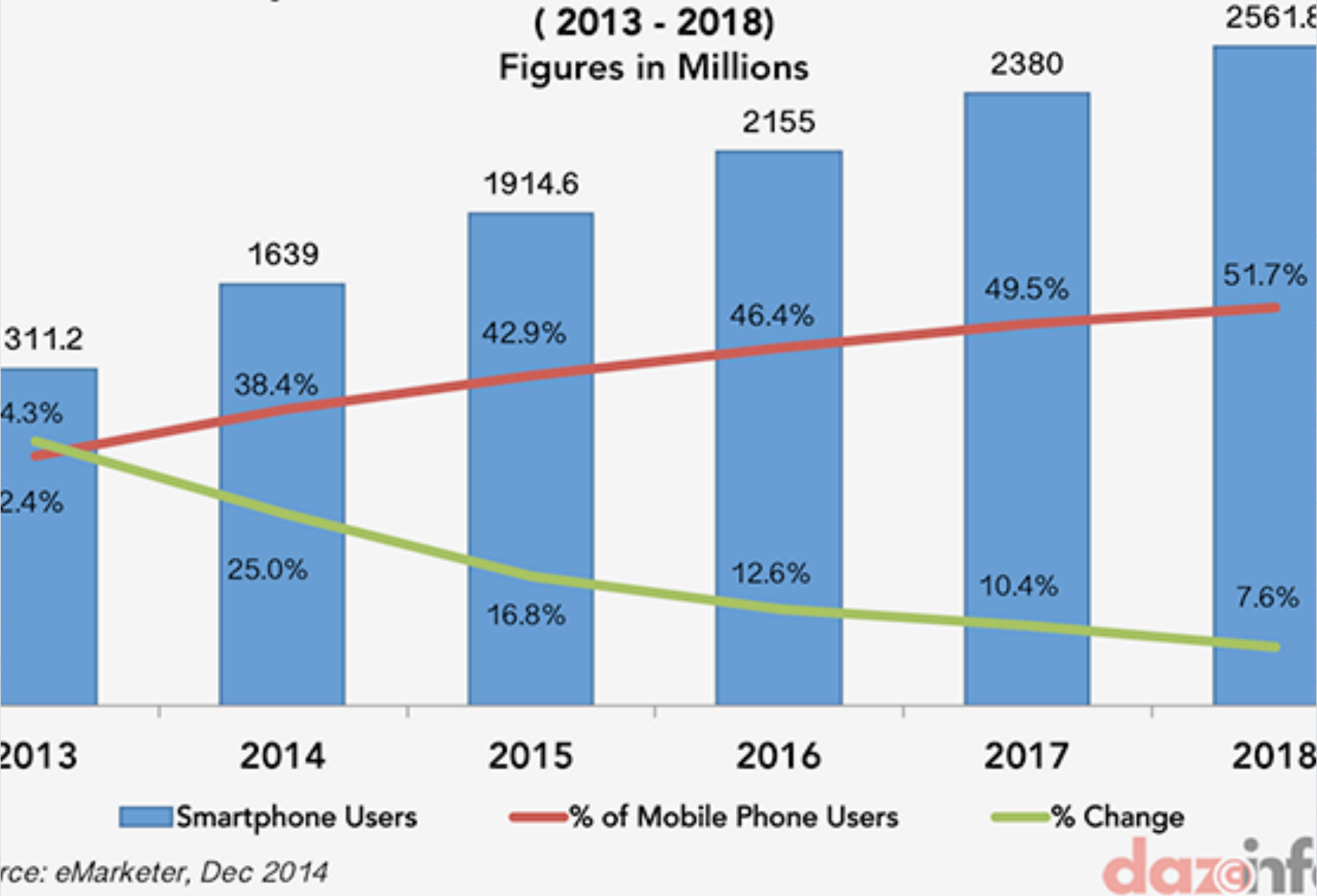
Trust, Privacy and Cooperation With Mobile CrowdSensing

Nirupama Bulusu
Portland State University

Smartphone Users and Penetration Worldwide

(2013 - 2018)

Figures in Millions



rce: eMarketer, Dec 2014

iPhone Sensors

Sensors are vital organs of phones making them smart.

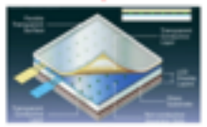
Without them it loses its smartness and metamorphoses into a machine converting sound *- typically and most efficiently the human voice -* **into electronic signals suitable for transmission over long distances.**
Just an innovation of the well-known Smoke Signal technique.

Ambient Light

This sensor can determine how much light is available in the area surrounding the device and automatically adjust the brightness of the screen in order to conserve battery life.

Touch Screen

The capacitive material sends raw touch-location data to the iPhone's processor. The processor uses software located in the iPhone's memory to interpret the raw data as commands and gestures.



Proximity

This sensor can determine how close the iPhone is to your face. This sensor is what helps the iPhone turn off its screen automatically whenever you hold the phone up to your ear. This is necessary to prevent accidental button clicks by the side of your head when talking.

Fingerprint

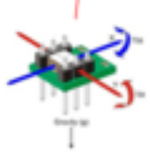
Touch ID is built into the home button, which is now built of laser-cut sapphire crystal so as not to scratch (which would prevent Touch ID from working). It features a stainless steel detection ring to see if the user's finger is there without pressing it.
The sensor will use capacitive touch to detect the user's fingerprint. The sensor has a thickness of 170 µm, with 500 pixels per inch resolution. The user's finger can be oriented in any direction and it will still be read. Apple says it can read sub-epidermal skin layers, and it will improve with every use.

Attitude

The Attitude (rotation) sensor provides the pitch, roll and yaw (azimuth) angles of the device relative to the normal horizon. It is a part of the Device-Motion information computed by the iPhone operating system from the main sensors, particularly the gyroscope.
Although the gyroscope provides precise measurements of the angular rates, calculating rotations only from the gyroscope are subject to a noticeable drift due to various inherent physical phenomena like gyro precession. The Device-Motion computation automatically uses measurements from the accelerometer to minimize the drift.

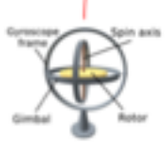
Accelerometer

The accelerometer sensor measures linear acceleration along three perpendicular axes, X, Y and Z. The movement of the mass along the double sided arrow measures the acceleration along that axis.
The linear acceleration sensor measures the acceleration effect of the device movement, excluding the effect of Earth's gravity on the device. It is a part of the Device-Motion information computed by the iPhone operating system from the main sensors, particularly from the accelerometer and the gyroscope.



Gyroscope

The gyroscope sensor measures rotational velocity along the Roll, Pitch and Yaw axes. It depends on the property of rotating mass. Like the accelerometer, gyroscope sensors for mobile devices utilize MEMS technology and may include an accelerometer.

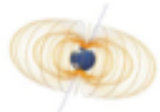


Moisture

The device also contains a sensor which isn't related to the interface. The water sensor is a little red tab that appears in the dock connector when the phone has been submerged in water. It can also appear as a red dot in the headphone jack.

Magnetometer

The magnetometer detects the Earth's magnetic field along three perpendicular axes. In addition to general rotational information, the magnetometer is crucial for detecting the orientation of the device relative to the Earth's magnetic north. Electrical appliances that generate magnetic fields disturb the magnetometer.



Gravity

The gravity sensor measures the Earth's gravity effect on the device. When the device is placed on a flat surface facing up, all of Earth's gravity appears on the -Z axis. Tilting the device "spreads" the gravity into the X and Y axes.
The gravity sensor is a part of the Device-Motion information computed by the iPhone Operating System from the main sensors, particularly from the accelerometer and the gyroscope.
Gravity is obtained when the linear acceleration data is subtracted from the raw readings of the accelerometer.

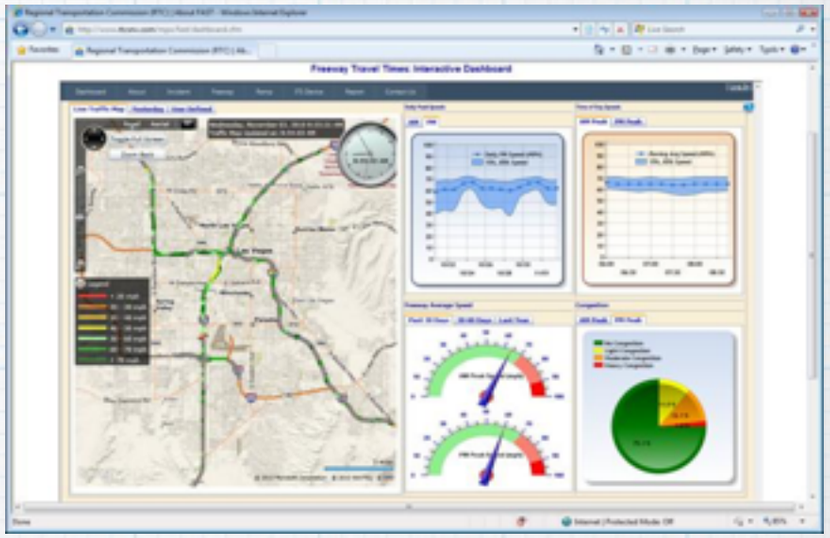
Barometer

The barometer senses air pressure to determine your relative elevation. So as you move, you can keep track of the elevation you've gained. It can even measure stairs climbed or hills conquered.

iPhone 6/6 Plus only

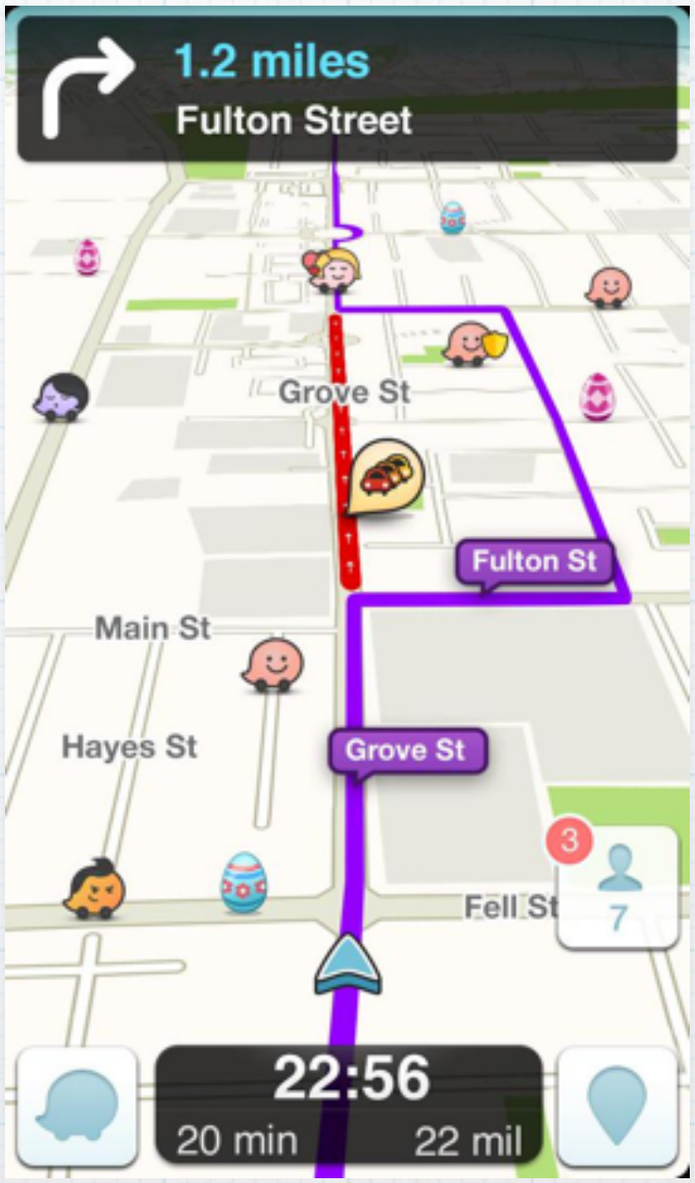
Mobile CrowdSensing

Raw data:
GPS,
audio,
images

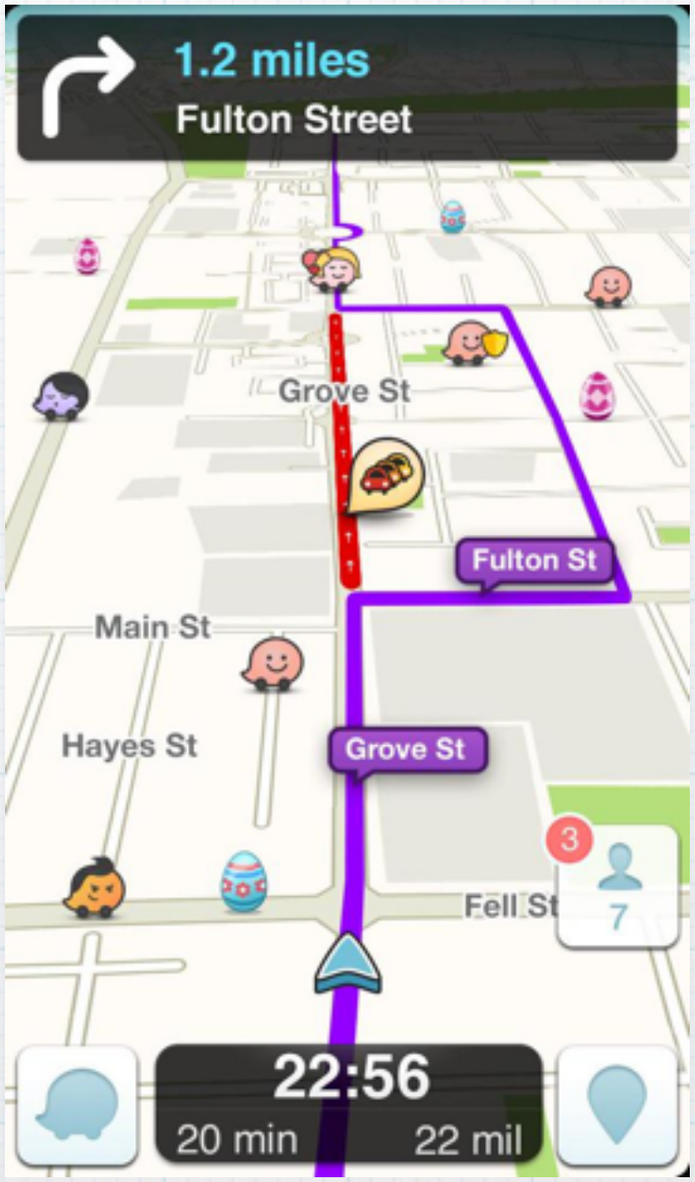


Example: Waze

* Drive, collect road goodies, earn bonus points

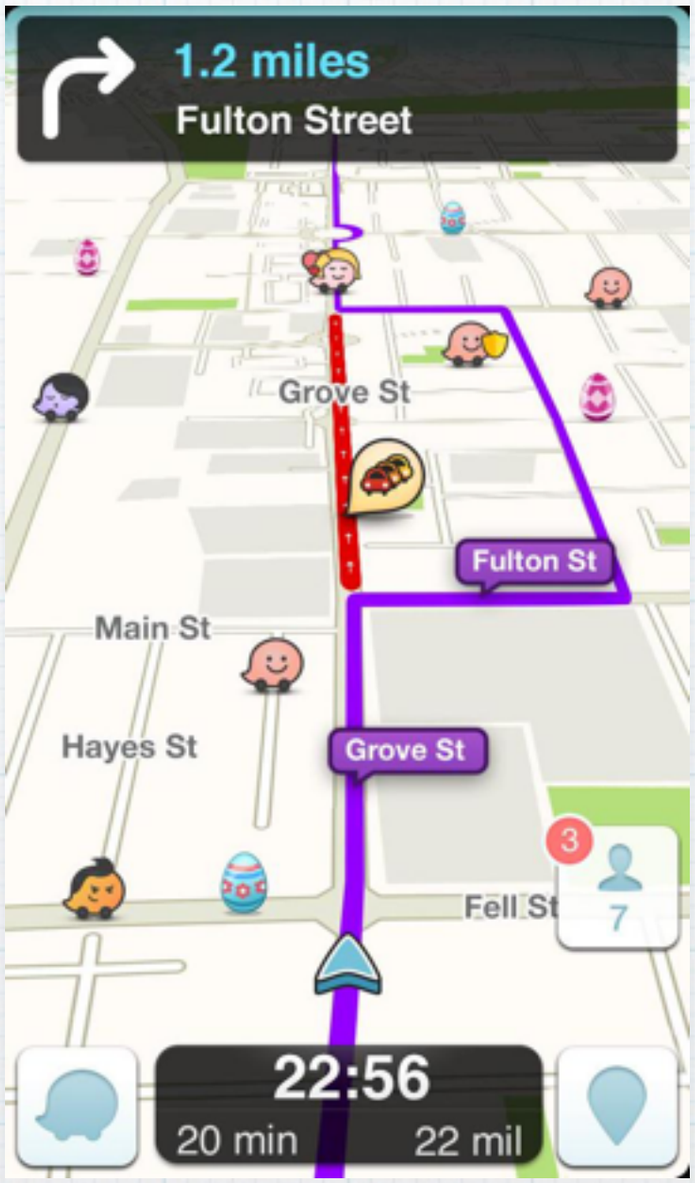


What could go wrong?

