ECE 445

Fall 2024

September 19, 2024

Project Proposal - Early Response Drone for First Responders

Kevin Gerard, Lohit Muralidharan, Aditya Patel
Team #11
TA: Manvi Jha

Table of Contents

1 Introduction

1.1 Problem

Every week, UIUC students receive emails from the Illini-Alert system regarding crimes that are committed, fires that are occuring, and other dangerous situations to be aware of. With the latest reported median response time of first responders to a 911 call being over 6 minutes in Champaign County [1], the situation to which emergency personnel are responding can drastically change from the initial details that were provided. This problem is even worse in rural areas where, for example, the average response time for Emergency Medical Services (EMS) is over 14 minutes [2]. To best be able to manage the event, first responders need as much accurate information as they can possibly receive. This way, the situation can be handled in a timely manner and the safety of everyone involved is prioritized. Thus, having eyes on the area before arriving can provide emergency response personnel with valuable information about potential hazards, individuals involved, and the severity of the event.

Over the past decade, the use of drones by first responders has significantly increased. For example, the city of Fremont, California's police and fire departments primarily use drones for reconnaissance, documenting crime scenes, and helping with search and rescue operations [12]. However, the use of drones as an early response technology, and in a sense acting as the first responder, is not yet a widely explored concept. Theoretically, this type of drone will reduce response times, improve the safety of all involved individuals, increase efficiency and prioritization, and massively aid understaffed departments [15].

1.2 Solution

Our solution is to construct a cost-effective drone that first responders can deploy and immediately fly to the location of an emergency event. While en route, they could use the drone's on board camera and computer vision capabilities to assess the situation at hand. There are multiple scenarios in which this drone could be particularly beneficial, such as:

- Police: monitor crime scenes and track suspicious individuals; provide aerial surveillance for events with a high density of people (such as sports games, concerts, or protests) to ensure everyone's safety
- Fire: monitor the spread of fire at the location; obtain information on what kind of fire it is (electrical, chemical) and any potential hazards
- Medical: assess the type and number of injuries suffered, and locations of patients

Our drone system consists of 4 different elements: a cloud storage, a backend, a frontend, and the drone itself. In order to create a baseline early response drone, we need to be able to control the drone as well as receive information from the drone such as capture frames, altitude, roll, pitch, and yaw. The capture frames and data will be visually displayed in the frontend. However, this data bundle will first be stashed onto a cloud storage, and when the backend is ready to receive the data, it will retrieve it. If time permits, we want to perform machine learning processing using object tracking and detection models on the backend software. The other data transmission that occurs is the sending of command signals from the frontend to the drone itself. In other words, whenever there is a keyboard click, we can visually see the key click which is uploaded to the cloud storage.

Note: while we do currently plan to use cloud storage via Firebase, as referenced above, we are currently researching possible ways to implement our design without the need for cloud storage. In this design, the drone's cellular chip will directly communicate with our C++ backend for data transmission.

1.3 Visual Aid

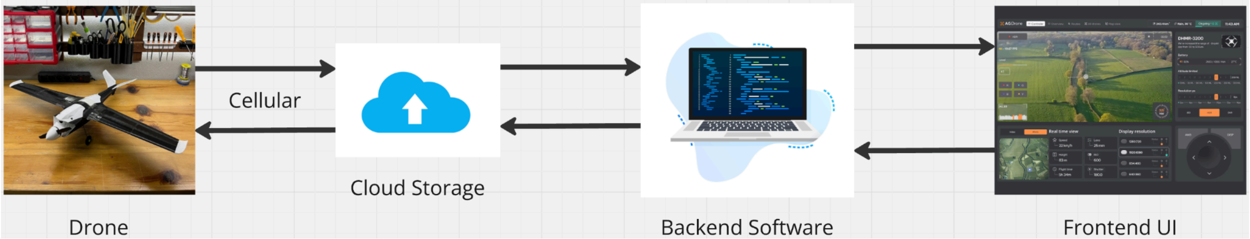

Figure 1: Primary components of the design

The fundamental components of our drone system are pictured above in Figure 1. The drone itself will either be constructed from foam board and will house the core electronics, such as the Printed Circuit Board (PCB), camera, cellular chip, servos, battery, and motor. After powering on the drone, the user will be able to connect to it by interacting with the intuitive User Interface (UI) run on the frontend software on their computer. The backend software will establish a connection between the frontend and embedded software running on the drone and transmit data between two through the cloud storage. Once this is completed, the user will be able to view the feed from the drone's onboard camera, various sensor data, and other critical flight information on the UI. Flight controls, such as yaw, pitch, roll, and throttle, can be inputted via keystrokes on the computer, and this data will then be transmitted through the network nodes to the drone over 4G cellular connectivity. Upon receiving the data, the drone's primary control loop will process it and act accordingly.

