SOUL:
An Edge-Cloud System for Mobile Applications in a Sensor-rich World

Minsung Jang, HyunJong Lee, Ketan Bhardwaj, and Karsten Schwan

October 28, 2016
Washington DC, USA
Presented for The First IEEE/ACM Symposium on Edge Computing

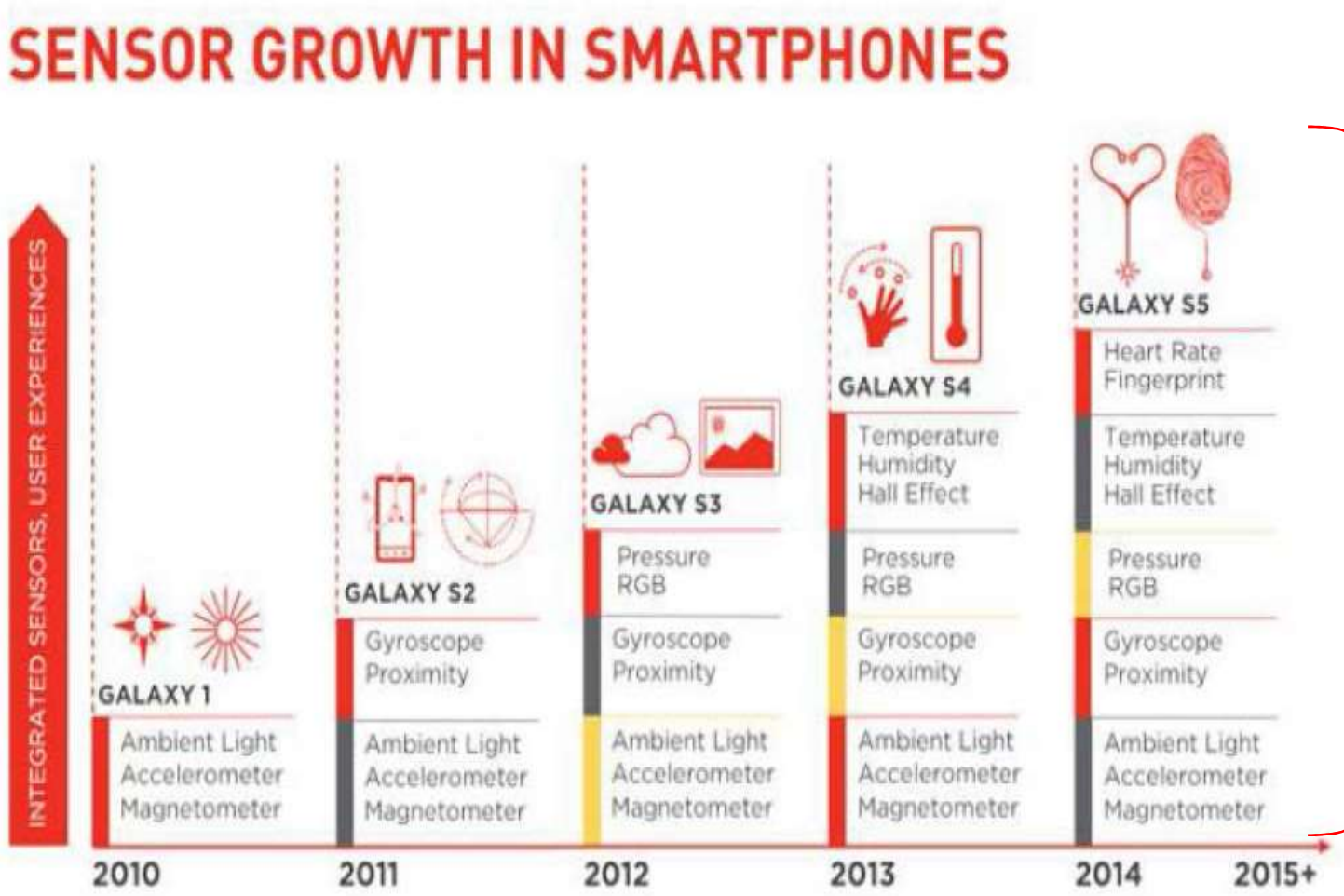
Georgia Tech

cercs

Contents

- Motivation
- Challenges
- SOUL
  - Mobile Apps and Sensors
  - Access Control Service
  - Programming Model
- Evaluation Results
- Conclusions

A trillion-sensor world is coming!



