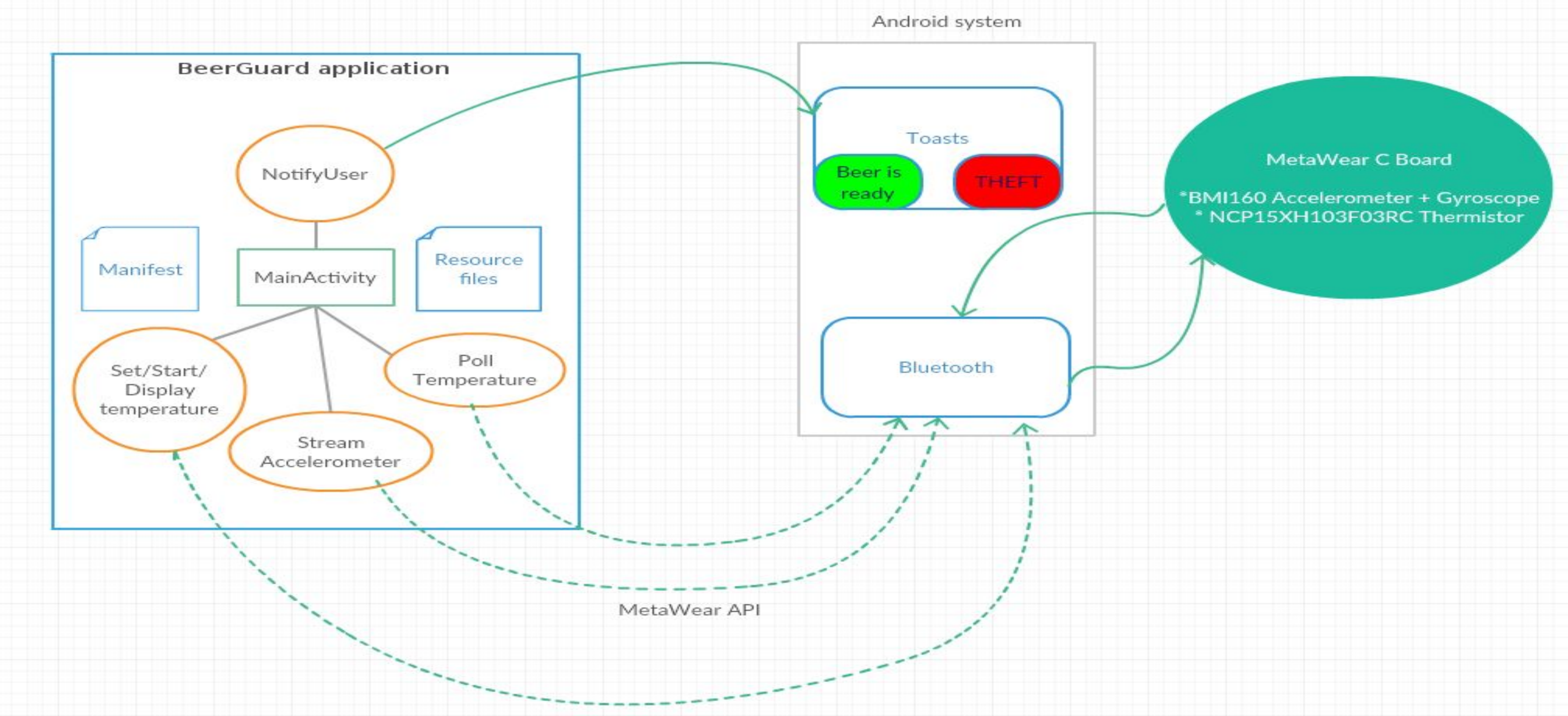
BeerGuard

High-level design

- Use MetaWear C board to devise a beer cooling/anti theft system
- Utilizes the thermistor & accelerometer to issue alerts when certain conditions are met:
    - Positive alert: Beer temperature is cold enough
    - Negative alert: Beer is being moved from the fridge before the cooling period is over

Design


BeerGuard application
NotifyUser
Manifest
MainActivity
Resource files
Set/Start/ Display temperature
Poll Temperature
Stream Accelerometer
Android system
Toasts
Beer is ready
THEFT
Bluetooth
MetaWear C Board
*BMI160 Accelerometer + Gyroscope
* NCP15XH103F03RC Thermistor
MetaWear API

What it does

- Working:
  - Anti-theft system seems to work as intended
    - Invokes a notification, 1 second vibration, alarm sound, and an alert
  - Temperature reporting seems accurate
    - Invokes a notification & alert when temperature is correct value
    - Alarm is disabled once temperature is correct value and user dismisses alert
    - Temperature reporting and alarm is disabled once user clicks “stop”
  - Seems to work alright as a service as well (once the app is closed)
  - Does not seem to crash but it’s probably vulnerable to something
- Issues:
  - Occasional, seemingly random inability for app to configure due to a Timeout error from the Metawear board. This is handled with exceptions, and is usually fixed by restarting the app
  - Also occurs with the FreeFall Demo for my phone, so I think I have a dud board and/or phone
    - Happens because of the .RESPONSE_TIME in Constants.java in the Metawear library is set to 1 second
    - Does not seem possible to change this without editing the Metawear libraries which is probably a bad idea

Architecture

- One main activity
- Call to retrieveBoard() on create
  - Which houses most of the accelerometer functions
- Various methods for alerts, notifications, vibrations, etc
- readTemp() houses temperature monitoring procedures
  - Temperature is permanently set to 10 degrees C
- Makes use of multithreading to ensure that temperature/alarm reporting continues smoothly after app is minimized.

User stories

Story	Acceptance criteria	Outcome
“As a user, I want start and stop monitoring & cooling my beer to an ideal temperature so I may feel safe in the knowledge that my drink is cooling properly & under supervision.”	● Must work when the beer is in a fridge, or ice bucket ● Must allow for a reasonably accurate starting/stopping functionality to signify that a beer is being cooled/has been cooled to the user's linking	Success: ● Board broadcasts surprisingly well even when inside a fridge ● Temperature reporting is accurate to three decimal places
“As a user, I want to be notified if my beer is removed from the fridge/ice bucket before the cooling period is over, which would be indicative of someone trying to take my beer.”	● Must work when the beer is in a fridge, or ice bucket ● Must report any significant fluctuations in accelerometer data from the sensor	Success: ● Board broadcasts accelerometer data from reasonable distances (about 15 feet, through a thin wall)

Misuser stories

Story	Mitigations	Outcome
“As a malicious user, I want to abuse the BeerGuard system by un-paring or otherwise tampering with the BLE connectivity between the MetaWear C board and the BeerGuard app.”	Ensure a relatively tamper-proof pairing mechanism between the MetaWear C board and the BeerGuard application. This may come in the form of a session token, derived from a cryptographically secure hash function, to be alloted to the user on every new "cooling" phase	Could not come up with a way to validate board aside from MAC address, which may easily be spoofed and is the sole identifier of the board.
“As a malicious user, I want to abuse a potential vulnerability in the anti-theft system so as to illegally procure someone else's beer without their knowledge”	Ensure that the anti-theft (accelerometer streaming) component is neither too sensitive, nor too lenient when used to issue theft warnings. This can be deduced with manual testing.	Impossible with current board, it is not sensitive enough. However, you would have to move very slowly whilst holding the sensor very steady, so you’ll probably get caught stealing anyway

Live demo

Questions?