1.4 High-level Requirements

1. Minimum Video Frame Rate: We anticipate our system will be able to display a video feed from the drone's onboard camera in 6 frames per second (FPS). However, as a minimum we plan to achieve 3-4 FPS.
2. Nodal Software System: Our design must comprise 3-4 primary nodes that efficiently intercommunicate to properly transmit and receive various data. These nodes include a C++ backend, a TypeScript frontend, the embedded drone software, and (optional) Firebase cloud storage as a medium between the drone and backend.
3. Preemptive Warning: For the safety of the drone operator and nearby individuals, the drone will monitor crucial parameters such as altitude, velocity, and battery levels to provide real-time alerts to the user via the frontend UI.

2 Design

2.1 Block Diagram

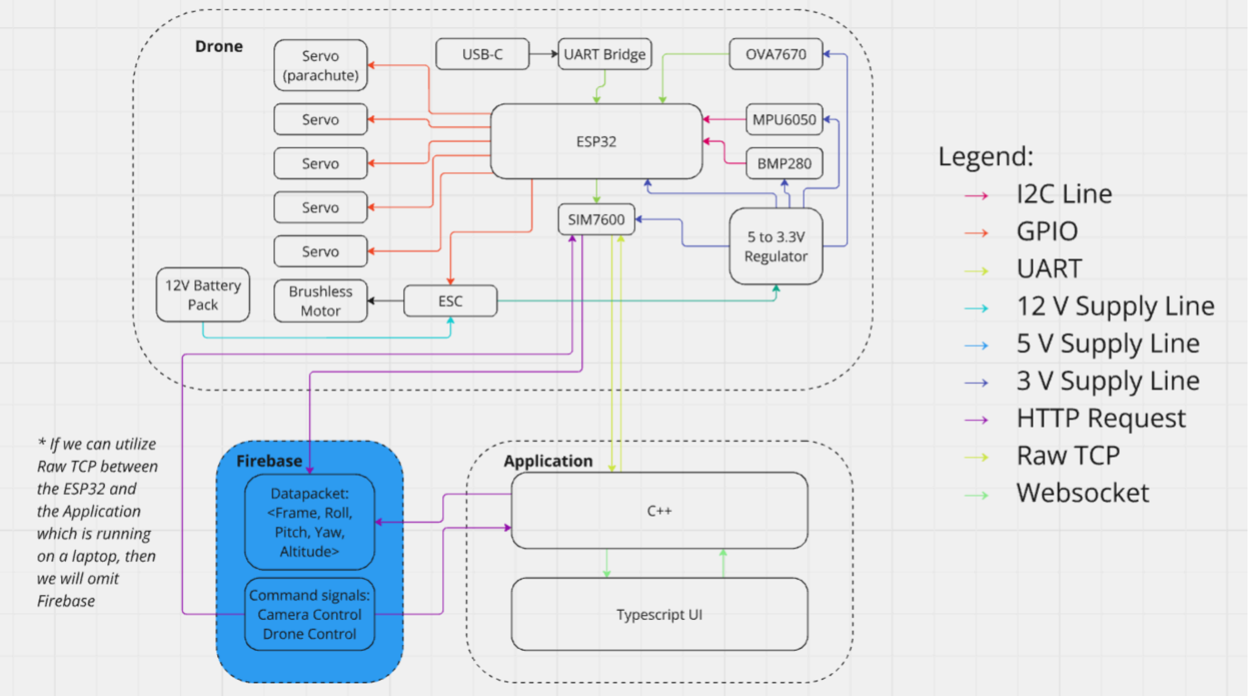

Figure 2: High-level block diagram for the overall drone system

2.2 Subsystem Overview:

In our overall project, there are 3 subsystems: The drone, the firebase database, and the application. Each of these subsystems have their own separate roles. The drone is used to retrieve commands and send sensor/visual data, firebase is used to store the data, and the application, which comprises a C++ backend, and a TypeScript UI, is supposed to process the data and provide a visual interface. **It is also important to note that the application lives in a laptop, and that our team may be able to completely remove the Firebase backend**.

