MICHIGAN STATE UNIVERSITY

Project Plan Presentation

Networked CV Algorithm Testing Harness

The Capstone Experience

Team Lockheed Martin Space

Eric Cui
Samuel Lain Hedden
Jonathan Lima de Paula
Tyler Przybylo
Shawn Russel
Natali Thiesse

Department of Computer Science and Engineering
Michigan State University

Spring 2026



From Students…
…to Professionals

Project Sponsor Overview

- Lockheed Martin’s space division, delivering spacecraft and mission systems
- Computer Vision team working with ML and CV algorithms
- Team is focused on Space Domain Awareness solutions

Project Functional Specifications

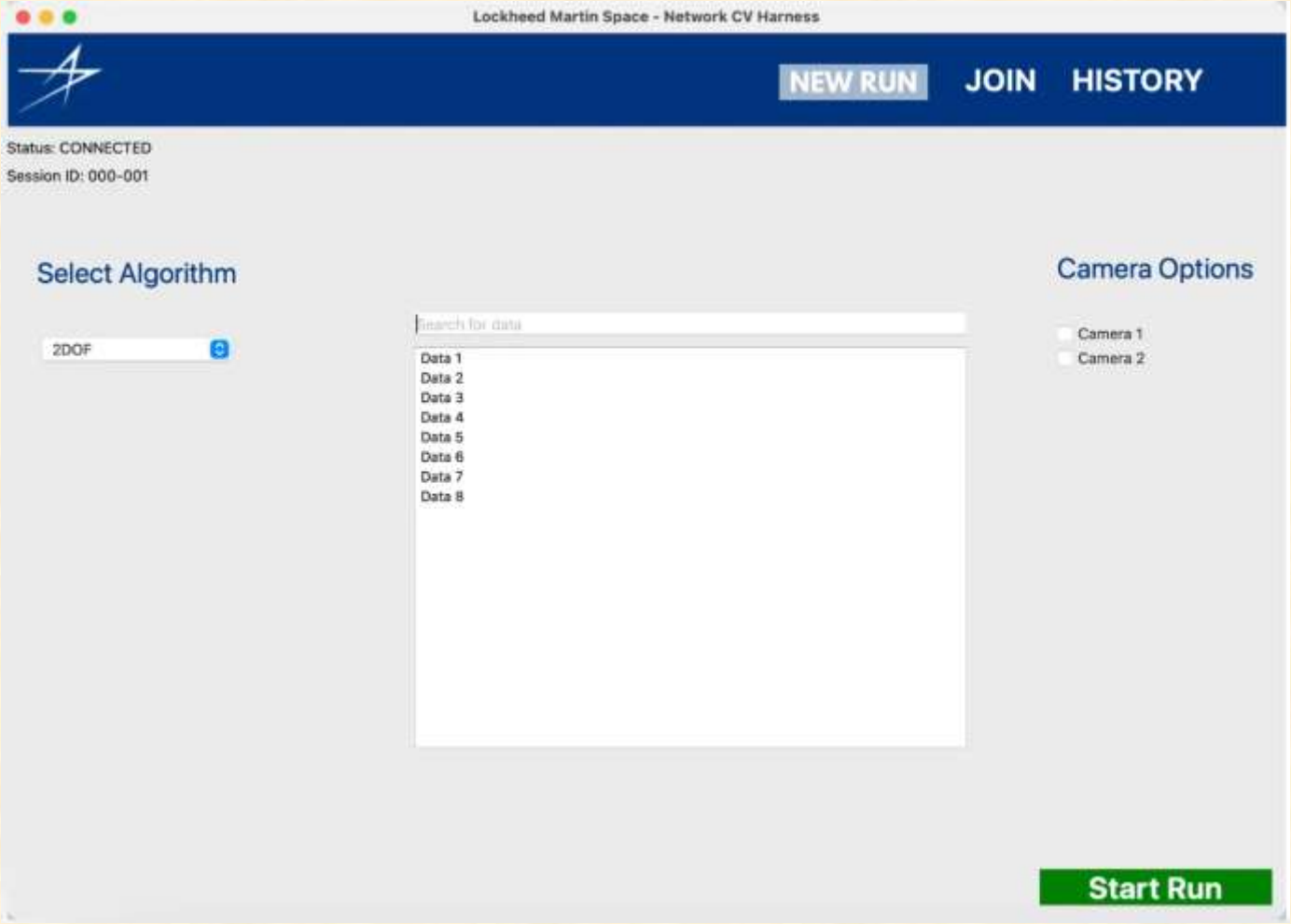
- CV testing is expensive to set up and hard to reproduce
- Unified harness to stream imagery, run algorithms, and visualize outputs
- Benchmarks accuracy against ground truth with error metrics/graphs
- Engineers can replay runs and compare performance over time

