Sensorium — The Generic Sensor Framework

Albert Rafetseder, Florian Metzger, and Lukas Pühringer
University of Vienna, Austria
albert.rafetseder@univie.ac.at

Yanyan Zhuang
University of Victoria, Canada
yyzhuang@cs.uvic.ca

Justin Cappos
Polytechnic Institute of NYU, USA
jcappos@poly.edu

***Abstract*—This contribution describes Sensorium, our framework for accessing sensor values on computing devices and making them available to other applications. Meanwhile, it allows users control the exposure of privacy-related data. Our goal is to bring the sensing capabilities of modern devices to a broader range of reseachers and experimenters via an open source framework. We also present a real application making use of Sensorium's virtues: For our web service Open3GMap, we crowd-source radio reception quality measurements in 3G networks. We combine the data into an open geo-information system.**

I. Introduction

Modern computing devices such as smartphones, laptops, and tablet computers are equipped with an increasing number of sensors: GPS, tilt, and acceleration meters quantify the physical position and orientation of the device; 3G, WiFi, and other interfaces gather data on the availability and signal quality of wireless networks; temperature and ambient light sensors deliver additional insight into a user's work and home environments.

On the other hand, different devices and platforms such as Android and iOS use very different interfaces into their sensors; privacy is another issue hardly tackled on any platform other than in a crude binary (allow/deny access) way.

Therefore, in this demo we introduce Sensorium, a generic sensor reading framework that funnels data from actual sensor drivers, implements fine-grained privacy control for the user, and provides generic outbound interfaces such as XML-RPC. We also show an application using it, Open3GMap, which visualizes mobile coverage data coming from Sensorium.

Sensorium can access all the information a device provides and makes them available to other applications. Up until now, it has been a challenging task for software developers (especially scientists and experimenters) to implement specialized sensor applications. Sensorium simplifies this task by providing a generic framework for interfacing sensors. In our current implementation, available for Android, most of the typical sensors are already implemented. Since giving access to sensor data also exposes the user's privacy, Sensorium displays all sensor readings that would be shared, and lets the user disable or set privacy levels for each sensor individually.

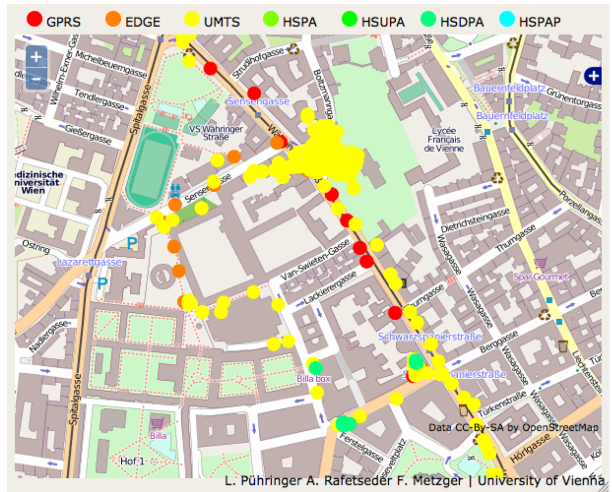

Fig. 1. The Open3GMap web page, displaying 3G coverage measurements extracted from Sensorium on top of OpenStreetMap.

Open3GMap[1] showcases the sensors framework. It comprises a web service displaying cellular access technology data points at their GPS locations collected by devices running Sensorium (see Figure 1). This solves a real-world problem: Currently, this kind of data is only available to mobile operators, which however are hindered by commercial interests to make them publicly available – at least in raw, unadorned form. Other projects such as OpenSignalMaps[2] and Sensorly[3], as well as corporations like Google and Apple collect these data, but are very restrictive regarding usage by other parties. This is not true for Open3GMap: We make the data points collected available as Open Data.

Obviously, other applications are possible. Since code and data are open-sourced, everyone can implement their great ideas.

II. Architecture

Figure 2 overviews Sensorium's architecture components. Sensorium implements *sensor drivers* on top of the operating system that take care of reading sensor values off platform-specific interfaces and push them upwards into the *registry*. Here, sensor data are timestamped and collected. On one side, data are prepared for local display, e.g. in a GUI or status widget. On the other side, a user-configurable *privacy layer*

