Chirp: Distributed Radar Network

Our Team

Jason Wang

MQTT Development

Oviya Seeniraj

Time Sync & Real-Time Adaption

Andrey Otvagin

Jetson/Node Development

William Ni

Integration

Vihan Jayaraman

Wireless Infrastructure

Motivation: RF Sensing = Seeing without Light

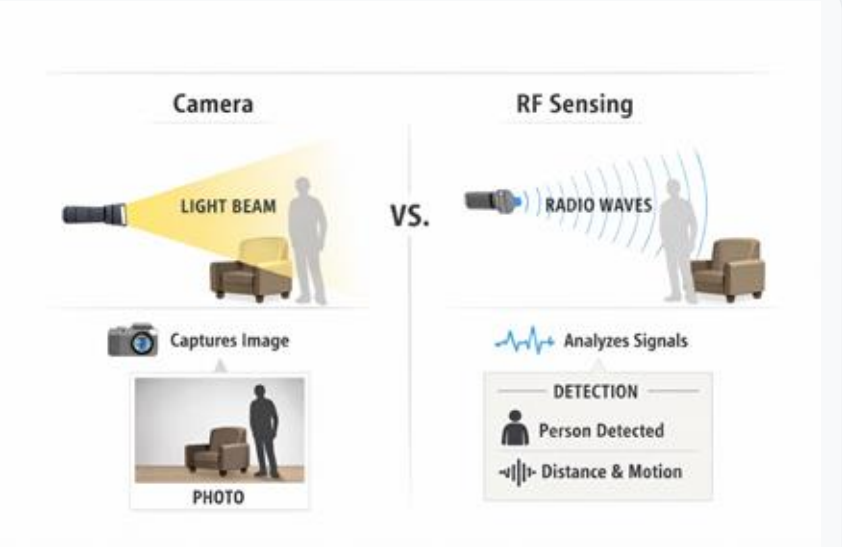


Fig. Camera vs RF Sensing

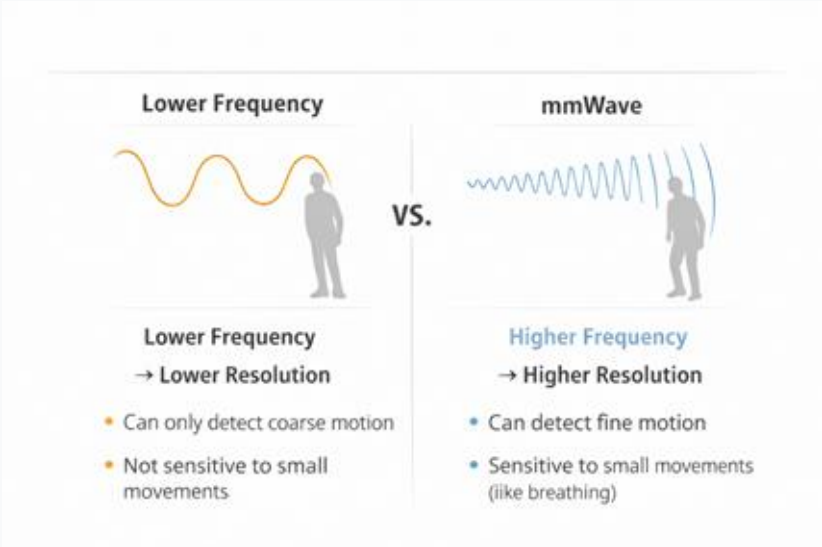
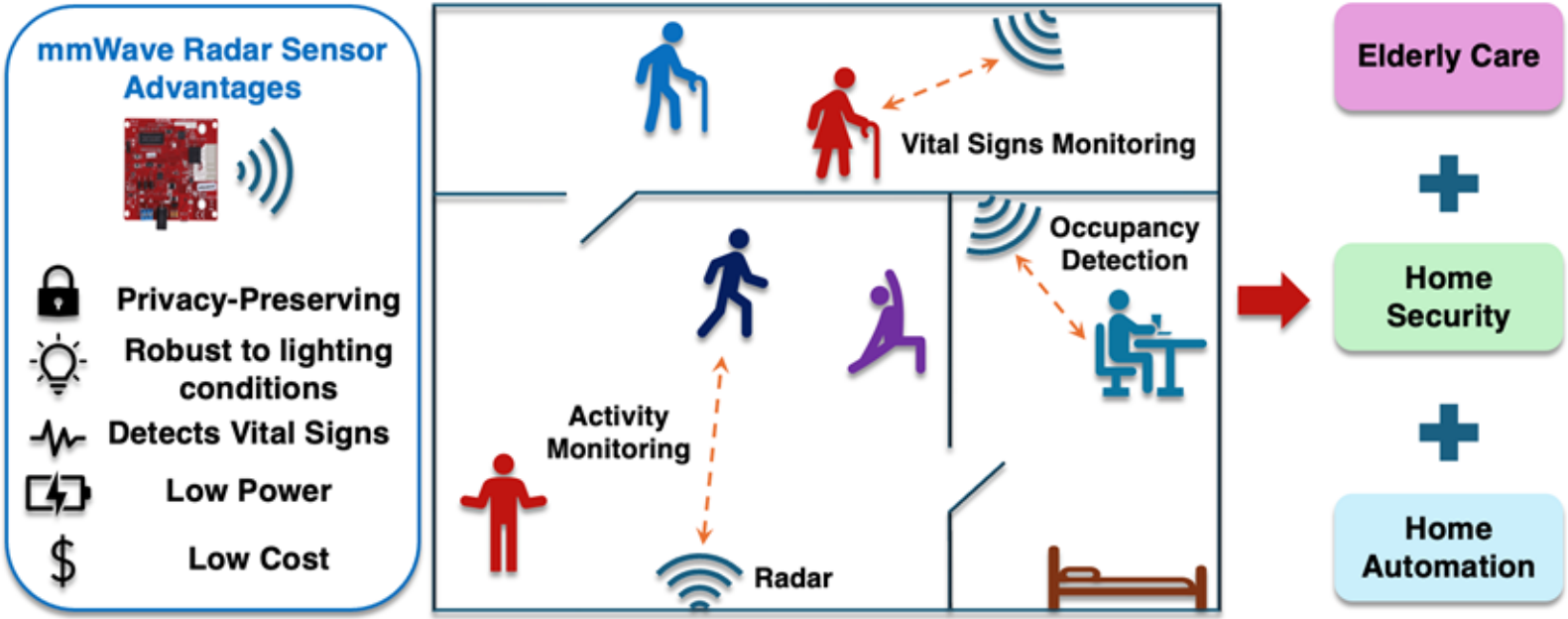


