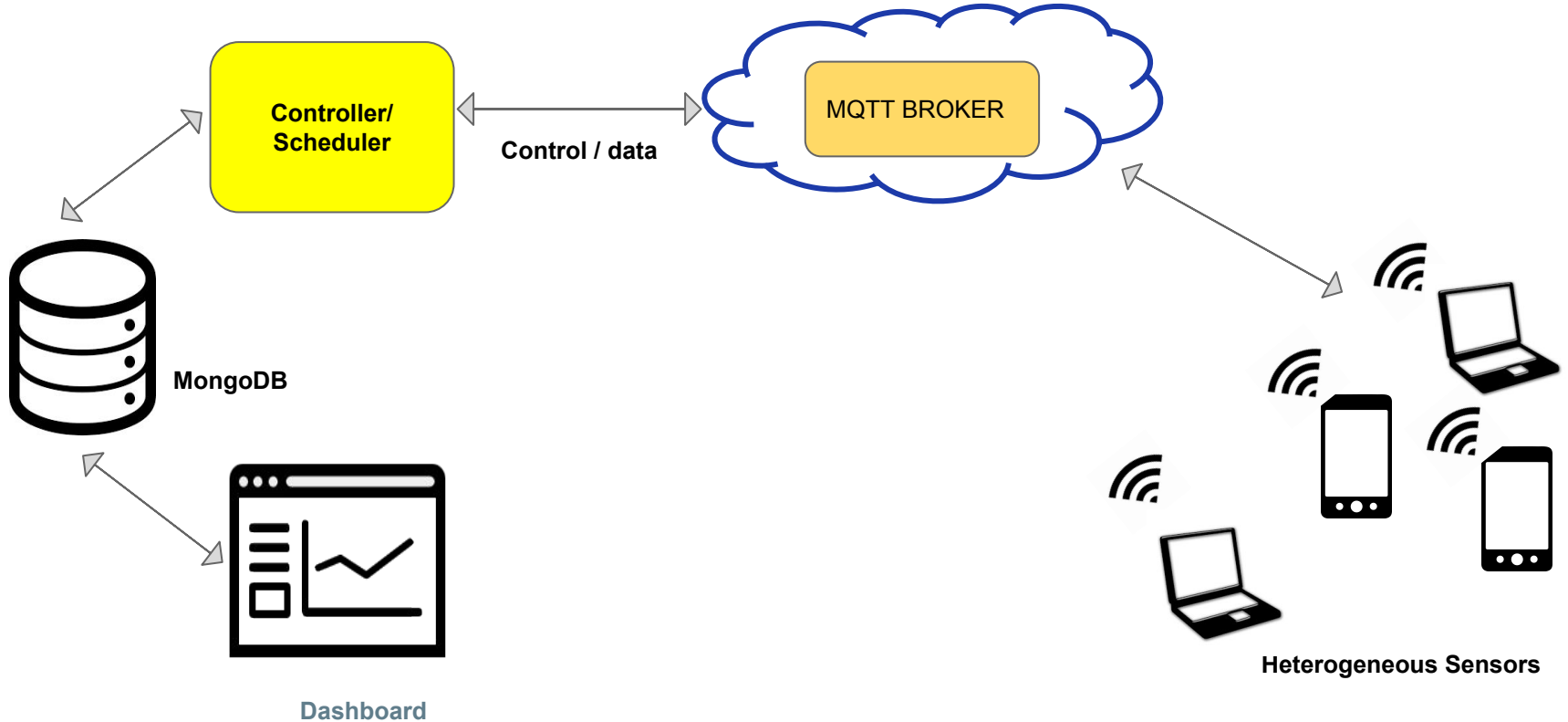




A crowdsourced spectrum sensing platform

Developed at Wings Lab, Stony Brook University
Santosh Ghosh, Ayon Chakraborty

Specsense Architecture

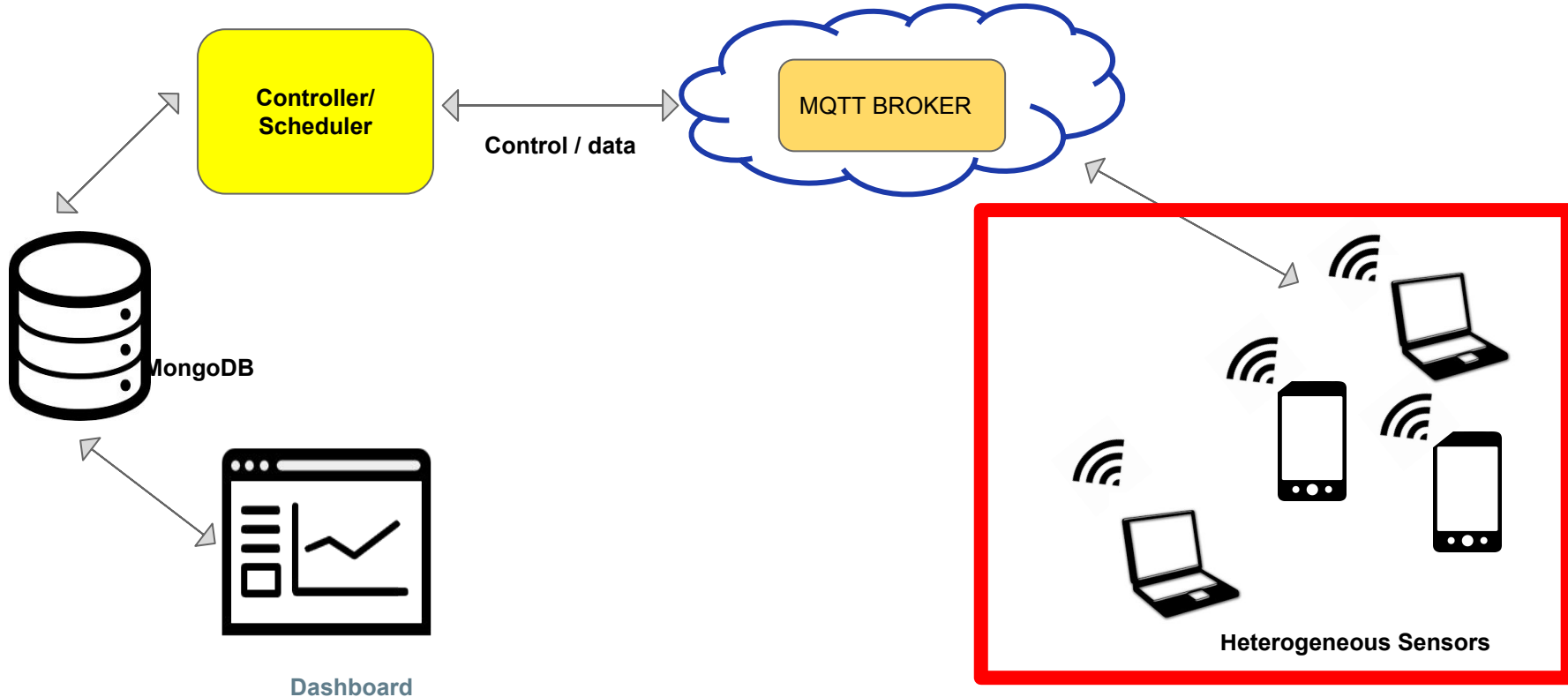


Work so far...

Component wise breakdown of progress.

Component Name	Purpose	Status
User Equipment (UE) Application in Android	Collect data from the mobile phones	Done
Python Controller	Control the UE by sending commands and collect the UE data in a database.	Mostly complete. Scheduler under development
Dashboard	Visual representation of UE data	Barebone framework developed.
Database	MongoDB database to store all UE data	Done

UE Application: Spectre



Spectre: Android Application

What is it?

This is an android application that aims to collect data from the mobile device and send it to the MqTT broker. It is built on top of two libraries.

Dependencies:

- 1) Python paho android service library
- 2) RTLSDR driver

See javadoc for a detailed insight as to how it works.

