decode

Data analysis methods and first results from pilots

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 732546

Project no. 732546

DECODE

DEcentralised Citizens Owned Data Ecosystem

D5.3 Data analysis methods and first results from pilots

Version Number: V1.0

Lead beneficiary: Eurecat

Due Date: March 2018

Author(s): Mirko Marras, Matteo Manca, Ludovico Boratto, David Laniado (EURECAT)

Editors and reviewers: Francesca Bria, Oleguer Sagarra, Javier Rodriguez, Pau Balcells (IMI), Job Spierings, Tom Demeyer (Waag Society)

Dissemination level:		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Approved by: Francesca Bria (Chief Technology and Digital Innovation Officer, Barcelona City Hall)
Date: 03/04/2018

This report is currently awaiting approval from the EC and cannot be not considered to be a final

Table of Contents

Abbreviations

ASIA Application of Integrated Attention Systems / Aplicatiu de Sistemes Integrats d'Atenció
BCN Barcelona
CityOS Operating System of the City
DTW Dynamic Time Warping
ICT Information and Communication Technology
IMI Municipal Institute of Information Technology / Institut Municipal d'Informatica
IoT Internet of Things
IRIS Incidents, Claims and Suggestions / Incidencies Reclamacions i Suggeriments
LBK LB Keogh
MVP Minimum Viable Product
ODI Open Data Infrastructure
TW ThoughtWorks
WP Work Package

1 Introduction

Giving back to people the ownership of their data has become a strategic aspect for cities and citizens in order to overcome the current model of data extractivism[1]. The problem with this model of data silos and data centralisation in the hands of a few corporations is not only that citizens' data are often used for goals they are not aware of. We are also missing opportunities for leveraging such data for public good and individual or collective goals which are not of primary interest for the companies which manage the data. Furthermore, even when open data are available, they often remain just raw data, which citizens without technical skills cannot benefit from.

In response to these problems, DECODE's vision of data as a common good implies a democratization of data processing and exposure based on transparent algorithms and intuitive interfaces for information visualization and exploration. With this in mind, the DECODE infrastructure will be deployed through pilots in Amsterdam and Barcelona described in more detail in D1.1[2]. They aim to demonstrate how the use of technology can allow citizens to effectively and responsibly control the use of their data. Applied in real-world scenarios, these pilots serve to exemplify how data can be managed in a decentralised way, shared and used following a different paradigm than the current one. The general considerations related to designing privacy interfaces and user experience are detailed in D4.7[3].

Barcelona Now, connected to the IDigital Decidim and CitizenSense pilots, aims to make data not only available to and responsibly shareable by citizens, but also useful to them, closing the cycle by providing a visual environment populated by interactive interfaces for data exploration and visualization. Furthermore, the collected data will be exposed in a unified format through API, so that they can easily be accessed by external services or connected to other applications. This means that external developers and entrepreneurs should be able to develop distributed data-driven applications based on decode specifications that are privacy enhancing and rights preserving. Even though the preliminary prototype is based on public data from Barcelona, illustrated in D5.1[4], its modularity and flexibility make it easily deployable with data from Amsterdam or other cities. The overall aim is to demontrate the innovative concept of data commons thorugh real world pilots, by providing tools that let the citizen control who can access their personal data and under what circumstances, ensuring data entitlements by enabling adaptive, smart, context-dependent access

[1] Francesca Bria "People shuld control their digital identity": https://www.citymetric.com/horizons/people-should-control-their-digital-identity-barcelona-s-chief-technology-officer-decode

[2] See "D1.1 Pilot Scenarios and Requirements | DECODE." 30 Sep. 2017,

[3] See "D4.7 Privacy Interface Guidelines | DECODE." 30 Dec. 2017

[4] See "D5.1 Barcelona Open Data, Sentilo and IRIS API available." 31 Dec. 2017

rules.

In these first steps, while individual and colelctive citizens' data shared via the DECODE infrastructure[5] are not yet available, we have started to design and implement a prototype version of the environment that gives unified access to open data from the city and use them to populate a set of basic interactive visualizations. Then, we have started to define a roadmap for the improvement of the environment and the integration with other pilots.

This deliverable comprises the following contributions:

- Build an easy-to-deploy framework capable of collecting, processing, exposing and visualizing heterogeneous data. The code is released under open source license[6].
- Develop a front-end interface that enables citizens to explore data, arrange and share visualizations, and compare information coming from different data sources.
- Implement a prototype version of the environment operating on data coming from public repositories and systems in Barcelona. The live demo is accessible online[7].
- Experiment with unsupervised data mining methods to analyze the collected data, extract knowledge from them, identify patterns and get insights about relevant city issues.
- Define a development roadmap for the improvement of the first prototype version and the integration of data from IDigital Decidim and CitizenSense pilots to leverage DECODE entitlement capabilities and DECODE framework for building "data commons" datasets.

An academic article describing the first version of this prototype has been accepted for publication in the proceedings of The Web Conference, former WWW conference, and will be presented on April 23-27, 2018 in Lyon, France (Marras et al., 2018).

In the DECODE project, we follow a Lean&Agile methodology outlined in detail in D1.7[8]. Based on this, Barcelona Now and the aspects described in this document are to be considered part of an ongoing experiment, and they can change according to pilot development.

This deliverable is structured as follows. First, Section 2 briefly presents the existing urban data exploration tools that relate to our contribution. Then, Section 3 describes Barcelona Now's environment and Section 4 showcases its capabilities. Finally, Section 5 depicts the roadmap and Section 6 discusses conclusions and future directions.

[5] See "D1.4 First version of DECODE Architecture Chapter 2.7.5 | DECODE " 31 Oct, 2017.
[6] https://github.com/DECODEproject/bcnnow
[7] http://bcnnow.decodeproject.eu
[8] See "D1.7 Project methodology and policy review | DECODE." 31 Aug. 2017,

2 Existing Tools for Urban Data Exploration

The proliferation of massively interconnected devices and sensors around citizens raises big challenges both regarding the infrastructures that manage the complexity of the generated data and the visual interfaces that leverage such data to support users in decisions and actions (Bibri & Krogstie, 2017).

Over the years, the increasing amount and variety of open datasets available in raw and heterogeneous formats (e.g. plain text, JSON, RDF, CSV, API) have mainly fostered analysis by data scientists. Tailored architectures and urban dashboards (Aguilera et al., 2016; Cheng et al. 2015; Chilipirea et al., 2017, Zdraveski et al., 2017) are just recently emerging to unlock this hidden potential and help to better understand city dynamics, including noise pollution (Navarro et al., 2017), mobility patterns (Kaltenbrunner et al., 2010, Gong et al. 2017)and electricity usage (Massana et al., 2017).

Both academia and companies have investigated different ways to operationalize city data on urban dashboards, but only available from public datasets (Kitchin, 2014). We briefly discuss existing implementations and how they relate to our proposal.

Madrid Dashboard[9] aims at experimenting with smart city services and fine-tuning them before deployment at large scale. The platform currently offers two services, namely people flow monitoring based on wireless tracking and environmental monitoring. For instance, the authors stated that such services can be applied to dynamically control the university heating, ventilation, and air conditioning systems.

UK Dashboard[10] summarizes quantitative data about the major United Kingdom cities on a single screen. It primarily shows weather, environmental, transportation, and energy demand with color-coded numerical values. Even though it provides a bird-eye city view, the representation of data is not translated from raw numerical values to a format that is easier to be digested by non-technical users.

Edmonton Dashboard[11] enables users to visualize pre-defined snapshots of city data together with descriptive indicators under consistent color-coding, iconography and fonts. The dashboard allows deeper historical analysis with interactive tools to filter data.

Amsterdam Dashboard[12] enables professionals to visualize data on a simple map view, displaying points representing discrete information, and geographical partition views, where each partition displays a certain category on which city elements are projected.

[9] http://ceiboard.dit.upm.es/dashboard
[10] http://citydashboard.org/choose.php
[11] https://dashboard.edmonton.ca/
[12] https://data.amsterdam.nl

Dublin Dashboard[13] integrates data from the city council, the government departments and several existing smart city and social applications. It contains hundreds of data representations grouped in different modules, including overall statistics from the city and information from key points.

Boston Dashboard[14] helps users to monitor the city with a set of baseline metrics and plans on several areas, including education, infrastructure, and housing. The platform does not have a real user dashboard, but provides a smart tool to enable professionals to develop their visualizations from a limited set of types, requiring technical operations.

CityPulse[15] is an open source framework to provide access to historical data and events that the user can visualize across multiple contexts. The ability to store cleaned and summarized data for post-processing makes it possible to analyse the status of the city not only on the go but also at any point in time, enabling diagnosing of any incidents or relevant situation that might have occurred.

The existing dashboards include limited pre-defined use cases which can only address the most typical needs without personalization capabilities. This is one of the reasons the impact of open data on public good is still limited (Gharaibeh et al., 2017). On the other hand, customizing applications that can support end-users' goals requires expertise not expected of most of them. Offering ad-hoc interfaces through which non-expert users can interactively personalize applications has shown promising results to reduce the mentioned gap (Paternò & Wulf, 2017). By applying this idea in the urban domain within the DECODE project, we aim to empower curious citizens with little or no technical expertise to get personalized insightful knowledge from urban data, either publicly available from existing open datasets or responsibly shared from DECODE.

With respect to the mentioned solutions, our proposal has several advantages in terms of openness, flexibility, usability and personalization, so it can be used in a wider range of scenarios and use cases in real-world settings.

[13] http://www.dublindashboard.ie
[14] https://boston.opendatasoft.com
[15] https://github.com/CityPulse/CityPulse-City-Dashboard

3 Reference Architecture

The environment is composed of a back-end subsystem acting as data aggregator and manipulator, accompanied by a web-based front-end subsystem. The front-end enables citizens to leverage the data provided by the back-end to create interactive visualizations and personalized dashboards. The figure below depicts the architecture.

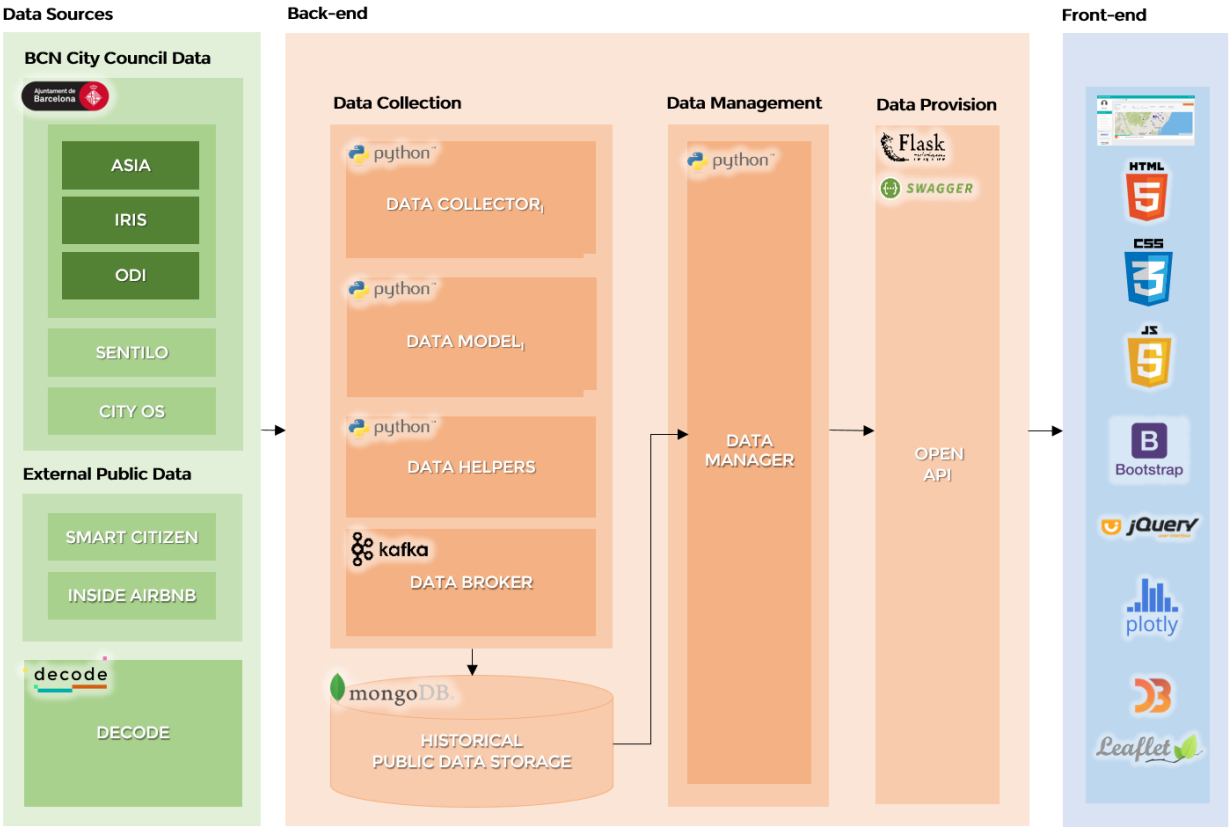

Figure 1. Reference architecture of Barcelona Now.

3.1 Backend

3.1.1 Data Modeling

The integration of data from different sources with heterogeneous access methods and formats requires a pre-processing step in order to provide a unified view of them. This is performed by taking data structured under the original data source schema and transforming it into data structured under an internal data schema, so that the target data is a standardized internal representation of the source data.

To this end, we have selected JSON[16] as standard storing format due to its flexibility. All the geographic coordinates have been converted to the WGS84 EPSG:4326[17] standard, while the datetime fields have been converted to the ISO8601[18] format. For each data source, attributes' names have been translated in English. Finally, we have designed a specific and unified record structure, so that it will be easier accessing and filtering heterogeneous data source via unified temporal and geographic queries, while a payload field stores all the information that is specific of that data source.

The record structure that each collector is responsible to provide is defined as follows:

Figure 2. Base record structure.

The base record has two composite fields: *location*, which includes all the geographic information associated to the record, and *payload*, which contains all the fields which are typical of that specific data source. Specifically, the location *field* part of the base record is defined as follows:

Figure 3. Location field structure.

[16] http://json.org/
[17] http://spatialreference.org/ref/epsg/4326/
[18] https://www.iso.org/iso-8601-date-and-time-format.html

The payload field structure depends on the specific data source that we considered, so we have a different structure for each data source.

Figure 4. Payload field structures.

3.1.2 Data Collection

The collectors folder is organized in a way that the collectors which use pull technology and the collectors which use push technology are subdivided in two different folders.

Each collector has its folder with three files: a configuration file, a payload definition file, and an execution file. The first one includes a Python class that defines a JSON record with all the configuration parameters needed by the collector. The second one contains a Python class with all the record attributes which are specific of that kind of data source, getter and setter methods, and a method which returns a JSON record containing all the attributes. The third one contains a Python class with four methods: *start()* is the main method whose purpose is to loop for all the files to be accessed or all the API calls to be executed. For each file or API access, it calls the *sendRequest()* which takes a URL as parameter and returns the data retrieved from that URL. According to the access method, *sendRequest()* is able to get data with different formats (e.g. API, XML, CSV, JSON) by using the proper Python package. Once the data is retrieved, *start()* calls *saveData()*. This method has the purpose of accessing each record represented in the original format, calling *buildRecord()* to create an instance of the record coded in the internal unified format for that data source and using *store()* from *StorageHelper* to save the record into MongoDB. Sending records to Kafka is also possible just by updating the corresponding JSON attribute of the main configuration file of the project (i.e., bcnnow-master/config/Config.py).

