A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring

A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring

Federico Montori, Luca Bedogni, Luciano Bononi
Department of Computer Science and Engineering (DISI)
University of Bologna, Italy
Email: {federico.montori2, luca.bedogni4, luciano.bononi}@unibo.it

***Abstract*—The Collaborative Internet of Things (C-IoT) is an emerging paradigm that involves many communities with the idea of cooperating in data gathering and service sharing. Many fields of application, such as Smart Cities and environmental monitoring, use the concept of crowdsensing in order to produce the amount of data that such IoT scenarios need in order to be pervasive. In our paper we introduce an architecture, namely SenSquare, able to handle both the heterogeneous data sources coming from open IoT platform and crowdsensing campaigns, and display a unified access to users. We inspect all the facets of such a complex system, spanning over issues of different nature: we deal with heterogeneous data classification, Mobile Crowdsensing (MCS) management for environmental data, information representation and unification, IoT service composition and deployment. We detail our proposed solution in dealing with such tasks and present possible methods for meeting open challenges. Finally, we demonstrate the capabilities of SenSquare through both a mobile and a desktop client.**

I. Introduction

The Internet of Things (IoT) paradigm is projected to have a huge impact on the life of human beings. Reports show the huge increase in the number of devices per person, projecting a skyrocketing 50 billions by 2020, which translates in more than 6 devices per person, on the average [1]. This is mainly devoted to the price reduction for the devices, which makes them more accessible for everyone, and the increasing number of services that these devices bring to the users in collective realities [2].

A. Motivation

Often, each IoT manufacturer provides its own cloud, its own services, generating IoT-based islands, or the so-called "Intranet of Things". This, in turn, forces the end user to use his or her own devices only in a manner that is foreseen by the manufacturer, without the possibility of exploiting new services that could offspring when relying on the data offered by heterogeneous devices. Overcoming such limitations is a hard task for people with networking and computer programming skills, nearly impossible for people without such capabilities. Still, this is an important feature of the future IoT, as users should be able to customize their own services by making sense of the different available data streams rather than being stuck with what has been devised by the producer. This is particularly true when considering collaborative scenarios that need individuals to work together as a crowd.

Sensing the environment and detecting potential hazards for citizens and nature has always been a crucial point of interest in environmental sciences. In particular, the field that matches with our research falls into the category of Community-Based Monitoring (CBM). Such paradigm is defined as a *"process where concerned citizens, government agencies, industry, academia, community groups, and local institutions collaborate to monitor, track and respond to issues of common community environmental concern*" [3]. Environmental CBM exists and has been deployed in several projects and it is categorized on top of both the capabilities and the awareness that are granted to participants [4]. This trend together with the impressive growth of IoT technologies for the environmental monitoring, led to the concept of Collaborative IoT (C-IoT). It is generally defined as a paradigm that breaks the aforementioned silos existing nowadays in vertical markets by enabling communication and interoperability across humans, enterprises and governmental entities [5]. Indeed, such concept is still poorly investigated in literature and existing works often focus on one or few of its aspects without taking into account the big picture.

B. Problem statement

The main concept upon which our work is founded is the capability of the end user to be a necessary part of the data gathering instead of a mere service consumer. Crowdsensing is the enabling concept for such paradigm; according to it, users can actively report data that can ideally be used either by them or by others, with a high spatial coverage and without the need for deploying monitoring sensor networks directly in the field. Moreover, as most of the sensors are already available in modern smartphones, every user can contribute in the data gathering even without owning dedicated sensors. This solution, known as Mobile Crowdsensing (MCS), has the advantage of being cost effective, while also providing real time data.

Crowdsensing brings along several research challenges, among which we concetrate on data reliability, in terms of both availability and correctness. Sensed data and environmental information is intended to be the main component of context-aware systems [6], thus it is important to ensure that data is meaninful and available continuously. We can quantify the data relevance and the effectiveness of a solution through the concept of Quality of Context (QoC), introduced in [7].

QoC is defined using five fundamental metrics: the data *precision*, which is how faithfully data reports real conditions, the *probability of correctness* and the *trustworthiness*, which are related to the reliability of the data and its source, the *resolution*, which denotes the information granularity and the *up-to-dateness*, which stands for how fresh is the data. Hence, the objective of any data retrieval task for context-aware systems is to keep such indices at a valuable level, depending on the scenario and the target service. Although in this paper we focus only on data availability through merging of sources – thus data quality is not addressed here – it is important to keep an eye on all these concepts at once.

Moreover, we consider of paramount importance to bring the technology closer to the user, and to make it accessible and useful regardless of how the data is produced, where it is stored, and how it is handled. Users should be able to create their own services based on their needs through the combination of available raw data streams. We believe that customized and ad-hoc aggregated services are a necessary point of arrival of massive data gathering, in such a way that end users can get the maximum outcome from the whole available knowledge. Furthermore, in order to better provide a broad spectrum of choices and possibilities, stakeholders may appeal to user-aided campaigns in which data gathering coexists with the users' proposals.

C. Proposal

In this paper we propose SenSquare, an architecture which fills the gap about dedicated services, by integrating heterogeneous data sources together, and making them available to the end user with an accessible interface. The heterogeneous data streams, usually not well-formed, are classified through Machine Learning techniques, and homogenized in different data classes. The users have then different possibilities to create their own services, through a web interface or through a mobile application. Several services templates are offered to the end user, who can choose to make use of them or start from scratch in creating his or her own template. Subsequently, such service is bound to a certain area of interest for each user, who will eventually be notified of updates concerning it. The data needed to fulfill the service requirements will either be gathered through reliable sources, if available, or through other unreliable resources. For instance, a template "heat index" service may require both a temperature value and a humidity value. An interested user can deploy such service in an area of interest, and the system will provide the data needed by fetching the available sources. Moreover, a user might be interested in enriching the heat index template service with an additional information, for instance the amount of airborne pollen due to allergies. Hence, he or she can extend the template service by simply adding the appropriate data and deploy it in the area of interest. In the end, services created by users are also available for the whole community, for which each member can take advantage of them or extend them further. We introduced for the first time SenSquare in [8], however we establish in this paper as further novelties the instantiation of customized and aggregated services, the use and classification of heterogeneous open data streams and the development of a mobile and a desktop application for the end users.

In this paper, we provide the following four novel contributions:

- We present the details of SenSquare, a C-IoT platform specifically for, but not limited to, CBM and environmental monitoring.
- We present the results of a user survey within the scope of Crowdsensing acceptance.
- We present an algorithm able to classify public available data streams, by means of Natural Language Processing (NLP) techniques, and machine learning.
- Finally, we describe the implementation of the SenSquare platform, along with the possibility for the users to compose their own IoT services through dedicated applications.

The remainder of this paper is structured as follows: Section II presents related work from literature; Section III motivates our study, by presenting the results of a survey which we conducted to assess the viability of our proposal by introducing a crowdsensing campaign; Section IV presents our architecture, detailing all of its components; in Section V we present our algorithm for the classification of heterogeneous data streams; Section VI highlights the different methods the users have to interact with SenSquare to create their own services; Finally, Section VII concludes this paper by summarizing our work.

II. Related Works

In this section, we review works from literature about the different aspects that fall within this vast area, which are *CBM*, *Existing platforms*, *Data Collection*, and *Mobile CrowdSensing (MCS)*.

A. Community-Based Monitoring (CBM)

The literature about human collaborative actions taken upon CBM and IoT concepts is vast, indeed CBM can also be activated in different guises. More in detail, We refer to "consultative CBM" whenever citizens are participating in collecting data and measurements without being necessarily involved in observing the results neither in decisions taken upon them. We name as "collaborative CBM" the paradigm in which participants are still the primary source of information, however they can get access to the outcomes and can take decisions on future directions. Collaborative CBM presents a more complex structure of user pool – for instance, it can include citizens, stakeholders, producers and consumers – and can be further categorized. As an example, it can be pushed to "transformative CBM", in which the actual demand and the goals of each campaign come directly from the end users, the citizens in most cases. Hence, it is clear how consultative CBM, being driven by the government or a certified institution, has a clear goal and is able to provide long-term datasets. Nevertheless, it is dramatically linked both to the issuer's resources and to appropriate incentive techniques. For such reason, incentive techniques are considered to have a crucial influence on their penetration, indeed several incentive

methods, from monetary to gamification-based ones have been extensively addressed in literature [9]. On the other hand, collaborative CBM presents an intrinsic advantage for the participant, thus it needs less explicit incentives to reach a satisfactory coverage. However, the power given to both malign and inexpert users might be dangerous for the data credibility [10]. An example of one of such campaigns is given by the Louisiana Bucket Brigade [11], an environmental health and justice organization collecting participants' reports and initiatives concerning petrochemical pollution through eye-witnesses.

