Supersensors: Raspberry Pi Devices for Smart Campus Infrastructure (short paper)

Kristian Hentschel, Dejice Jacob, Jeremy Singer, and Matthew Chalmers
School of Computing Science University of Glasgow
Glasgow, UK
Email: firstname.lastname@glasgow.ac.uk

***Abstract*—We describe an approach for developing a campus-wide sensor network using commodity single board computers. We sketch various use cases for environmental sensor data, for different university stakeholders. Our key premise is that supersensors—sensors with significant compute capability—enable more flexible data collection, processing and reaction. In this paper, we describe the initial prototype deployment of our supersensor system in a single department at the University of Glasgow.**

I. Introduction

What does a 21st century university campus look like? At Glasgow, we are redeveloping our campus at a cost of €1bn over the next decade. One headline objective is to create a 'smart' campus. However it is not clear what this means in practice. This paper explores possible infrastructure and use cases for a prototype smart campus testbed. As a starting point, we assume that all campus users have mobile devices (smartphones or tablets) and that there is extensive wifi coverage across the university premises, indoors and outdoors.

Our key idea is to embed physical sensors directly into the fabric of the campus. These might include footfall-sensors on walkways, temperature sensors in rooms, and sound sensors in public spaces. In general, such small-scale sensors should be widely distributed and unobtrusive. They should be low-power electronic devices, with minimal costs for procurement, installation, operation and maintenance. To allay concerns over privacy, no personal data should be collected directly by the sensors. Instead, data collection and reporting should focus on environmental factors, rather than personal ones. For instance, sound sensors will monitor ambient volume rather than the actual sound signal waveform.

There are many use cases for this style of environmental sensing data infrastructure. In a university administrative context, a plausible use case would be 'collect room utilization statistics' across centralized teaching spaces. For students, the use case might be 'locate a quiet study area'.

These Internet of Things (IoT) based sensors and end-user scenarios are largely standard, e.g. [1], [2]. However our project features the following three novel characteristics:

1) We are advocating the use of *supersensors*. Our sensor nodes are moderately powerful Linux servers, hosted on Raspberry Pi devices. They are capable of local compute operations, as well as transmitting data to a centralized database. Each node can behave autonomously to carry out tasks like sending tweets, processing data, dynamic reconfiguration and communicating with other local devices.
2) We use *ad-hoc networking* for nodes in poorly connected locations. The old university buildings have particularly thick stone walls, so wifi coverage or signal strength is impaired. Our system will support sensor-to-sensor communication via bluetooth, or delay-tolerant networking with mobile phones acting as data mules.
3) We employ *sensor fusion and machine learning* techniques. In addition to deployed sensors, we will harvest other feature data such as university timetable data and social media streams. We can process this multidimensional data using machine learning techniques to infer events.

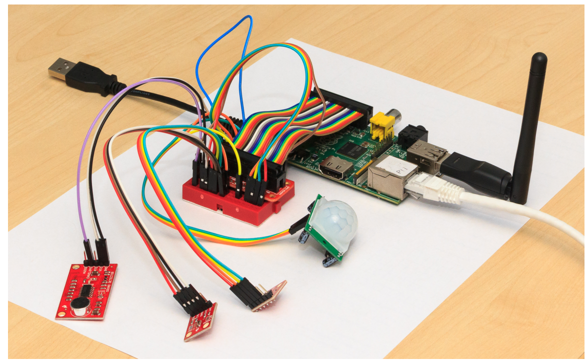

Fig. 1. The Raspberry Pi with its many exposed interfaces and powerful CPU makes for a flexible base platform to evaluate sensor devices, and prototype data collection and compute applications.

These innovations increase the applicability of our scheme. While our smart campus sensor system is currently only in prototype stage, i.e. deployed in a small part of a university environment with limited use cases, we argue that the system should scale up effectively and be useful in a wide variety of contexts.

II. Sensor Nodes

A. Specification

Each individual sensor node is built around a Raspberry Pi device, see Figure 1. This single-board general-purpose computer has a small footprint ($90 \times 60 \times 20\text{mm}^3$) and relatively low power requirement ($< 3W$). The hardware is mass-produced and readily available. Over eight million boards have been sold to date [3].