The execution file of each collector should be inserted into the cron job whenever a collector requires to be periodically executed to get data from the original source. The collector process used for each public dataset is detailed in D5.1[19].

3.1.3 Data Storage

The public data records are stored into MongoDB. More precisely, a MongoDB database called "decode" includes a collection for each data source whose collector has been developed and executed by the cron job. At the moment, we have the following seven collections, each one of them with the following cardinality:

- ASIA: 8,959 records.
- Bicing: 5,914,262 records.
- City Equipment Items: 35,494 records.
- Inside Airbnb: 17,953 records.
- IRIS: 530,439 records.
- City Points of Interest: 525 records.
- Sentilo's Sensors: 1,029,289 records.
- SmartCitizen Sensor's: 17,708,383 records.

The above-mentioned collections are continuously updated. The communication between each Python collector and MongoDB or each Python collector and Kafka is carried out by using pymongo or kafka-python respectively, two Python packages whose purpose is to provide a standard interface to access MongoDB and Kafka from Python code. For the attribute "coordinates", part of the field "location", we created a 2dsphere[20] index that calculates geometries on an earth-like sphere and supports all geospatial queries for inclusion, intersection and proximity. An index has been also created for the attribute "timestamp". These indexes have been created for each collection.

3.1.4 Data Exposure

Until this development step the APIs are privately deployed. They have a unique method which allows developers to search for available records by passing in appropriate parameters. These parameters are specified as follows:

- *source* string (query): name of the data source where the records must be searched;
- *filter* string (query): comma-separated attribute-condition pairs;
- *field* string (query): comma-separated attributes to be returned;
- *group* string (query): comma-separated attributes to group by;
- *aggregator* string (query): comma-separated function-attribute pairs;
- *sort* string (query): comma-separated ordering-attribute pairs;

[19] See "D5.1 Barcelona Open Data, Sentilo and IRIS API available, Chapter 4 | DECODE" Dec. 2017

[20] https://docs.mongodb.com/manual/core/2dsphere/

- *skip* integer (query) number of records to skip for pagination;
- *limit* integer (query) maximum number of records to return.

The APIs are currently deployed by a Flask script which receives the parameters, validates them, prepares a query to be done on MongoDB according to MongoDB's predefined syntax, and arranges the results in a JSON record to be returned.

The public documentation and the public deployment of the APIs are under definition on Swagger[21] and will be completed for the next deliverable, including more API methods. The public APIs can be leveraged for hackathons in the future .

3.1.5 How It Works

The backend aggregator and manipulator prototype is mostly written in Python. It can be deployed inhouse or in the cloud (e.g., Amazon Web Services[22]). It collects streaming and non-streaming data coming from heterogeneous data sources and with different formats (e.g., JSON, CSV, RDF, API), translates such data to an internal format based on international standards (WGS84 for geographic coordinates, ISO8601 format for timestamps), stores them into a database and provides access to them through API.

Each data source can be integrated into the back-end subsystem with the creation of its own data collector using either pull technology (i.e., collectors periodically request data to the data sources) or push technology (i.e., data sources send data to collectors whenever available). Leveraging pre-defined templates of collectors, developers can easily integrate a new data source by selecting the appropriate template, modifying it to set the mapping between the original data structure and the internal common data structure and adding it to the cron jobs list. The current version of all the collectors is available here[23] into the code repository stored in Github.

Once the collector accesses the original data through the *sendRequest()* method, the received records are translated from the original schema to the common schema shown in Figure 2 through the *buildRecord()* method. Proper helpers (i.e., *GeneralHelper*, *LocationHelper*, *TimeHelper*, and *StorageHelper*) provide functions to easily support the conversion.

Subsequently, since the back-end subsystem aims to provide historical access to data and is expected to receive geo-temporal queries, each translated record is stored into an instance of MongoDB. It natively supports large datasets and geo-operations on them. The storage task is carried out by each collector through the *saveData()* method which calls the pre-defined method *store()* of the *StorageHelper* class. On MongoDB, one table is created for each data source. In order to allow efficient search within

[21] https://app.swaggerhub.com/apis/bcnnow/bcnnow/0.0.1
[22] https://aws.amazon.com
[23] https://github.com/DECODEproject/bcnnow/tree/master/apps/backend/data/collectors

geographic areas or time intervals, proper database indexes are created on location coordinates and timestamps for each collection.

Finally, the data are made externally available via API using Flask. The Flask scripts simply receive the query parameters (i.e., data source name, filtering conditions, record fields to be returned, grouping conditions and aggregated functions to be computed on groups, and sorting attributes) and perform the corresponding query on MongoDB. The flexibility and control given by Flask makes it easier to create a unique point for providing access to public data, but also to private data which require to be accessed on-the-fly (e.g., from DECODE).

The back-end components and processes enable the simple integration of new data producers by just developing and inserting their own collector in the collectors' list and data consumers (e.g., web/mobile applications) by leveraging the provided public API.

3.2 Frontend

The front-end dashboard prototype consists of a web application implemented via HTML and JavaScript with the support of jQuery[24] and jQueryUI[25] libraries. The Bootstrap[26] library is used to build responsive layouts and arrange widgets in grids. The interactive visualization of data in charts and maps can be handled by D3.js[27], Plotly[28], and Leaflet[29] libraries. The communication between the front-end and back-end subsystems is built on AJAX and uses JSON for transmitting information in both directions. More precisely, the frontend Javascript scripts calls backend APIs with different parameters based on the type of data required by a given widget. The technologies we used are compatible with existing browsers, ensuring platform-independence.

The primary components provided by the subsystem are a visual interface to create and explore interactive visualizations and a set of functionalities to enable citizens to organize such visualizations in different dashboards in accordance with their needs and goals. Figure 3 showcases a dashboard. On the left sidebar, the interface gives access to the widget creation tool and the dashboards previously created.

[24] https://jquery.com/
[25] https://jqueryui.com/
[26] https://getbootstrap.com/
[27] https://d3js.org/
[28] https://plot.ly/
[29] http://leafletjs.com/

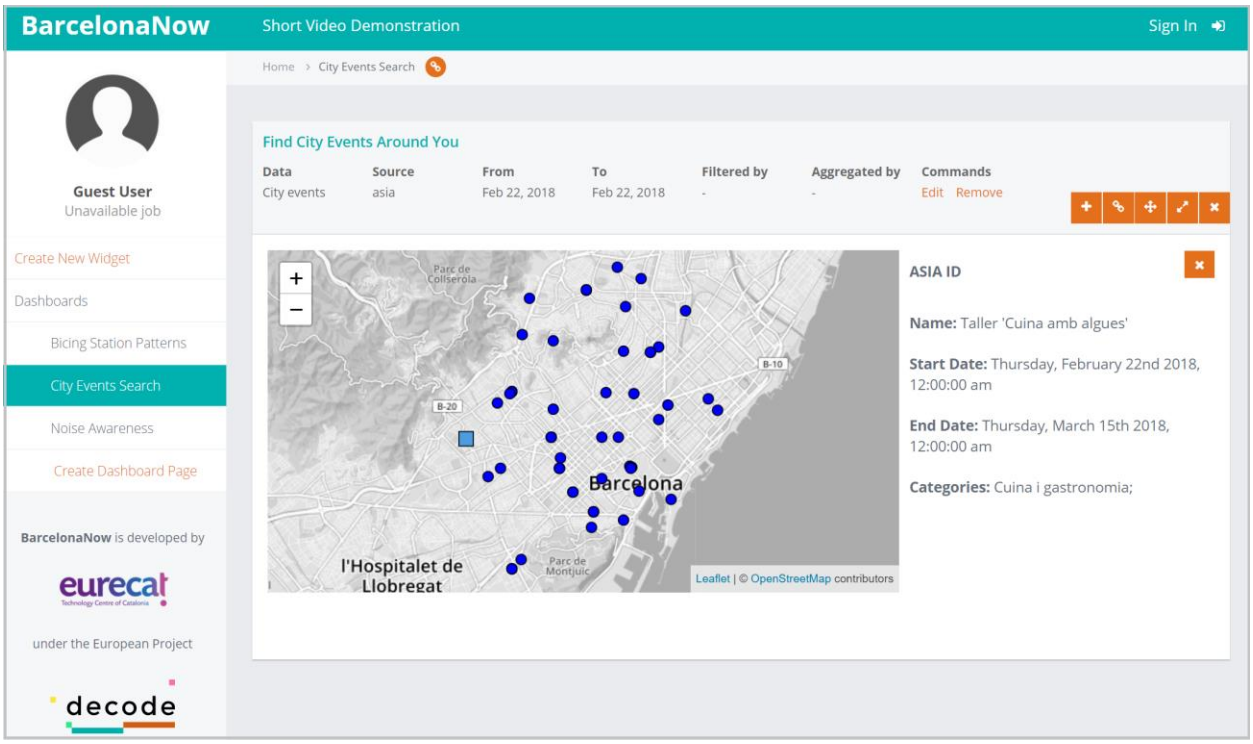

Figure 5. Sample dashboard overview.

The creation tool allows users to interactively compose and manage widgets with visualizations fed by a subset of the data provided by the back-end subsystem. Each widget indicates a user-defined title and the last modification time. Furthermore, for each included data source, the widget lists the name together with the set of user-selected parameters. Interacting with the icons on the widget header, the user can insert one or more data sources to be visualized inside it, and eventually specify additional parameters, such as the time range, the time granularity (e.g., daily, weekly, monthly, yearly), the visualization type (e.g., markers, heatmap), the geographical aggregation (e.g. census section, neighborhood, district), and filtering keywords (Fig. 4). Large part of these filtering options has been already developed. The other options will be available and extended in next steps.

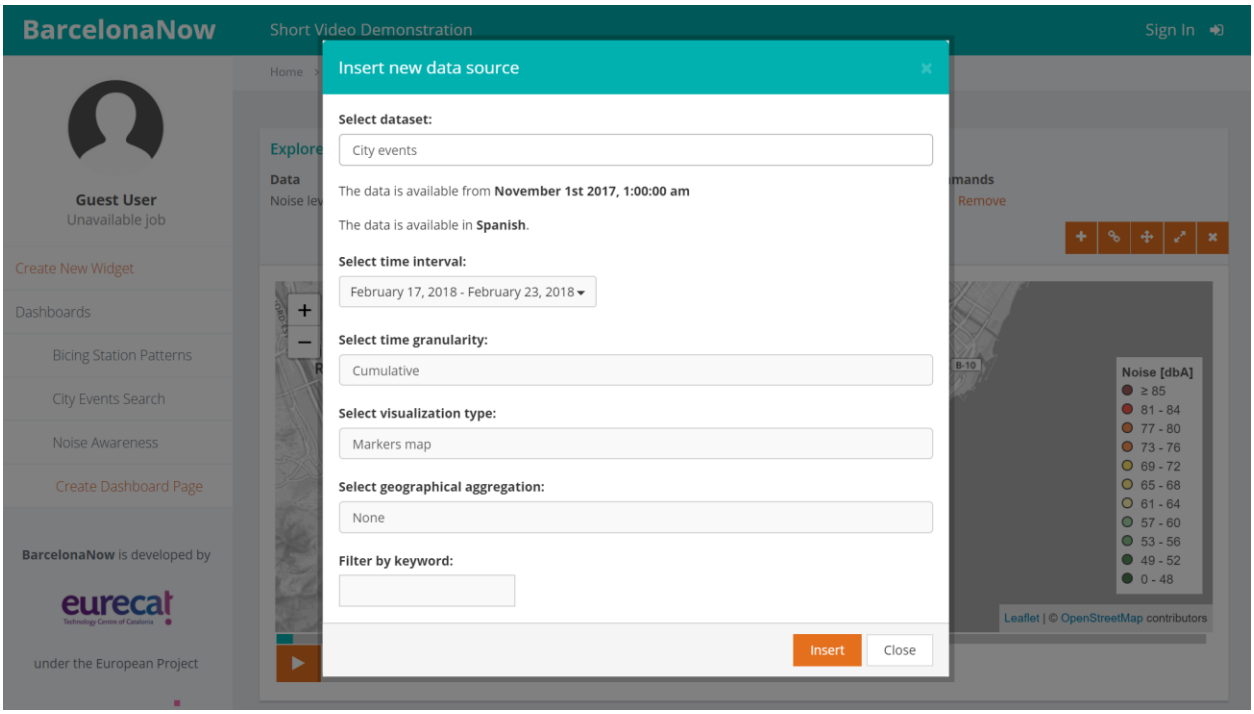

Figure 6. The interface to insert a new data source.

On the center, the preview widget panel depicts how the visualization appears with the current settings. The legend shows the meaning of different colors. In addition to this, users can inspect and personalize the detailed views regarding the individual elements depicted on the visualization. For instance, markers-based visualizations provide functionalities to select markers and interact with a panel providing further exploration.

The environment enables users to define one or more dashboards where they can group the widgets to be monitored on a single screen based on different user-defined thematics, goals, and needs. Each widget can be directly moved between different dashboards by drag and drop and shared via short custom links to allow other users to view and reuse them. The whole dashboard can be shared as well. The sharing methodology includes a set of actions and parameters to reproduce the visualizations without sharing the data itself. This could also support more advanced scenarios where the environment is deployed in a setting involving individual users with different access permission to one or more subsets of the same data source. By clicking on the link, the individual widget or the whole dashboard is reproduced, so that the current user can explore and modify it. We envision that each user can have a personal repository where their widgets are saved to be accessed in future sessions.

3.3 Deployment

This section report the python packages required for running the environment, and the main shell commands to setup the backend and the frontend subsystems.

3.3.1 Python Requirements

When installing software, and Python packages in particular, it's common to get a lot of libraries installed. Each of these packages has its own version. Requirements files give a way to create an environment: a set of packages that work together. The following list provides the Python packages required by the current version of BarcelonaNow to be fully working.

Data Modeling
bson==0.5.0
xmltodict==0.11.0
pandas==0.21.0
python-dateutil==2.6.1
DateTime==4.2

Data Collection and Pre-Processing
numpy==1.13.3
pyproj==1.9.5.1
requests==2.18.4
Shapely==1.6.2.
urllib3==1.22

Data Storage
pymongo==3.5.1
kafka-python==1.3.5

Data Exposure
Flask==0.12.2
Flask-Compress==1.4.0
Flask-Cors==3.0.3
Flask-RESTful==0.3.6.

3.3.2 Backend Setup

To run the backend, a list of technologies must be installed on the server where the environment will run. BarcelonaNow currently requires Python 3.5 and MongoDB. The following list provides the set of the base commands required to setup the backend.

Install Python (>=3.5)

$ sudo apt-get install python3.5

Install and run MongoDB
$ sudo apt-get install -y mongodb-org
$ sudo service mongod start

Clone the repository
$ git clone https://github.com/DECODEproject/bcnnow.git

Install all the Python requirements
$ pip install -r bcnnow-master/requirements.txt

Run BarcelonaNow API
$ nohup python3 bcnnow-master/backend/logic/api/v0/app.py &

In the case developers choose to use Kafka to store records, it should be firstly installed.

