Target Tracking using Drone Swarms

Project Members & Speaking Order

1. Finlay Cross – Drone Swarm Lifecycle & Software Porting

2. Benjamin Ireland – Drone Trajectory Control & Network Protocol Integration

3. Se Hyun Kim – Target Implementation

4. Ronniel Padua – Drone Management/Visualisation Application

Raith Fullam - Network Communication Protocol

(SENG, Assessed Separately)

Project Code: **E24**

Sponsor: Wireless Research Centre

Background

Why?

NZ has many threatened insects that are almost impossible to track through conventional means. Because of this little is known about their behaviour.

How?

By attaching harmonic radar tags to the insects, their movements can be tracked. Using drone swarms, detailed information about the insects can be obtained.

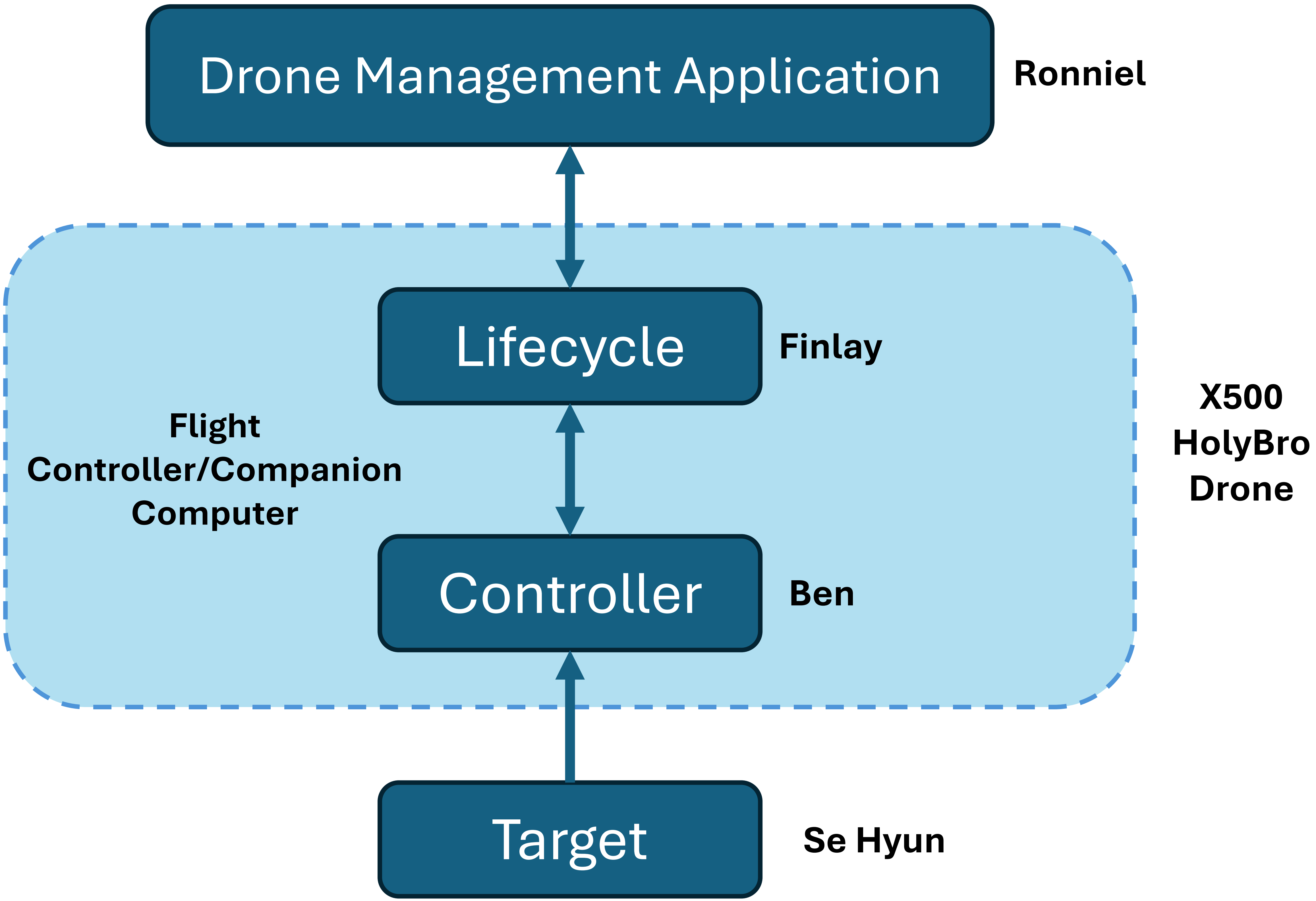
Other Applications:

Search & Rescue, Infrastructure Monitoring, Surveillance

Project Overview

- WRC’s Application:
  - Reliable tracking of threatened insects.
- Continued Project
- Subprojects:
  - Flight Controller
    - Trajectory Control System
    - Drone Swarm Lifecycle
  - Target Implementation
  - Drone Management Application
  - *Drone Coordination Protocol ~ SENG*

System Hierarchy



Sub-Project Goals

Successful Four Drone Swarm in the Field

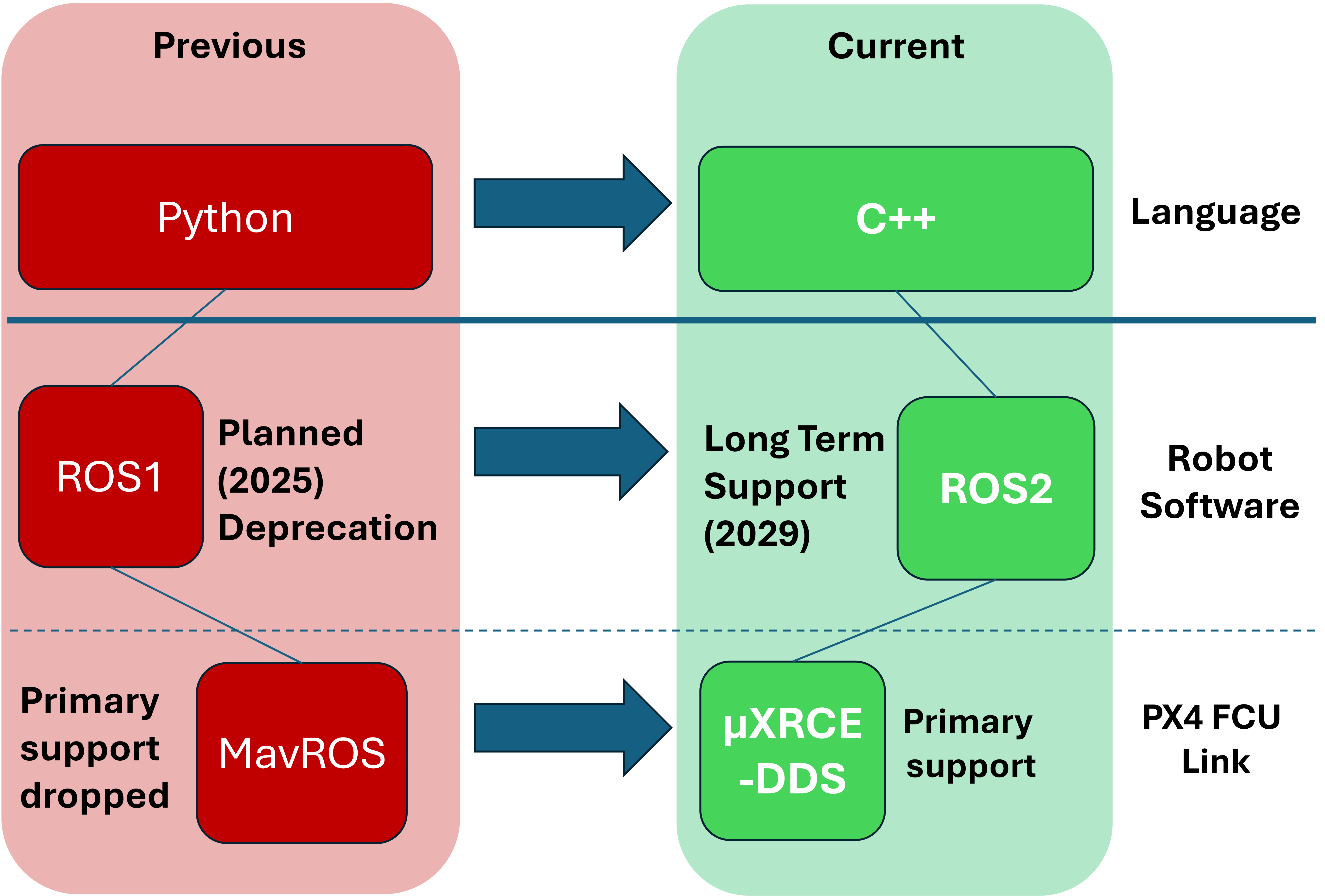
Port Software
Address Performance Concerns of last year

