

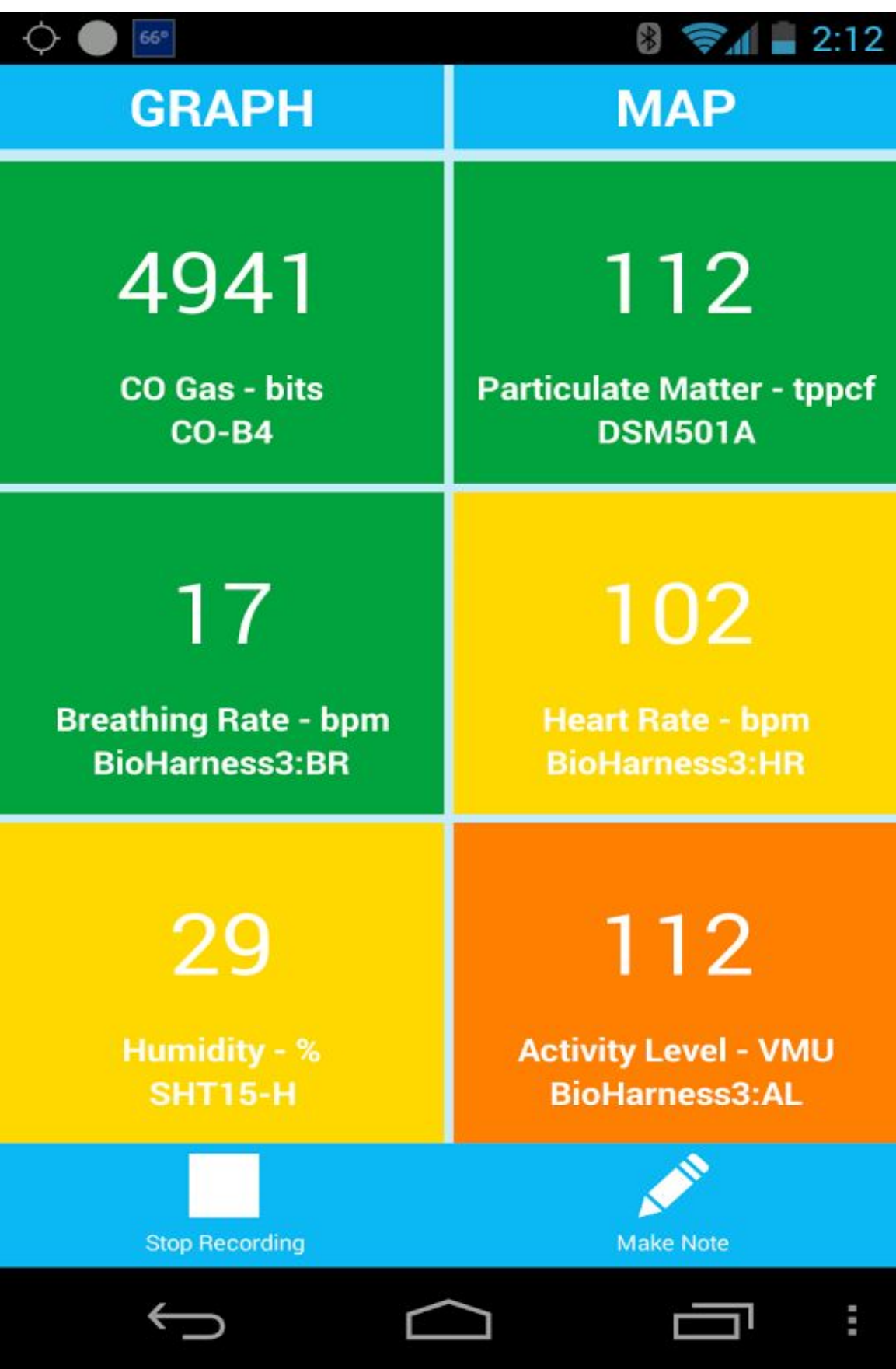
# A Prototype Implementation of IoT Platform for Smart Cities

Mobile Team: Joe Zuhusky, Soumie Kumar  
 Networking Team: Wesley Painter, Zal Bathena  
 Server Team: Hongtao Cheng, Wenliang Zhao  
 New York University, Courant Institute of Mathematical Sciences

