

Smart City Control Room Dashboards: Big Data Infrastructure, from data to decision support

P. Bellini, D. Cenni, M. Marazzini, N. Mitolo, P. Nesi, M. Paolucci

DISIT lab (<http://www.disit.org> <http://www.km4city.org>)

University of Florence, pierfrancesco.bellini@unifi.it, daniele.cenni@unifi.it,
mino.marazzini@unifi.it, paolo.nesi@unifi.it, michela.paolucci@unifi.it

Abstract: Smart City Control Rooms are mainly focused on Dashboards which are in turn created by using the so-called Dashboard Builders tools or generated custom. For a city the production of Dashboards is not something that is performed once forever, and it is a continuous working task for improving city monitoring, to follow extraordinary events and/or activities, to monitor critical conditions and cases. Thus, relevant complexities are due to the data aggregation architecture and to the identification of modalities to present data and their identification, prediction, etc., to arrive at producing high level representations that can be used by decision makers. In this paper, the architecture of a Dashboard Builder for creating Smart City Control Rooms is presented. As a validation and test, it has been adopted for generating the dashboards in Florence city and other cities in Tuscany area. The solution proposed has been developed in the context of REPLICATE H2020 European Commission Flagship project on Smart City and Communities.

Keywords: smart city dashboard, decision support system, widget, control room.

1. Introduction

In the development of a Smart City there is a great emphasis to the set-up of the so-called Smart City Control Room, SCCR. A SCCR is an area in which all the data are collected, aggregated and where high-level data/results are summarized and made accessible for the decision makers and shared to the city operators. In large metropolitan cities, the SCCR includes large panels/monitors (even covering large walls) in which the status of the city is reported in real-time presenting the view of the city with some synthesis, predictions, alert of data regarding: mobility, energy, social activities, environment, weather, public transportation, people flow, health, water, security, ICT, governmental, first aid, civil protection, police (118/112/911), fire brigade, hospital triage, and thus almost all the city resources expressed via KPI (Key Performance Indicator). Most of the KPI are representative of the status of resources deployed in the city and may be not geo-localized. Some of the city monitored resources are representative of the critical infrastructures for the city functionality and for the life of city users such as: transportation, energy,

security, health, water, civil protection, ICT, etc. In medium sized cities, the daily management of city resources is performed by a set of *city operators*, which could be legally independent with respect to the central municipality. Thus, they autonomously manage their control rooms, accessing and rendering their own data to take their own decisions which may be limited in scope, and according to some defined protocols and strategies, [1], [2]. For example, when the energy network has a problem in an area of the city the energy is rerouted to reach the all possible subareas via a different path; when the water service network has a problem on major distribution tubes, the water is provided in other means (for example using tanks); in presence of traffic congestion the red-light timing is acted to facilitate the flow and bus paths are changed/rescheduled according to planned schemas.

Once identified and understood the needs of having an integrated SCCR, it is necessary: (i) to choose what must be shown on the panels on walls; (ii) to choose what must be shown on computers of the operators in the rooms and remotely connected; (iii) to understand how the data have to be collected and computed (in the case of prediction and early warning). The ingestion, aggregation and data analytics processes are very complex to be managed since the information is heterogeneous (different format, providers, protocols, etc.) and the total amount of data is a Big Data problem, moreover, the final indicators, provisions and suggestions calculated by these complex processes must be easily understood by the observers of the panels. It is a problem of data representation which also must consider the competence of the final users: citizens /observers/operators/experts/decision makers/etc. In most cases, the final users have to be trained to understand the data and graphics representations. They must become confident on what they see to understand in deep all the single details represented on the screen. They are not going to have time to learn when a critical event happens. Some data or events are easy to be represented and understood in a dashboard, for example: a traffic representation observing the city map with red, orange, yellow, and green segments on the streets; a sensors measuring some values like temperature, humidity percentage, etc.; while it is more difficult to comprehend other, more complex, kinds of

data/results/provisions, for example: tables of pollution and pollution; traffic flow trends by numbers, etc. To make it easy to read even the most complex metrics, it is possible to use: alarm signals in red, blinking signal, etc. [2].

From the technical point of view, the tools for rendering information on SCCR are typically called Dashboards and are supported by Big Data aggregation tools [4]. The Dashboards should be capable to present real time data in several different manners with real time updates on screen autonomously H24 7/7 days, according to the refresh time of each data source, and have to be interactive to allow the users to make drill down activities on data to better understand the situation and the context. Dashboards 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.). In most cases, business intelligent tools may access to data with faceted indexing and search, for example in SOLR or ElasticSearch, [24], [25]. Those kinds of Dashboards are focused on single view of data, filtering and drilling down on data, rather than representing the city KPI and status. Examples of tools for drilling down on time and facet can be obtained by using Apache Banana ([26]), or HUE ([27]).