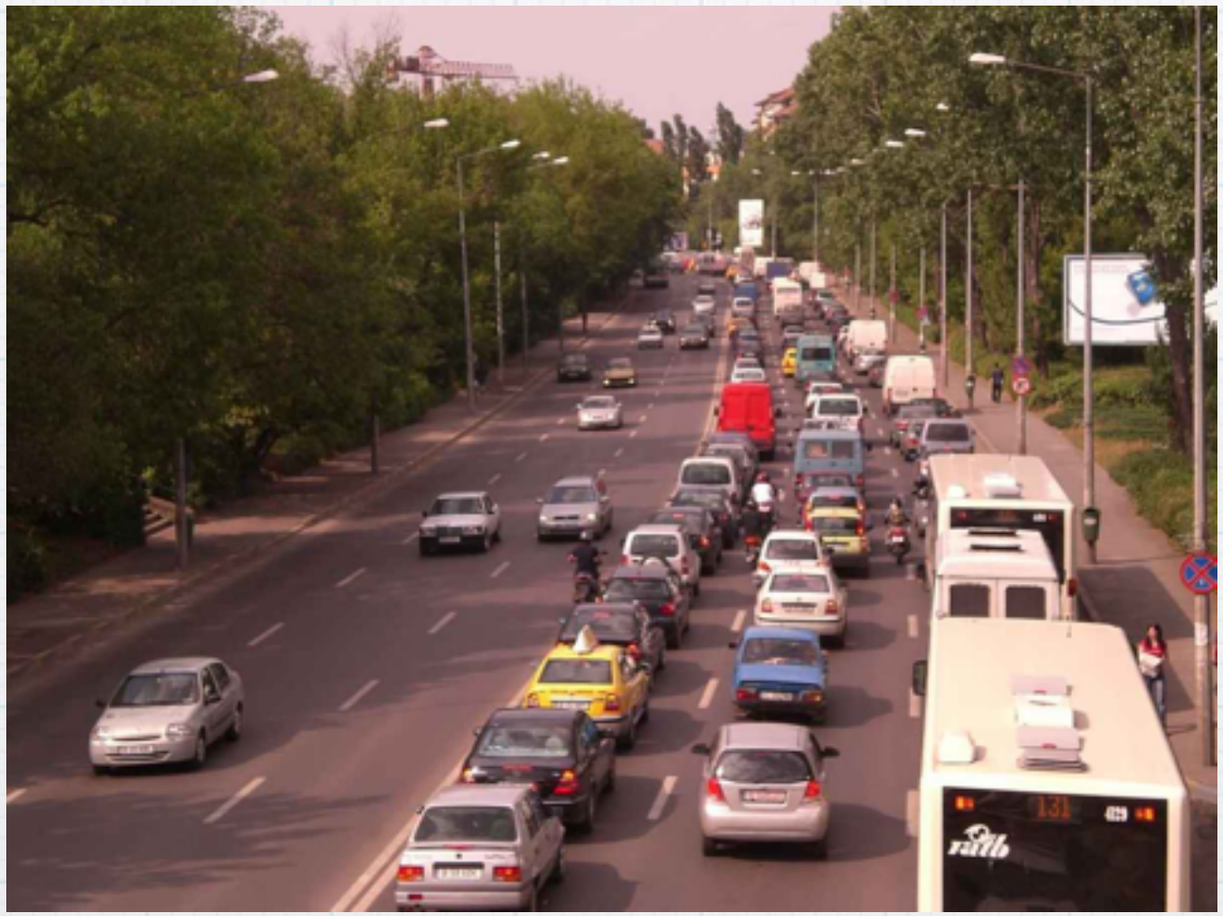


What could go wrong?

* Users could collect road goodies without actually being there!



Do we really care?



"What happens when an app becomes so popular it is basically a public utility? For a school project, Shir Yadid and Meital Ben-Sinai, fourth year students at Technion, hacked the incredibly popular Waze GPS map, an Israeli made smartphone app that provides directions and alerts drivers to traffic and accidents. The students created a virtual traffic jam to show how malicious hackers might create a real one."

-Kelsey Atherton, Popular Science, March 2014.

The Problem

* Online applications that encourage open participation/contribution remain vulnerable to spurious information.

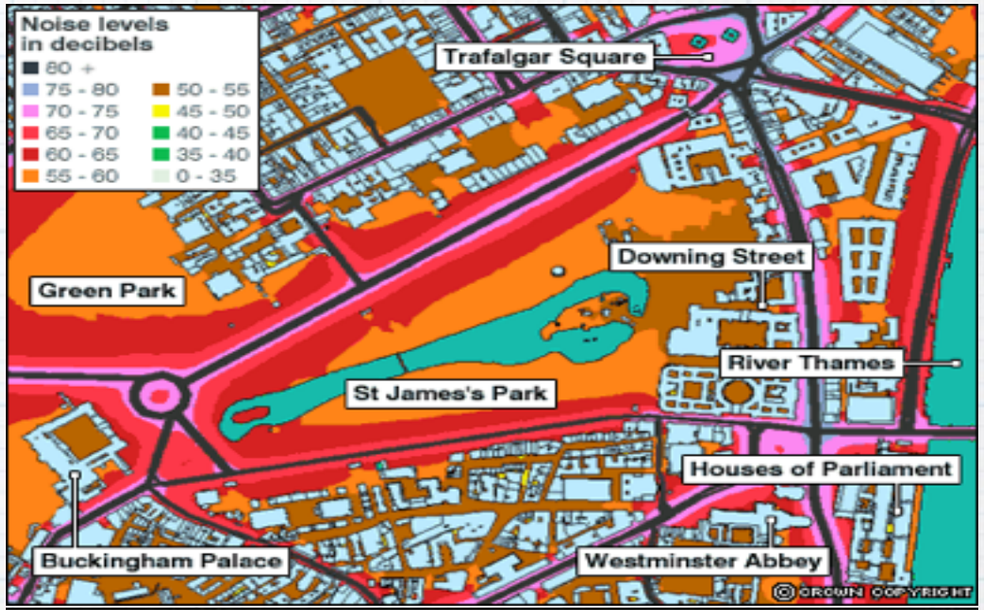
The Problem

* Fabricated data in crowd-sourced sensing applications

* How can a data consumer - receiving data from sensors not under its control - trust that the data is a true representation of the real-world phenomenon being sensed?

My Work

* Urban - Monitoring Noise Pollution (2009 -) [Rana10]
* Commerce - Price Dispersion Monitoring (2007 -) [Bulusu08]
* Current Focus
  * Mobile Health - Lung Sound Assessment, Health Trend Finder



This Talk

* Trust
* Time permitting
  * Privacy
  * Collaboration

Trust & Privacy
(Joint Work with Akshay Dua, Wu-chang Feng & Wen Hu) [Dua09, Dua14]

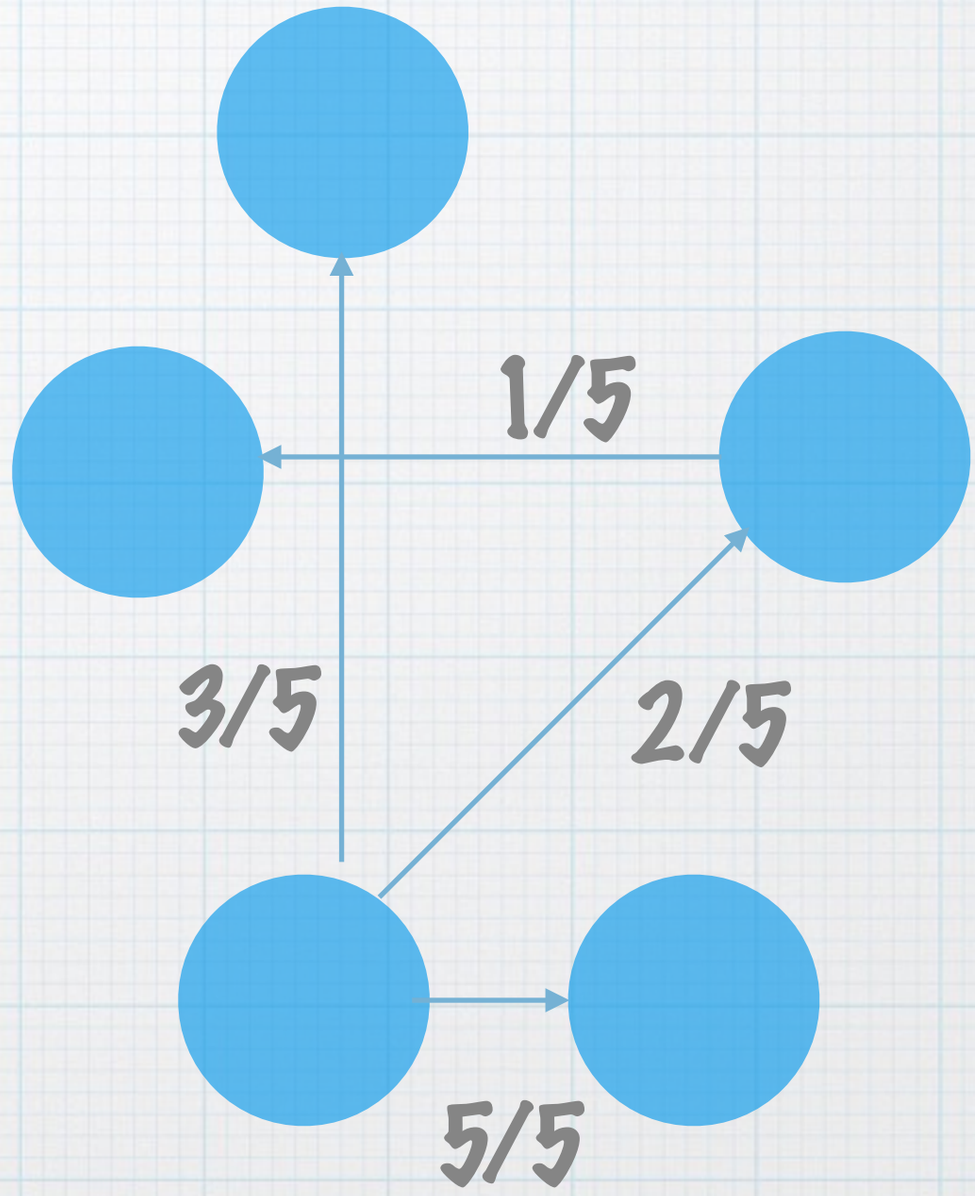
Existing Approaches

* Reputation Ratings
* Anomaly Detection
* Device Monitoring

Reputation Ratings

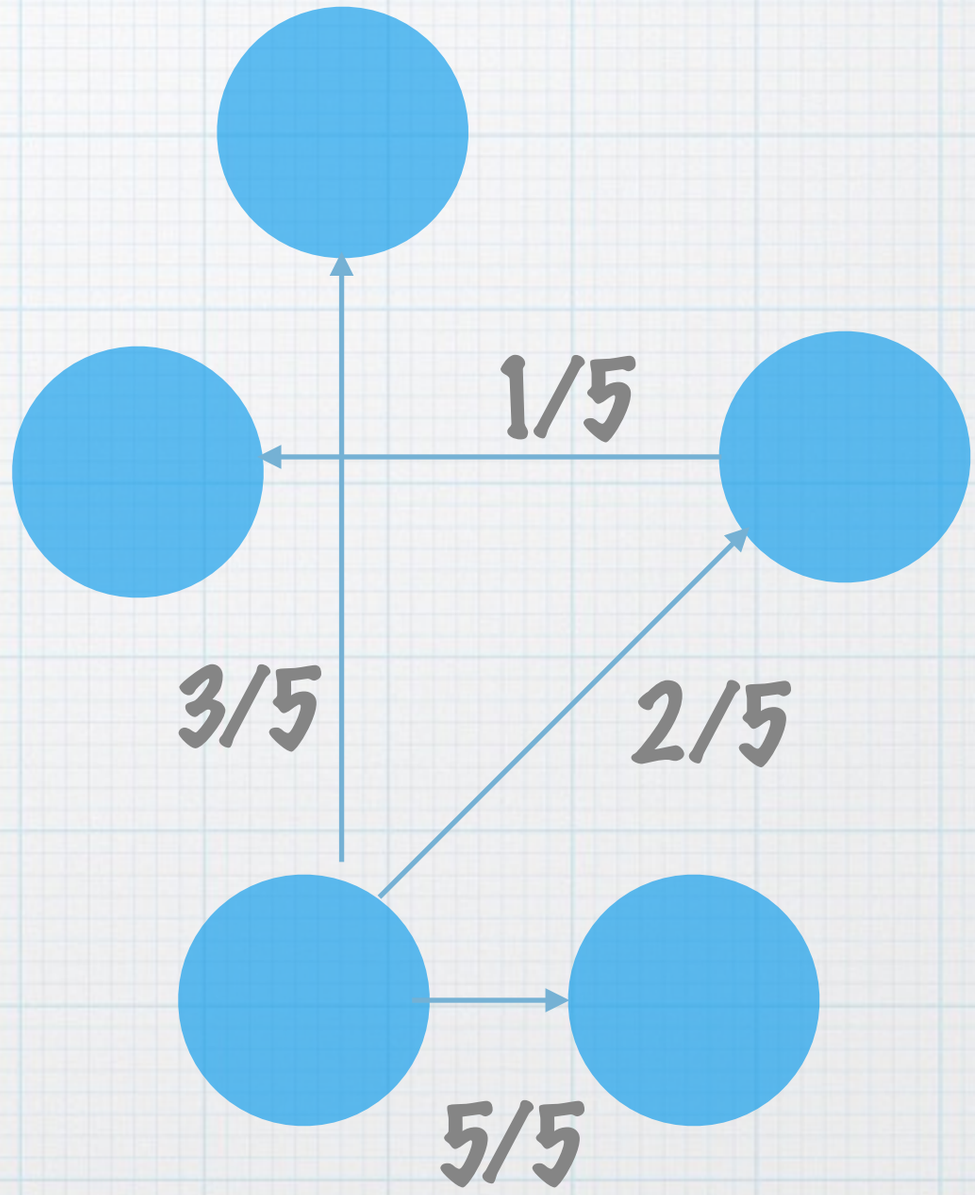
* Users rate each other; information from users with higher ratings considered more trustworthy

* Related Work: [Ganeriwal et al. 2008], [Jang and Ismail 2002], [Liang and Shi 2008]



Reputation Ratings

* Issues:
  * Assumes a model where users interact with each other.
  * Assumes users can correctly judge the integrity of information.

