# SEMTA Post-Processing Server

## Overview

* The SEMTA system uses a remote server to collect data and perform post-processing tasks such as data fusion and tracking.
* This system is implemented in two Python scripts. Data processing is performed by Tracking.py, while Server.py implements a simple HTTP server using Flask.

## Server Deployment

* A Docker container using Docker Compose has been configured for installation & deployment to any target operating system:
  + Install Docker and Docker Compose using preferred package manager
  + Set desired port in PostProcessing/docker-compose.yml, defaults to 5000
  + Issue the following commands within the “PostProcessing” directory:
    - docker-compose build
    - docker-compose up -d

## Client Usage

* Radar data is currently packaged in a .mat file, for use with MATLAB simulation and processing results. This format should be modified for file format of final use case.
  + The .mat file contains the following fields:
    - ‘range’
      * Estimated range to target
      * Floating point (# of radar units) x (# of frames) array
    - ‘vel’
      * Estimated Doppler range rate of target
      * Floating point (# of radar units) x (# of frames) array
    - ‘az’
      * Estimated azimuth bearing of target
      * Floating point (# of radar units) x (# of frames) array
    - ‘steer’
      * Antenna array steering direction
      * Floating point (# of radar units) x (# of frames) array
    - ‘SNR’
      * Estimated signal-to-noise ratio of target reflection
      * Floating point (# of radar units) x (# of frames) array
    - ‘hit\_list’
      * True/false indicator of whether a target detection occurred
      * Boolean (# of radar units) x (# of frames) array
* Radar measurement data is uploaded to server using HTTP POST request.
  + Default URI is ‘/’, for example http://127.0.0.1:5000/
  + File is attached to request body using ‘form-data’ format, with key “file”
* If file is successfully sent, it will be saved in the “PostProcessing/Input” folder.
  + A folder within “PostProcessing/Output” is generated using the name of the file.
  + .csv files of tracked target coordinates, along with .png files of resulting scatter plots, are saved to the new output file.
  + One of each type of output file is created for each single-unit tracking result, plus one for the multistatic results.
* Postman REST client API can be used to verify connection and correct operation.
  + GET request to root URI of ‘/’ should return HTTP status 200 if connected successfully

## Open Issues

* System will need to be made compatible with final system software.
  + .mat file will be replaced with output format used by radar system.
* System will need to be made compatible with asynchronous arrival of results.
  + Database using system such as MySQL or MongoDB can be implemented for this case.
* HTTP server will need to be exposed to WAN.
  + HTTP server currently runs on localhost/127.0.0.1