Fig. Sensing in mmWave vs Lower Frequency bands

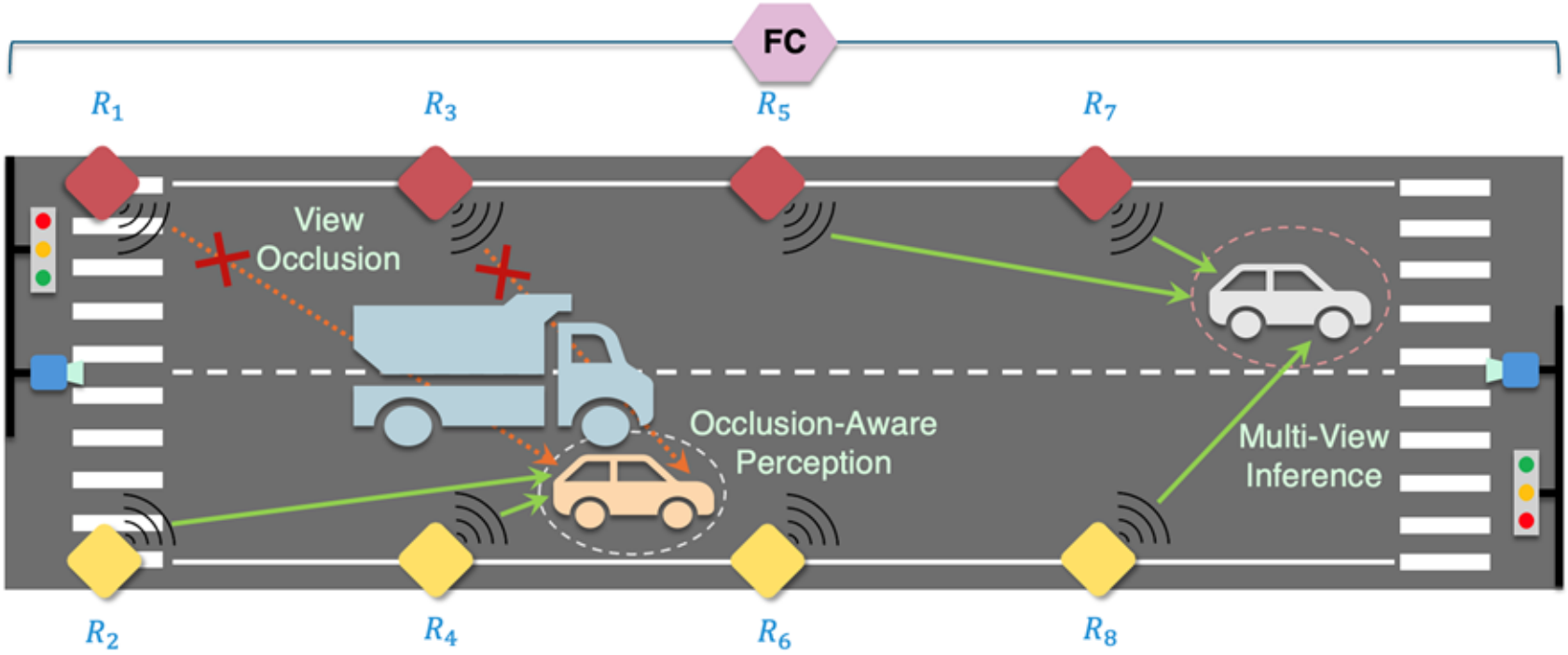
Motivation: mmWave Applications

Smart Homes enabled by mmWave Radar Sensors



Motivation: mmWave Applications

Collaborative Sensing on Roadside Deployments

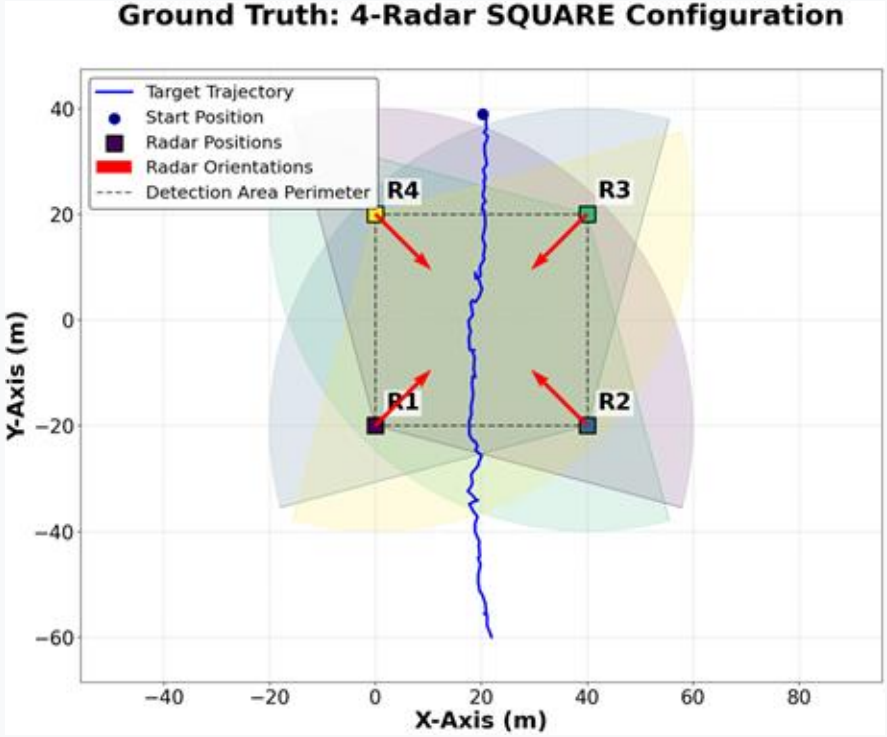