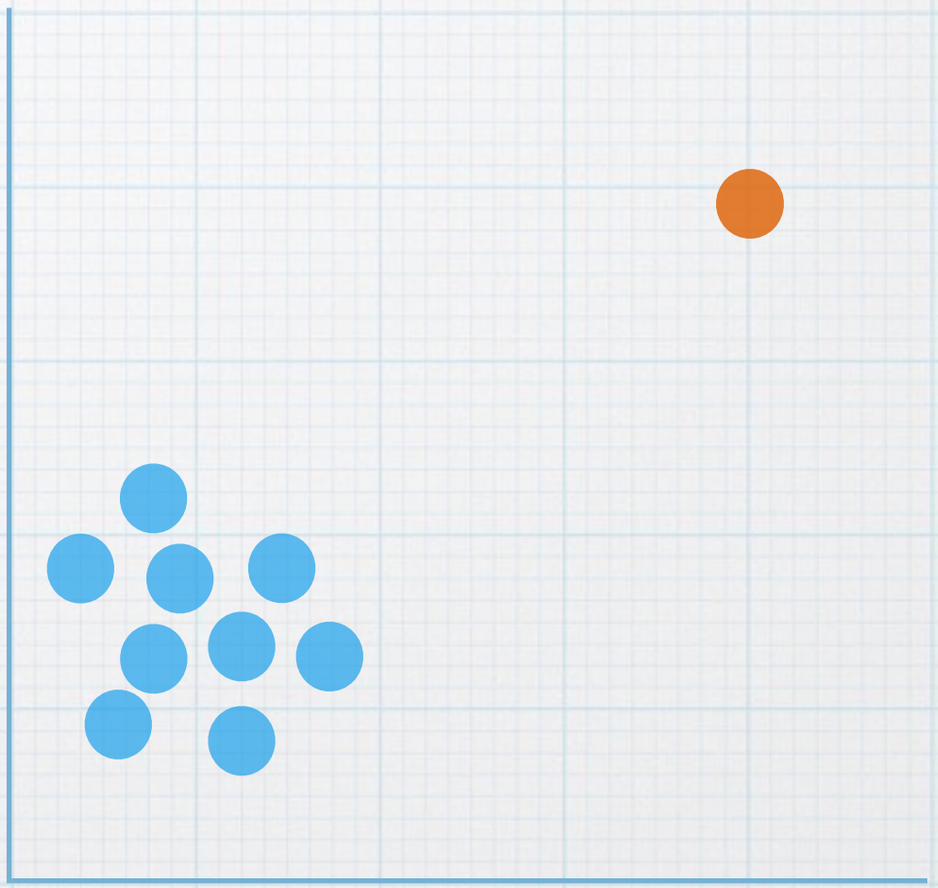


Anomaly Detection

* Ignore data that does not look normal

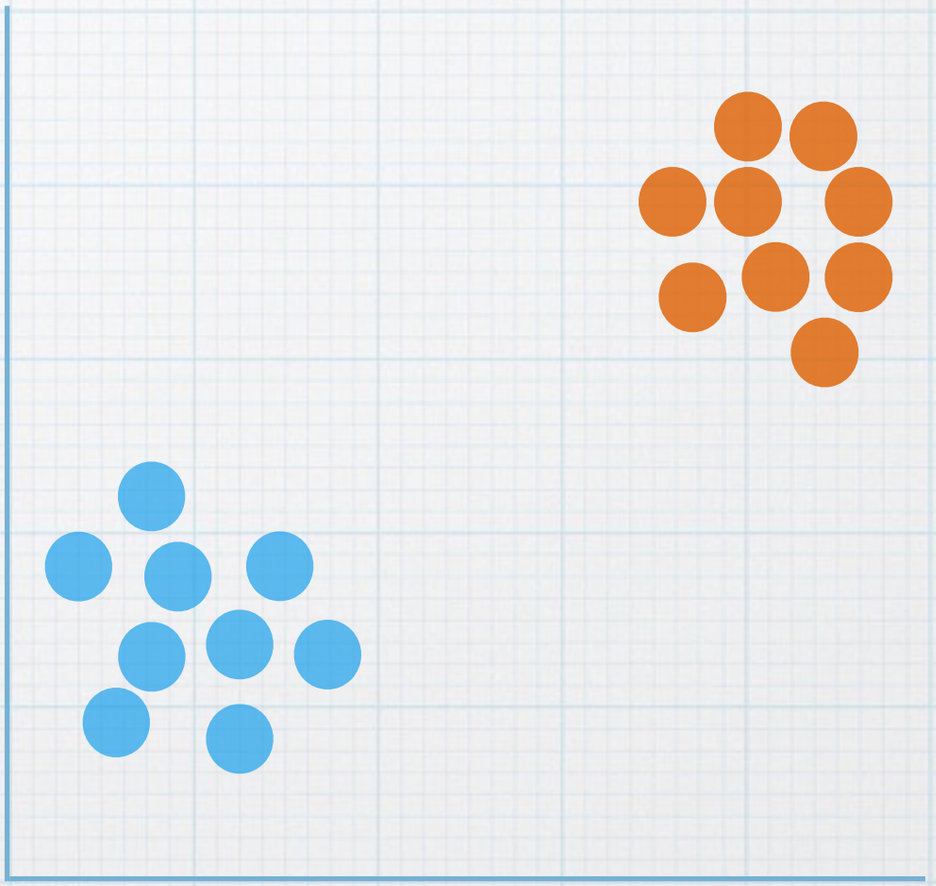
* References:

[Chitradevi et al. 2010]
[Chatzigiannakis and Papavassiliou 2007]
[Rassam et al 2012]
[Livani and Abadi 2010]



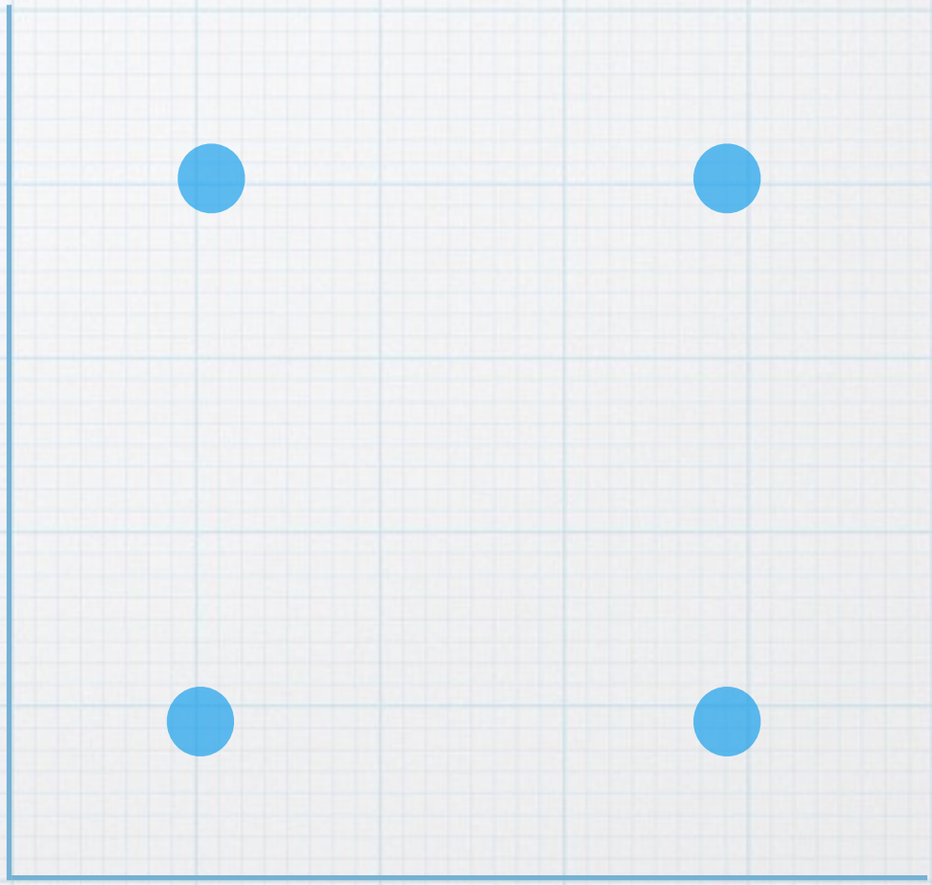
Anomaly Detection

* Issues
  * can be duped by a participant that emulates multiple colluding data sources



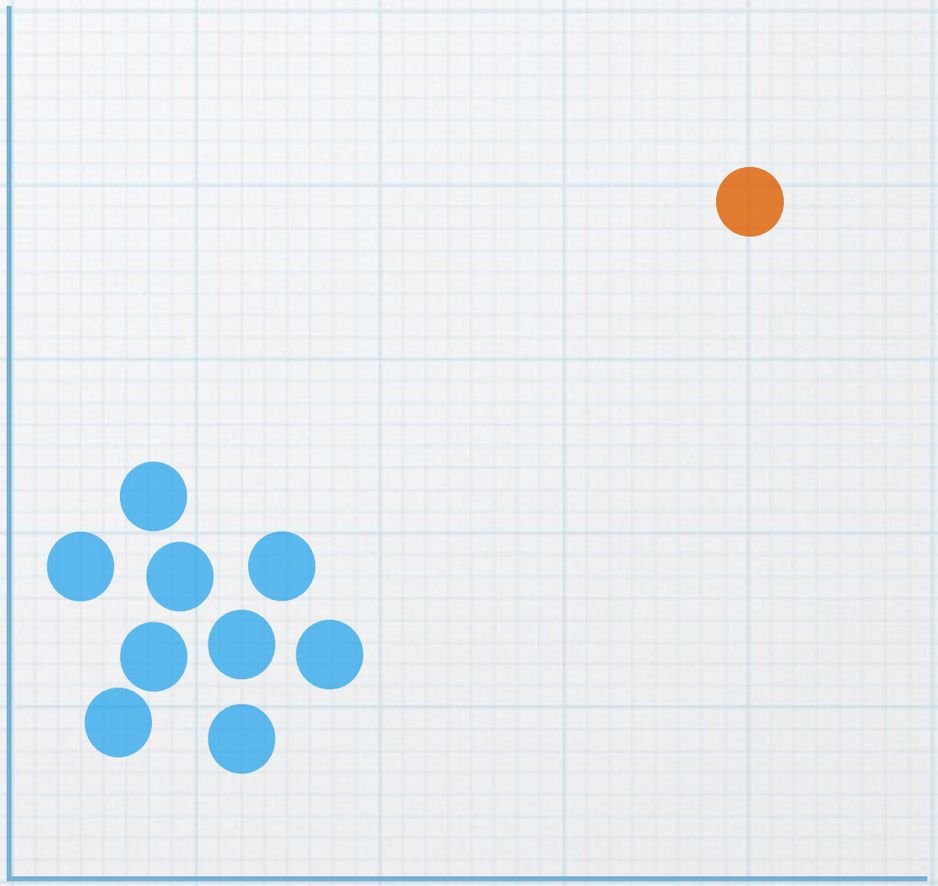
Anomaly Detection

* Issues
  * accuracy depends on number and distribution of data sources



Anomaly Detection

* Issues
  * may categorize new information as anomalous



Device Monitoring

* [Modine 2007] [Schramm 2007]
* Scan device to ensure only expected process and data exist
* Issues:
  * Intrusive
  * Prone to False Positives

Design Goals

Goals	Reputation Ratings	Anomaly Detection	Device Monitoring	Ideal Solution
Independent Verification			✓	✓
Emulation + Collusion			✓	✓
Non-intrusive	✓	✓		✓

Design Goals

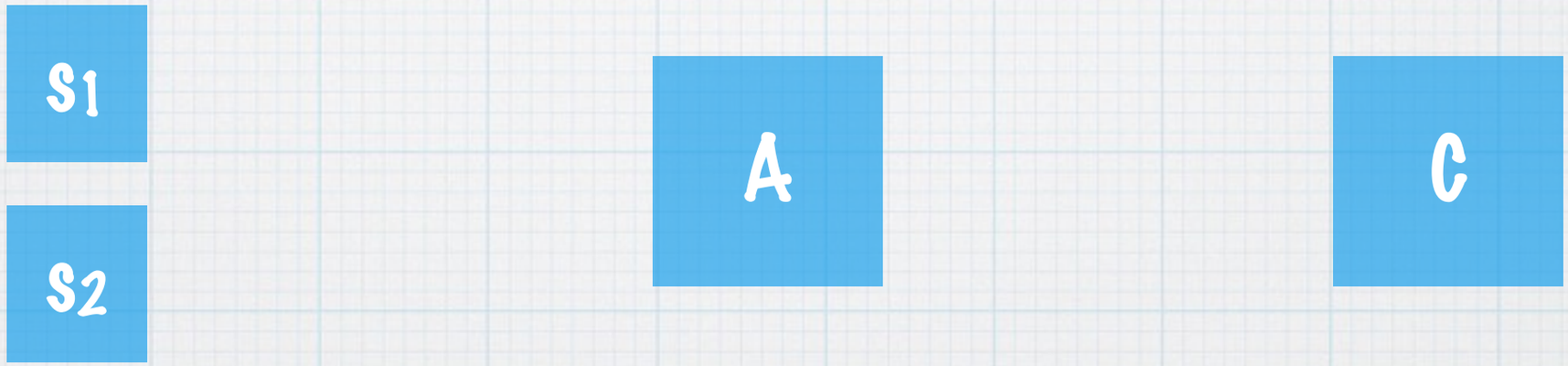
Goals	Reputation Ratings	Anomaly Detection	Device Monitoring	Trust-but-Verify
Independent Verification			✓	✓
Emulation + Collusion			✓	✓
Non-intrusive	✓	✓		✓