[1] http://homepage.univie.ac.at/albert.rafetseder/o3gm

[2] http://opensignal.com/

[3] http://www.sensorly.com/

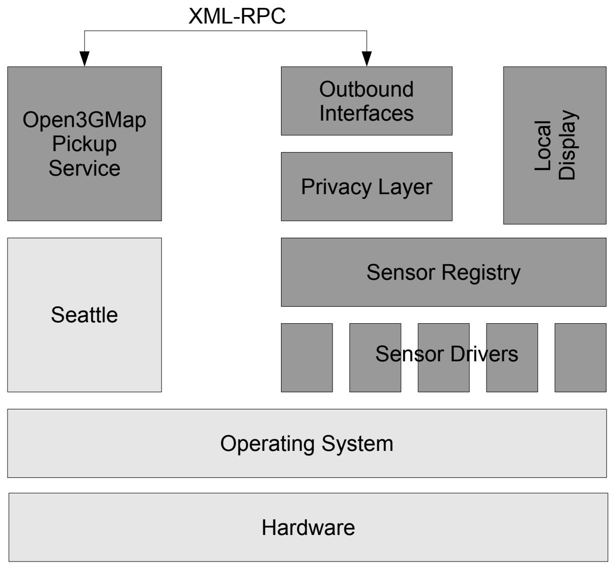

Fig. 2. Sensorium architecture with Open3GMap *pickup* and *server*. Components described in this paper are shown dark gray.

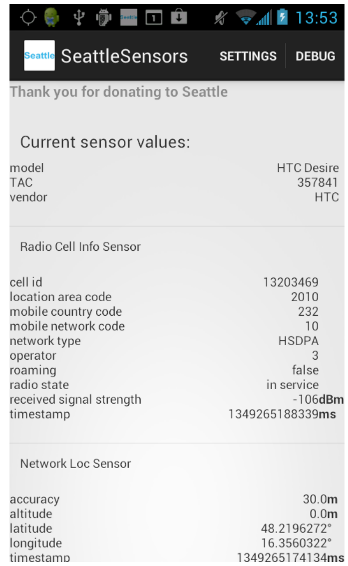

Fig. 3. Sensorium screenshot.

might allow for full sensor access from above or reduce the precision of values (e.g. round GPS coordinates); it could salt and hash sensor values for improved privacy, or completely deny access to individual (or all) sensors. Finally, other applications running on the same device are free to connect to Sensorium's *outbound interfaces* to register for sensor updates or poll data.

Due to the layered architecture, it is very simple for contributors to add their own implementations of layers or swap them out for their own altogether. Consider a scenario when a contributor wishes to include a sensor we do not yet provide a driver for. All that needs to be implemented is code interfacing the actual sensor, and the lightweight API into our sensor registry. Similarly, additional local display methods, privacy enhancements, and outbound interfaces might be implemented.

III. Implementation

Our current implementation of Sensorium[4] runs on the Android platform. To provide a unified interface for accessing the sensor data we incorporated an XML-RPC library[5] that listens for connections on localhost, meaning that only applications running on the same device can access it. The pickup code[6] to collect sensor values runs on top of the renowned Seattle[7] platform, which is also available for Android, and allows us to remotely and securely access the collected data. The example application we implemented to make use of Sensorium, Open3GMap[8], is based on JavaScript and the OpenLayers[9] library. All of our code is dual-licensed under GPLv3 and the BSD license.

[4] https://github.com/fmetzger/android-seattle-sensors
[5] https://code.google.com/p/android-xmlrpc/
[6] https://homepage.univie.ac.at/albert.rafetseder/o3gm_pickup.repy
[7] https://seattle.cs.washington.edu/
[8] https://github.com/lukpueh/Open3GMap
[9] http://openlayers.org/

Sensorium consists of a base registry service with a common interface which each sensor implementation can easily be plugged into. All values are also displayed for the user as seen in Figure 3. The privacy layer automatically anonymizes all gathered values before making them available through XML-RPC in accordance with the user's current privacy settings.

We currently provide sensor implementations for generic device information, e.g. device name and battery status; mobile radio data such as current access technology and cell information; location data provided by the mobile network and GPS; and WiFi and Bluetooth information, including recent scan results.

IV. Demo

During our live demo, the audience will be able to interact with both Sensorium running on devices we will bring along, and with Open3GMap and its database. We will present the graphical interface and configuration options of Sensorium and how privacy settings are reflected in the sensor values available on outbound interfaces. Furthermore, the steps of reading out data through the XML-RPC interface and picking up data by Seattle will be demonstrated. Live data will be fed into and displayed on Open3GMap.

Additionally, we plan to give helpful pointers to start one's own Sensorium-based sensor-reading project in just a few easy steps.

V. Conclusion

Our Sensorium framework attempts to bring the sensing capabilities of current generation devices to a broader range of developers and experimenters. Directly using Sensorium, which is freely available from the Google Play store, or adopting the available source code gives everyone the chance to build projects like the mobile coverage Web service Open3GMap we presented.