



HAL
open science

When Service-oriented Computing Meets the IoT: A Use Case in the Context of Urban Mobile Crowdsensing

Valerie Issarny, Georgios Bouloukakis, Nikolaos Georgantas, Françoise Sailhan, Géraldine Texier

► To cite this version:

Valerie Issarny, Georgios Bouloukakis, Nikolaos Georgantas, Françoise Sailhan, Géraldine Texier. When Service-oriented Computing Meets the IoT: A Use Case in the Context of Urban Mobile Crowdsensing. ESOC 2018 : 7th European Conference on Service-oriented and Cloud Computing, Sep 2018, Como, Italy. pp.1-16, 10.1007/978-3-319-99819-0_1 . hal-01871515

HAL Id: hal-01871515

<https://inria.hal.science/hal-01871515>

Submitted on 11 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

When Service-oriented Computing Meets the IoT: A Use Case in the Context of Urban Mobile Crowdsensing

Invited paper

Valérie Issarny¹, Georgios Bouloukakis¹, Nikolaos Georgantas¹,
Françoise Sailhan², and Géraldine Texier³

¹ Inria, Paris, France; `first.lastname@inria.fr`

² CNAM Paris, France; `first.lastname@cnam.fr`

³ IMT Atlantique/IRISA/UBL, Rennes, France;
`first.lastname@imt-atlantique.fr`

Abstract. The possibilities of new mobile distributed systems have reached unprecedented levels. Such systems are dynamically composed of networked resources in the environment, which may span from the immediate neighborhood of the users - as advocated by pervasive computing - up to the entire globe - as envisioned by the Future Internet and one of its major constituents, the Internet of Things. This paper more specifically concentrates on urban participatory mobile distributed systems where people get involved in producing new knowledge about the urban environment. Service-oriented and cloud computing are evident baseline technologies for the target mobile distributed systems. Service orientation provides the abstraction to deal with the assembly of the relevant heterogeneous component systems. The cloud provides the infrastructure to deal with the gathering and analyses of the observations coming from the sensing infrastructure, including from people. However, cloud-based centralized solutions come at a price, regarding both resource consumption and privacy risk. Further, the high heterogeneity of the participating nodes results in diverse levels of sensing accuracy. This paper provides an overview of our past and ongoing research to overcome the challenges facing urban participatory mobile distributed systems, which leverages mobile collaborative sensing, networking and computing. The experience with the Ambiciti platform and associated mobile app for monitoring the individual and collective exposure to environmental pollution serves as an illustrative use case.

Keywords: IoT · interoperability · middleware · mobile crowdsensing · urban sensing systems · multiparty calibration.

1 Introduction

The Service Oriented Computing –SOC– (or Service Oriented Architecture –SOA) paradigm has proved particularly appropriate for ubiquitous and mobile

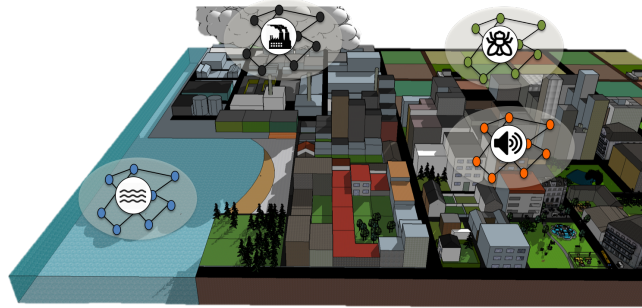


Fig. 1. Monitoring the urban environment using the fixed and mobile IoT.

computing systems [14]. Through the abstraction of networked devices and their hosted applications as services delivered and consumed on demand, the SOA approach enables the assembly of systems whose component systems can be retrieved and composed statically as much as dynamically, thanks to service discovery and service interaction protocols. While mobile services incorporate and apply the fundamental principles of SOA, they present a number of specifics that push certain challenges related to service oriented systems to their extreme and additionally introduce new unique research challenges. Such specifics relate to: (i) dynamism – open mobile environments are much more volatile than typical service environments, with services emerging and disappearing in arbitrary ways without prior notification; (ii) heterogeneity – a direct consequence of ad hoc mobile environments is that no safe assumption can be made about the technological and business features of the services encountered; and (iii) the equation among QoS expectations on services, scalability, and required resources is hard to solve, due to the resource constraints of mobile environments. Dealing with the identified specifics gets even more complex if we consider both traditional computing services and services attached to the physical world by means of sensors and actuators, i.e., Things [9].