3.3.3 Frontend Setup

To run the frontend, a list of technologies must be installed on the server where the environment will run. In fact, the backend must be running in order for the frontend to run. Since the current version of the frontend is a simple one-page dashboard which integrates a set of Javascript files to manage the provided functionalities, we just need to make publicly accessible the folder bcnnow-master/apps/frontend containing the one-page dashboard (index.html) and the required scripts. The following list provides the commands required to setup environment frontend.

Copy the frontend folder into a publicly-accessible folder
e.g., $ cp 'bcnnow-master/apps/frontend '/var/www/html/'

Open your browser at the link pointing to the publicly-accessible folder
e.g., http://localhost:80

4 Use Cases

The capabilities of the environment will be demonstrated in this preliminary step through its deployment with open data related to the city of Barcelona, from different sources, in the context of use cases connected to individual and collective needs. For the definition of the use cases we built on personas defined in D1.1. After describing the included data sources, we will show some sample functionalities of the environment for visualizing data from a single source and combining data from multiple data sources as multiple layers within maps.

4.1 Current Data Sources

More detailed information about the data sources is reported in D5.1. Here we summarize basic information about each source included, distinguishing official from third party data sources. The Barcelona's City Council data sources are:

- **ODI (Open Data Infrastructure)**[30]: it is a freely and openly-accessible data portal of the city, which contains data related to public services. Currently, data from the Bicing bicycle sharing system, the list of points of interest of the city and the list of different types of equipment have been integrated. *API. JSON, RDF*.
- **Sentilo**[31]: IoT sensor platform of the city. Currently a closed platform, accessible only to partners, which also has a public area. The platform allows to connect and consume sensors in a federated way. It contains mainly sensors from city council services and pilot projects that have been connected as proof. *API. JSON*.
- **IRIS (Incidències, Reclamacions i Suggeriments)**[32]: it is the "citizen relation system" of the city. An infrastructure that handles all the citizen interactions with the city via phone, person, email or other online means. It stores citizens demands and requirements, which are classified according to a predefined ontology and sent to its allocated supervisors. The city council has then roughly 30 days to attend to those petitions, some of which are tagged with geolocated information. *FILE. JSON*.
- **ASIA (Aplicatiu de Sistemes Integrats d'Atenció)**[33]: it is the system in charge of city equipment. This system fulfills two functions. One is to keep an up to date record of all city buildings and their usage, while the second is to keep a day to day agenda of all public activities held on public buildings. *FILE. XML*.

[30] http://opendata-ajuntament.barcelona.cat/
[31] http://www.sentilo.io/
[32] http://opendata-ajuntament.barcelona.cat/data/en/dataset/iris
[33] http://opendata-ajuntament.barcelona.cat/data/en/dataset/agenda-mensual

- **CityOS (City Operating System)**[34]: internal data lake of the city council, currently under final stages of development, scheduled to start working by January 2018. Currently, initial use cases are being considered for the platform, but eventually the data from ODI, Sentilo, IRIS, ASIA and many other city council sources will be connected to the infrastructure. *ACCESS NOT AVAILABLE*. *FORMAT NOT AVAILABLE*.

Other public data sources are:

- **Smart Citizen**[35]: it is a platform to generate participatory processes of people in the cities. Connecting data, people and knowledge, the objective of the platform is to build productive open indicators and distributed tools. *API. JSON*.
- **Inside Airbnb**[36]: it is an independent, non-commercial set of tools and data that allows users to explore how Airbnb is being used in cities. It is a mission-driven activist project which provides data and tools for an informed debate on the impact of Airbnb on residential communities, also for Barcelona. *FILE. CSV*.

4.2 Single-Level Visualization Design

As a first example, we started from the interviews and persona definitions from D1.1, in which noise emerged as an important issue that concerns citizens in Barcelona. It even motivated a group of citizen from the neighbourhood of Gracia to install sensors to measure the acoustic levels in their area.

Indeed, living in a large city like Barcelona means becoming acquainted with noise generated by transport, industry, people, social events, construction activities and so on. However, such noise is not equally distributed across neighbourhoods. Therefore, given the importance of noise as a source of information about citizens' and city quality life, we have leveraged the Barcelona Now dashboard to design a first visualisation showing a time-evolving map of environmental acoustic levels based on data provided by outdoor sensors spread all over the city. The measurements have been collected from Sentilo through Smart Citizen, a platform aimed at building open indicators and collaboratively connecting data, people and knowledge.

Therefore, beginning with a single data source (50 noise sensors, data updated every minute, historical data of 10 months, about 17 millions of observations in total), we demonstrate the dashboard capability of creating a new data visualization showing a time-evolving map of acoustic levels.

[34] http://ajuntament.barcelona.cat/digital/en/digital-transformation/city-data-commons/cityos
[35] https://smartcitizen.me
[36] http://insideairbnb.com/

Figure 7. Interactive visualization of noise temporal patterns based on 50 sensors. The right part of the widget shows the time series for the two sensors selected on the left (shown as squares).

The visual interface exposes temporal patterns over the acoustic level measured in A-weighted decibels (dBA). Hotter colors represent higher levels of noise, while colder colors depict more peaceful levels. Playing with the controls to start and stop the data animation over a range of time, it is possible to recognize both daily and weekly patterns characterising different neighbourhoods, and to identify, for example, the areas of the city that are more affected by noise pollution during the night or the weekend. By clicking on one or more sensors in the map, it is possible to activate a second panel next to the map, showing the temporal patterns of the selected sensors.

We now consider Carme, the proto-persona defined in D1.1, a woman with no special technical skills, who is concerned by the increase in noise in her area during night. Opening BarcelonaNow and selecting to view noise data on the map, Carme can focus on acoustic levels at 1am in the night, and get an overview of night noise levels across the different areas of the city. She can then select a noise sensor close to her place and compare it to other sensors, to compare the daily cycle of noise in her area with other areas.

As shown in Figure 7, the tool allows for comparison of temporal patterns for the two sensors selected, shown as squares on the map. In the case shown, two sensors in two peripheral areas of the city have been selected. The sensors exhibit a very similar trend in the selected period: a typical circadian pattern with low levels of noise during night, with a peak in both cases around 8am; however, a noticeable difference between the two sensors emerges as the second one, shown in blue, reaches more silent levels of noise during night.

In this way, Carme can not only check whether her area is more noisy than others on a specific moment in time, but also whether it tends to be more noisy during the night than during the day, with respect to a comparable area.

Other analogous visualizations can be created with any kind of data coming with a location and a timestamp, such as events in the city from ASIA, or the density of bikes into the bike-sharing stations from ODI. A citizen could for example see whether a certain area is more or less likely than another to have available bike sharing bicycles in the morning between 8am and 9am, the time in which many people take a bike from the service for going to work.

4.3 Multi-Level Visualization Design

While many questions and needs can be answered by just inspecting one source of data, like shown in the examples in the previous section, in some cases in order to get helpful knowledge and insights from the data it may be necessary to cross information from more than one source.

As a second example, we then show the capability of combining multiple data source that is inherent to the visualizations made available by the environment. To this end, we use the dashboard to create a widget including two data sources displayed into stacked layers with two different types of visualization.

In particular, in the example shown in Figure 8 the widget combines data from noise sensors with data from InsideAirbnb. The former are shown as coloured circles, like in the previous example, while the latter are represented through a heatmap that highlights with hotter colours areas having a higher density of Airbnb listings.

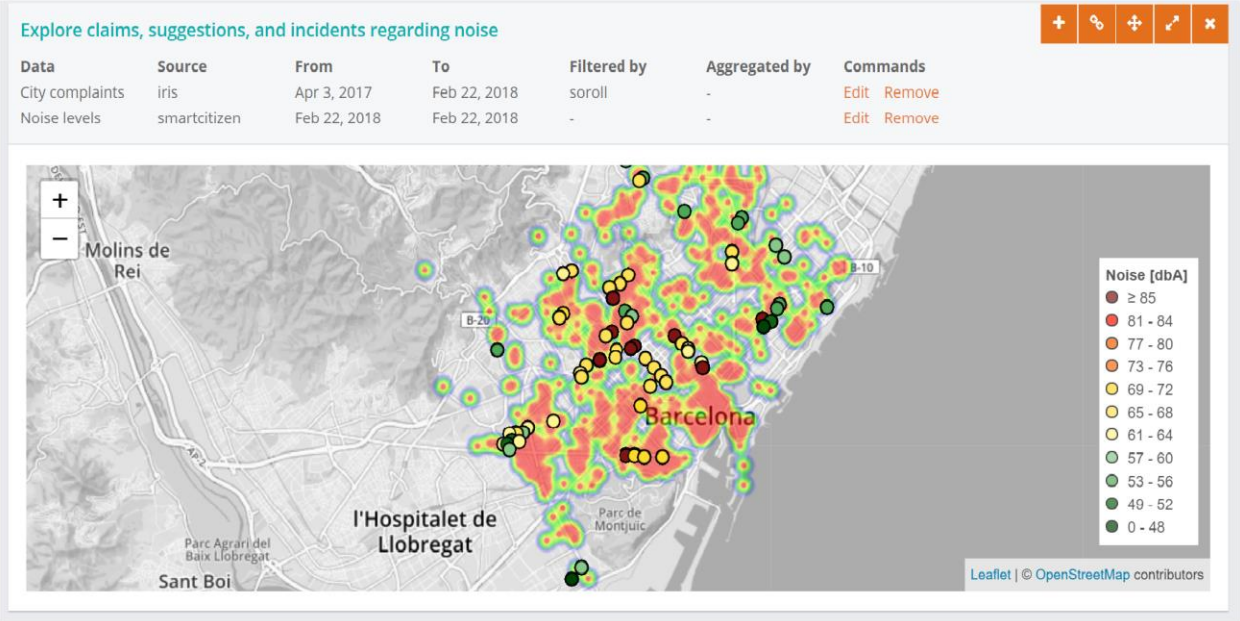

Figure 8. Combined visualization of noise levels (coloured circles) and Airbnb listing density (heatmap with hotter colors indicating higher density of listings in the corresponding area).

In this way, the user can put in contrast the information to get insights and correlations, and to answer questions such as: do areas with high density of short-term rent listings have higher levels of noise during night? These are just some example questions.

Other data sources can be inserted into the debate to enrich the information available to better understand the city dynamics. One example may be showing how citizens complaints about noise from IRIS are distributed across neighborhoods in comparison with the density of short-term rent listings or the location of the noise sensors to detect possible issues, such as whether noise sensors monitor areas affected by a large number of noise complaints. Other visualization types can be leveraged in a similar way.

In this way, the possibilities offered by the dashboard exceed the ones foreseen by who has created the tool, who has developed a visualization module, who has released a dataset or who has added a data source to the system: any citizen or organized group interested in investigating a particular issue in the city or in their specific area is empowered to unveil knowledge and discover urban patterns by leveraging the proposed tool to inspect the relationship between different phenomena.

4.4 Dashboards Creation and Personalization

The environment can be used to arrange the visualizations on multiple personalized dashboards. A dashboard is defined as a set of visualizations that can be monitored on a single view, grouped by topic or type. Each visualization constitutes a widget and a

dashboard can contain multiple widget. Users can create a new dashboard by simply leveraging the "Create a new dashboard" functionality provided on the left sidebar. Furthermore, users can personalize each dashboard by adding or removing widgets and can move widgets between dashboards. Dashboards can be shared, so that what is created by a citizen can constitute a starting point or an inspiration for others, and the collaborative use of the tool can result into a co-creation experiment to foster collective intelligence.

The login functionality is not yet fully implemented, but once login through the DECODE wallet will be implemented, users will be able to access to their personal account and customize their personal space with a combination of dashboards to meet their specific needs and monitor them along different sessions. With the data coming from DECODE, users can also combine such data together with public data already integrated in the environment to get personalized visualization, taken into consideration the privacy guidelines established in D4.7.

5 Data Analysis Methods

The visualizations which can be created with BarcelonaNow expose the data without further processing, apart from the preprocessing and transformation into a common format. As illustrated in the previous section, this approach can enable users to extract knowledge from the data according to specific needs, and also to discover patterns and possible relationships or dependences between different data sources. However, while this can arguably suite most contexts, in some case just exploring the data as they are may not be sufficient, and more complex automatic transformations may help to respond to specific information needs, or get a deeper understanding of city dynamics.

To equip citizens with advanced tools to understand and analyse data, we performed some experiments based on the currently available data, focusing especially on noise sensors. Noise sensors represent one of the richest data sources included in the current prototype, and, at the same time, related to a relevant city issue, acoustic pollution. Noise emerged as an important concern for citizens in the interviews in Barcelona from D1.1; furthermore, it has a strong connection with the CitizenSense pilot, in which users will be able to share noise data from their devices with selected individuals or groups.

In the following sections, we first present a descriptive analysis of the noise sensor data, then we report our unsupervised learning experiments for clustering sensors in order to detect patterns in the noise levels. The same methods are applicable on other datasets which contains time series, such as Sentilo data and Bicing data. The idea is to integrate the resulting techniques into the dashboard and to enable users to create visualizations based on the results of such analysis techniques. Finally, at the end of the section we show a sample analysis that combines noise data with data from other sources to answer relevant questions about the city.

5.1 Descriptive Analysis of Noise Sensors Data

By using the Smart Citizen API, we collected noise observations coming from 50 outdoor noise sensors spread all over Barcelona from January 1, 2017 until October 24, 2017. The acoustic levels were measured in A-weighted decibels (dbA) which consider the sound pressure difference between the average local pressure and the pressure in the sound wave. For instance, quiet libraries are below 40 dbA, houses are around 50 dbA and diesel trucks 90 dbA. The sensors are geographically spread along 10 districts and 25 neighbourhoods. Each district has five sensors in average, while each neighbourhood has two sensors in average. The total number of observations is around 17.708.383 and the measured values range from 0 to 122.6 dbA with an overage of 67.80 dbA (std. dev. 14.10). The average number of observations per sensor is 354.167 (std. dev. 92.492).

The SmartCitizen dataset mainly consists of two files:

1. The "sensors.csv" file lists id, name, latitude, and longitude for each noise sensor.
2. The "observations.csv" file lists the measured noise value, timestamp, and sensor id.

id	longitude	latitude	district	neighbourhood
3955	2.158188	41.397726	Gràcia	la Vila de Gràcia
3956	2.162031	41.394363	Eixample	la Dreta de l'Eixample
3957	2.158731	41.396953	Gràcia	la Vila de Gràcia
3958	2.156858	41.401533	Gràcia	la Vila de Gràcia
3959	2.166330	41.390883	Eixample	la Dreta de l'Eixample
3960	2.158644	41.374877	Sants-Montjuïc	el Poble Sec
3961	2.156256	41.374898	Sants-Montjuïc	el Poble Sec
3962	2.186253	41.413444	Sant Martí	el Camp de l'Arpa del Clot
3963	2.172321	41.397387	Eixample	la Dreta de l'Eixample
3964	2.156430	41.375380	Eixample	Sant Antoni

timestamp	id	value
2017-01-01 00:00:05	3955	63.0
2017-01-01 00:01:05	3955	67.8
2017-01-01 00:02:05	3955	65.7
2017-01-01 00:03:05	3955	68.8
2017-01-01 00:04:05	3955	67.1
2017-01-01 00:05:05	3955	69.7
2017-01-01 00:06:05	3955	68.3
2017-01-01 00:07:05	3955	67.2
2017-01-01 00:08:05	3955	66.6
2017-01-01 00:09:05	3955	66.6

Figure 9. Sample data from the sensors.csv file (left) and the observations.csv file (right).