Command Signals:

As mentioned above, we will have a visual interface that acts as a command control built on TypeScript. From this command control, we can monitor key presses (W, A, S, D) for controlling the drone and arrow keys for controlling the position of the camera. These monitored signals will then be sent to the C++ backend to be stored on a queue using websockets. If we are completely able to remove the firebase cloud storage, we can then immediately send this data using raw TCP sockets to an SIM7600 module. In order for any data transfer to occur, whether it is between websockets, raw TCP/UDP, HTTP request, the 2 devices in which the data transfers must lie under the same network. The reason why our C++ backend can interface with the TypeScript frontend is because both applications share the same subnet/network ID by living under the same Wifi network. We are not sure if the SIM7600 and the C++ backend can share the same network because our laptop will be utilizing the UIUC Wifi network and the SIM7600 uses the cellular network. The only way to be completely sure is by purchasing the components and doing tests early on. However, if our conclusions are that the SIM7600 cannot connect with the UIUC Wifi, our backup plan is simply to store everything on a cloud storage and retrieve data from the cloud storage using the SIM7600 which we know is possible. Once the SIM7600 receives the commands, it will process the commands and send appropriate decisions to the ESC and Servos.

Sensor/Visuals Data Transfer:

The other transfer of data occurs in the reverse direction where the drone will use the SIM7600 to send sensor/visual data to the C++ backend. This, again, may use the Firebase cloud storage as a medium if we are not able to establish some communication protocol between the C++ backend and the SIM7600. The sensor data that we can retrieve are approximate altitude, from pressure using a barometer, and roll, pitch, and yaw from an IMU. We may add an accelerometer as well, but that depends on how complex we want this design to be. However, once the sensor data/visuals are sent to the C++ backend, we may run an object tracking model such as Deepsort to analyze the bird's eye view if time permits. This data is then queued and sent to the TypeScript frontend. On the frontend, not only are we able to send command signals through key types, but we are able to visually see what the camera sees, and visualize the roll, pitch, yaw, and altitude.

2.3 Subsystem Requirements:

1. C++ Backend: For the C++ backend, we are planning on utilizing the Boost Asio/Beast Library. We believe that the Boost library is an industry standard library and will work well since it supports various protocols. For example, we can create raw TCP sockets, Websockets, and HTTP requests. These can come in handy when we need a middle man between the drone and the TypeScript frontend.

2. Frontend: For the frontend, we will use TypeScript because like Boost, this scripting language is also widely adopted by industry. TypeScript react compared to other frameworks like angular and vue is extremely user friendly as it combines both TypeScript and html in a single file. We can also create powerful visualizations with ease using TypeScript.

3. Cloud Storage: The two largest cloud storages are Google Cloud and AWS. Firebase is a part of Google Cloud which also provides free usage for non-commercial usage. As of right now, the drone is not set up for commercial use which is perfect for research and development. Again, as mentioned above, this is used as a medium like the queues in the C++ backend. If possible, we are going to completely eliminate the need for Cloud storage because in reality, scaling cloud storage is extremely expensive.

4. Drone system: The drone system comprises various sensors and devices which is shown below:

Devices	Amount	Role
ESP WROOM 32	1	The ESP WROOM 32 is the brain of our drone. Essentially. This is what we use to command each sensor, machinery, and cameras to perform their roles. We can flash arduino C code to tell the ESP how exactly it should command each device connected to it.
IMU (MPU6050)	1	The IMU is used to provide a 6-axis. From this, we will send roll, pitch, and yaw data using the SIM7600 to the frontend through many mediums.
Barometer (BMP280)	1	The barometer can be used to provide us pressure values. There are methods online to provide an accurate altitude measurement using the pressure value received.
Cellular Module (SIM7600)	1	The cellular module will be the main source of data transmission and receiver from either the Firebase

		Cloud Storage or the C++ backend if the Cloud Storage is removed.
ESC & Brushless Motor	1	The ESC and Brushless motor is the main source of propulsion for our drone. We will have some sort of method to control the speed of this drone mid-flight using the Frontend.
Servos	6	We will require 6 servos for controlling the Flaps, Ailerons, Rudder, and Camera View.
OV7670	1	The camera will be utilized to produce frames to send through the SIM7600 so that we can perform ML processing and display to the Frontend.
Foamboard	TBD	We will be utilizing foamboard to create the overall structure of the drone. Foamboard is easily repairable and cheap compared to alternatives such as LW-PLA and carbon fiber.

