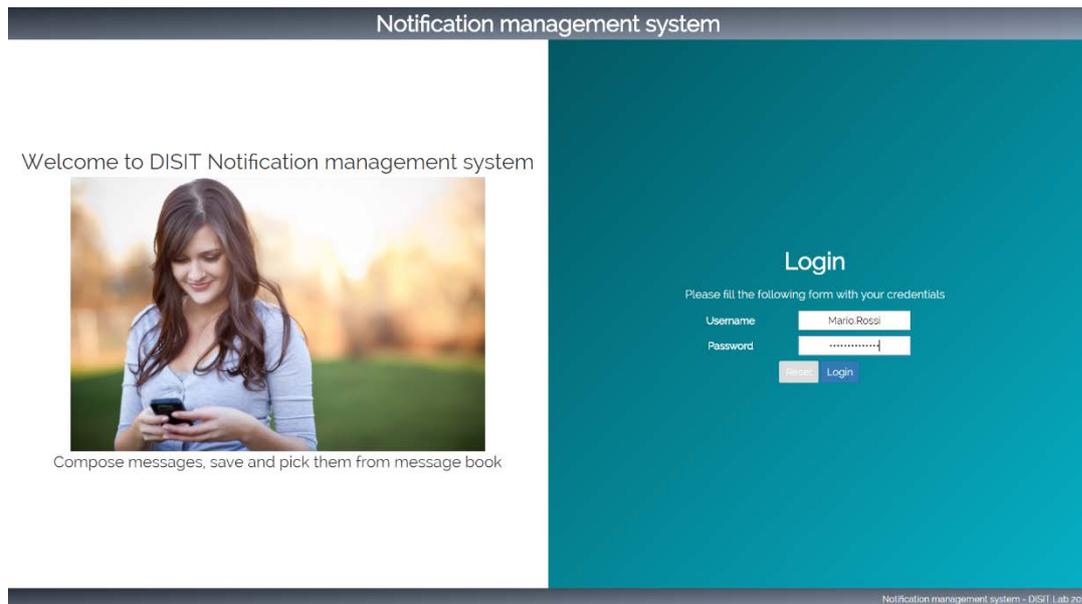


Figure 1: Login

12.3.2. Profiles and Notification pop-up

After logging in you can see the **Events generators management** section (12.3.4 Events generators management).

Please, pay attention to the footer.



Figure 2: Footer of the Events generators management section

From left to right:

- *Logout button:* You can log out in every moment you need by clicking this button.
- *Username:* The account's username logged in.
- *Application:* The client application associated to the User (e.g. Dashboard Manager in this case).
- *User type:* The user type (e.g. ToolAdmin) indicates which things you can see and do in this tool.

ToolAdmin is the only user type that is allowed to see and manage any notifications related to each event generator belonging to each application (4.4 [Events generators management](#)) and to consult all past events (4.8 [Events log](#)).

The different user types (e.g. Manager or AreaManager) can only see and manage notifications related to their event generators.

Furthermore, they can only consult past events associated to their event generators.

- *RT events*: this toggle allows to activate or deactivate the notification pop-up (by default, the value is set ON).

When one or more events occur and you are logged into the Notificator, a notification pop-up appears at the bottom left of the page for a few seconds.

The screenshot shows the 'Events log' interface. At the top, there are navigation tabs: GENERATORS, MESSAGE BOOK, ADDRESS BOOK, REST BOOK, EVENTS LOG (active), and CLIENT APPLICATIONS. Below the tabs are filters for Text filter, Start date (2017-07-31 00:00:00), End date (2017-08-11 09:22:00), and Application (All). A table lists events with columns: Generator container, Generator name, Generator type, User, Event time, Event type, Application, and Link. A notification pop-up is visible over the table, titled 'New events occurred', listing events for 'Ataf RT' and 'Temp'.

Generator container	Generator name	Generator type	User	Event time	Event type	Application	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:38:03	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:37:32	27 < Value <= 54 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:37:02	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:36:28	27 < Value <= 54 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:36:02	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Engagement created	EngagementCreated	marazzini	2017-08-09 17:35:58	Value <= 11 - Low	Dashboard Manager	Link
ToolAdmin - Public	Temp	DCTemp1	marazzini	2017-08-09 17:35:42	6 < Value <= 30 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:35:25	27 < Value <= 54 - Ok	Dashboard Manager	Link

Figure 3: Notification pop-up with RT events set to ON

In the notification pop-up, you can see one or more events with the following details:

- *Generator (Event generator)*: application entity (e.g. a metric or a widget) that fires events of a certain type.
- *Event (Event type)*: the description of the nature of an event (e.g. a threshold rule).
- *Time*: when the event occurs.

To see more event details, go to the **Events log** section (12.3.8 Events log)

12.3.3. Client applications (TBD)

THIS FUNCTIONALITY IS ACTUALLY NOT YET AVAILABLE.

It allows you to add, edit or delete client applications in Notificator.

Likely only Tool Admin users will be able to use it, managing the following fields of the database table Notificator.clientApplications:

- *Application name*;
- *Application LDAP name*: Application name, without spaces.
- *Login API URL*: API address that each client application must expose to allow the Notificator to search the user who tries to log in among those of the client application itself.
- *Container title label*: generator container name (e.g. in the Dashboard Manager System the name is "Dashboard").
- *Generator title label*: generator name (e.g. in the Dashboard Manager System the name is "Widget").
- *Generator type label*: generator type name (e.g. in the Dashboard Manager System the name is "Metric type").



- *Generator link label: generator container link name (e.g. in the Dashboard Manager System the name is "Dashboard link").*

12.3.4. Events generators management

The *events generators management* window lets you manage associations between generators, event types, messages and recipients with a user-friendly interface.

Dashboard title	Widget title	Metric type	User	Dashboard link	Add/edit/delete notifications
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini		
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini		
ToolAdmin - Public	Temp	DCTemp1	marazzini		
ToolAdmin - Public	Engagement created	EngagementCreated	marazzini		

Figure 4: Events generators management window

- *Application menu*: a drop-down menu containing the client applications manageable in the Notificator (e.g. Dashboard Manager, Twitter Vigilance, ...).

In the following lines we have decided to explain the “generic” names of the labels; for this reason, you can notice some differences with the name of the labels in Figure 4.

In fact, for each application, there are specific names related to its entities (see section 12.2 System entities and their lifecycle).

These specific names have been chosen by the user who added the client application to the Notificator (see section 12.3.3 Client applications (TBD)).

From left to right:

- *Container title*: the name of an event generator container (e.g. a dashboard title).
- *Generator name*: the event generator’s name (e.g. a widget).
- *Generator type*: the generator type’s name (e.g. a metric).
- *User*: the user to whom the generator belongs.
- *Container link*: you can go to the container page clicking the container link (e.g. the dashboard page).
- *Manage events and notifications*: this button opens a new window in which you can add, edit and delete notifications related to the chosen generator (details are explained in the next page).
- *Search filter*: when you type something into this field, the whole table is filtered, showing only rows containing the search string.

The table is paginated with ten containers (maximum) per page. If there are more than ten records you can change page with these buttons that appear under the table.

Every toggle in the table header sorts the corresponding column:



- First click to have a descending order.
- Second click to have an ascending order.
- Third click to get back to original order.

Clicking the “Edit notification” button you can manage events and notifications.
The following window will be shown:

Add/edit/delete notifications									
Dashboard title	Widget title	Metric type	User				Dashboard link		
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini						
<input type="text" value="Search"/> 									
Event type	Notification name	Active	Validity start	Validity end	Subject	Text	Edit notification	Delete notification	
Value <- 27 - Bad	Park free - Bad	Yes	2017-08-03 12:57:00	2018-08-03 12:57:00	[[Autol]]	Dear recipient, the f...			
27 < Value <- 54 - Ok	Park free - Ok	Yes	2017-08-03 14:10:00	2018-08-03 14:10:00	[[Autol]]	Dear recipient, the f...			
<input type="button" value="Close"/>									

- *Event type* (e.g. a threshold rule): the kind of event for this notification.

Figure 5: Manage events and notifications window

- *Notification name*: the name given in the notification’s creation phase.
- *Active*: a notification can be active or not.
- *Validity start / Validity end*: these dates describe the notification validity period. In other words, only events occurred in this period will be notified
- *Subject*: the subject of the message related to the notification.
- *Text*: the message body related to the notification.
- *Edit*: you can edit a notification.
- *Delete*: you can delete a notification.
- *Search filter*: when you type something into this field, the whole table is filtered, showing only row containing the search string.

12.3.4.1. Add a notification

To add a new notification, you must click the “Plus” button on the top right from the window previously described in the **Events generators management** section (12.3.4 Events generators management).

Add/edit/delete notifications									
Dashboard title	Widget title	Metric type	User				Dashboard link		
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini						
<input type="text" value="Search"/> 									
Event type	Notification name	Active	Validity start	Validity end	Subject	Text	Edit notification	Delete notification	

A new window will be shown:

Figure 6: Add e-mail notification modal window

You must fill the following fields:

- *Notification name*: a name of your choice for this notification. It will not appear in the messages sent.
- *Active*: you can decide through this toggle button to set a notification in an active status or not.
- *Validity start / Validity end*: these dates describe the notification validity period. In other words, only events occurred in this period will be notified.
By default, a notification is active from the time of the “window opening” to one year later.
- *Recipients from address book*: clicking on this field you can choose the addresses from a drop-down menu. Addresses are taken from the address book.
Clicking the X button close to the recipients’ names you can delete the recipients you do not want to insert. Holding down CTRL button you can select more than one recipients in group.

Figure 7: Recipients’ shortcuts

- *Custom recipients*: in this field, you can write different addresses from those saved in the address book. The emails written must be separated by commas, spaces or commas followed by spaces.
- *Message source*: this toggle button allows you to pick a message from the message book or to compose a new one:

- *Message picked from the message book:*
 - Search your message through the “Search” filter:
when you type something into this field, the whole table is filtered, showing only rows containing the search string.
 - Click on the radio button in the “Chosen message” column to choose a message.
 - Confirm your choice through the “Confirm” button.

Chosen message	Subject	Text
<input type="radio"/>	Active users - Low	Dear recipient, the f.
<input type="radio"/>	Park free SMN - Low	Dea recipient, the fo.
<input type="radio"/>	Ataf RT - Speedo - Ko	Dear recipient, the f.
<input type="radio"/>	Ataf RT - Bar content - Low	Dear recipient, Eeve

Figure 8: Messages' table

- *New message:*
 - Write a Message subject.
 - Write a Message body.
Confirm through the “Confirm” button.

Add e-mail notification

Notification name	Active	Validity start	Validity end
<input type="text"/>	<input checked="" type="checkbox"/> Yes	9/08/2017 15:31	9/08/2018 15:31

Recipients from address book	Custom recipients
<input type="text"/>	<input type="text"/>

Message source	Event type
<input type="button" value="New"/>	27 < Value <- 54 - Ok

Message subject

If you leave the subject blank or filled with the **[[Auto]]** placeholder, the system will generate the following automatic subject: Application name - Container title - Generator name - Generator type

Message body

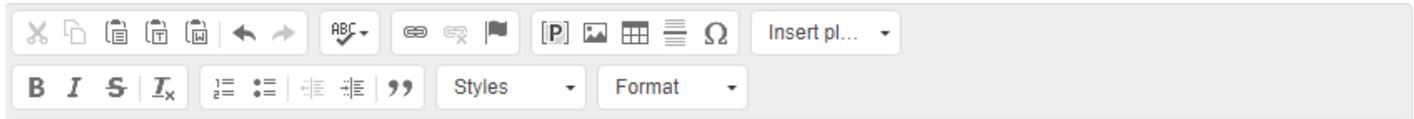
You can add event details automatically selecting the **[[EventDetails]]** placeholder from the editor menu or adding it manually to message body. If the placeholder isn't added event details will be automatically appended at the end of message body.

Dear recipient,
the following event has occurred:
[[EventDetails]]
Regards,
DISIT Notification System

Figure 9: Add e-mail notification modal window with fields for a new message

If you leave the subject blank or filled with the **[[Auto]]** placeholder, the system will generate the following automatic subject: *Application name - Container title - Generator name - Generator type*

You can add event details automatically selecting the `[[EventDetails]]` placeholder from the editor menu or adding it manually to message body. If the placeholder isn't added, event details will be automatically included at the end of the message body.



First line (from left to right):



You can cut, copy, paste, paste as plain text and paste from Word.

Furthermore, you can undo and redo.



You can use SCAYT (Spell Check As You Type) for different languages.



You can link, unlink and create anchors.

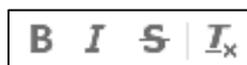


You can add placeholders, images, tables, horizontal lines and special characters.



You can insert `[[EventDetails]]` placeholder clicking this button.

Second line (from left to right):



You can write with bold, italic and strikethrough characters.

You can also remove the previously format used.



You can insert or remove numbered list, bulleted list.

You can also decrease or increase indent and block quote.

Styles ▾

Normal ▾

You can edit the block style and the paragraph format as you need.

- *Event type*: clicking on this field you can choose the event type associated with this notification from a drop-down menu.

To save your notification you must click the “Confirm” button, while to go back without saving you must click the “Abort” button.

12.3.4.2. Edit a notification

To edit a notification, you must click the “Edit” button from the window described in the *Events generators management* section (4.4 Events generators management).

A new window will be shown:

Edit e-mail notification

Notification name	Active	Validity start	Validity end
<input type="text" value="Ataf RT - Speedo - Ko"/>	<input checked="" type="checkbox"/> Yes	<input type="text" value="24/07/2017 10:55"/>	<input type="text" value="31/07/2017 10:55"/>
Recipients from address book		Custom recipients	
<input type="text" value="Mino Marazzini <mino.marazzini@gmail.com>"/>		<input type="text"/>	
Message source		Event type	
<input type="button" value="Book"/>		<input type="text" value="Value <= 50 - Ko"/>	
<input type="text" value="Search"/>			
Chosen message	Subject	Text	
<input type="radio"/>	Active users - Low	Dear recipient, the f..	
<input type="radio"/>	Park free SMN - Low	Dea recipient, the fo..	
<input checked="" type="radio"/>	Ataf RT - Speedo - Ko	Dear recipient, the f..	
<input type="radio"/>	Ataf RT - Bar content - Low	Dear recipient, [[Eve..	

Figure 10: Edit e-mail notification window

You can change the following fields to manage an event in a different way:

- *Notification name*: a name of your choice for this notification. It will not appear in the messages sent.
- *Active*: you can decide through this toggle button to set a notification in an active status or not.
- *Validity start / Validity end*: these dates describe the notification validity period. In other words, only events occurred in this period will be notified.
By default, a notification is active from the time of the “window opening” to one year later.
- *Recipients from address book*: clicking on this field you can choose the addresses from a drop-down menu. Addresses are taken from the address book. Clicking the X button close to the recipients’ names you can delete the recipients you do not want to insert. Holding down CTRL button you can select more than one recipients in group.