Lifecycle Development
Create a comprehensive mission lifecycle manager

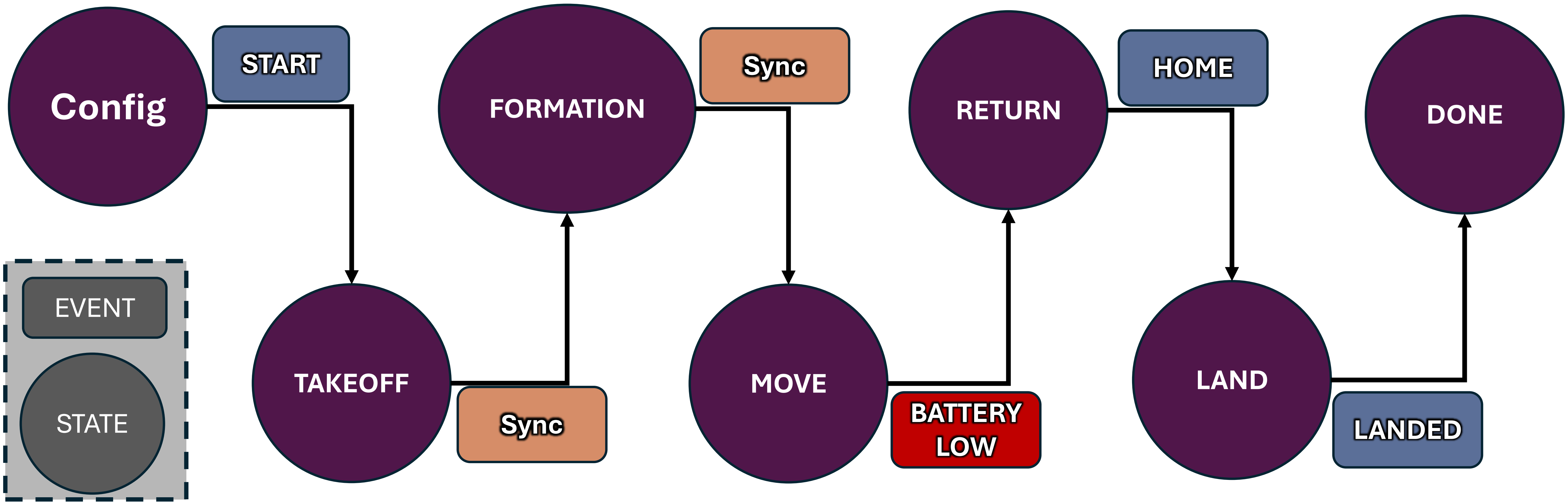
Scalability
Ensure the swarm is scalable to larger swarms

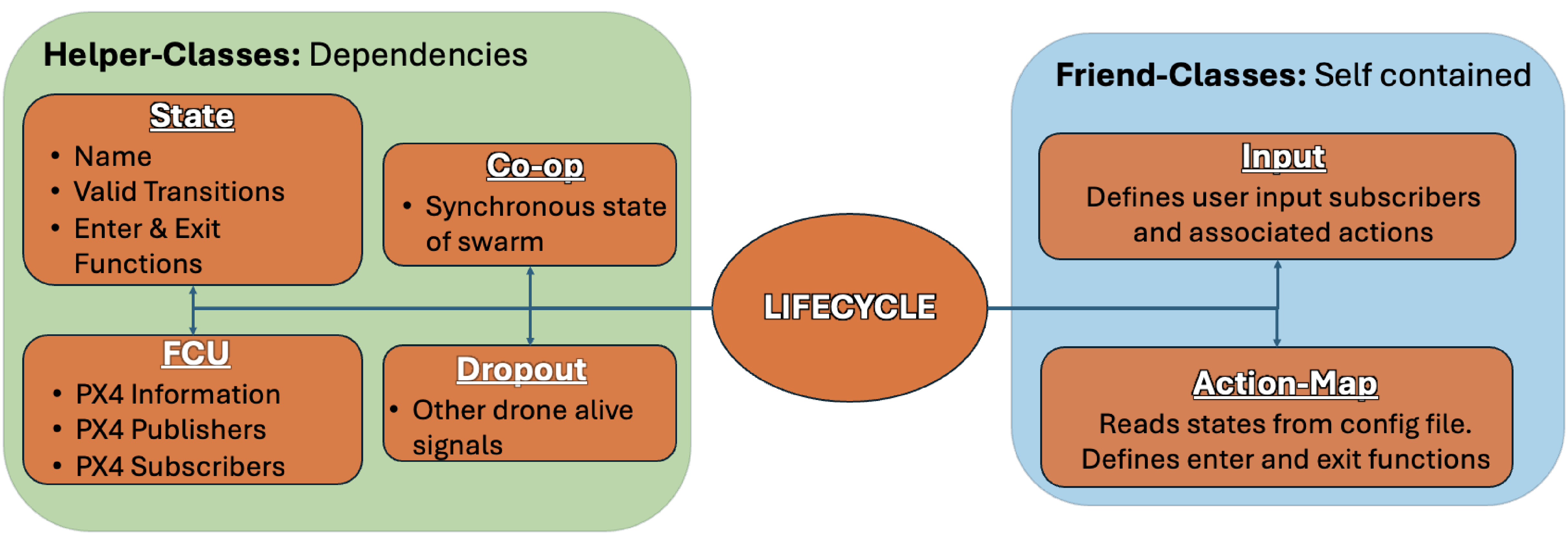
Long Term support
Ensure the project is viable far in the future

Software Redesign

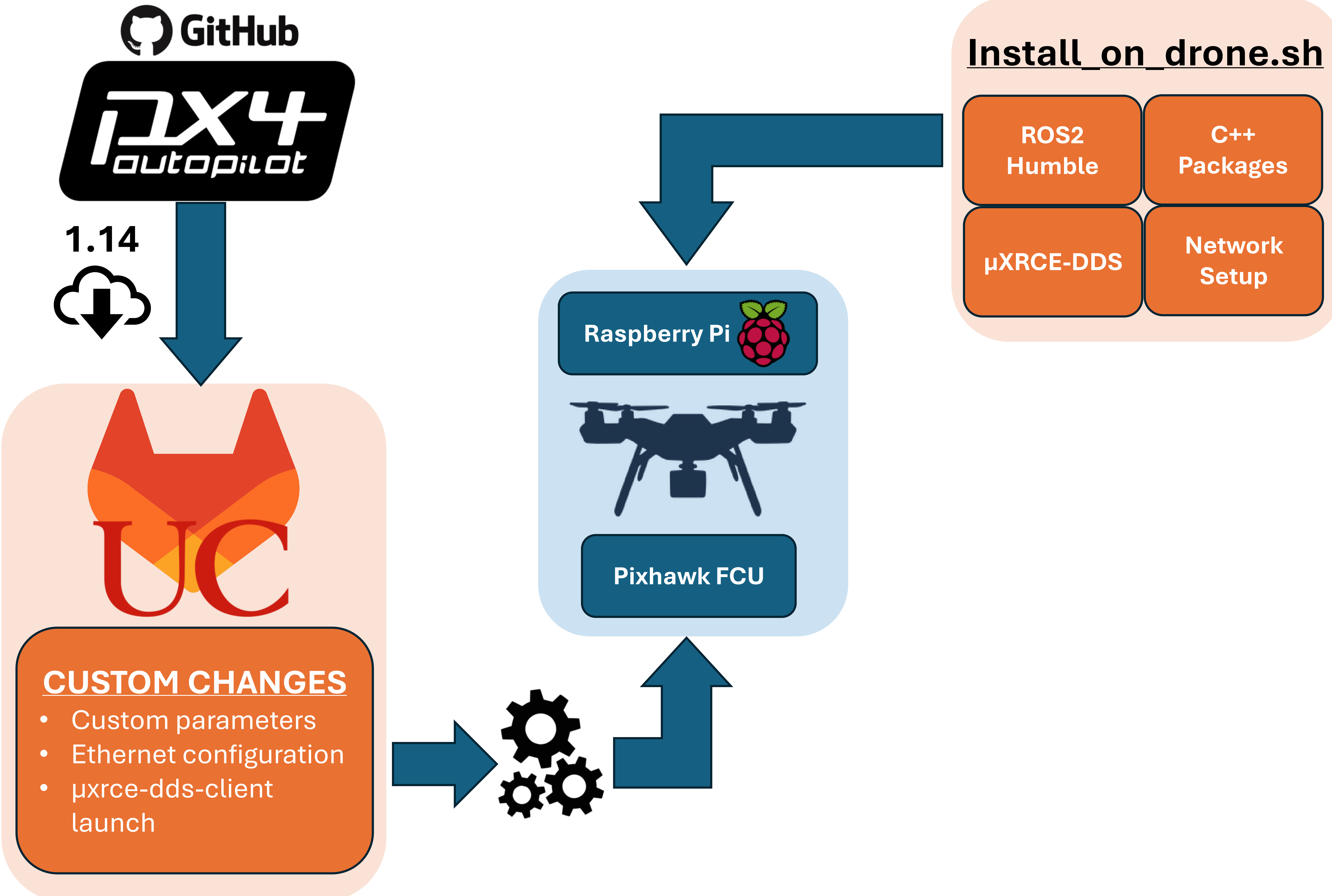


Lifecycle Implementation





Automatic Drone Setup



Project Review & Future

Achievements:

- Updated to long term support software and firmware
- Software Redesigned in C++
- Createm api style Input
- Automated drone setup
- Lifecycle restructured to be event based

Final Steps:

- Successful multi-drone field test
- In depth Documentation
- Code clean up

Recommendations:

- More detailed lifecycle states and events

Trajectory Controller & Protocol Integration

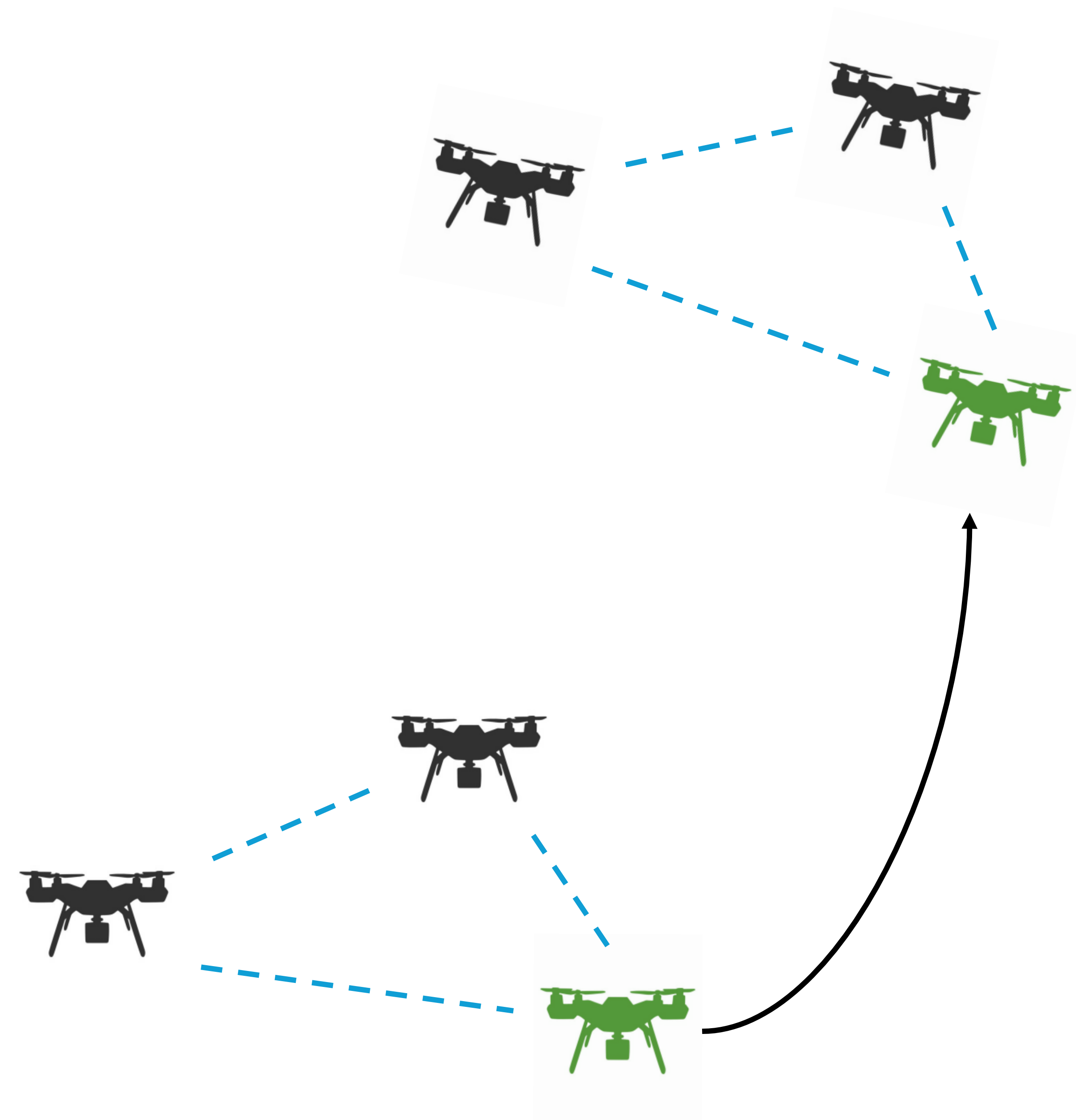
Sub-project Goals

- Provide reliable target tracking capability.
- Address performance and scalability issues of the previous iteration.
- Decoupling of formation and trajectory.

Sub-project Objectives

- Design of the drone's trajectory controller and swarm configuration.
- Integration with the drone coordination protocol.