Source: https://www.qualcomm.com/news/snapdragon/2014/04/24/behind-sixth-sense-smartphones-snapdragon-processor-sensor-engine

Georgia Tech

cercs

Motivation

- "Of the hundreds of thousands of apps available in app stores, today, fewer than 0.5 percent employ sensors," (WCA Summit 2012)

- **30% of the top 100 apps in each category (5000 apps) use the Android location service.**
- **Why?**

5	1	17
6	0	126
7	0	2
8	0	60
Total(%)	81 (1.62%)	14427 (1.92%)

Table 1: The number of apps based on sensor use.

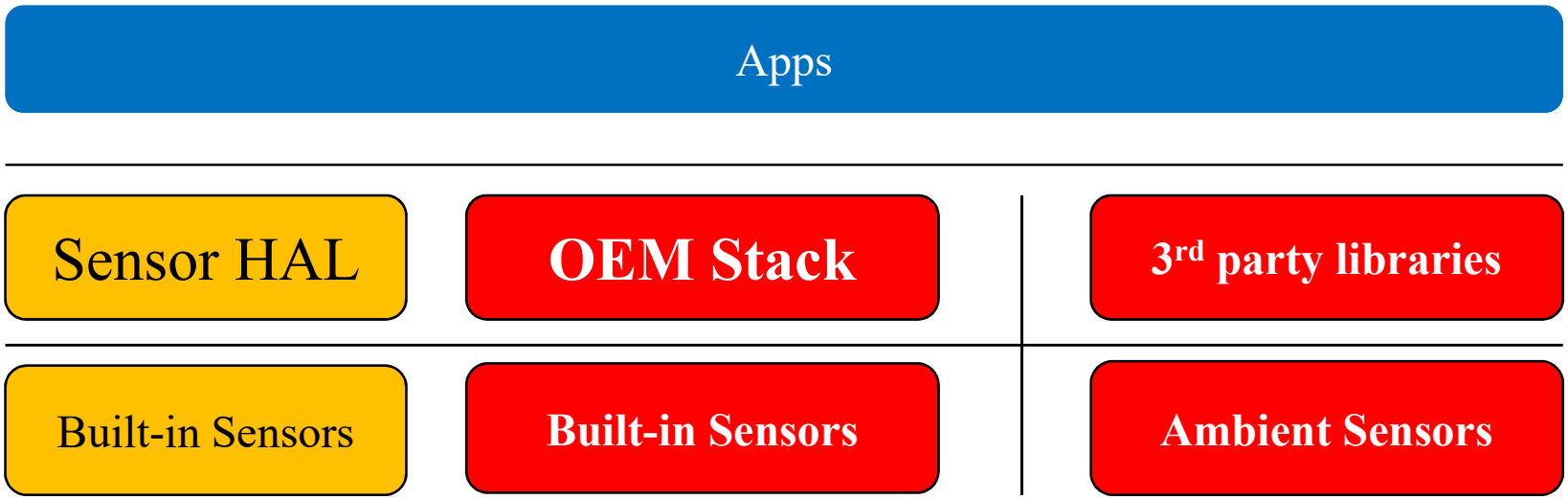
light	6	353

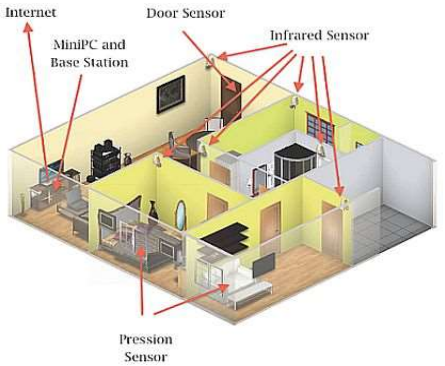
Table 2: The most commonly used sensors. All permissions start with *android.hardware.sensor*.

the numbers retrieved on May 20, 2015

Challenges (1)