B. Existing platforms

On the technological side, as stated in Section I, we are facing a situation in which IoT technologies are mainly exploited in isolated architectures, designed as close systems and often not compliant with the outer world [2]. Many IoT-based frameworks, commercial and not, are based on this concept. This is the case of Cumulocity [12], AllJoyn [13], Xively [14] and ThingWorx [15]. Such solutions are powerful when managing all the entities implemented within them, however they behave as IoT islands which have little or no interoperability with others. This can lead to data redundancy and unavailability when dealing with environmental monitoring, i.e. when data is interesting for the common benefit. Moreover, they also bind the user with a specific manufacturer, which might not provide all the devices he or she needs, therefore limiting the potential of such infrastructure. Indeed, many unifying architectures were proposed in scientific literature, trying to overcome such problem and trying to present a layer for integrating heterogeneous IoT networks into one, scalable and distributed, IoT global system. One of the most notable effort is given in [16], in which the authors envision a unique, Internet-like, architecture for the IoT, presenting an accurate three-layered module, in which things are abstracted, semantically annotated and virtualized. The author's approach aims at generating automatically IoT services for end users, thanks to a reasoning engine that processes historical data and user preferences as well as sensor perception and neighbor discovery. Such work, together with many others that can be found in literature, are certainly valuable, however they do not assume that IoT systems' owners and manufacturers are driven by different interests and may not be willing to share their data. In our work, we chose as a baseline to have a strong data pool, which we gather from open and governmental sources, in order to reduce the risk of lack of participants. Such issue is addressed in the Sensing as a Service (SaaS) model [17], which analyzes social and economical perspectives of today's and tomorrow's Smart Cities and defines a Sensor Publishers layer of separate business entities devoted to deal with sensor owners, an approach slightly different from ours. Regardless of the differences, such work, together with the notable one provided in [18], present undeniably valuable guidelines for the deployment of smart services in Smart Cities. The latter provides also a proof of concept through the installation of an IoT island in a urban scenario.

C. Data Collection

The data collection regarding the primary environmental measurements, such as temperature, humidity, light intensity, noise, pressure, wind strength and many others, is currently considered an easy and inexpensive task. For this reason, crowdsensing, intended as the community-based data collection through either embedded or general-purpose devices, has been found to be the basis for many research activities and projects. Heterogeneous data collected from either users or open data sources often has the drawback of being unlabeled and sparse and, therefore, its meaning is hardly intelligible. For such reasons, a data integration and classification layer is necessary in order to understand the semantics of the data collected. The most related reference that deals with the classification of heterogeneous data streams coming from publicly available and potentially unreliable sources is the work carried out in [19]. The authors extract user-annotated sensed data from a public platform, however, differently from our work, they infer the trustworthiness and reliability of such values in relation with a reference value, which is necessarily taken from a certified source (e.g. the well-known Forecast.io, now replaced by DarkSky[1] for the weather data). On the one hand this is an efficient classification solution, taking into account the measurements instead of the annotations, on the other hand it limits the classified data streams to only the ones for which a certified value is retrievable. It is also worth noting how certified values are given at a wide area granularity (often per-city), which, for some types of data, might be inaccurate. Examples include the noise level, which varies dramatically when the measurement is taken close to a highly crowded street or in a house backyard, or the temperature, which drops in parks and rises in congested roads due to cars. In general, heterogeneous data classification is a strongly widespread research area, which has been studied over decades by many researchers and typically algorithms are modeled over specific data sets, while they perform badly over others (this is known as Wolpert's *no free lunch theorem* [20]). Nevertheless, it is worth mentioning some recent research efforts such as the one carried out in [21], in which a genetic algorithm that dynamically selects a combination of well-known classification algorithms is proposed. We also experienced a wide use of clustering algorithms for class inference in heterogeneous datasets, such as the one presented in [22] for land use tagging. Clustering algorithms are seldom used when a large volume of manually annotated data is available, since they rely on unsupervised or semi-supervised bases, and are less application-specific.

D. Mobile CrowdSensing (MCS)

An interesting development of the C-IoT is MCS, which transposes the concept of crowdsensing in an ecosystem where smartphones and portable objects are committed to report phenomena of common interest through environmental sensing or manually produced data [23]. MCS can be applied through participatory sensing, in which users manually report

[1] https://darksky.net/

occurrences of certain phenomena, useful when such data require a significant human interpretation, and opportunistic sensing, in which an application reports periodically sensed data without the need of human interaction. In both cases, crowdsensing campaigns need to deal with the citizen participation, normally fostered through user motivation, low quality data and location awareness data mining, which are both challenging and currently studied problems [24]. The clear advantage of MCS is the huge amount of data samples that can be gathered due to the paramount spread of mobile general-purpose devices, which grants a large spatial and temporal coverage and permits to observe a phenomenon through a significant number of different measurements [25]. MCS has been widespread in a paradigm in which requesting users generate tasks and responding user can accept and execute them; one of the most famous implementation of such paradigm is given by Medusa [26], which provides an ad-hoc programming language for non-expert users for the task generation. Although task generation and acceptance is the most widely used standard for MCS, we propose a different approach, based on a totally opportunistic basis, as explained in [8]. A plethora of applications making use of the smartphones' sensors have been proposed in literature. This is the example of SecondNose [27], which collects environmental data in order to infer a number of indices concerning air quality and pollution. It is also integrated with specific portable multi-sensors in order to enlarge the number of detectable pollutants (such as benzene). Other applications make use of the microphone in order to keep track of the noise levels in different areas of a city, exploiting the concept of Mobile Learning (ML) [28], assigning the measuring task to a dedicated class of citizens (e.g. the bicycle couriers [29]), or focusing on a particular source of noise pollution (e.g. the traffic on highways [30]). MCS is also imposing itself as one of the main actors within the scope of the smart cities, taking part in the concepts of the smart parking – using a number of sensors in order to detect either when the user is parking [31] or the empty parking spots around the user [32] – of the activity recognition in transportation mode [33], of the emergency management [34], of the city mapping [35]. Many MCS applications make use of the GPS geo-fencing [36] in order to limit zones of interests [37].

Given the current status of C-IoT and MCS, we observed how most of the application relying upon such concepts are commonly driven by specific campaigns that focus on a tiny portion of the aspects they can cover. To our knowledge, our work is the first attempt proposing a global platform able to cope with heterogeneous data coming from different available sources for environmental monitoring.

TABLE I
SUMMARY OF ALL THE ALTERNATIVE FRAMEWORKS PRESENTED IN SECTION II.

[11]	CBM	Louisiana Bucket Brigade, platform collecting observations about petrochemical pollution.
[12]	Middleware	Cumulocity, REST-based commercial IoT framework
[13]	Middleware	AllJoyn, commercial IoT framework for home automation implementing a software bus.
[14]	Middleware	Xively, framework for IoT automation implementing a message bus, APIs and an open data repository.
[15]	Middleware	ThingWorx, transport agnostic IoT commerical framework for home automation.
[16]	Middleware	Three-layered architecture for IoT, promoting enrollment of stakeholders.
[17]	Middleware	Sensing as a Service, paradigm for the interoperable IoT promoting stakeholders recruitment.
[18]	Middleware	Padova Smart City, deployment of smart grid and healthcare services through WSNs.
[19]	Classification	Open data reliability analysis, based on comparation with ground truth data.
[26]	MCS	Medusa, a programming language for task generation in MCS.
[27]	MCS	SecondNose, collection of environmental data for air quality.
[28]	MCS	Monitoring noise pollution through Mobile Learning.
[29]	MCS	Monitoring noise pollution through bycicle couriers.
[30]	MCS	Focusing the monitoring of noise pollution on dedicated places such as highways.
[31]	MCS	Collaborative smart parking though activity recognition.
[32]	MCS	Collaborative smart parking through empty spot detection using the magnetometer.
[37]	MCS	MoST, platform for activity recognition and geofencing.

TABLE II
TABLE SHOWING COLLECTED DEMOGRAPHIC DATA ABOUT THE INTERVIEWED PEOPLE.