Solution Approach

* First address the data integrity problem
* Then add privacy as a constraint

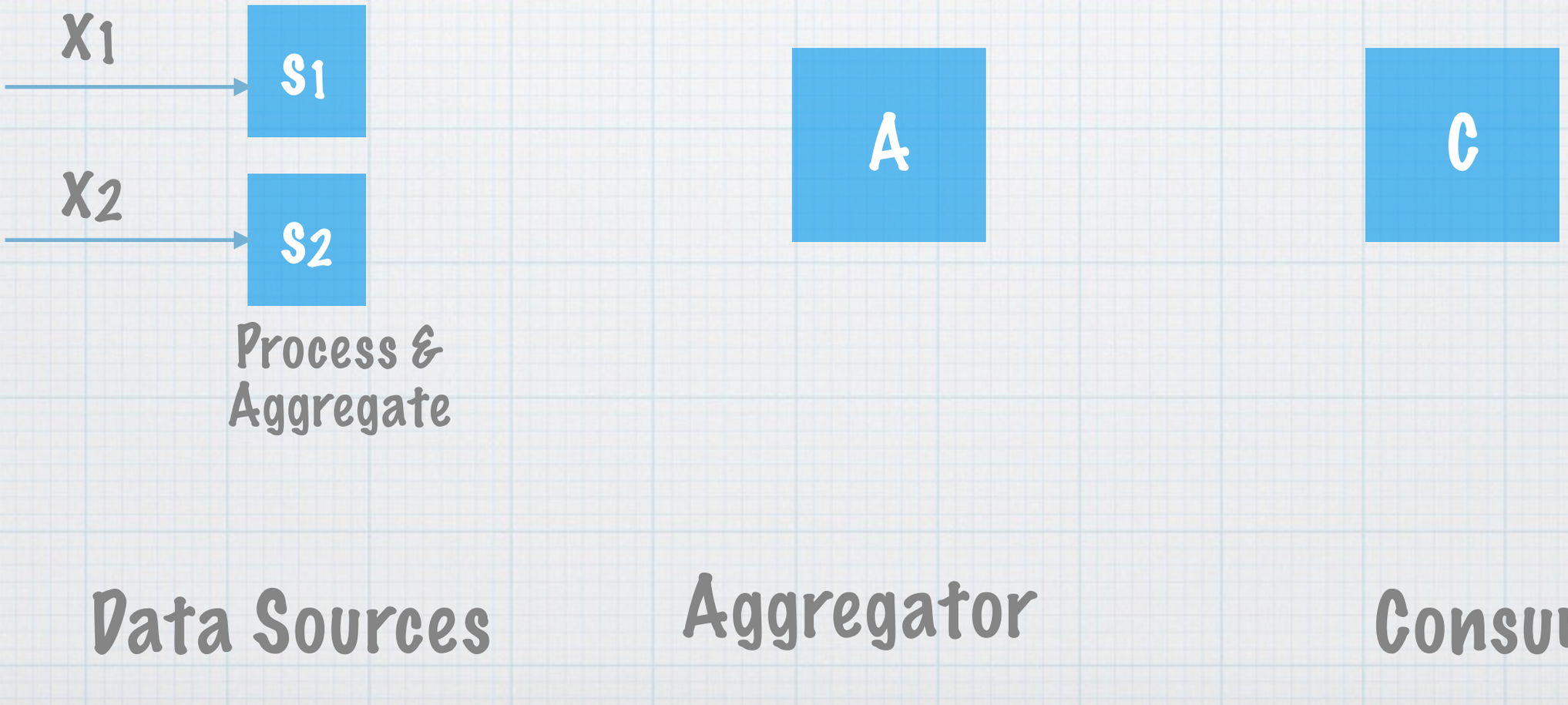
The Trust-but-Verify Model

* Trust the sources, but probabilistically verify the information they send



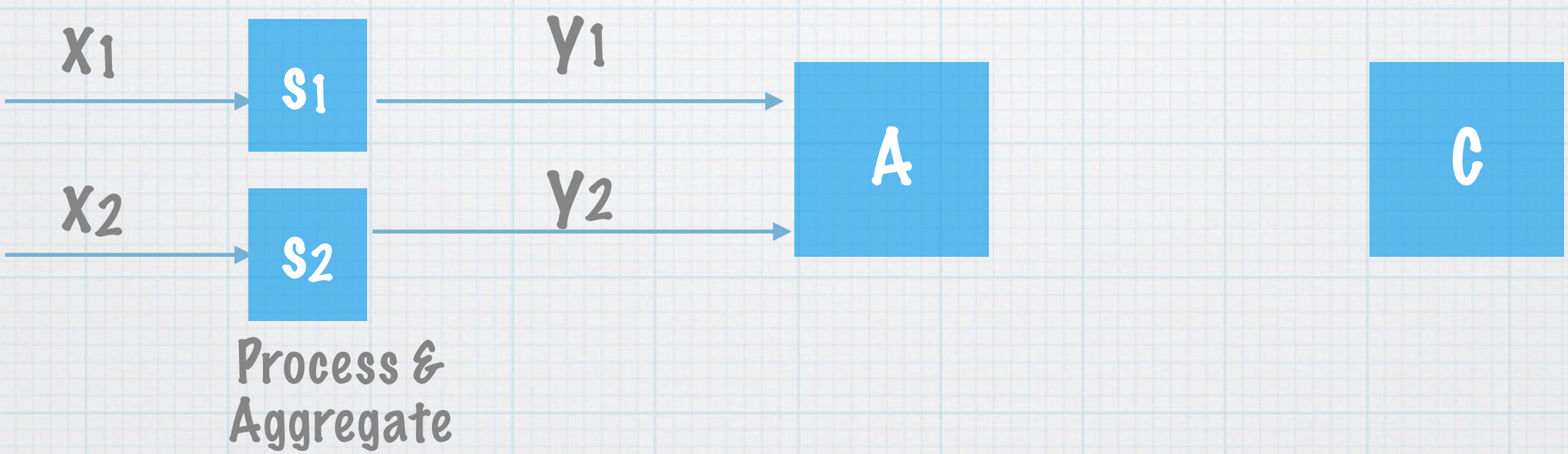
The Trust-but-Verify Model

* Trust the sources, but probabilistically verify the information they send



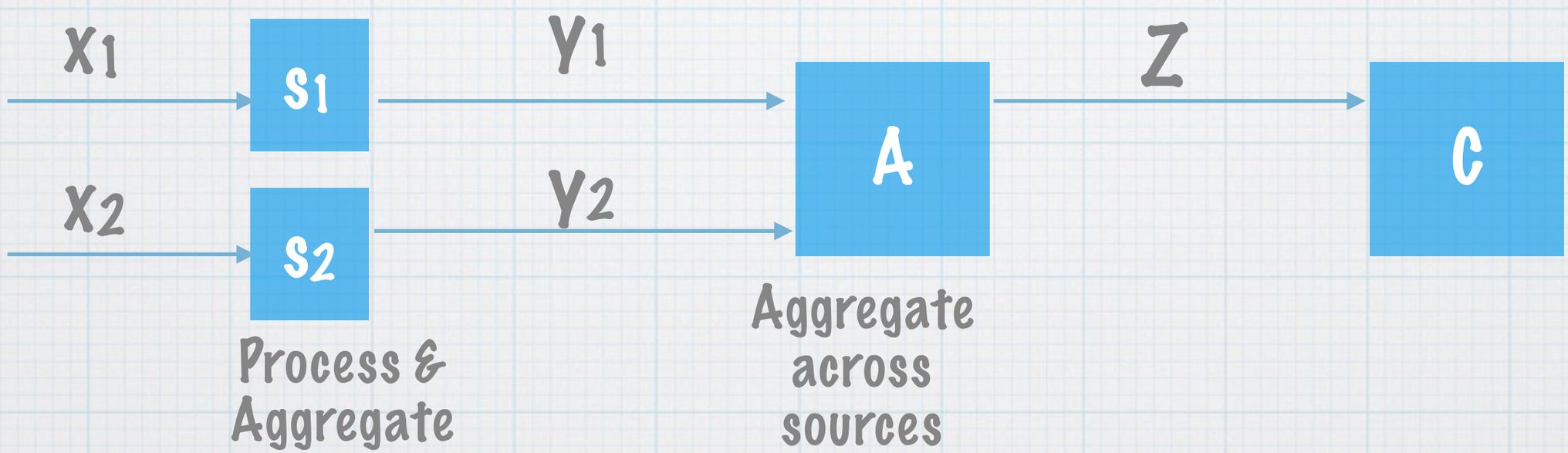
The Trust-but-Verify Model

* Trust the sources, but probabilistically verify the information they send



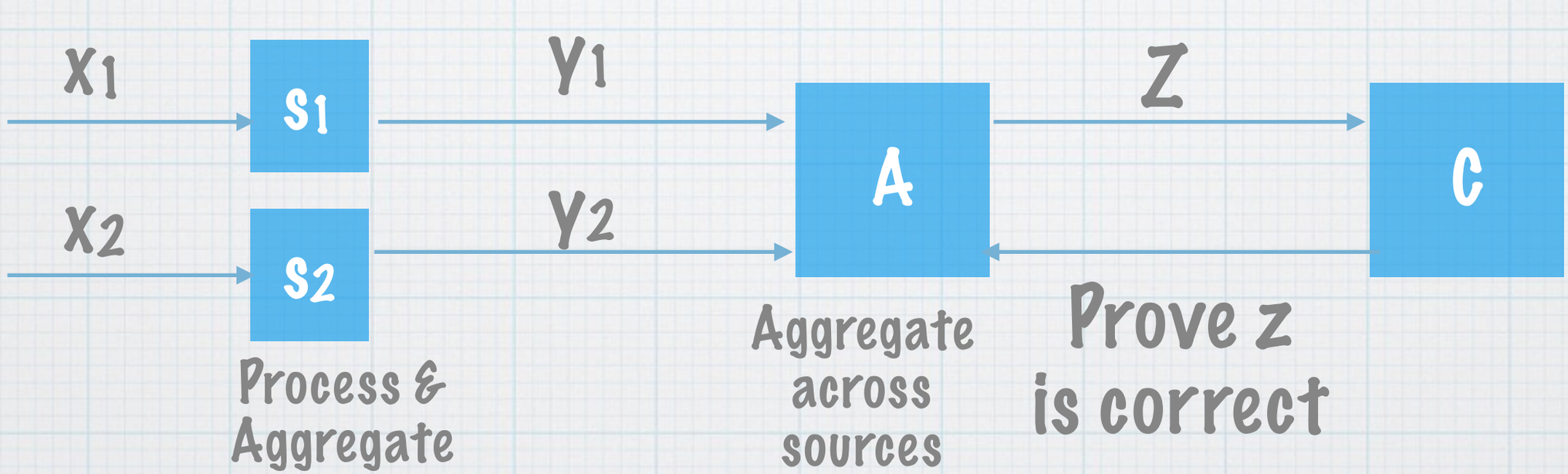
The Trust-but-Verify Model

* Trust the sources, but probabilistically verify the information they send



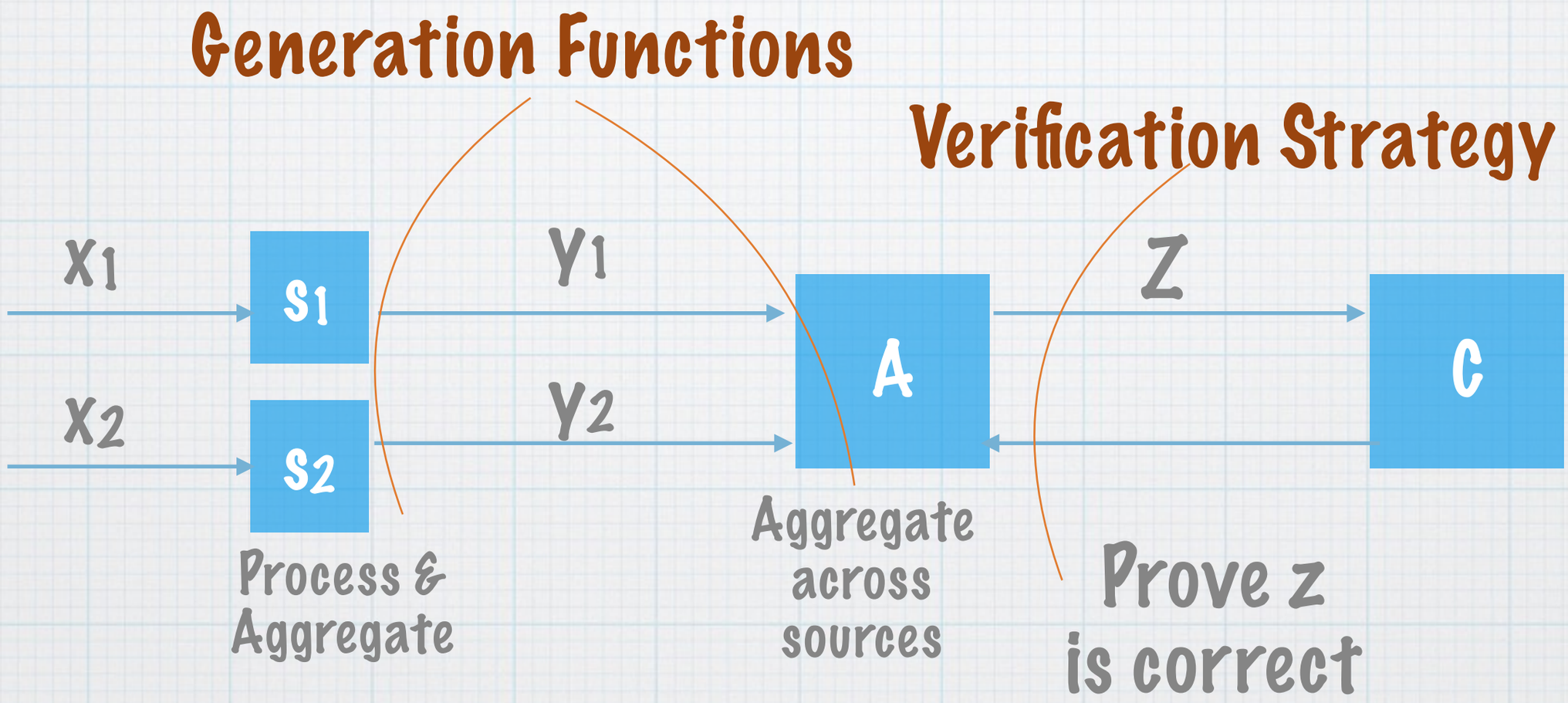
The Trust-but-Verify Model

* Trust the sources, but probabilistically verify the information they send



The Trust-but-Verify Model

* Goal: Develop verification function to enable C to check z.



Applying Trust-but-Verify

* Step 1: Identify the data to verify
  * Sensory information collected from the environment

Applying Trust-but-Verify

* Step 1: Identify the data to verify
* Step 2: Identify generation functions
  * Functions in the sensor's device drivers

Applying Trust-but-Verify

* Step 1: Identify the data to verify
* Step 2: Identify generation functions
* Step 3: Build verification functions
    * How do I convince the data consumer that the functions in the sensor's device drivers were faithfully executed?

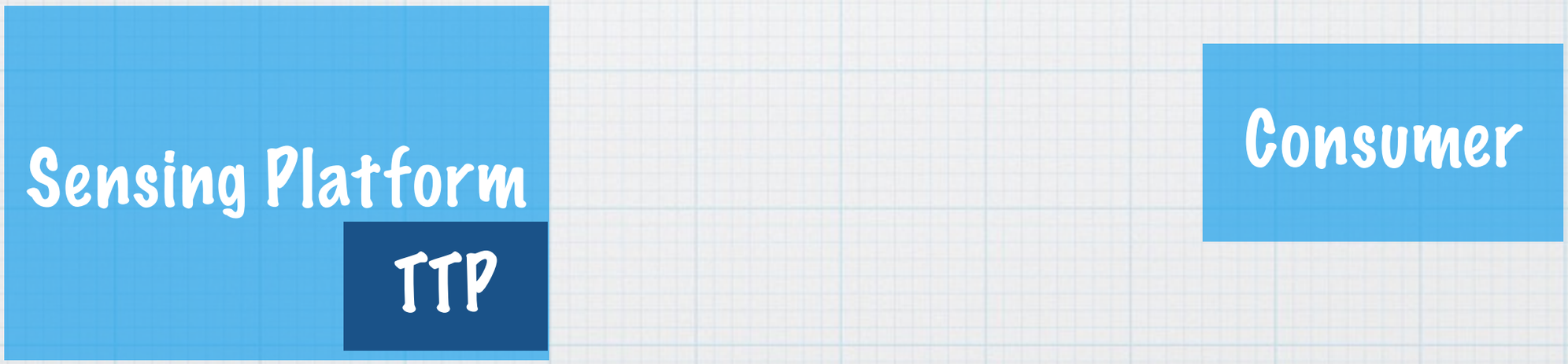
Build a "closed box" Trustworthy Sensing Platform