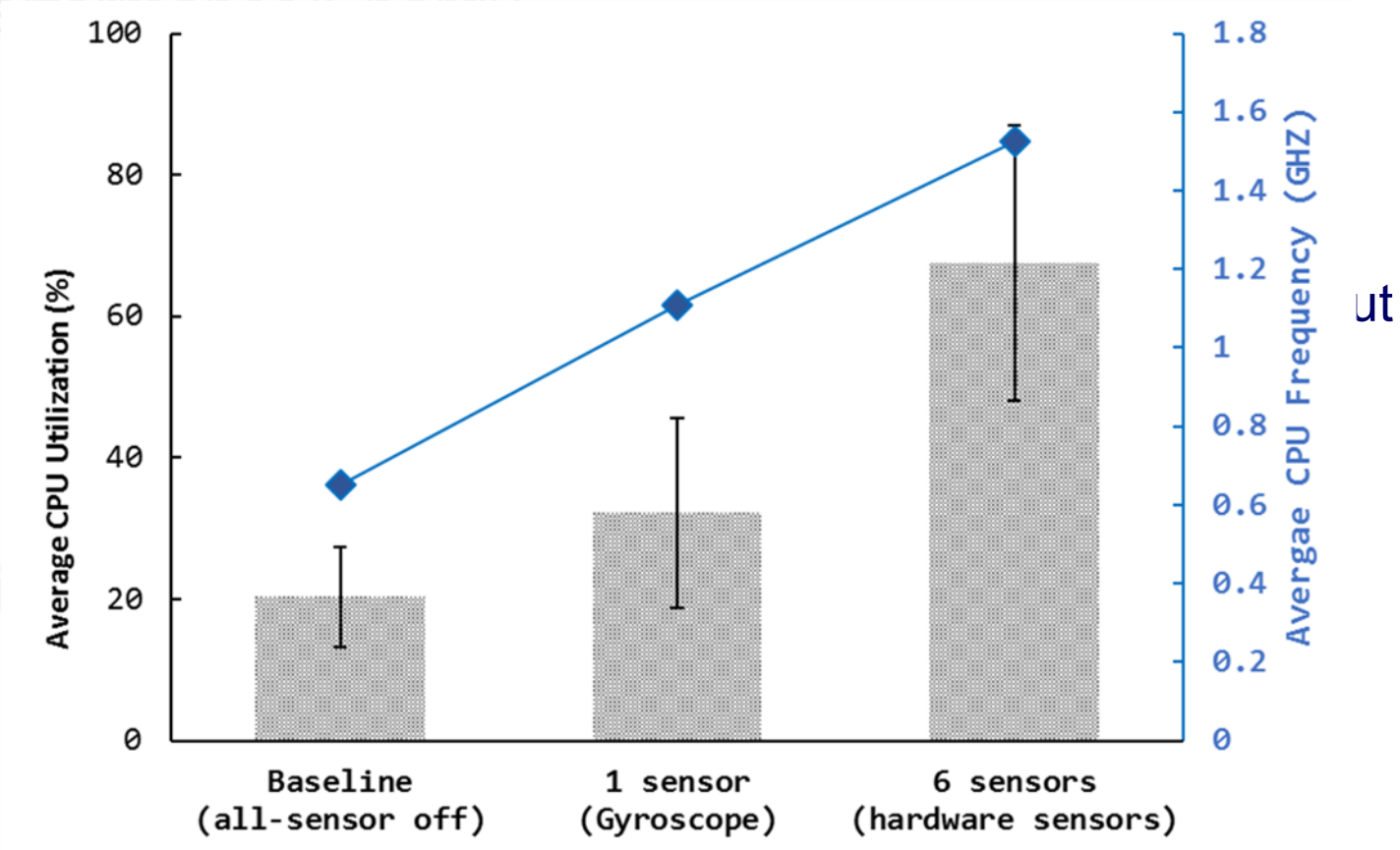
- Fragmented Sensor Ecosystem in mobile platforms
  - backward compatibility/ Existing apps support





Challenges (2)

- Pe
  - –
  - –
  - –
  - –
- M
  - –
  - –



† This measurements used the Perf tool on the Nexus 5 phone with KitKat 4.4)

SOUL – Approach

- Consistent abstraction and APIs: On- and off-device sensors
  - Aggregate: Virtualized instances consisting of physical sensors/actuator and corresponding services
  - The compatibility for existing sensor-based applications
- Flexible Offloading: leveraging edge or remote clouds resources
  - Processing-on-Demand instance
  - Automatic resource remapping when user context is changed
- Access Control at runtime: Using user's social relationship
  - Finer-grained sharing operations: Read vs Glance