Goal & Purpose

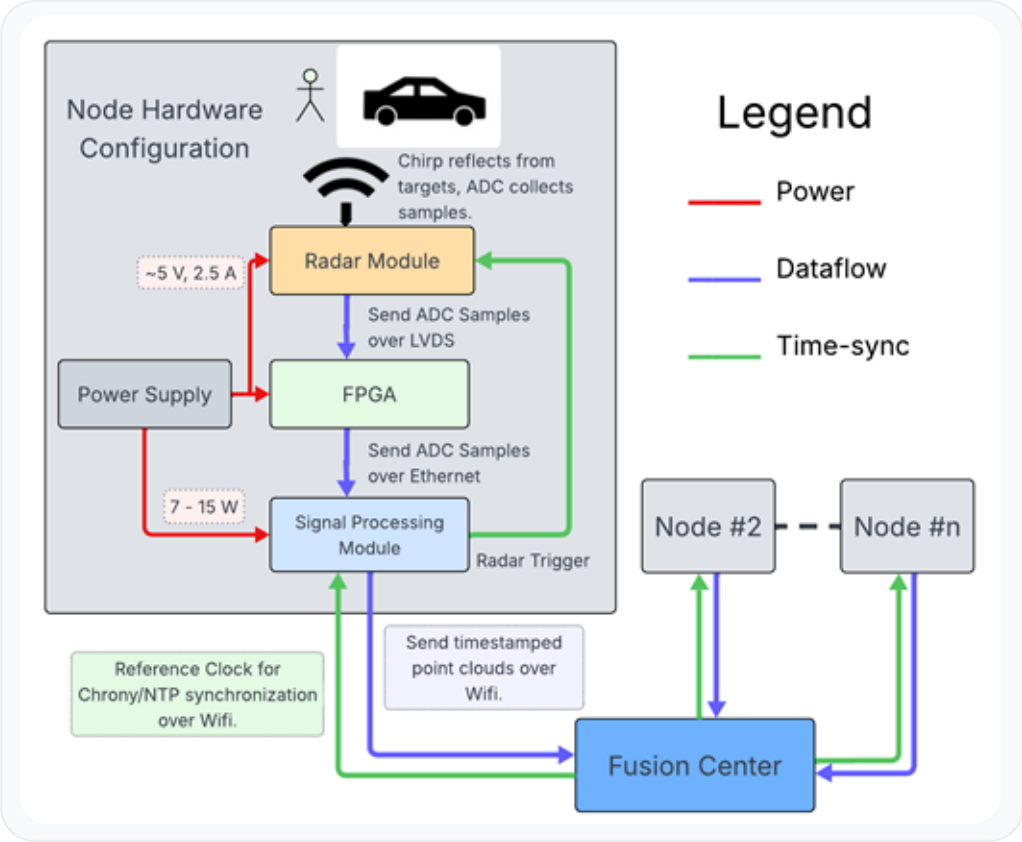
Demonstrate *Spatial Calibration* for distributed radar nodes

Achieved through 3 main steps:

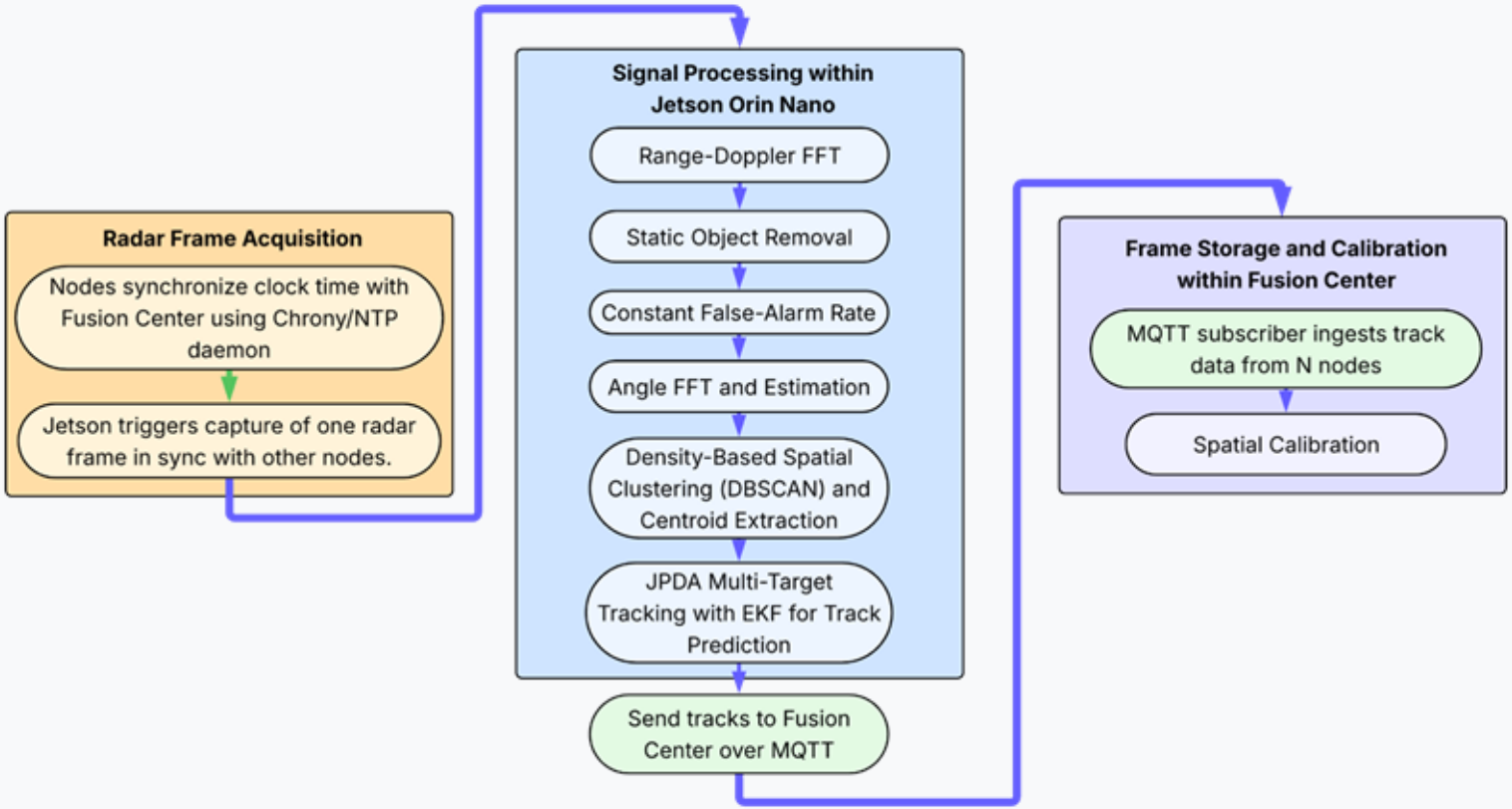
- Time Synchronization
- Reference Tracking
- Relative Pose Estimation