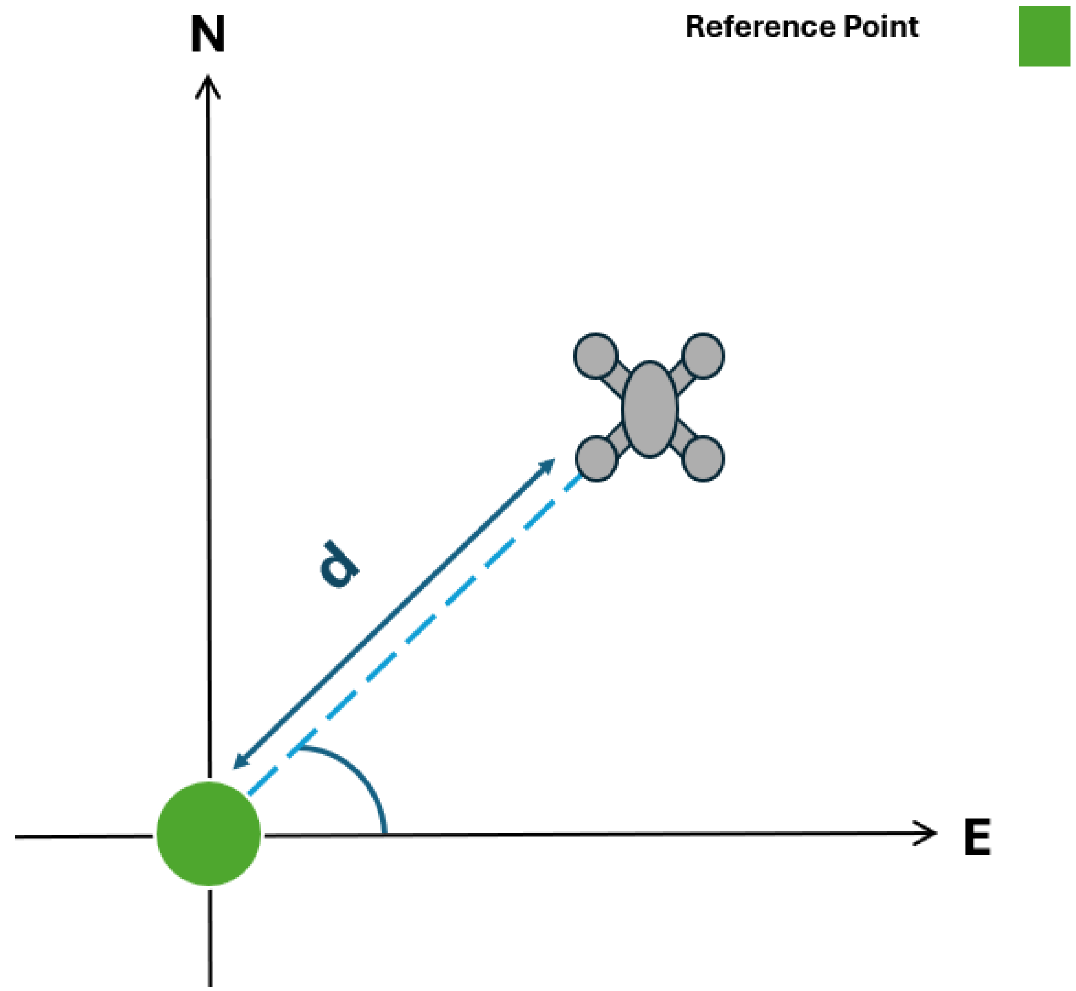
Previous Swarm Implementation

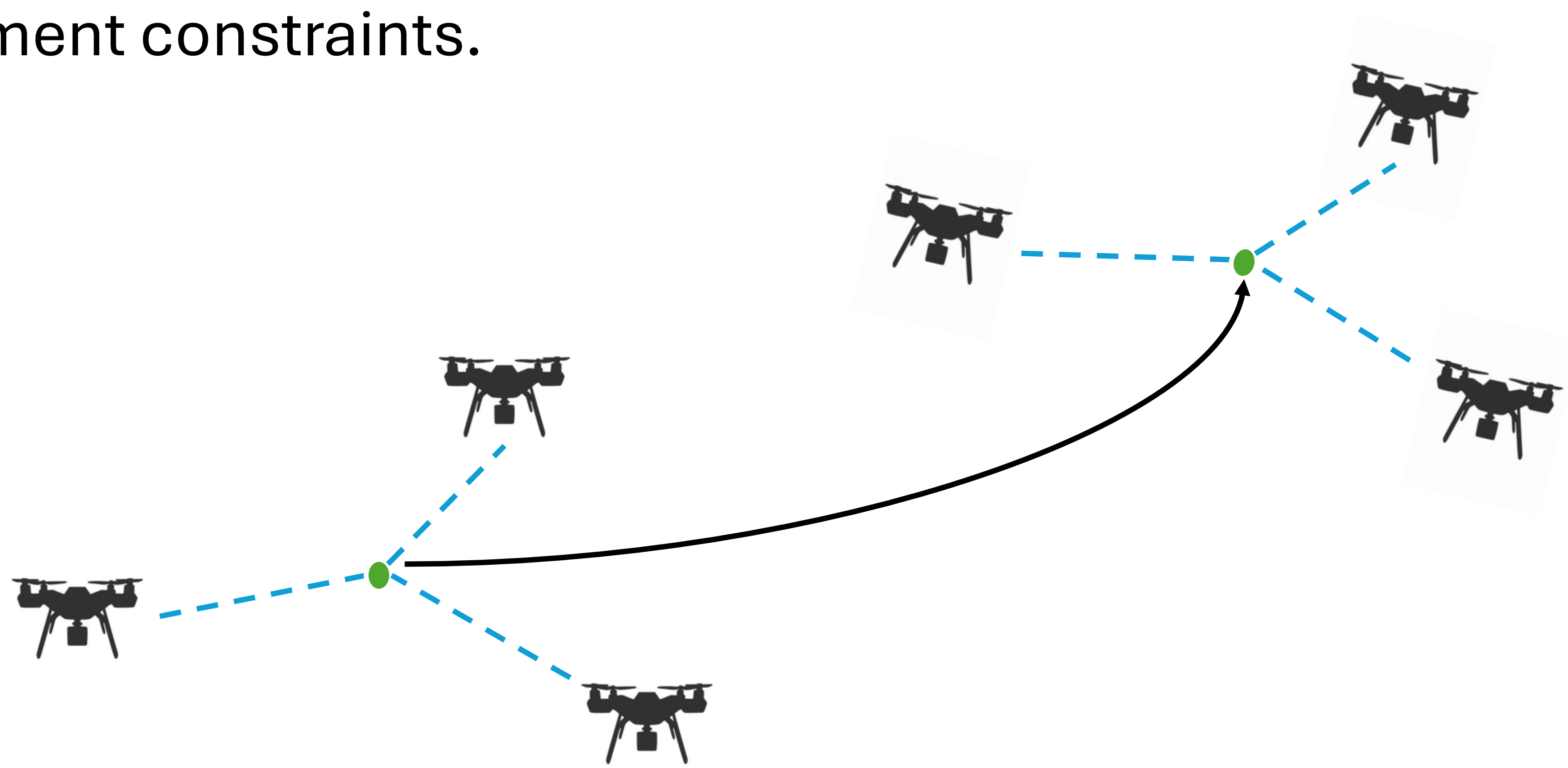


Swarm Design

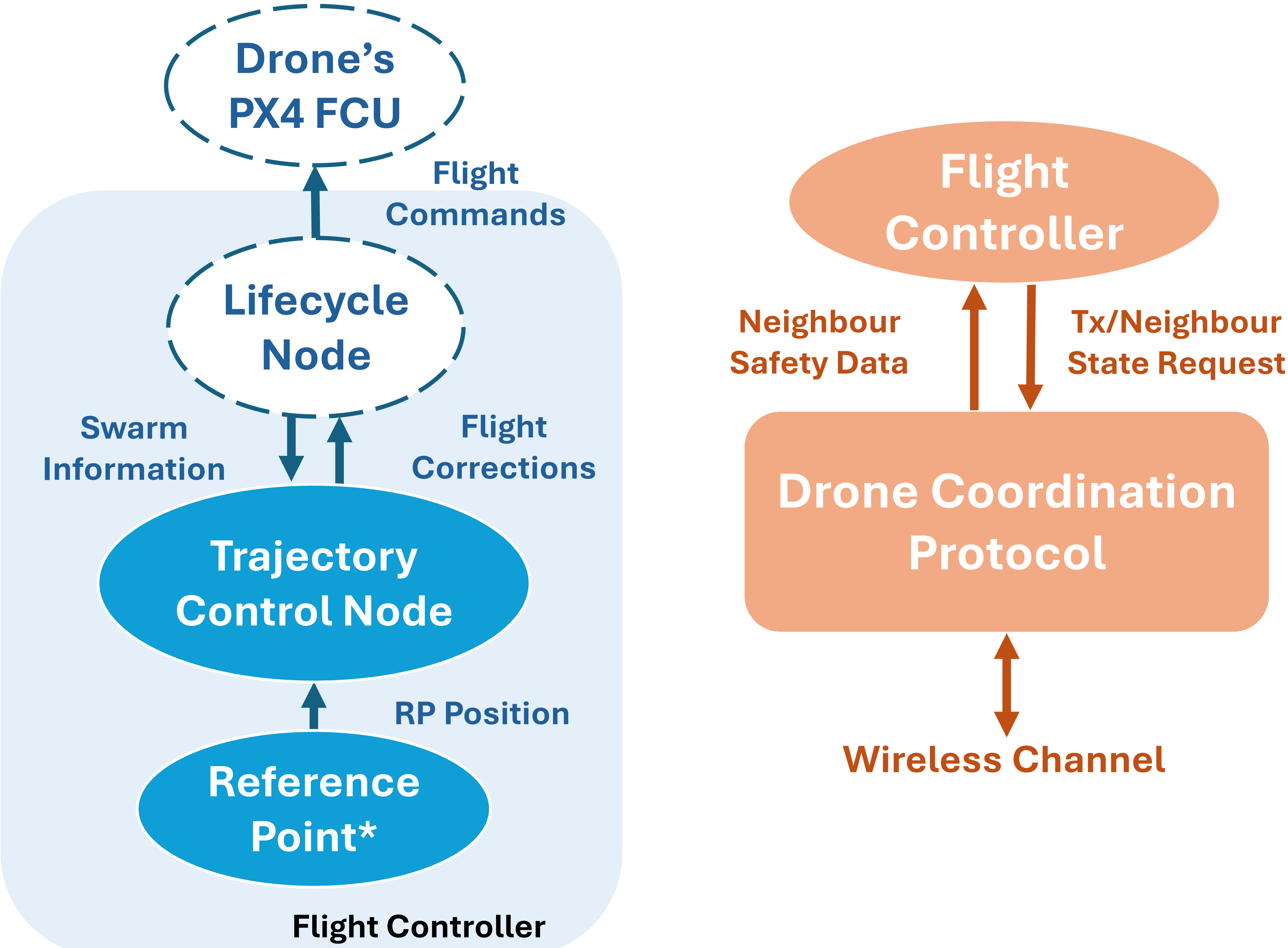
Proposed Solution

- Maintain position to Reference Point.
- RP superimposes itself over target.
- Simplifies individual drone computation and configuration.
- Position, angle, and height displacement constraints.





Software Architecture & Network Integration



Integration & Testing

Simulation

- The drone swarm has been tested in the *gazebo* simulation software.

Hardware Integration

- Software updates & drone reconfiguration.
- Middleware integration and FCU software customization.

Field Testing & Validation

- Field testing to date, test plans and evaluation.

Project Conclusion

Achievements

- Fully operational simulation of the drone swarm.
- Simplified swarm configuration to improve scalability.
- Separation of trajectory and formation activities.
- Initial field tests validated formation forming and singular drone target tracking.

Wrapping up

Validation of target tracking and multi-drone formations.

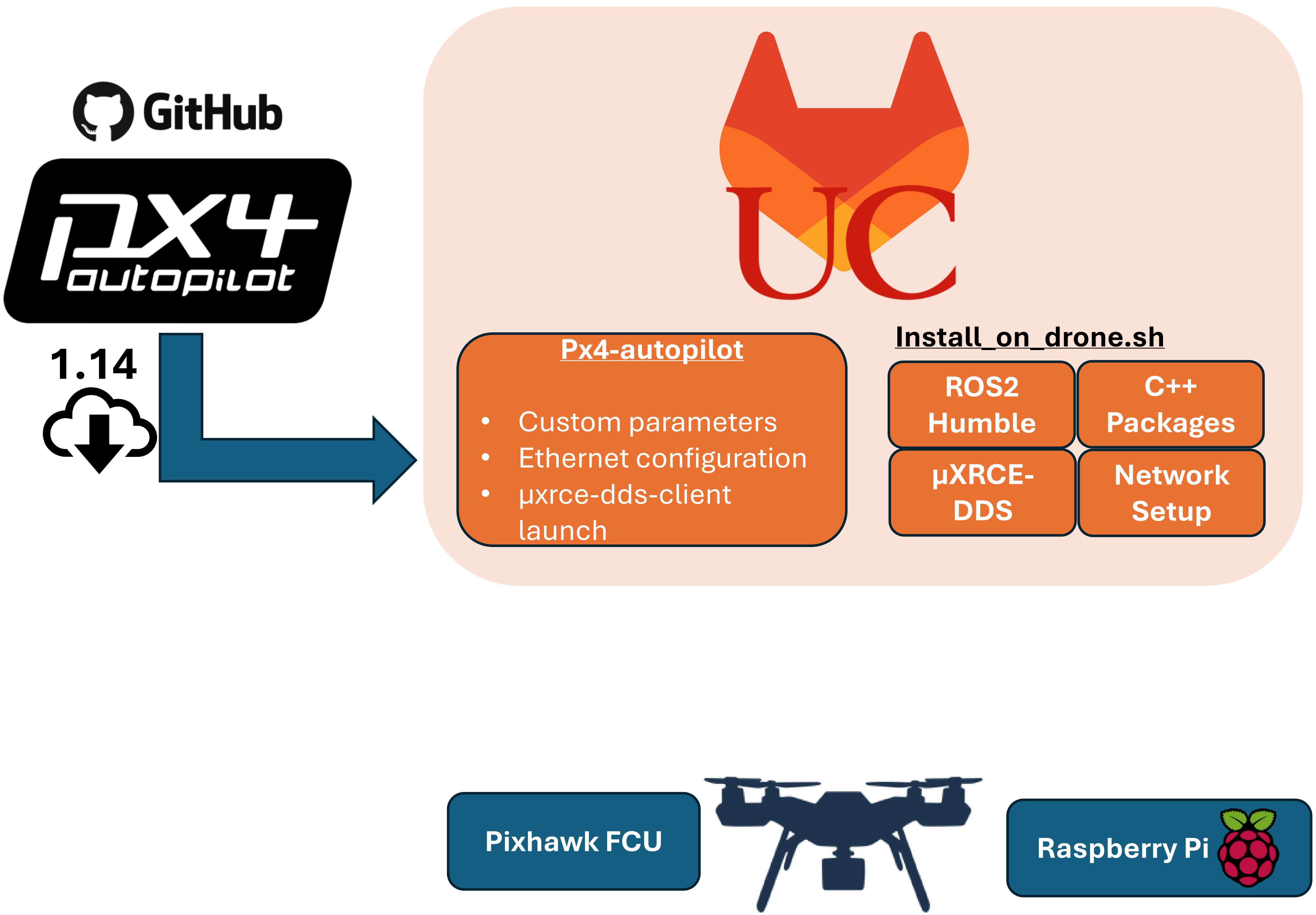
Recommendations

The current implementation is promising and should be enhanced in the project's continuation.