Project Design Specifications

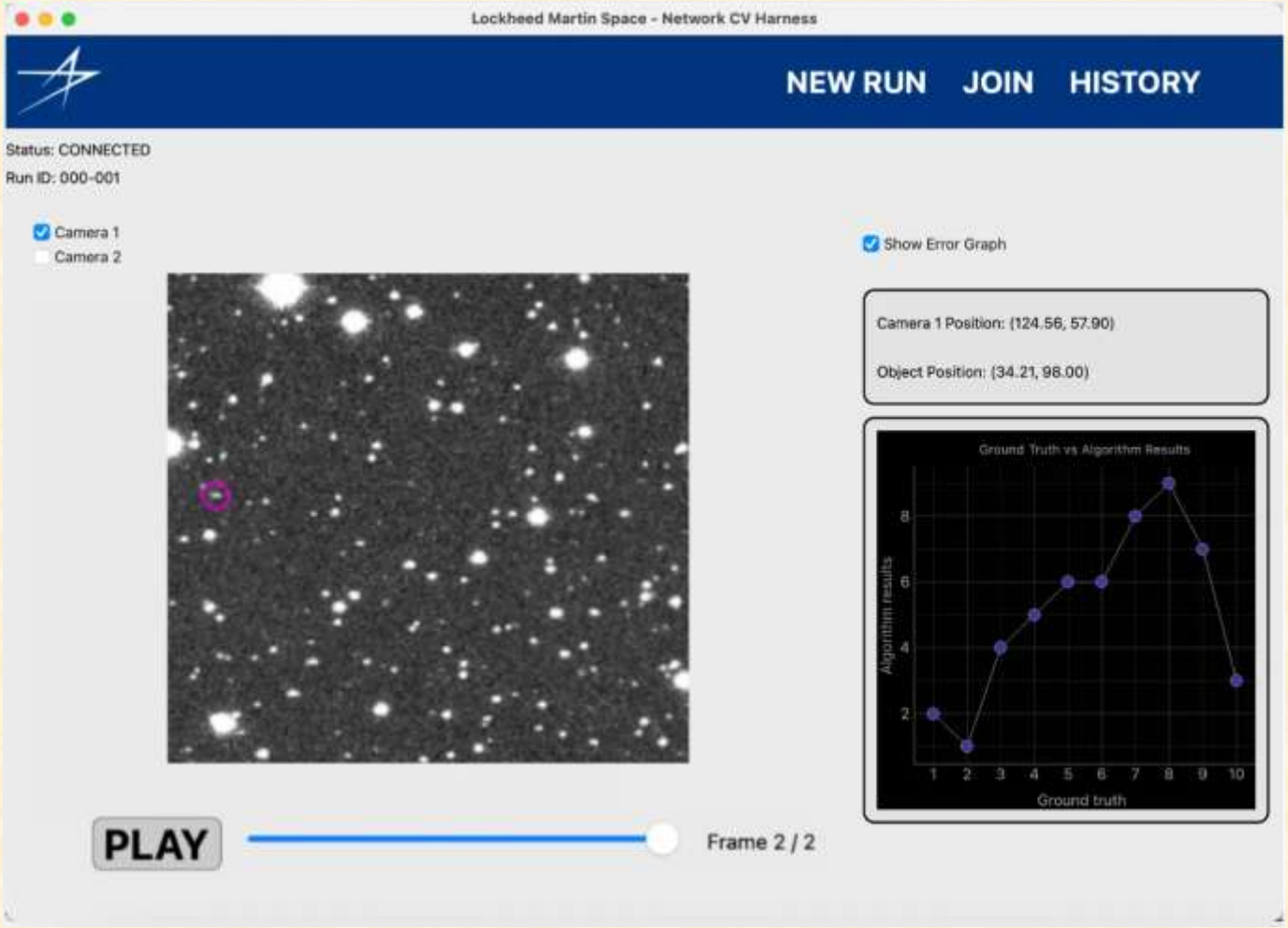
- Create a run by selecting a dataset, algorithm, and camera view
- During a run:
  - Livestream of results
  - Video playback features
  - Selectable error graph
  - Metadata available
- Join an active run to watch the same session from another client
- View previous runs