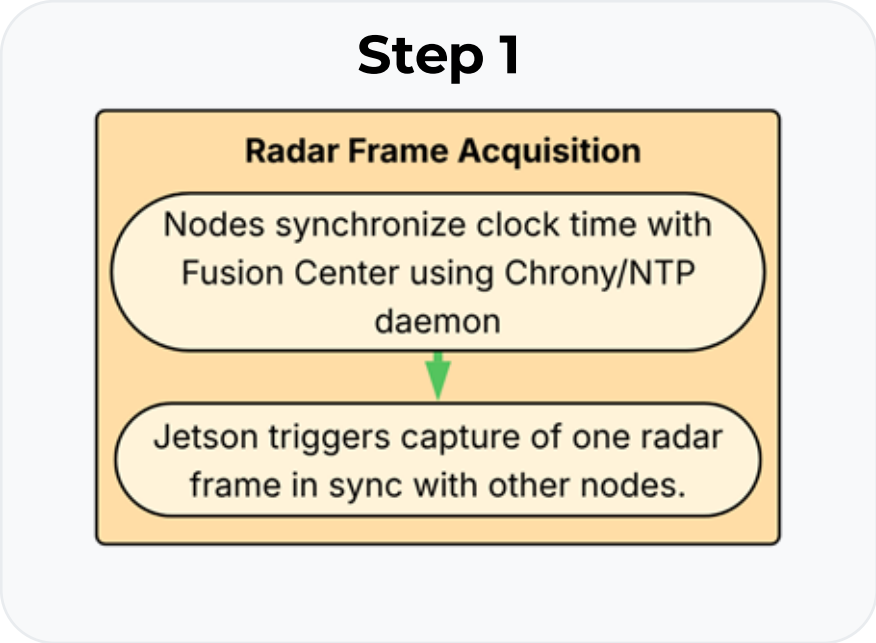
Block Diagram



Dataflow Overview



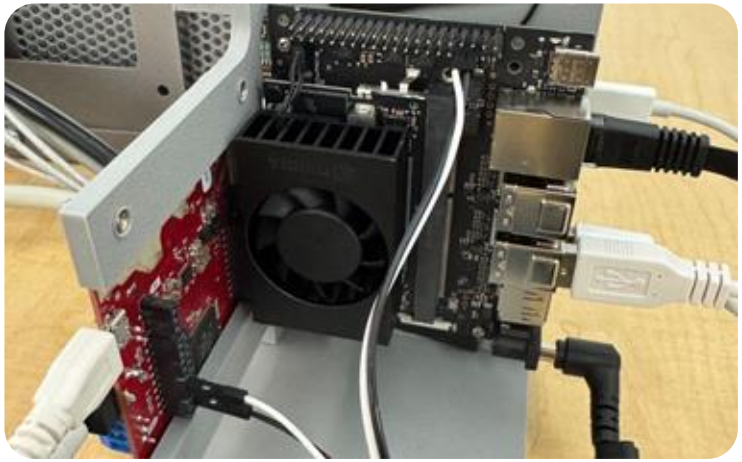
Dataflow Overview: Acquiring Radar Frames



Time Synchronization through Chrony NTP Daemon

Precise Synchronization

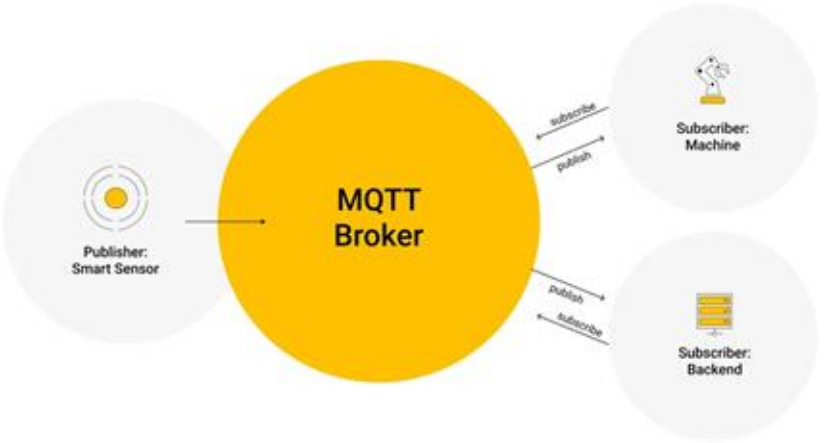
- Spatial Calibration requires that nodes capture frames at the same time
- Synchronize Nodes using “chrony”
- Setup Radars to capture a frame via a GPIO pulse
- Chrony NTP server served off of Fusion Center
- **~ 3 ms accuracy**



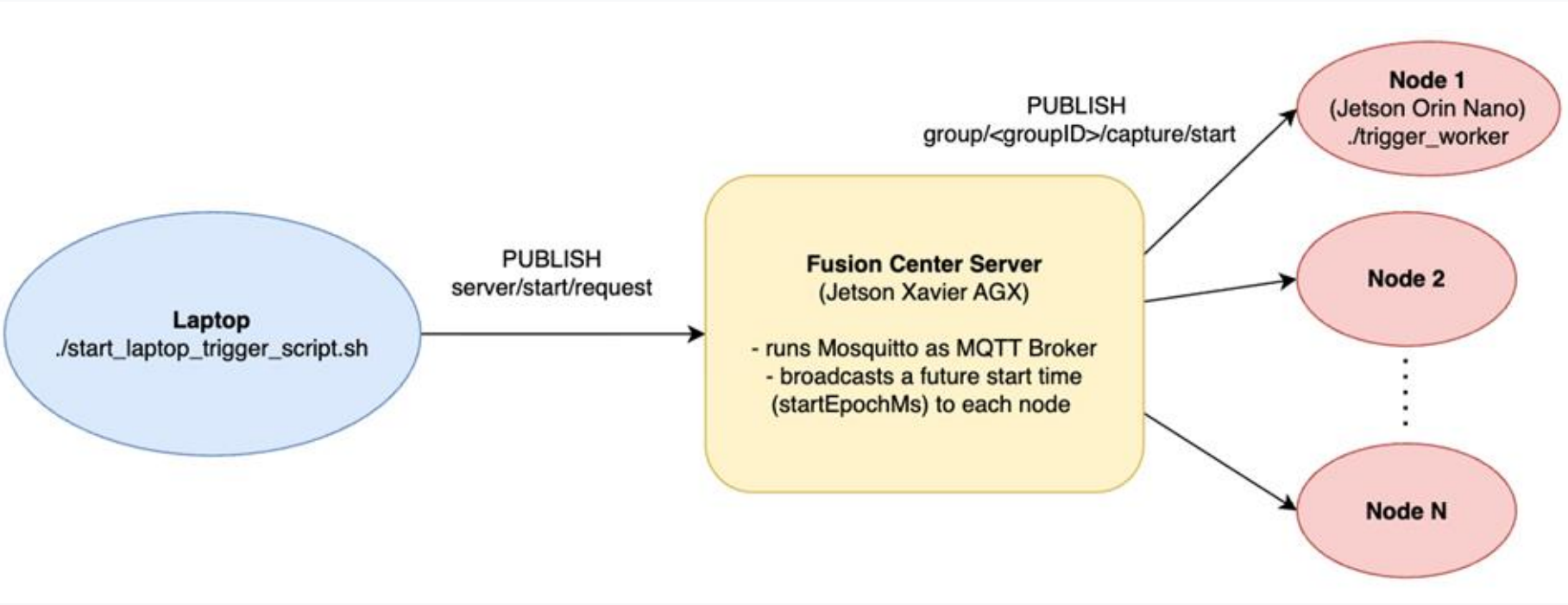
MQTT Overview for Communication between Nodes

What is MQTT?