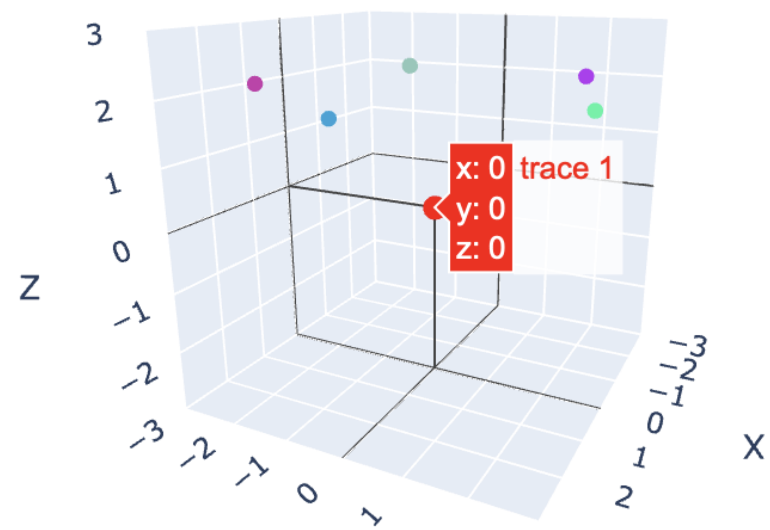
Future Objectives

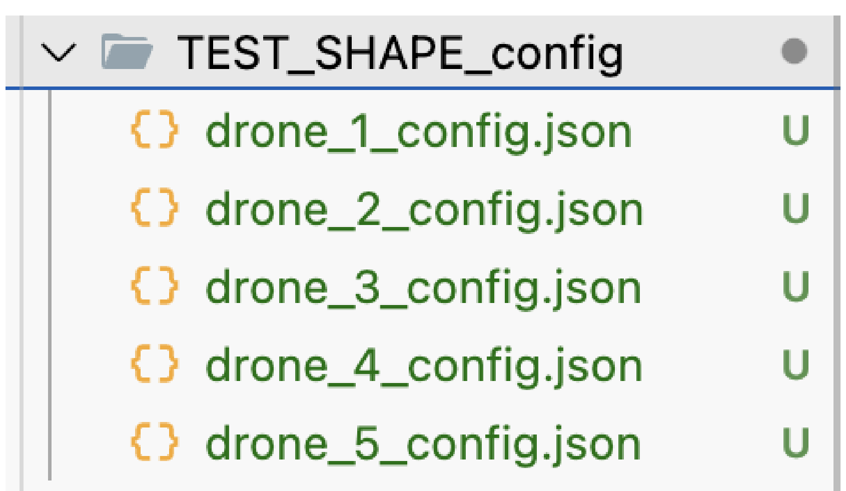
- Complete Network Protocol Integration
- Simulation -> Real World Performance Estimation
- Collision detection

Automatic Drone Setup



Configuration App





Dash App

Proof of concept to
Quickly create new formation

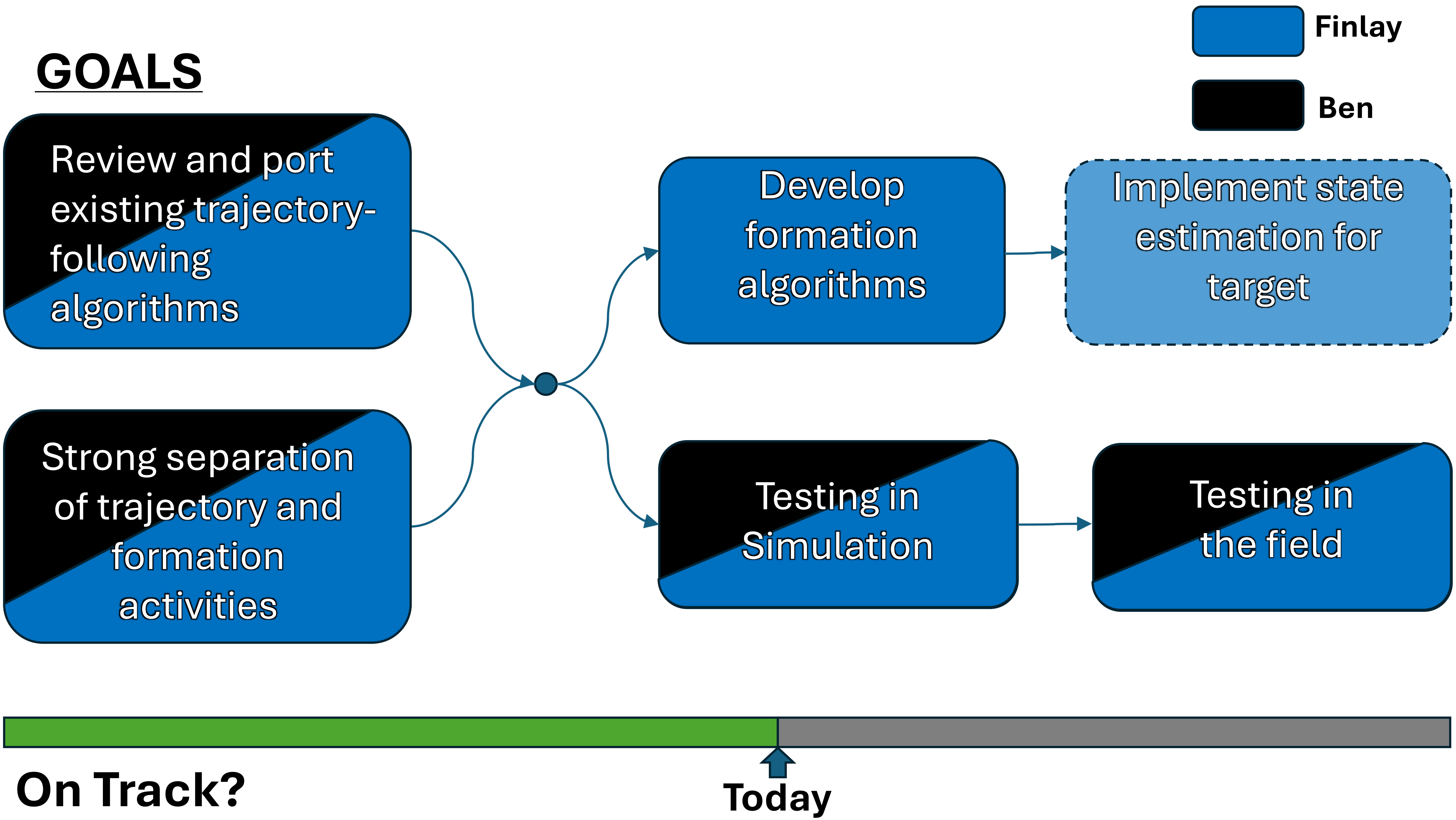
Reference Point
- Origin {0,0,0}

Add Node
- x, y, z control

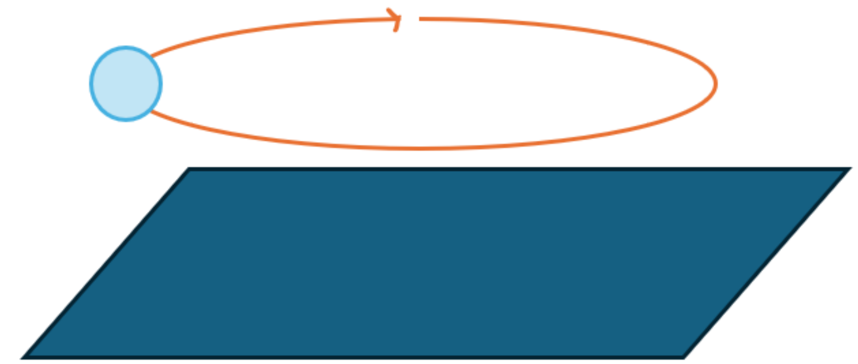
Create polygon
- Number of points
- Radius
- Height

