

# Demonstration Abstract: Sensor Mockup Experiments with SmartLab

G. Larkou, M. Mintzis, S. Taranto, A. Konstantinidis, P.G. Andreou and D. Zeinalipour-Yazti

Department of Computer Science,

University of Cyprus, 1678 Nicosia, Cyprus

{glarkou, mmintz01, staran01, akonstan, panic, dzeina}@cs.ucy.ac.cy

**Abstract**—In this demonstration paper we present *SmartLab*<sup>1</sup>, an architecture for managing a cluster of both *Android Real Devices (ARDs)* and *Android Virtual Devices (AVDs)* via an intuitive web-based interface. Our architecture consists of several exciting components for re-programming and instrumenting smartphones to perform application testing and data gathering in a facile manner, as well as executing mockup experiments by “feeding” the devices with GPS/sensor readings. We will particularly demonstrate the various components of our architecture that encompasses smartphone sensor data collected by mobile users and organized in our distributed NoSQL document store. The given datasets can then be replayed on our testbed comprising of real and virtual smartphones accessible to developers through our Web 2.0 user interface. We present the applicability of our architecture through various mockup experiments over different application scenarios.

**Index Terms** — mockups, testbeds, mobiles, big-data. (*keywords*)

## I. INTRODUCTION

Re-programming smartphones and instrumenting them for application testing and data gathering at scale is currently a tedious, time-consuming process that poses significant logistical challenges. To this end, we have implemented *SmartLab* [1], a comprehensive architecture for managing a cluster of both *Android Real Devices (ARDs)* and *Android Virtual Devices (AVDs)*, which are managed via an intuitive web-based interface. Our current architecture is ideal for scenarios that require *fine-grained* and *low-level* control over real smartphones [2], [3], e.g., OS, Networking, DB and storage, security, peer-to-peer protocols, but also for scenarios that require the engagement of physical sensors and geo-location scenarios.

*SmartLab* has been inspired by *PlanetLab* [4], which has pioneered global research networks; *MoteLab* [5], which has pioneered sensor network research, EU’s *WISEBED* project [6], which federated different types of sensor networks and *Amazon Elastic Compute Cloud (EC2)*. None of the aforementioned efforts focused on smartphones and thus those testbeds had fundamentally different architectures and desiderata. *SmartLab*’s current hardware consists of over 40 Android devices that are connected through a variety of means (i.e., *wired*, *wireless* and *virtual*) to our *private cloud (data-center)*. Through an intuitive web-based interface, users can upload and install Android executables on a number of devices concurrently, capture their screen, transfer files, issue UNIX shell commands, perform mockup experiments by “feeding” the devices with GPS/sensor readings and many more.

In the context of smartphones, a *mockup* refers to the process of extending an AVD’s or ARD’s particular sensor or

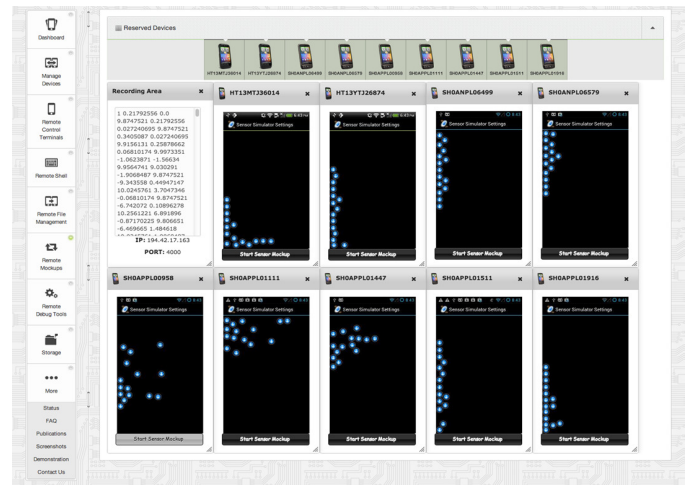


Fig. 1. An example mockup experiment in *SmartLab*, where a user feeds real smartphones with sensor readings from a big-data store and overviews the results through a Web 2.0 User Interface.

