Blaine Harris
Fall 2025

WaveTap

Local ADS-B Receiver & SDR Utilities

Blaine Harris, September 5th, 2025

GitHub Repo: https://github.com/bwharris1125/CS7319_Project_WaveTap/

Project Overview

Main Functions

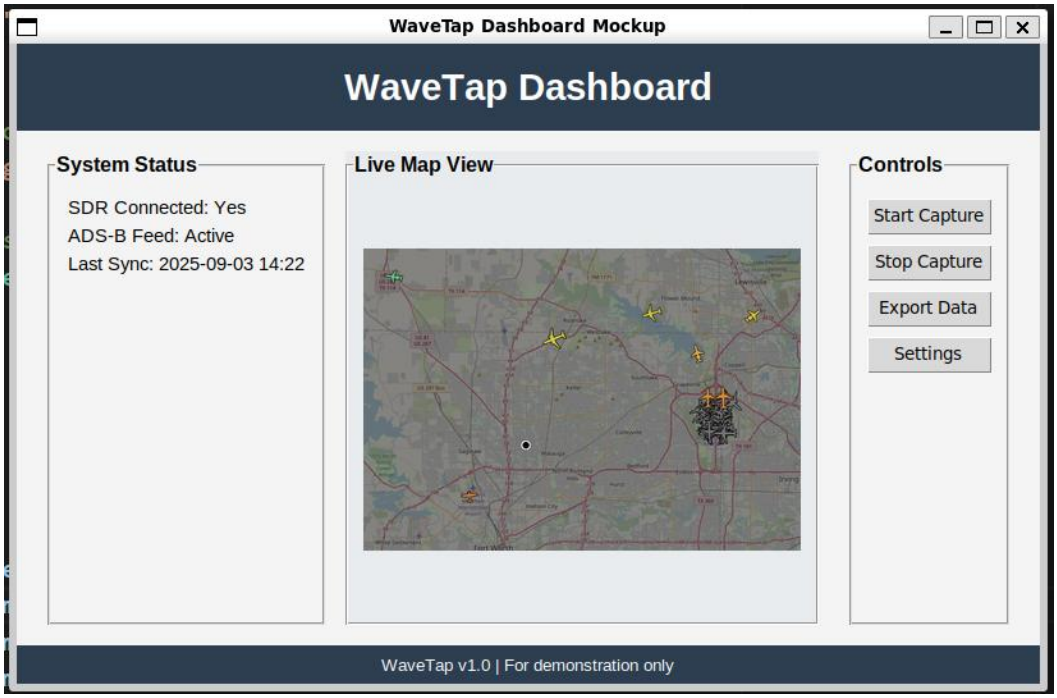
WaveTap is a microservices based platform for software defined radio (SDR) applications. For this project, the initial scope will be limited to ADS-B which is a public transmission from aircraft that provides tracking information. WaveTap allows users to tap into the radio waves all around them, monitor air traffic, and provides the expandability to capture additional signals using the SDR hardware. The microservices architecture allows for the application to be distributed across multiple devices to allow for easier maintainability, expandability, and modularity.

Expected Results

A distributed SDR platform where users can easily interface and modify components to utilize SDRs in a configurable and simple way. This will be comprised of multiple containers that perform actions such as receiving the SDR stream, decoding signals from the SDR, storing received data, and presenting data to an end user. Additionally, the application will provide a simple API interface that allows for simple and expandable

Prototype Screens

Due to my background in embedded systems, this application will primarily act as a distributed network of utilities for using a SDR. Therefore, a simple Python Tkinter UI is planned for interacting with the service. This python-based UI will interact using a web socket and should be able to run on any device with python.



The UI is focused on presenting interactions to control the physical hardware (the SDR), present live data of the ADS-B messages being received, and provide interfaces to store, capture, and view previous data.
Note, this UI is focused on the single use case of receiving ADS-B messages, but a similar interface can be developed and deployed for any application of the SDR. For other signals such as FM radio or first responder radio.