The Raspberry Pi device has a variety of interfaces for attaching hardware sensor devices. We use the I2C bus for simpler sensors (light, temperature, motion, sound) and USB for more complex sensors (Wifi). The system runs a standard Linux distribution. We add several low-utilization daemon services to gather realtime sensor readings and cache the data in local memory. This data is periodically flushed to a remote server via a secure TCP/IP connection.

Each board currently requires two physical connections, for wired ethernet and DC power supply via mains transformer. These connectivity requirements limit the deployment options for our sensors, so we are considering a switch to wireless networking and renewable power (e.g. solar).

The Raspberry Pi board is relatively inexpensive. There are various generations of boards; we are using generation 1 model B at present. Each board costs around €30. However in this era of disposable compute devices, Raspberry Pi boards are given away 'free' with computing magazines [4]. The cost of the sensor devices is minimal, around €20 in total.

Individual sensor node configuration is supported. Configuration options include:

1) hostname of data collection server.
2) frequency at which each sensor gathers data.
3) frequency at which aggregated data is pushed to server.
4) which sensors are enabled/disabled on this node.

Configuration is possible via TCP/IP connection to individual nodes. Also, all nodes can be configured directly from the server.

Each supersensor that is physically embedded in the campus has a Quick Response (QR) code label. (We intend to support beacon-style Bluetooth URL payloads too.) This per-node label directs people to information about the Glasgow smart campus infrastructure in general, as well as a stream of data readings from this local node.

B. Benefits

This section discusses the advantages and disadvantages of supersensors, i.e. 'fat' sensor nodes that have sufficient compute capability to do more than simply supplying a stream of readings to a server.

While larger nodes will use more electrical power (i.e. have higher operational costs) and will be more expensive to purchase (i.e. have higher installation costs), we feel that the Raspberry Pi device is *below* the impracticality threshold.

The power draw is small enough to embed in campus architecture. If there is one supersensor in each public room, this is a small power cost in relation to lighting that space. Further, the initial cost of the sensor is small, its physical footprint is negligible and it can be easily secured. We do not expect significant 'sensor vandalism' [5] given the low cost of the sensors, the security of the university campus and the non-intrusive nature of the sensing [6]. However a planned future project will look at new ways to assure the trustworthiness of this infrastructure.

Supersensors allow for richer on-node computation. This might include intelligent filtering of sensor data. It is possible to capture and buffer incoming data while network connectivity is disrupted. It is also straightforward to use ad-hoc networking techniques or delay-tolerant networking protocols to relay data to the server.

One key issue at Glasgow that currently prevents outdoor footfall sensing on paths across campus is the logistical difficulty of providing a wired network connection to externally mounted boxes. Instead, we propose to use wireless ad-hoc networking where we can (with only a subset of the Pis having wired network connections).

Edge computation is supported in our system. This might include per-node event triggers, that are defined and monitored on each supersensor device directly. Further compute tasks might include local aggregation of data such as map/reduce [7] or per-node user queries executing on-device.

The most significant advantage of using general-purpose compute devices for sensor infrastructure is that they are remotely reprogrammable, which allows us to retarget the sensing infrastructure, either using new sensors or employing existing sensors in new ways.

III. Software Platform

We have implemented a distributed data gathering application using a flexible micro-services architecture and where appropriate making use of well-known Linux idioms and existing open source software. Figure 2 shows the implemented services. They are partitioned into those intended to run on each sensor node, and those providing a centralized service. As all services are configured with a network address this distinction is flexible, and they can be re-deployed easily. Each is registered with `systemd`, providing a reliable restart and configuration mechanism. On the server side, we connect our own applications to proven scalable open source software for a persistent document data store (MongoDB), and a performant publish-subscribe system, cache, and queuing system (REDIS). We make use of Protocol Buffers for serialising and parsing transmitted data using the `proto3` format to specify message types and service endpoints. Using these service contracts, we can later implement different services in different languages if needed and maintain interoperability. Our services are currently implemented in Python as this language is well supported on the Raspberry Pi, and provides libraries for easily connecting to the various hardware interfaces required by our choice of sensors, including I2C, SPI, and general-purpose IO pins.