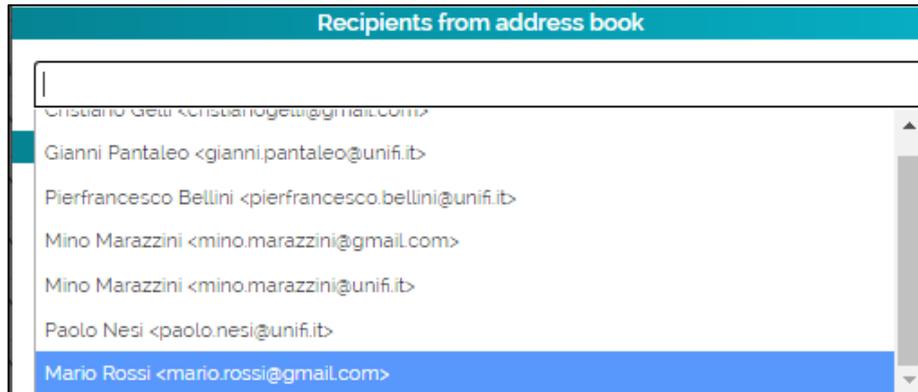


Figure 11: Recipients' shortcut

- *Custom recipients*: in this field, you can write different addresses from those saved in the address book. The addresses written must be separated by commas, spaces or commas followed by spaces.
- *Message source*: this toggle button allows you to pick a message from the message book or to compose a new one:
- *Message picked from the message book*:
 - Search your message through the “Search” filter: when you type something into this field, the whole table is filtered, showing only rows containing the search string.
 - Click on the radio button in the “Chosen message” column to choose a message.
 - Confirm your choice through the “Confirm” button.

Chosen message	Subject	Text
<input type="radio"/>	Active users - Low	Dear recipient, the f
<input type="radio"/>	Park free SMN - Low	Dea recipient, the fo
<input type="radio"/>	Ataf RT - Speedo - Ko	Dear recipient, the f
<input type="radio"/>	Ataf RT - Bar content - Low	Dear recipient, SEve

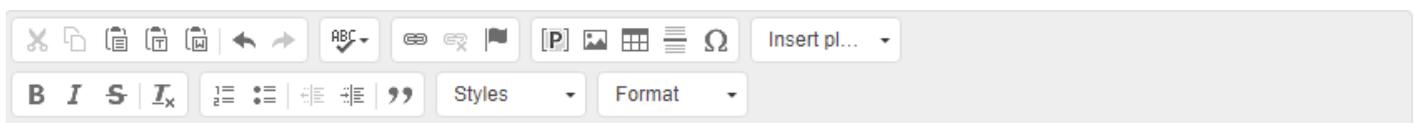
Figure 12: Messages' table

- *New message*:
 - Write a Message subject.
 - Write a Message body.
 - Confirm through the “Confirm” button.

Figure 13: Edit e-mail notification modal window with fields for a new message

If you leave the subject blank or filled with the `[[Auto]]` placeholder, the system will generate the following automatic subject: *Application name - Container title - Generator name - Generator type*

You can add event details automatically selecting the `[[EventDetails]]` placeholder from the editor menu or adding it manually to message body. If the placeholder isn't added event details will be automatically appended at the end of message body.



First line (from left to right):



You can cut, copy, paste, paste as plain text and paste from Word.

Furthermore, you can undo and redo.



You can use SCAYT (Spell Check As You Type) for different languages.



You can link, unlink and create anchors.

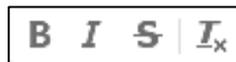


You can add placeholders, images, tables, horizontal lines and special characters.



You can insert `[[EventDetails]]` placeholder clicking this button.

Second line (from left to right):



You can write with bold, italic and strikethrough characters.

You can also remove the previously format used.



You can insert or remove numbered list, bulleted list.

You can also decrease or increase indent and block quote.



You can edit the block style and the paragraph format as you need.

- *Event type*: clicking on this field you can choose the event type associated with this notification from a drop-down menu.

To save your notification you must click the “Confirm” button while to go back you must click the “Back” button.

12.3.4.3. Delete a notification

To delete a notification, you must click the “Delete” button from the window described in the **Events generators management** section (12.3.4 Events generators management).

Then you must click the “Confirm” button in the pop-up shown.

To go back you can click the “Abort” button.

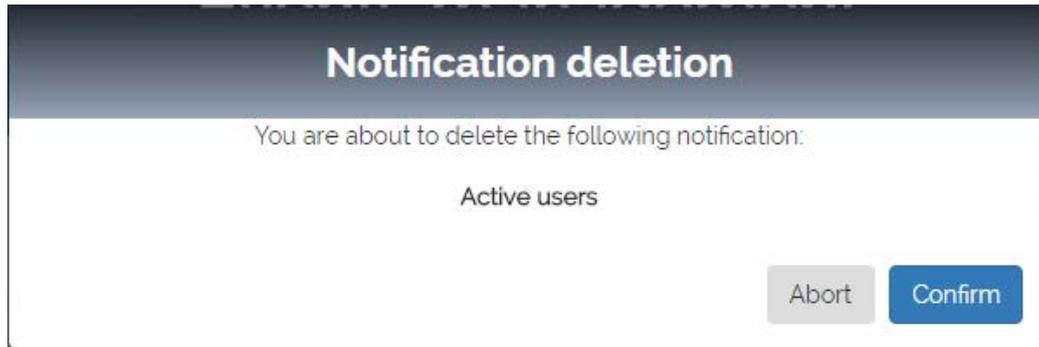


Figure 14: Notification deletion modal window

12.3.4.4. Set the validity period

During the notification creation, it is possible to set the notification validity period by specifying the start-end dates. To set the validity period of the notification, you can write manually the time or follow these steps (the same procedure is valid for the *Validity start date* and for the *Validity end date* fields):

- Click in the field.
- Choose the year:
 - Click the couple *Month Year* (e.g. July 2017).

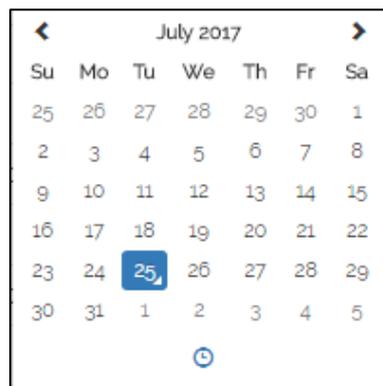


Figure a: Calendar for time filter

- Click the *Year* or slide years clicking the buttons at the sides of the *Year* (e.g.2017).

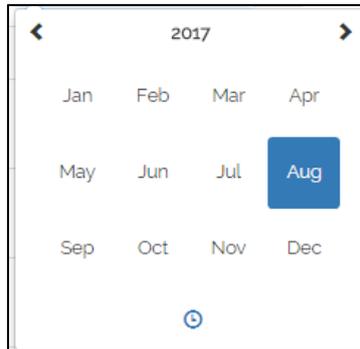


Figure b

- Choose the *Year* in the window or slide the decades clicking the buttons at the sides of the *Decade* (e.g. 2015-2026).

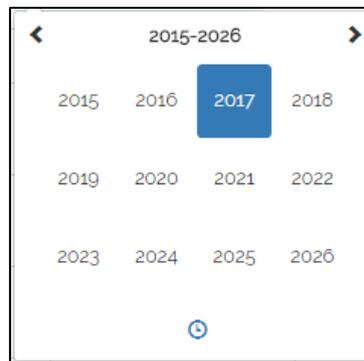


Figure c

- Choose the month:
 - Click the couple *Month Year* (e.g. July 2017).



Figure d

- Choose the *Month* (e.g. August)

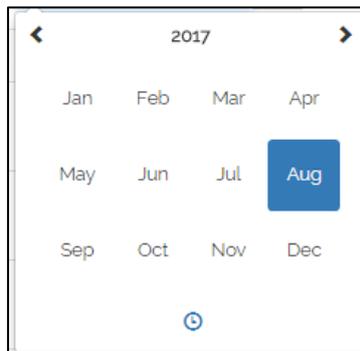


Figure e

- Choose the day:
 - Choose the *Day* (e.g. 25).

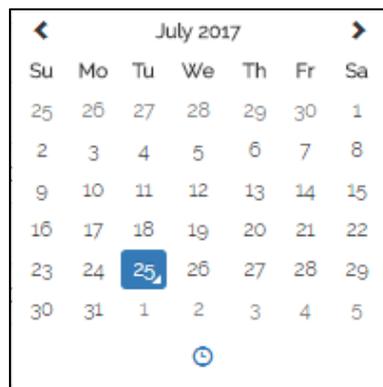


Figure f

- Choose the time clicking the little clock (Fig.g) and choose the time clicking the buttons above and below the “hour”, the “minute” and the “second” number (Fig.h).



Figure g

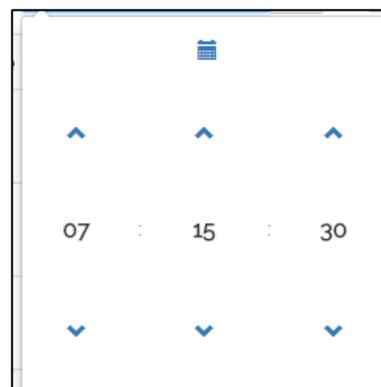
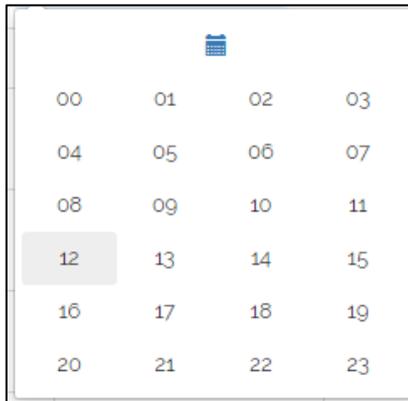
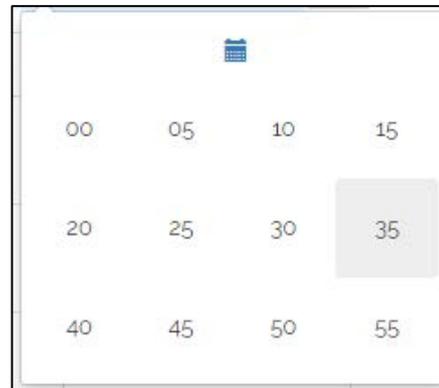


Figure h

You can also choose the time clicking the “hour” number (e.g.7 in Fig.h) obtaining the window in Fig.i or clicking the “minute” (e.g.15) or the “second” number (e.g. 30) obtaining the window in Fig.l.



00	01	02	03
04	05	06	07
08	09	10	11
12	13	14	15
16	17	18	19
20	21	22	23

Figure i

00	05	10	15
20	25	30	35
40	45	50	55

Figure l

12.3.5. Message book

The message book contains messages to be sent when an event occurs.

You can choose them in the **Events generators management** section (12.3.4 Events generators management) by clicking the corresponding menu item.



Figure 15: Message book window

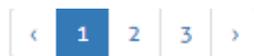


A new window will be shown with a table listing the available messages and specifying the following columns:

- *Search filter*: when you type something into this field, the whole table is filtered, showing only rows containing the search string.
- *Subject*: the subject message.
- *Text*: the text message.
- *Edit*: you can edit the message.
- *Delete*: you can delete the message from the message book.

The table is paginated with ten messages (maximum) per page.

If there are more than ten records you can change page with these buttons that appear under the table.



The toggle in every column sorts the relative column:

- First click to have a descending order.
- Second click to have an ascending order.
- Third click to get back to original order.

12.3.5.1. Create a new message

To create a new message, you must click the “Plus” button available on the top right of the *Message Book* window.

The following window will be shown:

Add e-mail

Message subject

If you leave the subject blank or filled with the **[[Auto]]** placeholder, the system will generate the following automatic subject:
Application name - Container title - Generator name - Generator type

[[Auto]]

Message body

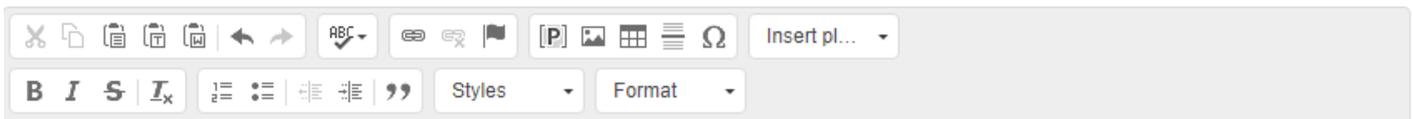
You can add event details automatically selecting the **[[EventDetails]]** placeholder from the editor menu or adding it manually to message body. If the placeholder isn't added event details will be automatically appended at the end of message body.

Dear recipient,
the following event has occurred:
[[EventDetails]]
Regards.
DISIT Notification System

Cancel Confirm

Figure 16: Add e-mail window

- *Message subject*: write a subject for your message.
If you leave the subject blank or filled with the **[[Auto]]** placeholder, the system will generate the following automatic subject: *Application name - Container title - Generator name - Generator type*
- *Message body*: write a text for your message.



First line (from left to right):



You can cut, copy, paste, paste as plain text and paste from Word.

Furthermore, you can undo and redo.



You can use SCAYT (Spell Check As You Type) for different languages.



You can link, unlink and create anchors.

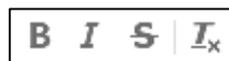


You can add placeholders, images, tables, horizontal lines and special characters.



You can insert `[[EventDetails]]` placeholder clicking this button.

Second line (from left to right):



You can write with bold, italic and strikethrough characters.

You can also remove the previously format used.



You can insert or remove numbered list, bulleted list.

You can also decrease or increase indent and block quote.



You can edit the block style and the paragraph format as you need.

You can add event details automatically selecting the `[[EventDetails]]` placeholder from the editor menu or adding it manually to message body. If the placeholder isn't added, event details will be automatically included at the end of message body.

By including the `[[EventDetails]]` placeholder in the message body, the following event details will be shown:

Field name	Description	Possible value
Event time	When the event occurs	2017-07-17 17:12:06
Application	Application related to the event	Dashboard Manager
Container title	The name of an event generator container	Km4City (link)
Generator name	The event generator's name	Engagement sent
Generator type	The generator type's name	Km4City_Query_Day
Event type	Description of the nature of the event	Value < 4500.0 more than 1 time(s)
Registered value	Registered value by one or more sensors	739
User	The user who created the event generator	Softec

To save your message, click the “Confirm” button; while to go back without saving, you must click the “Cancel” button.

12.3.5.2. Edit a message

To edit an existing message in the *Message Book* window, you must click the corresponding “Edit” button.