We analyze the noise cycles regarding the level of noise measured by different sensors. First, we aggregate together the cycles from all the sensors, to infer general noise patterns in Barcelona. Then, we focus on the local cycles, one for each sensor.

Before starting the analysis, we analyzed the amount of observations provided by each sensor in order to remove the ones that have measured only few values. Considering the analysis period (i.e., 01/01/2017 - 24/10/2017, 296 days) and the update frequency (i.e., every minute), each sensor should have provided a total of 60*24*296 = 426,240 observations. However, due to measurement errors or faults, no sensor has provided all the expected number of observations. To maintain coherent our analysis, we selected only those sensors that have provided at least 90% of the number of values expected (i.e., 383,616) and have worked properly (e.g., some sensors have measured always the same value, 100 dbA). Therefore, 30 out of 50 sensors have been used for the remaining of the analysis. The total number of observations they provided is around 12,263,032 and the measured values range from 18.4 to 113.4 dbA with an overage of 64.65 dbA (std. dev. 6.61). The average number of observations per sensor is 408,767 (std. dev. 10,173). Then, we removed the random errors that occur in the data measurement process and we managed the missing values for some sensors during certain periods. It can be due to several factors, such as faulty equipment and environmental factors. Two established methods of dealing with noise in data mining are binning and rolling smoothing. Binning divides data into buckets or bins of equal size. Then, the data are smoothed by using either the mean, the median, or the boundaries of the bin. Rolling smoothing uses both past and future values around the point of interest. In our case, we implement binning and rolling smoothing using the mean.

First, considering the average noise level over all sensors during the day, we get an idea of the global noise level in Barcelona.

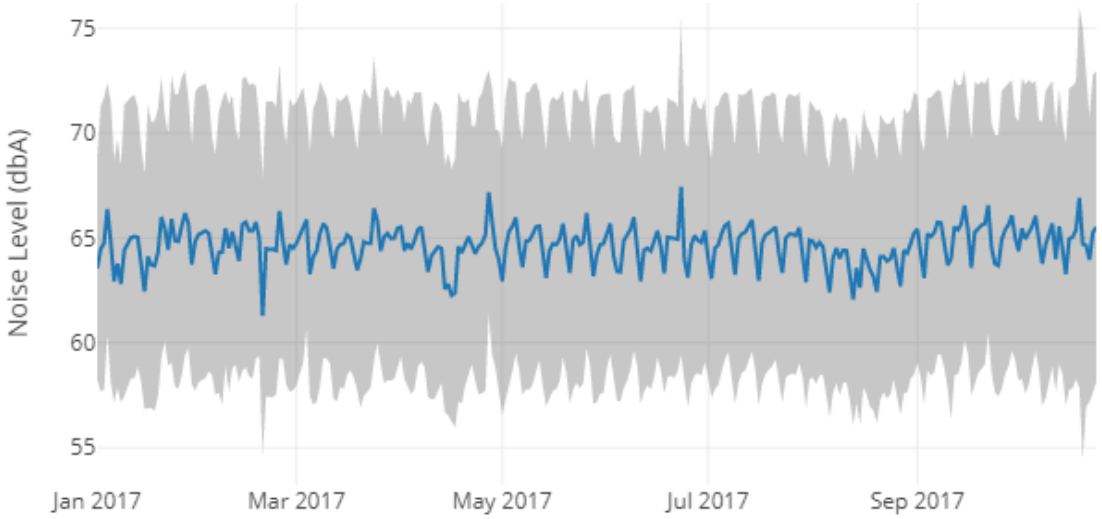

Figure 10. Barcelona average daily noise during 2017. Gray areas correspond to mean ± one std. dev.

In Figure 10, the large standard deviations in gray show that the noise levels greatly vary among the different sensors and, therefore, among different areas of the city. Moreover, the average noise levels by day highlight the presence of a global weekly circadian trend of the city. More precisely, we can observe that the average noise level remains around 62 dbA during weekends, then during the workweek days it goes up at 66 dbA, and finally it goes down when the next weekend starts. With this visualization, we can preliminarily observe that workweek days are generally louder than weekend days in the city.

Second, considering the typical daily routines, we suppose that two different patterns for workweek (Monday-Friday) and weekend (Saturday-Sunday) exist.

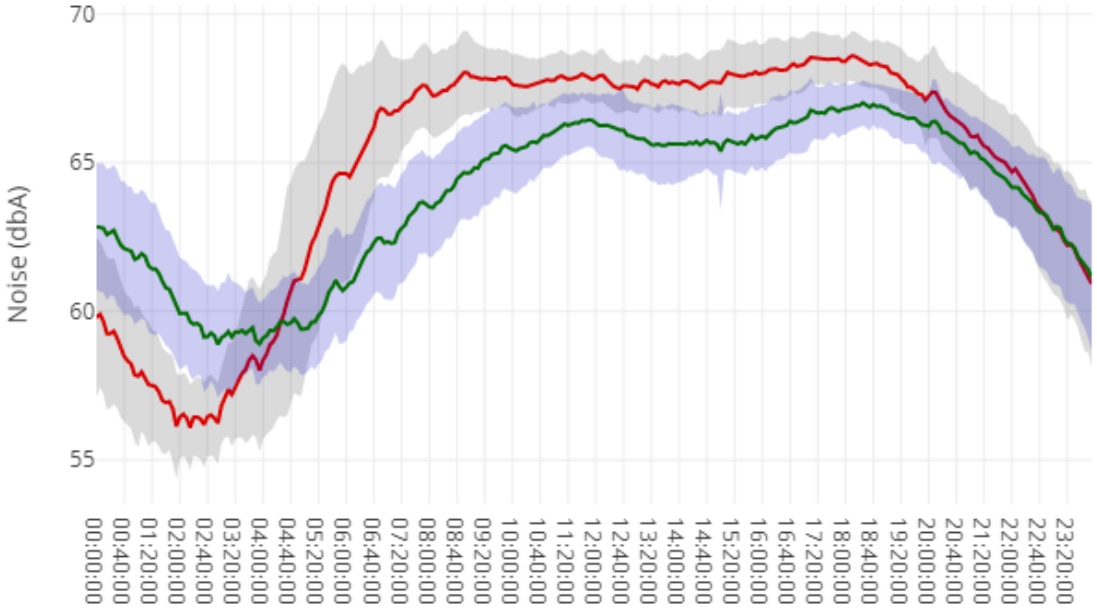

Figure 11. Barcelona average noise on workweek days (red) and weekend days (green). Transparent areas correspond to mean ± one std. dev.

In Figure 11, we plot the average cycles of noise levels for workweek days from Monday to Friday (red curve) and weekend days (green curve).

By analyzing the acoustic levels during workweek days, we observe the global minimum level of around 60 dbA between 2:00 and 3:00. Then, the curve slope rapidly increases until 8:00 when the noise levels start to remain stable at around 70 dbA. The curve goes down after 20:00. The pattern during weekend days is slightly different in the sense that the curve is less skewed, indicating that the noise variation during the whole day is less pronounced. From 13:00 until 16:00, noise seems to decrease by around 2 dbA. Local maximums are detected at 12:00 and 19:00. The curves cross each other at 4:30 and 23:30. Thus, we can observe that workweek days are more peaceful from 23:30 to 4:30, weekend days from 4:30 to 23:30. However, the large standard deviations show that such patterns could be strongly influenced by the area where the sensor is located.

By moving locally, we take a look at the data measured by four specific noise sensors, each one of them exhibiting a different pattern.

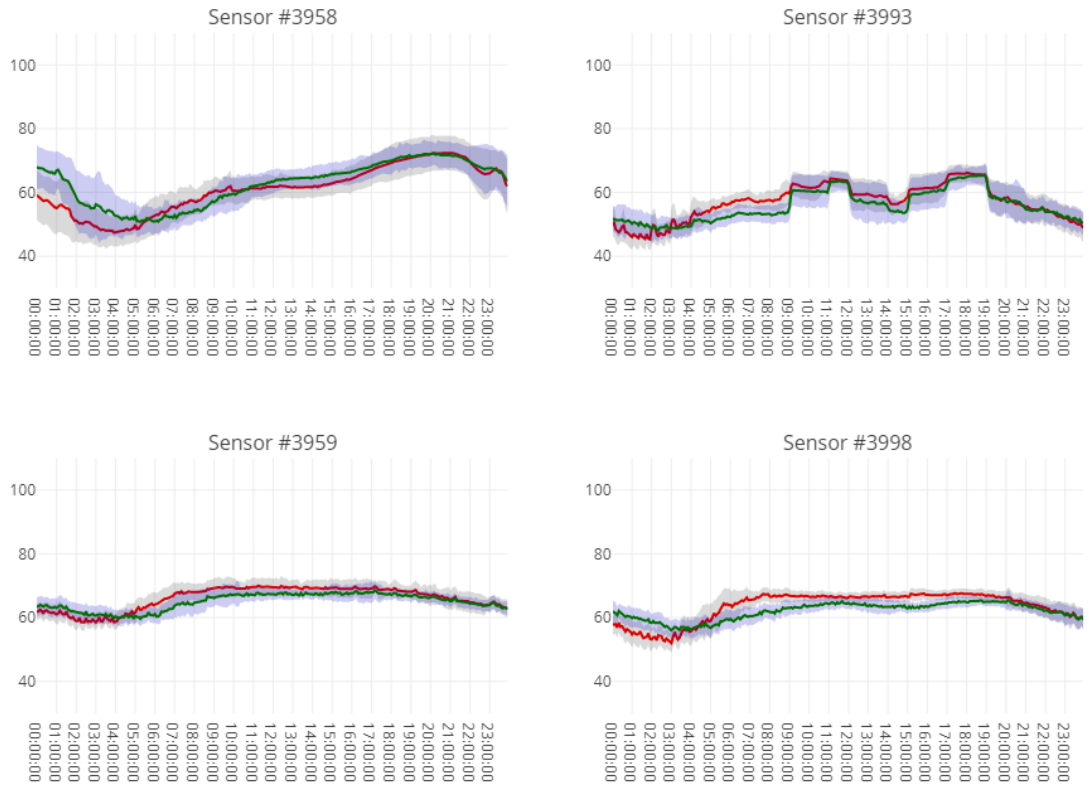

Figure 12. The average noise from sensors on workweek days (red) and weekend days (green). Transparent areas correspond to mean ± one std. dev.

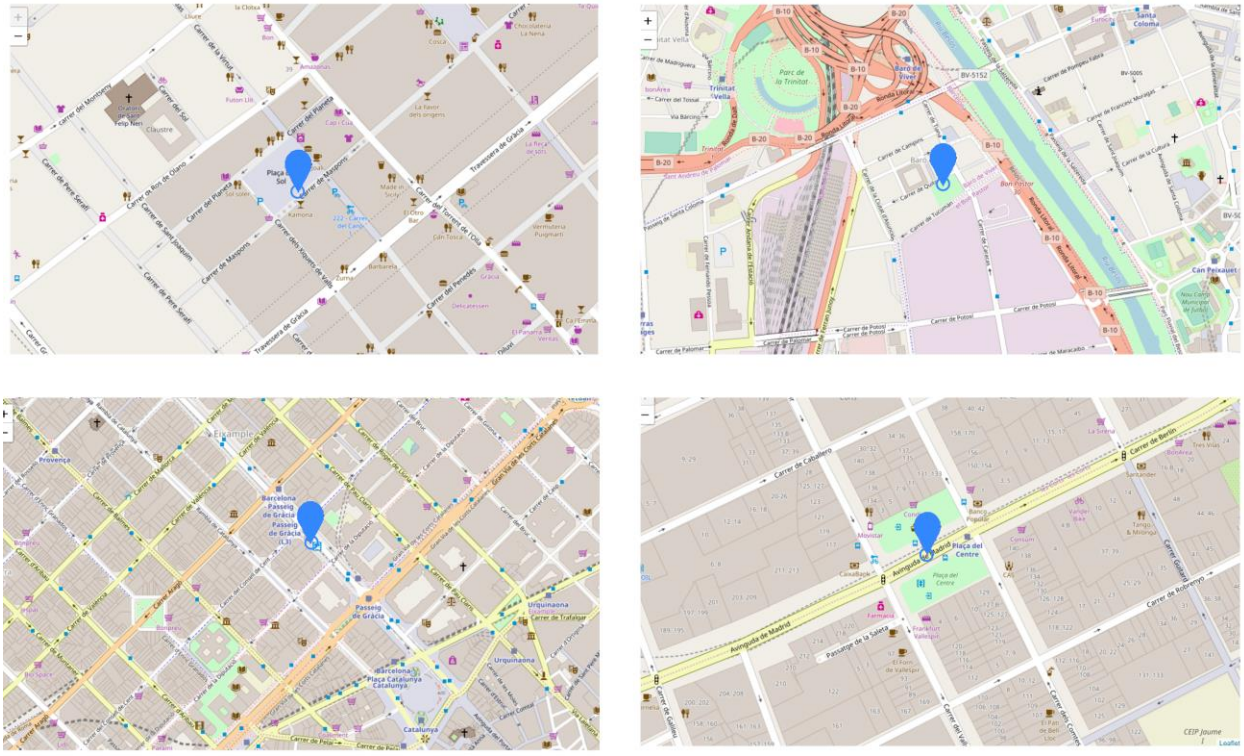

Figure 13. Geographical area where are placed Sensor #3958 (upper-left), Sensor #3993 (upper-right), Sensor #3986 (bottom-left) and Sensor #3998 (bottom-right).

From Figure 13, we observed that *Sensor #3958* is placed in Plaça de Sol, one of the most renowned and crowded areas of Gracia. It has terrace cafes and at night people convene to drink and be merry. Indeed, the CitizenSense pilot involves neighbours from Plaça del Sol, interviewed for D1.1, interested in measuring acoustic levels in the area to get evidence of the noise and support their complaints and demands. Figure 12 shows that the area exhibits a different pattern with respect to the global workweek-weekend trend. In fact, both curve slopes increase from a global minimum of 47 dbA for workweek days (50 for weekend days) during the early morning at around 3:30 to the global maximum of 72 dbA for workweek days (71 for weekend days) during the evening at 20:00. In general, their trends are quite similar, and we observe that they cross each other more than once. Workweek days are more peaceful during night and early afternoon, while weekend days are during morning. Late afternoon and evening are almost equal in the two curves. The difference in the first part of the day could derive from the typical citizens' routines. Considering that the sensor is placed near a lively square, on weekend days citizens are more motivated to go outside during early afternoon to eat or relax and during night to have fun, so noise increases. On the other hand, on workweek days citizens spend their time at work or school, so the square is not crowded and the noise is lower until late afternoon. We can arguably suppose that noise measured by this sensor is generated by people-related activities.