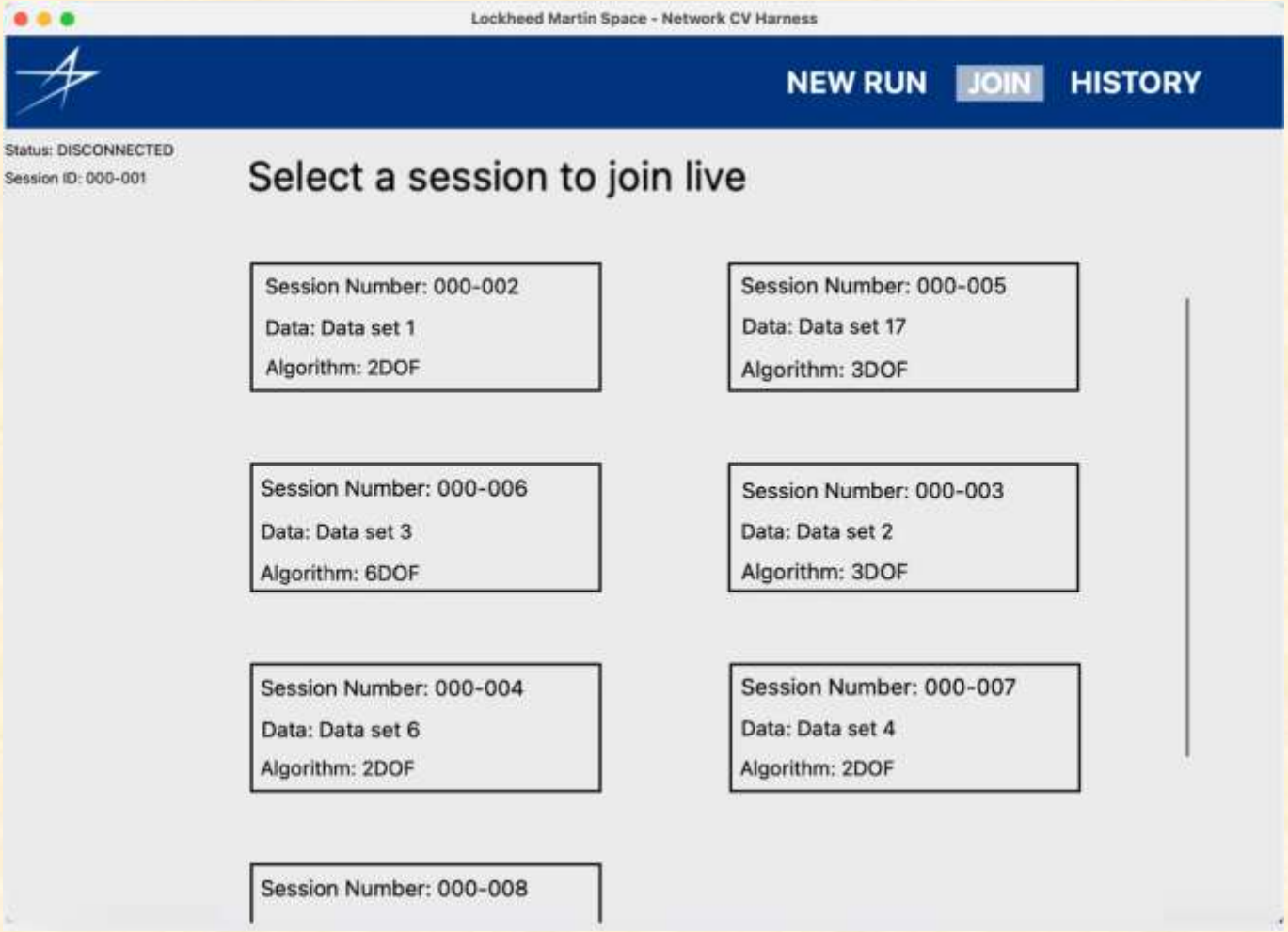
Screen Mockup: Creating a Run



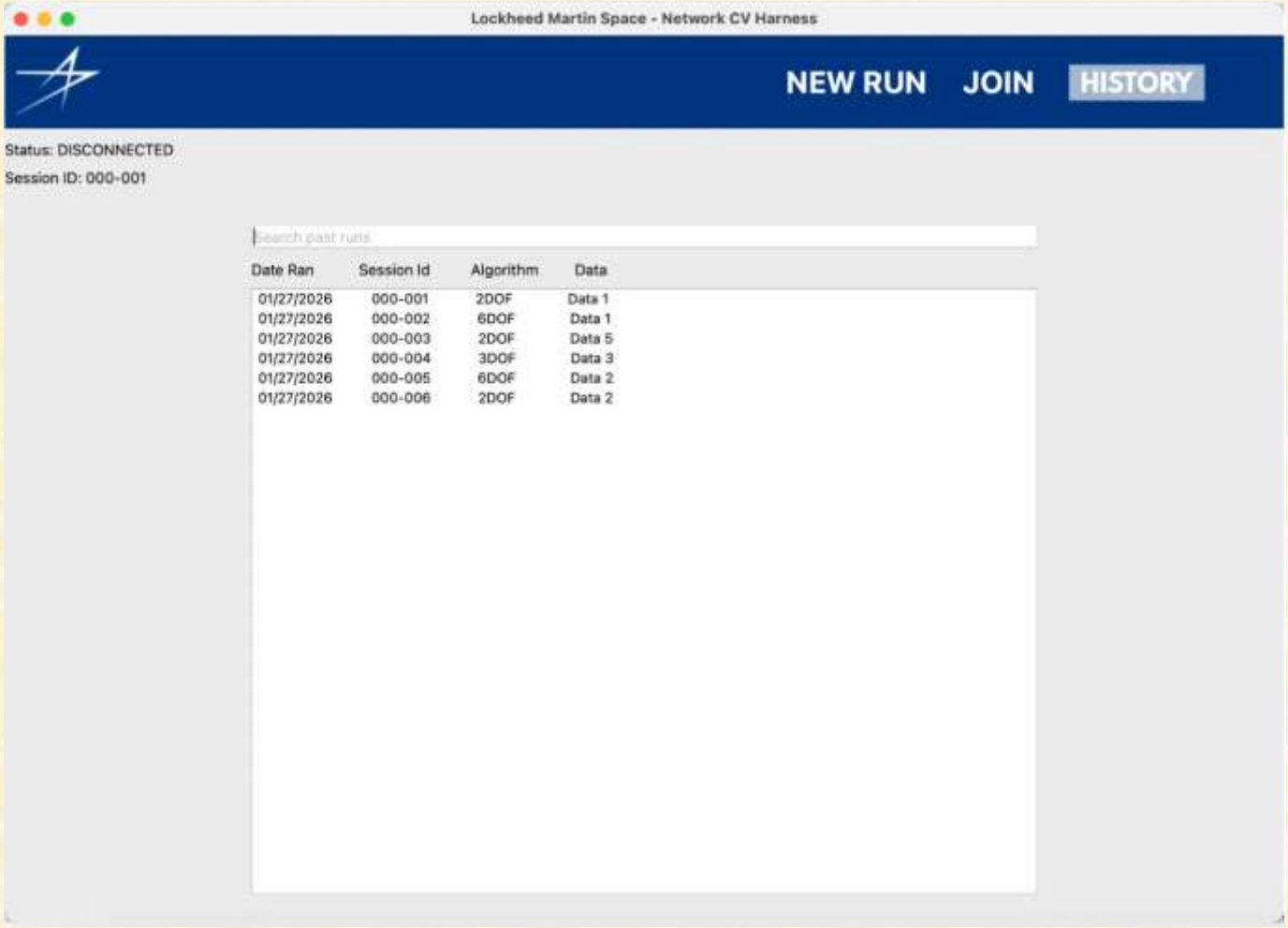
Screen Mockup: During a Run



Screen Mockup: Join a live session



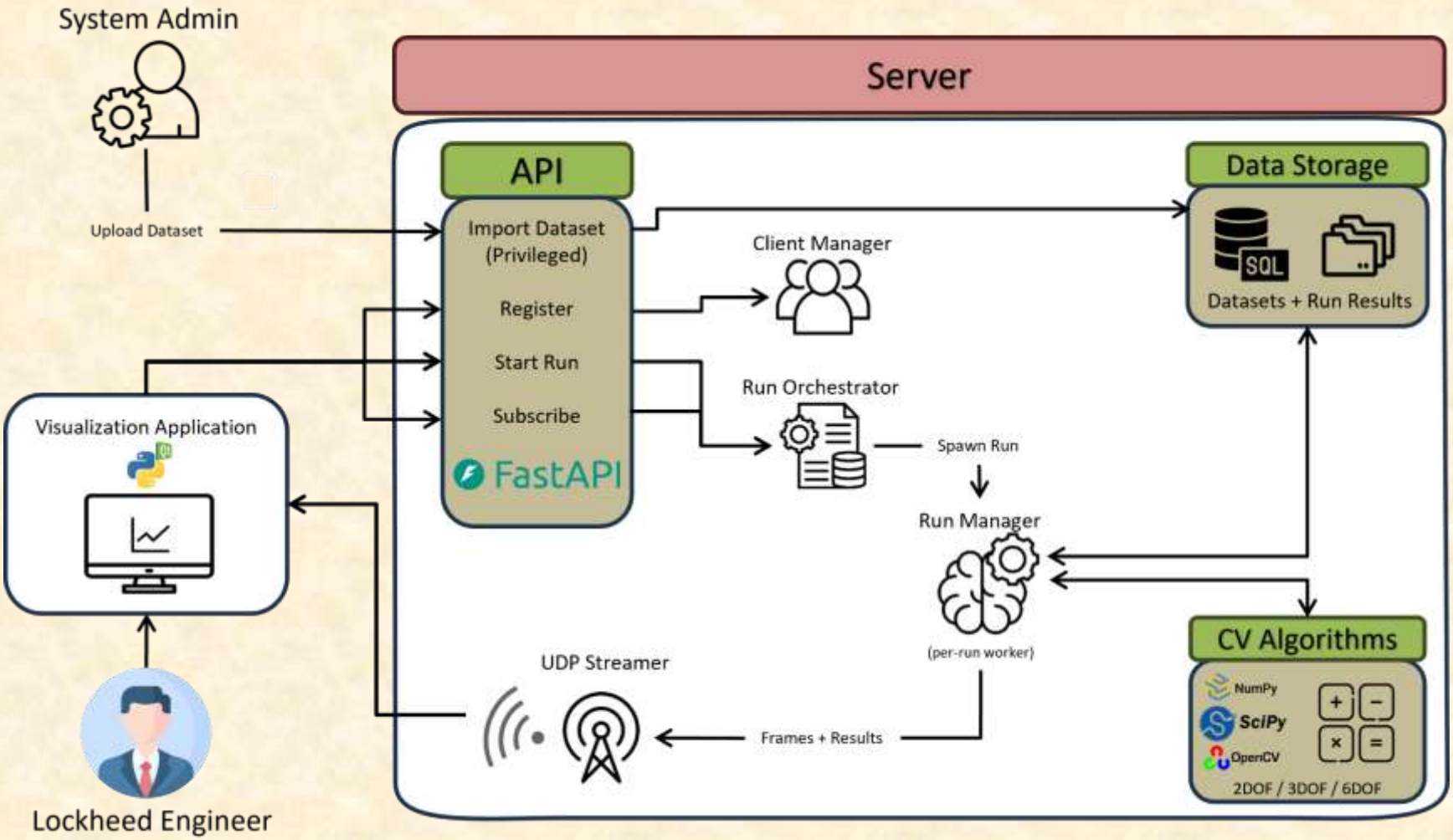
Screen Mockup: History Page



Project Technical Specifications

- REST API Server:
  - Built with FastAPI
  - Handles communication between Server and Client
- Algorithm Processing Layer:
  - Uses OpenCV, NumPy, and SciPy to run the selected algorithm (2DOF, 3DOF, or 6DOF) on image frames
  - It will compute the detected objects position, distance, and rotation
- Database:
  - Uses relational database to store frame data, ground truth data, algorithm results, and configuration information
  - Enables replaying processed data back to the client without needing to recompute the algorithm results