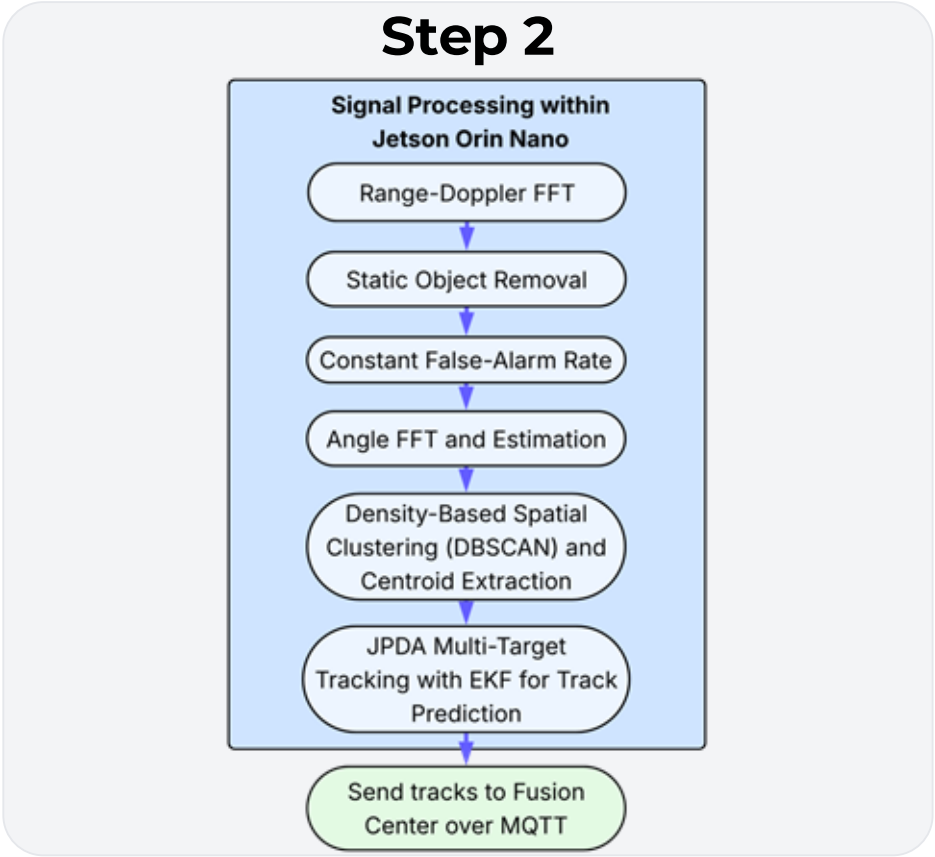
- **Message Queueing Telemetry Transport:** A lightweight, common protocol in IoT.
- **Pub/Sub Model:** Nodes communicate by publishing or subscribing to specific “topics”.
- **Centralized Routing:** The MQTT Broker handles all message distribution between nodes.



MQTT for Synchronized Trigger from Laptop



Dataflow Overview: Processing Radar Data



Radar Signal Processing Pipeline Purpose

CORE OBJECTIVE

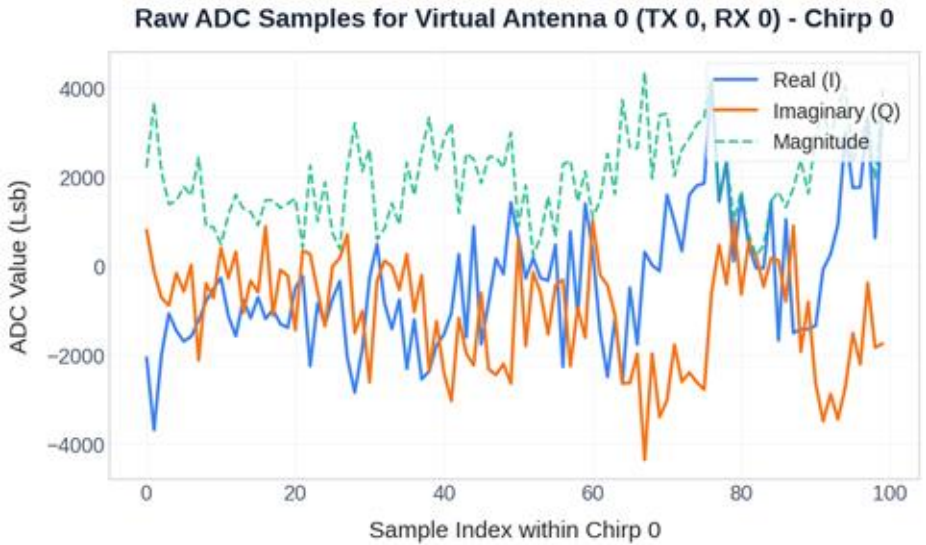
Stream of Raw ADC samples → Object Location & Velocity