SOUL - Design

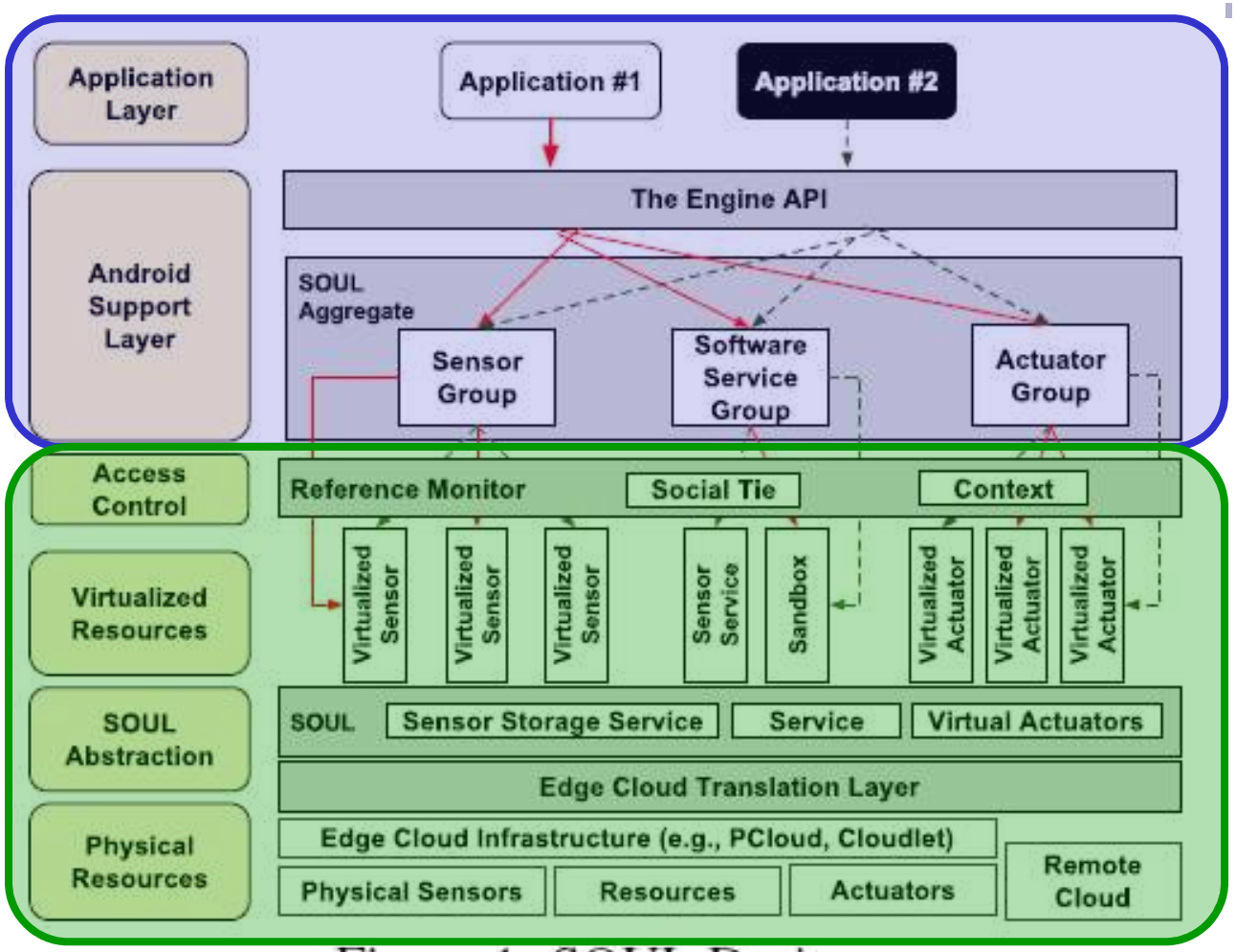


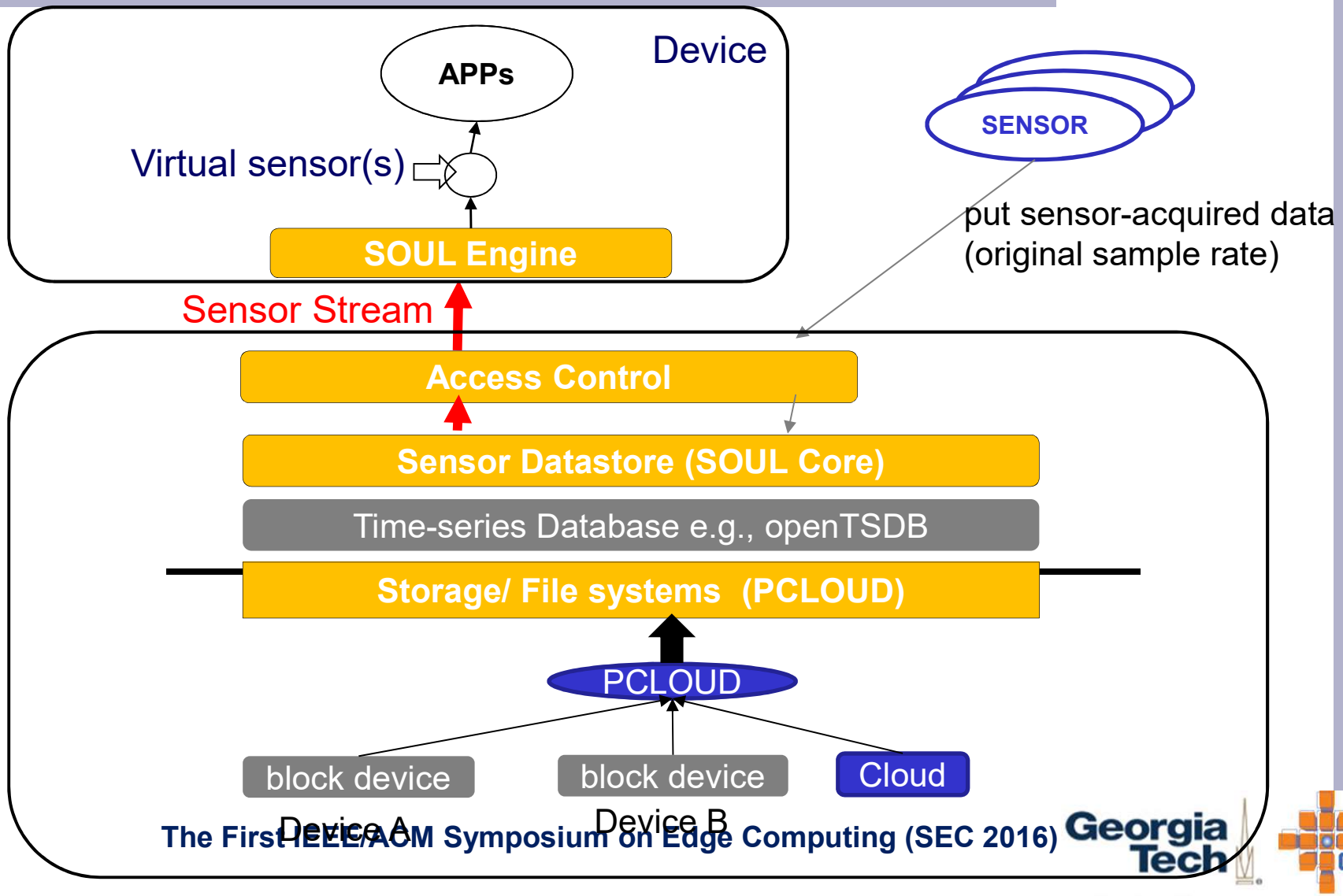
Figure 1: SOUL Design

SOUL Engine on Device

SOUL Core on an edge cloud (PCloud)

