

# Prototype 1 – Initial Testing

Gentleman's Guild of Engineering Excellence

PREPARED FOR 1103A – ENGINEERING DESIGN

NOVEMBER 03, 2024

## **Table of Contents**

<b>Table of Contents</b> .....	2
1. Foreword .....	3
2. Prototype Summary .....	3
A. Sequence Diagram .....	4
3. Summary of tests.....	5
A. Compare Coordinates and Validate Authority.....	5
B. Documenting Entry via Live Capture .....	6
Armed Test Example Case:.....	6
Disarmed Test Example Case: .....	7
C. Location tracking based on API.....	7
D. Breach and Admin Notifications. ....	7
4. Simple Analysis of Critical Components .....	9
5. Updated Target Specifications .....	10
6. Updated Test Plan .....	10
Hard-Time.....	10
Soft-Time Tasks.....	10
7. Proof of Concept: SIM based No-Go Zones. ....	11
Objective:.....	11
Scope:.....	11
Requirements: .....	11
Methodology:.....	11
Success Criteria: .....	11

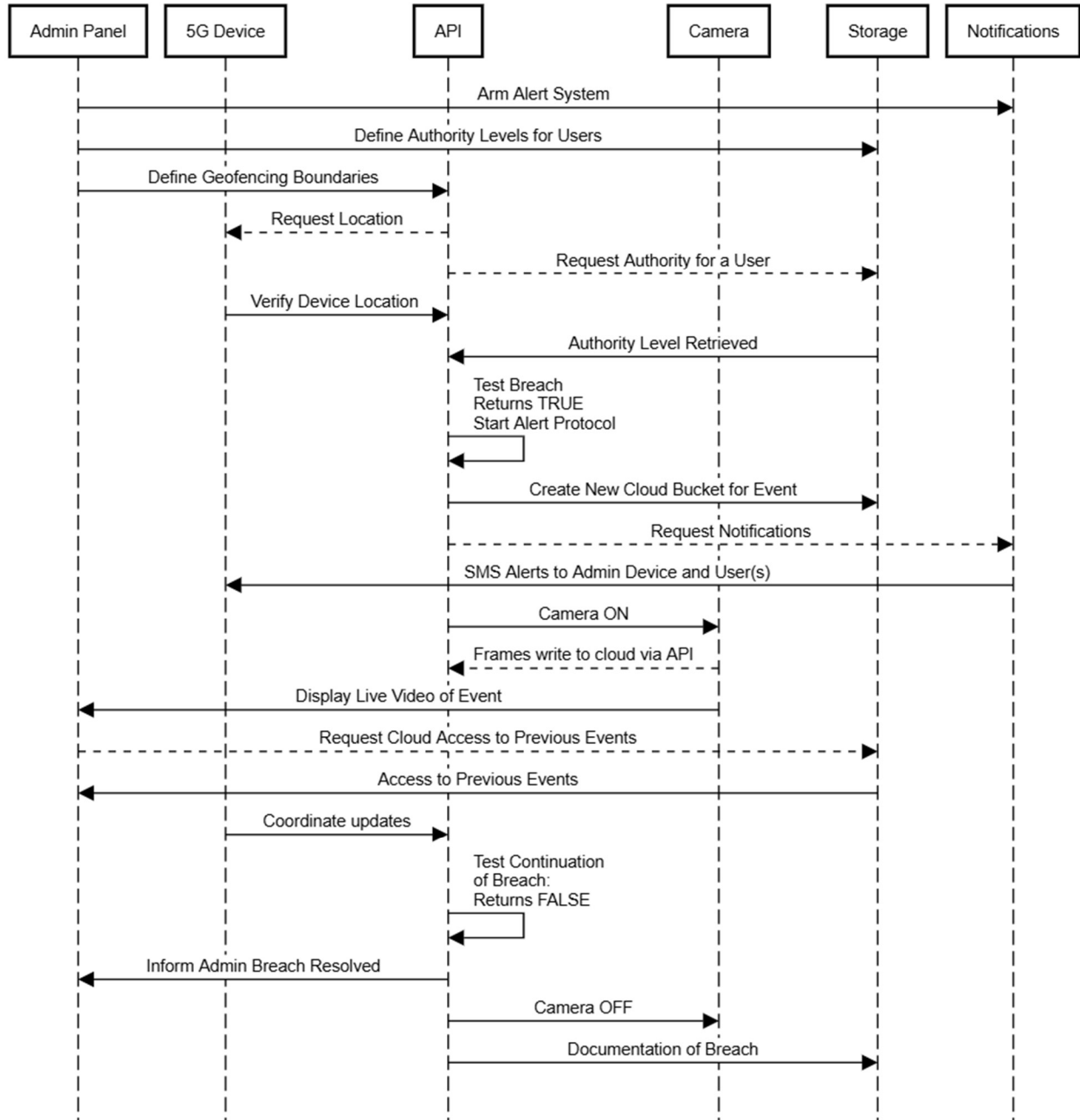
## 1. Foreword

In brief, our restriction zone security prototypes are currently limited to independent functions because of limited access to Shabodi's developer services. Thus, they have been tested in independence, and aspects related to data transfer are still in development. A sequence diagram has been included in this deliverable for the reader, outlining the order of function calls for operation clarity.

## 2. Prototype Summary

The purpose of this initial prototype is to establish the relationship between location data gathering and geofencing analytics, as well as its downstream effects. Functions of our rugged security prototype include, gathering of 5G device location, coordinate and authority level validation, and live capture of unwanted entry events. Our security system works with Shabodi's location API to gather coordinate information of devices connected to a local 5G network. A separate function evaluates the authentication level of the device and compares its coordinates against set boundary zones stored locally. If the device location registers an alert, this is received by an alert subsystem which