Example Output: Human 1

Position: x: 1 m, y: 3 m

Velocity: v_x: 1 m/s, v_y: 0.5 m/s

Timestamp: 12h:34m:56s



Pipeline: Raw Data to Range Doppler Map

STEP 01

Input Stream

Data provided as a stream of raw ADC samples

STEP 02

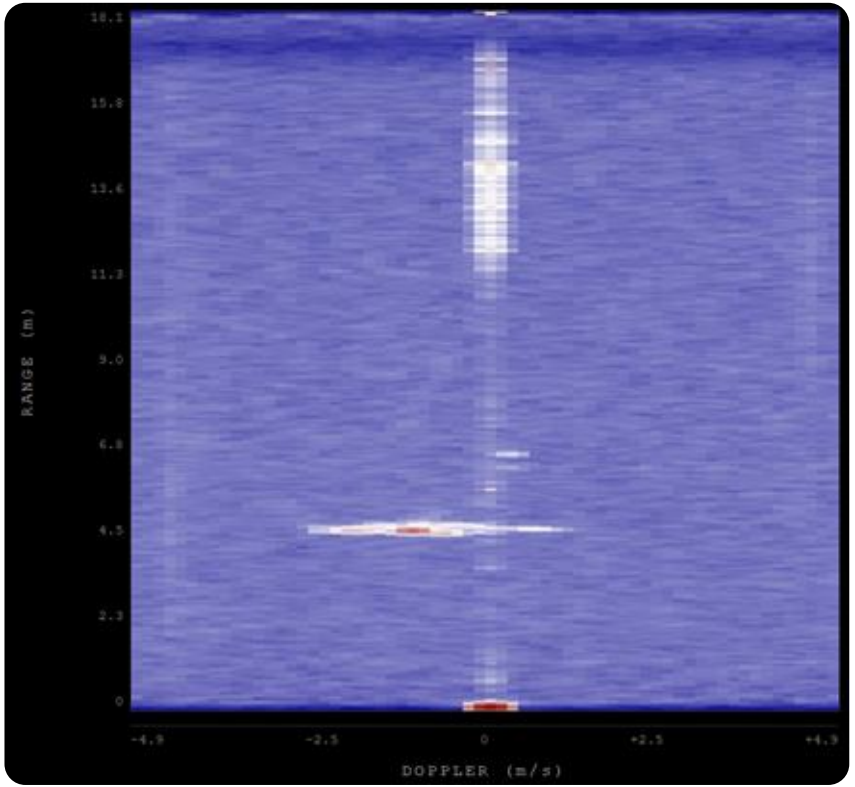
Processing

2D FFT to generate a Range Doppler Map

STEP 03

Static Filtering

Subtract power of stationary background to isolate moving targets



Visualizes magnitude per Range-Doppler bin

Pipeline: CFAR for Points of Interest

STEP 04

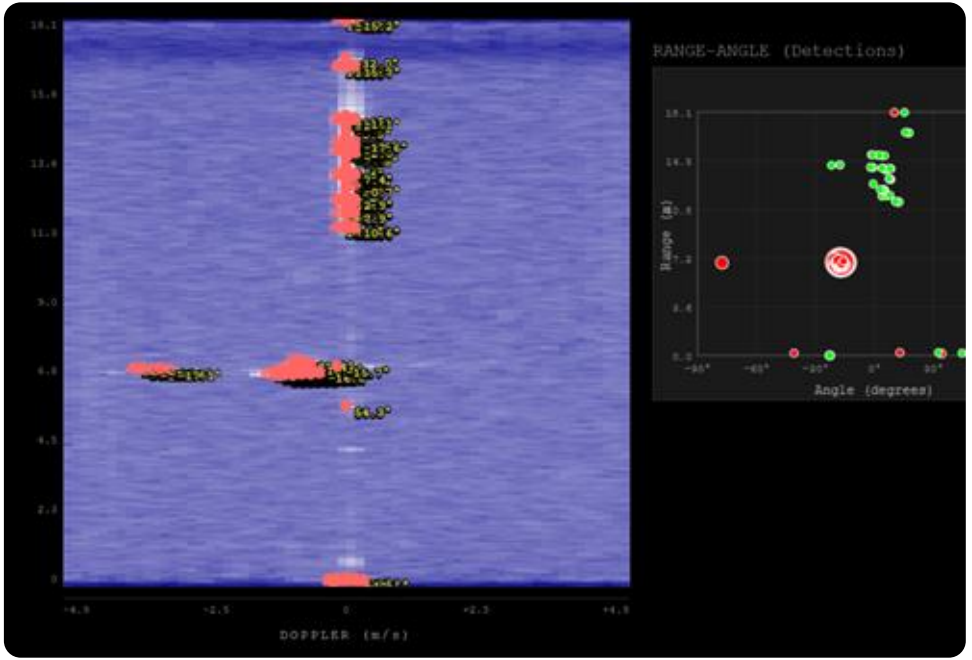
Detection

CFAR to identify potential points of interest

STEP 05

Angle Estimation

Calculate the angles of each of those points of interest

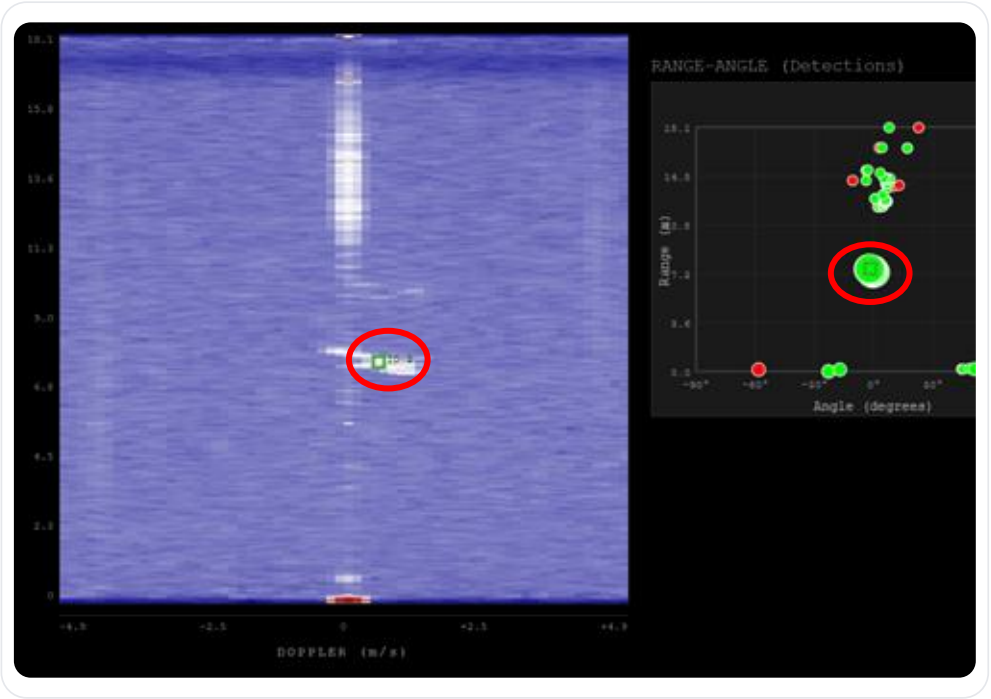


Pipeline: CFAR Point Clustering into Centroid

STEP 06

Clustering

Cluster CFAR Points (Range, Doppler, Angle) into Centroids using DBScan

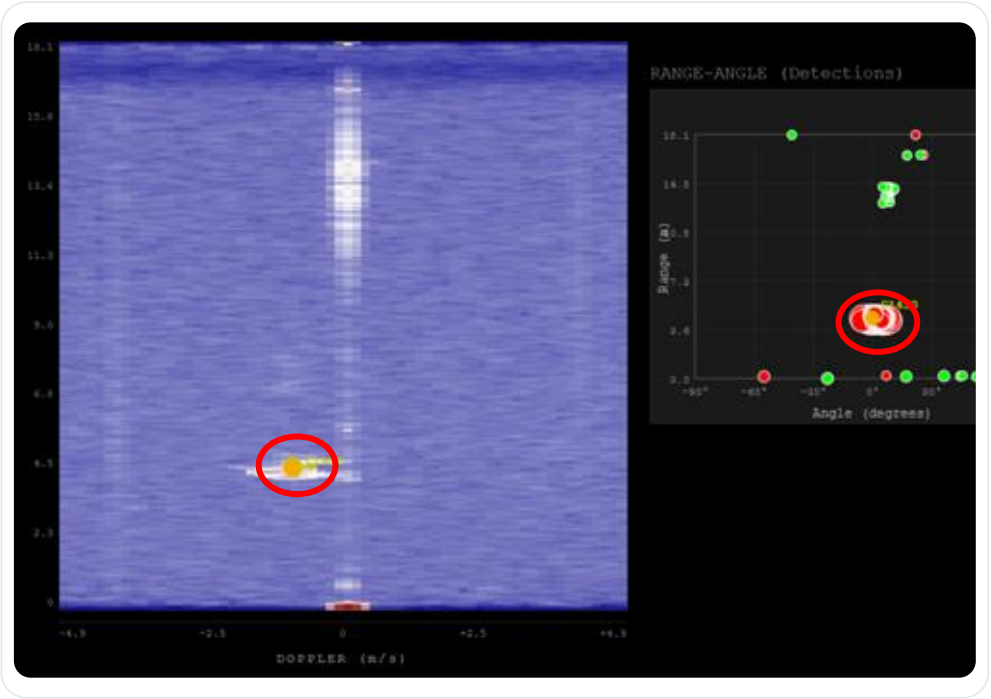


Pipeline: Filtering Centroid Noise