The same window of the previous paragraph (in edit mode) will be shown and you will be able to:

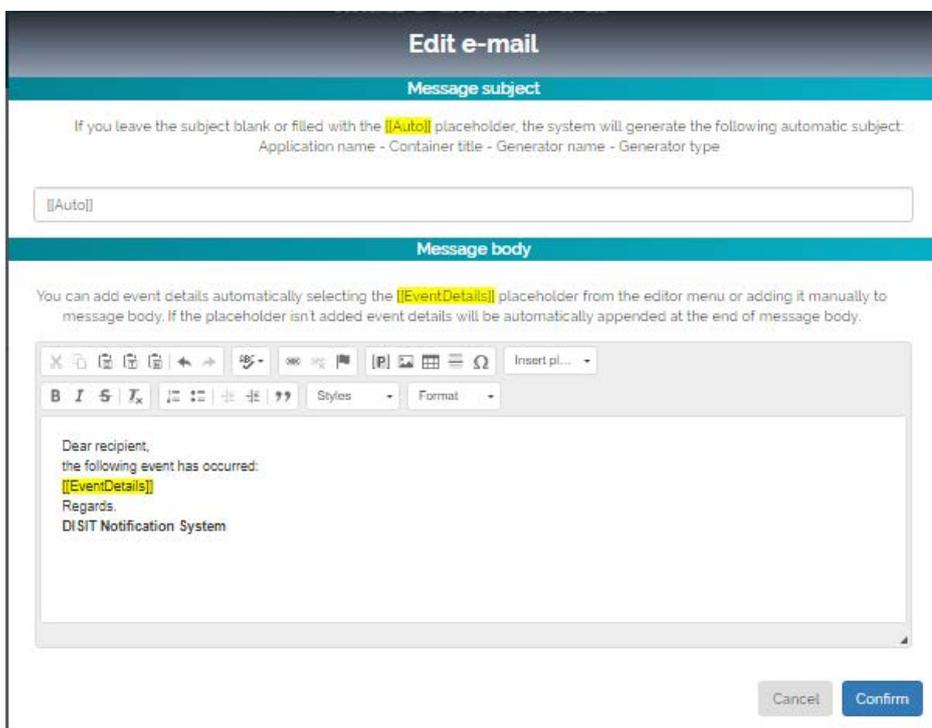
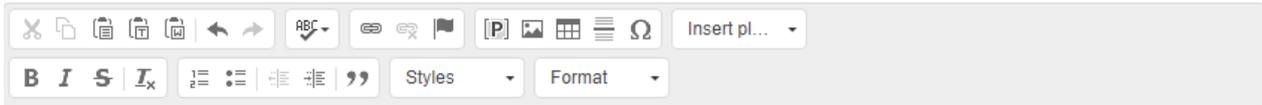


Figure 17: Edit e-mail modal window

- Edit the subject.
If you leave the subject blank or filled with the **[[Auto]]** placeholder, the system will generate the following automatic subject: *Application name - Container title - Generator name - Generator type*
- Edit the text.



First line (from left to right):



You can cut, copy, paste, paste as plain text and paste from Word.

Furthermore, you can undo and redo.



You can use SCAYT (Spell Check As You Type) for different languages.



You can link, unlink and create anchors.

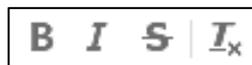


You can add placeholders, images, tables, horizontal lines and special characters.



You can insert **[[EventDetails]]** placeholder clicking this button.

Second line (from left to right):



You can write with bold, italic and strikethrough characters.

You can also remove the previously format used.



You can insert or remove numbered list, bulleted list.

You can also decrease or increase indent and block quote.



You can edit the block style and the paragraph format as you need.

You can add event details automatically selecting the `[[EventDetails]]` placeholder from the editor menu or adding it manually to message body. If the placeholder isn't added event, details will be automatically added at the end of the message body.

To save your message, click the “Confirm” button; while to go back without saving, you must click the “Cancel” button.

12.3.5.3. Delete a message

To delete an existing message in the *Message Book* window, you must click the corresponding “Delete” button.

Then you must click the “Confirm” button in the new window.

If the “Delete” button in the *Message Book* list is grey, it means that the message is used in one or more notifications and you can't delete it.

To go back you must click the “Cancel” button.

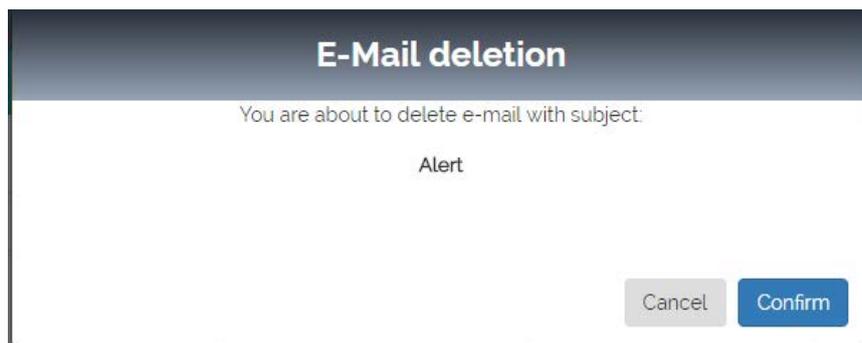


Figure 18: E-mail deletion window

12.3.6. Address book

The address book contains the recipients information needed to notify an event when it occurs.

To see the list of saved recipients, click the *Address Book* menu item in the **Events generators management** section (12.3.4 Events generators management).

Address	First name	Last name	Organization	Edit	Delete
cristianogelli@gmail.com	Cristiano	Gelli	Student		
gianni.pantaleo@unifi.it	Gianni	Pantaleo	DISIT		
mario.rossi@gmail.com	Mario	Rossi	DISIT		
mino.marazzini@gmail.com	Mino	Marazzini	Personal		
mino.marazzini@unifi.it	Mino	Marazzini	DISIT		
paolo.nesi@unifi.it	Paolo	Nesi	DISIT		
pierfrancesco.bellini@unifi.it	Pierfrancesco	Bellini	DISIT		

Figure 19: Address book window

- *Search filter*: when you type something into this field, the whole table is filtered, showing only rows containing the search string.
- *Address*: recipient's e-mail address.
- *First name*.
- *Last name*.
- *Organization*: organization to whom the recipient belongs.
- *Edit*: you can edit some recipient's informations.
- *Delete*: you can delete a recipient from the address book.

The table is paginated with ten users (maximum) per page.

If there are more than ten records you can change page with these buttons that appear under the table.



The toggle in every column sorts the relative column:

- First click to have a descending order.
- Second click to have an ascending order.
- Third click to get back to original order.

12.3.6.1. Add a new recipient

To add a new recipient in the *Address Book* list, you must click the “Plus” button available on the top right of the *Address Book* window.

The following new window will be shown:

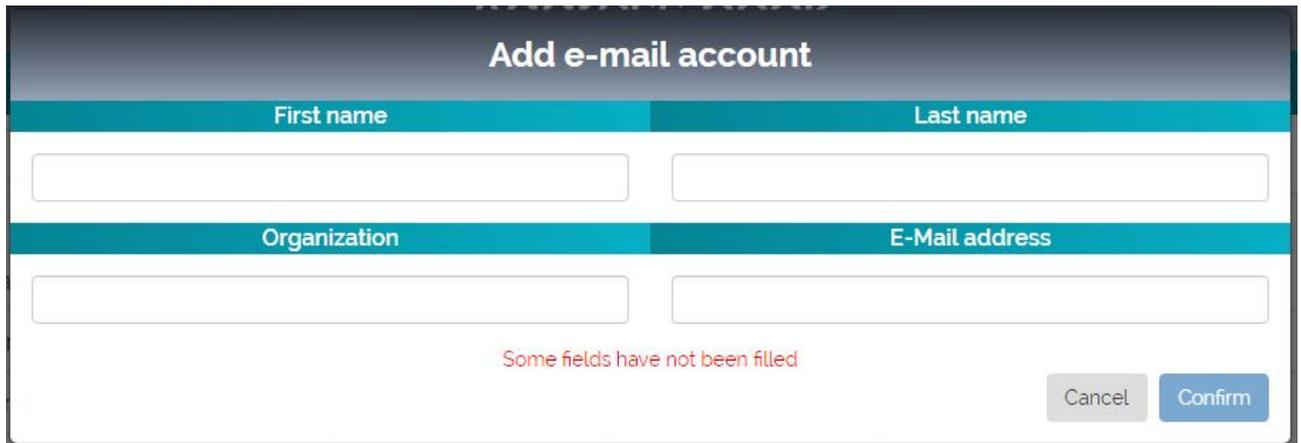


Figure 20: Add e-mail account window

- *First name and Last name*: write the recipient’s first and last name you want to add.
- *Organization*: write the recipient’s organization name.
- *E-mail address*: write the recipient’s e-mail address.

To save your recipient you must click the “Confirm” button, while to go back without saving, you must click the “Cancel” button.

12.3.6.2. Edit a recipient

To edit an existing recipient in the *Address Book* list, you must click the “Edit” button in the *Address Book* window.

The same window shown in the previous paragraph (in edit mode) will be opened and you will be able to:

- Edit the recipient’s name.
- Edit the recipient’s organization.
- Edit the recipient’s e-mail address.

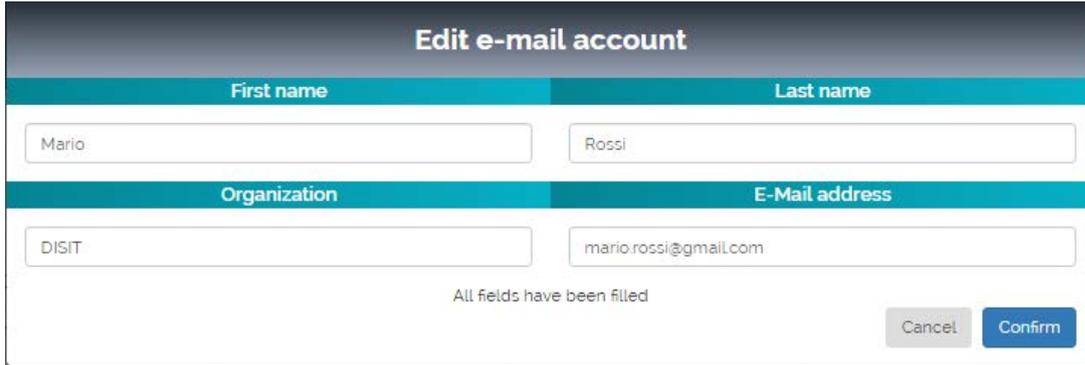


Figure 21: Edit e-mail account window

To save your recipient you must click the “Confirm” button, while to go back without saving, you must click the “Cancel” button.

12.3.6.3. Delete a recipient

To delete an existing recipient, you must click the “Delete” button in the *Address Book* window.

Then you must click the “Confirm” button in the new window shown.

If the “Delete” button is grey means that the recipient is involved in a notification and you can’t delete it.

To go back you must click the “Cancel” button.



Figure 22: E-mail deletion window

12.3.7. Rest book (TBD)

NOT YET AVAILABLE

12.3.8. Events log

In the events log you can see past events and their details through a time filter.

Generator container	Generator name	Generator type	User	Event time	Event type	Application	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:38:03	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:37:32	27 < Value <= 54 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:37:02	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:36:28	27 < Value <= 54 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:36:02	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Engagement created	EngagementCreated	marazzini	2017-08-09 17:35:58	Value <= 11 - Low	Dashboard Manager	Link
ToolAdmin - Public	Temp	DCTemp1	marazzini	2017-08-09 17:35:42	6 < Value <= 30 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:35:25	27 < Value <= 54 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:35:02	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:34:21	27 < Value <= 54 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:34:01	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:33:18	27 < Value <= 54 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:33:01	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:32:15	27 < Value <= 54 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:32:01	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-09 17:31:12	27 < Value <= 54 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-09 17:31:00	Value <= 50 - Bad	Dashboard Manager	Link

Figure 23: Events log window

- *Text filter*: the whole table is filtered, showing only rows containing the search string.
- *Generator container*: the name of an event generator container (e.g. a dashboard).
- *Generator name*: the event generator's (e.g. a widget) name.
- *Generator type*: the generator type's (e.g. a metric) name.
- *User*: the user to whom the container belongs.
- *Event time*: the moment when an event occurs;
- *Event type*: the description of the nature of an event (e.g. a threshold rule).
- *Application*: the application where the generator container is created (e.g. Dashboard Manager).
- *Time filter*: here you can specify the start and end dates. These dates describe a time filter to show past events' details.
By default, the table shows the events of the current day.
- *Application*: an application drop-down menu to filter events associated to the application chosen.

The table is paginated with seventeen events (maximum) per page.

If there are more than ten records you can change page with these buttons that appear under the table.



The toggle in every column sorts the relative column:

- First click to have a descending order.
- Second click to have an ascending order.
- Third click to get back to original order.

When one or more events occur, the table is updated.

You can see new rows in the table related to the events occurred only if the table is not filtered. As shown in the following image, new events are highlighted in light blue color.

Generator container	Generator name	Generator type	User	Event time	Event type	Application	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-11 09:24:58	27 < Value <= 54 - Ok	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-11 09:24:41	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Ataf RT	Ataf_Rt	marazzini	2017-08-11 09:24:41	Value <= 50 - Bad	Dashboard Manager	Link
ToolAdmin - Public	Engagement created	EngagementCreated	marazzini	2017-08-11 09:24:31	Value <= 11 - Low	Dashboard Manager	Link
ToolAdmin - Public	Park Free - Speedo	Park_Free	marazzini	2017-08-11 09:23:54	27 < Value <= 54 - Ok	Dashboard Manager	Link
Temp	DCTemp1	marazzini	2017-08-11 09:23:46	6 < Value <= 30 - Ok	Dashboard Manager	Link	
Park Free - Speedo	Ataf RT	Ataf_Rt	marazzini	2017-08-11 09:23:41	Value <= 50 - Bad	Dashboard Manager	Link
ee - Speedo	Park_Free	marazzini	2017-08-11 09:22:51	27 < Value <= 54 - Ok	Dashboard Manager	Link	

Figure 24: Events log with new events

12.3.8.1. Filter events

1. To filter the events list, type something in the *Text filter* field.
2. To edit the time filter, you can write manually the time or follow these steps:
 - Click in the validity field.
 - Choose the year:
 - Click the couple *Month Year* (e.g. July 2017).

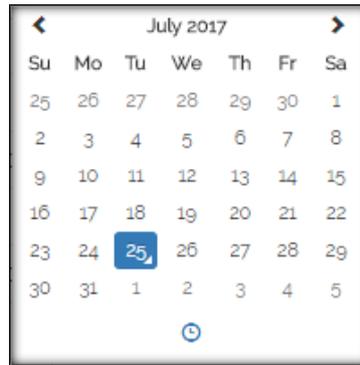


Figure a

- Click the *Year* or slide years clicking the buttons at the sides of the *Year* (e.g.2017).