Table 1: Drone hardware component descriptions

2.4 Physical Design

Our design is going to be a V-tail winged drone with a camera mount below the drone. This will require a total of 2 servos for the wings, 2 servos for the V-tail, 2 servos for the camera, 1 servo for parachute, and 1 motor for creating thrust for the drone. In terms of the wing, we plan to create a larger wing area than most drones for easier controllability for users during flight. It is also important to note that we will be utilizing foam board for the entirety of the drone. Foam board is extremely lightweight and replaceable compared to its alternatives such as LW-PLA 3d printing material and carbon fiber. This is mainly due to the fact that foam board is extremely easy to cut with tools like box cutters compared to carbon fiber and plastic. In the next couple sections, we will describe how to control the drone and how to generate lift.

For steering the direction of the drone, we utilize flaps and aileron. As mentioned earlier, our drone is a V-tail drone. Therefore, we have a total of 2 flaps and 2 ailerons. Each wing has a single aileron and each tail in the V-tail has a single flap. These control the roll, pitch, and yaw of the drone mid flight due to knockbacks caused by airflow. For example, if the left aileron is angled at -45 degrees and the right aileron is angled at 45 degrees, the plane will roll clockwise. If the left aileron is angled at 45 degrees and the right aileron is angled at -45 degrees, the plane will roll counter clockwise. Similarly, the tails control yaw and pitch. When both flaps are up, the plane will pitch downwards. If both flaps are down, the plane will pitch upwards. If the flaps are opposite of one another, this will alter the yaw in a similar fashion to the ailerons with roll.

In order to control the flaps and aileron of the drone, we utilize servos to control the flaps on the drone. Essentially, we utilize a servo that is connected to a rod, and the rod is connected to a piece connected to the aileron/flap. It is important to mention that the servos will be mounted directly on the wing for controlling the ailerons, and mounted within the fuselage of the drone for the flaps.

2.5 Drone Operation

Upon receiving power, the drone performs preliminary checks on its hardware components and establishes a stable cellular connection with the cloud/backend. Once everything is verified, the drone enters an idle state, transmitting sensor data and camera feed to the user while awaiting flight commands. During flight, the drone continuously monitors sensor data and communication stability, processing user commands to control its servos and motors. If any errors or communication losses occur, the drone will disable its motor, deploy a parachute, and notify the user before shutting down. The drone will also shut down upon receiving a user command to do so, halting all hardware operations.

2.6 Tolerance Analysis

Looking at our entire design, we anticipate that the drone’s cellular capabilities will be the most difficult to implement and pose the greatest risk to our plan. The cellular module we plan to use, the SIM7600, theoretically should support 4G connectivity and up to 5 Mbps upload speeds. This would be enough to stream video at a frame rate higher than we expect to achieve; however, there is not much information available on the usage of this chip in a drone setting.

Camera Data Calculation:
Resolution: 640x480 pixels
Color Depth: 24 bits per pixel (3 bytes per pixel for RGB)
Frame Rate: 5 FPS (as specified)

Raw Frame Size:
Pixels per frame = 640×480 = 307,200 pixels
Raw size per frame = 307,200 × 3 (bytes per pixel) = 921,600 bytes
Using the 10:1 compression ratio (JPEG Compression (Lossy)), the compressed frame size is:
921,600 / 10 = 92,160 bytes per frame

At 5 frames per second, the camera data rate will be:
Data rate for camera = 92,160 bytes per frame×5 frames per second = 460,800 bytes per second = 460.8kB/s

Data rate for sensors = 50 bytes per second

The total data rate for the camera and sensor data is:
460,800 bytes per second+50 bytes per second = 460,850 bytes per second

The protocol overhead is:
Overhead = 460,850×0.05 = 23,042.5 bytes per second ≈ 23,043 bytes per second
So, the total data rate is 0.483893MB/s × 8 = 3.87Mbps

The SIM7600 module supports LTE upload speeds of up to 5 Mbps. With the total data rate calculated as 3.87 Mbps, we are well within the limits of the SIM7600's upload capacity on a 4G LTE network.

This is within the 5 Mbps upload limit of the SIM7600 on an Long Term Evolution network, so the system should work under these conditions without hitting bandwidth limitations.