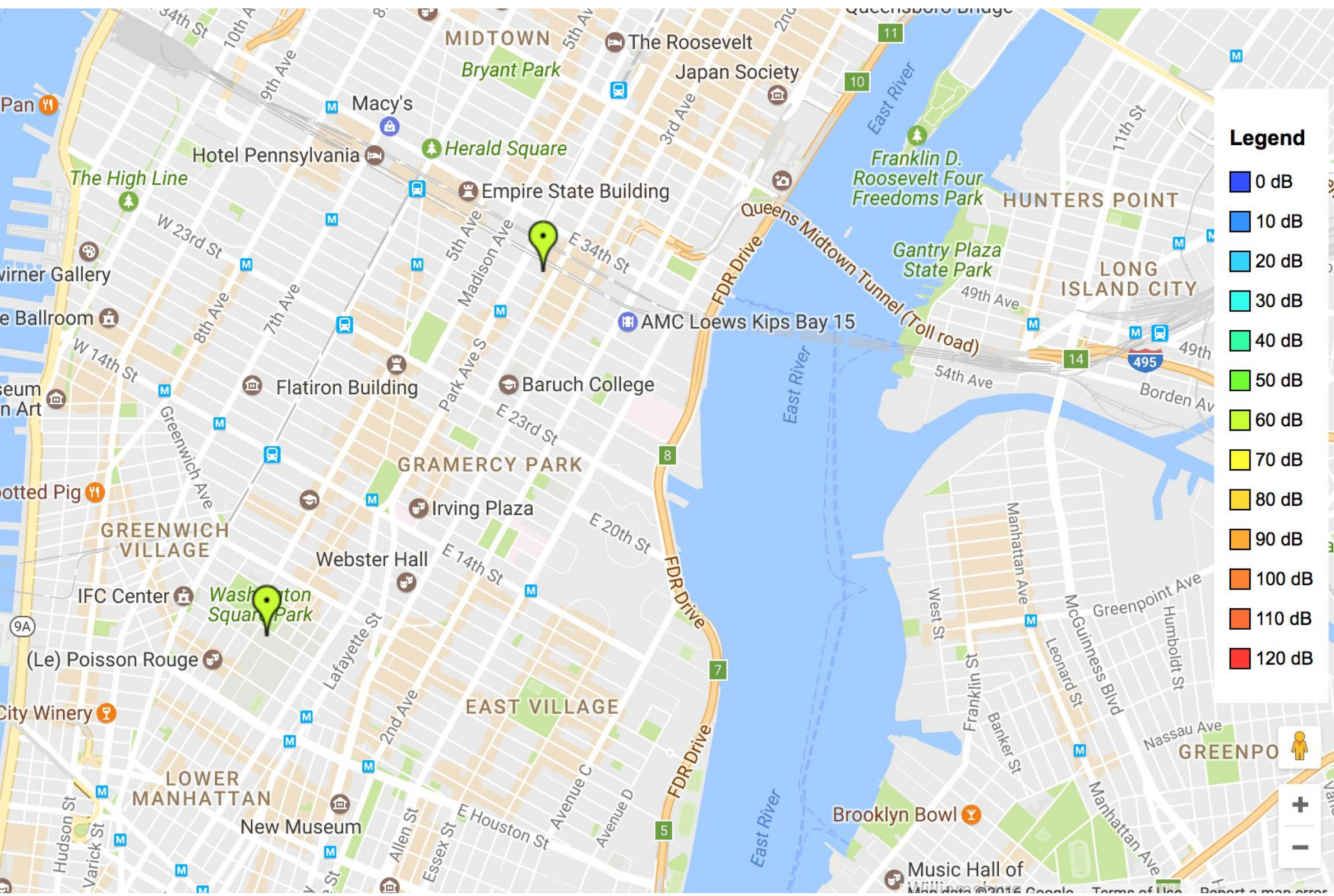
## Introduction

Our IoT Platform for Smart Cities is a prototype command and control system for large-scale coordination of mobile sensing resources. With this platform, large numbers of mobile sensors, including standard sensors such as GPS and microphone, as well as, mobile phone peripherals such as pollution sensors, can be issued fine-grained commands from the scheduler based on the state of the mobile sensing ecosystem.

The system is powered by a controller that tracks the state of connected clients and issues commands based on a custom scheduler. Data from the clients is stored in a MySQL backend, which can be queried and used to generate a web interface.