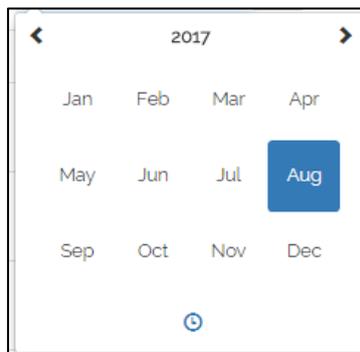


Figure b

- Choose the *Year* in the window or slide the decades clicking the buttons at the sides of the *Decade* (e.g. 2015-2026).



Figure c

- Choose the month:
 - Click the couple *Month Year* (e.g. July 2017).



Figure d

Choose the *Month* (e.g. August)

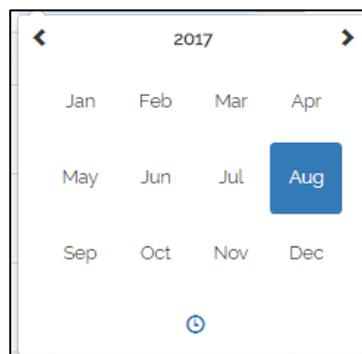


Figure e

- Choose the day:
 - Choose the *Day* (e.g. 25).



Figure f

- Choose the time clicking the little clock (Fig. g) and choose the time clicking the buttons above and below the “hour”, the “minute” and the “second” number (Fig. h).



Figure g

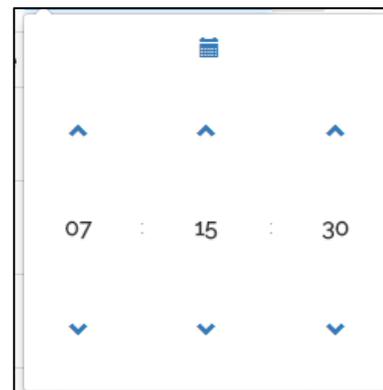


Figure h

You can also choose the time clicking the “hour” number (e.g.7 in Fig. h) obtaining the window in Fig. i or clicking the “minute” (e.g. 15) or the “second” number (e.g. 30) obtaining the window in Fig. l.

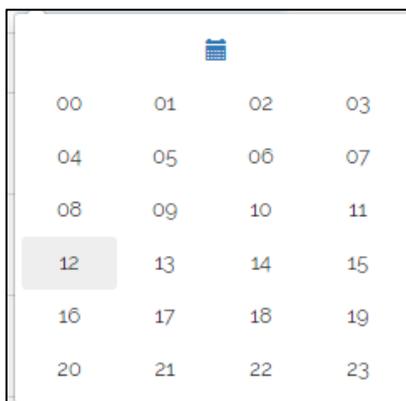


Figure i

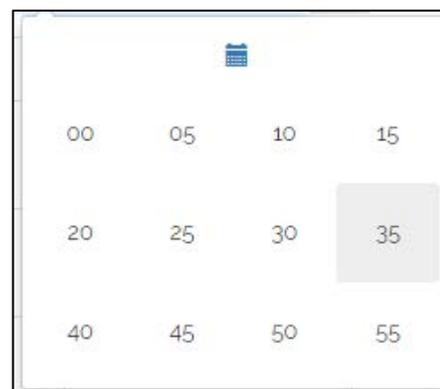


Figure l

3. Select the application in the drop-down menu to filter events related to the application of interest.
4. Click the “Filter” button to save the new filter settings (Search, time and application filters).

Annex I – Available metrics

In the following table, metrics available in the system managed by the DISIT Lab (<https://dashboard.km4city.org/management/index.php>) are listed and described.

Name	Description	Area	Source	Widget type
Ataf_Rt	Percentage of real time signals provided by ATAF	Mobility	Osservatorio Regionale per la Mobilità ed i Trasporti	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Pie chart • Single content • Speedometer • Time trend
BikeSharing	Fake metric simulating bike sharing data	Fake area	Disit	<ul style="list-style-type: none"> • Radar series • Pie chart • Table • Line series • Curved line series • Bar series
Bus_Active_Rt	Number of active busses tracked in real time	Mobility	Osservatorio Regionale per la Mobilità ed i Trasporti	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
Bus_Active_TT	Active busses on the basis of the ATAF timetable	Mobility	Osservatorio Regionale per la Mobilità ed i Trasporti	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
Bus_Position_Map	Real time position of ATAF busses	Mobility	Osservatorio Regionale per la Mobilità ed i Trasporti	<ul style="list-style-type: none"> • Service Map
Bus_State_Lines	Percentage of delays/advance on the available bus lines	Mobility	Osservatorio Regionale per la Mobilità ed i Trasporti	<ul style="list-style-type: none"> • Pie chart • State ride ATAF
Button		null	null	<ul style="list-style-type: none"> • Button
CarSharing	Fake metric simulating car sharing data	Fake area	Disit	<ul style="list-style-type: none"> • Radar series • Pie chart • Table • Line series



Name	Description	Area	Source	Widget type
				<ul style="list-style-type: none"> • Curved line series • Bar series
Car_Flow	Number of inbound/outbound cars per day (interpolation and sum based on sensors)	Mobility	Osservatorio Regionale per la Mobilità ed i Trasporti	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
CitizensSatisfacion	Fake metric simulating citizens satisfaction index	Fake area	Disit	<ul style="list-style-type: none"> • Radar series • Pie chart • Table • Line series • Curved line series • Bar series
CreatedEngagement	Km4City Created Engagements	Statistics	Disit	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
DCHum1	DataCenter Humidity 1	Datacenter	Disit	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
DCHum2	DataCenter Humidity 2	Datacenter	Disit	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
DCTemp1	DataCenter Temperature 1	Datacenter	Disit	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer



Name	Description	Area	Source	Widget type
				<ul style="list-style-type: none"> Time trend
DCTemp1float	DataCenter Temperature 1	Datacenter	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
DCTemp2	DataCenter Temperature 2	Datacenter	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
Delay_IngestionAvm	AVM ingestion delay	Mobility	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
Delay_IngestionAvm_Line17	Ingestion delay line 17	Mobility	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
Delay_IngestionAvm_Line31	Ingestion delay line 31	Mobility	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
Delay_IngestionAvm_Line6	Ingestion delay line	Mobility	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer

Name	Description	Area	Source	Widget type
				<ul style="list-style-type: none"> Time trend
EngagementCreated	Km4City Engagements Created	Statistics	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
EngagementSent	Km4City Engagements Sent	Statistics	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
EngagementUserEval	Km4City Engagements User Eval Events	Statistics	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
EngagementViewed	Km4City Engagements Viewed	Statistics	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
EventsOverview	Fake metric simulating an overview over city events	Fake area	Disit	<ul style="list-style-type: none"> Radar series Pie chart Table Line series Curved line series Bar series
Events_FI_Day	Number of events in Florence per day	Entertainment	Comune di Firenze	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend

Name	Description	Area	Source	Widget type
ExternalContent	It allows to visualize content coming from external websites and webtools	External content	Disit	<ul style="list-style-type: none"> External content
FirstAid	Florence main first aids status	Statistics	Disit	<ul style="list-style-type: none"> WidgetFirstAid with 3 different visualization mode: single hospital totals mode; single hospital details mode; multiple hospitals overview mode.
FoodPreferences	Fake metric simulating food preferences	Fake area	Disit	<ul style="list-style-type: none"> Radar series Pie chart Table Line series Curved line series Bar series
Healthcare	Fake metric simulating healthcare data	Fake area	Disit	<ul style="list-style-type: none"> Radar series Pie chart Table Line series Curved line series Bar series
InternetConnections	Fake metric simulating internet connections data	Fake area	Disit	<ul style="list-style-type: none"> Radar series Pie chart Table Line series Curved line series Bar series
Km4City_Query_Day	Number of queries per day on Km4city API (mobile or web)	Statistics	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
List_Events_FI_Day	List of the events of the day in florence with various information	Entertainment	Comune di Firenze	<ul style="list-style-type: none"> Events
MentionsTrends_FI_Day	Main trends/citations on the Twitter Vigilance Florence channel per day	Social network	Disit - Twitter Vigilance	<ul style="list-style-type: none"> Trend mentions
Meteo_Rt	Percentage of working weather measures	Meteo	Consorzio LaMMA	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Pie chart Single content

Name	Description	Area	Source	Widget type
				<ul style="list-style-type: none"> • SmartDS • Speedometer • Time trend • State ride ATAF
N_Active_Operators	Number of active operators	Statistics	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
N_Active_Tourists	Number of active tourists	Statistics	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
N_Active_Users	Number of active users	Statistics	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
N_triples	Number of M triple in the RDF store	Statistics	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
OD_Available	Number of available Open Data	Statistics	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
Park_Free	Percentage of free parking areas	Mobility	Osservatorio Regionale per la Mobilità ed i Trasporti	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content

Name	Description	Area	Source	Widget type
				<ul style="list-style-type: none"> • Time trend compare • Pie chart • Single content • Speedometer • Time trend
Park_Free_FortezzaFiera	Percentage of free areas in the Fortezza Fiera Parking	Parcheggi	Disit	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Pie chart • Single content • Speedometer • Time trend
Park_Free_MercatoCentrale	Percentage of free areas in the Mercato Centrale Parking	Parcheggi	Disit	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Pie chart • Single content • Speedometer • Time trend
Park_Free_Parterre	Percentage of free areas in the Parterre Parking	Parcheggi	Disit	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Pie chart • Single content • Speedometer • Time trend
Park_Free_StazioneSMN	Percentage of free areas in the Santa Maria Novella Parking	Parcheggi	Disit	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Pie chart • Single content • Speedometer • Time trend
Park_Rt	Percentage of working parking	Mobility	Osservatorio Regionale per la Mobilità ed i Trasporti	<ul style="list-style-type: none"> • Bar Content • Column content • Gauge chart • Generic content • Time trend compare • Pie chart • Single content • SmartDS

Name	Description	Area	Source	Widget type
				<ul style="list-style-type: none"> Speedometer Time trend State ride ATAF
Pollution	Fake metric simulating air pollution	Fake area	Disit	<ul style="list-style-type: none"> Radar series Pie chart Table Line series Curved line series Bar series
Pollution2	Pollution2	Fake area	Disit	<ul style="list-style-type: none"> Radar series Pie chart Table Line series Curved line series Bar series
Previ_Meteo	Meteo measures in a municipality	Meteo	Consorzio LaMMA	<ul style="list-style-type: none"> Prev meteo
Process	Process status	Process status	Disit	<ul style="list-style-type: none"> Process
ProtezioneCivile	Data provided by the Civil Protection	API	Disit	<ul style="list-style-type: none"> Protezione civile
Q1	Energy consumption in the district Q1	Energia	Comune di Firenze	<ul style="list-style-type: none"> Bar content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
QAria_FI_GRAMSCI_NO2	Concentration of NO2 @ FI-GRAMSCI	Air Quality	ARPAT	<ul style="list-style-type: none"> Bar content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
Recharge_Columns_Position_Map	Position of the recharge stations for electric vehicles	Mobility	Osservatorio Regionale per la Mobilità ed i Trasporti	<ul style="list-style-type: none"> Service Map
Rivers	Fake metric simulating rivers data	Fake area	Disit	<ul style="list-style-type: none"> Radar series Pie chart Table Line series Curved line series

Name	Description	Area	Source	Widget type
				<ul style="list-style-type: none"> Bar series
Scce_CPU	CPU percentage of use on the SCE in absolute (data from SCE)	Statistics	Disit	<ul style="list-style-type: none"> Scce
Scce_Job_Day	Number of processes launched on the SCE per day	Statistics	Disit	<ul style="list-style-type: none"> Scce
Scce_Mem	Memory used by the SCE in absolute (data from SCE)	Statistics	Disit	<ul style="list-style-type: none"> Scce
Sensors_Rt	Percentage of working sensor flow	Mobility	Osservatorio Regionale per la Mobilità ed i Trasporti	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Pie chart Single content SmartDS Speedometer Time trend State ride ATAF
ServiceMap_Query_Day	Number of queries to the ServiceMap tool per day	Statistics	Disit	<ul style="list-style-type: none"> Bar Content Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
Services_Access	Number of access to services per day (web e mobile)	Statistics	Disit	<ul style="list-style-type: none"> Column content Gauge chart Generic content Time trend compare Single content Speedometer Time trend
Services_Duplicate	Percentage of service duplicated (name, address, civic number)	Statistics	Disit	<ul style="list-style-type: none"> Bar content Column content Gauge chart Generic content Time trend compare Pie chart Single content Speedometer Time trend
Service_Active_FI	Number of active services in Florence	Statistics	Disit	<ul style="list-style-type: none"> Bar content Column content Gauge chart



Name	Description	Area	Source	Widget type
				<ul style="list-style-type: none"> • Generic content • Time trend compare • Single content • Speedometer • Time trend
SmartDS_Process	Result of a project on the martDS tool	Statistics	Disit- SmartDS	<ul style="list-style-type: none"> • Pie chart • SmartDS
Temperatures	Fake metric simulating temperature statistics	Fake area	Disit	<ul style="list-style-type: none"> • Radar series • Pie chart • Table • Line series • Curved line series • Bar series
Tourists	Fake metric simulating tourists flow	Fake area	Disit	<ul style="list-style-type: none"> • Radar series • Pie chart • Table • Line series • Curved line series • Bar series
TweetsRet_Florence_Day	Number of tweet/retweets on the Florence channel per day	Social network	Disit- Twitter Vigilance	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
UrbanFaults	Fake metric simulating urban faults data	Fake area	Disit	<ul style="list-style-type: none"> • Radar series • Pie chart • Table • Line series • Curved line series • Bar series
WeatherStations	Fake metric simulating weather stations data	Fake area	Disit	<ul style="list-style-type: none"> • Radar series • Pie chart • Table • Line series • Curved line series • Bar series
WifiStream_Aps	Number of active Access Wi-Fi Points in Florence active in the last 5 mins.	Network	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend

Name	Description	Area	Source	Widget type
WifiStream_Events	Number of events in the Firenze Wi-Fi in the last 5 mins.	Network	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
WifiStream_Users	Number of Firenze Wi.-Fi users	Network	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
Wifi_Conn_Day	* Number of Wi-Fi connections per day	Network	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer • Time trend
Wifi_Operative	Percentage of working Wi-Fi access points with respect to those that should working in Florence.	Network	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Pie chart • Single content • Speedometer • Time trend
Wifi_Power_5Ril	Percentage of Firenze Wi-F that provide a value of the power for at least 5 measures.	Network	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Pie chart • Single content • Speedometer • Time trend
Wifi_Remark_Day	Number of Firenze Wi.-Fi events per day	Network	Disit	<ul style="list-style-type: none"> • Bar content • Column content • Gauge chart • Generic content • Time trend compare • Single content • Speedometer



Name	Description	Area	Source	Widget type
				<ul style="list-style-type: none">Time trend

Annex II – Examples of Dashboards

Event oriented: a dashboard for controlling the status of city with respect to a large event (such as the visit of Pope, or US President). In that case, the dashboard would be dedicated to monitor the paths that would be probably used, the status of traffic in those area, the number of police and security resources in those area, the TV cameras, the hospitals, the other events and micro-events (accidents, crashes, fights), etc.