This paper specifically focuses on the design of service-oriented systems supporting the analysis of urban-scale phenomena through the composition of the diverse relevant sensing services, from services supported by urban infrastructure networks to services gathering observations from end-users (Fig. 1). The related research challenges are numerous, including the ones due to the aforementioned specifics of mobile services. Here, we concentrate on the challenges that we have been investigating in relation with the development and deployment of an urban service monitoring the individual and collective exposure of the population to environmental pollution (Section 2). Thanks to the sensors that people may increasingly “hold” and/or “wear” while moving, we are potentially able to monitor urban phenomena at a very fine grain across time and space. However, while the potential is high and examples in the area of traffic monitoring evidences it, the ability to monitor diverse urban phenomena at scale

is far from being straightforward. There are indeed many hurdles arising, among which the following ones that we have been and are currently investigating:

- The monitoring of urban phenomena involves analyzing observations across time and space, but also across application domains (e.g., BMS, traffic, health). As a result, the thorough analysis of urban phenomena involves the ability to compose services across horizontal (location dependent) and vertical (application dependent) boundaries. In addition, the urban system architecture must ease such a composition of services at scale. Toward that goal, we investigate system architectures that build upon the pub/sub communication paradigm, which has proven well suited for open, large-scale dynamic systems. Further, we adopt the edge computing paradigm, which includes leveraging the processing power of the mobile nodes. Moreover, the system architecture must ease the integration of highly heterogeneous Things, which goes along with supporting interaction across diverse protocols (Section 3).
- While the SOC paradigm together with semantic technologies allow exposing the capabilities of the connected Things despite their high heterogeneity, the large diversity of Things also manifests itself from the standpoint of the accuracy of the delivered observations. It is then essential to assess the quality of the provided observations, which may result in either filtering out or correcting part of the observations. Obviously, the ability to correct observations leads to gathering more knowledge, for which we study automated, collaborative multi-party calibration (Section 4).
- The scale of the urban systems significantly challenges the networking and computing infrastructures, especially when the system keeps monitoring phenomena over time. The adoption of the pub-sub communication paradigm together with edge computing in our architecture contributes to overcoming the challenge by fostering communication and computing within the relevant geographical area(s). In addition, the mobility of nodes is worth exploiting to cross geographical regions and/or application domain boundaries. In that direction, we explore the offloading of the urban WSN-based IoT infrastructure networks to "peer" mobile nodes that act as crowdsensors in the same application domain. This allows saving the energy due to communication in the WSN and thereby increasing the WSN lifetime (Section 5).

The above list is only a subset of the challenges that IoT-based systems need to face to enable a thorough analysis of urban phenomena. This paper then concludes with an overview of other challenges that we and/or others are investigating and for which service-oriented and cloud computing are essential baseline technologies but that need to be revisited to handle the specifics of IoT-based systems.

2 Monitoring the exposure to the urban environmental pollution: A use case

According to the EU 1996 Pollution Directive (www.eea.europa.eu/policy-documents/council-directive-96-61-ec-ipcc), " 'Pollution' shall mean the direct or in-

direct introduction as a result of human activity, of substances, vibrations, heat or noise into the air, water or land which may be harmful to human health or the quality of the environment, result in damage to materiel property, or impair or interfere with amenities and other legitimate uses of the environment.” With the majority of the population now living in cities, it is increasingly critical to be able to monitor the exposure to environmental pollution in urban centers. In particular, the research community suggests that the social cost of noise and air pollution in EU -including death and disease- could be nearly EUR 1 trillion, while the social cost of smoking in the EU has been estimated to be EUR 544 billion [17].

The development of the IoT, including the one of the sensing technologies that smartphones now embed, together with advancements in machine learning, allow for the deployment of cloud-based platforms that collect and aggregate the many relevant urban data sources to produce street-level hourly maps about the environmental pollution (e.g., [21]). Also, mobile apps that serve collecting observations about the environmental pollution enable to inform users about their individual exposure and related impact on their health [7]. As part of our research, we have developed a cloud-based platform and related mobile app for monitoring the individual and collective exposure to the environmental pollution [6]. In particular, the app implements noise sensing using the phone’s microphone while accounting for the relatively low accuracy of the resulting observations [22]. In a nutshell, the mobile app implements a crowdsensing application that periodically transfers the collected observations to the cloud server, which filters out the observations that are not deemed accurate enough. The cloud server implements data assimilation techniques to integrate observations from various data sources with mathematical models to simulate the state of a system of an urban phenomenon [20].

The research resulted in the Ambiciti solution (ambiciti.io) that enables monitoring the individual exposure to noise and air pollution (Fig. 2-Left), while producing street-level maps about environmental pollution (Fig. 2-Right). In summer 2015, we deployed the first version of Ambiciti, which initially supported noise pollution monitoring only. Since autumn 2016, Ambiciti also monitors the exposure to air quality. The urban-scale deployment of Ambiciti –especially in Paris– has led us to identify a number of recommendations for the IoT-based sensing system and supporting architecture [10, 12], which we have been and are still investigating. The major challenge to face is to enable and promote the gathering of observations of sufficient quality at scale so as to enable the thorough analysis of urban phenomena. The following reports on our complementary studies toward that goal. All these studies focus on developing distributed collaborative protocols that contribute to increasing the accuracy of the knowledge produced at the edge, while reducing the overall resource consumption associated with data processing and communication.