Save
- Save Name

Progress & Goals



Scope Overview



Previous Target Node

Used virtual preplanned path node as target

New Target Node

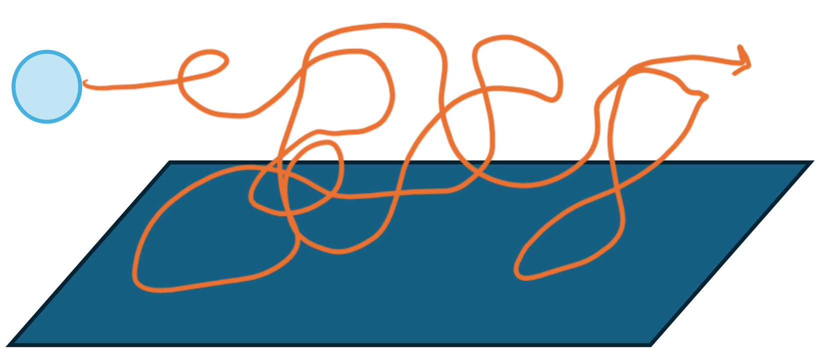
Develop a Real-Time tracking target

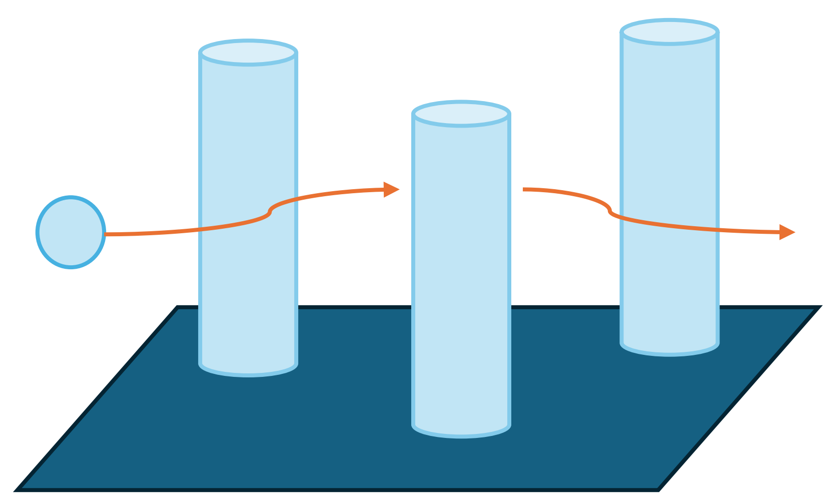
Advantages of Real-Time Tracking Target

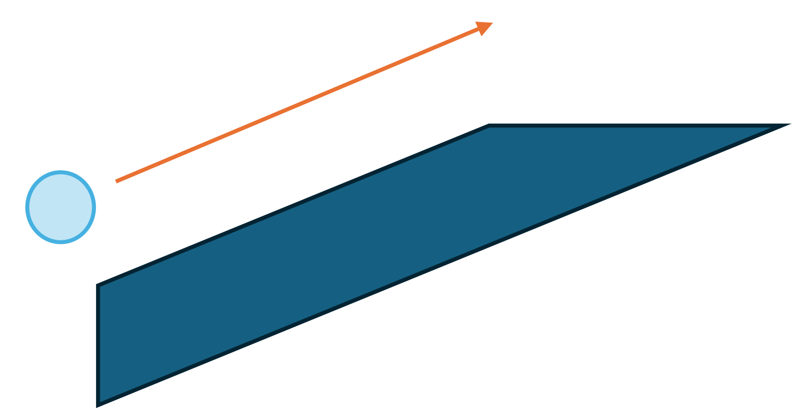
Increased flexibility in movement and timing

Increased scalability to different environments

Mimic real-life missions, more accurate testing







Target Requirements

- Need a precise tracking method
- Need a tracking method suited to the testing environment

	GPS	Wi-Fi RTLS	Active RFID
Accuracy	2.5 m	5 m	3 m
Range (Maximum)	Global	15 m	80 m
Power	High	High	Moderate
Cost (NZD)	Free (Provided by WRC)	Free (Wi-Fi capability installed inside drone)	40 + (260 x N), N = number of drones. (Tag [4] + Reader [5] x N)
Ref. pt. Compatabible	Yes	No	No
Choice:	**GPS**		