Moreover, the concept of Dashboard for SCCR is also often confused with the data aggregator tool that is a fundamental tool for the Control Room and city control in general and can be regarded as the back office of the Control Room. Many solutions for control rooms and their back-offices has been proposed such as those of IBM [5] on services for citizens, business, transport, communication, water and energy; [6] on governmental, educational, e-health, safety, energy, transport and utilities; etc. Most of these solutions present a multi-tier architecture ranging from 3 to 6 layers [7], [8], [9].

In this document, the Dashboard solution developed in the context of REPLICATE research and development projects of the European Commission is presented. Replicate is an SCC1 project of the European Commission on H2020 ([10]). The solution proposed is based on Km4City Smart City Ontology ([11]) and data aggregator [4], [10], [22]. Please note that the Dashboard Management System of DISIT Lab is released as Affero GPL Open Source on GitHub, see DISIT lab page. The present solution is managing more than 1.2 million of complex events/data per day.

The paper is structured as follows. In Section 2, the requirements of smart city control room are discussed. Section 3 presents the adopted smart city architecture. In Section 4, the dashboard system for the smart city control room is presented with its architecture. Section 5 reports a set of experimental results and lesson learnt. Conclusions are drawn in Section 6.

2. Smart City Control Room Requirements

In this section, the main requirements of Dashboard systems for SCCRs are summarized. They have been collected during the above-mentioned research project by interviewing a number of operators and decisions makers belonging to several cities and nationalities.

A SCCR dashboard system is substantially a Decision Support System tool, DSS, since it provides evidence of critical conditions, and may offer solutions. On this regard, it may integrate/exploit artificial intelligence algorithms, for example, reporting prediction, identifying anomalies, manifesting early warning, providing relationships among entities exploiting inference geospatial reasoning about what is located in the city: resources, structure, people, areas, critical infrastructures, etc. [10], [17], [19], [20].

According to our analysis, a Dashboard system for smart city must be capable to:

- Show dashboard on web browser in a H24/7 modality;
- show data on widgets according to several graphic paradigms (tables, graphs, histograms, maps, Kiviat, lists, tv camera, heatmaps, weather, critical city events, etc.) with a level of interactivity and animation;
- show data on autonomous and connected/synchronized widgets;
- collect, show and keep update on screen data with automated refresh for each view, and real time data according to the even driven paradigm;
- show data both real time and historical, allowing the drill down on time, space and relationships among data and city entities;
- collect and show data coming from different big data and classic data sources (SQL, NoSQL, RDF, P2P, API, SOLR, etc.) also in aggregated manner;
- compose the Dashboards as a set of graphic and integrated widgets that can be separately set up assigning several parameters: data source, size, colors, shape, etc.;
- work with large amount of data providing high performances, as short response time;
- compute alarms, and provide support by a flexible notification system capable to send alerts, activate tickets for maintenance, automate actuators, post on social, etc.;
- provide actuators widgets together with showing graphs, and capable to act on IOT Devices;
- provide support for collaborative production of dashboards and for co-working;
- provide support for embedding dashboards into third party web pages;
- provide data engine for collecting connection response time on different protocols, and for verifying the consistency of web pages via HTTP;
- integrating with IOT Applications by managing real time data and connecting its actuators to real time IOT applications;

- integrate dashboards in more complex dashboard systems;
- support authentication and authorization with the most general approaches such as LDAP, Kerberos, etc.;
- collect and get data from batch resources and in real time, using a large range of protocols and formats.

This means that each Dashboard should be composed by several configurable Widgets, each of them can collect data coming from several data sources. On each data stream, one or more criteria as firing conditions should be set up for the notification of alerts, intervention, etc.

In small cities, with about 100.000 inhabitants the number of relevant data sources to be integrated by the data aggregator and represented in Dashboard can be in order of 20-30 while in larger cities they can rapidly grow for the presence of multiple operators for each utility. Thus, the complexity is also greater as the number of stakeholder actors involved in.

Therefore, before starting with the development of the proposed Dashboard solution, several state-of-the-art solutions and proposals have been analyzed. As nonfunctional requirements, the Dashboard system must be scalable, interoperable with several tools, open source, robust, usable, secure in protecting data views according to the GDPR [17] and flexible.

A set of solutions, both commercial and open-source, have been analyzed to identify a functional platform to be adopted. Most of the solutions which are present at the state of the art derive from business intelligence solutions (e.g., SpagoBI, Tableau, OpenDataSoft, etc. [13], [14], [16]), in which the tools provide some data mart (data virtualization) tool to access data sources and thus have powerful tools in this sense. On the contrary, they provide limited capabilities and tools on rendering and dashboard for control rooms that must stay H24/7, rendering specific kind of structured data. For these

reasons, several specific custom solutions have been proposed by many cities such as: London (<https://data.london.gov.uk/>), Amsterdam (<http://citydashboard.waag.org/>), Dublin, etc. On the bases of the analysis made, regarding the solutions available on the market, none of them satisfied all the requested functionalities and functional aspects, above described, and this is the reason why we decided to start the development of our solution, ([14]).