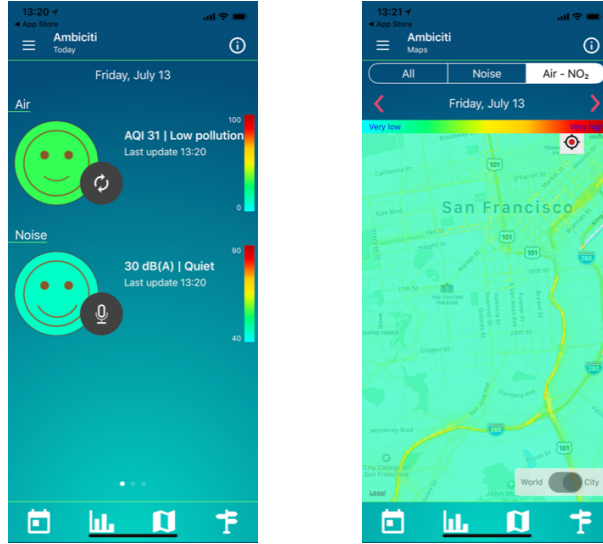


Fig. 2. Monitoring the exposure to environmental pollution using Ambiciti.

3 System architecture for the urban IoT

Ubiquitous computing devices featuring sensing capabilities are deployed in a variety of *application domains*, such as smart cities, smart factories, resource management, intelligent transportation and healthcare to name a few. This enables analyzing observations of different domains across time and space in the so called *urban environment*. However, enacting IoT urban systems is still raising tremendous challenges for the supporting infrastructure from the networking up to the application layers. Key challenges [1, 18] relate to *deep heterogeneity*, *high dynamicity*, *scale*, and many others.

To support the deployment of large-scale, heterogeneous and dynamic IoT urban systems we rely on the architecture depicted in Fig. 3. This comprises the following layers:

- **Device:** This layer serves applications from multiple domains. Each application involves heterogeneous devices and processing capabilities in mobile nodes.
- **Edge:** At this layer, we adopt the Edge computing paradigm to collect multi-domain data from end-users being in several regions. We define a region as a bounded geographical area with common features (e.g., a university campus). Additionally, to deal with the heterogeneity of IoT applications (exposed as services), we deploy *software interoperability artifacts* at this layer.
- **Cloud:** Several domains of IoT applications require powerful processing capabilities (e.g., for executing machine learning algorithms) and thus, at this layer we enable the use of Cloud computing resources.

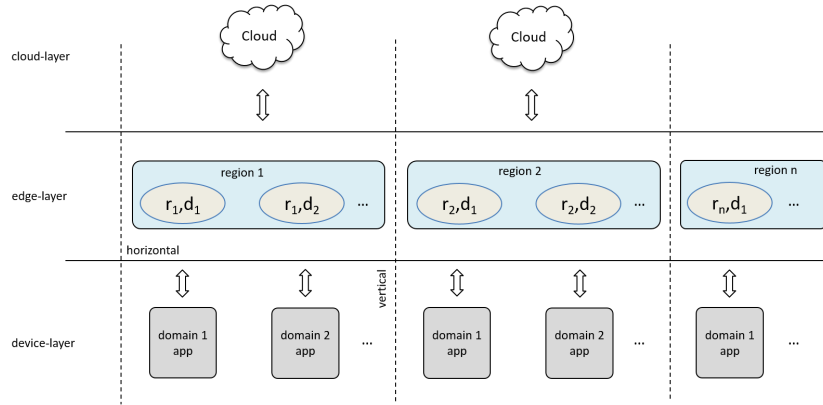


Fig. 3. System architecture for the urban IoT.

3.1 Supporting wide-scale IoT apps

Mobile IoT applications are typically deployed on resource-constrained devices with intermittent network connectivity. To support the deployment of such applications, the pub/sub interaction paradigm is often employed, as it decouples mobile peers in both time and space. In a pub/sub system, multiple peers interact via an intermediate *broker* entity – publishers produce *events* characterized by a specific *filter* to the broker. Subscribers subscribe their interest for specific filters to the broker, which maintains an up-to-date list of subscriptions.

To support distributed applications spanning a wide-area, the pub/sub system has to be implemented as a set of independent, communicating brokers, forming the *broker overlay*. Let $B = \{b : i \in [1..|B|]\}$ be the set of pub/sub brokers. As depicted in Fig. 4, in such architectures [3], peers can access the system through any broker that becomes their home broker. Then, based on the peers' input, the pub/sub system ensures the delivery of the events produced towards the interested subscribers. As still shown in Fig. 4, a message broker can be assigned to support a specific region or one or more application domains inside a region. This depends on the corresponding administration of the related region's vertical markets.

Building an IoT application over a pub/sub infrastructure, requires the selection of an appropriate protocol (e.g., MQTT). Such a protocol enables IoT devices to access the broker overlay and push/receive events. Additionally, to create the pub/sub broker overlay, the corresponding *message broker* implementation (e.g., VerneMQ, HiveMQ, etc) must be selected. Each message broker technology has different capabilities, such as: compatibility with different protocols, support for clustering (i.e., forming a broker network), provision of performance features, etc. To enable the interconnection between message brokers that employ different protocols for message routing (e.g., MQTT and AMQP in Fig. 4), *adapter* software components can be utilized. Usually, such components are supported by the corresponding message broker technology (e.g., VerneMQ).