Sensing Platform

Consumer

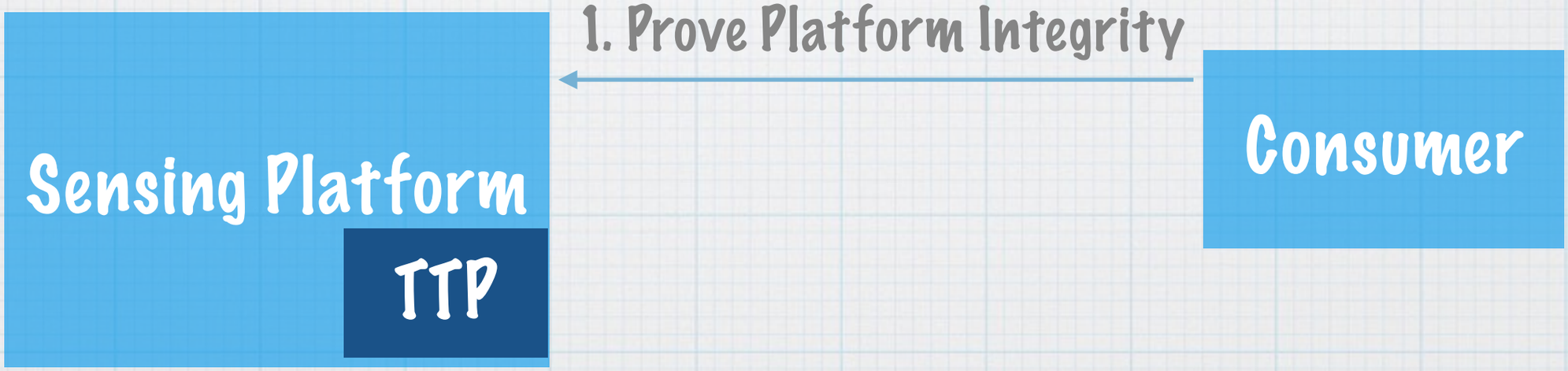
Build a "closed box" Trustworthy Sensing Platform

* Establish a Trusted-Third-Party (TTP) inside sensing platform



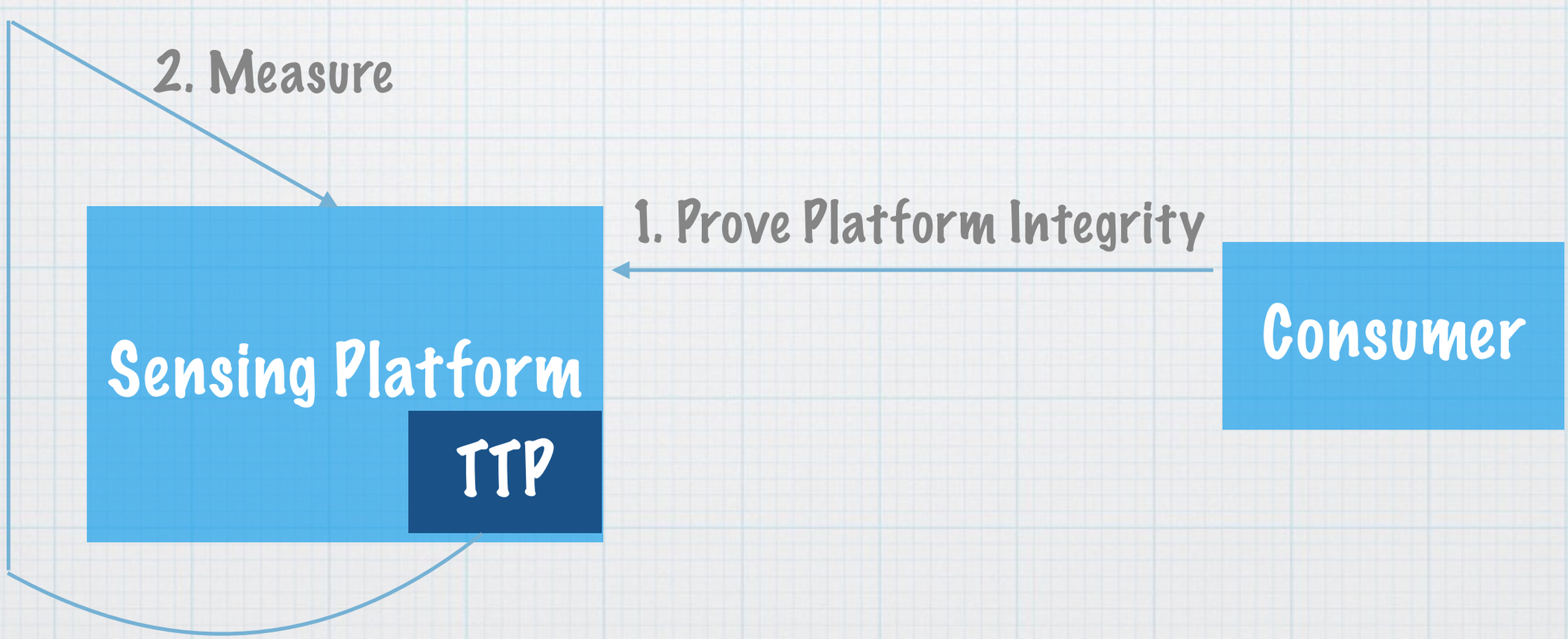
Build a "closed box" Trustworthy Sensing Platform

* Establish a Trusted-Third-Party (TTP) inside sensing platform



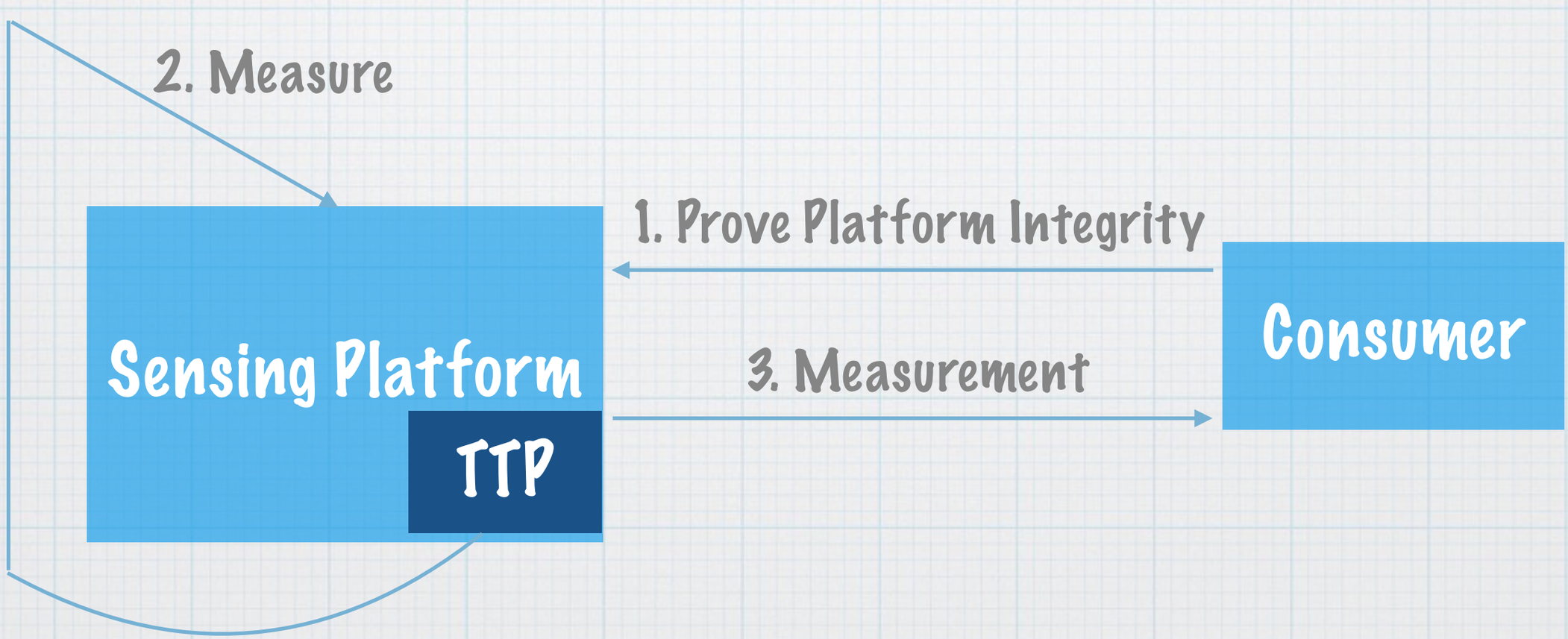
Build a "closed box" Trustworthy Sensing Platform

* Establish a Trusted-Third-Party (TTP) inside sensing platform



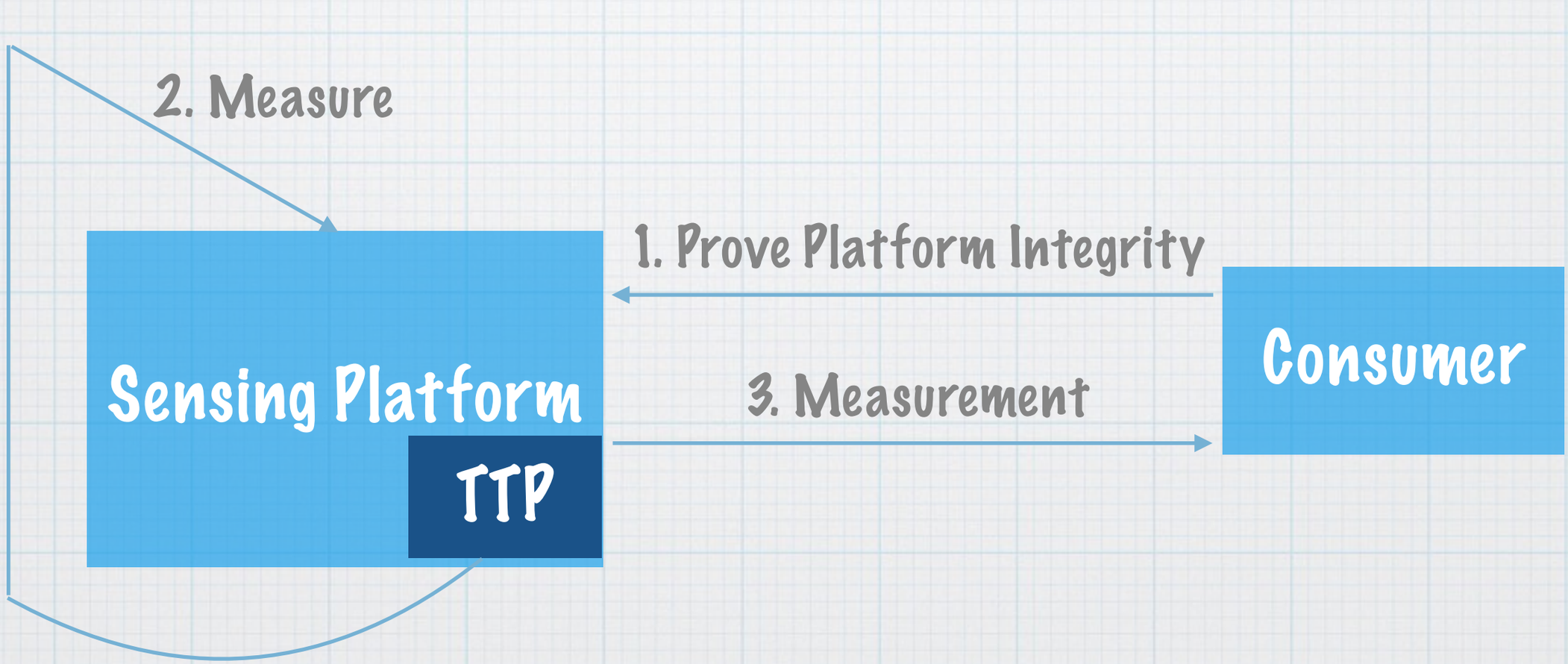
Build a "closed box" Trustworthy Sensing Platform

* Establish a Trusted-Third-Party (TTP) inside sensing platform

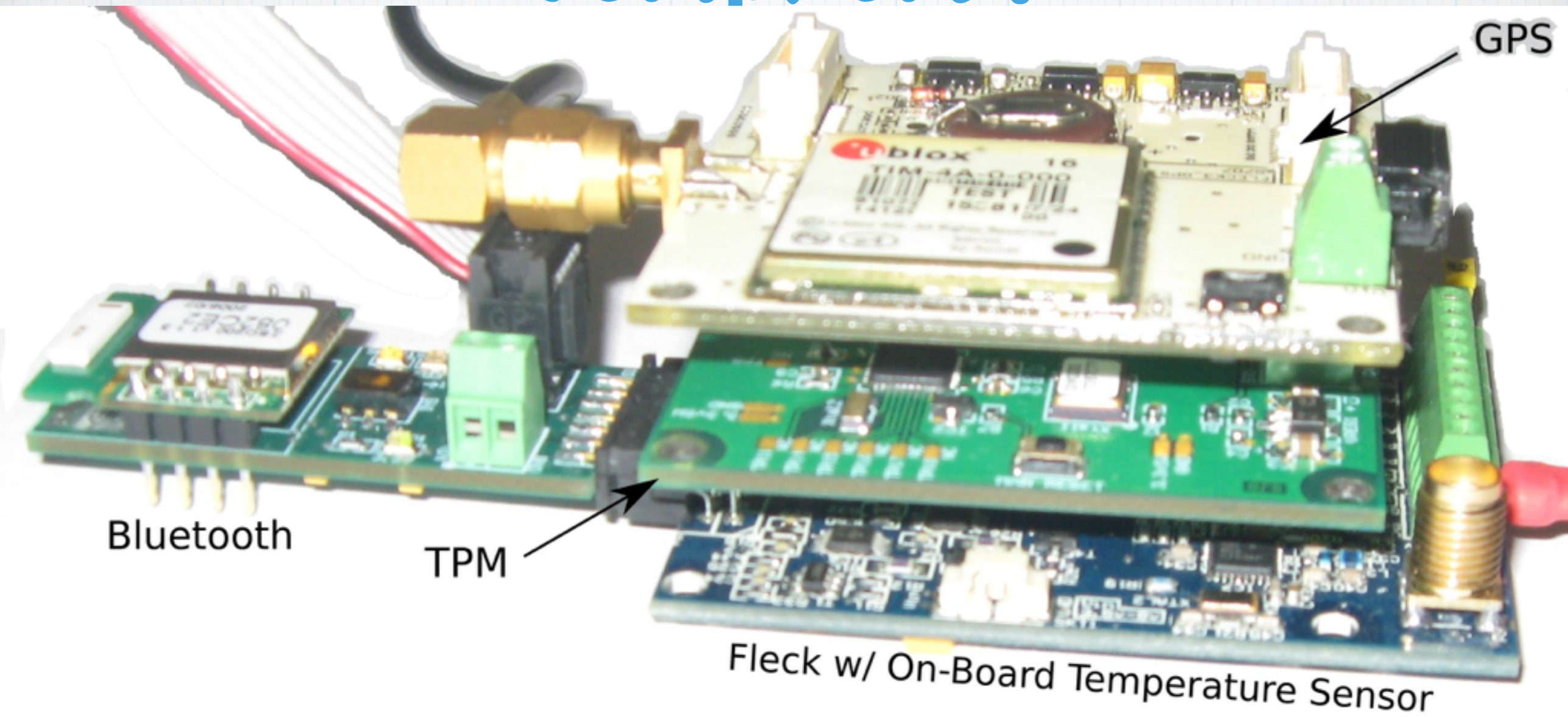


Build a "closed box" Trustworthy Sensing Platform

* Trust in the platform induces trust in the faithful execution of the generation functions



The Trusted Sensing Peripheral



* TTP: Trusted Platform Module (TPM): Enables trusted boot [Trusted Computing Group]
* Fleck: Provides secure execution [Francillon and Castelluccia [2008]]

Time Required for Platform Attestation

Task	Compute Time (sec)	Transmit Time (sec)
Single Attestation	1.72 (+/- 0.01)	0.3 (+/- 0.1)

* Attestation: TPM's RSA signature over SHA digest of instruction memory
* Sensing platform: 8 KB of memory; 8 MHz Atmega micro controller