Each sensor node has one required service: The *collector* buffers data received from each of the optional *sensor* services and forwards it to the *measurements* server at a configured maximum frequency. A *heartbeat* mechanism provides monitoring of the sensor node health and allows the server to reply with updated configuration for the collector and sensors as required. Connections are initiated by the distributed nodes.

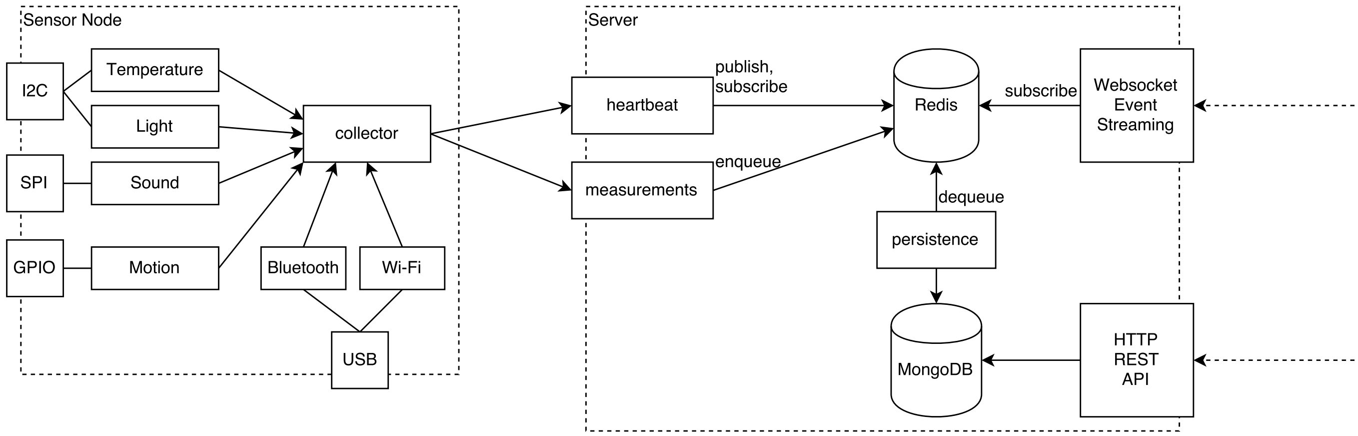

Fig. 2. A flexible micro-services architecture connects sensors to central server.

This way, losing network connectivity is not a problem since data is stored locally, and the only well known name on the network that has to be configured is the main server's address.

The server services, REDIS, and MongoDB are currently all running on a single commodity server. The use of a publish-subscribe model for real time event streams and work queues for data storage and processing tasks, make it easy to add load balancing and distribute the data store across a cluster of multiple machines.

Users of our data are provided with two read only interfaces: An *HTTP REST API* allows for polling current and historical node and sensor data. A streaming *websockets* server provides a near-realtime stream of sensor change events and changes in a node's availability. Due to the flexible publish-subscribe model, we can easily add more specialized tasks that publish higher level events as we discover new use cases and data processing opportunities. We also plan to add an interface for autonomously reconfiguring the sensor nodes, e.g. to adjust the measurement frequency of specific sensors in response to certain events.

Our flexible and well-specified service architecture implements a test-bed for prototyping new applications and trying out various sensors. The system is designed to cope well with both scaling up and future additions: We plan to extend our Bluetooth sensor usage to collect and forward data from less powerful beacons or battery powered sensors in the vicinity. This will be closely integrated with the ad-hoc mesh network for sensors without their own reliable internet connection. We also anticipate using the 'spare' cycles on the Raspberry Pi devices for edge computing experiments and using our existing monitoring infrastructure to record utilisation data and processing results from these.

IV. Use Cases

In this section, we outline a selection of use cases for the smart campus. We report which of these have been implemented already. The other use cases are still under development at Glasgow.