Tourism oriented: a dashboard reporting the major events in the city, the number of arrivals in the city, the number of people in the major points of the city, the number of accesses to the museums, etc.

As a conclusion, after the production of a set of Dashboards some of them were selected for trial and are reported in the web site for your public access from [Http://www.km4city.org](http://www.km4city.org) where most of them are presenting data that can be rendered on screen to public. This does not mean that the published data are open and that can be downloaded to be reused and published/exploited for other purpose. Moreover, the Dashboard can also contain data that cannot be visualized by public, for safety reasons, and thus are protected by some conditional access solution. For instance, since they are sensitive data describing the city status in real time or by predictions.

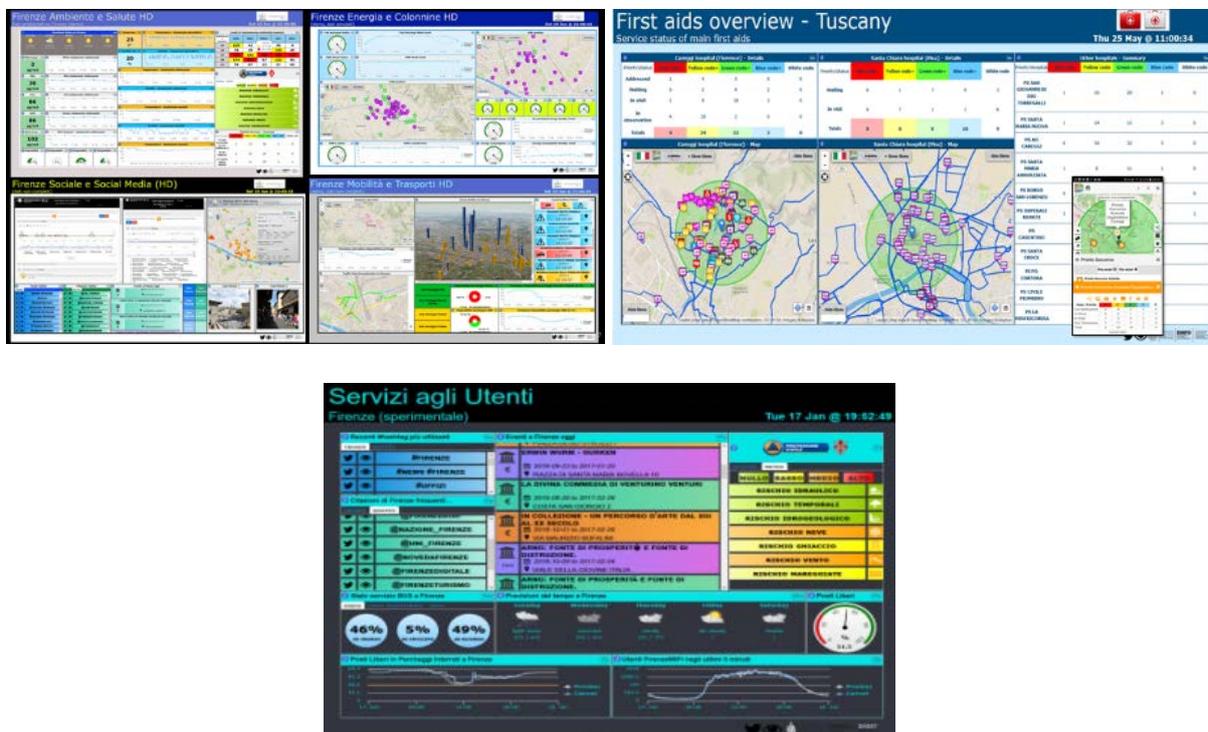
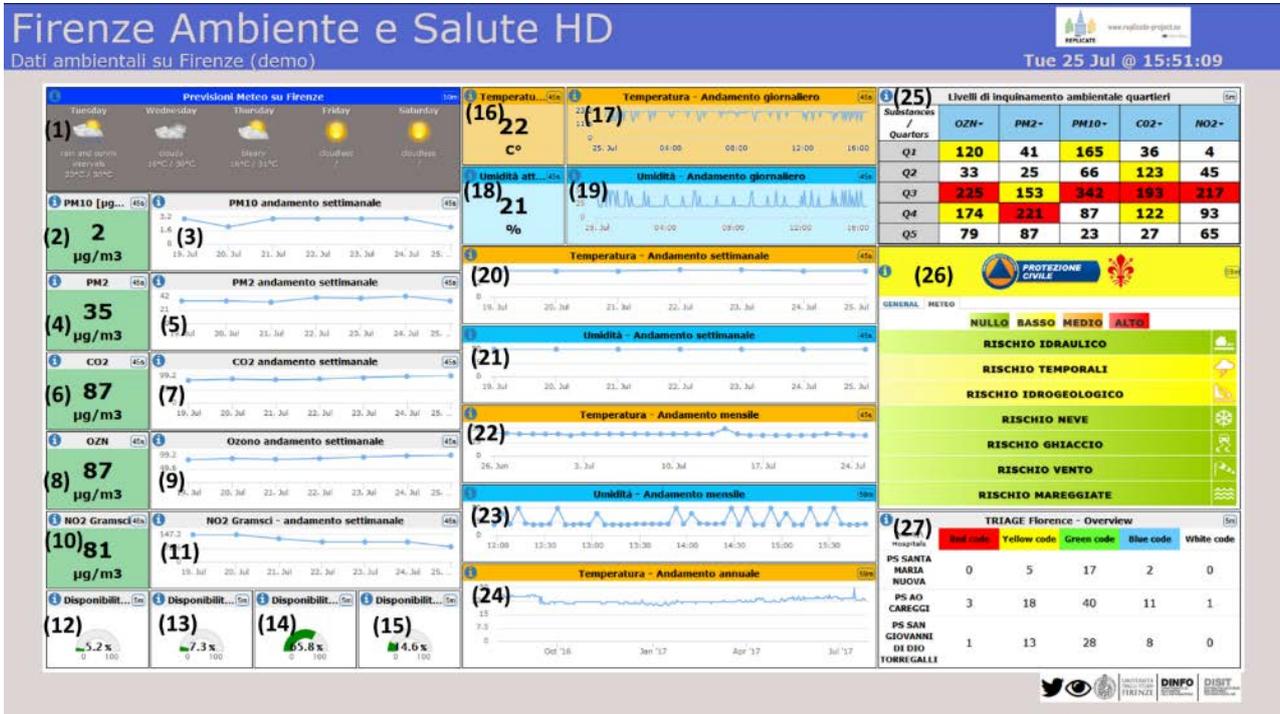


Figure 3: Florence Smart City Dashboards, dashboard reporting first aid status, and a final user dashboard for hotels

- [REPLICATE Dashboard Environment Full HD](#)

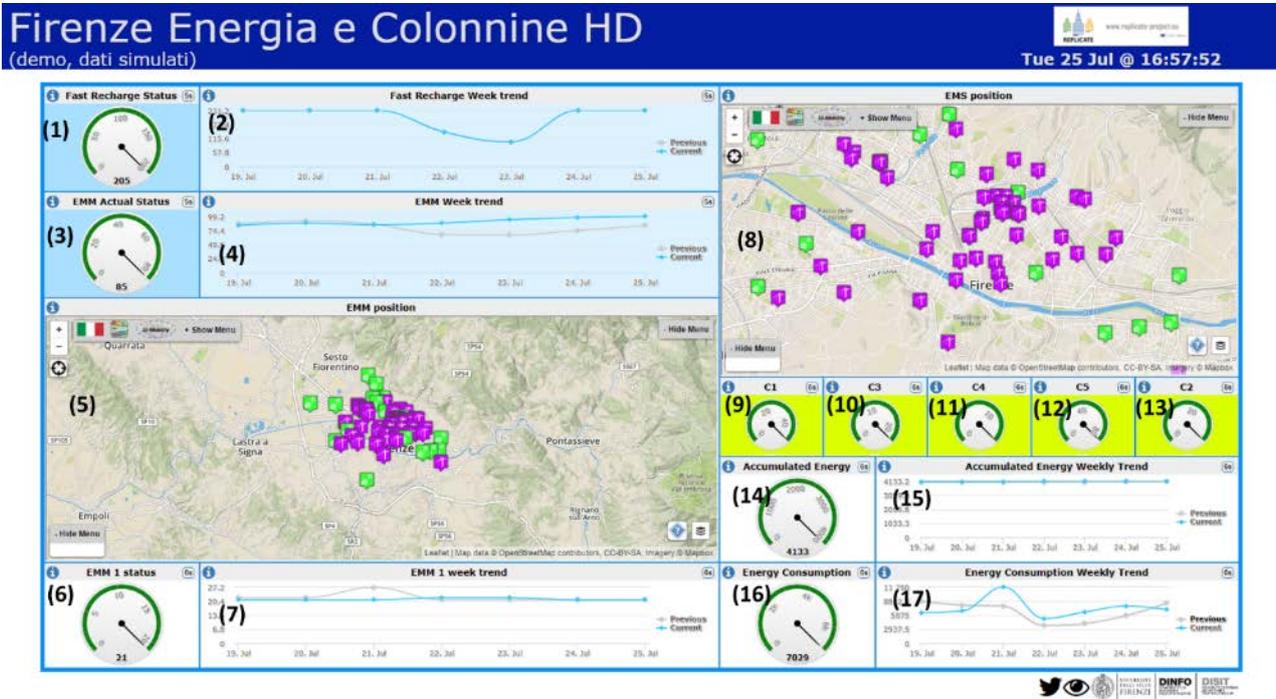
This dashboard shows data referred to environment and health, as described below in detail.



- 1) Meteo forecast in Florence
- 2) PM10 real time value
- 3) PM10 trend in the week
- 4) PM2 real time value
- 5) PM2 trend in the week
- 6) CO2 real time value
- 7) CO2 trend in the week
- 8) OZN (ozone) real time value
- 9) OZN (ozone) trend in the week
- 10) NO2 (Nitrogen dioxide) real time value
- 11) NO2 (Nitrogen dioxide) trend in the week
- 12) Real time parking slots availability for the Santa Maria Novella parking
- 13) Real time parking slots availability for the Mercato Centrale parking
- 14) Real time parking slots availability for the Fortezza parking
- 15) Real time parking slots availability for the Parterre parking
- 16) Temperature in Florence
- 17) Temperature in Florence daily trend
- 18) Humidity in Florence
- 19) Humidity in Florence daily trend
- 20) Temperature in Florence weekly trend
- 21) Humidity in Florence weekly trend
- 22) Temperature in Florence monthly trend
- 23) Humidity in Florence monthly trend
- 24) Temperature in Florence annual trend
- 25) Levels of environmental pollution in Florence neighborhoods
- 26) Civil protection real time alerts
- 27) First aid cues status divided in colored severity levels in real time for the three main hospitals in Florence

- [REPLICATE Dashboard Energy Full HD](#)

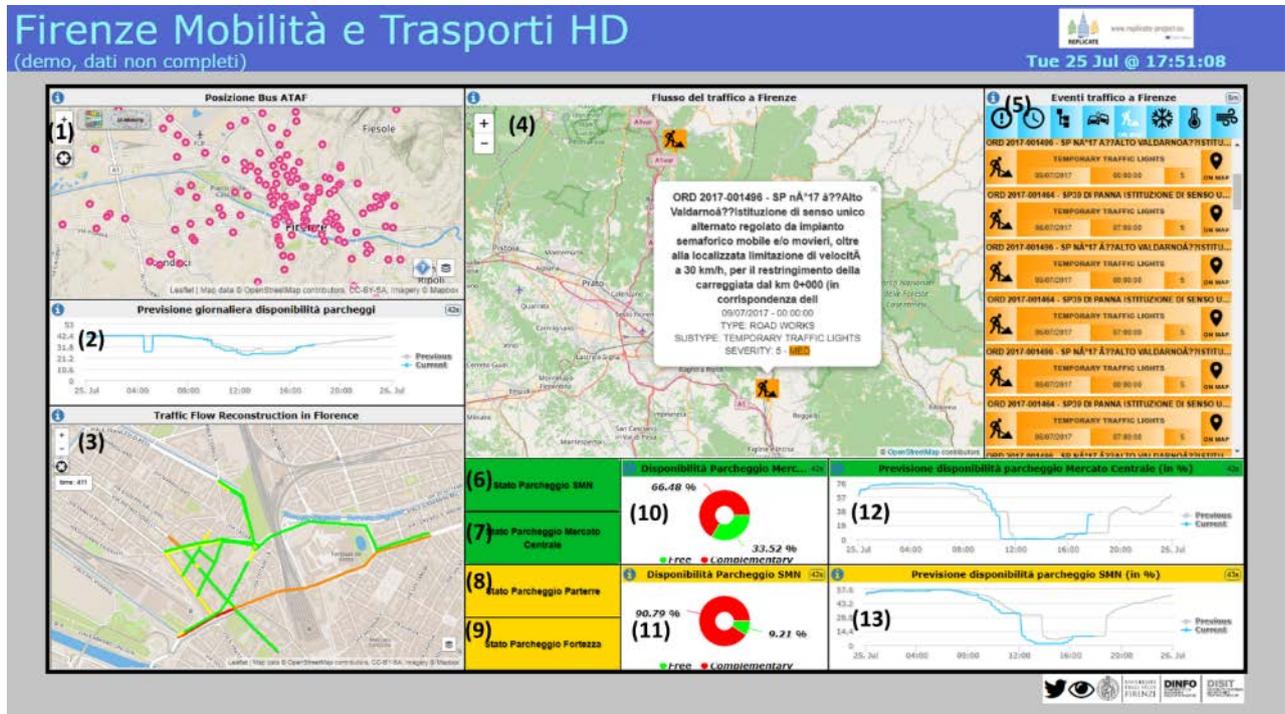
This dashboard shows data referred to energy aspects, as described below in detail.



- (1) Energy consumption of the fast recharge stations
- (2) Fast recharge stations energy consumption weekly trend
- (3) Energy Metering Manager energy consumption
- (4) Energy Metering Manager energy consumption weekly trend
- (5) Energy Metering Manager map (e.g. position of the e-charging stations)
- (6) Single e-charging station energy consumption value. The value is updated according to the selected e-charging station on the map above.
- (7) Single e-charging station energy consumption weekly trend
- (8) Energy management system (EMS) position
- (9) Building 1 real time energy consumption value (and aggregation of EMS data)
- (10) Building 2 real time energy consumption value (and aggregation of EMS data)
- (11) Building 3 real time energy consumption value (and aggregation of EMS data)
- (12) Building 4 real time energy consumption value (and aggregation of EMS data)
- (13) Building 5 real time energy consumption value (and aggregation of EMS data)
- (14) Total Energy accumulated in the day
- (15) Accumulated energy weekly trend
- (16) Total Energy consumed in the day
- (17) Energy consumption weekly trend (current week compared with the previous)

- [REPLICATE Dashboard Mobility Full HD](#)

This dashboard shows data referred to mobility and transport aspects, as described below in detail.



