

# Reflection Document - Drone Conflict Detection & Visualization

## 1. Project Overview:

This project was developed as part of the FlytBase internship assignment. It detects spatial and temporal conflicts between a primary drone's mission and other drones in the airspace. The system processes data, calculates positions over time, checks for conflicts, and generates an animated visualization.

## 2. Design Choices:

- Used JSON for mission data for easy editing and readability.
- Created modular Python scripts: `io.py` for input handling, `trajectory.py` for position calculation, `detector.py` for conflict checks, and `visualizer.py` for animation.
- Chose a second-by-second simulation for simplicity and clarity in visual outputs.
- Visualization with Matplotlib to produce both live animations and GIF outputs.

## 3. Scalability:

For large-scale deployment (1000+ drones):

- Use spatial indexing (KD-Tree) to speed up distance checks.
- Employ parallel processing or distributed computing for trajectory simulations.
- Store and query trajectory data in a spatial database like PostGIS for efficiency.

## 4. Tools and Resources Used:

- Programming Language: Python 3.x
  - Libraries: NumPy, Matplotlib
  - Development Environment: VS Code
  - AI Assistance: VS Code Copilot, Replit AI for code suggestions and debugging.
- All AI-generated code was reviewed and tested manually before inclusion.

## 5. Lessons Learned:

- Improved understanding of spatio-temporal data processing.
- Learned effective ways to visualize movement and conflicts in 2D animations.
- Gained experience in structuring a small but complete software project with clear modular components.

## 6. Future Improvements:

- Add 3D visualization with altitude and time.
- Implement real-time streaming conflict detection.
- Build a web dashboard for uploading missions and viewing results online.