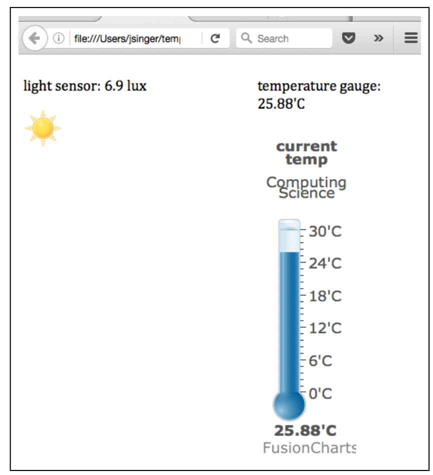

Fig. 3. Web dashboard showing environment state of an office

A. Room Temperature Monitor (done)

Several staff offices have been equipped with supersensors featuring temperature and light monitors. These simple environmental sensors log readings with a frequency of 0.3Hz. Figure 3 gives a webpage rendered view of the sensors.

Studies show that environmental factors like temperature [8] and air quality [9] can have significant effects on the productivity of office workers. Our web dashboard allows users to identify optimal rooms for hot-desk working. It also allows managers to monitor current environmental conditions in the building.

B. Free Meeting Rooms (in progress)

We have installed a supersensor node with sound and motion sensors in each public meeting room in the School of Computing Science. Note that the sound sensor monitors a sound envelope rather than a true audio stream. Generally, public meeting rooms should be booked via a shared calendar system. However the calendar data is often unreliable: sometimes ad-hoc meetings occur without bookings if a room is seen to be empty, and at other times scheduled meetings may be cancelled without rescinding the calendar booking.

A remotely accessible sound sensor in each room allows building users to find a meeting room that is currently free. We define 'free' as a meeting where the sound level is lower than a certain threshold, and has been below this threshold value for a few minutes.

We are currently developing a smartphone app that superimposes meeting room sound levels on top of a schematic plan of the building. This app has been requested by end-users, and we will evaluate its deployment in detail.

One possible extension of this use case involves machine learning. We intend to *train* each supersensor to recognise different kinds of events based on different sound envelope patterns, i.e. acoustic event detection [10], Ideally, the training would be localized on each node due to the varying acoustics of each room. This would require significant compute capability, which is provided by the powerful ARM supersensor processor. It may be preferable to extract features from the true sound signal rather than the envelope, to achieve better event recognition accuracy. We will revisit this decision at a later date. The audio processing could take place on each supersensor node, to avoid the privacy implications of transmitting sensitive audio recordings across the network.

C. Room Occupancy Census (to do)

The estates management team at Glasgow is keenly interested in building utilization. They regularly monitor whether teaching venues are in use, and if used, whether they are occupied by an appropriate number of people. For instance, it is inefficient to hold a class for 10 students in a 200-seat lecture theatre.

A summary of recent studies [11] suggests that 20-40% utilization is normal in a UK university. The UK Higher Education Space Management Group (SMG) is a sector-wide working committee that studies the problem of utilization. Their report [12] presents data from a range of UK universities, with the summary statistic that 27% utilization is the sector average—this implies that the university provides $3.7\mathrm{m}^2$ of space for every $1\mathrm{m}^2$ actually in use.

The SMG report acknowledges that the data is incomplete and expensive to collect. A local census at Glasgow to monitor utilization occurs once per semester. The management reserves the right to penalize under-utilization on the part of individual departments. Generally, a university administrator with a clipboard traverses the campus and makes manual observations during actual lecture events. In a smart campus, it would be possible to use embedded sensors to detect whether a particular room is in use, and how many people are in that room. Binary room utilization might be as simple as checking the value of a sound sensor or motion sensor. More fine-grained measures of occupancy levels would require complex sensor fusion, possibly based on counts of unique wifi MAC addresses or motion sensors. Some researchers have investigated using audio stream processing to infer the number of people in a room [13] but this generally depends on multiple individuals conversing during a meeting, rather than a single lecturer delivering a monologue. Ideally, we could use machine learning to tune such occupancy prediction over time. It would be sensible to *train* a supersensor, by supplying oracle occupancy values during an initial learning phase.

D. Custom Event Triggering (to do)

So far, we have incorporated five different sensors (light, temperature, motion, sound and wifi) with individual sensor nodes. In the previous use cases, sensor data is relayed to the server, where event processing occurs for end-user applications. However it is also possible to support per-node complex events. These events might be defined by individual users for particular localized nodes. Example might be a sudden increase in sound, light and temperature, which could indicate a fire or an explosion.