GPS with custom values as well as supporting the addition of sensors that may not exist in the hardware of a particular ARD (e.g., NFC). In order to support both GPS and other sensor mockups in *SmartLab* (e.g., *accelerometer*), on both ARDs and AVDs, we opted for a custom module, coined the *Remote Mockup (RM) Library* that shall be thoroughly demonstrated during the demonstration. Figure 1, shows an example scenario where a user feeds nine reserved smartphones with sensor data input (i.e., *accelerometer*, *orientation*, *gyroscope*).

## II. SMARTLAB TESTBED ARCHITECTURE

Our platform consists of several modules that span from the hardware layer, to the device server layer and administrative tools, to the user interface layer and the Remote Mockup Library, in particular, highlighted in this demo.

### A. Hardware Layer and Supported Connection Modalities

*SmartLab* consists of several Android Real Devices (ARDs) and Android Virtual Devices (AVDs), constructed using tools in the Android SDK (e.g., *android create avd*, *mksdcard*). Additionally, it supports a variety of connection modalities: most of the devices are directly connected to our datacenter in ARD-Local mode, utilizing USB hubs, but more smartphones are also connected using the ARD-Remote mode (i.e., *WiFi/3G*). This mode is particularly promising for scaling the testbed outside our Department (e.g., ARD-Internet mode, where latencies span beyond 100ms.)

<sup>1</sup>Available at <http://smartlab.cs.ucy.ac.cy/>

## B. Device Server Layer and Administrative Tools

The Device Server (DS) is a complete Linux OS image having the SmartLab subsystems and ADB installed. Each DS is also equipped with a local web server which is responsible to host the administrative tools required for maintenance purposes similarly to routers and printers. More specifically, SmartLab supports a variety of administrative tools: i) A “wipe” tool, which is able to simultaneously “factory reset” a number of ARD-Local or AVD devices; ii) A “backup” tool, which is able to simultaneously backup all settings, applications and files from an ARD-Local or AVD device and store them in a distributed file system; and iii) A “restore” tool, which is able to restore all settings, applications and files from a backup file (.ab) to a number of supported target devices simultaneously.

## C. User Interface and Data Layers

SmartLab implements several modes of user interaction with connected devices using either *Websocket-based interactions*, for high-rate utilities, or *AJAX-based interactions* for low-rate utilities. In particular, SmartLab supports: i) *Remote File Management (RFM)*, an AJAX-based terminal that allows users to push and pull files to the devices; ii) *Remote Control Terminals (RCT)*, a Websocket-based remote screen terminal that mimics touchscreen clicks and gestures but also enables users recording automation scripts for repetitive tasks; iii) *Remote Debug Tools (RDT)*, a Websocket-based debugging extension to the information available through the Android Debug Bridge (ADB); iv) *Remote Shells (RS)*, a Websocket-based shell enabling a wide variety of UNIX commands issued to the Android Linux kernels of allocated devices; and v) *Remote Mockups (RM)*, a Websocket-based mockup subsystem for feeding ARDs and AVDs with GPS or sensor data traces encoded in JSON for trace-driven experimentation.

## D. Remote Mockup (RM) Library

A mockup provides part of a system’s functionality enabling testing of a design. In the context of Android, mockups refer to the process of extending an AVD’s or ARD’s particular sensor or GPS with custom values. Additionally, one important benefit of mockups is that these can support the addition of sensors that may not exist in the hardware of a particular ARD (e.g., NFC, WiFi Direct, etc).

In order to support both GPS and other sensor mockups in SmartLab, (e.g., *accelerometer, compass, orientation, temperature, light, proximity, pressure, gravity, linear acceleration, rotation vector and gyroscope sensors*) on both ARDs and AVDs, we opted for a custom module, coined the Remote Mockup (RM) Library. Our RM library establishes a socket server on DS feeding devices with sensor or GPS readings encoded in JSON format and stores them in a NoSQL repository. As this functionality is completely outside the ADB interaction stream, we were required to provide each application with a custom library, coined `SLSensorManager.jar`.