3. Smart City Architecture

This section presents the overview of the Smart City architecture which is presently in place in the Tuscany area. With the aim of producing a smart city infrastructure for stimulating sustainable mobility, smart energy, and smart utility in the city, a data aggregator has been developed (see **Figure 1**). It presents a front-end layer for the City Dashboard and control room, Smart City API for web and mobile App, decision support tools, personal assistants via mobile App, participative portals, crowd sourcing, etc. The data aggregation infrastructure also supports Data Analytics and Data Intelligence based on integrated data collected from public administrations open data, private data from operators, and personal data coming from social media and city users. In this paper, the architecture enabling the construction of the Control Room in terms of Dashboards and Data Aggregator is described.

In the Km4City layer, **City Operators and Data Brokers** provide data which are collected by using streams, data driven and/or ETL processes which are scheduled on the Big Data processing back office based on DISCES (Distributed Smart City Engine Scheduler) tool. Among the data collected those provided in Open Data from the municipalities, Tuscany region (Observatory of mobility), LAMMA weather agency, ARPAT environmental agency, etc., and several private data coming from City/Regionals Operators: mobility, energy, health, cultural heritage,

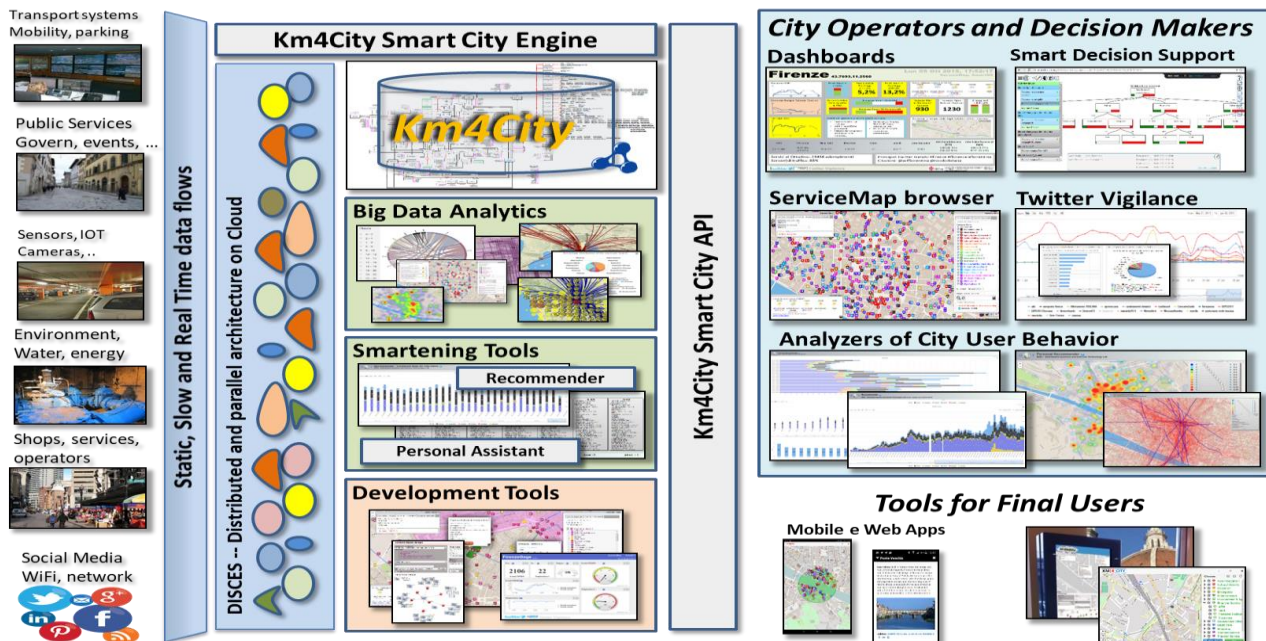


Figure 1: Km4City Architecture for Smart City.

services, tourism, wine and food services, education, wellness, environment, civil protection, weather forecast, etc. Once the data are collected, the back office activates several processes for improving data quality, reconciling data and converting data into triples for the RDF store of the Knowledge Base (when needed) [10], [4], implemented by using a Virtuoso triple store. DISCES is allocating processes on several virtual machines allocated on the cloud according to their schedule.

For semantic aggregation of data and service, it has been decided to exploit and improve the Km4City Ontology ([10]) as the main ontological model. Km4City is modeling multiple domain aspects related to mobility, services, Wi-Fi, cultural services, energy, structure (streets, civic numbers, public structures, geographic locations, green areas), sensors, busses, smart sensors, parking, city services, transportation, events, pharmacies, hospital and real time data of first aid, environment (with pollution and pollination, weather forecast, weather conditions), and private mobility with fuel prices.

In the smart city architecture, in addition to the RDF store for the knowledge base, a number of noSQL Stores (namely: HBase and MongoDB, [28], [29]) are adopted for storing tabular data as those arriving from sensors and user profiles, respectively.

4. Dashboard System Architecture

In Figure 2, the general architecture of the Dashboard solution is presented. The main components of the architecture are described in the following.