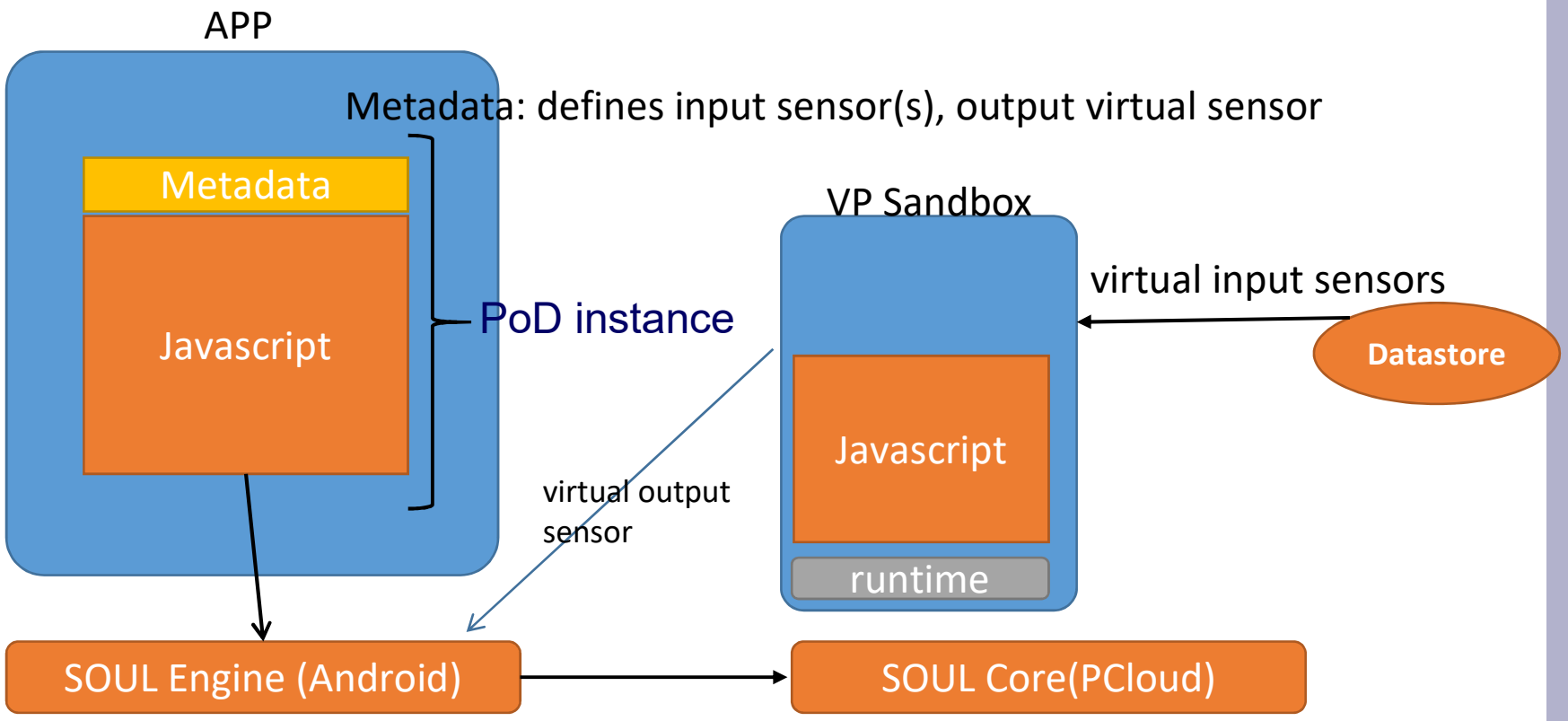
Georgia Tech cercs

Sensor Stream



Processing On Demand

- Inject app's processing code to SOUL



Access Control

- Helping sensor owners properly decide different access privileges of their sensors
  - Scalable solution handling ever-increasing diverse sensors
- Enabling people to easily set up access policies
  - Social relationships in SNS
  - The current context where the request is made
- Reference Monitor and Policy Generator

Access Control: Policy Generator

- Models to predict social ties from the interactions observed in SNS
- SNS to estimate social relationships (social tie)
- Why capturing the context?
  - Not all social relationships can be captured by an SNS
  - Generating the template for guests resulting from the context
- How to determine the context?
  - Location, Common events

Evaluations – Access Control

- Test settings
  - A Facebook account: 2675 postings, 3458 comments and 2270 LIKES
  - Time taken to model this account: 453 minutes

Table: The elapsed time to create a policy

Source	Time in milliseconds
Social Tie-based	6
Context-based	
– Facebook Event	123
– Google Calendar	90