- (1) Real time position of busses in Florence estimated on the basis of the timetable (in alternative if they are not received the scheduled position according to the timelines are reported).
- (2) Parking availability vs the previous week trend (a prediction trend can be shown as well)
- (3) Traffic flow representation and reconstruction from the scattered sensors to the extrapolation in the other points of the city.
- (4) Map showing the traffic events (listed on the (5))
- (5) List of the traffic events. By clicking one event, it is shown on the map on the left (in alternative the 3D maps with the status of the traffic flow sensors and/or parking is reported.)
- (6) Button to activate the real time availability of slots for Santa Maria Novella parking
- (7) Button to activate the real time availability of slots for Mercato Centrale parking
- (8) Button to activate the real time availability of slots for Parterre parking
- (9) Button to activate the real time availability of slots for Fortezza parking
- (10) Real time slots availability in percentage (change on the basis of the parking selected, Santa Maria Novella or Mercato Centrale)
- (11) Single parking availability prediction compared with the previous week trend (change on the basis of the parking selected, Santa Maria Novella or Mercato Centrale)
- (12) Real time slots availability in percentage (change on the basis of the parking selected, Parterre or Fortezza)
- (13) Single parking availability prediction compared with the previous week trend (change on the basis of the parking selected, Parterre or Fortezza)

- [REPLICATE Dashboard Social Monitoring Full HD](#)

This dashboard shows data referred to mobility and transport aspects, as described below in detail.



- (1) Social monitoring in real time with sentiment analysis. The scope is to monitor the social media sentiment to react in case of unexpected events, it also include the last tweets in real time, the sentiment analysis in real time, etc.
- (2) Daily trend social monitoring via Twitter
- (3) Heatmap showing the real time position/density of people in Florence on the basis of the Wi-Fi streaming
- (4) Most cited hashtag in the Social Media Florence channel
- (5) Most cited users in the Social Media Florence channel
- (6) Public events in Florence
- (7) Web Cam buttons related to different locations to be shown in the “Cam Firenze 1” Widget (9)
- (8) Web Cam buttons related to different locations to be shown in the Widget (10)

Other dashboards

- [Service map 3d](#)

A 3D map representing data related to traffic. In particular Car parking slots availability (blue pillars) and road sensors data (orange pillars).

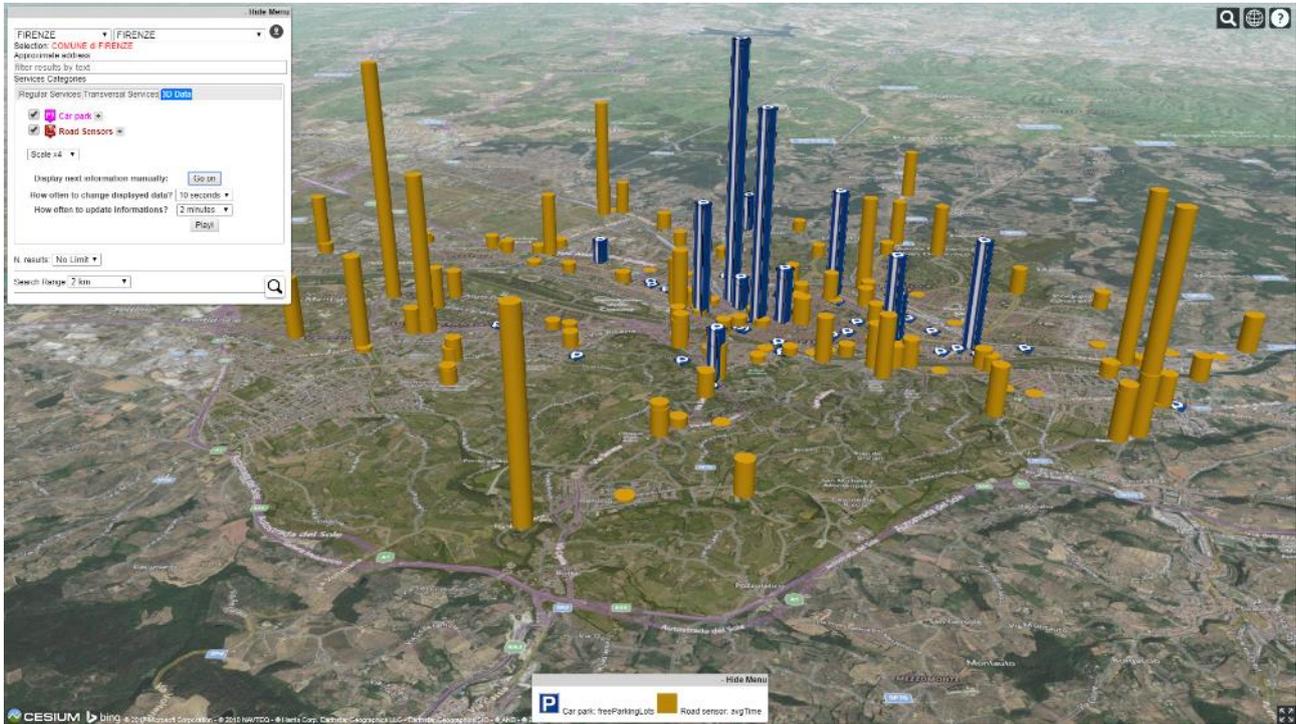
Regarding the Car parking, pillars can represent the following data:

- parking capacity,
- free parking availability,
- occupied parking slots,
- occupancy

Regarding the Road sensors data pillars can represent the following data:

- Average Distance
- Average Time

- Occupancy
- Concentration
- Vehicle Flow
- Average Speed
- Threshold Percentile
- Speed Percentile



- [Cross Widget interaction sample dashboard](#)

[This dashboard shows the functionality related to the interaction between two different widgets, i.e. by clicking one widget, another widget is updated accordingly.](#)

Cross widget interaction 1

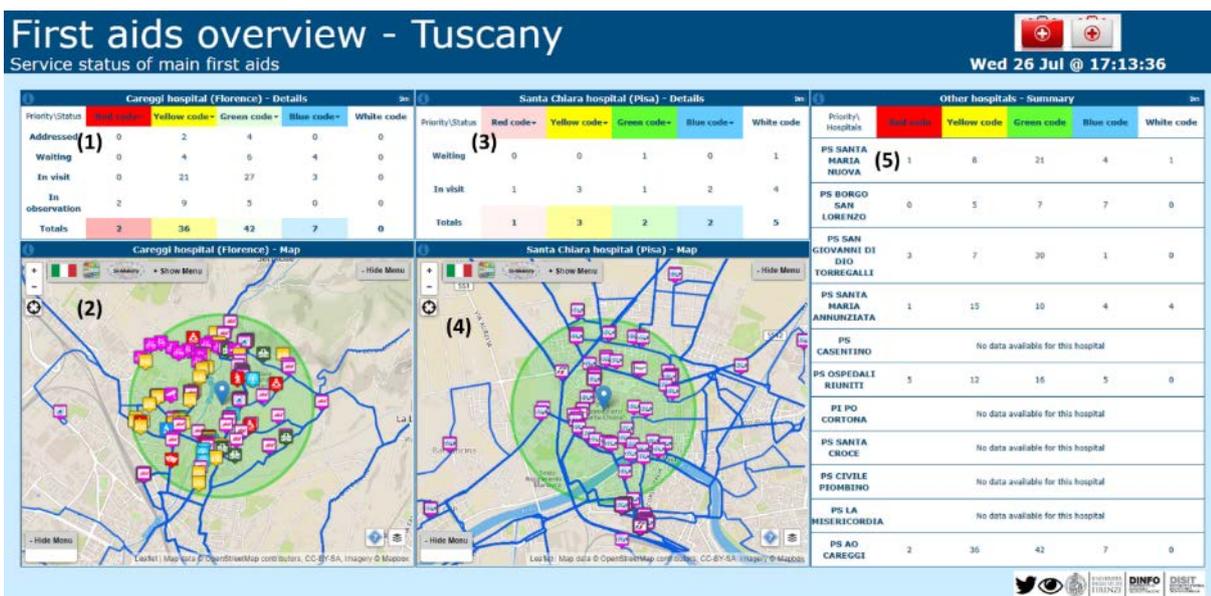
Artistical events, traffic events, city cams

DOT
Tue 6 Jun @ 18:50:21

- (1) Automatically scrolling up and down list of cultural and public events in Florence. By clicking one event in the list, the corresponding position of the clicked event is shown on the maps on the left (map (2) or map (3) in accordance to the event type)
- (2) Map showing the position of the events type “Exhibitions” only;
- (3) Map showing the position of all the other events types, excluded “Exhibitions”;
- (4) Buttons activating four different cameras to be shown in the widget (5)
- (5) Camera view
- (6) Automatically scrolling up and down list of traffic events in Florence. By clicking one event in the list, the corresponding position of the clicked event is shown on the maps on the left (Incidents are shown on map (7), while the other events are shown on map (8));
- (7) Map showing the position of the traffic events “Incidents” only;
- (8) Map showing the position of the all the others traffic events, excluded “Incidents”;
- (9) Buttons activating four different cameras to be shown in the widget (10)
- (10) Camera view

- [First aid sample dashboard](#)

This dashboard shows real time data related to First Aid cues status in Florence Hospitals.



- (1) Real time status of the cues in the Careggi hospital in Florence. Different colors identify the different severity of intervention;
- (2) Map showing services and bus lines available in an area close to the Careggi hospital;
- (3) Real time status of the cues in the Santa Chiara hospital in Pisa. Different colors identify the different severity of intervention;
- (4) Map showing all the bus stops and bus lines close to the Santa Chiara hospital in Pisa;
- (5) Real time status of the first aid cues in the major hospitals in Tuscany. Different colors identify the different severity of intervention.

- [Twitter Vigilance dashboard](#)

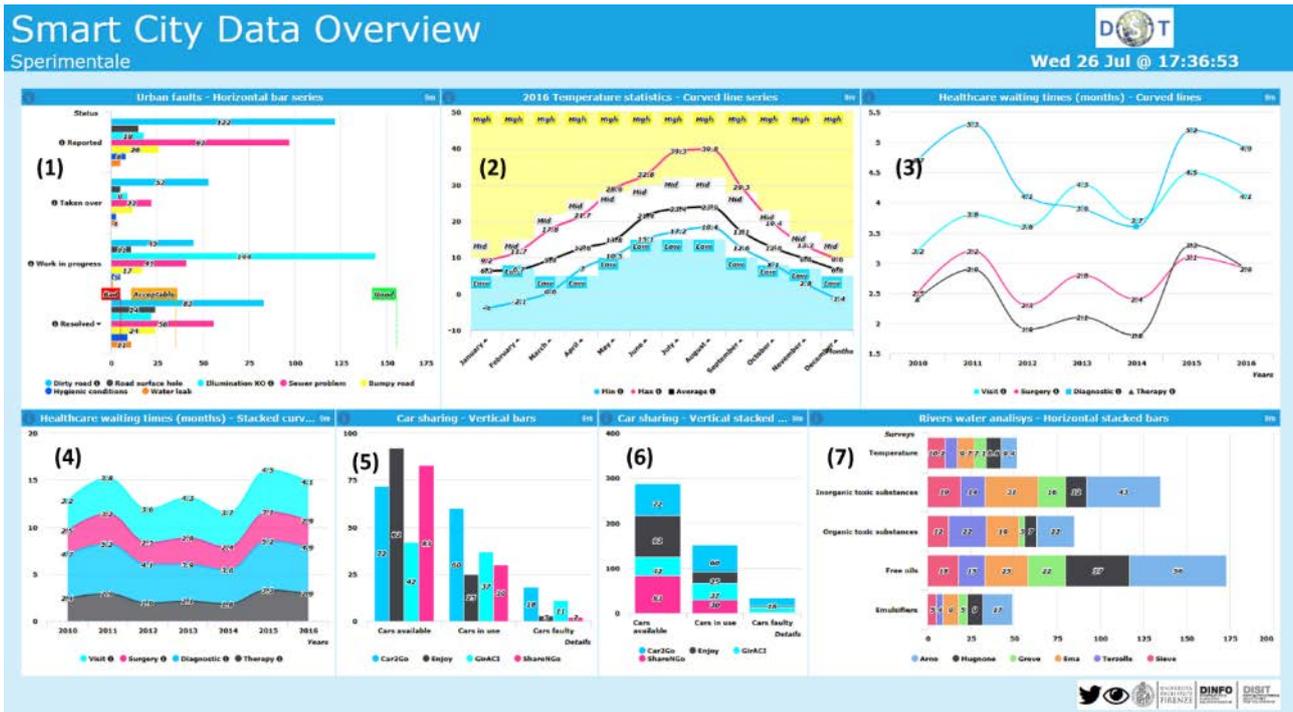
- (1) Daily trend social monitoring via Twitter

- (2) Social monitoring in real time with sentiment analysis. The scope is to monitor the social media sentiment to react in case of unexpected events, it also include the last tweets in real time, the sentiment analysis in real time, etc.
- (3) Most cited hashtag in the Social Media Florence channel
- (4) Most cited users in the Social Media Florence channel
- (5) Informative text



- [Smart City Data Overview sample dashboard](#)

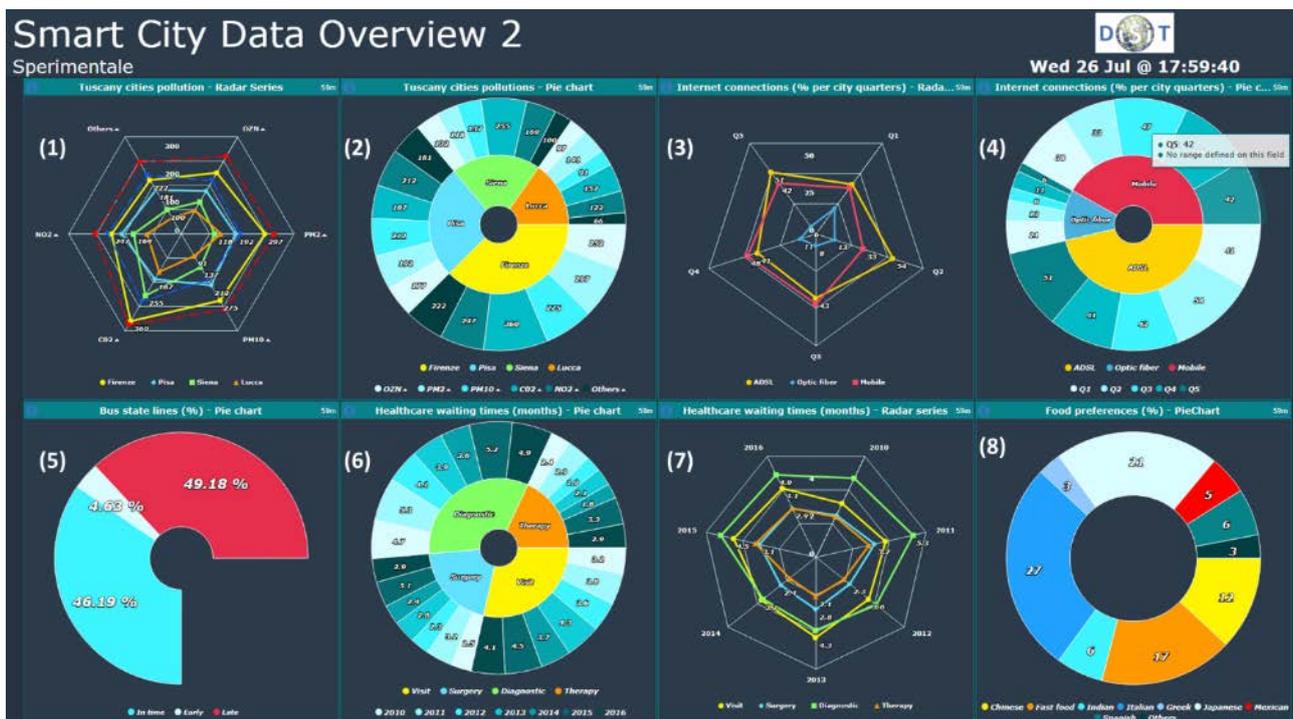
This dashboard shows different data representations, with different widgets graphic styles.