3 Testing and Demonstration

For the purposes of testing our project, we will spend a sizable amount of time unit testing each of the individual components involved in the drone's physical hardware, as well as electrical and software subsystems. To the best of our ability, we will ensure reliability, stability, safety, and robustness of our drone before flying it in the air. When we are confident that our complete drone system is ready for in-air testing, we will first need to register our drone with the Federal Aviation Administration, since we anticipate our drone will weigh more than the unlicensed limit of 0.55 lbs. This process is straightforward, but requires an additional $5 registration fee.

Upon registering our drone, we will then be able to fly and test our drone in-air. However, many high-traffic outdoor areas on campus, such as quads, do not allow the usage of drones. We have identified some local fields that should serve as ideal locations for testing our drone, since there is little to no civilian presence; however, we must first confirm with the university or local government that these spaces are free for drone use. For the Final Demonstration, we will be able to show the drone's bespoke software suite and basic electronics indoors, without flying the drone. We will then either supply a pre-recorded video of the drone's in-flight performance, characteristics, and features, or perform an in-person exhibition of our drone flying in the air if a suitable location is close enough.

4 Ethics and Safety

Ethical Issue: The drone's ability to provide real-time video surveillance and tracking in public or private spaces raises serious concerns about privacy. Since drones may operate in environments where individuals have a reasonable expectation of privacy, such as homes or sensitive locations, the deployment of the drone has to be properly considered.

- IEEE/ACM Code of Ethics: Both the ACM Code of Ethics (section 1.6) [3] and the IEEE Code of Ethics (section 1) [4] emphasize respecting the privacy and autonomy of individuals. The project must avoid infringing on people's right to privacy.
  Prevention Measures: To avoid privacy violations, the drone should have defined operational protocols, such as geographic boundaries or "no-fly" zones, to avoid areas where privacy could be compromised. Additionally, transparent communication with the public about the drone's use and purpose can help to mitigate privacy concerns.

Ethical Issue: Since the drone collects and transmits sensitive data, including video feeds and sensor data, there is the potential risk of data breaches or tampering.

- IEEE/ACM Code of Ethics: Section 2.9 of the ACM Code of Ethics [3] stresses the need to design systems that protect the privacy and security of data. The IEEE Code of Ethics also advocates for ensuring data security and avoiding harm (section 1) [4].
  Prevention Measures: To ensure data security and integrity, encryption protocols should be applied to the communication channels (e.g., TCP encryption methods). Additionally, multi-factor authentication (MFA) and strict access controls should be enforced, especially if cloud storage is used.

Ethical Issue: Drones can pose physical risks, such as collisions with buildings, people, or wildlife, or interference with other airborne vehicles. Ensuring the physical safety of the public is a critical consideration in both the design and operation of the drone.

- IEEE/ACM Code of Ethics: The IEEE Code of Ethics emphasizes prioritizing public safety (section 1) [4]. Similarly, the ACM Code of Ethics (section 1.2) stresses the responsibility to avoid harm [3].
  Prevention Measures: The drone should be equipped with collision avoidance systems (ECS shutdown on command) and programmed to adhere to established FAA drone regulations, including maintaining a safe altitude and distance from populated areas [5]. Geofencing technology can also ensure the drone does not operate in restricted or hazardous areas. Regular maintenance checks and firmware updates should be part of the operational protocol to avoid malfunctions.

Ethical Issue: Deploying drones in public spaces without the public's consent or knowledge may undermine trust and raise concerns about government or institutional overreach. Citizens may feel uncomfortable or surveilled if drones are present in their neighborhoods or public spaces without clear communication.

- IEEE/ACM Code of Ethics: The ACM Code of Ethics (section 1.7) emphasizes the need to honor confidentiality and avoid misleading the public [3]. The IEEE Code of Ethics underscores the importance of transparency and honesty (section 5) [4].
  Prevention Measures: To address this, it is important to work with local law enforcement and municipal agencies to ensure transparency regarding how, when, and where the drones are deployed. Regular public consultations and open communication channels will help maintain trust. Additionally, signage or notifications should be placed in areas where the drone is actively surveying.

Ethical Issue: The project must adhere to relevant federal, state, and local regulations governing drone usage. This includes obtaining necessary certifications from the Federal Aviation Administration (FAA) and following safety regulations.