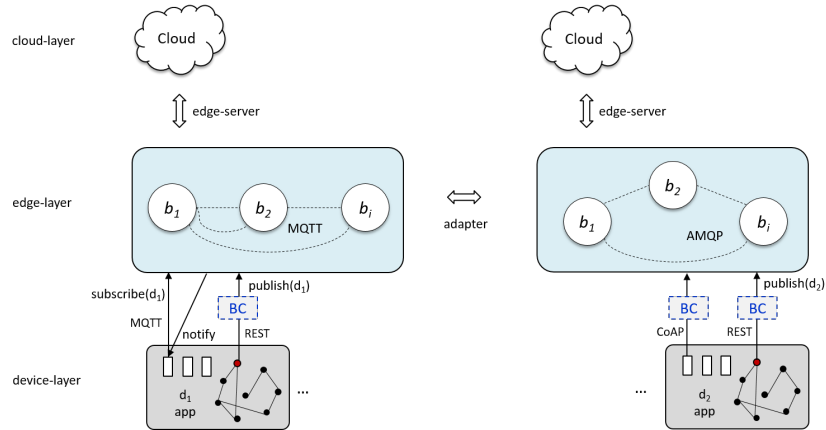


Fig. 4. Multi-domain/region pub/sub system.

An important design-time decision is the provision of resources for message broker deployment. In this work we assume the deployment of brokers at the Edge. To enable the interconnection between a message broker (Edge) and the cloud, we employ an intermediate *edge-server* which corresponds to a pub/sub subscriber that collects data and forwards them to the cloud.

3.2 Supporting Heterogeneous IoT apps

Mobile environments can be very heterogeneous, both locally but also when reaching out to the whole IoT. Service heterogeneity concerns both business semantics and communication middleware. The former issue has led to a wide use of ontologies and related technologies in SOA. As for the latter issue, it is due to the fact that mobile IoT devices may use different communication contexts. This calls for support for heterogeneous interaction styles, namely message-driven, event-driven, and data-driven styles. Different interaction styles apply to different needs; for instance, asynchronous, event-based pub/sub is more appropriate for highly dynamic environments with frequent disconnections of involved entities. This fact makes the various service bindings supported by most service technologies too stringent, since they comply with a single (client/server) message-based interaction style. Service models should be able to abstract and comprehensively specify the various service bindings of mobile IoT devices. This further implies extending the notion of service and introducing adequate service interaction modeling. Moreover, interoperability mechanisms are required between heterogeneous interaction styles based on transformation mappings between their models.

Our work in this context has focused on the problem of interconnecting systems that employ different interaction styles or paradigms [5]. We include in particular the client/server, publish/subscribe, data streaming and tuple space paradigms, which cover the majority of communication middleware protocols

enabling system interaction. Our overall approach generalizes the way to design and implement service-oriented distributed applications, where the employed interaction paradigms are explicitly represented and systematically integrated.

As a first step, we identified the semantics of the four principal interaction paradigms and elicited a connector model for each paradigm. Our models represent the essential semantics of interaction paradigms, concerning *space coupling*, *time coupling*, *concurrency* and *synchronization coupling*. These four categories of semantics are of primary importance, because these are end-to-end semantics: when interconnecting different interaction paradigms, we seek to map and preserve these semantics. Inside each interaction paradigm, we identify further one or more of four *interaction types*: *one-way*, *two-way synchronous*, *two-way asynchronous* and *two-way streaming* interaction. These interaction types incorporate the above semantics of end-to-end interactions.

Following the previous step, we introduced a higher-level connector model that comprehensively represents the semantics of various middleware protocols that follow the four base connector models, hence it can represent the majority of the existing and possibly of the future middleware protocols. In particular, this connector models the four interaction types supported by the base connector models in an abstract way. Furthermore, we introduced a generic interface description language for specifying interfaces in an abstract way. This language enables the definition of operations provided or required by an application component that follow the four interaction types.

By relying on the above abstractions, we have introduced a solution to seamlessly interconnect IoT devices that employ heterogeneous interaction protocols at the middleware level [2]. Interconnection is performed via one or more message brokers lying at the Edge as presented in the previous section. More specifically, a dedicated software component called a *Binding Component (BC)* performs the conversion between the middleware protocol of an IoT device and the message broker protocol as depicted in Fig. 4.

Based on our modeling abstractions, we have developed a systematic approach for the automated synthesis of BCs. The development of BCs is a tedious and error-prone process, which can highly benefit from our automated systematic support. This can help application developers integrate heterogeneous IoT devices inside complex applications. Furthermore, the automated BC synthesis is essential for applications relying on the dynamic runtime composition of heterogeneous devices, where there is no human intervention. Our solution was introduced as a core component of the H2020 CHOReVOLUTION project (chorevolution.eu) to enable heterogeneous interactions in services & Things choreographies. Currently, it supports the following middleware protocols: REST, CoAP, MQTT, WebSockets, SemiSpace and DPWS. It is released as open source software (gitlab.inria.fr/zefxis).