- (1) Horizontal bar series showing the status of some urban problem. Different colors represent the different types of problem. The vertical axis (ordinate axis) represent the status of the problem (Reported, Taken over, Work in progress, Resolved);
- (2) Curved line series widget showing the yearly temperature trend, for minimum, maximum and average temperature;
- (3) Curved lines widget showing the average annual healthcare waiting time from 2010 to 2016;
- (4) Filled curved lines areas widget showing the average annual healthcare waiting time from 2010 to 2016 (same data of (3) but represented in a different mode);
- (5) Vertical bars widget representing the car sharing status, divided in Cars available, Cars in use and damaged cars. Different colors identify different car sharing brands;
- (6) Vertical filled bars widget representing the car sharing status, divided in Cars available, Cars in use and damaged cars. Different colors identify different car sharing brands (same data of (5) but represented in a different mode);
- (7) Horizontal stacked bars widget showing the data of rivers water analysis. Different colors identify different rivers. The vertical axis (ordinate axis) represent the different parameter analyzed (temperature, inorganic substances, organic substance, oils, emulsifiers).

- [Smart City Data Overview 2 sample dashboard](#)

[This dashboard shows different data representations of the radar series widget and pie chart widget, with different graphic styles.](#)



- (1) Radar series widget showing data on pollution for different Tuscany cities;
- (2) Pie chart widget showing data on pollution for different Tuscany cities (same data as (1));
- (3) Radar series widget showing data on percentage of internet connection for the different neighborhoods in Florence;
- (4) Pie chart widget showing data on percentage of internet connection for the different neighborhoods in Florence (same data as (3));
- (5) Pie chart widget showing data on percentage of busses in time, in advance, in delay;

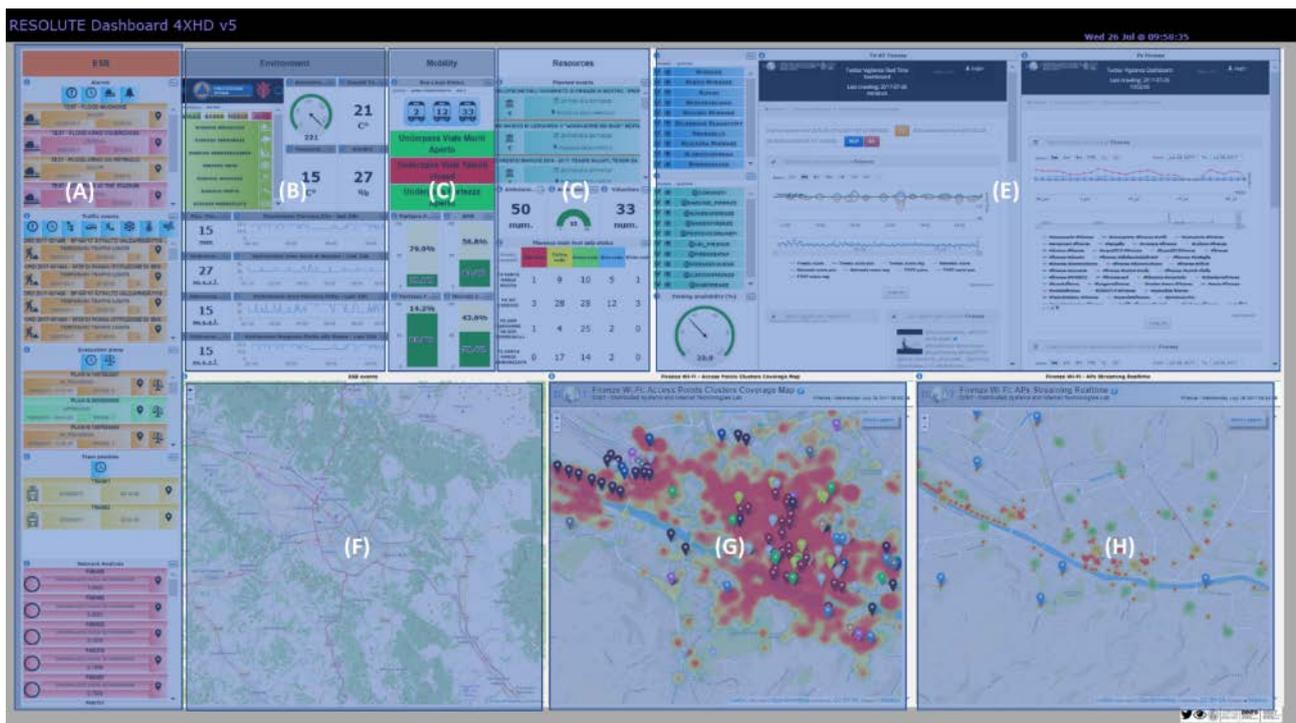
- (6) Pie chart widget showing the average annual healthcare waiting time from 2010 to 2016;
- (7) Radar series widget showing the annual, average waiting time for the different types of hospital services;
- (8) Pie chart widget showing the food preferences in percentage. Different colors identify the different type of cuisine (Chinese, Fast food, Indian, Italian, Greek, Japanese, Mexican, Spanish, Others).

- **Resilience dashboard**

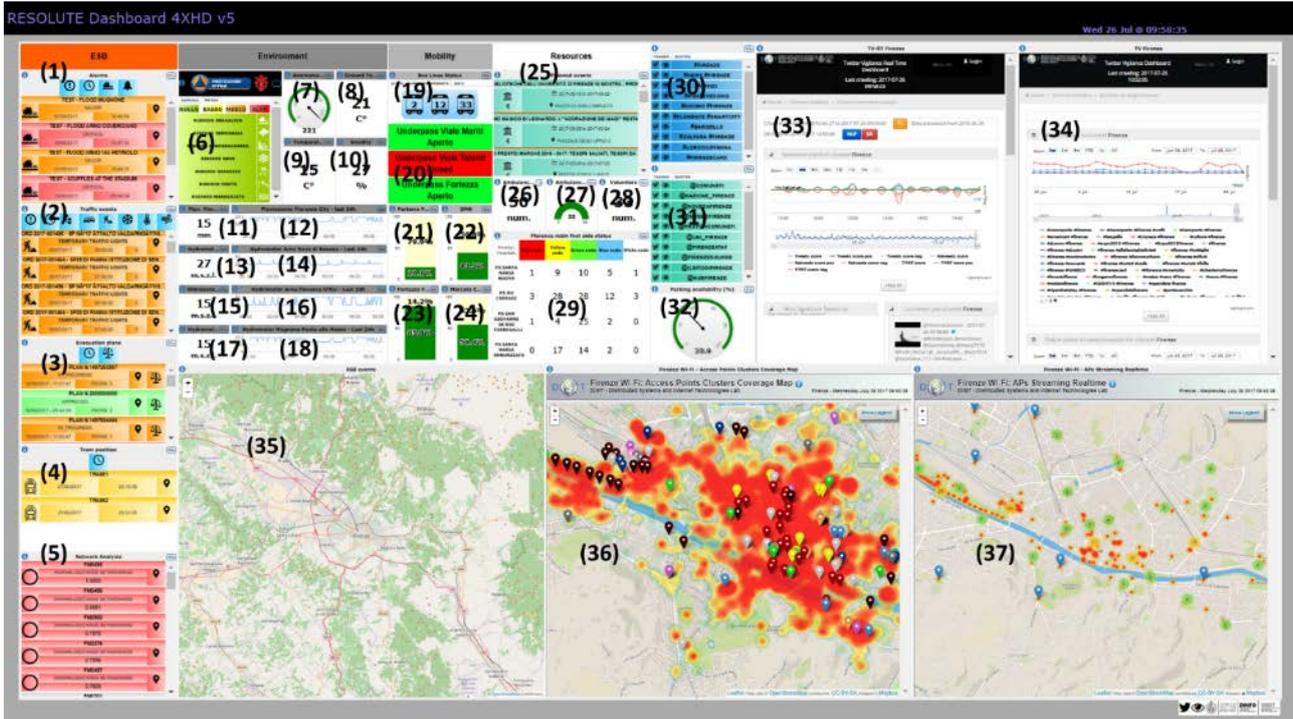
The Resilience Dashboard represents the synthesis of the status of the Urban Transport System (UTS) and it is thought for being installed in each control room of the interested stakeholders. The Dashboard has been designed to be used by different users such as Civil protection, Mobility dept., Urban Police, etc.

As shown in the following image, the widgets have been grouped according to the domain they refer to:

- A. messages coming from the different DSS and published in the ESB (ESB column);
- B. indicators referred to the most important environmental aspects (Environment column)
- C. indicators referred to mobility aspects (Mobility column)
- D. indicators referred to resources availability in the system (Resources column)
- E. Social media section
- F. Location on the map of the events received by the ESB (1)
- G. Clustering representation of the WiFi Access Points
- H. Real time people movements based on the wifi streaming



In details, the data represented in the resilience dashboard are the following:



- (1) Alarms
- (2) Traffic events
- (3) Evacuation plans
- (4) Real time tram positions
- (5) Network analysis
- (6) Civil protection alerts
- (7) Anemometer value
- (8) Ground temperature
- (9) Actual temperature
- (10) Actual humidity in percentage
- (11) Pluviometer value in Florence city
- (12) Pluviometer trend in Florence city in the last 24 hours
- (13) Hydrometer value in the Arno river, Nave di Rosano zone
- (14) Hydrometer trend in the Arno river, Nave di Rosano zone in the last 24 hours
- (15) Hydrometer value in the Arno river, Uffizi zone
- (16) Hydrometer trend in the Arno river, Uffizi zone in the last 24 hours
- (17) Hydrometer value in the Arno river, Ponte alle mosse zone
- (18) Hydrometer trend in the Arno river, Ponte alle Mosse zone in the last 24 hours
- (19) Real time busses status
- (20) Underpasses status
- (21) Parterre parking slots availability (in percentage)
- (22) Santa Maria Novella parking slots availability (in percentage)
- (23) Fortezza parking slots availability (in percentage)
- (24) Mercato Centrale parking slots availability (in percentage)
- (25) Public and cultural events list in the city
- (26) Number of available ambulances
- (27) Number of used ambulances
- (28) Number of available volunteers
- (29) First aid status of the cues in the main hospitals in Florence
- (30) Most cited hashtags in the Florence social media channel
- (31) Most cited users in the Florence social media channel
- (32) Percentage of parking available in Florence

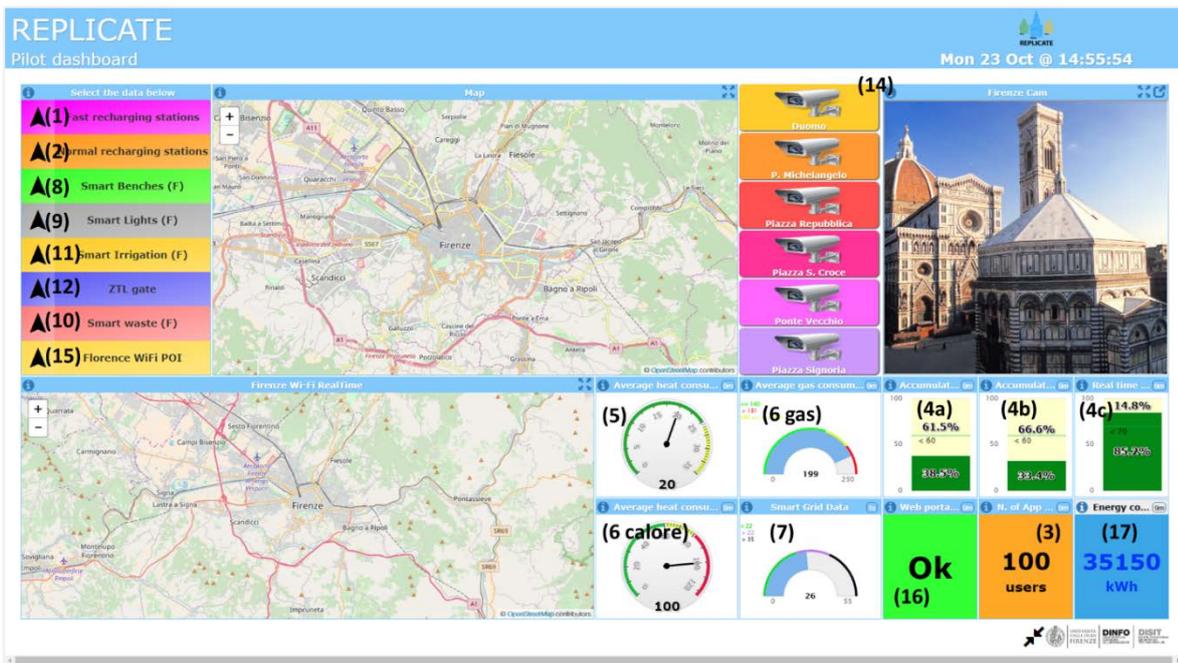
- (33) Social monitoring in real time with sentiment analysis. The scope is to monitor the social media sentiment to react in case of unexpected events, it also include the last tweets in real time, the sentiment analysis in real time, etc.
- (34) Daily trend social monitoring via Twitter
- (35) Map showing the location of the event listed in the ESB column (A)
- (36) Heatmap showing the Clustering representation of the WiFi Access Points
- (37) Heatmap showing the real time position/density of people in Florence on the basis of the Wi-Fi streaming

- [Smart City Control Room Console](#)



Entry point to access different dashboards realised for the REPLICATE H2020 project.