Male	65.3%
Female	34.7%
Age 18-25	31.7%
Age 26-35	53.5%
Age 36-45	5.9%
Age 46-55	4.0%
Age 56-65	5.0%
Living in city (or town) center	57.4%
Living in the first outskirts	16.8%
Living in the periphery	13.9%
Living in the countryside	11.9%
Ownership of the roof	48.5%
Ownership of the garden	44.6%

III. USER SURVEY

In this section we present the results of a survey we conducted in order to assess whether users are willing to participate in a data gathering campaign hosting one or more crowdsensing elements in their everyday life. More in detail, we proposed to the users two different ways of participating. First, we proposed to the participants to host a small multi-sensor device, acting as a weather station, in an outer part of their house (e.g. the rooftop, the windowsill, the balcony or the garden, if present). The device is embedded in a small box not bigger than a 5cm-sided cube and hosts sensors for measuring temperature, humidity, pressure and environmental noise level. To report the data, the weather station has to be connected to the Internet, therefore we asked the participant to share their Wi-Fi connection with such devices. As an alternative, the device should report the data either through cellular connection or through some other long-range technology, e.g. LoRa, rather than Wi-Fi, resulting in an increased cost for the device distributor. We still consider such approach

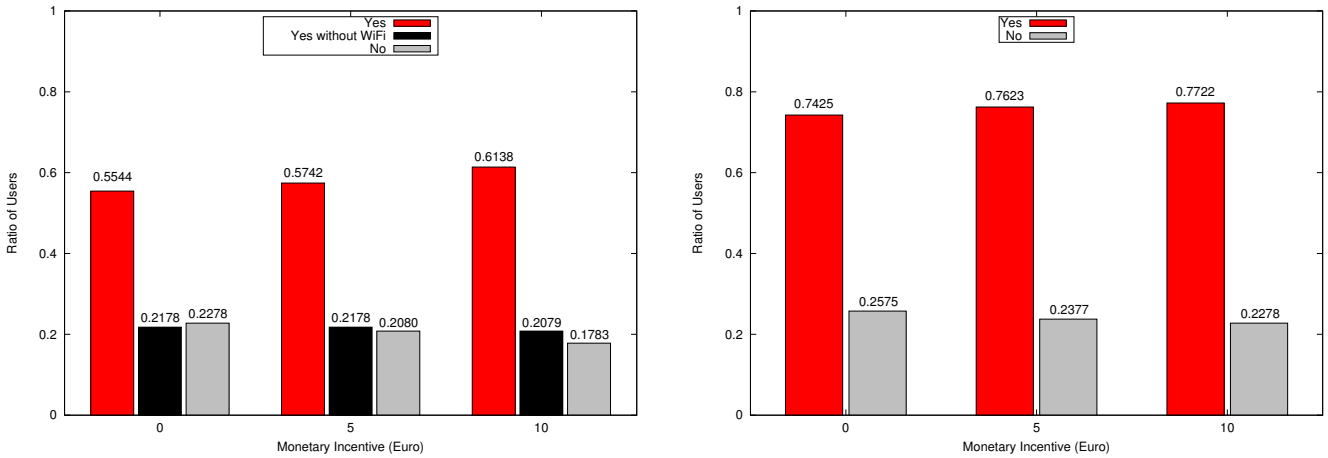


Fig. 1. The results of the survey. Figure 1(a) shows the results concerning the weather station to be installed at home, while Figure 1(b) shows the results for the smartphone APP.

as crowdsensing, since, even though users do not materially own the appliance, they have complete control on it. Second, we asked the participants about their willingness to install a mobile application in their personal smartphones, which runs in background and reports periodically sensed data to a central entity. For both installments we assure that the participant will get a personal consumer application able to monitor the data that their device, either the smartphone or the weather station, is sending to the remote platform.

We surveyed personally more than 100 individuals, all of them living within the Italian region Emilia-Romagna, which counts 9 different provinces and around 348 different municipalities. As a matter of fact, we do not intend to provide statistics over the general user acceptance of a crowdsensing paradigm, rather we wanted to prove that the population of a sample region tends to be positive towards an environmental crowdsensing campaign. The user survey involved human beings of different ages, both females and males, and it is organized in three main sections: (A) The first section is about some general questions about the users participating in the survey, which made possible to report the participants demographics, outlined in Table II. Such questions concern their age, their gender, in which context they live and the ownership of outer parts of their house. (B) We asked the participant whether he or she is willing to install the above mentioned weather station in the outer parts of his or her house and report the data to our central database. The user can answer with a plain "Yes", a plain "No" or "Yes (without sharing the Wi-Fi)". Should the user select the plain "Yes", the survey skips to section C, otherwise the user is offered to answer the same question including a monetary reward of 5 Euro per month and, would he or she answer neither in this case with a plain "Yes", the reward is increased to 10 Euro per month. After such proposal, regardless of the answer, the survey continues with the subsequent section. (C) Here, the user is asked about his or her willingness to install the previously mentioned mobile application – in this case the only possible answers are "Yes" and "No". The flow is similar to the previous one, thus, if the user is not willing to install it, a monthly reward of 5 Euro is proposed and it is increased to 10 Euro in case of another negative answer. Both in sections B and C, if the user states to be willing to participate only in exchange of a monetary reward, he or she is asked whether is willing to accept such reward supplied in the form of a discount or an offer regarding a particular class of stakeholders (e.g. a discount for train tickets or for mobile phone costs).

Focusing on Figure 1(a), we see that more than the half of the users is willing to install the reporting weather station and share the Wi-Fi connection. Interestingly, the number of users who either do not want to install it or simply to share the Wi-Fi is more or less constant regardless of the reward entity. Figure 1(a) reports the results of the survey concerning the installation of the mobile application on the participant's smartphone. The behavior is similar to the previous case, except for the fact that in this case we did not give the possibility to install the app without sharing a connection, which is needed in order to report the data. Nearly 80% of the participants replied that they would install the crowdsensing application on their smartphone. Again, similarly to the previous case, increasing the reward does not have the desired effect of increasing the number of positive users as well. Instead, their percentage remains more or less constant regardless of the reward entity. An interesting aspect which emerges by comparing Figure 1(a) and Figure 1(b) is the fact that the users are more willing to install a mobile application reporting data rather than trusting the installation of a third-party device in their houses. Moreover, it is interesting to note that 73% of the participants who replied "No" or "Yes without WiFi" to the installation of the weather station for a 10 Euro per-month, answered positively to the installation of the mobile application question without reward.

Finally, the vast majority of users requiring a reward surprisingly accepted the alternative form of reward that we proposed. This suggests that a number of stakeholders, spanning from telecom companies to transportation companies to municipalities, are potentially motivated to take active part in such campaign as distributors. Indeed, a vast amount of meaningful sensed data is a powerful source of knowledge that can

help such stakeholders in planning and decision making. An example of such alternative form of reward has been adopted in the crowdsensing campaign issued by Doxa, an Italian institution for market researches. In particular, they proposed to the users to install a mobile application called *DoxaMeter*[2] which monitors the cellular connectivity and offers a monthly discount of 5 Euros for purchases on amazon.it.

IV. SenSquare

In this section we focus on the contributions brought by SenSquare, the interoperable data platform upon which the experiments and algorithms presented within this paper are carried out. SenSquare has been recently proposed as a unique architecture for an ecosystem relying on Crowdsourced data capable of providing aggregated services for the common benefit [8]. Several IoT-based data platforms are currently establishing as isolated systems, unable to communicate or share information with each other, therefore we are facing data redundancy, produced by different sources often for the same purpose. In order to overcome this issue, SenSquare is currently developed within the scope of a unifying architecture, able to host data coming from different realities and standardize the way in which data is presented. In our case, we integrated data coming from official and governmental sources, open data sources and Crowdsensed sources, for which we will outline differences and details in Section IV-A.

In general data is structured in "Data Streams" (DS), which are instantiated with the static meta information about the type of raw data provided and are periodically updated by the data source with new measurements. Each new measurement necessarily comes with a spatial data (GPS coordinates) and temporal data (timestamp). The source, especially in case of an end user contributing to the measurements by means of a Crowdsensed approach, might be adhering to a campaign which establishes some sort of reward related to the measurements provided. In such cases, the source is typically not providing data at a constant rate, indeed, each new data update provided is coupled with a response, which contains a configuration stating when or where to provide the next update. Such decision is taken according to a set of rules, which are established by the industrial or governmental stakeholders that are interested in the Crowdsensed data. The algorithm behind this mechanism is described in [8] and summarized in Section IV-B, while the architecture of the platform and its components are outlined in detail in Section IV-C.

A. Data Sources

The main contribution of SenSquare is given by its common approach in order to cope with different sources. In particular, it combines results with a wide variety of different characteristics that span over low or high reliability and low or high update rate. In this section we summarize the sources that we considered in the current state of the implementation of the system.

We label as *Reliable Resources* those coming from governmental or official institutions, for which environmental sensing is the main business focus. They are considered reliable for several reasons that include the high precision of their appliances and the guarantee of their perseverance in providing information. This may include entities such as the Environmental Protection Agency (EPA)[3], which is the institution for environmental monitoring in the United States. In our case we took into account the Regional Agency for the Environmental Protection in Italy (ARPA), the public administrative agency for the environmental monitoring. Given its regional management, the task of unifying data coming from all parts of Italy is particularly hard. In fact, each of the 20 Italian regions implements its own version of the data platform, thus, in order to retrieve the data, a per-region ad-hoc parser and retriever has to be implemented.