- Client Visualization:
  - Created with PyQT
  - Enables the user to view algorithm results in real-time through a video player
  - It is used to select the configuration (algorithm, cameras, dataset) and submit jobs.

Project System Architecture



Project System Components

- Hardware Platforms
  - Windows Computer
- Software Platforms / Technologies
  - REST API (FastAPI, Python)
  - Database (PostgreSQL)
  - Client (PyQT)
  - Containerization (Docker)
  - Algorithms (OpenCV, NumPy, SciPy)

Project Risks

- Risk 1: 6DOF / SLAM complexity
  - 6DOF integration can be very dependent on the use case. Not much documentation on the internet and the algorithm is very complex.
  - After doing initial research, build on our work on 2DOF/3DOF, work closely with the client, and test frequently
- Risk 2: UDP streaming to multiple clients
  - Coordinating multi-client streaming and session management
  - Use a publish/subscribe model: one run publishes, multiple clients subscribe, server streams to all subscribers
- Risk 3: UDP packet loss / out-of-order
  - Frames or results may drop or arrive out of order
  - Design our UDP packets with redundancy in mind, including the results of the algorithm in every UDP packet, and have client side retry if packet loss grows above a certain threshold
- Risk 4: Algorithm speed/latency
  - Algorithms may not keep up with real-time frame rates; blocking can stall the app
  - Introduce a small starter latency to allow for compensating for future lag

Questions?