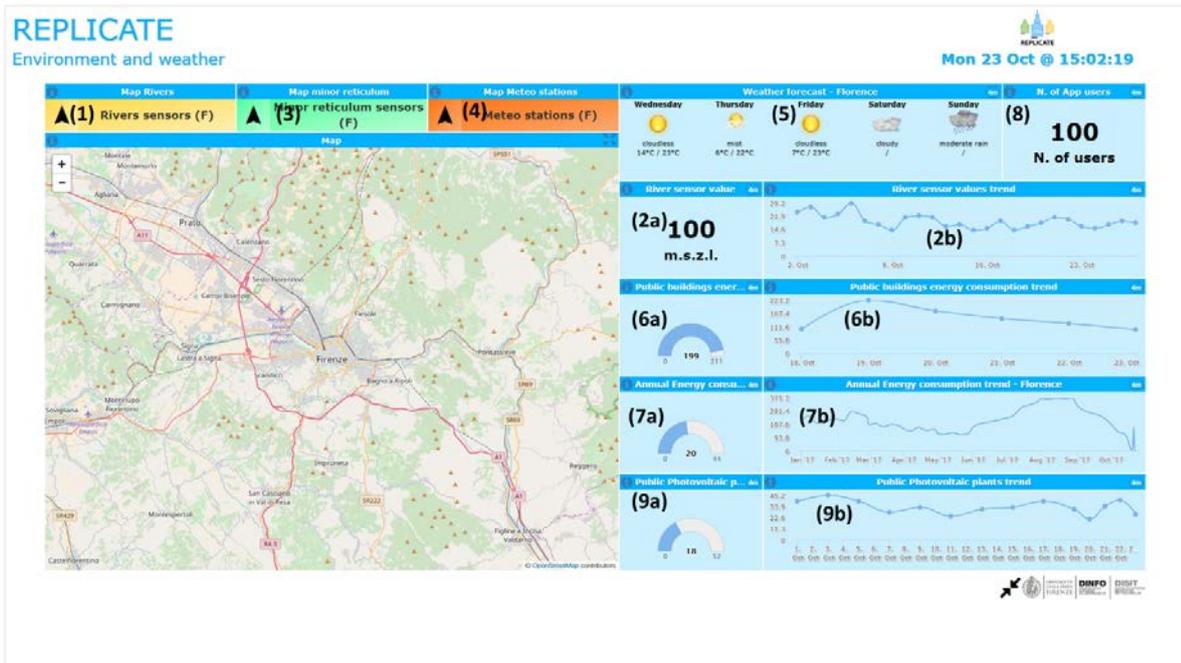
- [REPLICATE pilot dashboard](#)



- 1) Position of ReCharging station location and status (free/occupied, consumption), Fast

- 2) Position of ReCharging station location and status (free/occupied, consumption), normal
- 3) Total number of users of the Energy mobile app
- 4) Status of the energy accumulator at Le Piagge (real time (a), stored (b), used (c))
- 5) Energy termic consumption at Le Piagge buildings
- 6) Consumption data for heat (6 calore) and gas (6 gas)
- 7) Energy Smart Grid Data
- 8) Smart Benches location and real time data (on widget 4a)
- 9) Smart light location and real time data (on 4b and 4c)
- 10) Smart Waste location and real time data (on 7)
- 11) Smart irrigation location and real time data (on 4c)
- 12) ZTL gates location and real time flow
- 13) People flow in real time, WiFi based
- 14) Video camera location and real time video
- 15) WiFi POI in Florence
- 16) Status of the Web portal
- 17) Energy data consumption inserted via mobile App

- [Environment and weather dashboard](#)



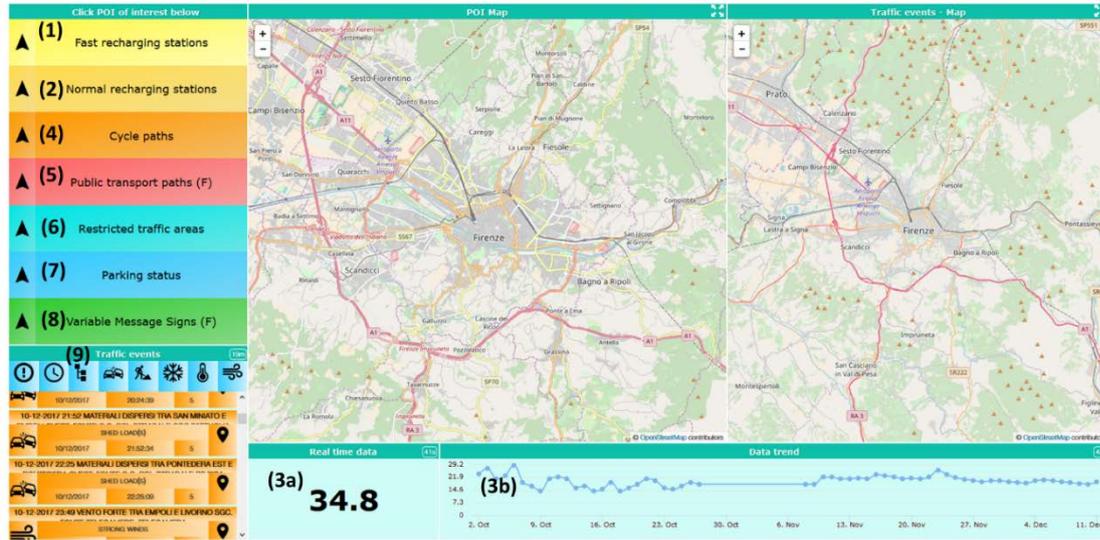
- Rivers level location (1) and real time data (2a) and trend (2b)
- Minor reticulum rivers level location (3) and real time data (2a) and trend (2b)
- Meteo station location (4)
- Meteo station real time data (2a and 2b)
- Meteo forecast (5)
- Energy consumption public buildings – lighting: real time data (6a) and trend (6b)
- Total annual energy consumption in the municipality of Florence: real time data (7a) and trend (7b)
- Total number of energy mobile app users (8)
- Energy production of public photovoltaic systems: real time data (9a) and trend (9b)

- [Mobility and Transport dashboard](#)

Mobility and Transport

REPLICATE

Mon 11 Dec 15:58:04



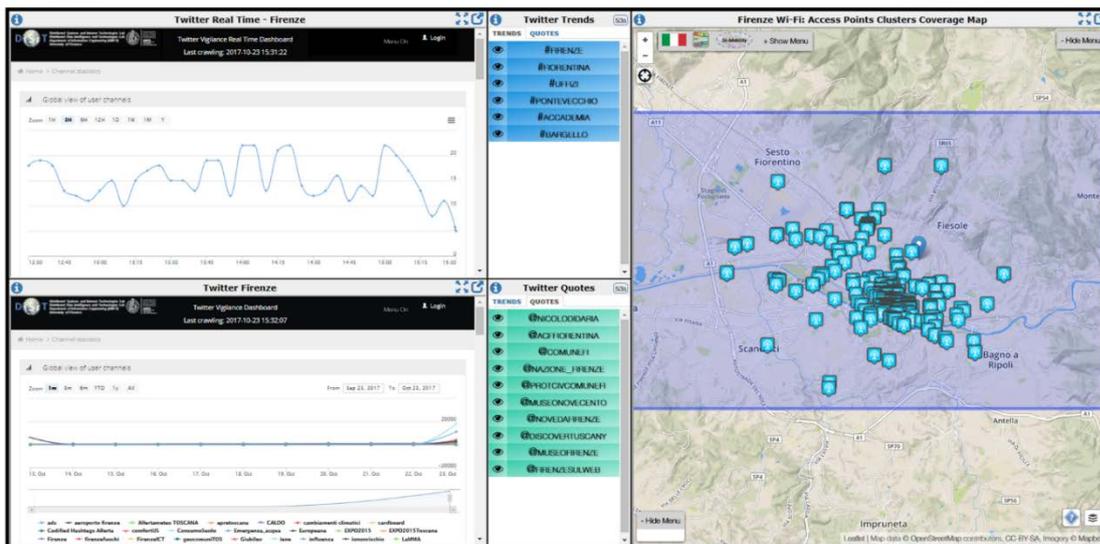
- Fast ReCharging station location (1), real time data (3a) and trend (3b)
- Normal ReCharging station location (2), real time data (3a) and trend (3b)
- Bicycle paths (4)
- Public transport paths and bus stops (5)
- Restricted traffic areas location (6) and real time data (3a)
- Parking location (7), real time data (3a)
- Variable Message Signs (8)
- Traffic events (9)

- [Web and Social dashboard](#)

REPLICATE

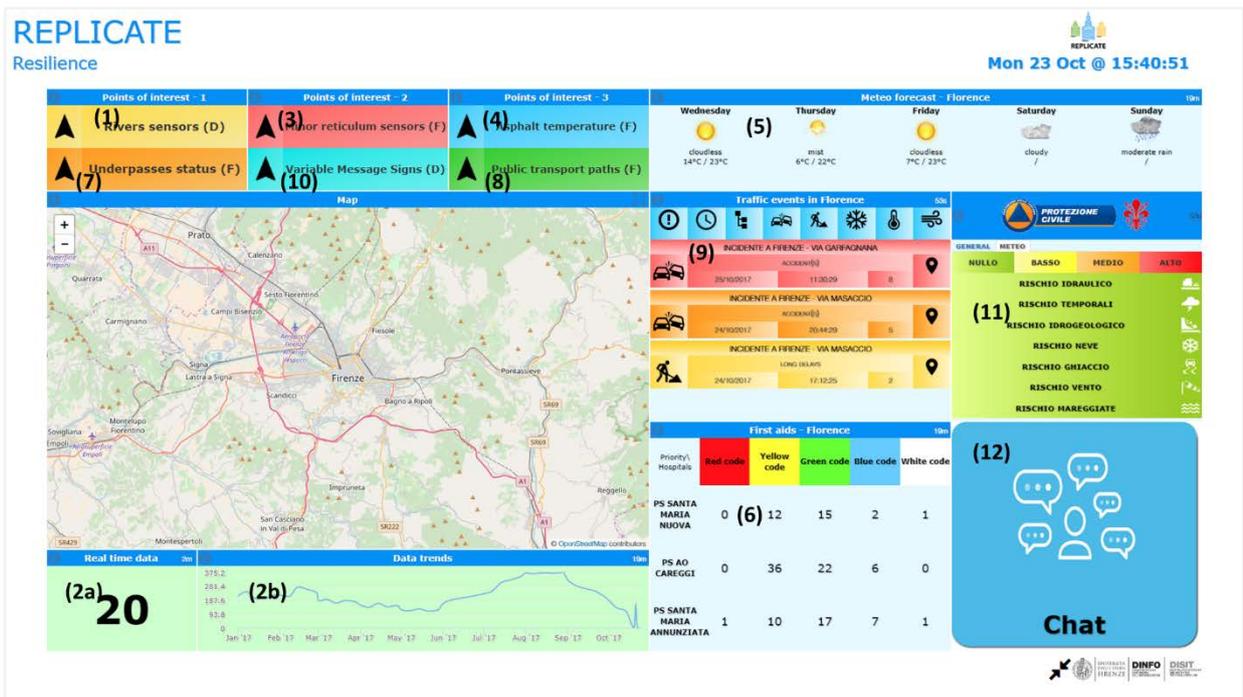
Web and Social

Mon 23 Oct @ 15:32:17



- People flow real time heatmap based on WiFi (1)
- Twitter real time sentiment analysis (2)
- Twitter daily analysis (3)
- Twitter most used hashtags (4)
- Twitter most quoted users (5)

- [Resilience dashboard](#)



- Rivers level location (1) and real time data (2a) and trend (2b)
- Minor reticulum rivers level location (3) and real time data (2a) and trend (2b)
- Asphalt temperature location (4) and real time data (2a) and trend (2b)
- Meteo forecast (5)
- First Aid cues real time status in main hospitals in Florence (6)
- Underpasses location and status (7)
- Public transport paths and bus stops (8)
- Traffic events in Florence (9)
- Variable Message Signs (10)
- CivilProtection alerts in real time (11)
- Private chat with services and security stakeholders (12)

- [Real Time Sensors via ServiceMap3D](#)

This Dashboard represents real time data in a 3D map



- [Tuscany weather dashboard 1](#)

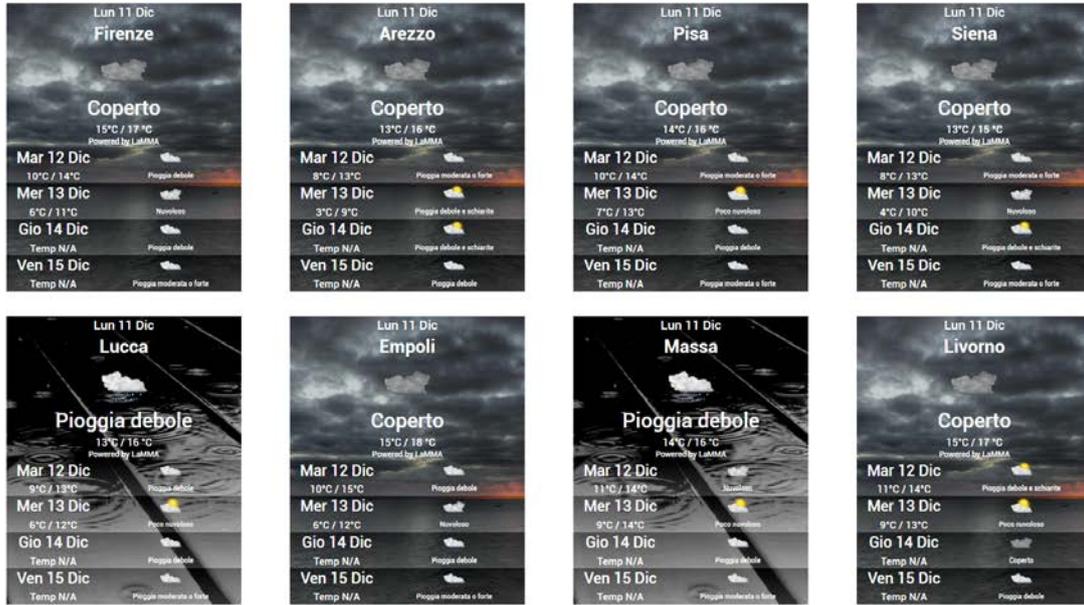


- [Tuscany weather dashboard 2](#)

Tuscany weather dashboard 2

Widgets verticali - Italiano

Mon 11 Dec 18:30:18



- [Tuscany weather dashboard 3](#)

Tuscany weather dashboard 3

Compact widgets - English and italian

Mon 11 Dec 18:32:02