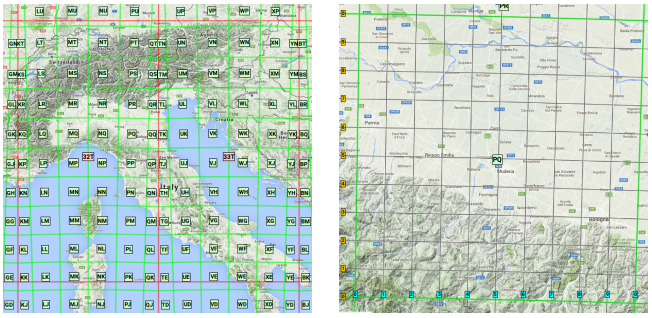
We label as *Unreliable Resources* those coming from open data platform, on which normally users can upload their data as open data channels for their personal purposes. An example is given by OpenSignal[4], a crowdsourced coverage map of 3G and 4G base stations. Our previous work focuses on data unification coming from ThingSpeak and SparkFun, two well-known open data platforms, from which we extracted more than 100,000 public data channels [2]. Such data is considered unreliable as there is no guarantee about its veracity neither about what it actually measures. Indeed, data streams within each channel are not labeled with a defined data class, which has to be inferred upon the name given to the streams themselves. This opens up several issues, for instance we could possibly exclude valuable results due to their bad labeling, i.e. a temperature value could be named with a pointless name and thus not classified as meaningful. On the other hand, we could even include measurements that are not valuable for our system. One example is given by the vast amount of values of temperature taken from indoor environments or from objects. These are possibly categorized as temperature values, however, since they are not reporting values related to the common environment, they are not interesting and can be damaging when combined with other streams, for example when calculating a local mean. In order to assign a data class to data streams extracted from Unreliable Resources, we designed a classification algorithm, as it is explained in Section V. On the other hand, Unreliable resources have the advantage of being a significant amount and their number is in constant growth [2], making a worldwide coverage possible. On top of these premises, we expect the inference of their veracity to be feasible. This can be achieved, for each measurement, through machine learning algorithms taking into account neighboring measurements about the same class. Furthermore, Unreliable Resources tend to have a significantly higher update rate, providing information with a satisfactorily high temporal granularity.

Finally, we consider *Crowdsensed Resources* to be the main contribution to our system, mainly because they constitute the set of resources that our platform is able to control. Our Mobile Crowd Sensing (MCS) ecosystem has been already covered and explained in detail in [8] and will be summarized

[2]http://doxameter.it/

[3]https://www3.epa.gov/

[4]https://opensignal.com/



(a) (b)

Fig. 2. MGRS encoding of Northern Italy. Figure 2(a) shows the subdivision of GZIs in 100 km sided squares, while Figure 2(b) shows the subdivision of a 100 km sided square in a 10×10 grid.

in Section IV-B. It represents a system currently implemented as a part of our platform and referring to a mobile application as its client. Such system has been shown to be useful and respecting automatically the constraints imposed by our server, resulting in a reduction of data redundancy. Furthermore, in Section I we described an example device that takes part in our campaign and falls into the category of Crowdsensing Resources. It is worth mentioning that our Crowdsensing paradigm provides user privacy through anonymity: whenever participants' devices submit a set of measurements, the user's identifier is not saved, thus there is no plain way to infer when two measurements come from the same user, neither to track the user's path. Thus, our crowdsensing paradigm can be seen as non-intrusive, at least following the guidelines considered a common practice in recent EU projects, such as Preciosa[5] and PARIS[6]

B. Spatial Encoding and Constraints

SenSquare is based on a centralization of the decisional and computational capabilities, in order to make the data providers and consumers, the "satellites" of the system, as simple as possible. As stated previously, a subset of such satellites might be attending a campaign instantiated by a number of stakeholders. Such entities are typically assuring some kind of reward in exchange for the measurements taken. Several studies have been centered on how to convince end users in contributing to such campaigns [9], some of them suggest a monetary reward, some others rely on special offers by the stakeholders themselves, gamification [38] has been taken into account as well. The survey we proposed, already covered in Section I, gave us valuable suggestions on how to cope with this aspect.

In order to deal with such paradigm, for each measurement uploaded by one of the satellites, our system replies with a configuration, which is a set of temporal and spatial constraints. A temporal constraint is a timeout that establishes when the next measurement has to be uploaded. It is calculated on top of the data type and it is specified on a per-zone basis. A spatial constraint is encoded as a square zone that limits the validity of the data and it is different for each data type and, as for the time constraint, specified on a per-zone basis (which has to be equal or bigger than the square zone delimiting the validity). Every time a data producer exits the defined zone the measurement needs to be updated, as a consequence the spatial constraint is typically only for mobile producers. We chose Military Grid Reference System (MGRS) [39] as an encoding scheme for square zones since it natively supports hierarchy. More in detail, MGRS divides the world in 6° by 8° rectangular geographic areas (with some exceptions at the poles due to the spherical shape of the globe) and labels them with an unique identifier called Grid Zone Identifier (GZI). Each of such areas is subdivided in 100 km sided squares, which are in turn hierarchically subdivided in 100×100 smaller squares until a precision of 1 meter is reached. In such way each GZI is followed by two 5-digits coordinates representing the x and the y of a 1 meter sided square. Such coordinates can be masked in order to represent the "parent" of a zone, meaning that a different number of digits for the coordinates (from 0 to 5) identifies squares of different magnitudes. For a graphical feedback, Figure 2 gives the MGRS representation of Northern Italy at different granularities.

As an example $32T\ PQ\ 5731\ 2957$ identifies a 10 meters sided square located in GZI $32T$, while $32T\ PQ\ 57\ 29$ identifies a 1 km sided square. The interested reader can refer to [8] for further and more specific details on spatial encoding and constraints.

C. System Architecture

As stated previously, the architecture of SenSquare is structured as a client-server centralized system, in which no communication link is possible among clients. As can be seen in Figure 4, the central entity is the *Central Coordination Unit (CCU)*, which is separated in several modules, devoted to different tasks. Each module refers to the *Central Database (CDB)*, which is the entity collecting the raw data about all the relevant measurements, all the data streams, the rules, data about the users and the instantiated services. The structure of the CDB is outlined in Figure 3 with a relational diagram.

In particular, the system is characterized by two types of users: the stakeholders, which, as said, propose data collection campaigns, and the participants, which are users able to produce data and/or willing to consume data in the form of services. Each user is owning zero or more devices, which are the physical entities committed to sense the environment. Each device can produce one or more data streams, which are characterized by a single data type, marked with a pre-defined data class. Finally, each stream refers to a set of measurements, to which, for each update, a new measurement is added. A special case is given by the data channels retrieved from the open data platforms, which are considered as single devices although they are not necessarily physical devices or can refer to multiple ones. Each stakeholder can submit a number of rules, which may be referring to a particular zone, and, whenever a participant decides to attend to the campaign proposed by a particular stakeholder, its subscription is registered onto

[5]Preciosa Project FP7-ICT-2007-2, project n.224201
[6]PARIS Project FP7-SEC-2012-1, project n.312504

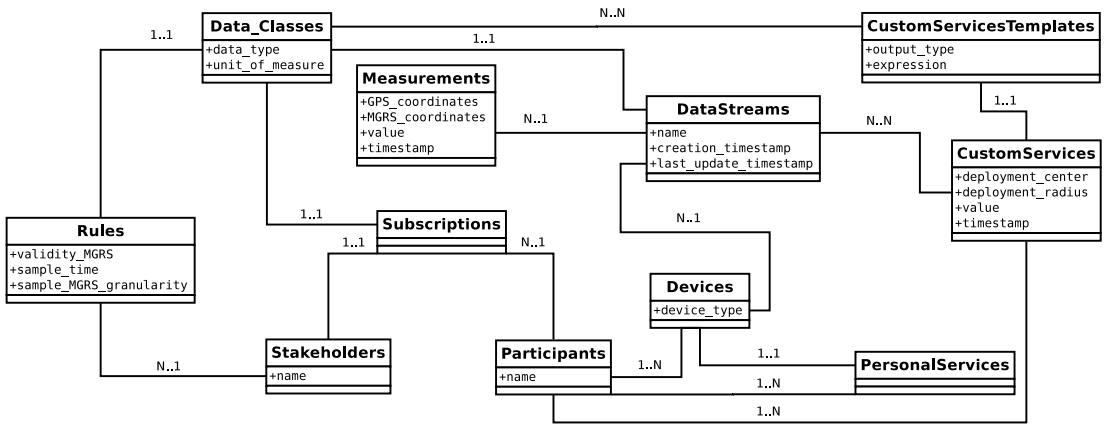


Fig. 3. Diagram outlining the relations between the database entities.

a specific table. Furthermore, raw data is commonly not made for being accessible to everyone as is. In particular, whenever a participant is willing to consume a particular stream of data, its request is conveyed through the instantiation of a service. The data aggregation is established upon the creation of custom services templates, which specify the type of data to be consumed, and instantiated in a certain zone as custom services. In addition, personal services are special instances of services monitoring a single device and available exclusively to the device owner. Services and templates are described more in detail in Section VI.

The *Data Retriever (DRet)* is the module committed to retrieve periodically data coming from the "static sources", which are both the governmental sources publishing in their own open repositories and the open data platforms from which we extract valuable and Crowdsensed data posted intentionally by end users. For each of such sources we implement different parsers in order to cope with the heterogeneity with which the data is presented. For further information, Section IV-A explains in detail which data sources we consider and how we retrieve data.