Evaluations - micro benchmark

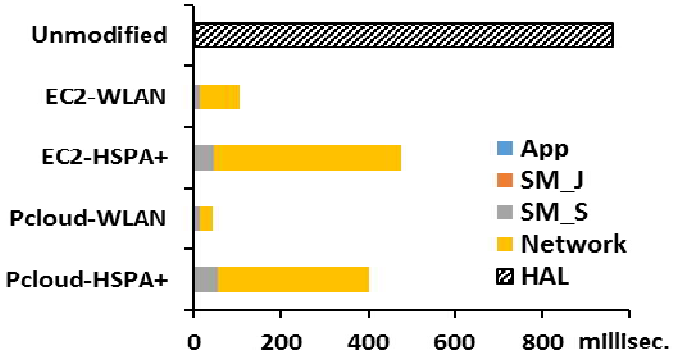


Fig. 1. Elapsed time per layer

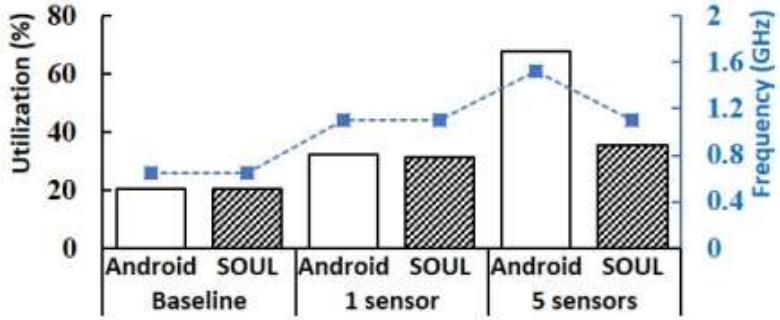


Fig. 2: The CPU overhead in Android vs. SOUL

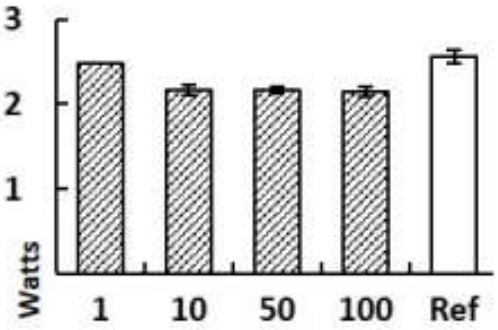


Fig. 3: Power consumption as the number of sensors increases

Evaluation – PoD

- Using the Kalman filter for post-processing of sensor data

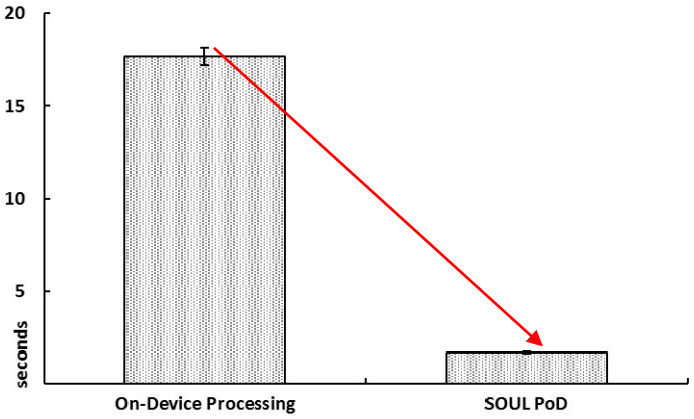


Figure 4. The elapsed time

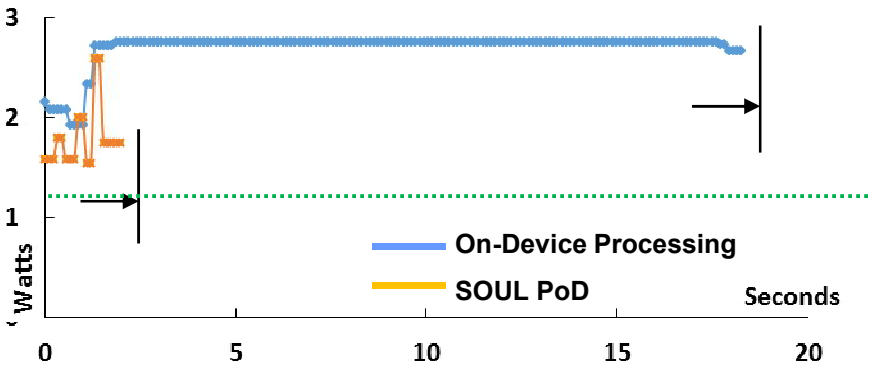


Figure 5. Comparison of power consumption

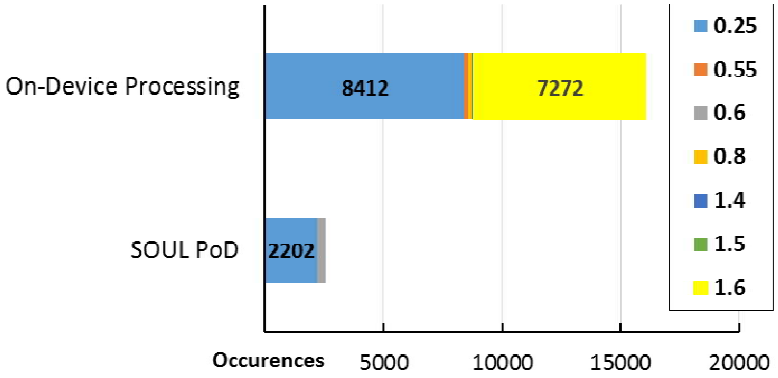
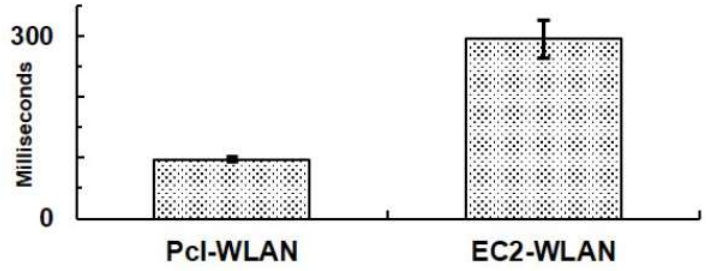