4 Overcoming the low accuracy of mobile crowdsensors

Current crowdsensing approaches are primarily focused on collecting data at a large scale, as in particular supported by the pub/sub system architecture of the previous section. However, without any suitable calibration mechanism, smartphones tend to disclose inaccurate readings. To tackle this issue, we propose to collaboratively and automatically calibrate in the field, the sensors embedded in and/or connected to the smartphones. Thus, when some mobile sensors are in the same spatial vicinity, uncalibrated sensors get automatically calibrated –without involving the end users. Compared to existing solutions that iteratively calibrate one by one a unique mobile sensor at a given time, our approach supports the joint calibration of several uncalibrated sensors, thus reducing the overall calibration delay. This leads to a multi-party, multi-hop calibration for which the history of calibrated sensors is used to assess the best calibration hyperpath, which is the one that minimizes the accumulated calibration error. Our approach towards accurate urban sensing further distinguishes from the traditional WSN setting by enabling mobile, consumer-centric smartphones to interact with both stationary and mobile sensors in their proximity so as to calibrate the smartphones’ sensors. In particular, our solution leverages the presence of two types of sensors:

- *Reference sensors* correspond to high fidelity sensors (e.g., a sound meter in the case of noise sensing) that are regularly calibrated by experts. Reference sensors are typically government-run (see Figure 2 in [13] for an illustration of the few reference sensors measuring noise in the city of Paris) and their measurements are generally made publicly available. They are wirelessly networked [16] as detailed in the next section.
- *Mobile sensors* correspond to low cost and less accurate sensors that are owned by individuals (they typically include the sensors embedded in smartphones) and that need to be remotely calibrated.

Then, the reference sensors that belong to the urban infrastructure support the calibration of any mobile sensor that is within sensing range. And, in turn, the newly calibrated sensors may be used to calibrate the uncalibrated sensors they meet (as they move). The following outlines our solution whose detailed presentation may be found in [15].

Multi-Party Calibration. During a rendezvous, any uncalibrated sensor i attempts to calibrate using the calibrated measurements provided by the m nearby sensors, for which we leverage multivariate linear regression.

As illustrated in Fig. 5, there is a linear dependency between the measurements produced by a non-calibrated sensor and the m surrounding calibrated sensors. Thus, any reading $y_i(t)$ of the uncalibrated sensor i at time t can be expressed as the following linear function:

$$y_i(t) = \beta_0 + \beta_1 x_1(t) + \beta_2 x_2(t) + \dots + \beta_m x_m(t) + e_i(t) \quad (1)$$

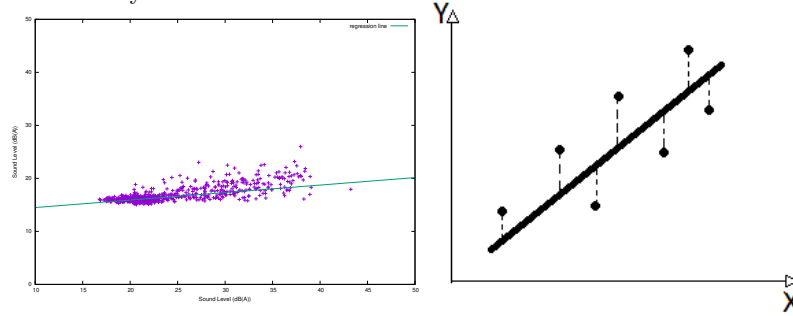


Fig. 5. An uncalibrated smartphone calibrates using the calibrated measurement of one calibrated smartphone. Regression lies in fitting a straight line to bivariate data using least square regression. The regression minimizes the vertical residual (dashed line on the right figure) by keeping to a minimum the squared vertical distance from each data point to the line.

where: $x_j(t)$ ($j \in [1, \dots, m]$) denotes the calibrated measurements of the m sensors; β_0, \dots, β_m represent the unknown (and fixed) regression coefficients; and $e_i(t)$ is the residual noise, with $t \in [t_1, \dots, t_p]$. The goal is to find the regression coefficients $\beta_0, \beta_1, \dots, \beta_m$ that minimizes the residual noise. For this, we rely on the least square method, which minimizes the sum of the squared distance of the data points from the line measured perpendicular to the x axis (see Fig. 5).

Multi hops, multi-party calibration. Thanks to the above multivariate linear regression, the calibration coefficients of an uncalibrated smartphone sensor i may be estimated based on the readings provided by the surrounding sensors that are met during a multiparty rendezvous. However, sensor i should not systematically (re-)calibrate: it must do so only if the quality of the regression is sufficient. In particular, a high residual error (i.e., e_i in Equation 1) reflects a poor correlation between the readings provided by the surrounding sensors, and in such a case, the conditions are not met for an effective calibration. This may occur when some smartphones are in bags/pockets and others are handheld.