- **Data Aggregator** is a set of tools for collecting data

reconciling them to the same city entities. To this end in the proposed architecture the Km4City Aggregation has been adopted to have all data fit into the Km4City Ontological model. Thus, Data are provided from the Aggregator with Smart City API (rest CALL), SPARQL, SOLR and/or SQL queries.

- **Data Collector** is a multi-process engine (also called Dashboard Engine) for acquiring data from multiple data sources: SQL, noSQL, RDF/SPARQL, API, SOLR, etc. ([25],[28],[29],[30],[31]), by using multiple protocols: HTTP, HTTPS, ODBC, WebSocket, etc. The Data Collector needs to have a configuration for each acquisition process, which produce a result, also named **Metric** or Measure. Some of these **Metrics** may be saved into a local data base for historical reason. To finalize the queries to be performed for collecting Metrics, specific tools may be used for **Data Mart** such as database browsers and drilling down into 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 Data Collector, with its processes to acquire Metrics, is also capable to perform the real time assessment of firing conditions.
- **Firing Assessment** allows to compute the firing conditions on all the data streams reaching the dashboard system. In the case in which, a Firing Condition becomes true, a message event is sent to the Notificator service. Firing Conditions can be estimated:

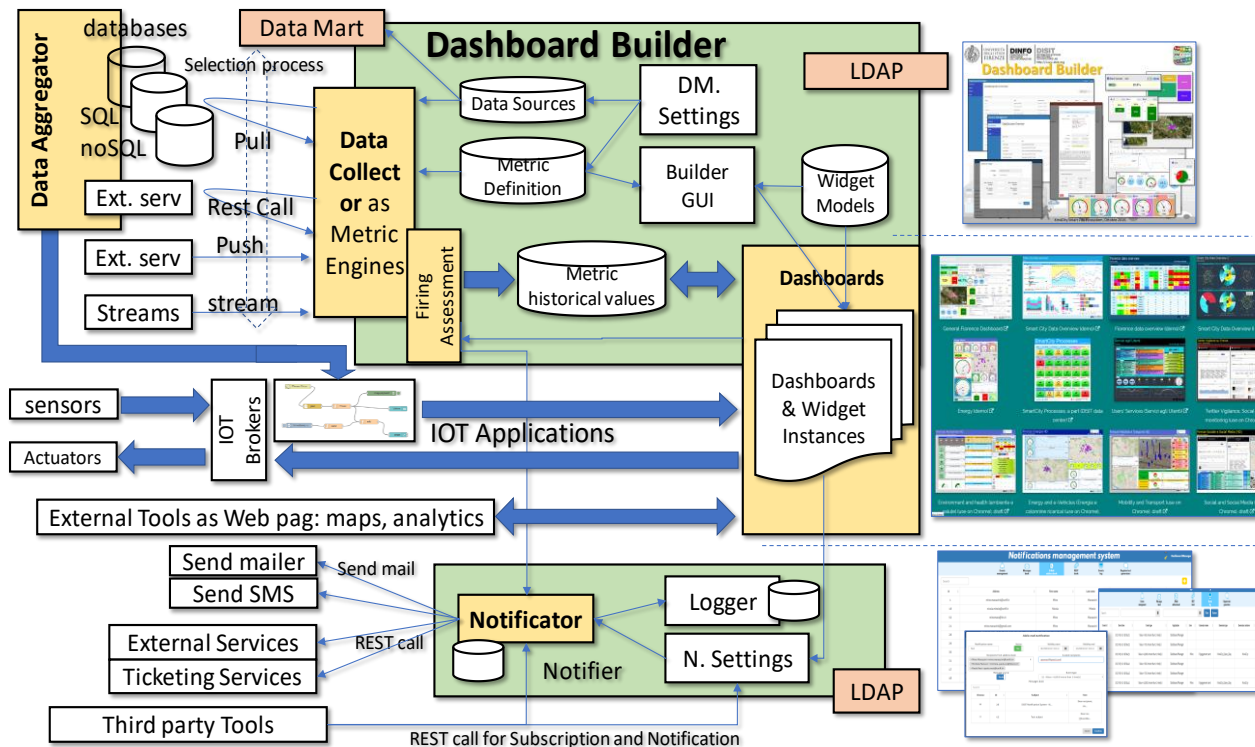


Figure 2: Architecture for Smart City Dashboard Management System integrated with IOT.

from the field and from external services and

- 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.
- **Dashboard Builder** is the core tool for creating dashboards by using a graphic user interface. In the tool the user can set up **Data Sources**: IP address, protocol, user name and password for accessing at each specific the data source. Once the data sources are identified several **Metrics** can be defined. Then new Dashboards can be created taking interested **Metrics** and associating them with one or more **Widget**. 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, as temperature trend in a graphic along the last 24 hours, last week, last month, etc. Therefore, the composition of a **Dashboard consists of** placing and configuring a set of Widgets into the Dashboard frame. The Dashboards are created by using the visual interface of the **Dashboard Builder**.
 - **Integration with IOT**. This feature has been covered by (i) producing special MicroServices as blocks that can be used into IOT applications developed in NodeRED, (ii) connecting NodeRED applications with several IOT brokers, ([33]). Point (i) implied the development of a layer that allowed the traditional Dashboard Widget to be directly connected to (a) IOT Applications by using WebSockets, (b) IOT Brokers (for example, via NGSI, MQTT, etc.). To this end, one of the most suitable IOT Brokers resulted to be the Fi-Ware Orion Broker.