distributes physical and digital alerts to administrators and the physical device in question. The alert subsystem will act as a trigger for object data collection in the form of local video capture. Data written to local files will also be linked to an object bucket storage file on google cloud using their cloud API. Regulation of video resolution is to occur via Shabodi's bandwidth API. The resolution of the unwanted entry (UE) and thus disarming of the alert system is understood, and detailed in the final section of the deliverable, however, is currently under development.


## A. Sequence Diagram



### 3. Summary of tests

#### A. Compare Coordinates and Validate Authority

In the first test, the zones “zone A” and “zone B” were created using random coordinates and were assigned clearance levels of 5 and 3 respectively. A SIM card with an ID of “12345” was assigned a clearance level of 5, and the location of the SIM card was set to within the coordinates of zone A. When the code is run, the message “SIM 12345 has clearance to enter Zone A” is printed.

```
41  
42  sim_card = SIMCard(sim_id="12345", clearance_level=5)
43  sim_card.update_location()
44
45  check_clearance(sim_card, zones)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\devmy\GNG1103> & C:/Users/devmy/AppData/Local/Microsoft/Windows/
se"
SIM 12345 has clearance to enter Zone A
PS C:\Users\devmy\GNG1103> █
```

In the second test, the location of the zones and SIM card remained the same, as well as the clearance level for the zones. The clearance level of the SIM card was changed to 2. The following message was printed: “ALERT: SIM 12345 with clearance level 2 entered Zone A with clearance level 5”.

```
42  sim_card = SIMCard(sim_id="12345", clearance_level=2)
43  sim_card.update_location()
44
45  check_clearance(sim_card, zones)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\devmy\GNG1103> & C:/Users/devmy/AppData/Local/Microsoft/Windows/
se"
ALERT: SIM 12345 with clearance 2 entered Zone A with clearance level 5
PS C:\Users\devmy\GNG1103> █
```