Another aspect to consider for assessing the relevance of a given multiparty calibration relates to the history of past calibrations. We must compare the quality of the calibration parameters computed in the current rendezvous against the quality of the previous calibrations (if any). We thus maintain the history of multiparty calibrations using a weighted directed *hypergraph*: a multiparty rendezvous between sensor i and K sensors is represented by an *hyperedge* between sensor i and the K sensors. The quality of the regression established by i based on the readings provided by the K sensors is reflected by the weight of the directed hyperedge between i and the K sensors. Ultimately, the preferred calibration is the one that results in the lowest cumulated weight.

Summarizing the proposed multiparty calibration method, any smartphone (embedding a sensor) i participating in the collaborative calibration periodically performs the following process:

1. Smartphone i detects the presence of nearby sensing device(s), i.e., devices in the shared sensing range.
2. If any eligible rendezvous, i exchanges its sensing measurements (i.e., time series) in a synchronized manner so as to establish the linear relationship between the measurements of the nearby sensors and the raw measurements obtained locally.
3. The best calibration path is determined and the calibration function is set.

Overall, the key feature of the calibration system is to enhance the accuracy of the raw data sensing provided by a self-selected and potentially unskillfulness or distrustful crowd. As few anomalies or discordant observations may contaminate the overall knowledge, it is critical to identify, detect and treat such anomalies. To tackle this issue, we attempt to combine two complementary approaches : (i) we harness the power of the crowd to collaboratively sense and calibrate, (ii) we coordinate with city-owned sensors so as to support an initial bootstrapping calibration.

5 When crowdsensing meets the infrastructured IoT networks

The previous section shows how the participatory sensing benefits from an interaction with smart city sensing infrastructures to bootstrap the calibration of the crowdsensors and deliver more accurate measurements. This cooperation can be deepened to achieve a mutual benefit for crowdsensing and the urban IoT infrastructures. For this, we advocate the possibility of direct interactions between crowd smartphones and the infrastructured sensors that surround them. First – as illustrated in the previous section –, such interactions can benefit smartphones by allowing them to quickly and directly retrieve (higher quality) measurements taken around them. Second, smartphones can push their participatory engagement further: in addition to making their smartphone’s sensors available, they can also offer to share part of their network interconnection to offload the WSN and relay part of its data (as shown in Fig. 7).

In general, urban measurement infrastructures are implemented by Wireless Sensor Networks (WSN), which are more or less extensive and distributed in the city. The sensors perform measurements and send them towards a gateway (the WSN’s sink) in charge of publishing to the pub/sub system the collected data through its Binding Component (BC) as shown in Fig. 6. To do this, the WSN implements convergecast communication to allow the sink to collect data generated by each sensor in the network. With the exception of the leafs, each sensor sends its parents both the data it generates and the data transmitted to it by its children. The worst case is that of the nodes near the sink: the emission of the measurements they generate represents only a small fraction of their communications since they spend most of their time relaying their children’s data.

But, due to high deployment constraints in urban environments, sensors in the WSNs are often battery powered, which limits the lifetime of the whole urban

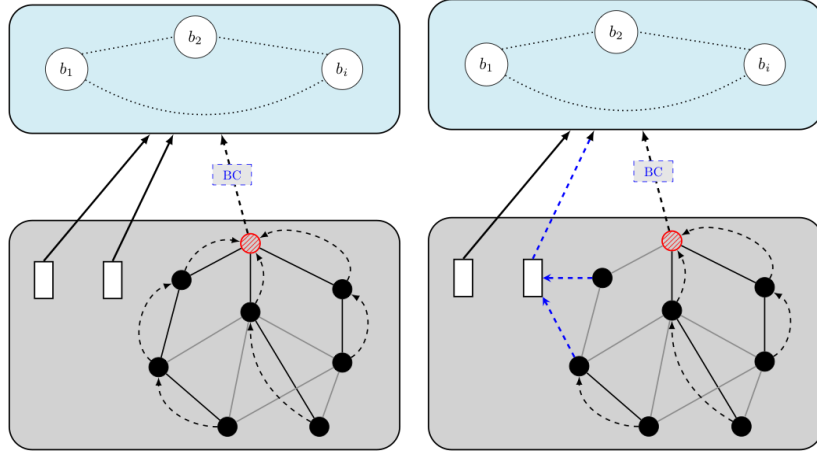


Fig. 6. Classical convergecast data collection in the WSN. **Fig. 7.** Smartphone offloading data from the WSN.

sensing infrastructure. However, the definition of the WSN lifetime depends on the policy of the network operator. If it does not tolerate any failure, the lifetime of the network is limited by the exhaustion of the energy of at least one sensor. Another strategy is to tolerate a given number of sensors that no longer work, in which case the lifetime of the network is determined by the number of sensors that remain connected to the sink despite the death of a subset of sensors. Whatever lifetime definition is chosen, a smartphone can help the infrastructure survive longer by giving it the ability to relay some of its data. Note that this requires that smartphones can connect to infrastructured sensors, for example via protocols such as Zigbee, 802.15.4 or Bluetooth Low Energy (BLE). The fact is that sensors are increasingly able to communicate and we believe that lack of compatibility is less a technical problem than a market problem. So, although with the exception of BLE, most smartphones do not yet integrate the main interfaces supported by the sensors, we think that this will evolve within a reasonable time. Therefore, we consider that direct communication between the infrastructure and smartphones is not an obstacle for our work and will be available in the future.