Dashboards are typically adopted for reporting KPI of the city and thus of specific infrastructures and services. This means that specific alerts and notifications have to be activated and managed at level of single Metric. On the other hand, the same Metric can be used on different dashboards and widgets for different purposes. So that, for each Widget of each Dashboard specific alerts and firing conditions can be set up. For example, when Metric M is adopted in Widget W of Dashboard D, the certain criterion C is saved and computed for firing (M, W, D, C). One or more Criteria can be defined (M, W, D, C1 ... Cc), each of them may produce multiple Notifications, N: (M, W, D, C1 (N1, ..., Nn), ..., Cc (...)).

Therefore, the solution has been to design and develop a **Notifier** to

- Accept registrations of possible tuples $\langle m,w,d,c \rangle$, to enable the reception of *Notification Requests*, that

are used to send Notifications according to different approaches;

- Accept registration by third-party tools, in addition to those of the Dashboard Builder, to send alarms about the firing of the registered conditions;
- Produce emails and REST calls, that can be used for calling SMS, as well as for the activation of maintenance ticketing system on OpenMaint tool for example ([32]);
- Log all the registrations and Notification Requests for further analysis and security evidence;
- Maintain and use a list of Notification recipients, 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.

To this end we suggest using specifically development tools, Such as the ServiceMap ([34]) which is used for knowledge base browsing over the city data as Km4City knowledge base, which is RDF store as well, exploiting geospatial reasoning and inference [Bellini et al., 2014]. In addition, the technical browsing on the RDF Graph Store may be needed to discover relationships. To this end, the LOG ([35]) tool for browsing into any RDF store by using SPARQL and visual interface has been developed in the past and now used in this context. This tool allows 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. [23].

As a conclusion, the generated Dashboards are **Dashboard Instances** which 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).

5. Experimental Results and Validation

The solution proposed in this paper has been developed in REPLICATE flagship research and development project SCC1 H2020 of European Commission for Florence City. It has been also used in other large projects such as Sii-Mobility Smart City Nazionale on Mobility and Transport of MIUR, RESOLUTE H2020 project for critical infrastructure and resilience of transport systems, and GHOST MIUR for Cagliari smart city experimentation. The proposed Dashboard solution is strictly connected with a number of tools of the Sii-Mobility/Km4City suite of tools which are used to

perform smart city analytics, semantic browsing, and decision support, etc., such as: ServiceMap ([34]), smartDS, system thinking decision support based on Bayesian models (<http://smartds.disit.org> [21]), Wi-Fi monitoring and predictions, smart parking prediction, traffic flow reconstruction and prediction, energy metering, first aid monitoring, environmental monitoring, social media monitoring and alerting, weather forecast, etc.; most of them based on clustering, machine learning, etc. [4].

In general, the decision makers in the city are politicians, assessors, and director of units. Some of the units have already adopted a high level of technology, for example, the mobility and transport, the civil protection, etc. In other units, the level of control is low since the technical activity is mainly delegated to City Operators such as: energy operators, water management, health care hospitals, environmental agency, waste management, police and alert (112, 118, 911), etc. All of them have their own monitoring system, that is tuned to

operators. The dashboard can organize data according to different views/paradigms: horizontal view (synthetic view reporting many different aspects of the city status) and vertical (or thematic).

5.1 Examples of Horizontal User Oriented

A horizontal dashboard contains a synthetic view reporting many different aspects of the city status, such as:

