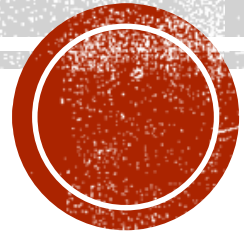
MONITORING SYSTEM



Aggressively Passive

Ben Nutter
Collin Gros
Joseph Mongome
Saul Armendariz

PROJECT OVERVIEW

- For the scope of this project given time constraints, we hope to have a basic notification system, with 2 sensors and an app interface.
- The 2 sensors we decided on was a temperature, and a motion detecting sensor.
- These sensors will be connected to a central processing unit, which will be our Raspberry Pi 3 (Model B), this pi will have a ui to help us set up the sensors and allow us to send push (in app notifications) or SMS.

KEY ARCHITECTURAL DRIVERS

- The central unit will be consistently monitoring our sensor values.
- We need a central unit to process the data or receive processed data and decide whether or not the information is within or outside our tolerances for triggers.
- After this data is interpreted what do we need? Depending on the user of our system, and how they choose to be alerted, it will be a push notification or SMS.

ARCHITECTURAL STYLES?

- Iterative Achitecture
  - We would have our initialization, which would be the setup of the central unit and which modules are connected to it, each iteration would be monitoring what the values of each sensor are.
  - These modules would each be able to trigger, so they would be running asynchronously but the data would synchronize at the central unit, process then reinitialize, which would be monitoring.
- Event-Based Architecture
  - Events for us would be when the sensors are triggered, based off that event it would then send a notification.
- We have chosen to go with Event-Based Architecture.

EVENT-BASED ARCHITECTURE

- We decided on this architecture because of the versatility, this type of architecture is normalized to unpredictability and asynchronistic, this will allow greater response times for our warnings and notifications.
- There are some risks though, mostly that the system can become opaque, in which EBAs are sometimes loosely coupled, this means that we can't always be sure what components are part of the system and which are their dependencies.
- Without having a firm grasp on this, there could be an event that triggers others leading to an unwanted chain of events.