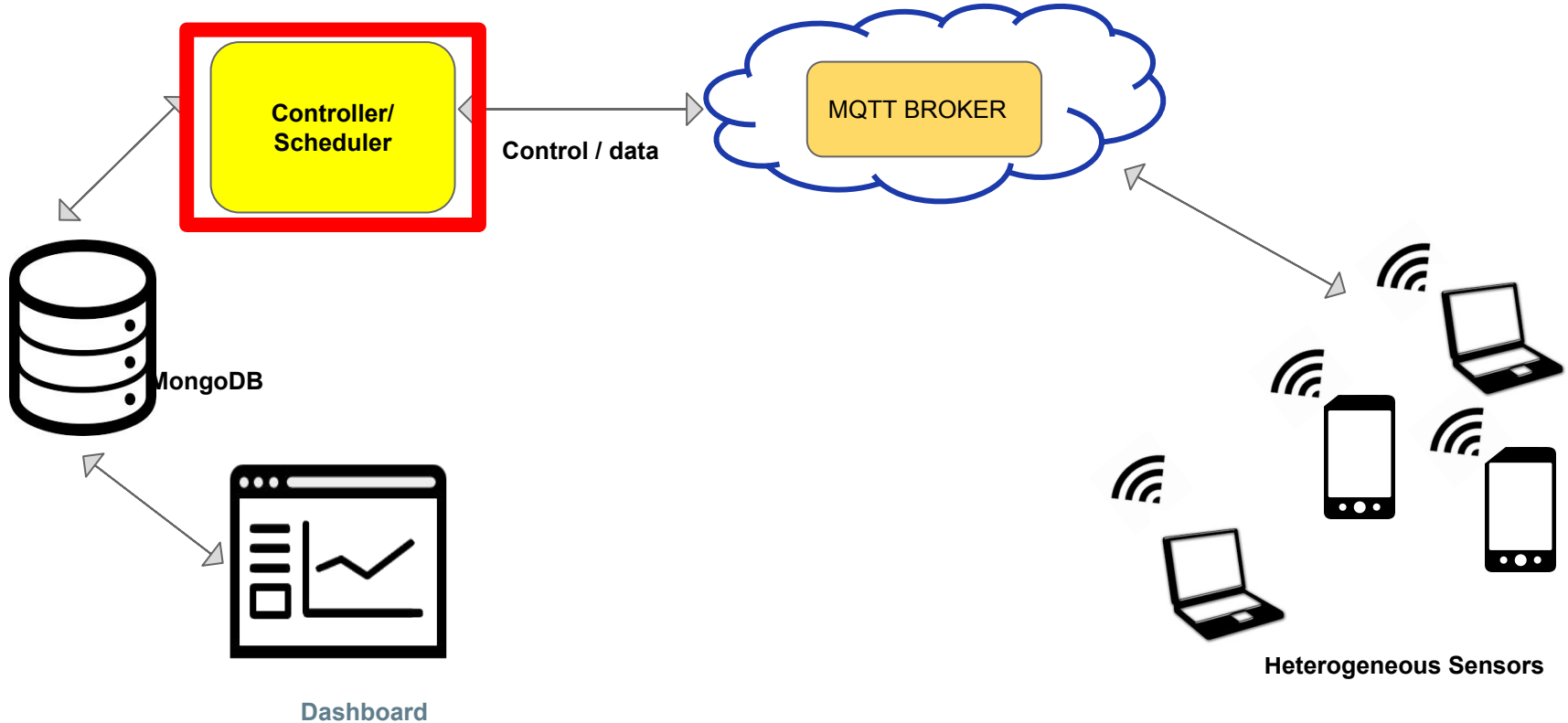
Spectre: Current Capabilities

- 1) Send data in two modes a) Wifi Mode (Testing purposes) b) RTLSDR mode (send spectrum data)
- 2) Handles reconnection/ lost connection to broker / network outage
- 3) Sends Hello/Bye message to let the system know when it is online/offline
- 4) Periodically updates location and battery power
- 5) Sends data over Wifi/Cellular whichever is available

Spectre: TODO

- 1) Service support. Currently we have a UI. But it should ideally run in a background service.
- 2) GPS service should not run always. It should run in two cases and then switch off by itself. The two cases are:
 - i) When controller sends a command to scan.
 - ii) When it periodically wants to update its location.
- 3) Include timing information in the northbound data so that performance metrics can be calculated later on.
- 4) Currently UEs are only mobile phones. Need to include laptops and Raspberry pi. Need to develop drivers for these types of UE

Specsense Controller



Specsense Controller

What is it?

It is a python program that is responsible for acting as a controller for scheduling all the data collection and which also acts as a consumer for all the data sent by the UEs.