Many studies have sought to optimize the lifetime of networks. They can be classified according to the techniques they propose. Some solutions seek to maximize the network lifetime by enhancing RPL energy efficiency [11]. For example with a RPL metric that integrates the available energy of the WSN nodes [8] to ensure that the nodes solicited for relaying data are the ones with the highest energy budget. However, using a routing protocol with such a very dynamic metric implies great routing instability. Other solutions are based on Delay Tolerant Networks communication techniques. In particular, the optimization of the data collection by a mobile node, usually through the optimization of the mobile node's trajectory based on variants of the traveling salesman problem [4].

In [19], we have presented how the cooperation of the crowd smartphones and the urban WSN can extend the measurement infrastructure’s lifetime based on both optimization and traffic engineering techniques. We proposed a model to integrate the crowd smartphones as mobile sinks able to offload a portion of the WSN data transmissions. We optimize the routing to take advantage of the path diversity offered by the WSN topology. With a simple energy model (each sensor has an energy budget given by the capacity of its batteries), we determine the number of communications a sensor is able to perform before exhausting its energy. We can also compute the network lifetime, in our case the time before at least one sensor runs out of energy. Our linear program determines the quantity of data that a sensor should send to each of its neighbors (including mobile nodes) to maximize the network lifetime. Although the efficiency of the solution is strongly linked to the location of mobile nodes, we have shown on a realistic example that when we allow a subset of network sensors to communicate with a mobile node, we can increase the lifetime of the network in significant proportions (in our example, we multiplied the lifetime of the network by 7). Our work opens several research issues that we are exploring as part of our ongoing and future work. One of these issues relates to our load balancing strategy that requires the introduction of proportional routing into the WSN routing mechanism. This modification also makes it possible to support temporary failures of the sensors by offering next alternative jumps if a node having to relay the information is not available at the time of emission.

6 Conclusion

The IoT holds the promise of blending the physical with the virtual world, while the power of the crowd enables gathering knowledge at unprecedented scales across both time and space. However, the vision raises tremendous challenges for the supporting system architectures, among which keeping pace with the target scale and overcoming the heterogeneity of the IoT nodes. The literature has shown that service-oriented and cloud computing are evident baseline technologies: service orientation brings abstraction to overcome heterogeneity and the cloud brings the computing infrastructure to aggregate the knowledge sensed within the IoT. However, this puts high demand on the networking infrastructure, while collaborative networking and computing at the edge allow for significant resource saving. Our research focuses on such collaborative distributed IoT-based system architectures at the edge, where we build upon our experience on developing and deploying an urban-scale system for the monitoring of the individual and collective exposure to environmental pollution.

This paper has outlined our most recent research in the area, spanning the study of: (1) a publish-subscribe system architecture enriched with software interoperability artifacts to support the interconnection of applications implementing heterogeneous communication paradigms; (2) in-network multi-party, multi-hop calibration so as to overcome the low accuracy of low cost sensors, and especially of the sensors embedded in smartphones; and (3) collaboration

between the smartphones running crowdsensing apps and the urban WSN-based IoT infrastructures toward increasing the lifetime of the WSN networks.

Crowdsensing- and IoT-based system architectures, from networking up to the middleware and application layers, have been studied extensively over the last ten years, while we have focused our presentation on our specific contributions over the last couple of years. The interested reader is referred to the bibliography provided in the cited references for an overview and analysis of related work.

We have many research challenges ahead of us and some of them have been mentioned along the sections. Among our ongoing research, we would like to stress the study of privacy-aware crowdsensing systems due to the tension that the topic creates: (1) On the one hand, we know that the more accurate and precise knowledge we get about the urban environment, the more we will be able to develop effective solutions to enhance the people's well-being and reduce urban resource consumption; (2) On the other hand, gathering such a precise knowledge also goes along with the ability to infer knowledge about people and even individuals. Part of the solution lies in policy making as for instance illustrated by the recent GDPR data protection policy by the EU. Further, crowdsensing systems must be designed to enforce privacy preservation. Distributed collaborative crowdsensing systems where the individual knowledge remains at the edge and ultimately on the user's device contribute to it. Still, the systems must support the appropriate tradeoffs so that relevant urban-scale knowledge may be computed to inform the city development.

Acknowledgments: The authors would like to thank the support of: the Inria@SiliconValley International Lab, CityLab@Inria Project lab, and the EIT Digital innovation activity Env&You. They also gratefully acknowledge the major contribution of their Inria colleagues, Vivien Mallet, Pierre-Guillaume Raverdy and Kinh Nguyen, to the development of the Ambiciti system solution.