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

Tourism oriented: a dashboard reporting the major

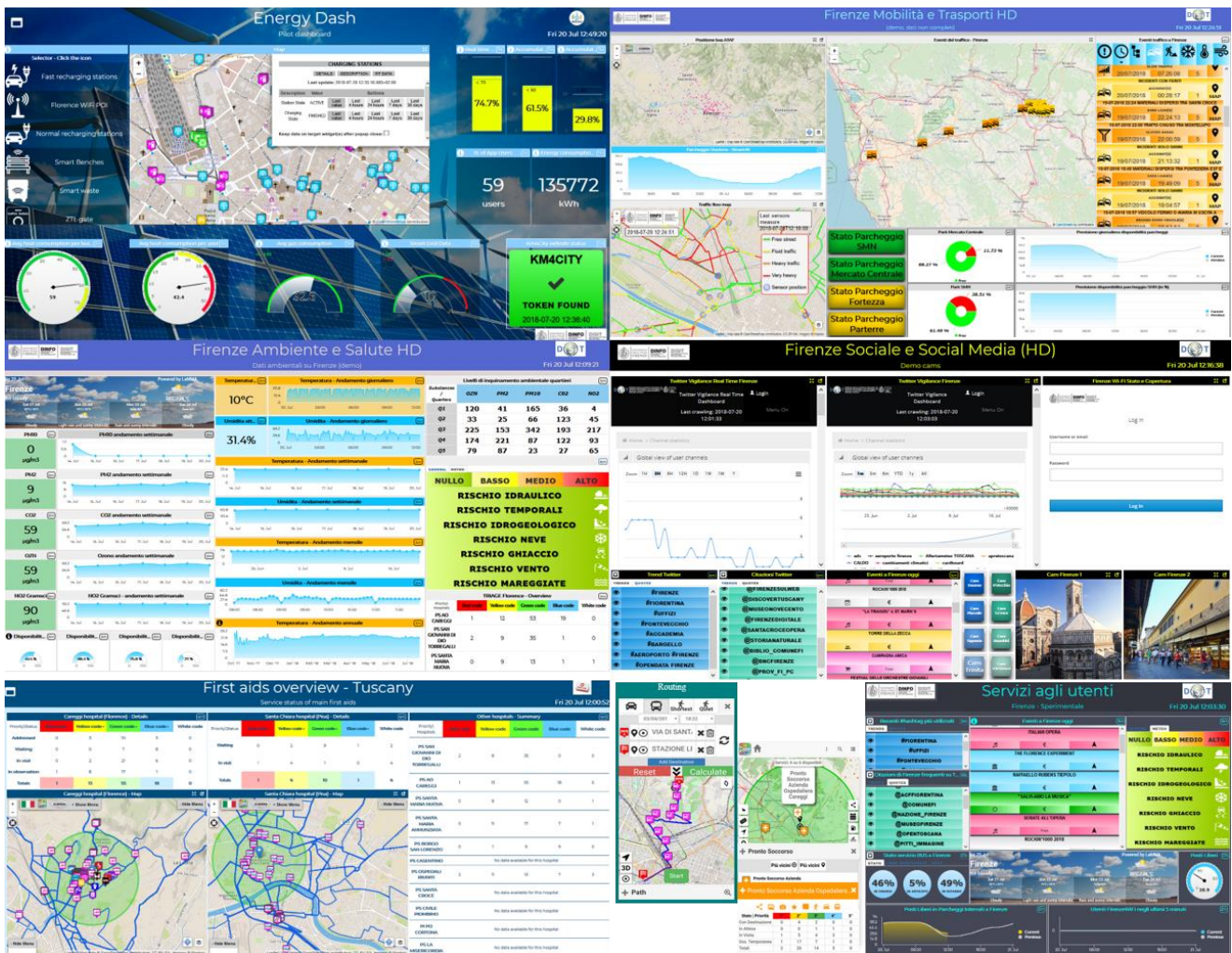


Figure 3: Florence Smart City Dashboards, dashboard reporting first aid status, and a final user dashboard for hotels.

vertical control their own information and status. In some cases, the general information about weather forecasts and status is shared among the different

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, the number of touristic busses arriving in the city and their paths, etc.

5.2 Example of Vertical thematic oriented

A thematic dashboard contains the available data related to a specific context. It can be viewed as an in-deep representation of one of the aspects reported in the horizontal dashboards. Some samples can be:

public transportation: position of busses in real time, number of active busses, average delay at the bus stops, number of active taxis with respect to the plan, number of recharging stations for public vehicles and their status, number of people on busses, percentage of busses with respect to the plan, number of events/incidents on traffic, status of the underpasses, status of the bridges, number of tickets, number of free lots in parking, events in the city, etc.

private mobility: level of traffic flow, traffic flow reconstruction with classic colors (green, yellow, orange and red), number of free slots in parking, number of cycling people on paths, number of vehicles entered into the RTZ per hour, number of vehicles entered in the city per hours, number of trucks on the main streets, number of shared bikes in percentage respect to the total available, events in the city, etc.

Energy: KW/h or GW/h consumed in the last hour for public services, KW/h or MW/h consumed in the last hour for recharge stations, KW/h or MW/h saved by public services since the usage of renewable energy production, KW/h or MW/h saved in the store and available for consumption, number of monitored meters grid, saved energy by following suggestion provided by Apps, Co2 saved by using e-Vehicle, etc.

Environmental data referring to different area of the city: temperature, humidity, PM10, PM2, CO2, wind, pollination, etc.; weather forecast; real time data from environment (temperature, humidity, wind velocity, etc.); level of water in the rivers; level of drinkable water; Tons of collected garbage; Tons of collected garbage differentiated; earthquake monitoring; etc.

Social: status and stream of the social media; the most cited users on Tweet; the most mentioned hashtags on Twitter.com; the sentiment analysis on Tweets connected to the city in the last minutes; number of people moving the main area; number of people arrived by train in the City; TV cameras about the main points of interest in the city; number and list of the major entertainments, political, and sport events in the city; etc.