The *Data Classifier (DClass)* is a module committed to assign a data class to each data stream in which is not indicated. In fact, not all the data sources specify the exact type of data to which the measurements refer to, thus we need to infer it using an NLP-based approach. The data classification is described in detail in Section V.

Finally, the *Crowd Sensing Configurator (CSConf)* is a server committed to listen for possible Crowdsensing clients. It replies seamlessly to each data upload with a time and a zone constraint, as described in Section IV-B, that specifies when and/or where the next update should be provided by such client. Such directions may either be followed more or less strictly by the data provider or not followed at all. This might clearly determine whether the reward exposed by the stakeholder that requested such constraints shall be delivered or not.

V. Batch Classification

Giving homogeneity to data is not an easy task. In this section we describe an additional step that had to be taken whenever collecting data coming from the Unreliable Resources that we considered: ThingSpeak[7] and SparkFun[8]. ThingSpeak is an open source data platform launched in 2010 by ioBridge which provides a personal cloud for each user where it is possible to store measurements and offering a straightforward API as well as an analysis tool for an easy access to such information. SparkFun Electronics Ltd. was founded in 2003 and founds its main activity as a microcontroller seller and manufacturer for open source hardware. It also hosts an open source data cloud with functionalities similar to ThingSpeak's ones despite negligible differences. We demonstrated in [2] that the coexistence of both platforms is relevant since the distribution of the users subscribing to such platforms is geographically driven, i.e. ThingSpeak users are significantly more frequent in Europe and Asia, while SparkFun is more popular in the United States. We also showed that the data coming from such sources is characterized by a compatible structure and is easy to integrate. In particular, it is organized in "data channels", which contain one or more data streams relative to a different class of data each. All the data streams are updated at the same time with a single operation.

Such data is not uniquely labeled with a data class and information about the type of data provided is not guaranteed to be reliable. Indeed data fields are labeled totally upon the user's decision, thus they may result sometimes not even related with the data itself. In SenSquare we introduced the entity "Data Class", which is a 4-character tag that uniquely identifies the data type together with its unit of measure, as can be seen in Figure 3. Such tag, together with the geolocation, is considered mandatory for the purpose of interoperability and service composition, which are the fundamental concepts that characterize the ecosystem. While geolocation cannot be inferred when not indicated, data class can be retrieved from the name of the stream, as well as the tags, the description and the name assigned to the data channel. Within the scope of the project we designed a classification algorithm able to infer the data class of a consistent percentage of the data streams. The algorithm uses a supervised learning approach, which we tested on a set of 2000 data streams, randomly extracted from

[7]https://thingspeak.com/
[8]https://data.sparkfun.com/

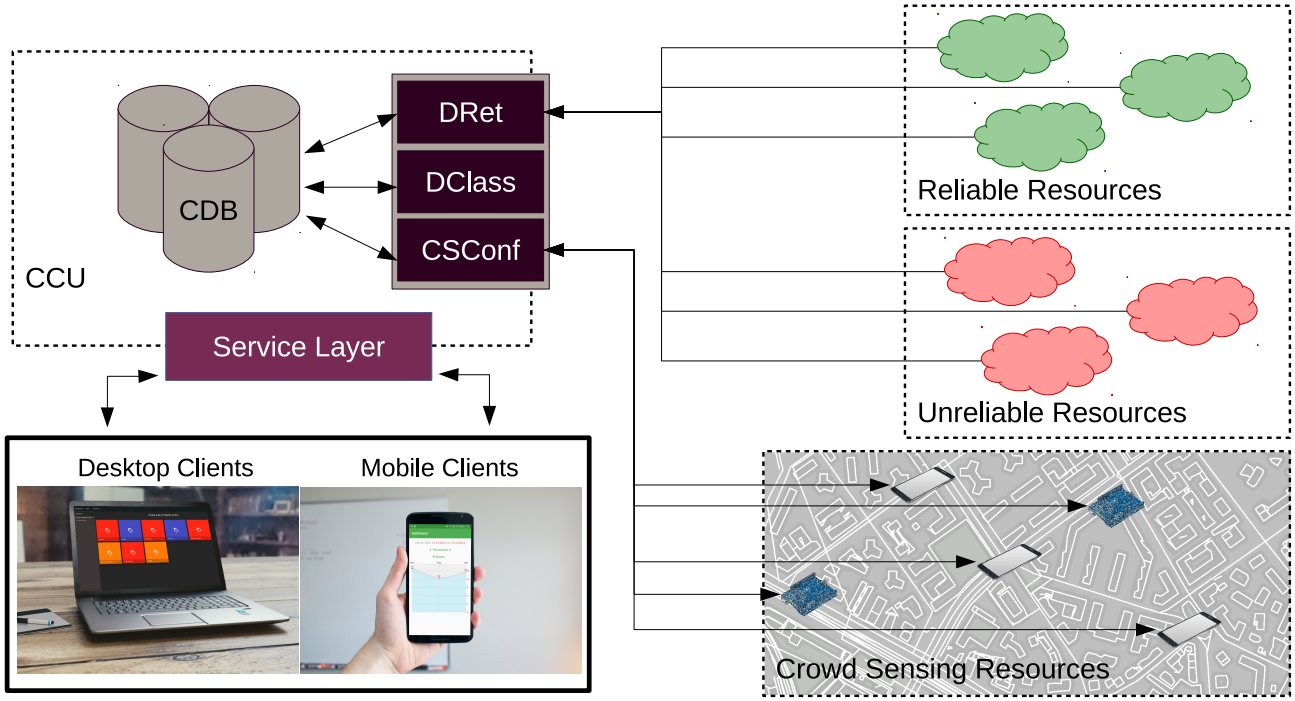


Fig. 4. Architecture of SenSquare, outlining all the different facets in which clients can contribute to data collection.

all the geolocated streams belonging to both the platforms. In order to perform the test we classified manually all the streams finding a set C of 33 different data classes. We also needed to exclude 800 data streams, for one or more of the following reasons: (i) their classification was considered too hard even for real people, for instance cases when the fields are named in a progressive way like *field1*, *field2* and so on and the channel description and tags are absent, (ii) the data or the stream label was considered meaningless, like in *motion*, *alarm*, *user* or when all measurements report a default value like 0, a typical behavior of test instances, (iii) the class was considered not pertinent to the purposes of our project (i.e. not related to environmental monitoring or not interesting for the society), for instance, streams reporting measurements of CPU usage. Hence, we ended up with 1200 consistent and manually annotated data streams, on which we built up our algorithm.

We chose to perform on our dataset DS a typical 10-fold cross validation [40], which has been widely accepted in literature over the years for the purpose of text classification in different application fields [41], [42], since it avoids loss of significant features when separating a dataset in training and test sets. As a consequence, for each iteration, we identified 120 streams as the test set Tst and the remainder was included in the training set $Train$. The algorithm works as follows: for each class $c \in C$, we initialized an empty dictionary D_c which we filled with $c(Train)$, i.e. the names of all streams in the training set belonging to such class. Then, for each stream name $s \in Tst$, we calculated its similarity with each class, selecting $distance(s, c) = min\{wd(s, t) \mid t \in D_c)\}$, that is, the minimum value of edit distance between s and all the terms in the dictionary for such class. For the purpose of calculating the edit distance, we used the Damerau-Levenshtein distance [43], after a text pre-processing in which we reduced every literal to lowercase and replaced all the separators with a unique identifier. We also observed that some streams have very short names, e.g. temperature is sometimes indicated only with "t" or "t1". For such reason we needed to normalize the edit distance calculation, i.e. we divided the Damerau-Levenshtein distance, which is by definition absolute, by the maximum value between the length of the two operands as:

$$wd(s,t) = \frac{DamerauLevenshtein(s,t)}{max(length(s), length(t))} \in [0,1]$$

Also, due to the extreme use of strong abbreviations in stream names, we chose not to use stemming. Afterwards, we assigned the stream s to the class c_0 such that $distance(s, c_0) = min\{distance(s, c) \mid c \in C\}$, that is, the class for which the minimum edit distance from s is the lowest among all classes. In order to obtain the metrics, we referred to the macro averaged F-measure ($MF\text{-}Meas$) [44] calculated as:

$$MF\text{-}Meas = \frac{2 \cdot MPrec \cdot MRec}{MPrec + MRec}$$

where $MPrec$ and $MRec$ are, respectively, the weighted averages of the precision and the recall calculated on top of the confusion matrix for each class. The weighted average is calculated by picking the actual number of observations belonging to each class as a normalization discriminant. Table III outlines the results we obtained from the cross-validation.

As it can be deduced from the table, our algorithm obtained a good baseline result. Due to the extremely heterogeneous variety of data streams, we still lack a way in which we can recognize the "garbage", i.e. all the data streams that are not suitable for the purposes of our system and thus we should

TABLE III
EVALUATION METRICS FROM THE DATA CROSS-VALIDATION.

	MPrec	**MRec**	**MF-Meas**
Mean	0.8883	0.8953	0.8918
Std. deviation	0.0317	0.0275	0.0290

TABLE IV
SYMBOLS USED IN THE ALGORITHM.