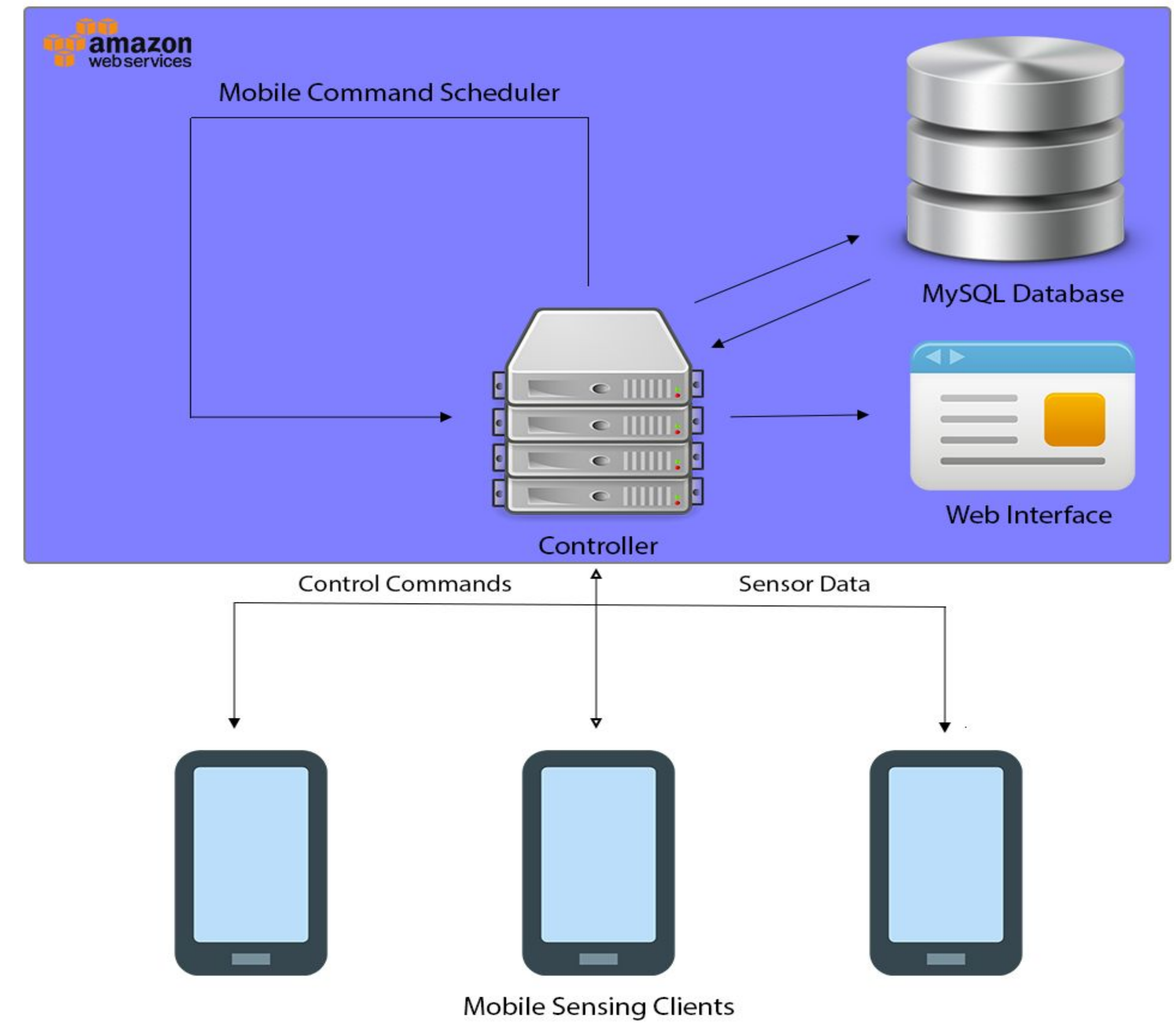


Mobile Interface



An example of noise map in Manhattan (data collected from android phones)