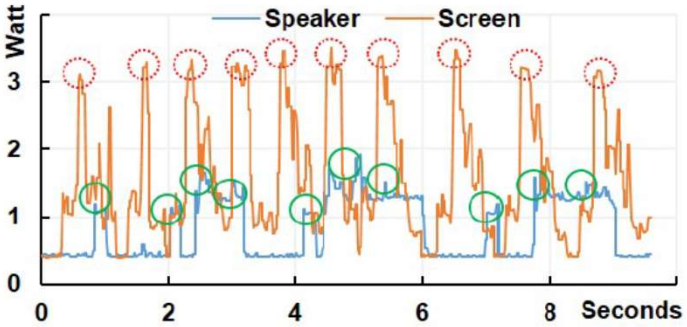
Figure 6. CPU frequency changes

Evaluation – SOUL aggregate

- Don't turn on the screen [Demo]
  - The idea of a SOUL aggregate



(a) Response time to state the current time. Error bars show 95% confidence intervals, repectively.

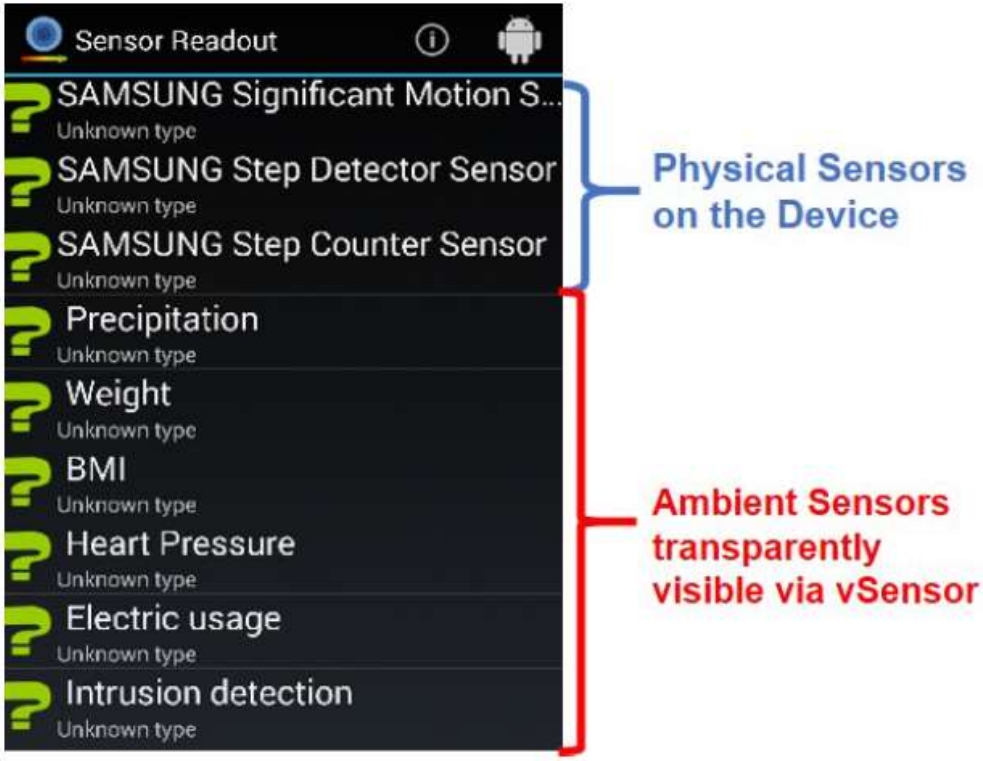


(b) Power consumption

Fig. 4: Results of the 'Don't turn on the screen' application. Concerning total energy consumption, the speaker and the screen consume 22.14 and 40.94 mWH, respectively. Each circle shows the moment that a user acknowledges the current time.

Evaluations – Existing Apps

- Backward Compatibility [Demo]



Conclusions

- SOUL allows apps to easily interact with and leverage available sensors and resources.
- SOUL expands complex sensor processing and integration activities beyond a single smartphone's computing, storage, and battery constraints.

Thank you!