Sensor #3993 is placed in Carrer de Quito, a second-level road in the outskirts of the city. From Figure 12, we observed that this sensor exhibits an interesting pattern with respect to the typical workweek-weekend pattern. The daily noise is overall acceptable. The daily time series has two humps, one in the morning and one in the afternoon. In these two time intervals, the noise goes over 60 dbA. Between the two humps, noise goes down, reaching levels below 60 dbA, at around 55 dbA. Analyzing the geographical area, we observe that the sensor is placed near some schools, one municipal center, and a square. We can arguably suppose that the noise pattern mainly depends on the traffic generated to reach these buildings in the morning and in the evening. The workweek-weekend patterns have the same shape.

Sensor #3986 is the first-level road Passeig de Gracia, one of the major avenues in Barcelona and one of its most important shopping, business and tourism areas, containing several of the city's most celebrated pieces of architecture. The average noise level remains almost the same during both workweek and weekend days. Since the road is located in the city centre, the noise is more pronounced with respect to other sensors spread along the city and less influenced by the typical workweek-weekend-day activities.

Sensor #3998 is placed in a first-level road, Avinguda de Madrid, on the outskirts of Barcelona, a residential area. It does not differ to much from the general city trend. During the day, the sensor measures peaceful and almost constant acoustic levels. This reflects the fact that the sensor is placed into a residential area, so less activities happen during night and traffic seems to be the most influential factor that shapes the noise levels, remaining almost the same during the whole day.

5.2 Temporal Clustering of Noise Sensors Data

The analysis conducted on four sample sensors highlights that all the noise sensors and, consequently, the city areas differ with respect to the measured noise if we take in consideration their workweek-weekend days patterns. We are now interested in grouping such patterns to find similarities among sensors and areas. This can allow one to identify groups of similar areas according to their noise level cycle, and for example to detect which neighborhoods have a cycle with relatively higher levels of noise during nights, and which are outliers with characteristics that are very different from average.

To this end, for each sensor, we considered two patterns, one for workweek and one for weekend. We clustered a total of 60 time series (i.e. 30 sensors x 2 patterns/sensor). We smoothed the time series using a 30-minute window. This also allows us to find correlations between workweek and weekend patterns, since it might also happen that

the weekend pattern of a sensor can be similar to the workweek pattern of another sensor, or vice versa.

5.2.1 Similarity Measure and Clustering Algorithm Definition

A cluster is a set of similar objects, where similarity is defined by a given measure. Thus, the first step was the definition of how to measure similarity between two time series. Then, the second step was the definition of the clustering algorithm to be applied.

Calculating the *Euclidean distance*[37] between two time series can give us a good idea of the similarity between them. The Euclidean distance between identical time series is zero and the Euclidean distance between very different time series is large. Therefore, we have used it as distance measure in the preliminary steps. However, with a good similarity measure, small changes in two time series should result in small changes in their similarity. With respect to Euclidean distance this is true for changes in the y-axis, but it is not true for changes in the time axis (i.e. compression and stretching). Thus, it often produces pessimistic measures when it encounters distortion in time axis. In addition to Euclidean distance, we have also tested the Dynamic Time Warping (DTW) distance. Such distance is widely adopted in time series analysis. It finds the optimal non-linear alignment between two time series. With it, the Euclidean distances between alignments are then much less susceptible to pessimistic similarity measurements due to distortion in the time axis. However, there is a price to pay for this: DTW is quadratic in the length of the time series used. If we perform DTW multiple times on long time series data, this can be prohibitively expensive. In order to speed things up, we used an approximation of DTW that has a O(n) time and space complexity, namely *FastDTW*[38]. After preliminary tests, we have observed that, in our current case, Euclidean distance and FastDTW produce very similar results, so we have included only the analysis conducted using Euclidean distance.

After defining a method to determine the similarity between two time series, the next step is to perform clustering. We can use the *KMeans*[39] clustering algorithm. With this algorithm, the number of clusters is set a priori and similar time series are clustered together. KMeans is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set into a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because different location cause different result. So, the better choice is to place them as much as possible far away from each other. The next step consists in taking each point belonging to a given data set and associating it to the nearest centroid. When no point is pending, the first step is completed and an early

[37] https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.spatial.distance.euclidean.html
[38] http://cs.fit.edu/~pkc/papers/tdm04.pdf
[39] http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

grouping is made. At this point, we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. Then, a new binding has to be established between the same data set points and the nearest new centroid. A loop is generated and the k centroids change the location step by step iteratively until the partition converges and no more changes are done.

KMeans is a partitioning algorithm. This class of algorithms is popular due to the ease of implementation and low computational cost; however, they have disadvantages because they can discover only clusters with convex shapes and the number of clusters is pre-defined. In future steps, we plan to analyze the results obtained by applying other existing clustering techniques: hierarchical clustering and density-based clustering.

5.2.2 Experiments

To find the optimal number of clusters, we carried out the "elbow" method[40] which finds the point after which an increase in the number of clusters implies only minor improvements, and looked at the silhouette, which is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation)[41]. As it can be observed in Figure 13, a curve elbow seems to be located at k=2 or k=3. We also calculated the silhouette score at all values of k until 10. The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. The highest silhouette score was found for k=2 (0.52), higher than for K = 3 (0.42) and for the other values of K until 10. Thus we distributed the time series in 2 groups according to the labels computed by KMeans. The cardinality of each cluster is shown in Figure 14, while the clusters compositions are depicted in Figure 15.

[40] https://en.wikipedia.org/wiki/Elbow_method_(clustering)
[41] https://en.wikipedia.org/wiki/Silhouette_(clustering)

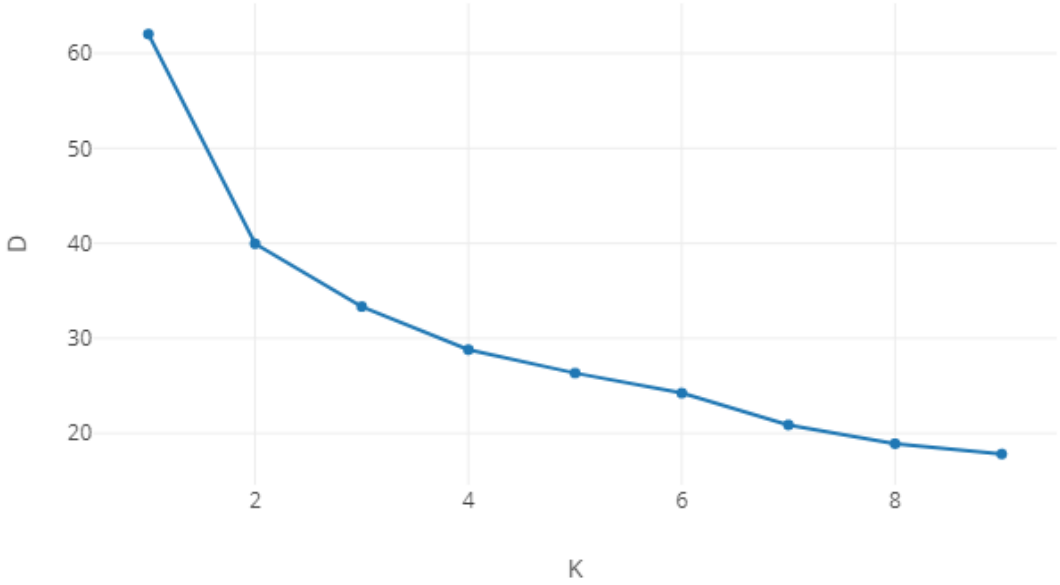

Figure 14. Performance increase for increasing values of K. K is the number of cluster, while D is the distortion (i.e., the sum of the squared distances between each observation and its closest centroid).

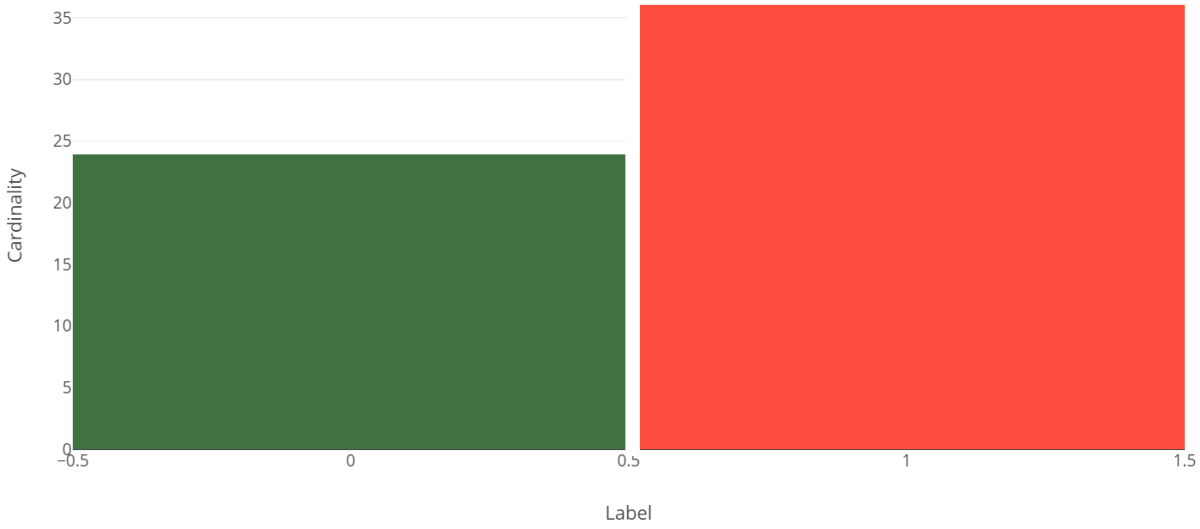

Figure 15. Histogram representing the distribution of time series among clusters.

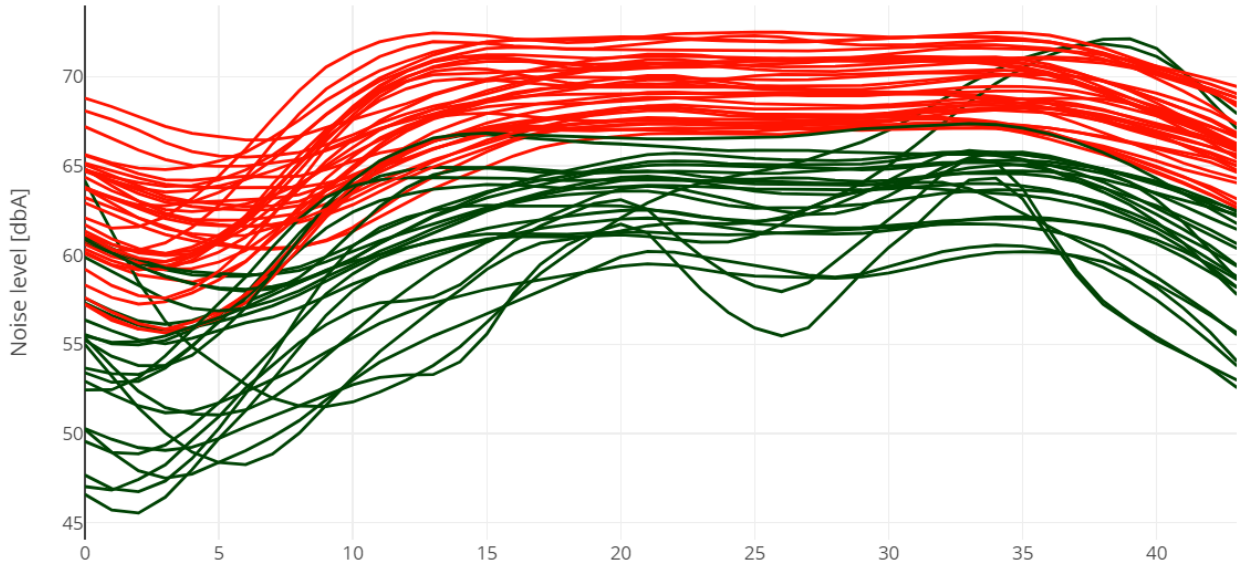

Figure 16. The workweek-weekend time series of all the sensors, coloured according to clusters assignments.

The time series appear well clustered. The clusters need further inspection to identify abnormal overlappings. To interpret the obtained results from a geographic perspective, we plot a map showing one marker for each sensor for workweek days. We do not plot the map for weekends since we discovered that, in almost all cases, workweek and weekend time series of the same sensor are assigned to same cluster.

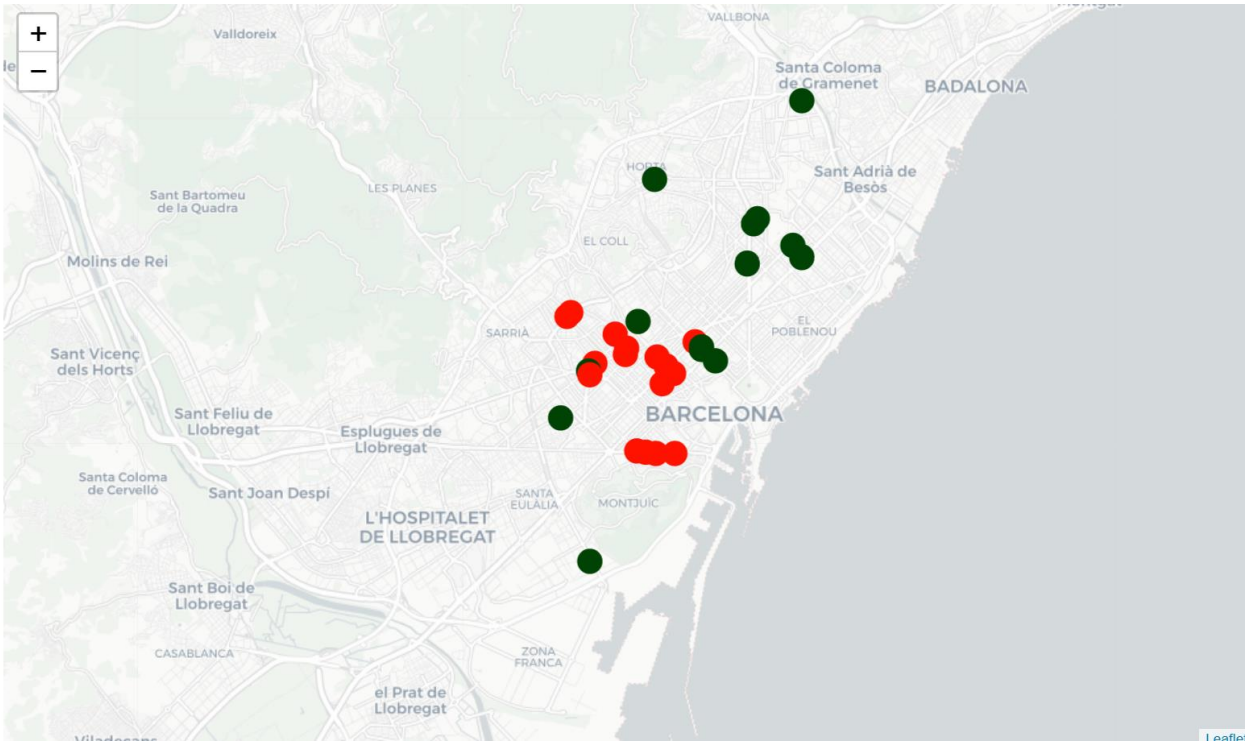

Figure 17. Geographical distribution of noise sensors along the city. The color of the marker represents the cluster within the sensor is assigned and the legend is the same used in Figure 13 for consistency.