Dependencies:

Python paho client for mqtt

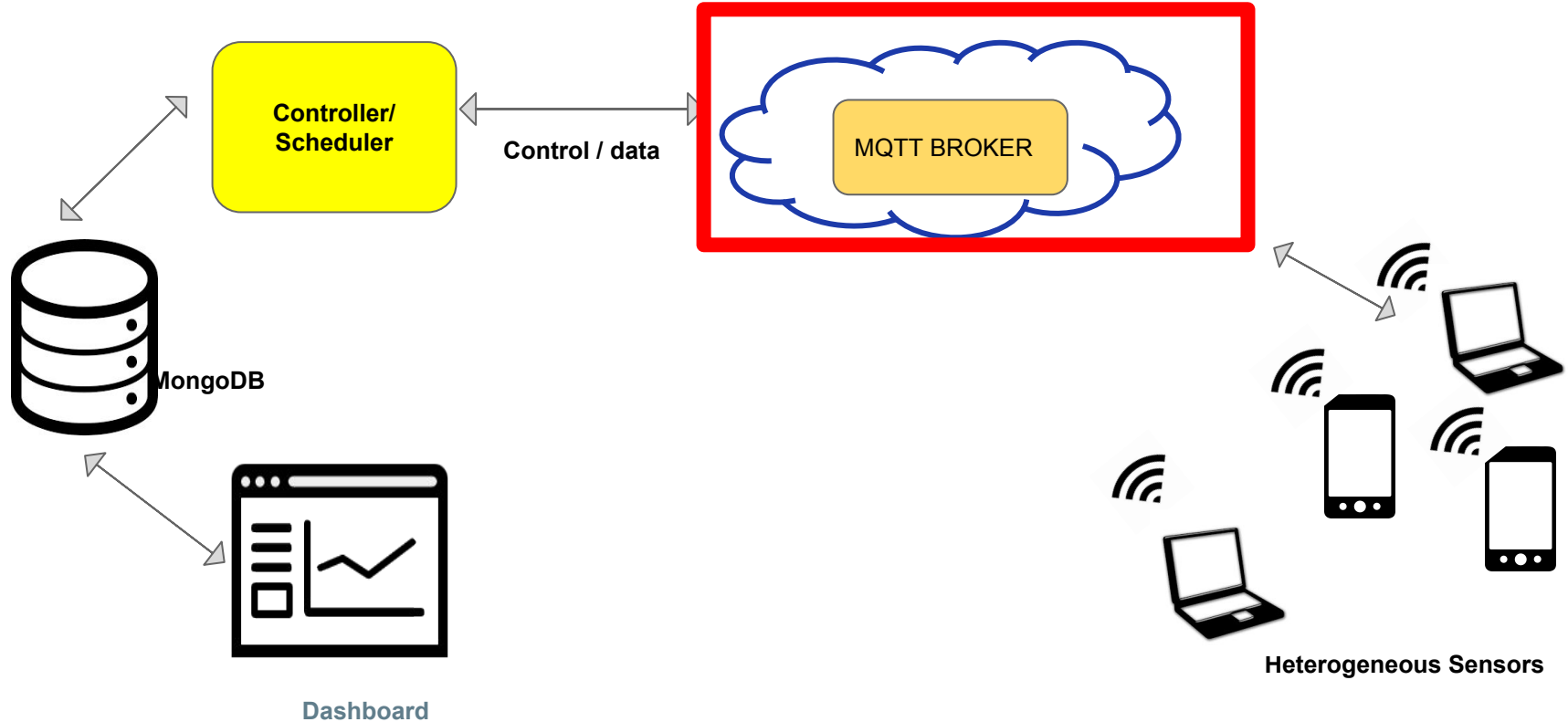
Specsense Controller: Current Capabilities

- 1) Collect all messages sent by UE and enter them in mongodb
- 2) Listen for incoming “manual” scan command that may come from the web front end.
- 3) Schedule scans on all the online UEs

Specsense Controller: TODO

- 1) Design the schedulers.
- 2) Design algorithms to select UEs. (Major work).

MqTT Broker: HiveMq



MqTT Broker

- 1) We are using Hive Mq broker.
- 2) We have two licenses for the broker.
- 3) Each broker can handle 2500 concurrent connections (as per the license obtained)
- 4) Detailed instructions for setting up in the hivemq setup file.

Future Work

Major:

- 1) Develop scheduler for efficiently scheduling scans.

Minor:

- 1) Develop RESTful API to query collected data. Support query like “What is the power of channel 15 at location X,Y?”
- 2) Enrich the visualization framework by incrementally adding graphs.

Tools Used

- 1) HiveMq : A commercial Mqtt broker.
- 2) Django Webserver
- 3) Android
- 4) MongoDB