Security: data also presented on the Social Dashboard describing the presences in the city of people; any kind of event in the city from entertainment, sport, political, religious, critical events on the road, etc.; eventual paths and area of the events; TV cameras observing the areas of the events; number of resources available for controlling the city and their deploy on the city map (cop, ambulances, 118, 911); aspects related to the environmental data; aspects related to mobility for

planning the evacuations; aspects related to the public transportation for eventual change the paths.

Health: data reporting the status of the triage in the major hospitals; position of the emergency stations; number and position of the ambulances; environmental data for hot waves, temperature, etc., which can increase the risk of stroke.

5.3 Validation

As a conclusion, after the production of a set of Dashboards some of them were selected for trial and are reported in the Km4city portal, [11], where most of them are presenting public data that can be rendered on screen. 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. Many examples of dashboards produced by the presented tool can be accessed from the Km4City Portal, [11].

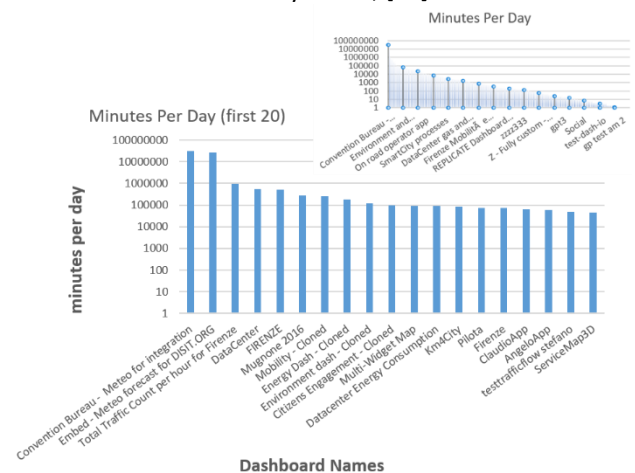


Figure 4: Minutes per Day. All dashboards and only first 20.

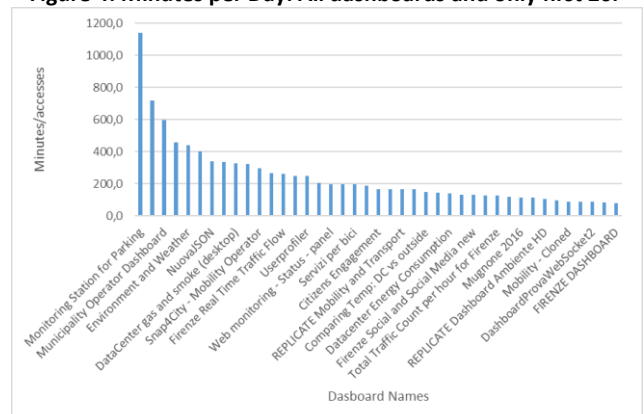


Figure 5: Average number of minutes the people stay connected on dashboard for each access

Actually, 653 Dashboards are present in the system: the total number of minutes of access to the dashboards is 62.410.108 (29.045,28333 total hours) and the total number of clicks on them is 1.742.798, 33% are public

and the 67% are private. Private dashboards are those accessible only by their author while the others are visible to all the web users. In Fig.4 it can be seen how many minutes per day each dashboard is accessed by the users: at the top right you can see the trend of all the dashboards, at the bottom left only the dashboards that have a total greater than minutes of access. In Fig.5, it is possible to see the average number of minutes the people stay connected on dashboard for each access (calculated as {total minutes per day}/{total accesses per day}). This second instagram, is relevant to evaluate which type of dashboard is best suited to stay in a control room means: the dashboards having the highest values are those kept longer on the screen and more comfortable as staying in a control room.

6. Conclusions

Smart cities are becoming more and more advanced, for this reason it is necessary to ingest a multitude of data not only static and periodic but in real time. This amount of data (Big Data) must be analyzed to provide information on the state of the city both to citizens and especially to decision makers. A fundamental aspect is to study and apply ad hoc methodologies to visualize the events in real time, in Smart City Control Room Dashboards. In this work we have analyzed the functional characteristics that these Dashboards must have to better represent the state of the city. The analysis has led us to conclude that there are no solutions on the market that meet all the requirements outlined, so it has led us to develop our own solution which fits the Km4City Big Data architecture.

7. Acknowledgements

Thanks to the European Commission for founding. All slides reporting logo of REPLICATE H2020 are representing tools and research funded by European Commission for the REPLICATE project. REPLICATE has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement n° 691735).