Summary

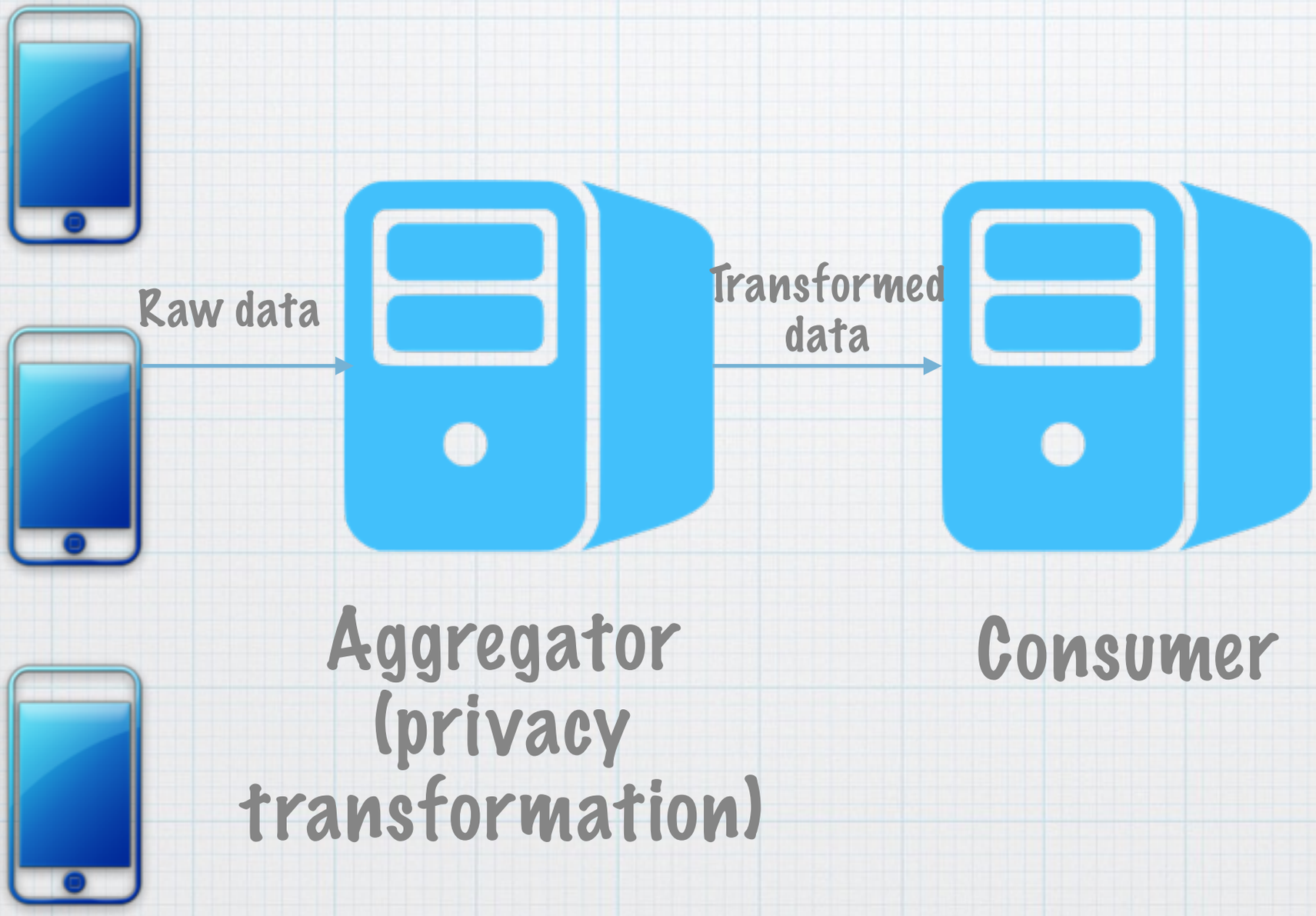
* Problem: How can a data consumer - receiving data from sensors that are not under its control - trust that the data is a true representation of the real-world phenomenon being sensed?
* Solution: Build a separate trustworthy sensing platform that cannot be altered or modified
* Limitations
  * Does not prevent PHYSICAL collusion among users
  * Can be fooled by doctored sensing environment
  * A separate platform users must carry around

Privacy in Mobile CrowdSensing

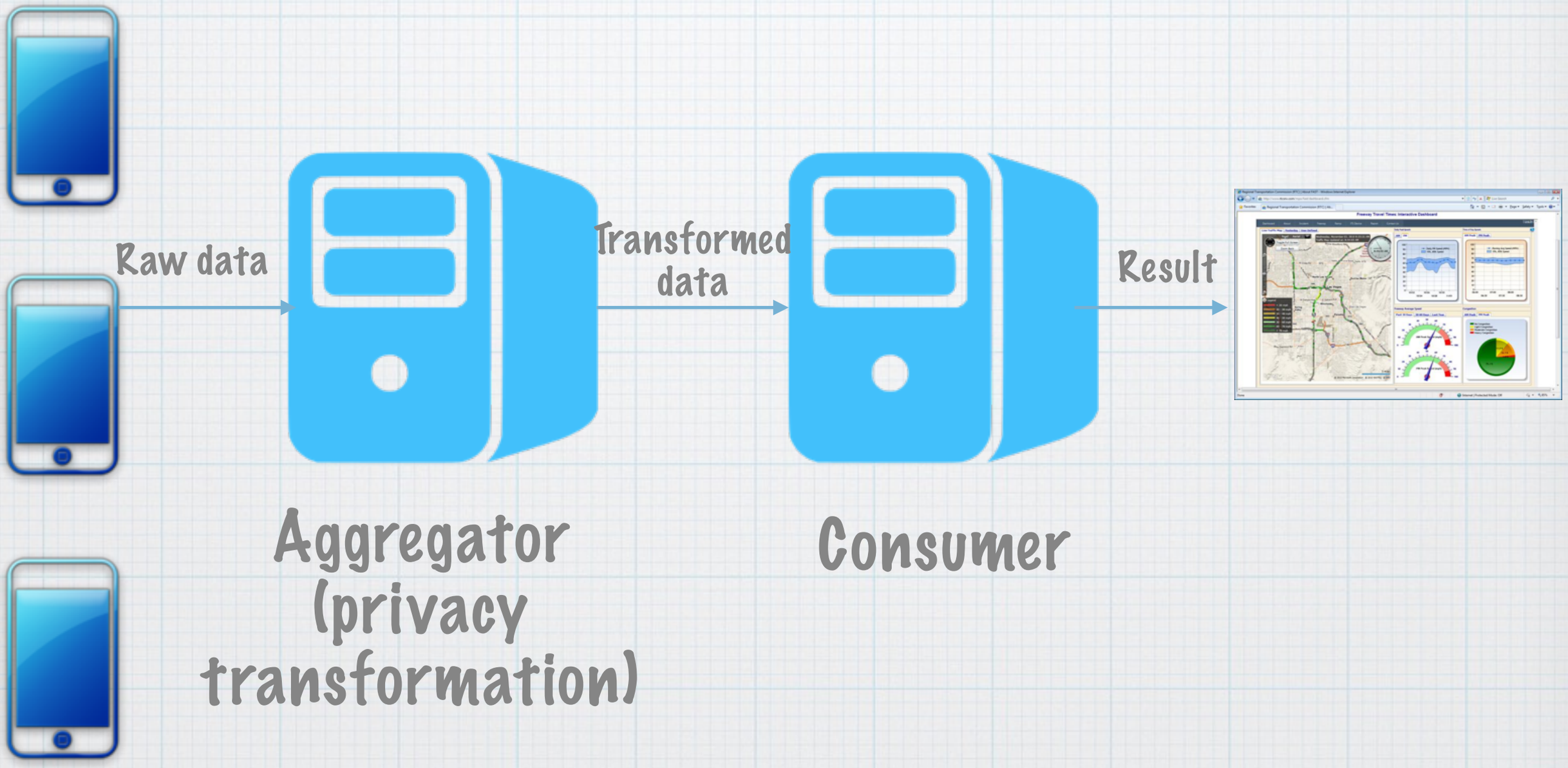
Privacy in Mobile CrowdSensing

Privacy in Mobile CrowdSensing

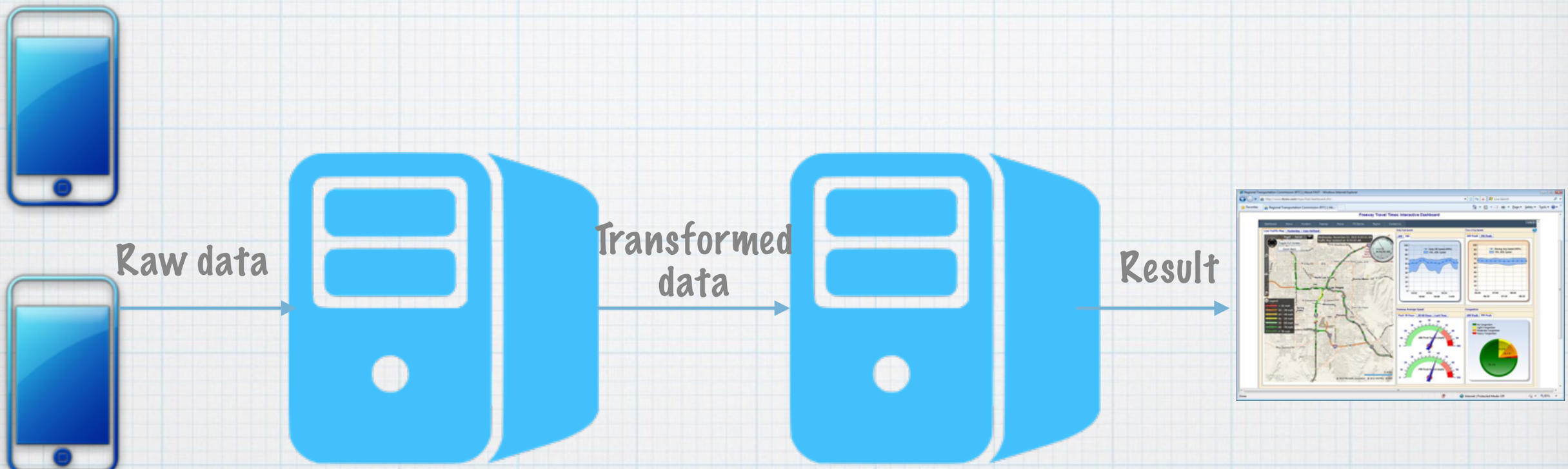
Privacy in Mobile CrowdSensing



Privacy in Mobile CrowdSensing



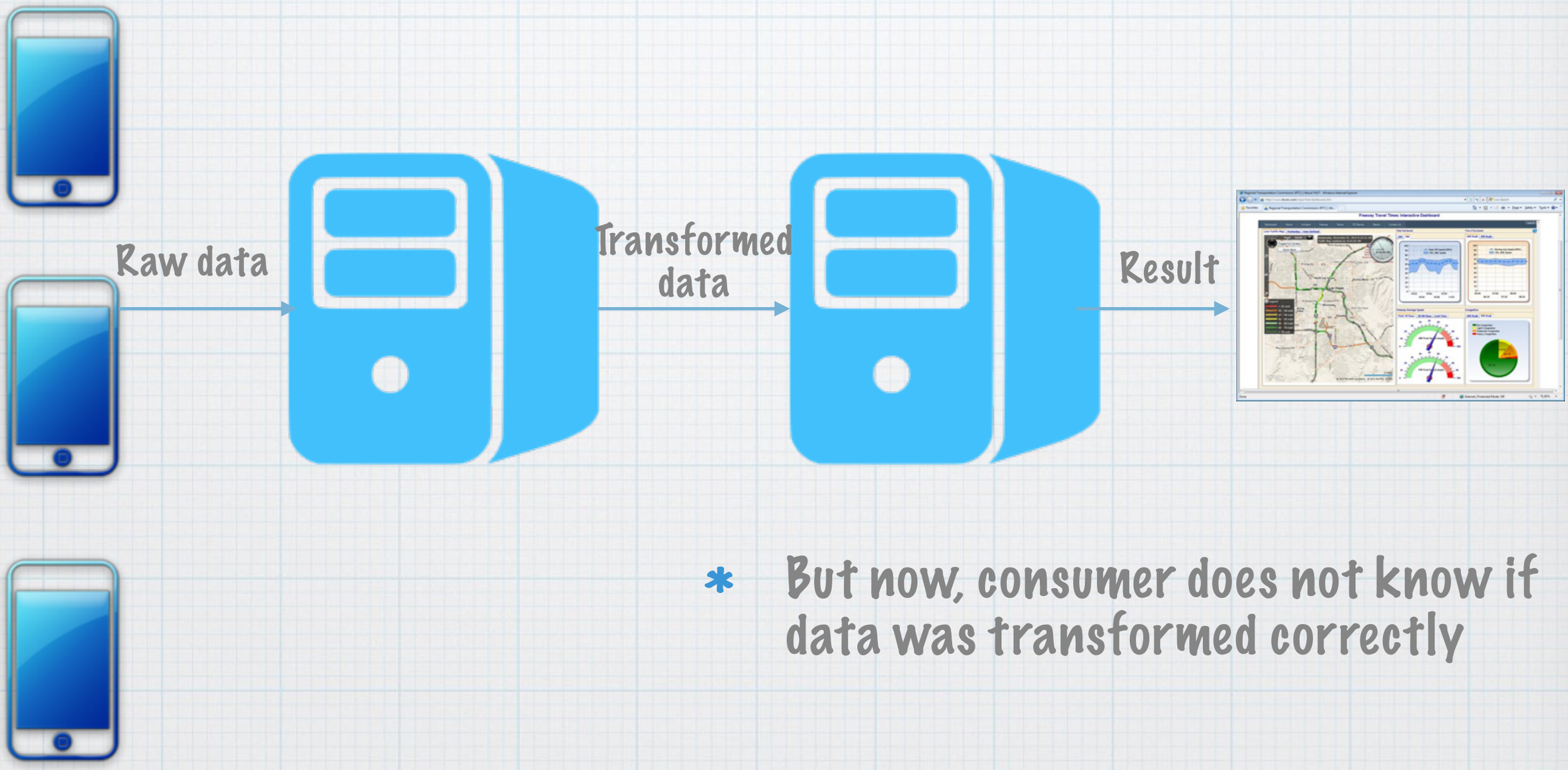
Privacy in Mobile CrowdSensing



* Trusted intermediary computes privacy transformation

* Example: location cloaking or averaging [Gruteser and Grunwald 2003], [Rastogi and Nath 2010], [Shi et al. 2011]

Privacy in Mobile CrowdSensing



* But now, consumer does not know if data was transformed correctly

Privacy in Mobile CrowdSensing

* Trusted intermediary computes privacy transformation
  * Example: location cloaking or averaging [Gruteser and Grunwald 2003], [Rastogi and Nath 2010], [Shi et al. 2011]
* But now, consumer does not know if data was transformed correctly

Related Work

* Location Privacy and Integrity are not generally addressed together
* PoolView [Ganti et al 2008]
  * Compute community statistics using perturbed data (e.g. average wight or speed)
  * No location privacy - data collection locations known
  * no integrity - statistics computed by trusted parties

Related Work

* Location Privacy and Integrity are not generally addressed together

* VPriv [Popa et al 2009]: Compute tolls over location paths

  * Integrity & weak privacy: Pseudonymous locations in the clear

Related Work

* Location Privacy and Integrity are not generally addressed together

* PrivStats: privacy-preserving data aggregation with accountability [Popa et al. 2011]

  * No location privacy: Data collection locations known

Applying Trust-but Verify: Integrity with Privacy

* Identify data to verify
    * Output of privacy transformation
* Identify generation functions
    * Privacy transformation
* Identify verification functions
    * How to convince the data consumer that the privacy transformation was executed faithfully while preserving data source privacy?