$Train$	Training set.
Tst	Test set.
D_i	Dictionary for the class i.
C	Set of all classes.
$name(t)$	Function extracting the name of the stream t.
$c_{real}(t)$	Real class to which stream t belongs, as annotated manually.
$c_{assigned}(t)$	Class to which our algorithm assigned the stream t as belonging to.
$wd(s,t)$	Our normalized edit distance between the strings s and t.
$distance(s,c)$	Minimum edit distance between the string s and all the strings in D_c.
TP_c	True positives with respect to class c.
FP_c	False positives with respect to class c.
FN_c	False negatives with respect to class c.
$MPrec$	Weighted average precision of the algorithm among all classes.
$MRec$	Weighted average recall of the algorithm among all classes.
$MF\text{-}Meas$	Harmonic mean of $MPrec$ and $MRec$.

not consider. We expect that, despite the algorithm can be improved to achieve better results, the classification should be partially user-aided in order to be precise enough. More precisely, in our experiments we encountered few anomalies that could suggest a reclassification in one way or another:

(A) In some cases there may exist $c_1 \neq c_0$ such that $distance(s, c_1) \approx distance(s, c_0)$ for a certain t_i, that is, two classes c_1 and c_0 for which an input t_i is suitable since the minimum word distance from them is equal or presents a negligible difference, which we established to be less than a parameter ϵ.

(B) In few cases $distance(s, c_0)$ is considered too high to be a determining condition for s to belong to c_0. In particular, such distance is higher than a defined threshold τ.

The fragment of pseudo-code in Algorithm 1 is intended to be explanatory for the sake of the current status of the classification algorithm. For the sake of clarity, we recall in Table IV all the symbols used in the algorithm.

Algorithm 1 Classify all the streams in the Test Set

```
# Fill the dictionary with the Training Set
for all t ∈ Train do
    i ⇐ c_real(t)
    append name(t) to D_i
end for

# Extract each stream to be classified from the Test Set
for all s ∈ Tst do

    # Calculate for each class the min word distance
    for all c ∈ C do
        distance(s, c) ⇐ min{wd(name(s), w) such that
            w ∈ D_c)}
    end for

    # Assign a class to the data stream
    compute c_assigned(s) such that
      distance(s, c_assigned(s)) = min{distance(s, c) such
        that c ∈ C}

    # Check if the assigned class is a true positive or not
    if c_real(s) = c_assigned(s) then
        increase the true positives (TP) of c_real(s)
    else
        increase the false negatives (FN) of c_real(s)
        increase the false positives (FP) of c_assigned(s)
    end if
end for

# Calculate precision and recall for each class
for all c ∈ C do
    precision_c ⇐ TP_c/(TP_c + FP_c)
    recall_c ⇐ TP_c/(TP_c + FN_c)
end for

# Extract final metrics as weighted averages
MPrec ⇐ weighted_avg{precision_c such that c ∈ C}
  with weight TP_c + FP_c
MRec ⇐ weighted_avg{recall_c such that c ∈ C}
  with weight TP_c + FP_c
MF-Meas ⇐ (2 * MPrec * MRec)/(MPrec + MRec)
```

VI. FRONT-END SERVICES

Data collected among the users are typically raw information about some defined place. By default, a participant should always be provided with the access to the measurements that his or her personal devices are producing. Most of the times, such measurements are concerning an environment of interest for such user, e.g. the place where he or she lives. This is the case of the weather station that we outlined in Section III, for which, as said, the access to the environmental data produced can be a valuable source of revenue. Such access is provided in the form of a service interface called Personal Service Instance (PSI), a simple view on the raw data related to the correspondent device. An implementation of it is outlined in Section VI-B in the form of a mobile application.

Besides the above mentioned considerations, with respect to the crowdsensing paradigm, our platform aims at providing all users with a global access to sensed data. For such reasons, we implemented a mechanism by means of which users can aggregate raw data streams and compose services, that can be exploited by other users too. We can simply think about all the well-known information that are obtainable by combining raw sensing measurement such as temperature and humidity. This is the case of the Heat Index, or humidex, which is a derived measurement calculated upon the values of temperature and humidity and it is commonly referred to

as the "human-perceived temperature". Another example is given by the Dew Point, which corresponds to the maximum temperature at which water vapor in the air will condense and form liquid dew. This is again dependent on the values of humidity and air temperature and can be calculated upon such measurements. Moreover, the definition of derived quantities can be extended to custom ones. For instance, within the scope of house automation, a participant might be interested in opening automatically the windows whenever the environmental temperature reaches a value over a certain threshold. However, at the same time, such participant might want to combine the value of temperature with some other due to certain requirements, e.g. he or she might be allergic to pollen, thus, if there is a high concentration of airborne pollen, the participant would rather use air conditioning. This approach is a simple example of data aggregation as a custom service that a user can create which, as a consequence, results in a combination of energy saving and safe health. In our proposed architecture, the service "aeration for pollen intolerant" is intended to be created only once and can be instantiated by several participants in different locations, provided that the right sensors are available in such places.

In order to give a more detailed shape to such definition, and to outline the components of our platforms devoted to such aspects, we define as Custom Service Template (CST) a combination of primary data classes through a mathematical expression. More in detail, a CST is characterized only by the output type, which can be either numeric (a floating-point number) or boolean, and the above mentioned mathematical expression E. Such expression is defined as:

$$E := c \mid DC \mid (E + E) \mid (E - E) \mid (E * E) \mid (E/E)$$

where c is a constant value and DC is a data class among the ones already defined in the CDB. The current implementation of the expression lacks some constructs, such as boolean operators and conditionals, which we aim to propose in a subsequent release of the system. Each CST is belonging to the participant who created it, although it is accessible to all the users and, upon creation, it is registered onto the CDB, as shown in Figure 3.

We then define as Custom Service Instance (CSI) an instantiation of a CST onto a specific region or area. A CSI must associate a unique instance of data stream for each of the data classes included in the expression belonging to the respective CST. In addition, all the selected data streams must be geolocated within a radius $r_{deployment}$ from the central point of instantiation $C_{deployment}$. Each CSI is registered onto the database, as shown in Figure 3, and belongs to the participant who instantiated it, which can be different from the owner of the respective CST. Furthermore, each CSI computes periodically the expression using the selected data streams and registers the output onto its database record. Although our implementation does not allow it yet, every CSI could be considered as a geolocated data stream itself, in order to permit its integration within another CST.

A. Web Interface

In this section we aim to show one real implementation of the service platform, developed as a RESTful Web application. Screenshots of the client interface are shown in Figure 5, on which we will go through with an example.

Figure 5(a) shows the interface with which the user can compose a CST by typing in a name, selecting an output type and following a guided procedure to compose the expression. In our specific case, the end user wishes to create a CST with which he or she can monitor the environmental conditions of interest in order to decide whether a jogging activity is not convenient due to a unhealthy combination of high humidity and high concentration of airborne PM10. As it is shown in the picture, the resulting expression is $100 - (PM10 * 0.84) - (Humidity * 0.63)$. With reference to our data class table, $Humidity$ is measured as a value between 0 and 100, while $PM10$, which stands for the concentration of coarse particulate matter (with a diameter between 2.5 and 10 μm) in the air, is measured in $\mu g/m^3$. The newly created CST, defined as *Jogging*, is intended to return a positive number if the conditions are considered acceptable, negative otherwise. Figure 5(b) shows the activity of a user attempting to instantiate the CST with the $C_{deployment}$ in the outskirts of the city of Bologna, Italy. The system automatically returns all the data streams measuring $Humidity$ and $PM10$ located within $r_{deployment}$ from it. The figure shows clearly two instances of Reliable data streams for the $PM10$ – in our case belonging to the institution ARPAE – and one instance of an Unreliable data source for the $Humidity$ – in our case coming from ThingSpeak – as spots in the map. In order to finalize the creation of a CSI, the user must specify one and only one data stream for each required data class. Figure 5(c) shows a view of the user interface over all the public CST, represented as colored square boxes. We remark that the CST can be made available by anyone, and displayed to any user. Hence, potential non-technical end users could simply instantiate an existing CST in the area of interest, without the need to build it from scratch. This brings more elasticity with respect to hyper-customized vertical solutions. On the left column a list of all the personal CSI is displayed (in our case it is showing the one created in Figure 5(b)). Figure 5(d) shows the detailed view over a personal CSI, including the location of all the monitored data streams in the map, their last update value and the last computed output value of the CSI itself.

B. Mobile Application and Widget

In this section we describe our Android mobile application, called Habitatest, which is composed by an Android activity to merge services together, and widgets to monitor the users' services.