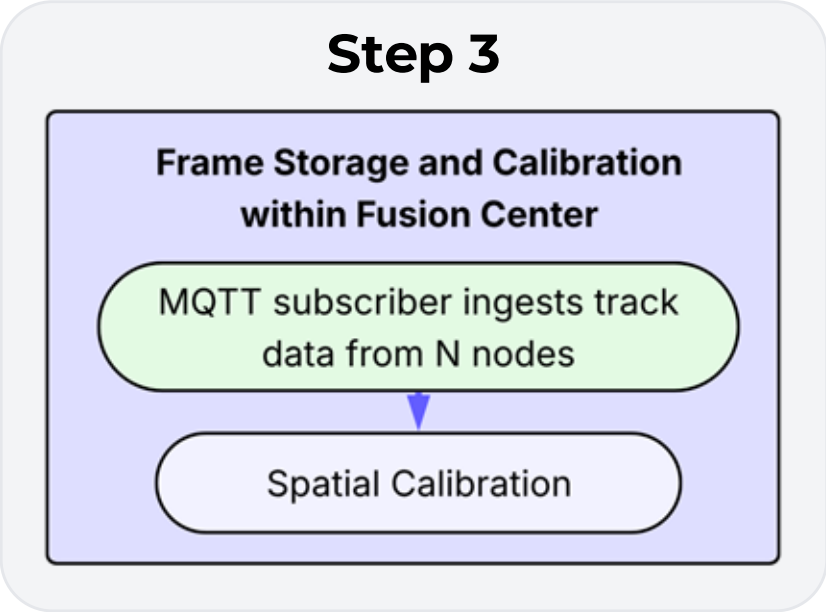
STEP 07

Noise Filtering

Filter Centroids across time using Extended Kalman Filtering (EKF) and Joint Probabilistic Data Association (JPDA)



Dataflow Overview: Spatial Calibration



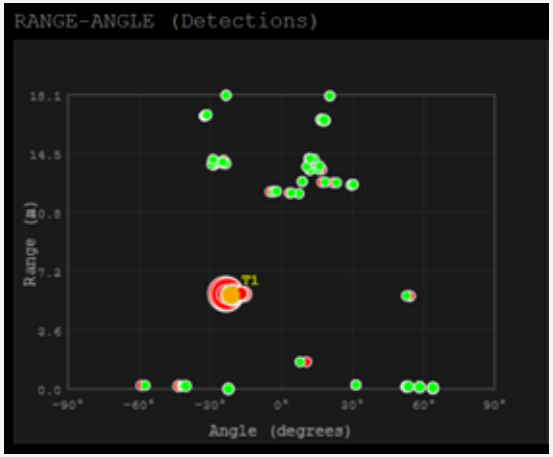
Spatial Calibration

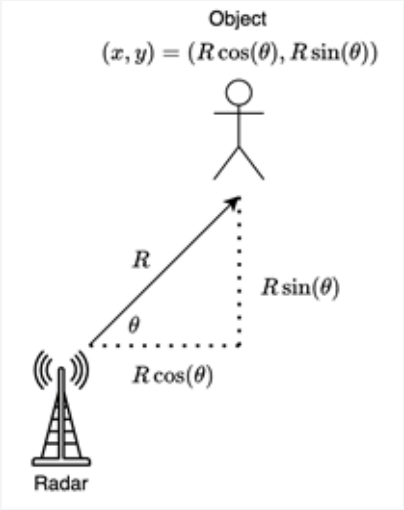
Goal:

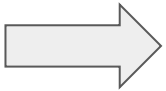
- Map all the trajectories together to create fused map
- Requires figuring out the translation & rotations

How?

- Publish frames w/ detection data
- Source common frames
- Turn detection list into centroids (blobs of ‘objects’)
- Algorithms outputting rotation and translation estimation
- Transpose node centroids w/ output to create bird’s eye map


RANGE-ANGLE (Detections)
18.1
14.5
10.8
7.2
3.6
0.0
Range (m)
T1
-90°
-60°
-30°
0°
30°
60°
90°
Angle (degrees)

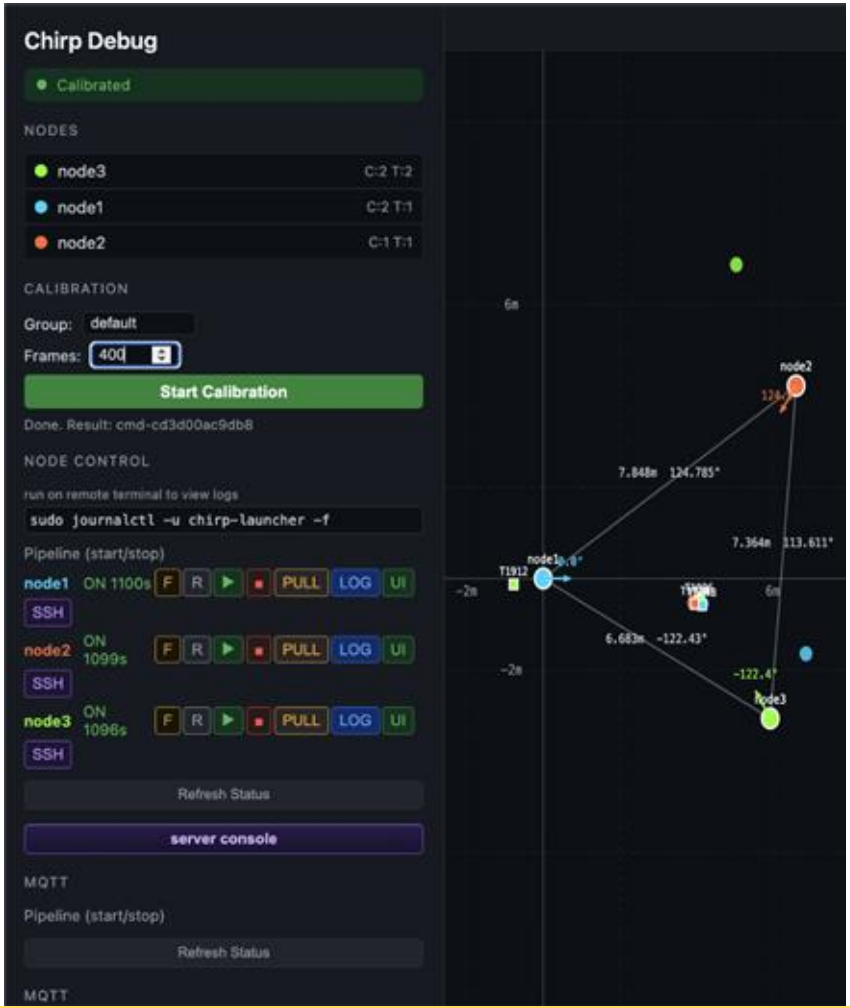

Object
$(x, y) = (R\cos(\theta), R\sin(\theta))$
R
$R\sin(\theta)$
θ
$R\cos(\theta)$
Radar