References

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials* (2015)
2. Bouloukakis, G.: Enabling Emergent Mobile Systems in the IoT: from Middleware-layer Communication Interoperability to Associated QoS Analysis. Ph.D. thesis (2017)
3. Bouloukakis, G., Georgantas, N., Kattapur, A., Issarny, V.: Timeliness evaluation of intermittent mobile connectivity over pub/sub systems. In: *Proc. of the 8th ACM/SPEC on Int. Conf. on Performance Engineering* (2017)
4. Garraffa, M., Bekhti, M., Létocart, L., Achir, N., Boussetta, K.: Drones path planning for wsn data gathering: A column generation heuristic approach. In: *IEEE Wireless Communications and Networking Conference (WCNC)* (2018)
5. Georgantas, N., Bouloukakis, G., Beauche, S., Issarny, V.: Service-oriented distributed applications in the Future Internet: The case for interaction paradigm

- interoperability. In: ESOC 2013 - European Conference on Service-Oriented and Cloud Computing. Lecture Notes in Computer Science, vol. 8135 (2013)
6. Hachem, S., Mallet, V., Raphaël, V., Raverdy, P.G., Pathak, A., Issarny, V., Bhatia, R.: Monitoring Noise Pollution Using The Urban Civics Middleware. In: IEEE Big Data Service (2015)
 7. Hachem, S., Mathioudakis, G., Pathak, A., Issarny, V., Bhatia, R.: Sense2Health: A Quantified Self Application for Monitoring Personal Exposure to Environmental Pollution. In: SENSORNETS (2015)
 8. Iova, O., Theoleyre, F., Noel, T.: Using multiparent routing in RPL to increase the stability and the lifetime of the network. *Ad Hoc Networks* **29** (2015)
 9. Issarny, V., Bouloukakakis, G., Georgantas, N., Billet, B.: Revisiting Service-oriented Architecture for the IoT: A Middleware Perspective. In: 14th International Conference on Service Oriented Computing (ICSOC) (2016)
 10. Issarny, V., Mallet, V., Nguyen, K., Raverdy, P.G., Rebhi, F., Ventura, R.: Dos and Don'ts in Mobile Phone Sensing Middleware: Learning from a Large-Scale Experiment. Proceedings of the 2016 International Middleware Conference (2016)
 11. Kamgueu, P.O., Nataf, E., Ndie, T.D.: Survey on RPL enhancements: a focus on topology, security and mobility. *Computer Communications* (2018)
 12. Lefèvre, B., Issarny, V.: Matching Technological & Societal Innovations: The Social Design of a Mobile Collaborative App for Urban Noise Monitoring. In: 4th IEEE International Conference on Smart Computing (2018)
 13. Maisonneuve, N., Stevens, M., Ochab, B.: Participatory noise pollution monitoring using mobile phones. *Information Policy - Government 2.0: Making Connections between citizens, data and government* **15**(1,2) (2010)
 14. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: State of the art and research challenges. *Computer* **40**(11) (2007)
 15. Sailhan, F., Issarny, V., Tavares Nascimento, O.: Opportunistic Multiparty Calibration for Robust Participatory Sensing. In: MASS 2017 - IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (2017)
 16. Santini, S., Ostermaier, B., Vitaletti, A.: First experiences using wireless sensor networks for noise pollution monitoring. In: Proceedings of the Workshop on Real-world Wireless Sensor Networks (REALWSN) (2008)
 17. SEP: Links between noise and air pollution and socioeconomic status – in-depth report 13 produced for the european commission, dg environment by the science communication unit, uwe, bristol. Tech. rep., Science for Environment Policy (2016), <http://ec.europa.eu/science-environment-policy>
 18. Teixeira, T., Hachem, S., Issarny, V., Georgantas, N.: Service oriented middleware for the internet of things: a perspective. In: ServiceWave – European Conference on a Service-Based Internet (2011)
 19. Texier, G., Issarny, V.: Leveraging the power of the crowd and offloading urban iot networks to extend their lifetime. In: LANMAN 2018: IEEE International Symposium on Local and Metropolitan Area Networks (2018)
 20. Tilloy, A., Mallet, V., Poulet, D., Pesin, C., Brocheton, F.: BLUE-based NO₂ data assimilation at urban scale. *Journal of Geophysical Research* **118**(4) (2013)
 21. Ventura, R., Mallet, V., Issarny, V., Raverdy, P.G., Rebhi, F.: Estimation of urban noise with the assimilation of observations crowdsensed by the mobile application Ambiciti. In: INTER-NOISE 2017 - 46th International Congress and Exposition on Noise Control Engineering Taming Noise and Moving Quiet (2017)
 22. Ventura, R., Mallet, V., Issarny, V., Raverdy, P.G., Rebhi, F.: Evaluation and calibration of mobile phones for noise monitoring application. *Journal of the Acoustical Society of America* **142**(5) (2017)