The predefined combination of events could be logically checked on the local node, then if the event is triggered, local computation could be executed, e.g. issue alarm via email or tweet. This is similar to existing 'if this then that' services, but the logic and processing is local to the sensor node, rather than executing on the server.

There may be potential issues due to nodes that are overloaded with queries. This is a possible vector for denial of service attacks. We need to determine managed approaches to event trigger registration. Another possibility is to offload query processing to nearby devices, i.e. dynamically shift the compute load. This is the approach of cyber foraging [14].

E. Robotic Support Infrastructure (to do)

In mobile robotics applications, it is useful to provide precomputed maps of the local environment. Our supersensor nodes can provide a distributed infrastructure for storing and broadcasting availability of local data sets. For example, a robot might create a map of a room's environment during its first visit, and store this at the local sensor node.

Furthermore, the live sensor status data might also advise autonomous robots to avoid or prioritize different rooms depending on how crowded they are.

Robots and other mobile devices may also use our planned ad-hoc networking facilities to access computational offloading or other remote services.

V. Related Work

Saffo [15] identifies sensors as the basis for a decade of technical innovation. He refers to 'smart artifacts'. To develop this further, we consider the campus to be a 'smart environment'. Kyriazopoulou [16] reviews the notion of a smart city and presents alternative architectural approaches to realizing a smart environment. Our solution maps onto the IoT category, in her review. Szabo et al. [17] present the notion of a smart campus. However they use this term to refer to students that engage in participatory sensing to crowd-source information about lecture timetables and event information. On the other hand, we focus on environmental (rather than academic and organizational) data with our sensors.

Kukka et al. [18] describe a study to predict users' information needs in a smart city. They ask users which information would be valuable, then build a system and observe which information is accessed most frequently. News and maps appear to be widely used; travel and event information much less used. We intend to conduct a similar study to determine campus users' information requirements.

Yoneki [19] presents a sensor system build around Raspberry Pi devices. Her key constraint is that she deploys this system in a remote rural environment. Her nodes are battery powered and use delay-tolerant networking to transfer data between nodes via data mules. Many of her solutions are also applicable for campus network deadspots in our system. Banerjee et al. [20] use Raspberry Pi boards for medical sensing applications. They justify the use of Raspberry Pi devices because of their small footprint and low cost, combined with relatively high compute performance. Ferdoush and Li [21] describe a wireless sensor network deployment using low-cost Arduino and Raspberry Pi boards. They contrast their solution with earlier TinyOS-based research platforms, which are much more difficult to configure and expensive to develop. Unlike our system, they use the Raspberry Pi boards as base stations for remote sensors which transmit sensor data to the Pi devices via Zigbee.

Wang et al. [22] describe the StudentLife system, which monitors student wellbeing on campus. Each student has a mobile phone which captures sensor data such as sound, light and motion. This data is fed into behavioural classifiers, which are then correlated against student grades and self-reported stress levels. The authors report significant correlation between captured sensor data with student stress and academic performance. Our system is much more coarse-grained, in that we are capturing data at room-level rather than student-level. However it could be used to perform similar longitudinal academic studies.

VI. Conclusion

We have outlined the motivation for supersensors, based on inexpensive Raspberry Pi devices attached to off-the-shelf sensors. These supersensor nodes are currently being deployed on the campus at the University of Glasgow, to support a range of use cases. The sensor data is stored in a NoSQL database server, and may be queried via a RESTful API. Our currently deployed system includes ten nodes deployed in a single university department. The typical network traffic generated by each node is 0.2 KB/s, the database grows by 11 MB per day. A variety of end user apps should benefit university administrators, students and staff alike.

References

[1] S. Dawson-Haggerty, A. Krioukov, J. Taneja, S. Karandikar, G. Fierro, N. Kitaev, and D. Culler, "Boss: Building operating system services," in *Proc. 10th USENIX Symposium on Networked Systems Design and Implementation*, 2013, pp. 443–457.