node3
node2

Fusion Center Front End

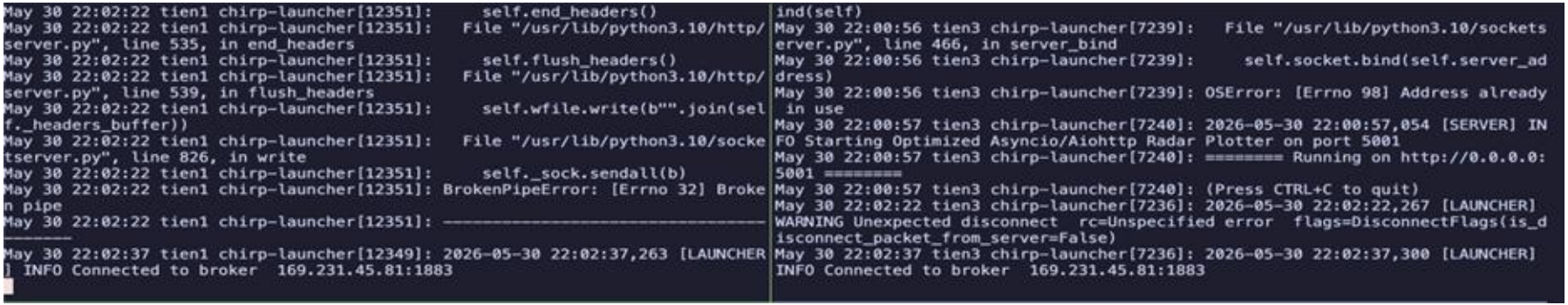
- Hosted on fusion center
- Main Features:
  - Pull from Github
  - per-Node RDM Visualizer (UI)
  - Local Logs
  - Start Calibration
  - Displays Calibration Results
- Allows future capstone team/researchers to quickly iterate and verify software.



Fusion Center Back End

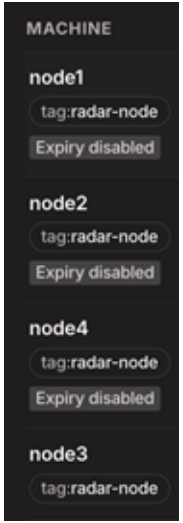
- Dockerized stack for easier deployment
    - Dashboard service for user interface
    - Calibration service for running node calibration
    - Bridge service for mqtt message ingest
    - NanoMQ mqtt broker service for handling mqtt transactions
    - Postgres service for storing centroids and tracks from nodes

Fusion Center Features



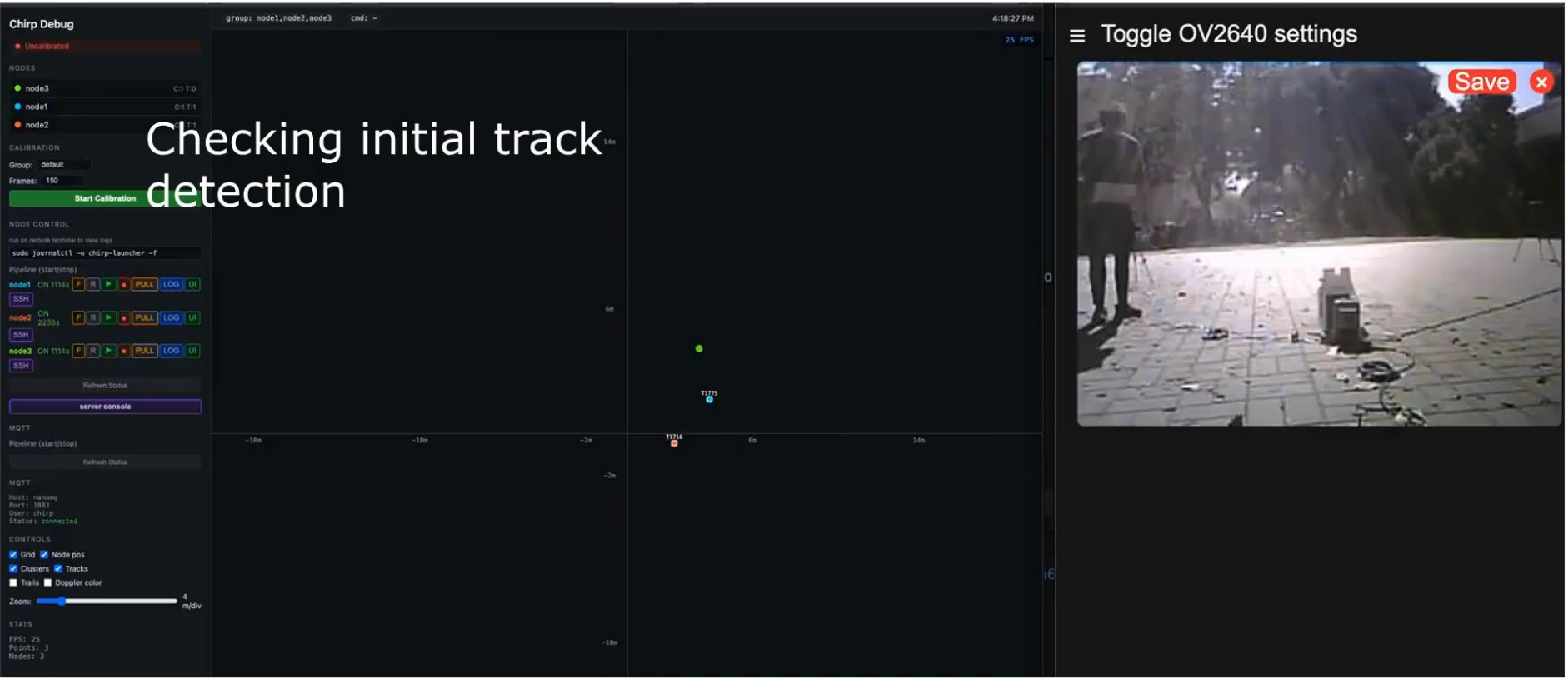
Web-accessible logs, per node

With Tailscale VPN and DNS resolution, nodes automatically connect to Fusion Center once powered on through daemon service.



DNS resolution

Spatial Calibration Demonstration



Thank you for listening!

Feel free to ask any questions.

Special thanks to: Anirban Banik, Tien Nguyen, Prof. Isukapalli, Prof. Madhow