## Overview

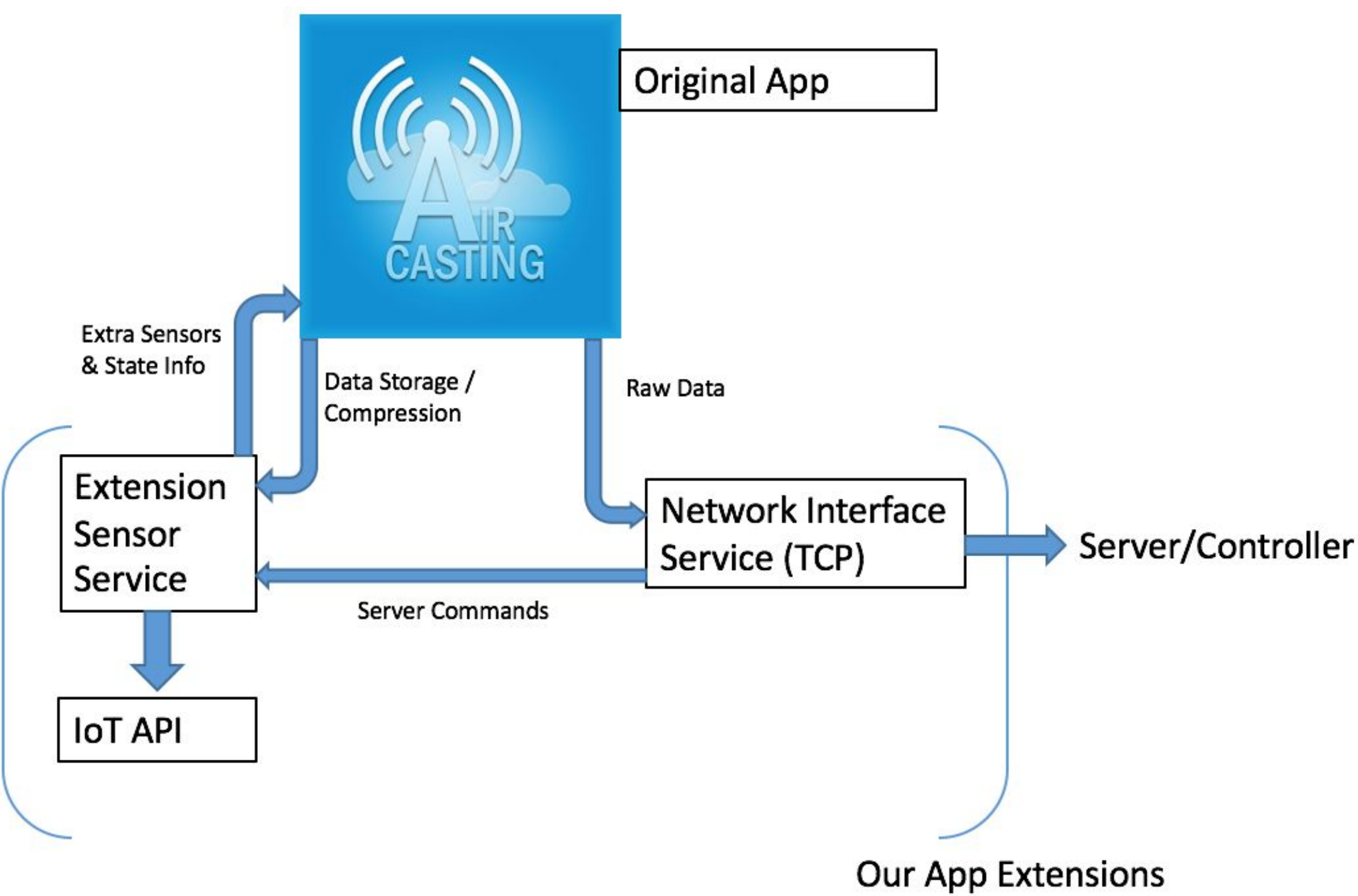


## Implementation Challenges

- Uniquely identifying mobile phones
- Multithreading server for high availability
- Thread synchronization for accurate scheduling
- Understanding the existing App to add extensions without making unrequired changes.
- Designing and implementing mobility support.
- Designing efficient message and signaling format for client/server communication
- Coordination across teams.

## Mobile App Details

- Mobile Clients based on *AirCasting* Android Open-Source Application
- Our IoT Platform extends the *AirCasting* App with two Android “Services”
- One Service provides access to the hardware sensors of the device/phone
- Another service provides interface to the Network Layer
- We maintain full backward compatibility with the original *AirCasting* App
- We generalize App capabilities to allow for any number of Sensors



Our App Extensions

## Server Details

- Two separate threads for running the controller and the data consumer
- Controller has a scheduler embedded to make decisions and send command to connected mobile phones periodically
- Parser for Json format data input, able to deal with compressed (gzip) or uncompressed data
- Seperate MySQL tables store sensing data for each device;
- Another thread queries database periodically (10 s)
- Google Map V3 API (javascript) generates real time mapping of information in html format
- Apache web server runs on AWS hosting the data visualization html page.
- The server visulization can be accessed from anywhere with any browser.

## Networking Details

- TCP sockets with a custom message structure facilitate command communications from the server to the client and data transfer back.
- Multi-threaded driver provides dedicated communication channels between the phone and server that are kept alive using custom keep-alive messages.
- Command message structure provides flexibility for future development of new commands.
- Clients with changing IP addresses due to changes in the mobile network automatically try to reconnect with the server.

## Acknowledgement

Professor Lakshmi Subramanian  
 Fatima Zarinni  
 Shiva Radhakrishnan Lyer