8. References

- [1] M. Azzari, C. Garau, P. Nesi, M. Paolucci, P. Zamperlin, "Smart City Governance Strategies to better move towards a Smart Urbanism", The 18th International Conference on Computational Science and Its Applications (ICCSA 2018), July 2 - 5, 2018 in Melbourne, Australia in collaboration with the Monash University, Australia.
- [2] C. Garau, P. Zamperlin, M. Azzari, P. Nesi, G. Balletto, M. Paolucci, [THE ROLE OF KM4CITY DASHBOARD IN URBAN POLICIES: GOVERNANCE STRATEGIES FOR DYNAMIC URBAN SYSTEMS](#) from 2nd International Conference on Smart and Sustainable Planning for Cities and Regions 2017, Bolzano/Bozen (Italy), 22-24 March 2017.
- [3] Few, Stephen. "Information dashboard design." (2006).
- [4] C. Badii, P. Bellini, D. Cenni, A. Difino, P. Nesi, M. Paolucci, Analysis and Assessment of a Knowledge Based Smart City Architecture Providing Service APIs, Future Generation Computer Systems, Elsevier, 2017.
- [5] IBM Institute for Business Value, "How Smart is your city? Helping cities measure progress", [online]. Available: oct 2013, http://www.ibm.com/smarterplanet/global/files/uk_en_uk_cities_ibm_sp_pov_smartcity.pdf
- [6] Alcatel-Lucent Market and Consumer Insight team, "Getting t about Smart Cities Understanding the market opportunity in the cities of tomorrow", Oct. 2013.
- [7] Anthopoulos L., Fitisilis P. "Exploring architectural and organizational features in smart cities." Advanced Communication Technology (ICACT), 2014 16th Int. Conference on. IEEE, 2014.
- [8] Filippini L., Vitaletti A., Landi G., Memeo V., Laura G.; Pucci P., "Smart City: An Event Driven Architecture for Monitoring Public Spaces with Heterogeneous Sensors," in Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on , vol., no., pp.281-286, 18-25 July 2010.
- [9] Chourabi, Hafedh, et al. "Understanding smart cities: An integrative framework." System Science (HICSS), 2012 45th Hawaii Int. Conference on. IEEE, 2012.
- [10] Replicate Project, <http://www.replicate-project.org>
- [11] Km4City, <https://www.km4city.org>
- [12] P. Bellini, M. Benigni, R. Billero, P. Nesi and N. Rauch, "Km4City Ontology Building vs Data Harvesting and Cleaning for Smart-city Services", International Journal of Visual Language and Computing, Elsevier, 2014.
- [13] SpagoBI, <http://www.spagobi.org/>
- [14] P. Bellini, D. Cenni, M. Marazzini, N. Mitolo, P. Nesi, M. Paolucci, "Smart City Control Room Dashboards Exploiting Big Data Infrastructure", The 24th International DMS Conference on Visualization and Visual Languages, DMSVIVA 2018, Hotel Pullman, Redwood City, San Francisco Bay, California, USA, June 29 - 30, 2018.
- [15] Tableau, <https://www.tableau.com>
- [16] OpenDataSoft, <https://www.opendatasoft.com>
- [17] GDPR, General Data Protection Regulation, <https://www.eugdpr.org>
- [18] Suakanto, Sinung, Suhono H. Supangkat, and Roberd Saragih. "Smart city dashboard for integrating various data of sensor networks." ICT for Smart Society (ICISS), 2013 International Conference on. IEEE, 2013.
- [19] McArdle, Gavin, and Rob Kitchin. "The Dublin Dashboard: Design and development of a real-time analytical urban dashboard." (2016): 19-25.
- [20] De Marco, Alberto, Giulio Mangano, and Giovanni Zenezini. "Digital Dashboards for Smart City Governance: A Case Project to Develop an Urban Safety Indicator Model." Journal of Computer and Communications 3.5 (2015): 144-152.
- [21] E. Bellini, P. Nesi, G. Pantaleo, A. Venturi, "Functional Resonance Analysis Method based Decision Support tool for Urban Transport System Resilience Management", IEEE Int. Smart Cities Conference (ISC2), 12-15 Sept. 2016, Italy.
- [22] Bellini P., DiClaudio M., Nesi P., Rauch N., "Tassonomy and Review of Big Data Solutions Navigation", as Chapter 2 in "Big Data Computing", Ed. Rajendra Akerkar, Western Norway Research Institute, Norway, Chapman and Hall/CRC press, ISBN 978-1-46-657837-1, eBook: 978-1-46-657838-8, july 2013, pp.57-101, DOI: 10.1201/b16014-4.
- [23] Bellini P., Nesi P., Venturi A., "Linked Open Graph: browsing multiple SPARQL entry points to build your own LOD views", <http://log.disit.org> International Journal of Visual Language and Computing, Elsevier, 2014.
- [24] Elasticsearch, <https://www.elastic.co/products/elasticsearch>
- [25] Apache Solr, <http://lucene.apache.org/solr>
- [26] Apache Banana, <https://github.com/Lucidworks/banana/>
- [27] HUE, <http://gethue.com/overview>
- [28] HBase, Apache HBase, <https://hbase.apache.org/>
- [29] MongoDB, <https://www.mongodb.com>
- [30] RDF <https://www.w3.org/RDF/>
- [31] SPARQL: <https://www.w3.org/TR/rdf-sparql-query>
- [32] Openmaint: <http://www.openmaint.org/en/project/how-it-works>
- [33] Node-Red, <https://nodered.org>
- [34] ServiceMap, <http://servicemap.disit.org>
- [35] LOG, <http://log.disit.org>