BU-353S4 USB GPS Receiver

- Operating Frequency of 1.575 MHz
- Channels = 48

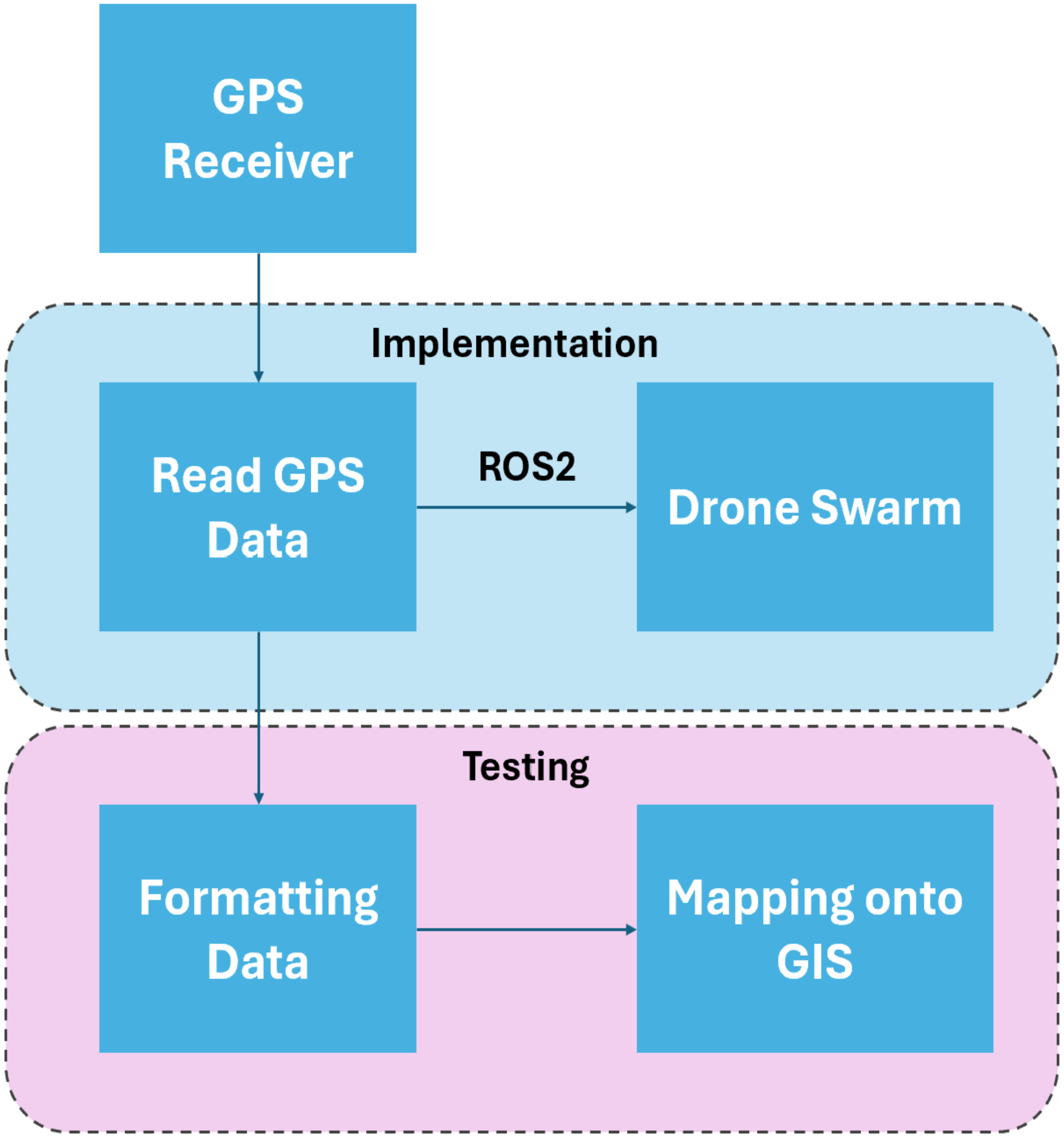
GPS Implementation

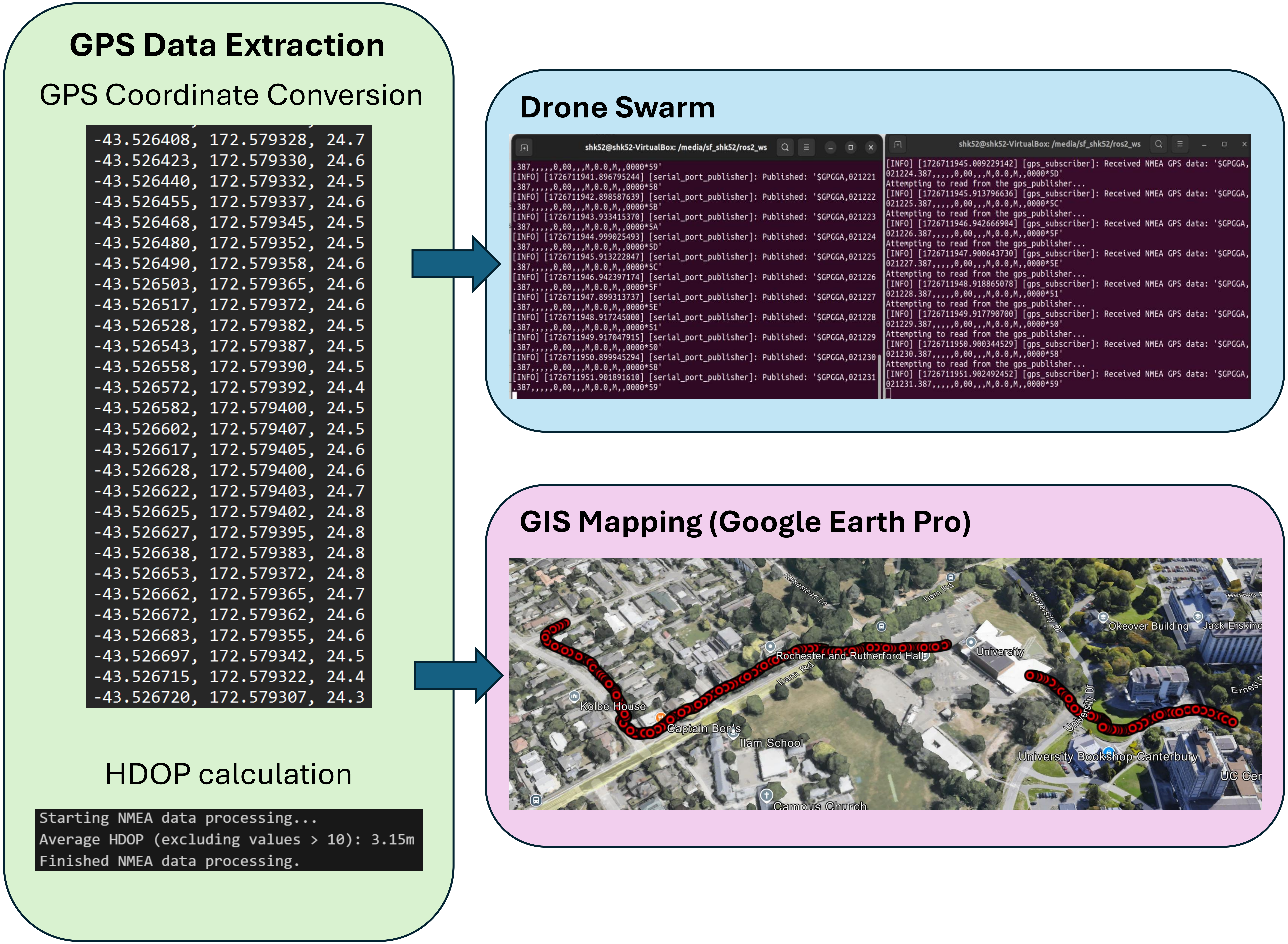
Implementation

- Read GPS Data from Receiver (10 times per second)
- Convert NMEA GPS data to Lat, Lon, Alt in degrees
- Send GPS Data to Drone Swarm using ROS2

Testing

- Read HDOP Data from Receiver (1 time per second)
- Log GPS Data into a text file
- Map out GPS Data using GIS Software





GPS Data Extraction
GPS Coordinate Conversion
-43.526408, 172.579328, 24.7
-43.526423, 172.579330, 24.6
-43.526440, 172.579332, 24.5
-43.526455, 172.579337, 24.6
-43.526468, 172.579345, 24.5
-43.526480, 172.579352, 24.5
-43.526490, 172.579358, 24.6
-43.526503, 172.579365, 24.6
-43.526517, 172.579372, 24.6
-43.526528, 172.579382, 24.5
-43.526543, 172.579387, 24.5
-43.526558, 172.579390, 24.5
-43.526572, 172.579392, 24.4
-43.526582, 172.579400, 24.5
-43.526602, 172.579407, 24.5
-43.526617, 172.579405, 24.6
-43.526628, 172.579400, 24.6
-43.526622, 172.579403, 24.7
-43.526625, 172.579402, 24.8
-43.526627, 172.579395, 24.8
-43.526638, 172.579383, 24.8
-43.526653, 172.579372, 24.8
-43.526662, 172.579365, 24.7
-43.526672, 172.579362, 24.6
-43.526683, 172.579355, 24.6
-43.526697, 172.579342, 24.5
-43.526715, 172.579322, 24.4
-43.526720, 172.579307, 24.3
HDOP calculation
Starting NMEA data processing...
Average HDOP (excluding values > 10): 3.15m
Finished NMEA data processing.
Drone Swarm
shk52@shk52-VirtualBox: /media/sf_shk52/ros2_ws
GIS Mapping (Google Earth Pro)
Rochester and Rutherford Hall
University
Okeover Building
Jack Erskine
Kolbe House
Captain Ben's
Ilam School
Ilam Rd
University Dr
University Bookshop Canterbury
Campus Church

Conclusion

Achievements:

Found suitable target method ☑

Capable of sending and receiving correct GPS data ☑

Various features to help test/debug the target ☑

Virtual Static Target → Real-Time Target ☑

Summary:

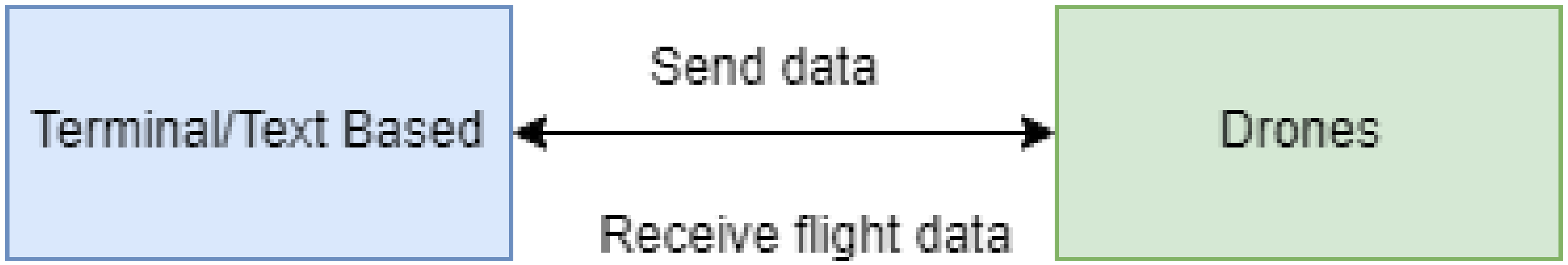
Future continuations of this project now have a fully independent target node which will be used to help test drone swarm behaviour

Recommendations:

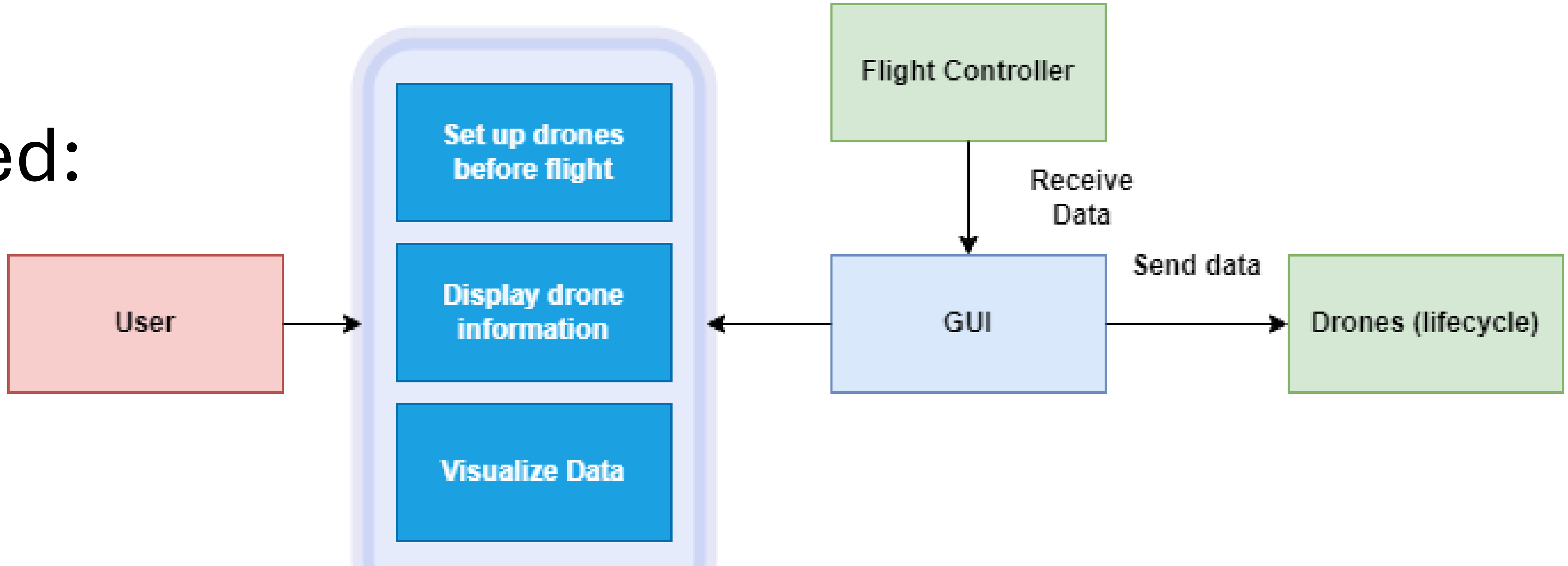
If further testing deems a more accurate GPS is required, the current GPS can be upgraded to GNSS RTK receiver which may help with accuracy.