Mapping clusters geographically enables us to identify different areas exhibiting similar daily behavior of noise levels amplitude. There are 6 green markers which are associated to peaceful areas during workweek days (avg. 55 dbA). These sensors share common characteristics: they are on the outskirt residential areas of the city and they are placed on second-level streets. While moving to the city centre, the noise levels increases and this is reflected into the clusters points distribution.

To provide robust analysis focused on the shape of the daily cycle rather than on the overall level of noise, we have scaled each time series to have zero mean and unit variance (i.e., amplitude scaling). We applied this scaling only to workweek and weekend time series since we have observed from the experiment above that they well describe the noise patterns for the sensors. To do amplitude scaling for each time series, we subtracted from each observation the mean of the values in that time series and we divided such difference by the standard deviation of the values in that time series. Then, we repeated the clustering on the resulting workweek and weekend time series. With this step, we were focused more on the shape of the time series rather than on the pure amplitude of noise levels. Clustering amplitude-scaled time series enables us to group time series which have similar shape even if they are shifted in the y-axis. As it can be observed in Figure 18, a curve elbow seems to be located at k=2. To select a value of k, we further calculated the silhouette score for all values of k until 10. The resulting silhouette score for k=2 was 0.51, while it was 0.53 for k=3, higher than for any other value of k until k=10. Therefore, we selected k=3.

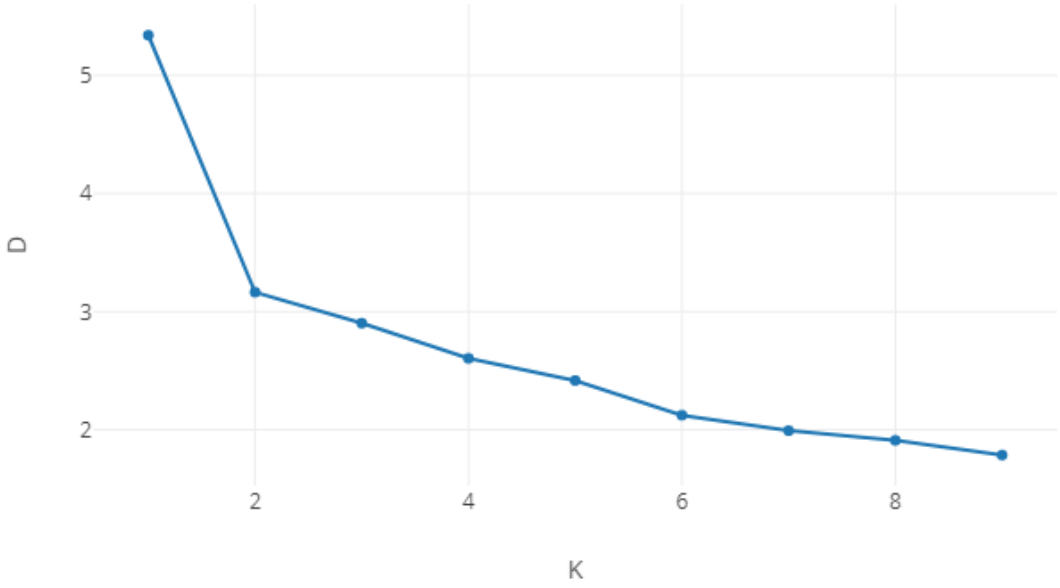

Figure 18. Performance increase in function of the number of clusters K. K is the number of cluster, while D is the distortion (i.e., the sum of the squared distances between each observation and its closest centroid)

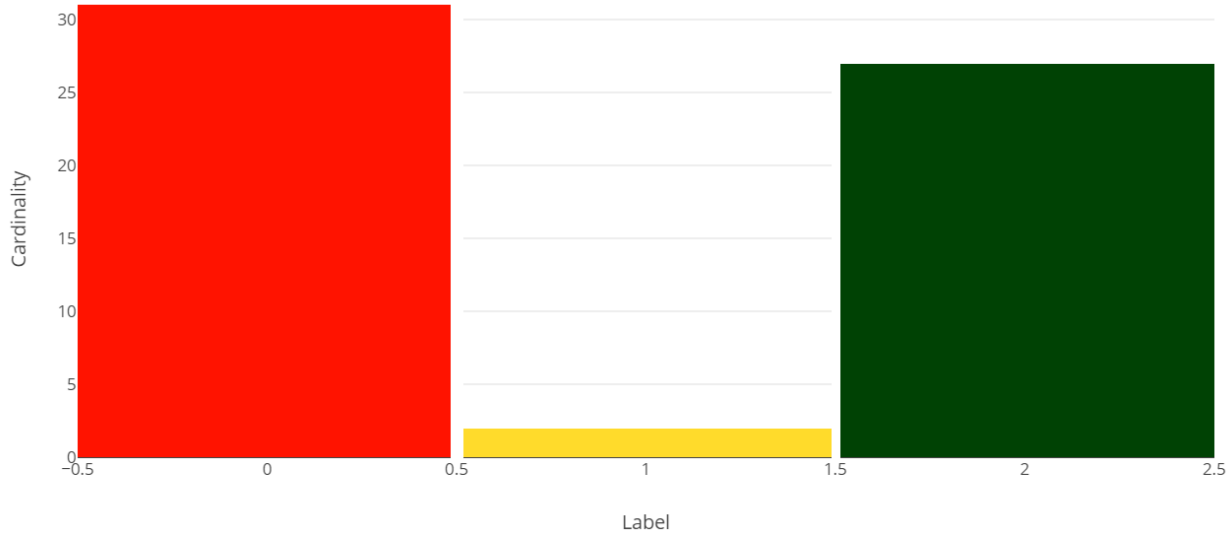

Figure 19. The distribution of time series in clusters for amplitude-scaled time series clustering.

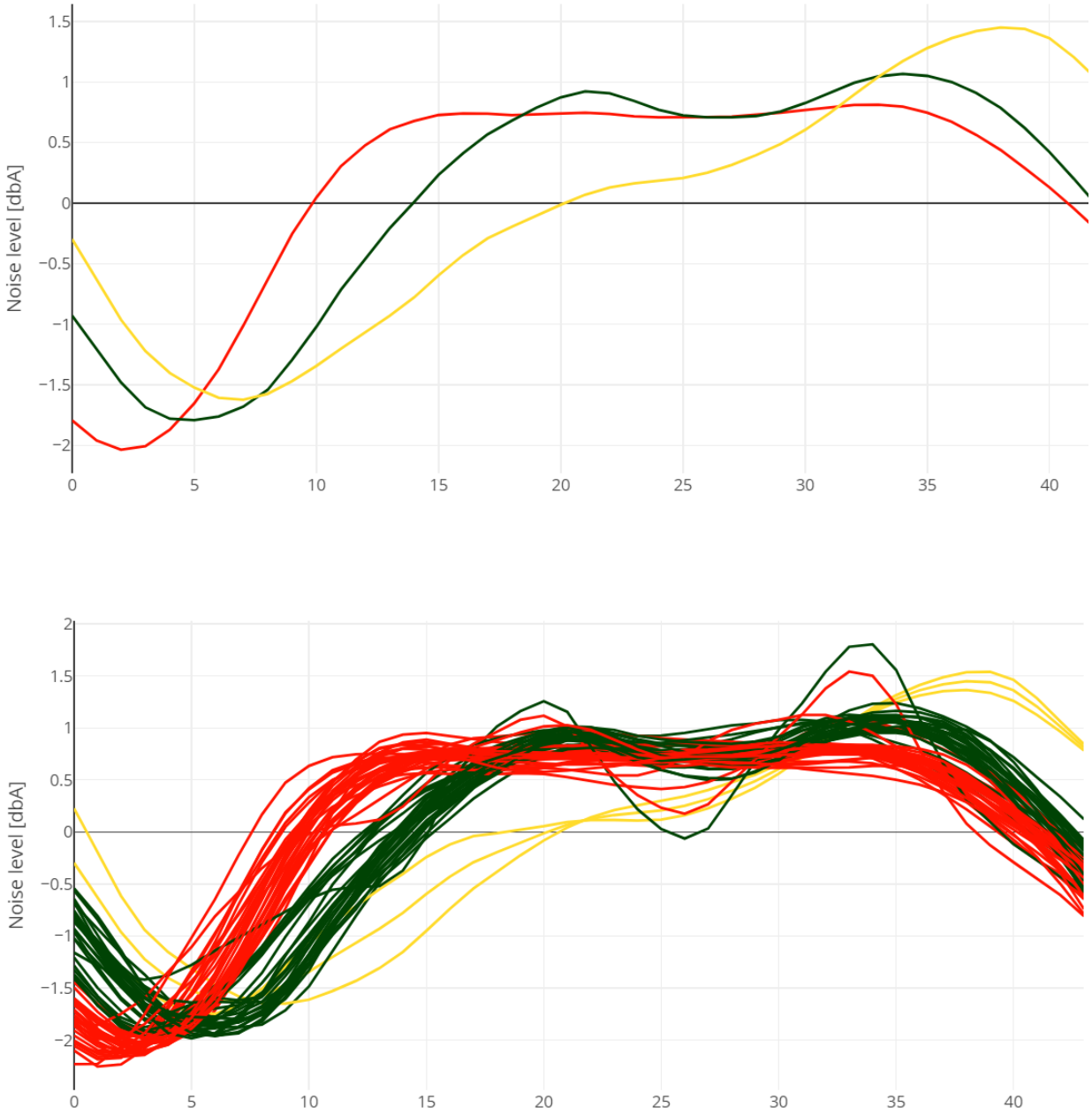

Figure 20. Centroids of clusters in amplitude-scaled time series clustering (upper). Amplitude-scaled time series distribution between clusters (bottom).

The red and green clusters contain almost the same number of time series, around 30 each, while the yellow cluster contains only two time series. From Figure 20, we observed that red and green clusters' centroids seem to reproduce the shape of workweek and weekend time series. From this, we can arguably think that the the clustering process which uses amplitude scaling can split groups of time series regarding the two parts of the week. More precisely, the red cluster seems to mostly reproduce the workweek daily behavior, while the green cluster the weekend daily behavior. In fact, by observing the clusters composition, we noticed that for 27 sensors the

workweek-weekend time series have been correctly splitted between the clusters. For the other 3 sensors, two of them (i.e., Sensors #3962 and Sensor #4009) have both workweek-weekend time series in the workweek cluster (Figure 21 and Figure 22). Workweek and weekend patterns of Sensor #3958 seems to have a completely different pattern on both workweek and weekend days, so the related time series have been placed in a separated cluster. As shown in Figure 21 and Figure 22, we observed that the clustering seems to work properly.

Overall, without performing amplitude scaling on time series, we can cluster time series with respect to the different levels of noise, while with amplitude-scaled time series, we can cluster them based on their shape, and their workweek-weekend patterns.

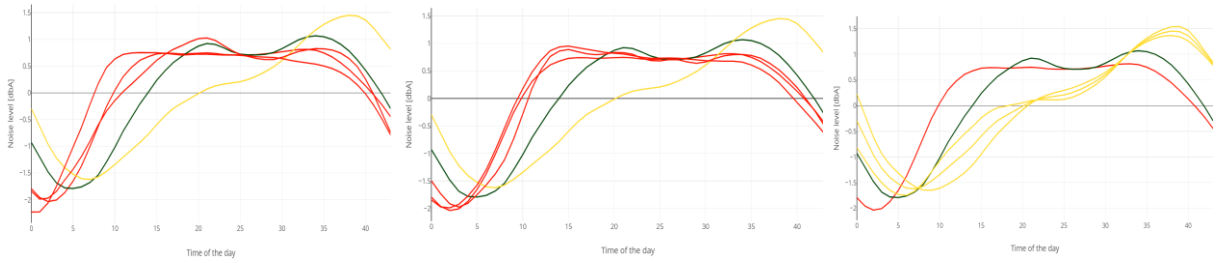

Figure 21. On the left, the workweek centroid and the workweek-weekend time series for Sensor #4009 are depicted in red, the weekend centroid in green. On the center, the workweek centroid and the workweek-weekend time series for Sensor #3962 are depicted in red, the weekend centroid in green. On right, the workweek centroid in red, the weekend centroid in green, and the workweek-weekend time series for Sensor #3958 together with the related centroid in yellow.

Figure 22. Geographical areas where Sensor #4009 (left), #3962 (center), #3958 (right) are placed.

The entire set of experiments was also repeated considering all the daily time series without aggregating them in workweek and weekend time series. We obtained comparable qualitative results; thus, we have not included the details in this document.

5.3 Combined Data Source Analysis

In this section, we are interested in putting in contrast the information coming from different data sources to get insights and correlations, and to understand the dynamics behind noise pollution. As an example, we combine the geographical distribution of complaints about noise from IRIS with clustering of noise sensors temporal patterns described in the previous section. This can be helpful for example tor a citizen to understand how a high amount of complaints in their neighborhood might be related to some aspect of the daily pattern of noise in the area. It can be also helpful to the city government policy makers to understand which kinds of noise temporal patterns are likely to produce more citizen complaints.

The IRIS dataset includes several types of complaints, incidents and suggestions that citizens send to the city council. In particular, one type refers to the complaints raised due to problems related to noise levels. The dataset specifically includes 2.910 noise complaints raised from January 2016 to December 2017. Their distribution based on the motivations that citizens indicated when sending the complaint is depicted in Figure 21.

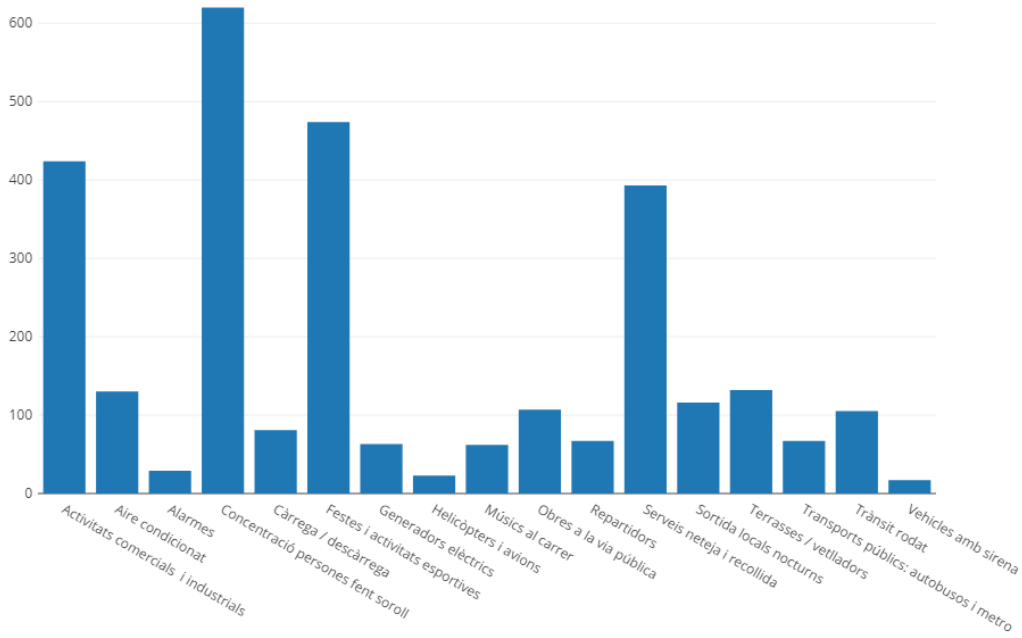

Figure 21. Distribution of complaints related to noise by subcategory, from IRIS data.