[2] A. Rowe, M. E. Berges, G. Bhatia, E. Goldman, R. Rajkumar, J. H. Garrett, J. M. F. Moura, and L. Soibelman, "Sensor andrew: Large-scale campus-wide sensing and actuation," *IBM Journal of Research and Development*, vol. 55, no. 1.2, pp. 6:1–6:14, 2011.

[3] E. Upton, "Raspberry pi 3 on sale now at $35," 2016, https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/.

[4] ——, "Raspberry pi zero out now," 2016, https://www.raspberrypi.org/magpi/raspberry-pi-zero-out-today-get-it-free-with-the-magpi-40/.

[5] T. Zia and A. Zomaya, "Security issues in wireless sensor networks," in *Systems and Networks Communications, 2006. ICSNC '06. International Conference on*, Oct 2006, p. 40.

[6] A. Soppera and T. Burbridge, "Maintaining privacy in pervasive computing — enabling acceptance of sensor-based services," *BT Technology Journal*, vol. 22, no. 3, pp. 106–118, 2004.

[7] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[8] O. Seppanen, W. J. Fisk, and Q. Lei, "Effect of temperature on task performance in office environment," 2006, http://escholarship.org/uc/item/45g4n3rv.

[9] P. Wargocki, D. P. Wyon, and P. O. Fanger, "Productivity is affected by the air quality in offices," in *Proc. Healthy Buildings*, 2000, pp. 635–640.

[10] A. Temko, C. Nadeu, D. Macho, R. Malkin, C. Zieger, and M. Omologo, *Computers in the Human Interaction Loop*, 2009, ch. Acoustic Event Detection and Classification, pp. 61–73.

[11] C. Beyroughy, E. K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A. J. Parkes, "Towards improving the utilization of university teaching space," *Journal of the Operational Research Society*, pp. 130–143, 2009.

[12] Space Management Group, "Space utilisation: practice, performance and guidelines," 2006, http://www.smg.ac.uk/documents/utilisation.pdf.

[13] Q. Huang, Z. Ge, and C. Lu, "Occupancy estimation in smart buildings using audio-processing techniques," *arXiv preprint arXiv:1602.08507*, 2016.

[14] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *Personal Communications, IEEE*, vol. 8, no. 4, pp. 10–17, 2001.

[15] P. Saffo, "Sensors: the next wave of innovation," *Communications of the ACM*, vol. 40, no. 2, pp. 92–98, 1997.

[16] C. Kyriazopoulou, "Smart city technologies and architectures: A literature review," in *Smart Cities and Green ICT Systems (SMARTGREENS), 2015 International Conference on*, 2015, pp. 1–12.

[17] R. Szabó, K. Farkas, M. Ispány, A. A. Benczúr, N. Bátfai, P. Jeszenszky, S. Laki, A. Vágner, L. Kollár, C. Sidló, R. Besenczi, M. Smajda, G. Kövér, T. Szincsák, T. Kádek, M. K´sa, A. Adamkó, I. Lendák, B. Wiandt, T. Tomás, A. Z. Nagy, and G. Fehér, "Framework for smart city applications based on participatory sensing," in *Cognitive Infocommunications, IEEE 4th International Conference on*, 2013, pp. 295–300.

[18] H. Kukka, V. Kostakos, T. Ojala, J. Ylipulli, T. Suopajärvi, M. Jurmu, and S. Hosio, "This is not classified: everyday information seeking and encountering in smart urban spaces," *Personal and Ubiquitous Computing*, vol. 17, no. 1, pp. 15–27, 2011.

[19] E. Yoneki, "Raspinet: Decentralised communication and sensing platform with satellite connectivity," in *Proc. ACM CHANTS*, 2014.

[20] S. Banerjee, D. Sethia, T. Mittal, U. Arora, and A. Chauhan, "Secure sensor node with Raspberry Pi," in *Multimedia, Signal Processing and Communication Technologies (IMPACT), 2013 International Conference on*, 2013, pp. 26–30.

[21] S. Ferdoush and X. Li, "Wireless sensor network system design using raspberry pi and arduino for environmental monitoring applications," vol. 34, 2014, pp. 103–110, the 9th International Conference on Future Networks and Communications (FNC'14)/The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC'14)/Affiliated Workshops.

[22] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell, "Studentlife: Assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 3–14.