In the third test, everything remained the same except the location of the SIM card was changed to a set of coordinates that fall outside out of both zones. The following message was printed: “SIM 12345 is not within any restricted zone”.

```
21     def update_location(self):
22         self.latitude = 45.746 # Replace with actual latitude
23         self.longitude = -73.983 # Replace with actual longitude
24
25     # Check if SIM card clearance level allows access to a given zone
26     def check_clearance(sim_card, zones):
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS C:\Users\devmy\GNG1103> & C:/Users/devmy/AppData/Local/Microsoft/WindowsApps/python3.9.0/python.exe
SIM 12345 is not within any restricted zone
PS C:\Users\devmy\GNG1103> |
```

## B. Documenting Entry via Live Capture

Local video capture devices allow for the capture of event-specific still frames and video. These video captures are stored to local file paths. In independent operation, the camera subsystem responds to alerts. This is verified by a Boolean test that requires user to input the alarm status. If input is ‘yes’, a frame is saved by the camera and saved to program file location.

*Armed Test Example Case:*

```
What is the status of the alarm? (ARMED or DISARMED responses only)
ARMED
The status of the alarm system is ARMED
True
```



Figure 1. Capture from Webcam saved to local file. True statement sends console a receipt that video frame is being read.

*Disarmed Test Example Case:*

```
What is the status of the alarm? (ARMED or DISARMED responses only)
DISARMED
The status of the alarm system is DISARMED
```

Local video is also activated on an armed/ disarmed basis, with frames written to the program file location at consistent intervals until the arming case resolves – currently in development. (As of current testing, multiple frame video capture resolves occurs only when input of “q” is received).

## C. Location tracking based on API

Location tracking based on API use is the core foundation of what allows the base function of the program to do its job. One major roadblock has prevented serious progress in this prototype testing. Shabodi’s API for sim tracking requires an access token, this access token is unable to be generated (a collective issue shared among all project groups). Though this roadblock is halting serious progress in this prototype testing it has not completely halted it. API’s that use sim tracking are all in ownership of private companies that we do not have access to. However, there are some tracking API’s that do not require sim card use.

Initial testing of code using an alternative API gave a confirmation that the code can reach an API and retrieve data, at set time intervals. Testing 0.5 second intervals, 1 second intervals and 5 second intervals, data was able to be retrieved. Retrieval of this location data was able to be stopped with a kill command as well.

This initial test would be considered LoFi and focused. It is stated as Low fidelity as the API found was not able to accurately determine the location of the device (servers were only present in the USA and Europe for locating). This although showed the code can retrieve data from an API at set intervals.

## D. Breach and Admin Notifications.

1. The code entered an infinite loop, intended to send a "Zone Breach Alert" notification every 5 seconds until manually stopped.
2. For each loop iteration, the script:
  - Send a "Zone Breach Alert" notification using the Pushcut webhook URL provided.

## Initial Testing of Security Functions

- Checked if the request to Pushcut was successful.
  - Printed a success message to confirm the alert was sent or an error message if it failed.
  - Increment a counter each time a zone breach notification was sent.
3. After sending 5 consecutive "Zone Breach Alert" notifications:
- The script sent a different "Admin Alert" notification to a designated admin.
  - Printed a confirmation that the admin alert was sent successfully (or an error message if it failed).
  - Reset the counter back to 0 to start counting for the next set of zone breach alerts.

This process means the test successfully simulated a repeated alert system, escalating to an admin notification if a zone breach alert is sent 5 times in a row. The drawback of this test is that I only have one phone to test it on, so the admin notification, and the breach notification were both sent to my phone, as the app needs a membership, so it won't display the first part of the message. Further refinement will be done using another platform with the intent of developing an app for notifications to user and admin devices.





## 4. Simple Analysis of Critical Components

### 1. Data input to the main system

Data input provides the information necessary for the system to function. For example, verifying device location, defining zone boundaries and setting authority levels for the users all require input data.

### 2. Data readily available from cloud for administrator access.

Administrators need the ability to recall previously stored data. Large network strain may decrease both the ability of our program to write to the cloud, as well as decrease the ability of administrators to view video information stored within the cloud. Cloud object storage must be thus readily available, noting that it cannot be provided on an (unsupervised) subscription basis.