The RM library can be embedded to any android application enabling interaction with the SmartLab GPS/Sensor subsystem running on DS. In fact, our library has precisely the same interface with the Android SDK `SensorManager` and consequently a user can override Android’s default behavior as shown next.

- Request Internet permission in the App Manifest:  

```
<uses-permission android:name="android.permission.INTERNET"/>
```
- Instead of invoking the default `SensorManager`:  

```
mSensorManager = (SensorManager)
    getSystemService(SENSOR_SERVICE);
```

use:  

```
mSensorManager = SLSensorManager.
    getSystemService(this, SENSOR_SERVICE);
```
- Connect to the SmartLab’s Remote Mockup using:  

```
mSensorManager.connectSimulator();
```

## III. DEMONSTRATION SCENARIO

**Interactive:** We will start our demonstration out by over-viewing the main components of SmartLab. We shall then present the complete interaction workflow, i.e., allocate devices, transfer files to remotely connected devices and initiate/control demonstration programs utilizing our internally developed modes of user interaction (RFM, RCT, RDT, RS and RM). More specifically, we plan to demonstrate how somebody can test both widely known applications available through the Android market and interesting applications that we have developed in-house for crowd-sourced trajectory matching [3], crowd-messaging applications, fine-grained indoor localization, peer-to-peer search [2] and others.

**Trace-driven:** We will provide pre-recorded sensory readings and a prototype application in order to demonstrate the Remote Mockup (RM) library. Moreover, we will provide a prototype application with recording functionality in order to demonstrate how a user is able to record sensory readings through SmartLab. Finally, we will present a complete tutorial on how the RM library can be integrated into an existing Android application and mock pre-recorded sensory readings to the Android application, similar to Figure 1.

## ACKNOWLEDGMENTS

This work was financially supported by the last author’s startup grant, funded by the University of Cyprus. It has also been supported by MTN Cyprus, EU’s COST Action IC903, EU’s FP7 MODAP project and EU’s FP7 Planetdata NoE.

## REFERENCES

- [1] Georgios Larkou, Constantinos Costa, Panayiotis G. Andreou, Andreas Konstantinidis and Demetrios Zeinalipour-Yazti. “Managing Smartphone Testbeds with Smartlab”, In Proc. of the 27th Intl. Conference on Large Installation System Administration, USENIX, *LISA’13*, 115–132, 2013.
- [2] Andreas Konstantinidis, Demetrios Zeinalipour-Yazti, Panayiotis G. Andreou, Panos K. Chrysanthis, and George Samaras. “Intelligent Search in Social Communities of Smartphone Users”, In Distributed and Parallel Databases, Volume 31, 115-149, 2013.
- [3] Demetrios Zeinalipour-Yazti, Christos Laoudias, Costantinos Costa, Michalis Vlachos, Maria I. Andreou, and Dimitrios Gunopulos. “Crowd-sourced Trajectory Similarity with Smartphones”, *IEEE Trans. on Knowl. and Data Eng.* 25, 6 (June 2013), 1240-1253, 2013.
- [4] Larry Peterson, Tom Anderson, David Culler, and Timothy Roscoe. “A Blueprint for Introducing Disruptive Technology into the Internet”, A blueprint for introducing disruptive technology into the Internet. *SIGCOMM Comput. Commun. Rev.* 33, 1 (January 2003), 59-64, 2003.
- [5] Geoffrey Werner-Allen, Patrick Swieskowski, and Matt Welsh. “Mote-Lab: a wireless sensor network testbed”, In Proc. of the 4th Intl. Symposium on Information Processing in Sensor Networks, *IPSN’05*. IEEE Press, Piscataway, NJ, USA, Article 68, 2005.
- [6] Geoff Coulson et al., “Flexible experimentation in wireless sensor networks”, In *ACM Communications*, Volume 55, Issue 1, 82-90, 2012.