In Figure 6(a) we show the main screen of the Habitatest app, where the user is able to select the services to monitor. The selection can be made either by inserting the ID string or by scanning the QR code which can be retrieved from the web service described in Section VI-A. The user can declare

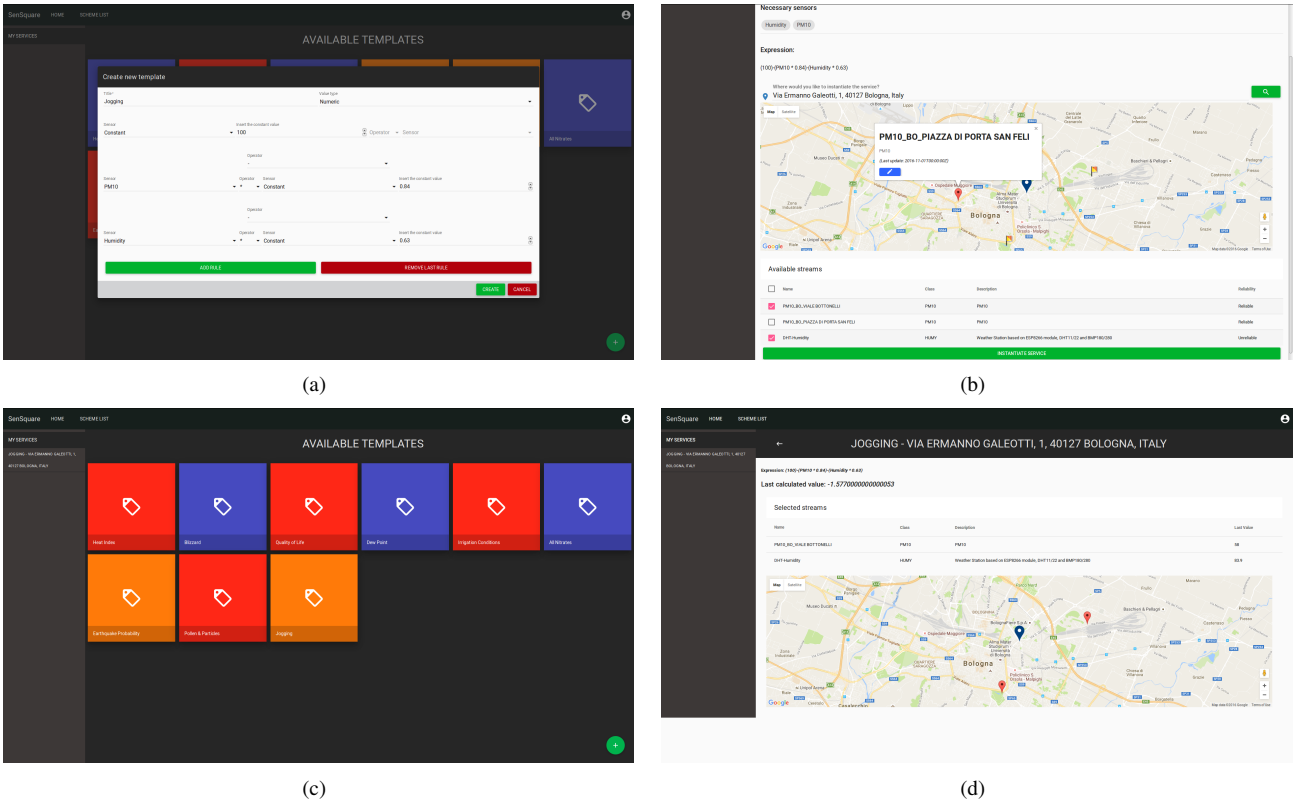


Fig. 5. Screenshots of the web interface for service creation and instantiation. Figure 5(a) shows the personalized service creation; Figure 5(b) presents the sensor to be choosen to instantiate a service; Figure 5(c) is the list of personalized services available to be instantiated, and Figure 5(d) shows th service running, displaying the desired value.

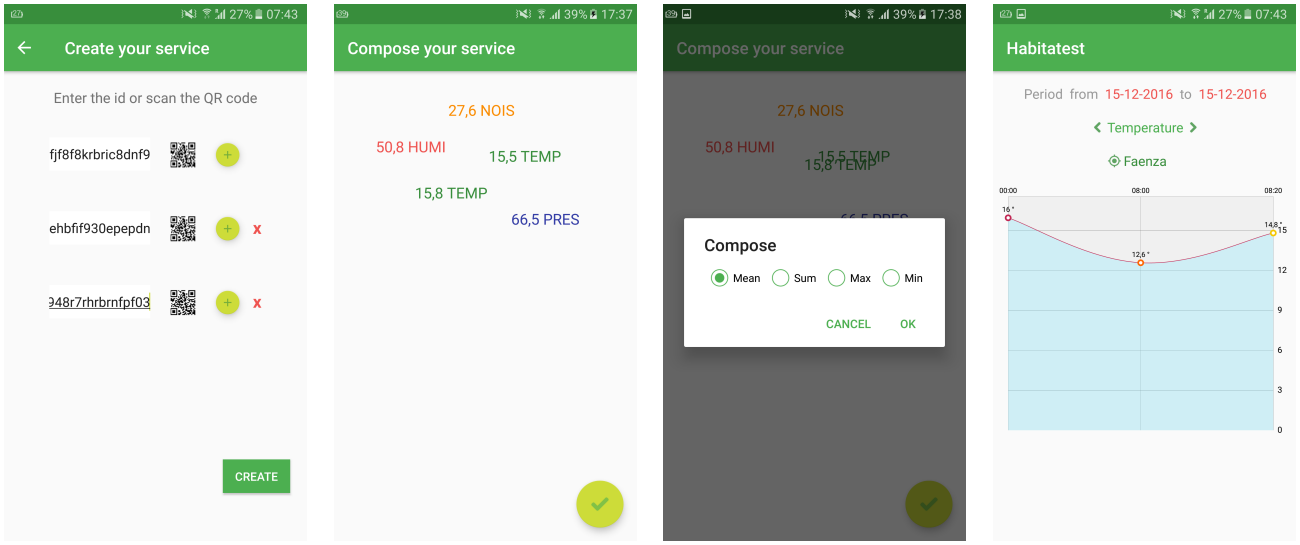


Fig. 6. The Habitatest APP. Figure 6(a) shows the service composition screen; Figure 6(b) is the one devoted to the same data classes merge, along with Figure 6(c) which is the merging method; finally, Figure 6(d) presents the charts about the desired service

any number of services of interest to be monitored through the Habitatest app and an update frequency. After having this step, the user is redirected to the activity we show in Figure 6(b), from which the user can select all the data is interested into. He or she can also merge together data of the same class through 4 different methods, which we show in Figure 6(c). The user can merge the data either by extracting the mean of the instances of the requested data class, their sum, the maximum or the minimum value.

The idea behind this choice is the fact that the user can select multiple services offering the same type of data, however only an aggregated value is given out of them. An example may be given by a number of temperature sensors installed close to the residence of the user, who, instead of selecting one or

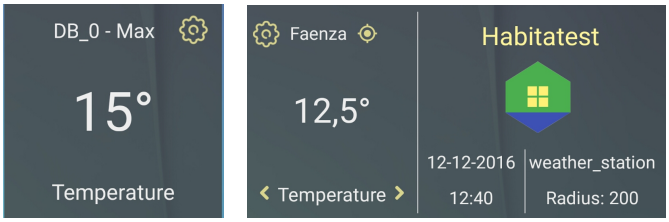


Fig. 7. The widgets provided by the Habitatest APP. In Figure 7(a) we show the simple one, while Figure 7(b) offers more information to the user, such as the service range validity, the date, and geolocation information.

the other, only wants to monitor the highest temperature in the area. Therefore, the user selects all of them in the screen pictured in Figure 6(a), drags and drops all the temperature sensors onto each other as shown in Figure 6(b), and chooses to get the maximum out of them as in Figure 6(c). In case the user is not interested in any of the data classes, he or she can just drag them out of the screen to remove them from the monitoring area. After the selection of the interesting data classes and, possibly, their aggregation, a local and personal service within the application is created. The service is not saved onto the CCU, it is available only to the user who creates it.

The user can then choose to monitor the service directly through the Habitatest app, or by using one of the widgets provided, as in Figure 7(a) or Figure 7(b). After selecting the smaller (Figure 7(a)) or the bigger version (Figure 7(b)), the user can then get updates directly on his or her home screen. If the user selected more than one data class to be monitored, the widget will give the possibility to the user to switch from one type of data to the other through a right and left arrow.

The Habitatest app runs then in the background, gathering all the information selected by the user and aggregating them together locally following the directions given, assuming that the user selected to do so. The user also has the opportunity to click on the data class to get the historian of the measurements, as shown in Figure 6(d).