### 3. User information captured.

UE events must trigger video capture at the initial instances of the breach, else there is the risk that the window of opportunity (where information is most readily available) will close. It is possible that increased latency due to network demands may reduce the probability of data acquisition, and so it is essential that video capture protocols are initiated at the front end of the program.

### 4. Accuracy of Location Data

Accurate location data is essential for enforcing zone boundaries as it prevents false alarms and ensures alerts are triggered only for genuine breaches. Without precise location data, the alert system could be compromised leading to both missed detections as well as unnecessary alerts.

### 5. Alert Activation

Alert activation is critical as it triggers the alert from detected breaches, notifying user about unwanted entries. Effective alert activation ensures timely responses to breaches, which helps mitigate security risks.

## 5. Updated Target Specifications

Target specifications have largely remained identical in comparison with initial design criteria, with several changes made based on new information surrounding API capabilities from client meeting 2:

- Position updates are to be requested in 5 second intervals for first comprehensive prototype. Once near-final prototype is created, stress test of 1 or 0.5 second update will commence, nearing real time tracking.
- Location is to operate within an acceptable accuracy of centimetres (~10cm radius).
- UE events trigger user alerts with an acceptable reaction time in milliseconds (<5ms).
- Alerts and UE identification is to be accurate to within the 95<sup>th</sup> percentile.
- Total bandwidth usage should operate below 15% total capacity.
- Video capture written to file at resolution of 480p and 10fps.

## 6. Updated Test Plan

### Hard-Time

1. Gathering data from SIM location: Nov 9 - Devin, John, Ghadi, Gordon, Sean
2. Sending data to the cloud: Nov 9 - Devin, John, Ghadi, Gordon, Sean
3. Testing multiple device tracking in real time: Nov 13, Devin, John
4. Integration of Location API into required functions: Nov 13 - Devin, John
5. Integration of Camera functions: Nov 13: Gordon, John
6. Boundary system: Nov 16 - Sean, Ghadi
7. Alerting from boundary: Nov 16, Sean, John  
(ability to identify SIM in zones/trigger system when in a no-go zone)
8. Detect person in an area: Nov 16 - Ghadi, John,
9. (reduced bandwidth)
10. Recognize person in area: Nov 16 - Gordon, Devin
11. (increase in bandwidth of cameras)
12. Send data to soft time: Nov - 23 Sean, John
13. UI development/refinement: Oct 27- Nov 27, Devin, John, Ghadi, Gordon, Sean

### Soft-Time Tasks

1. Include Bandwidth API to relevant functions Nov 9 – Gordon, Ghadi

2. Live Location Updates and Alert Responsiveness: Nov 9 - Devin, Sean
3. User Interface Rough Outline (login, cloud library access): Nov 9 – John
4. Admin Live Video Access Upon Request: Nov 11 – Gordon
5. Admin controls (dashboard options): Nov - 11 Devin, Ghadi, John

## 7. Proof of Concept: SIM based No-Go Zones.

### Objective:

Demonstrating the real-world feasibility and uses of sim location-based tracking used to restrict and notify users of unwarranted entry into areas.

### Scope:

- Tracking a device based on SIM data
- Setting and identifying zones and entries in/out of zones
- Alert users of unwanted entries and alert admins of entries
- Use of video feed to identify further identify unwanted users in zones

### Requirements:

- Reliable gather SIM data for location tracking
- Setting and storing user ID, Security clearance, and SIM data
- Ability to set multiple zones and their security clearance
- Ability to differentiate between security clearances in users and zones
- Reliably change bandwidth of cameras to be able to recognize people

### Methodology:

- Technology stack: Add location detection based on SIM data, as well as storing data with user ID's
- Testing: Scalable prototypes. One person, one zone. Two people two zones, etc.
- Limitations: Initial testing will be based on backend inputted location data, until API token is working.
- Limitations: Testing is limited to use of team technology (5 sim cards, 10 cameras, laptops and cell phones)

### Success Criteria:

- System must accurately identify entries in and out of zones >99% of the time
- Alerts must reach user phone within 1 second, admin phone within 6 seconds.