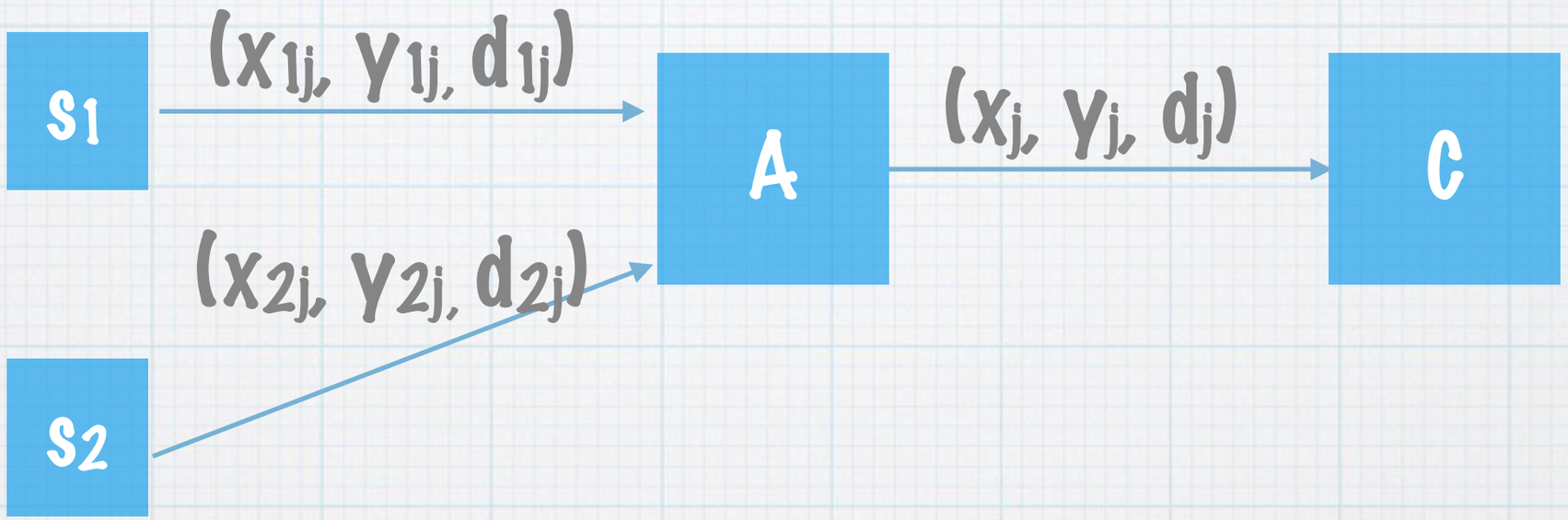
Homomorphic Commitments

* Idea: Check computations over inputs using functions of inputs
* Example
  * Blake receives the sum 5 from Eve. This is the sum of inputs from Alice and Bob.
  * Blake wants to know if Eve added these inputs faithfully.
  * But Alice and Bob do not want to reveal their inputs to Blake.

Homomorphic Commitments: The Idea

* Insecure version
  * Alice sends $f(3) = 2^3$, Bob sends $f(2) = 2^2$ to Blake
* Blake checks: $f(3) \times f(2) = f(5)$
* Indeed, $2^3 \times 2^2 = 2^5$; Blake is happy!

Assumed System Model



* Privacy-preserving Transformation: Mean of Inputs [Popa et al. 2011], [Rastogi and Nath 2010], [Shi et al. 2011], [Ganti et al. 2008]
* n >= k for k-anonymity [Samarati and Sweeney 1998]
    * Inferences from (xj, yj, dj) would apply to any of the k sources

LocationProof: Normal Operation

* Consider
  * k sources
  * publishing x_{ij}: longitude of source i at time instance j

LocationProof: C challenges A

S_i	A	C
interval j:		
sense x_{ij} s_i -> A: x_{ij} b= x_{ij}	x_j = Sum x_{ij} A -> C: x_j	$r_j = x_j/k$

How Long Before a Fabrication is Detected

* p: C's checking probability
* q: Probability with which A corrupts aggregates
* Expected number of successes before first failure (1 - pq)/pq

Data Source Overhead

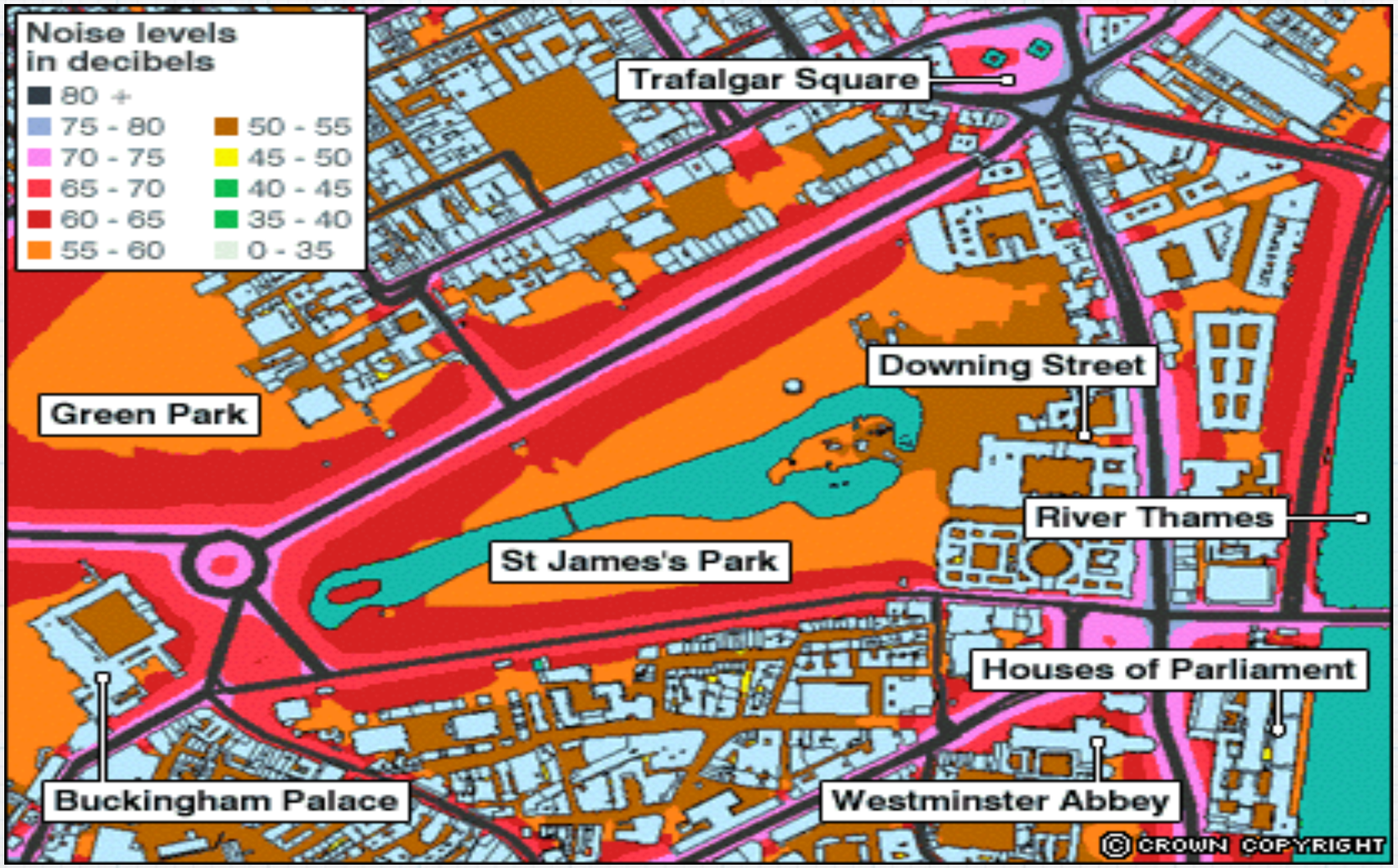
	Android	TSP
Current	144 mA	55 mA
Time	2.325 msec (+/- 1.26)	3.44 sec (+/- 0.03)
Energy Consumption	1.2 mJ	5 Joules

Summary: Privacy-preserving, High-Integrity CrowdSensing

* Problem
    * How to achieve simultaneous data integrity and privacy
* Solution
    * Use homomorphic commitments as a building block
* Limitations
    * Supports additive transformations only
    * Does not indicate location spread (future work)

Collaboration (Joint Work with Rajib Rana, Wen Hu, Chun-tung Chou & Salil Kanhere [Rana1 01])

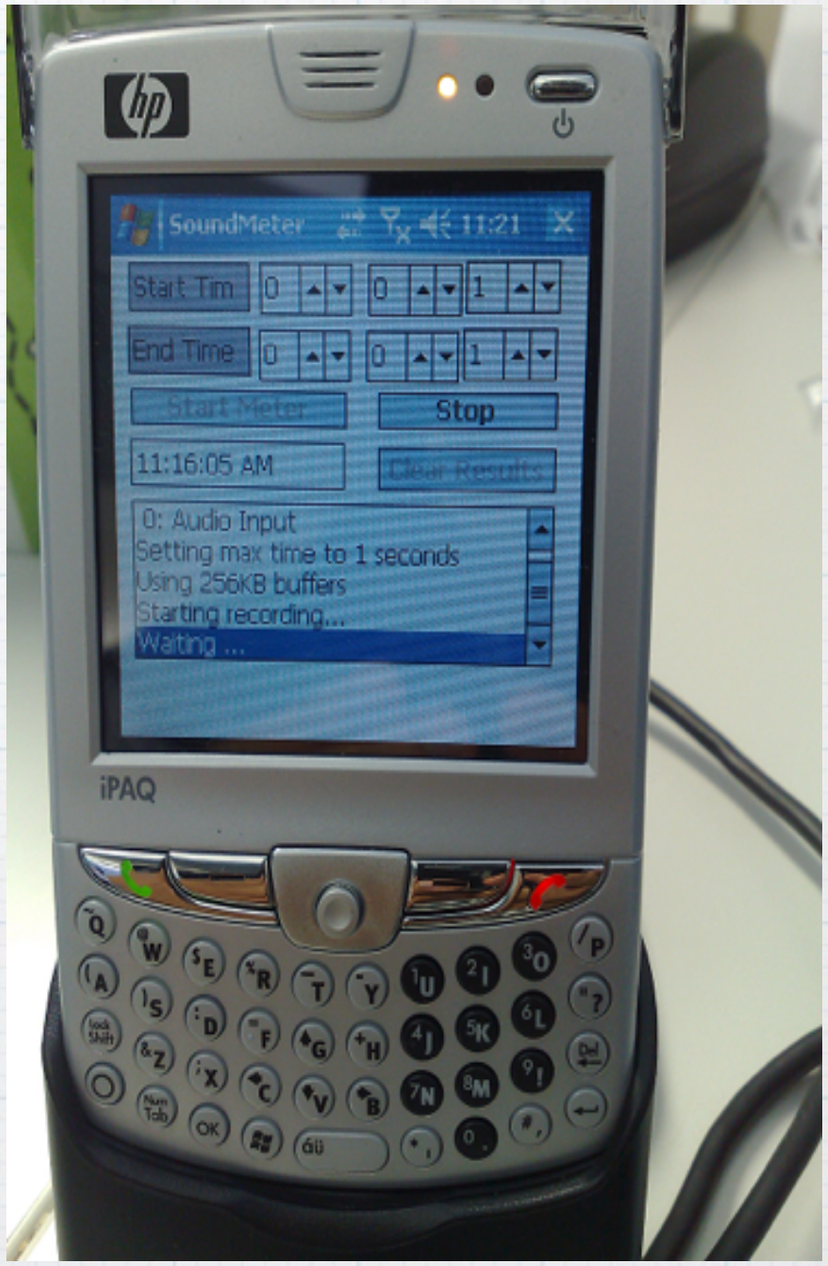
Noise Map



Existing maps are simulation generated. Can not be used for local action plans.

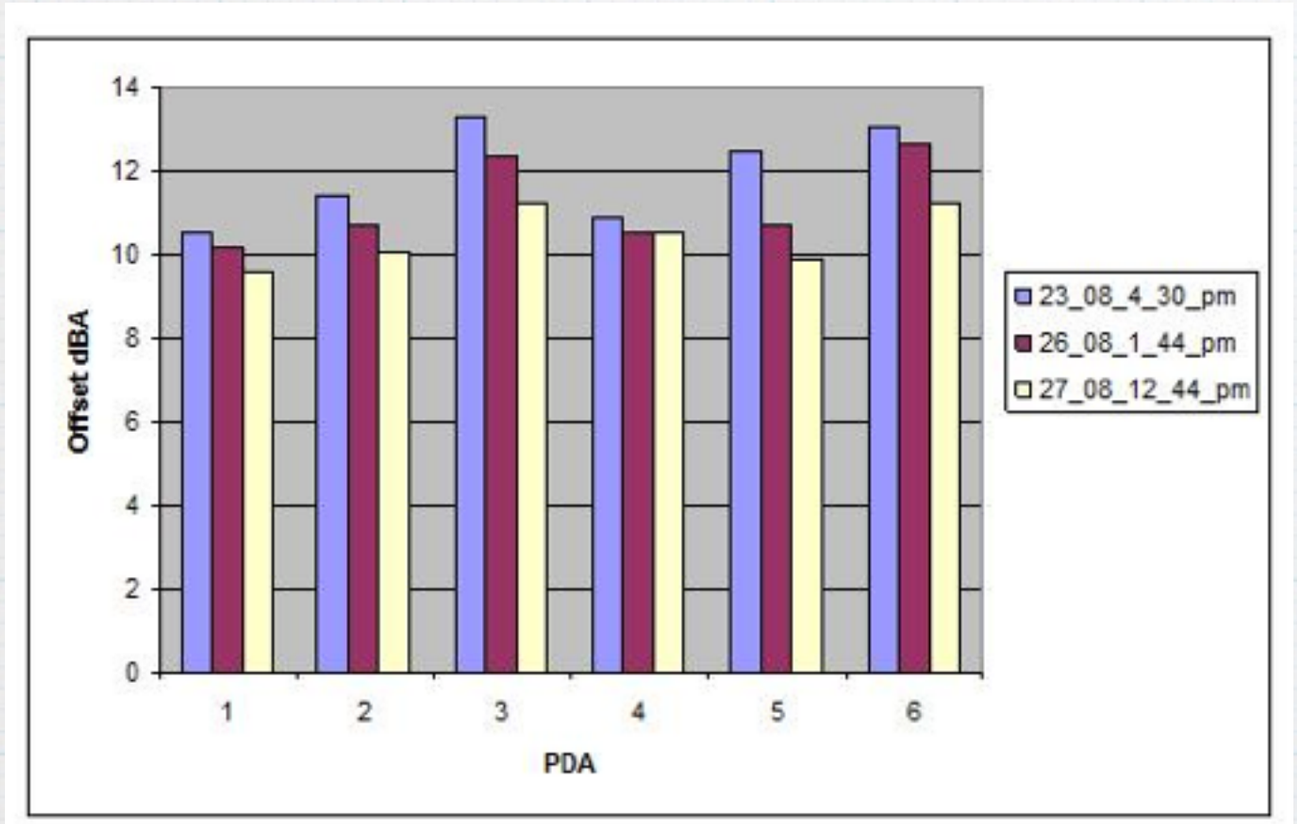
Objectives

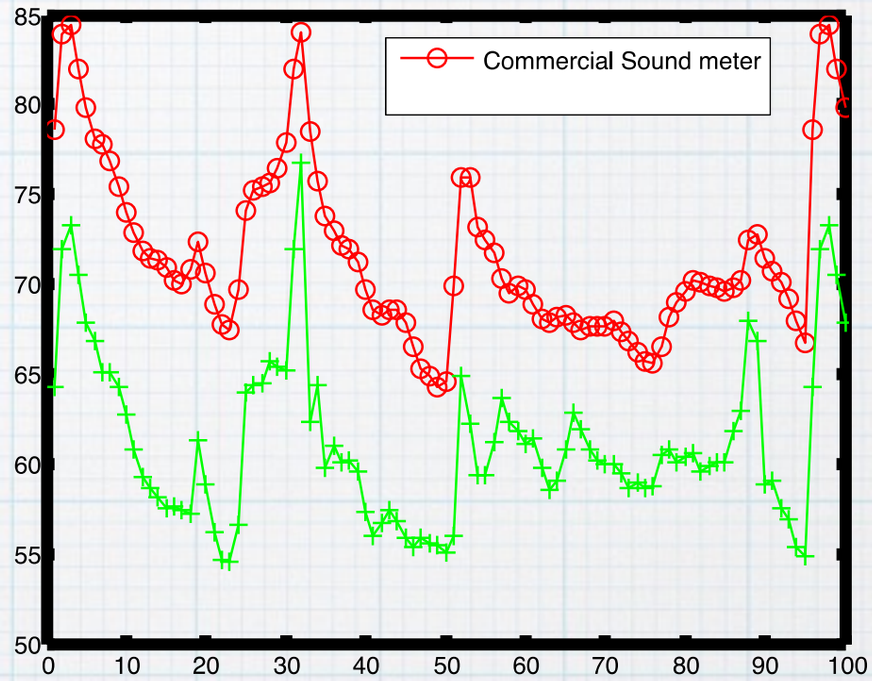
Reconstruction of Temporal-Spatial Noise Profile from Incomplete Audio Samples Collected via Mobile CrowdSourcing.