We combined a choropleth visualization of the noise complaints distributed along geographical sections from IRIS with the marker-based visualization of noise sensors clustered based on their workweek day pattern (Figure 14). Combining the data sources, we immediately observed that the areas mostly affected by noise complaints are not covered by any noise sensor, so it was not possible to detect evidence for noise sensor measurements in relation to noise complaints. In such case, citizens can create

and use the visualizations also to promote a data-driven proposal requesting additional noise sensors in the affected areas. For instance, considering the large amount of complaints in the Sant Antoni's census section and the absence of noise sensors there, a citizen who lives in that section could add a petition asking to install some noise sensor in the area. Furthermore, we noticed that the noise complaints mainly come from the outskirts of the city, even though a wide range of noise sensors located in the city centre measure extremely high levels of noise. However, the highest density of complaints is found in central areas such as Gotic, as illustrated below.

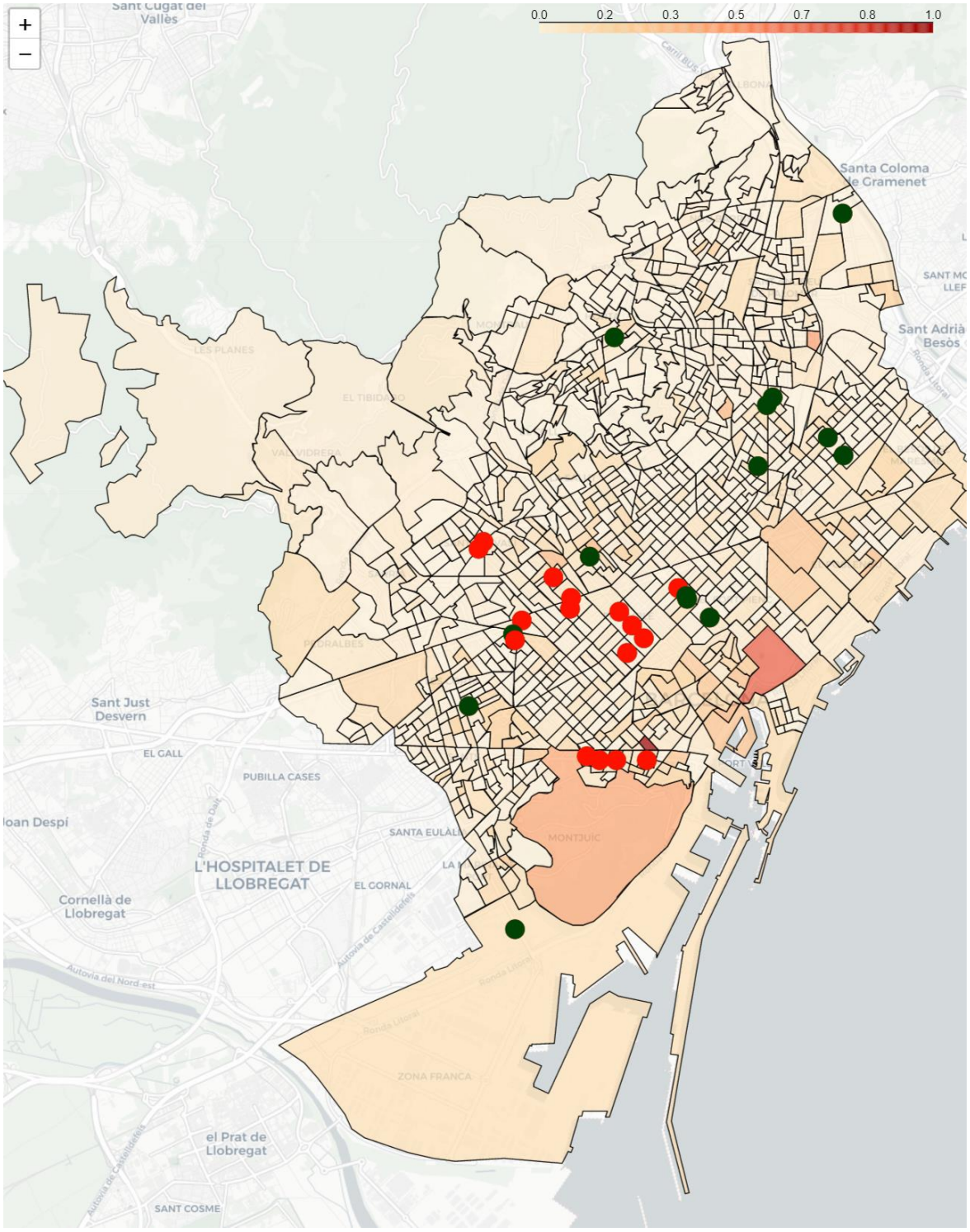

Figure 22. Combination of a choropleth of the noise complaints distributed along geographical sections from IRIS and a visualization showing noise sensors clustered based on workweek pattern from Smart Citizen.

The IRIS dataset reveals the exact latitude and longitude only for a small set of complaints, while only the geographical section is reported for the others. At this stage, we were interested in finding evidence of noise complaints in noise sensors data for the

areas where both of them are present (i.e. the section has a hot color and includes at least one sensor). This is useful to see whether the complaint was raised when the noise sensor had really experienced a change from its normal behavior. No sensor from the 30 ones selected to perform the analysis has this property. However, one of the sensors we discarded in the preliminary step, Sensor #4006, is placed within a geographical section where 4 complaints have been raised and has provided noise levels during the days when the complaints have happened. We went deep into the analysis of Sensor #4006 at the moment when the complaints were raised in the geographical section where it is placed. On May 8, 2017, the complaint has been caused by people making noise in the street. Since we do not have the exact time when the complaint has been raised, we can only arguably suppose that the complaint has been raised during night. By observing the noise levels during night, we have noticed that around midnight the sensor measured noise levels around 70 dbA, a higher level, which were not measured in the other days around the same time. On June 6, 2017, the reason behind the complaint was the noise made by commercial and industrial activities. From the reason, we hypothesize that it has been raised during the day. During the morning, we observed an abnormal peak at around 10:00. The same is true for the complaint raised on July 20, 2017. In that case, the peak has been more pronounced: over 75 dbA. On August 8, 2017 the reason of the complaint has been the noise generated by sanitation services. We can arguably suppose that the complaint has been caused by the peaks observed at around 23:00.

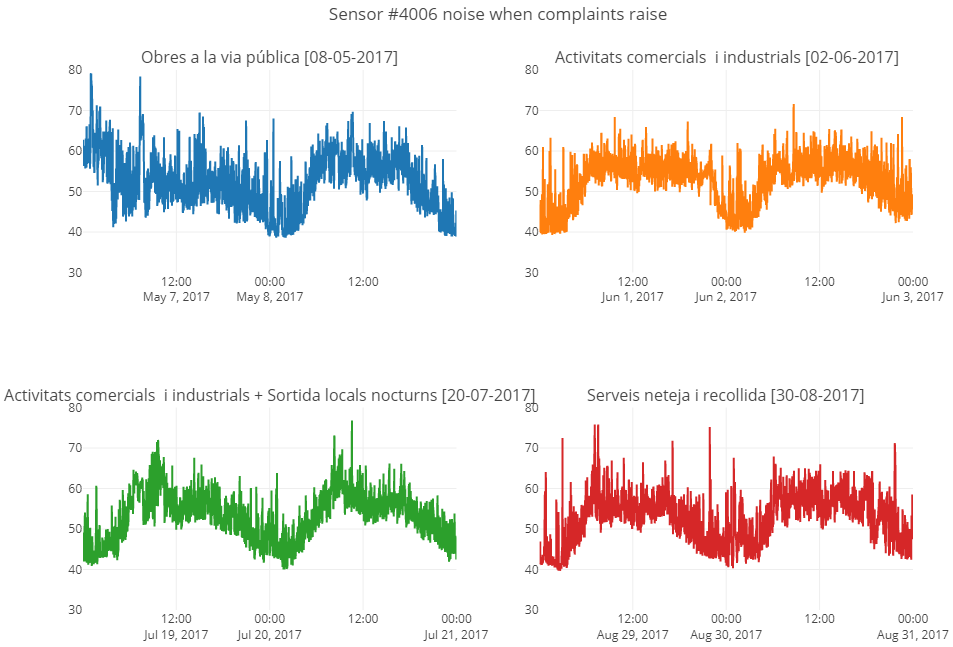

Figure 23. Temporal trend for four sensors from geographical sections and days correspondence with some complaints.

One issue with Figure 22 is that it shows the absolute number of complaints by census section, but census sections vary in size. Therefore, the areas with high amounts of

complaints in some cases correspond just to very big census sections. In order to take into account the size of each geographical section into the choropleth visualization, we calculated a normalized score to measure complaints distribution along geographical sections. More precisely, we computed a score by dividing the number of complaints in each section by the geographical size of that section, then we performed min-max normalization in the range [0,1] for all the scores. We used these scores to depict the choropleth map in Figure 24.

The sections with the higher scores are the number 170 in Sant Antoni, Eixample (50 complaints have been raised in a geographical area of around 30,000 m2) and the number 19 in Sant Andreu, Sant Andreu (22 complaints in around 50,000 m2). The normalized scores help to highlight the areas where complaints are more geographically concentrated. Therefore, with respect to the choropleth in Figure 22, sections such as the number 24 in Poble Sec, Sant Montjuic (21 complaints in 3,803,040.8 m2) and the number 43 in Sant Pere, Santa Caterina i la Ribera, Ciutat Vella (35 complaints in 540,757.38 m2) have been colored with lighter colors in Figure 24. Several complaints have been raised within them, but they are also geographically wider. The associations between section number and geographical area can be retrieved online on the Barcelona municipality web site[42].

Enabling citizens to compose and combine these kind of visualizations can help them to raise awareness regarding noise driven by data. Moreover, with DECODE, we expect that the set of available sensors will increase and citizens can exploit personalized visualization of noise levels with data responsibly shared with other citizens.

[42] http://www.bcn.cat/estadistica/catala/dades/inf/ele/ele40/Mapes5.htm

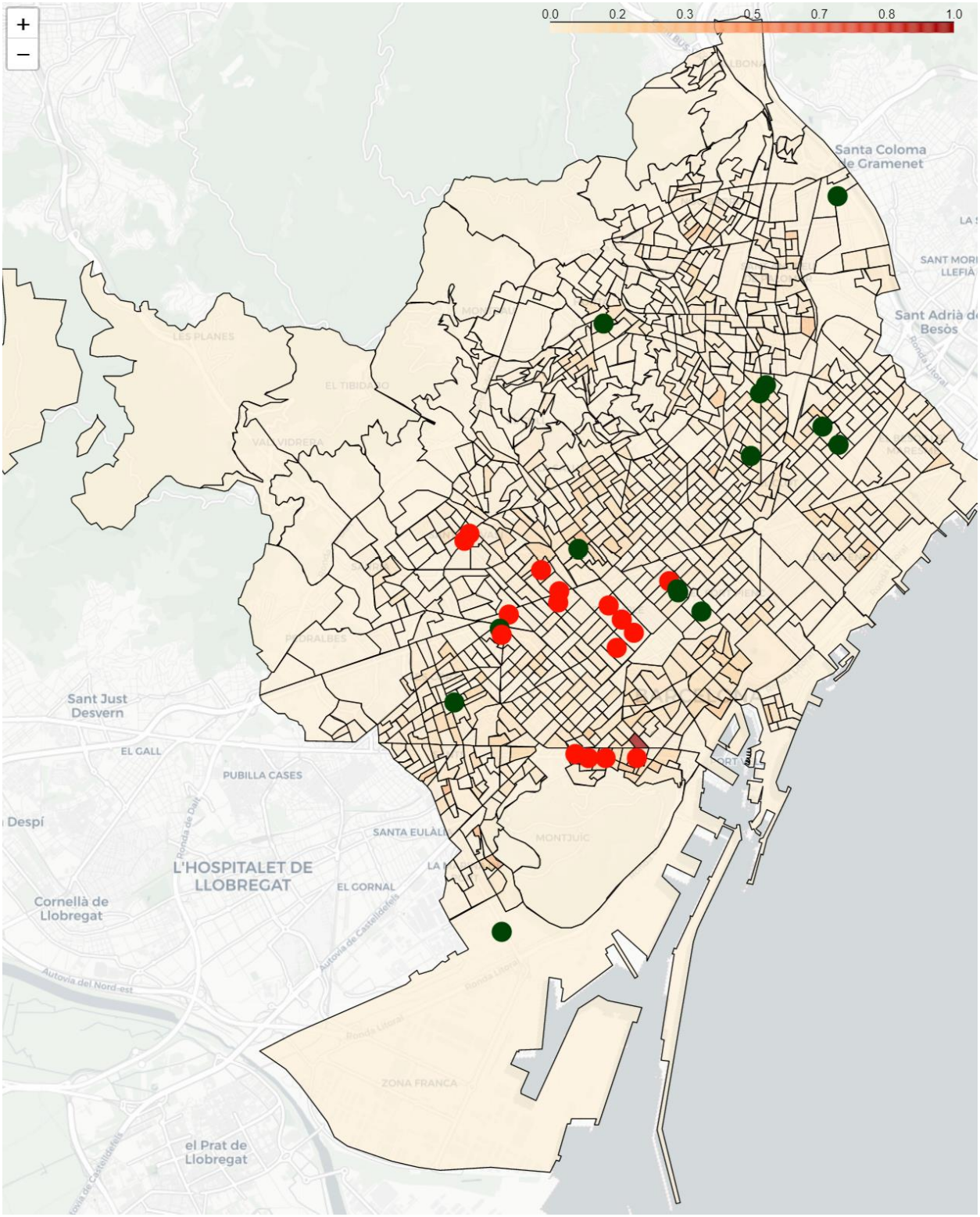

Figure 24. Combination of a choropleth of the noise complaints distributed along geographical sections from IRIS normalizing the number of complaints with the geographical size of the section.

6 Roadmap

The prototype system and the data analyses presented in the previous sections rely on public data from Barcelona, because private data shared through the DECODE infrastructure are not yet available. However, the system has been designed in order to be modular and to allow for integration of such data when available. New data sources can be added by adding a new collector as explained in Section 3, in case of public data to be stored in the system, and exposed through the API; in case of private data, they can be integrated as a parallel stream of data to be shown on the fly without storing anything on the backend, as all the visualization code runs on the client side. While here we have until now mostly focused on maps, which are especially relevant to understand urban data, in next developments we plan to leverage the idea of enabling citizens to compose their own visualizations for other visualizations types (e.g, donuts, bars, and so on). With this, we can provide data manipulation and exploration flexibility beyond maps, for a variety of use cases, including relevant use cases from the pilots in which not all data sources come with geo-temporal attributes (e.g., demographic statistics on the IDigital DECIDIM pilot, as described below).