VII. Conclusions

In this paper we introduced our proposal for the integration of heterogeneous data sources, namely SenSquare. Such platform possesses the capabilities to achieve the task of environmental monitoring for Smart Cities with a significantly fine granularity, joining the concepts of open data sources and crowdsensing, therefore exploiting the devices owned by the end users in order to produce valuable services. We proposed an architecture open to extensions in several ways and still allowing the coexistence of diverse data gathering methods. In order to give an unique interpretability to heterogeneous data we designed a common data structure which we rendered in a relational database. We demonstrated through a user survey the meaningfulness of our proposal, indeed we showed that more than the half of the citizens are willing to collaborate in a crowdsensing campaign at no reward. We also demonstrated the possibility of classifying correctly more than the 80% of the data streams coming from unreliable resources using a simple NLP approach, which can be combined with others to improve the accuracy even more. Finally, we presented our implementation of the IoT service composition, a capability that offers modular and customizable monitoring services, for which we also showed two clients: a web application and a mobile application. To fully exploit the IoT potential, data should be available to different services, so that users can customize them and tailor to their behavior. For this reason, future improvements of our system or other systems relying on the same concept need to push their efforts towards both collaboration and integration. It is worth noting that many users can produce sensitive information which are not intended to be shared. For this reason we took into account, for a future improvement, the possibility for a user of merging personal information in a decentralized location with other data streams from SenSquare, in such a way privacy, intended as sensible data safety, is preserved. Furthermore, such a high volume of data raises the problem of data quality, which we aim to address in future works.

References

[1] T. J. Barnett, A. Sumits, S. Jain, and U. Andra, "Cisco Visual Networking Index (VNI): Global Mobile Data Traffic Forecast Update, 2015-2020," *Cisco White Paper*, 2016.

[2] F. Montori, L. Bedogni, and L. Bononi, "On the Integration of Heterogeneous Data Sources for the Collaborative Internet of Things," in *2nd International Forum on Research and Technologies for Society and Industry (RTSI)*, 2016.

[3] G. Whitelaw, H. Vaughan, B. Craig, and D. Atkinson, "Establishing the Canadian community monitoring network," *Environmental Monitoring and Assessment*, vol. 88, no. 1-3, pp. 409–418, 2003.

[4] A. Lawrence, "No Personal Motive?' Volunteers, Biodiversity, and the False Dichotomies of Participation," *Ethics, Place & Environment*, vol. 9, no. 3, pp. 279–298, 2006.

[5] F. Behmann and K. Wu, *Collaborative Internet of Things (C-IoT): For Future Smart Connected Life and Business*. John Wiley & Sons, 2015.

[6] A. K. Dey and G. D. Abowd, "Towards a Better Understanding of Context and Context-Awareness," *Computing Systems*, vol. 40, no. 3, pp. 304–307, 1999.

[7] T. Buchholz, A. Küpper, and M. Schiffers, "Quality of Context: What It Is And Why We Need It," *Proceedings of the workshop of the HP OpenView University Association*, pp. 1–14, 2003.

[8] F. Montori, L. Bedogni, A. Di Chiappari, and L. Bononi, "SenSquare: a Mobile Crowdsensing Architecture for Smart Cities," in *IEEE World Forum on Internet of Things, WF-IoT 2016 - Proceedings*, 2016.

[9] L. G. Jaimes, I. J. Vergara-Laurens, and A. Raij, "A Survey of Incentive Techniques for Mobile Crowd Sensing," *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 370–380, 2015.

[10] C. C. Conrad and K. G. Hilchey, "A review of citizen science and community-based environmental monitoring: Issues and opportunities," *Environmental Monitoring and Assessment*, vol. 176, no. 1-4, pp. 273–291, 2011.

[11] "Louisiana Bucket Brigade." [Online]. Available: http://www.labucketbrigade.org/

[12] "Cumulocity Framework." [Online]. Available: https://www.cumulocity.com/

[13] "AllJoyn Framework." [Online]. Available: https://allseenalliance.org/framework/documentation

[14] "Xively." [Online]. Available: https://xively.com/

[15] "ThingWorx." [Online]. Available: https://www.thingworx.com/

[16] C. Sarkar, A. U. Akshay, R. V. Prasad, A. Rahim, R. Neisse, and G. Baldini, "DIAT: A scalable distributed architecture for IoT," in *IEEE Internet of Things Journal*, vol. 2, no. 3, 2015, pp. 230–239.

[17] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by Internet of Things," *Transactions on emerging telecommunications technologies*, vol. 25, no. 1, pp. 81–93, 2014.

[18] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[19] J. B. Borges Neto, T. H. Silva, R. M. Assunção, R. A. F. Mini, and A. A. F. Loureiro, "Sensing in the collaborative Internet of things," *Sensors*, vol. 15, no. 3, pp. 6607–6632, 2015.

[20] D. H. Wolpert, "The Lack of A Priori Distinctions Between Learning Algorithms," *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, 1996. [Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/neco.1996.8.7.1341

[21] M. N. Haque, N. Noman, R. Berretta, and P. Moscato, "Heterogeneous ensemble combination search using genetic algorithm for class imbalanced data classification," *PLoS ONE*, vol. 11, no. 1, 2016.

[22] T. Pei, S. Sobolevsky, C. Ratti, S.-L. Shaw, T. Li, and C. Zhou, "A new insight into land use classification based on aggregated mobile phone data," *International Journal of Geographical Information Science*, vol. 28, no. 9, pp. 1988–2007, 2014. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/13658816.2014.913794

[23] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[24] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to Mobile Crowd Sensing," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops, PERCOM WORKSHOPS 2014*, 2014, pp. 593–598.

[25] M. Talasila, R. Curtmola, and C. Borcea, "Mobile Crowd Sensing," in *Handbook of Sensor Networking: Advanced Technologies and Applications*, 2015, no. JANUARY 2014.

[26] M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan, "Medusa: a programming framework for crowd-sensing applications," *Proceedings of the 10th international conference on Mobile systems, applications, and services - MobiSys '12*, no. Section 2, p. 337, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2307636.2307668

[27] C. Leonardi, A. Cappellotto, M. Caraviello, B. Lepri, and F. Antonelli, "SecondNose: an air quality mobile crowdsensing system," in *Proceedings of the 8th Nordic Conference on Human-Computer Interaction Fun, Fast, Foundational - NordiCHI '14*, 2014, pp. 1051–1054.

[28] M. Zappatore, A. Longo, M. A. Bochicchio, D. Zappatore, A. A. Morrone, and G. De Mitri, "A crowdsensing approach for mobile learning in acoustics and noise monitoring," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. ACM, 2016, pp. 219–224.

[29] E. Kanjo, "NoiseSPY: A real-time mobile phone platform for urban noise monitoring and mapping," *Mobile Networks and Applications*, vol. 15, no. 4, pp. 562–574, 2010.

[30] S. Leao, K. L. Ong, and A. Krezel, "2Loud?: Community mapping of exposure to traffic noise with mobile phones," *Environmental Monitoring and Assessment*, vol. 186, no. 10, pp. 6193–6206, 2014.

[31] R. Salpietro, L. Bedogni, M. Di Felice, and L. Bononi, "Park Here! a smart parking system based on smartphones' embedded sensors and short range Communication Technologies," in *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, 2015, pp. 18–23.

[32] F. J. Villanueva, D. Villa, M. J. Santofimia, J. Barba, and J. C. Lòpez, "Crowdsensing Smart City Parking Monitoring," in *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, 2015.

[33] L. Bedogni, M. Di Felice, and L. Bononi, "By train or by car? Detecting the user's motion type through smartphone sensors data," in *2012 IFIP Wireless Days*. IEEE, nov 2012, pp. 1–6.

[34] T. Ludwig, T. Siebigteroth, and V. Pipek, "Crowdmonitor: Monitoring physical and digital activities of citizens during emergencies," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8852, 2015, pp. 421–428.

[35] G. Cardone, A. Cirri, A. Corradi, and L. Foschini, "The participact mobile crowd sensing living lab: The testbed for smart cities," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 78–85, 2014.

[36] D. R. Sanqunetti, "Implementing geo-fencing on mobile devices," 2006.

[37] G. Cardone, A. Cirri, A. Corradi, L. Foschini, R. Ianniello, and R. Montanari, "Crowdsensing in Urban areas for city-scale mass gathering management: Geofencing and activity recognition," *IEEE Sensors Journal*, vol. 14, no. 12, pp. 4185–4195, 2014.

[38] J. Hamari, J. Koivisto, and H. Sarsa, "Does gamification work? - A literature review of empirical studies on gamification," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2014, pp. 3025–3034.

[39] National Geospatial-Intelligence Agency, "Military Map Reading 201," Tech. Rep. [Online]. Available: http://earth-info.nga.mil/GandG/coordsys/mmr201.pdf

[40] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection 2 Methods for Accuracy Estimation," *Proc. of IJCAI'95*, pp. 1137–1145, 1995.

[41] R. J. Mooney and L. Roy, "Content-Based Book Recommending Using Learning for Text Categorization," *Pro- ceedings of the Fifth ACMConference on Digital Libraries*, pp. 195–204, 2000.

[42] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos, "An Evaluation of Naive Bayesian Anti-Spam Filtering," in *European Conference on Machine Learning*, 2000, pp. 9–17.

[43] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of the ACM*, vol. 7, no. 3, pp. 171–176, 1964.

[44] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," *Information Retrieval*, vol. 1, no. 1, pp. 69–90, 1999.