- IEEE/ACM Code of Ethics: Both the ACM and IEEE Codes of Ethics (ACM section 1.5 [3] and IEEE section 7 [4]) stress the importance of abiding by applicable laws and regulations.
  Prevention Measures: In Champaign, Illinois, the drone system must comply with FAA Part 107 regulations governing the operation of unmanned aerial vehicles (UAVs) [5]. This includes rules about keeping the drone within visual line of sight, avoiding flying over people without waivers, and operating during daylight hours. Additionally, all operators should receive proper drone pilot certification. The system should include built-in compliance features, such as geo-fencing to prevent the drone from flying in restricted areas (e.g., near airports) [6].

Safety Concern: Mechanical or electronic failures in critical drone components, such as the propulsion system or servos, could lead to accidents, resulting in potential harm to people, property, or the drone itself.

- Prevention Measures: Regular maintenance and pre-flight inspections will be mandatory to ensure all components are functioning properly. The drone will undergo routine performance checks to detect wear and tear early. Redundant systems should be implemented for critical functions, and real-time diagnostics will monitor the drone's systems during operation. This aligns with the IEEE Code of Ethics, which emphasizes prioritizing public safety (section 1), and the ACM Code of Ethics (section 1.2), which stresses the responsibility to avoid harm.

Safety Concern: Drone operation in sensitive environmental areas, such as wildlife preserves or nature reserves, could disrupt ecosystems or harm wildlife.

- Prevention Measures: The drone will be programmed with geofencing technology to avoid sensitive environmental areas. Specific operational protocols will be developed in collaboration with environmental agencies to ensure minimal disturbance. Flight altitude and duration will be adjusted in these areas to reduce any environmental impact. This complies with the IEEE Code of Ethics (section 1) and ACM Code of Ethics (section 1.2), which focus on avoiding harm.

Safety Concern: Loss of communication between the drone and the backend system could result in erratic or unsafe behavior, potentially causing accidents or loss of the drone.

- Prevention Measures: The communication system will incorporate failsafe protocols, such as returning to a safe, predefined location (home point) in case of a signal loss. Additionally, signal encryption and interference mitigation strategies (e.g., frequency hopping) will be used to ensure stable and secure communication. Regular testing in various conditions will be performed to assess the reliability of the communication link. This aligns with the ACM Code of Ethics (section 2.9) and IEEE Code of Ethics (section 1), which advocate for data security and reliability.

Safety Concern: The drone's batteries pose a fire hazard, especially in cases of overheating or physical damage.

- Prevention Measures: The drone will use certified, high-quality batteries with integrated thermal management systems to prevent overheating. The battery compartment will be reinforced to minimize the risk of damage in the event of a collision. Additionally, battery levels will be monitored continuously during flight, and pre-flight safety checks will include ensuring batteries are not overcharged or showing signs of wear. This supports the IEEE Code of Ethics (section 1) regarding public safety and the ACM Code of Ethics (section 1.2) regarding harm prevention.

5 Works Cited

[1] Illinois Department of Public Health, "EMS Median Response Times," *Illinois Department of Public Health*, 2019. [Online]. Available: https://dph.illinois.gov/topics-services/emergency-preparedness-response/ems/prehospital-data-program/emsresponsetimes.html. [Accessed: Sept. 12, 2024].

[2] H. K. Mell et. al., "Emergency Medical Services Response Times in Rural, Suburban, and Urban Areas," *National Library of Medicine*, Oct, 2017. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5831456/. [Accessed: Sept. 12, 2024].

[3] *ACM Code of Ethics and Professional Conduct*, Association for Computing Machinery, 2018. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: Sept. 17, 2024].

[4] *IEEE Code of Ethics*, Institute of Electrical and Electronics Engineers, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: Sept. 17, 2024].

[5] *Small Unmanned Aircraft Systems (Part 107)*, Federal Aviation Administration, 2016. [Online]. Available: https://www.faa.gov/uas/commercial_operators/part_107. [Accessed: Sept. 17, 2024].

[6] *Campus Drone Policy*, University of Illinois Urbana-Champaign. [Online]. Available: https://cam.illinois.edu/policies/drone-policy/. [Accessed: Sept. 17, 2024].

[7] "Drone as First Responder (DFR) | City of Fremont, CA Official Website," *Fremont.gov*, 2023. https://www.fremont.gov/government/citywide-initiatives/public-safety-initiatives/drone-as-first-responders-dfr. [Accessed: Sept. 17, 2024].

[8] "Drone as First Responder (DFR) - Skydio Public Safety Solutions," *Skydio.com*, 2024. https://www.skydio.com/solutions/public-safety/. [Accessed: Sept. 17, 2024].