Purpose of the GUI

Original:



Improved:

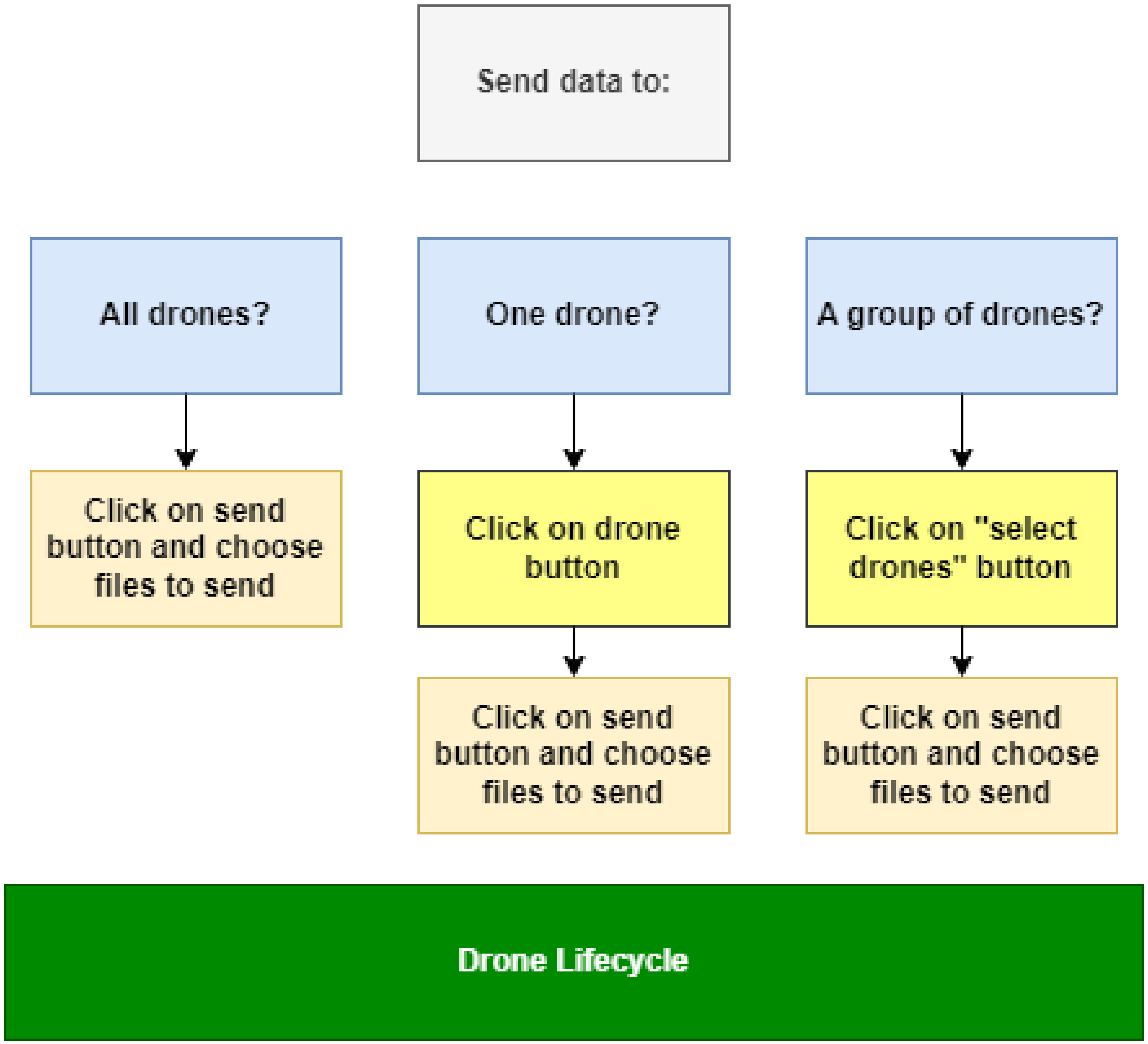


What Has Been Accomplished?

Display drone information	Store drone information	Set up drones before flight
See connection information	Accessible configuration file	Add new drones
		Set up SSH keys
		Enter a direct connection

GUI (Python-Tkinter)	Requirement	Send data to drones:	Increase code reliability
Necessary data:	ROS Configurations	Create ROS configurations	Add unit tests
	Python Webserver	Send files/folder	Add manual tests
Necesary Commands:	Activate drone files	Send commands	Add exception handling

Using the GUI

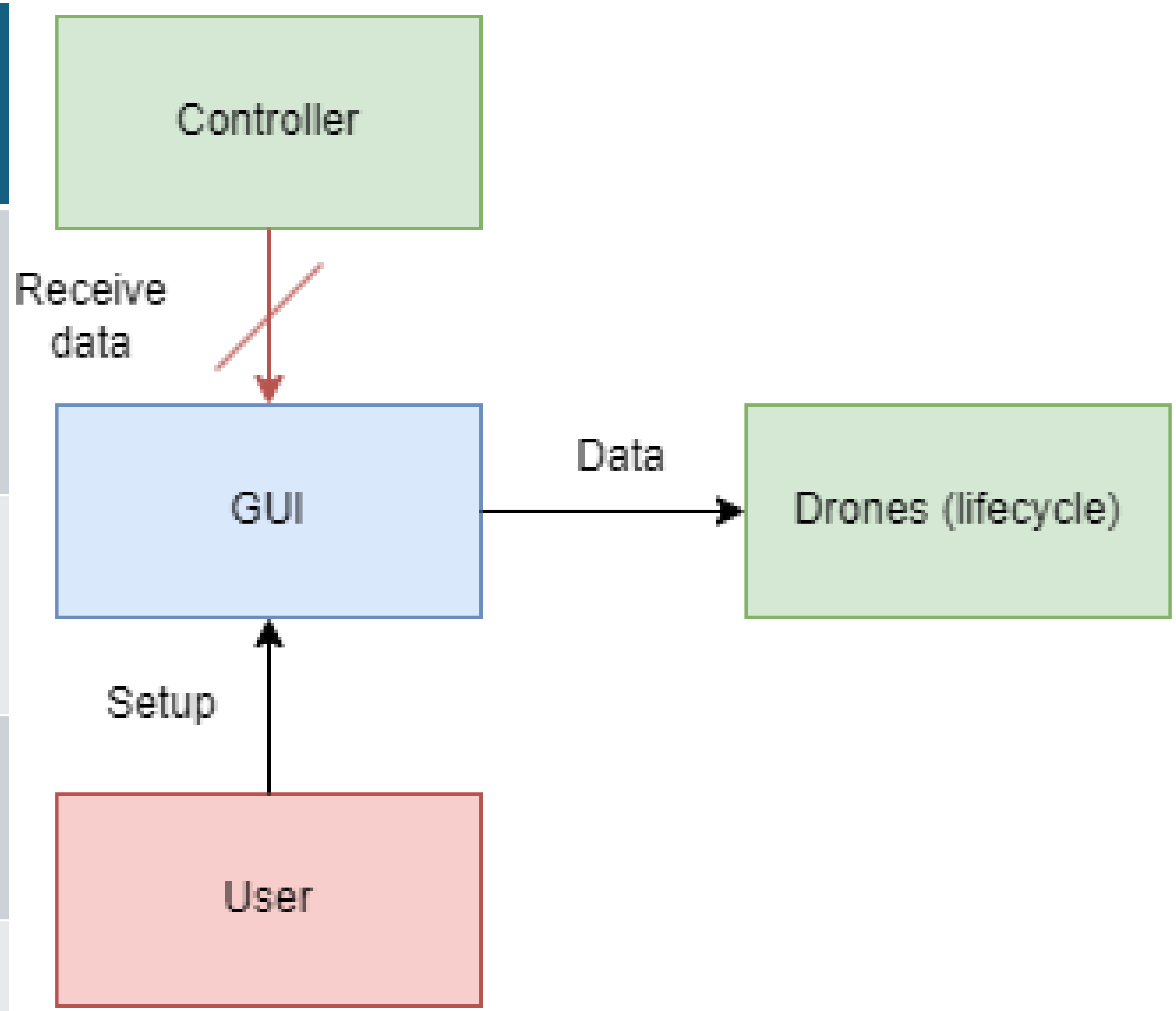


What Needs To Be Done?

Requirement:	Receive data from drone	What is data used for?
GUI	ROS Messages?	Visualize data from the drone
	Network protocol??	Update drone information
	Flask webserver?	

In Conclusion

What the GUI can do	Final steps
Send data to the drones	**Receive** data from the drones
Show various information about drone	**Update** drone information automatically
Set up new drones	**Recommendations to future groups**
Guarantee code works with tests	Use test-driven development approach to streamline development




Drone Configuration
Drone doublehelix Ip: 192.168.20.19 MAC: 10:10;10:10:10
Drone zygote Ip: 192.168.20.72 MAC: 00:c1:41:29:09:27
Drone drone-1 Ip: 192.168.10.1 MAC: n/a
Git Pull
Send a file to all drones
Send directory to all drones
Generate SSH Keys
Fly!
Create ROS zip folder
Add drones to file
Send files to multiple drones
Create drone configurations