PDA Based Sound Level Meter

Calibration



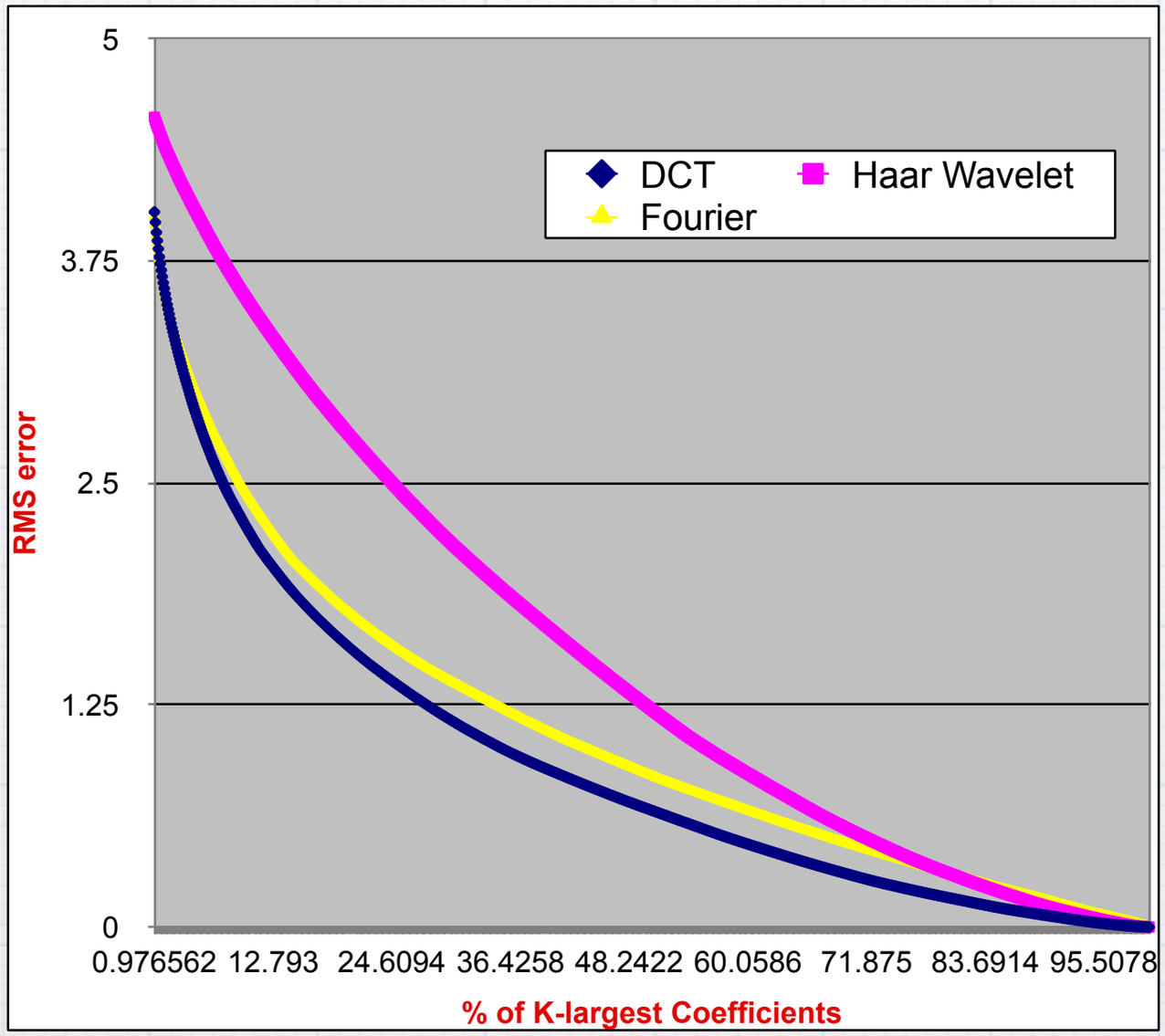


< 2 dBA over 3 days

Challenges

* Incomplete Sampling
  * insufficient samples
  * samples missing from area of interest
    * irregular sampling (oversampled and under-sampled areas)
    * sampling frequency may change with time of day
* Bandwidth limitations
* Approach - compressed sensing (random projection)

Compressibility of Spatio-Temporal Noise Profile

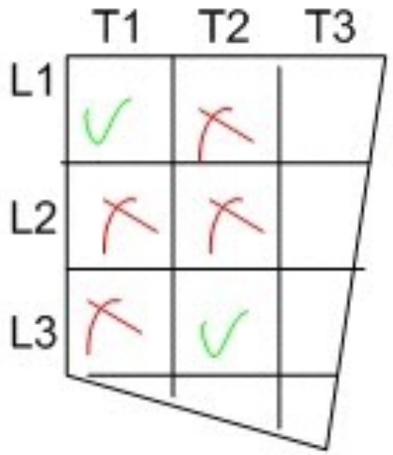


System Architecture

Base Station

2

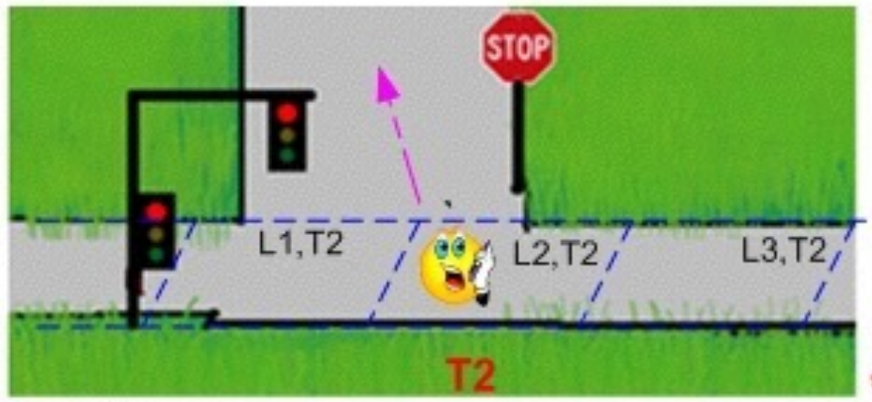
Volunteers sending noise measurements to Base station using GPRS/EDGE/Wi-Fi



Samples collected at base station

CS based Reconstruction of noise level form collected samples

Volunteers collecting noise samples while walking through street.



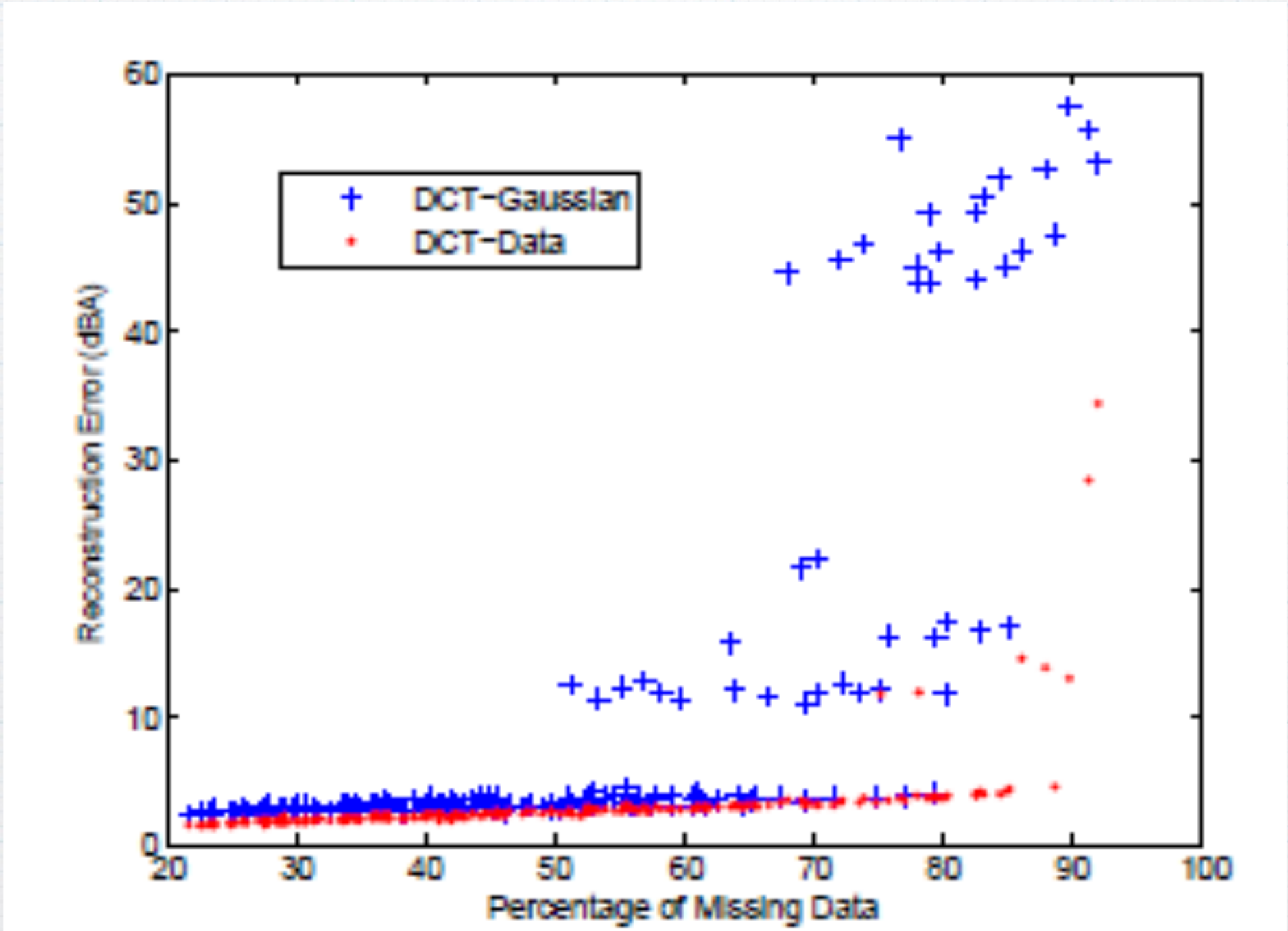
,T3,T4....,

1

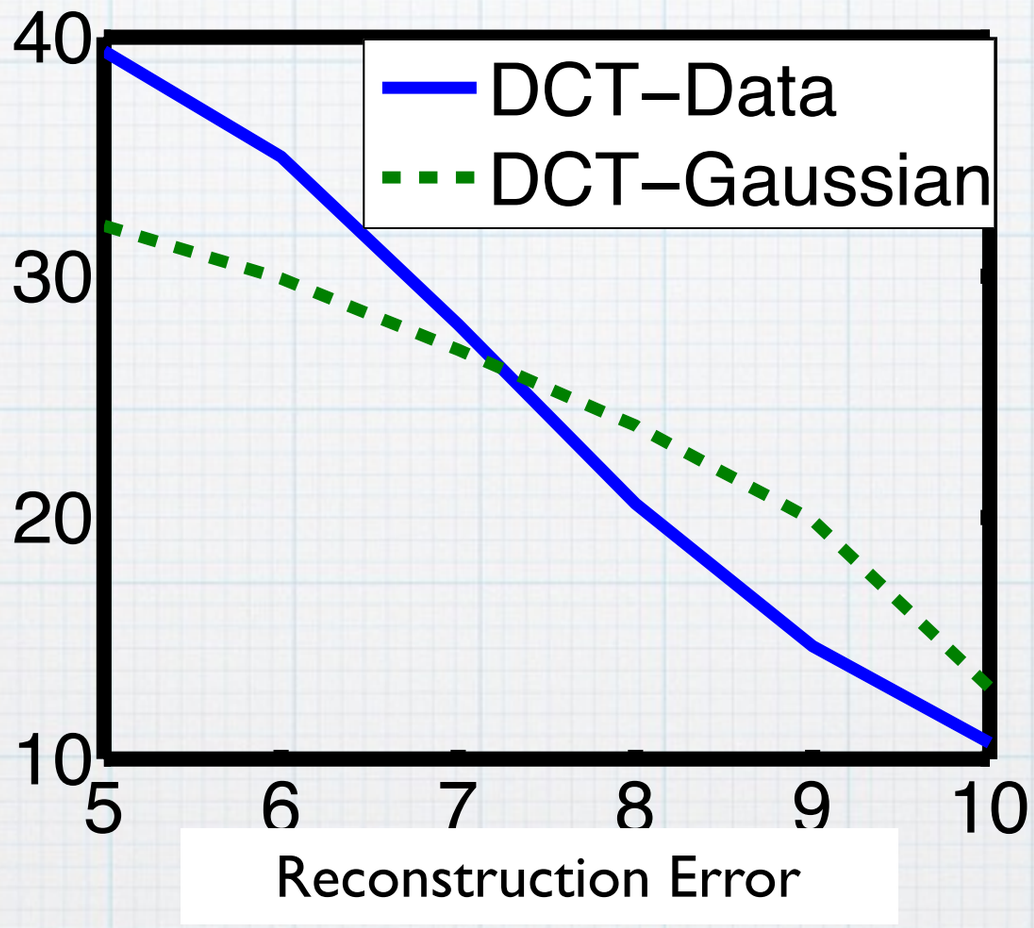
Data Aggregation

* Random Projection (DCT Gaussian)
* Raw Data (DCT Data)

Experimental Results



Experimental Results



Summary

* Mobile Phones as Sound Level Meters
  * compressibility of spatio-temporal noise profile
  * data aggregation using Random Projections
  * mobile phone based sound level meter accurate
* Adaptive sampling
  * better or equivalent reconstruction with fewer samples

References

- [Bulusu08] "Participatory Sensing in Commerce: Using Mobile Camera Phones to Track Market Price Dispersion", Nirupama Bulusu, Chun Tung Chou, Salil Kanhere, et al, In Proceedings of the International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense'08), Raleigh, North Carolina, November 2008.

- [Dua09] "Towards Trustworthy Participatory Sensing", Akshay Dua, Nirupama Bulusu, Wu-chang Feng, and Wen Hu, In Proceedings of the Usenix Workshop on Hot Topics in Security (HotSec 2009), Montreal, Canada, August 2009.

- [Dua14] "Combating Software and Sybil Attacks to Data Integrity in Crowd-Sourced Mobile Embedded Systems", Akshay Dua, Nirupama Bulusu, Wu-chang Feng, and Wen Hu, ACM Transactions on Embedded Computing Systems (TECS), September 2014.

References

- [Rana10] "Ear-Phone: An End-to-End Participatory Urban Noise Mapping System", Rajib Rana, Chun Tung Chou, Salil Kanhere, Nirupama Bulusu, and Wen Hu, In Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2010), SPOTS Trac, Stockholm, Sweden, April 2010.