BarcelonaNow aims to connect and unify the data generated in the use cases developed in the city of Barcelona and in the other pilots, to use them as test-beds for the study of the policy and technological framework needed to successfully bootstrap communities around a combination of publicly, private and collectively gathered datasets. With this objective in mind, we define the roadmap for further development of Barcelona Now in relation to the other pilots. The figures below depict the foreseen timeline of the next steps, and its correspondence with the related deliverables, as discussed and defined in the General Assembly in Amsterdam, on February 1st, 2018.

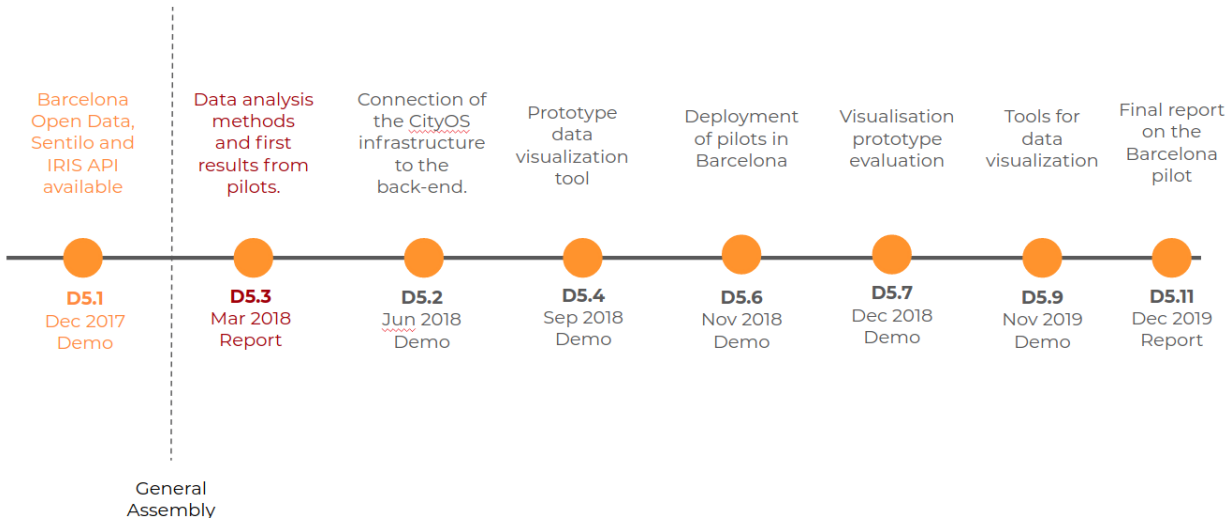

Figure 25. Timeline of the deliverables related to BarcelonaNow.

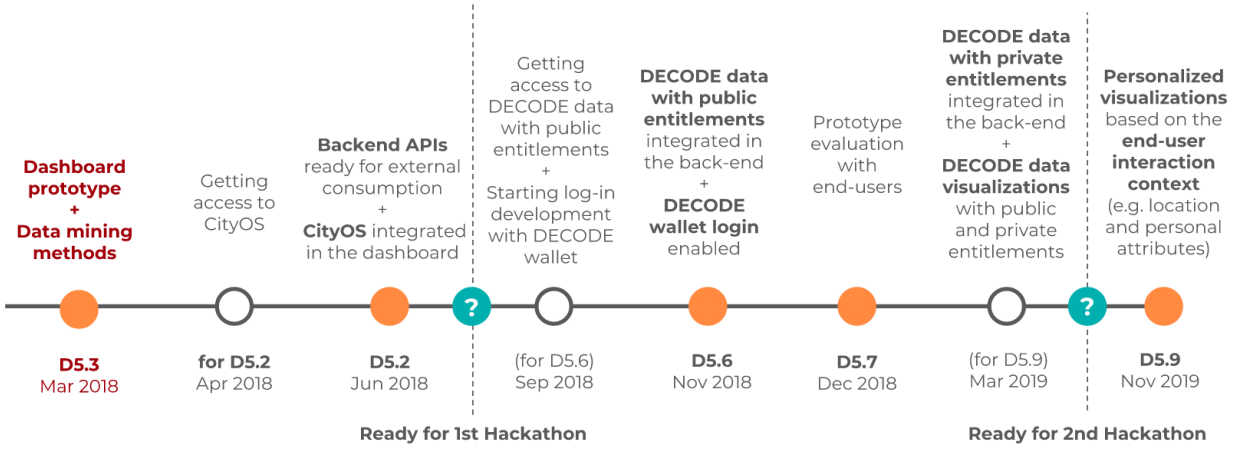

Figure 26. Timeline of the next steps in BarcelonaNow development and integration with the pilots.

The next two subsections focus on the integration of BarcelonaNow with the CitizenSense and iDigital Decidim pilots, introduced in D1.1 and briefly depicted through a schema in Figure 27.

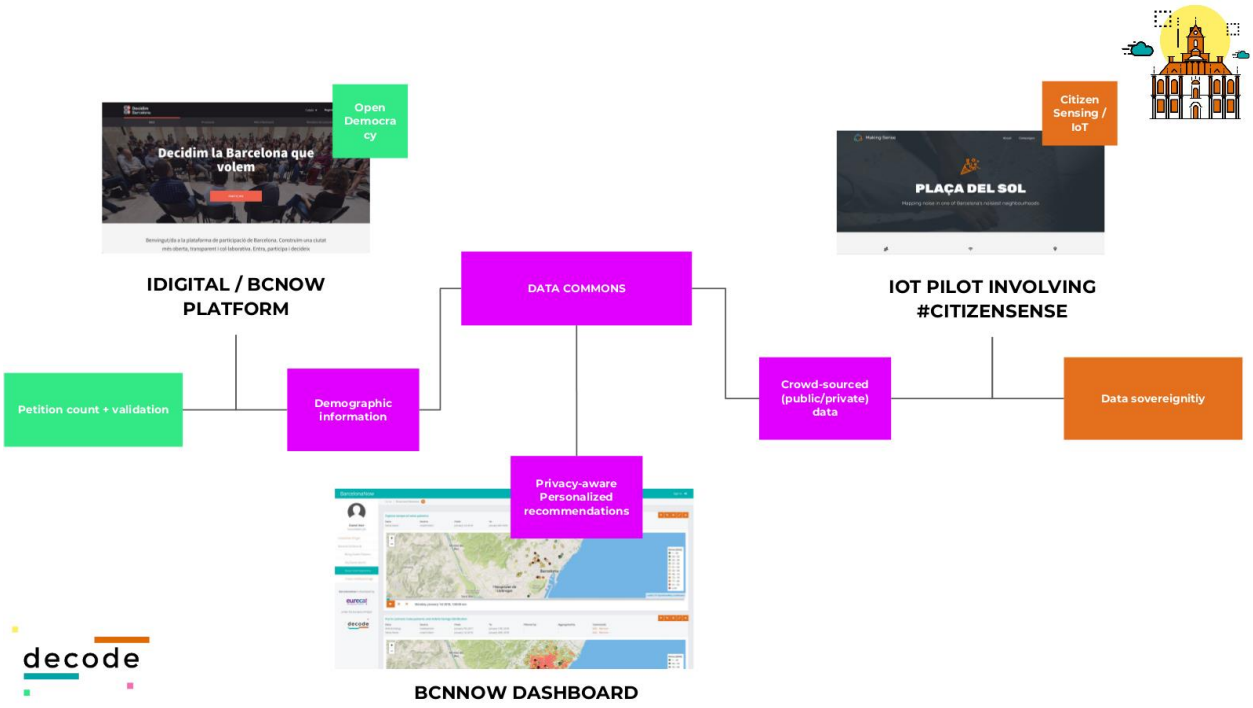

Figure 27. Explicative schema of the Barcelona pilots integration and relationship with BarcelonaNow

6.1 CitizenSense Roadmap

BarcelonaNow is currently leveraging the SmartCitizen infrastructure and the data gathered in the framework of the MakingSense project[43]. These datasets, however, are on the public domain at the moment, and its users have expressed interest in gaining, through the DECODE infrastructure, means to win governance tool over these datasets that allow them to define a variety of privacy access levels for their use according to different target use groups.

Once this pilot has started its deployment, its integration with Barcelona Now will result in the personalization of the visualizations based on data that users have permissions to access and/or information about the devices they have currently claimed (and hence control). Users will be able to log into BarcelonaNow with their DECODE wallet, and then through the dashboard they will access and visualize data they have permission to see. In relation to what the dashboard currently offers, with the DECODE platform, users can view different sensors and data analysis points on maps based on the sensors and the related observations they have permission to access. Possible combinations between this privately-shared data and public data already stored in Barcelona Now backend will be explored to provide advanced functionalities to users.

Private data from CitizenSense will not be stored into the Barcelona Now backend, but retrieved via the DECODE infrastructure of the MakingSense pilot and used upon request to depict visualization directly on the frontend (i.e., client side). The details of this process will be fleshed out as the pilots are deployed, and the different components of DECODE relating to the CitizenSense pilot are put into action. Based on the outcomes of the pilot and its reach, different potential levels of personalization will be achieved.

6.2 IDigital Decidim Roadmap

The iDigital Decidim pilot designed in D1.1 will enable users to securely take part in petitions related to deliberative processes. At the same time, this pilot will allow users to selectively disclose some of their demographic information in the form of aggregated counters at the time of petition support, thus creating a common pool of information appropriately anonymized that can be used for mapping citizen needs.

In relation to this, the functionalities developed by BarcelonaNow will be used to effectively enable citizens to explore visualizations related to the demographic and/or geographical distribution of users personal attributes in relation to petitions support in a privacy-preserving way. In particular, the dashboard will be used to generate

[43] http://making-sense.eu

personalized visualizations by enabling them to login to the dashboard through their DECODE wallet, and to share some of their personal attributes. Leveraging and properly adjusting the data visualization methods presented in this deliverable, the dashboard will show personalized views that can be of interest to the citizens. More precisely, the integration with IDigital Decidim will include the following aspects:

- Decidim data, including aggregated demographic information donated by the users via the iDigital Decidim DECODE app, will be integrated for visualization into the dashboard.
- Decidim participants will be able to log into BarcelonaNow through their DECODE wallet, and to access personalized information and visualizations on their petitions.

It is expected that the use of this tool fosters fact-based deliberative processes on the platform and engages users in argumenting and proposing actions about city issues in the Decidim platform in a data-driven way.

Also, this technology will help in closing the circle by shaping a clear use-case and a field of experimentation for the concept of data-commons, that is, a crowd-sourced dataset of citizen data which is democratically governed and used to solve societal challenges of public interest.

7 Conclusions

With the proposed environment, we have presented a new and comprehensive way of enabling citizens to define and tailor city-related data visualizations by themselves. Instead of relying on pre-packaged solutions that can only address limited and common needs, we enable citizens with little or no technical expertise to create their custom visualizations by leveraging the provided functionalities, while controlling the access and sharing of their own data in a privacy-enhancing manner.

Our proposal constitutes an ecosystem in which citizens, but also other stakeholders (e.g., policy makers and city administrators) can share and co-create visualizations to increase public awareness on city issues, supporting an open, transparent and democratic city. Furthermore, in line with DECODE principles sketched in the white paper, data are exposed through an API, so other services and interfaces can be developed on top of the backend. The code is released under open source licence, and is highly modular, so that other data sources can also be integrated by adding appropriate data collectors to the backend, and the infrastructure can be easily deployed for other cities.

In this document we have illustrated the system and demonstrated its capabilities with public data from Barcelona to respond to citizen needs, also in relation with user needs and persona definition from D1.1. Beyond providing an interface to explore rough data from the city, we have performed a deeper analysis of noise temporal patterns, and experimented unsupervised clustering algorithms to identify patterns by comparing noise trends across areas and over time, and to detect anomalies in the data. The results of this kind of analyses, such as the clusters obtained and their centroids, can be integrated into the dashboard as another possible layer, to offer enriched visualizations and enable more advanced explorations. The same methods can be applied to other data coded in the same format.

At the current stage, the proposed architecture has been deployed and demonstrated with public data from the city of Barcelona. In the next steps, as illustrated and discussed in the Roadmap section, aggregated demographic data from the IDigital Decidim pilot, and private data shared in a privacy-enhancing way through DECODE in the CitizenSense pilot will be integrated as further sources of data feeding the data exploration interfaces. Furthermore, the integration with the DECODE wallet for logging into Barcelona Now will make possible to offer personalized services and visualizations, based on user contextual information shared with the system, and on the permissions to access content shared by other users. This is where DECODE has the potential to provide the technological layer needed to make a step forward with respect to the existing paradigms of data silos and data extractivism, and to turn the vision of data as commons into a reality with benefits for citizens at both the individual and collective level.

In this way, Barcelona Now will represent a privileged interface to interact with the pilots, and to demonstrate the possibilities offered by the DECODE infrastructure.

References

Aguilera, U., Peña, O., Belmonte, O., and López, D. (2016). Citizen-centric data services for smarter cities. Future Generation Computer Systems.

Bibri, S. E. and Krogstie, J. (2017). Smart sustainable cities of the future: An extensive interdisciplinary literature review. Sustainable Cities and Society.

Cheng, B., Longo, S., Cirillo, F., Bauer, M., and Kovacs, E. (2015). Building a big data platform for smart cities: Experience and lessons from santander. In Proceedings of BigData Congress. IEEE.

Chilipirea, C., Petre, A.-C., Groza, L.M., Dobre, C., and Pop, F. (2017). An integrate architecture for future studies in data processing for smart cities. Microprocessors and Microsystems.

Gharaibeh, A., Salahuddin, M. A., Hussini, S. J., Khreishah, A., Khalil, I., Guizani, M., and Al-Fuqaha, A. (2017). Smart cities: A survey on data management, security, and enabling technologies. IEEE Communications Surveys & Tutorials, 19(4):2456–2501.

Gong, Y., Morandini, L., and Sinnott, R. O. (2017). The design and benchmarking of a cloud-based platform for processing and visualization of traffic data. In Proceedings of BigComp. IEEE.

Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., and Banchs, R. (2010). Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. Pervasive and Mobile Computing, 6(4):455–466.

Kitchin, R. (2014). The real-time city? big data and smart urbanism. GeoJournal, 79(1):1–14.

Marras, M., Manca, M., Boratto, L., Fenu, G., and Laniado, D. (2018). BarcelonaNow: Empowering Citizens with Interactive Dashboards for Urban Data Exploration. In Proceedings of the Web Conference (WWW'18). Forthcoming.

Massana, J., Pous, C., Burgas, L., Melendez, J., and Colomer, J. (2017). Identifying services for short-term load forecasting using data driven models in a smart city platform. Sustainable Cities and Society, 28:108–117.

Navarro, J. M., Tomas-Gabarron, J., and Escolano, J. (2017). A big data framework for urban noise analysis and management in smart cities. Acta Acustica united with Acustica, 103(4):552–560.

Paternò, F. and Wulf, V. (2017). New perspectives in end-user development.

Zdraveski, V., Mishev, K., Trajanov, D., and Kocarev, L. (2017). Iso-standardized smart city platform architecture and dashboard. IEEE Pervasive Computing, 16(2):35–43.