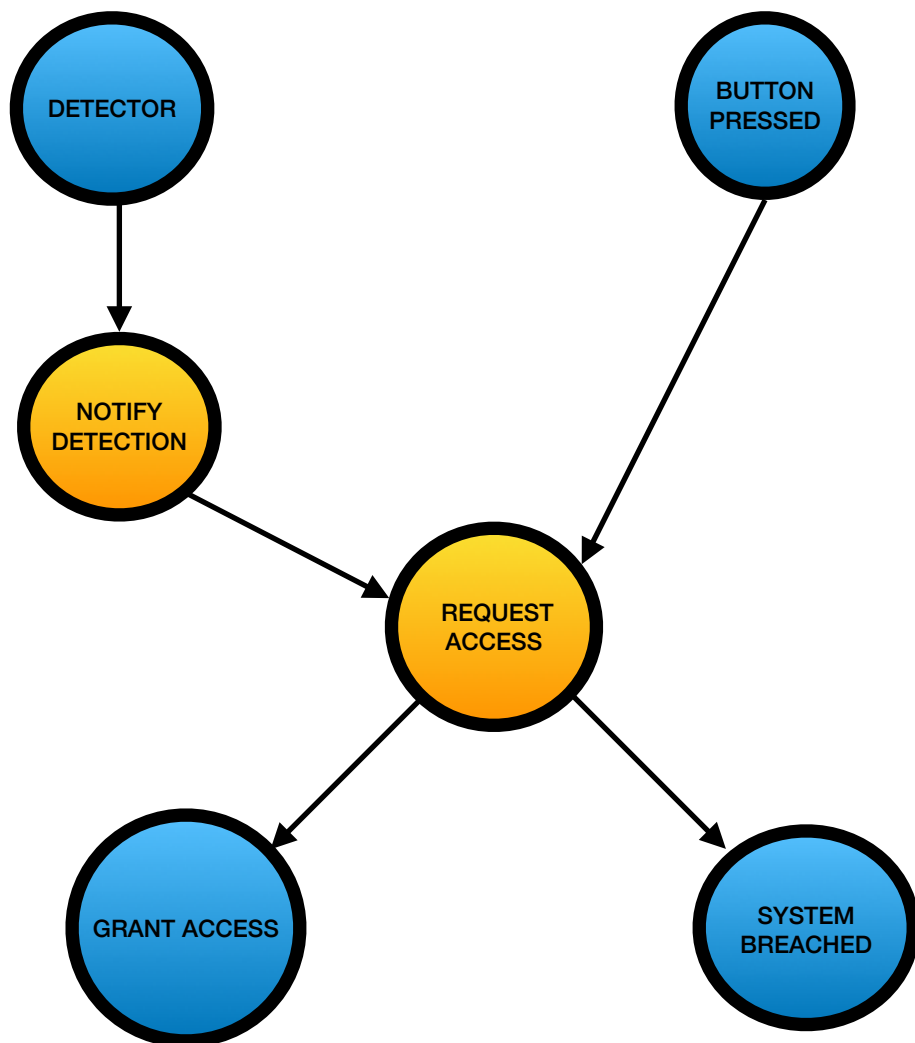
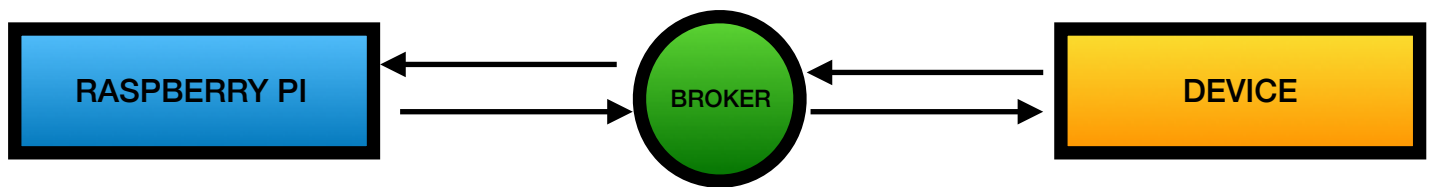


SPACEMAN SECURE

By Mohaimen Syed

SPACEMAN-SECURE is a security system that alerts the clients who are subscribed to the system if there is a someone/something within close proximity of the system and also detects if there are any intruders subscribed to the system. When the button on the raspberry pi is pressed a access request is sent to the client and if the client enters the correct passcode on his device, access is granted otherwise the all clients are notified of a system breach and also triggers an alarm to notify of a potential intruder.

MQTT PROTOCOL



This project demonstrates the use of two nodes – A device that is capable of compiling a python script and a raspberry pi model 3B+. The rpi is connected to a GrovePi+ and also a LCD RGB Display screen, two LED lights, a Button, a Buzzer and an Ultrasonic Ranger. The nodes communicate with each other using the MQTT Pub/Sub Protocol, via a broker which is "eclipse.usc.edu".

Initially the LCD display is green and show "SYSTEM LOCKED". Data is sent to the devices from the raspberry pi with the reading on the ultrasonic ranger and lets all clients know that there is an alien nearby, and when the button is pressed a request is send with the correct passcode. If the client enters the correct pass code, access will be granted turning on the blue LED and the LCD display screen will turn blue and show "ACCESS GRANTED". Otherwise if system is breached, all clients will be notified, the red LED light will flash and the buzzer will go on and off signaling an alarm, the LCD display will turn red and show "SYSTEM BREACHED".

The limitations faced include the i2c race case which I tried fixing using threading and locks but the instruction from the previous lab issues did not help much. The system cannot detect the specific device that has entered the wrong passcode. The system also does not inform the time when access was requested.

Each device that is subscribed can be issued a unique serial number. We can use time logs to enter the time of request and access and the Raspberrypi can store the serial numbers of all clients in a database and notify all clients who asked for permission and at what time and date, which can also be entered into a database. The lessons I take from implementing this system is that a good portion of time need to be spent not only adding new features but also running test case to ensure a flawless and error-free and secure system.