Project Design

Primary Architecture Style

This project follows option 1 outlined in the project guidance and utilizes a microservice architecture. This is composed of a microservice for each of the major functions of the application. SDR data stream capture, SDR stream decoding, resource coordinator/arbiter, event storage, API gateway (stretch goal), and a simple interface to view active and recorded data.

Architecture Diagrams

The following diagrams depict the proposed notional architecture for the WaveTap application. Diagrams were made using “Mermaid”, and the source for the diagrams can be viewed on the project repo.

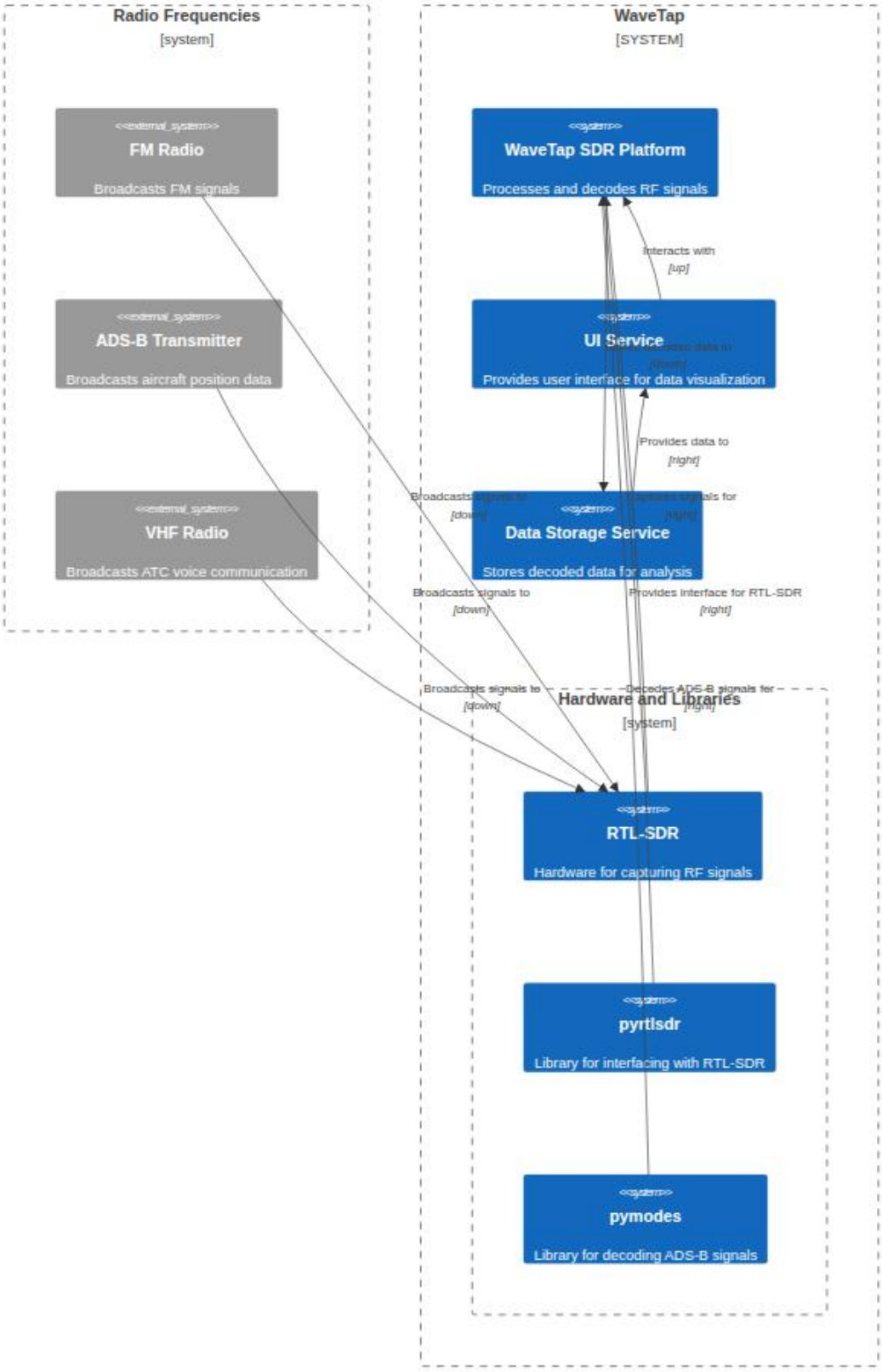


Figure 1: WaveTap Context Diagram

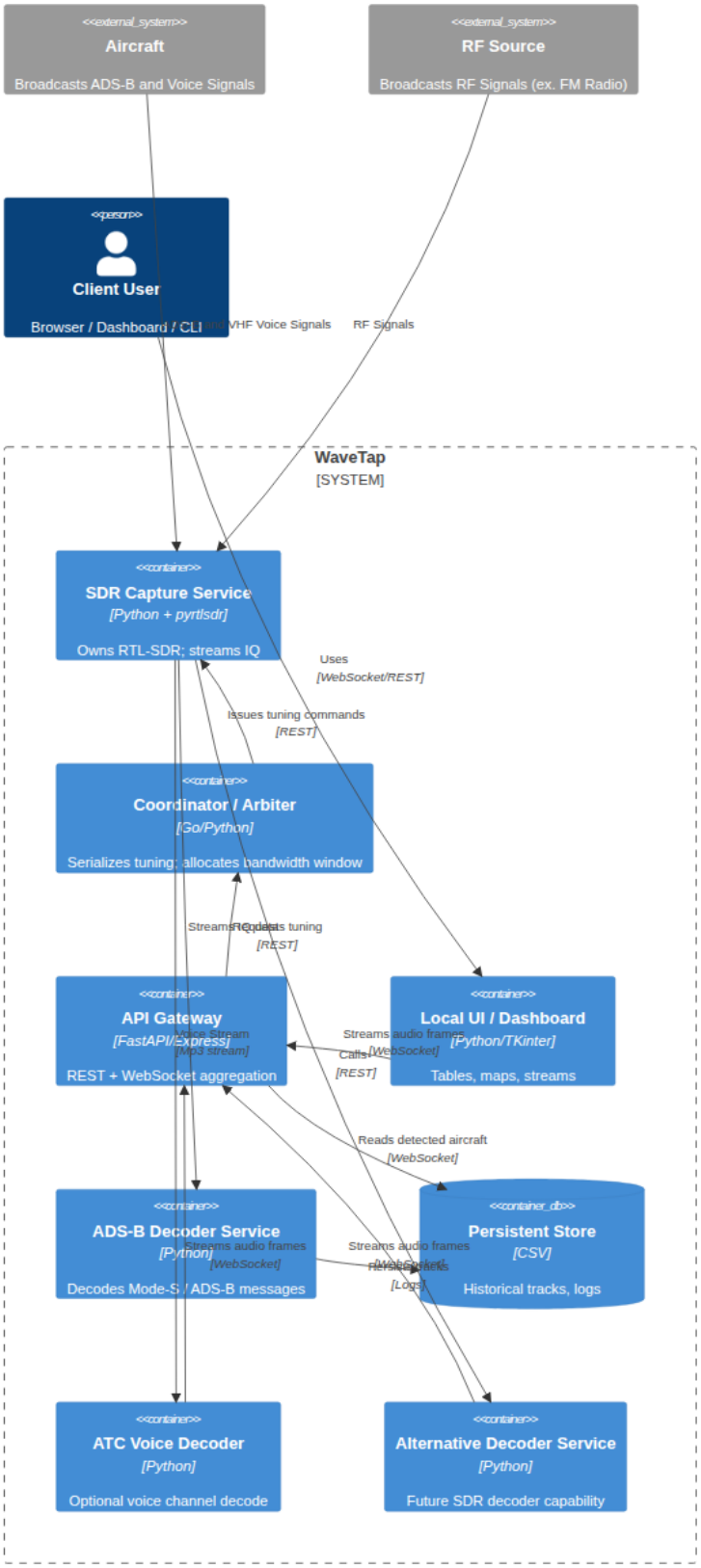


Figure 2: WaveTap Container Diagram

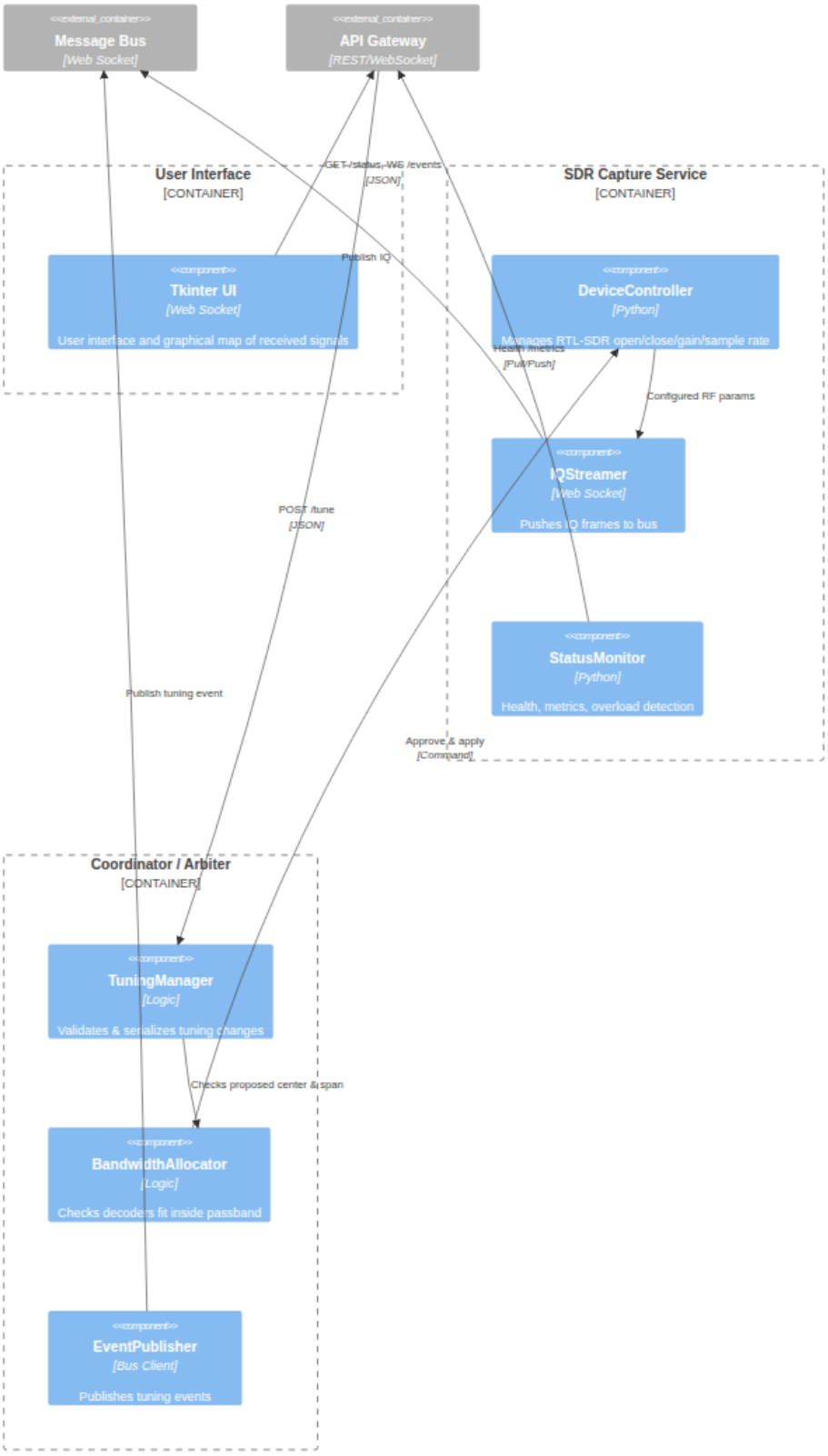


Figure 3: WaveTap Components Diagram

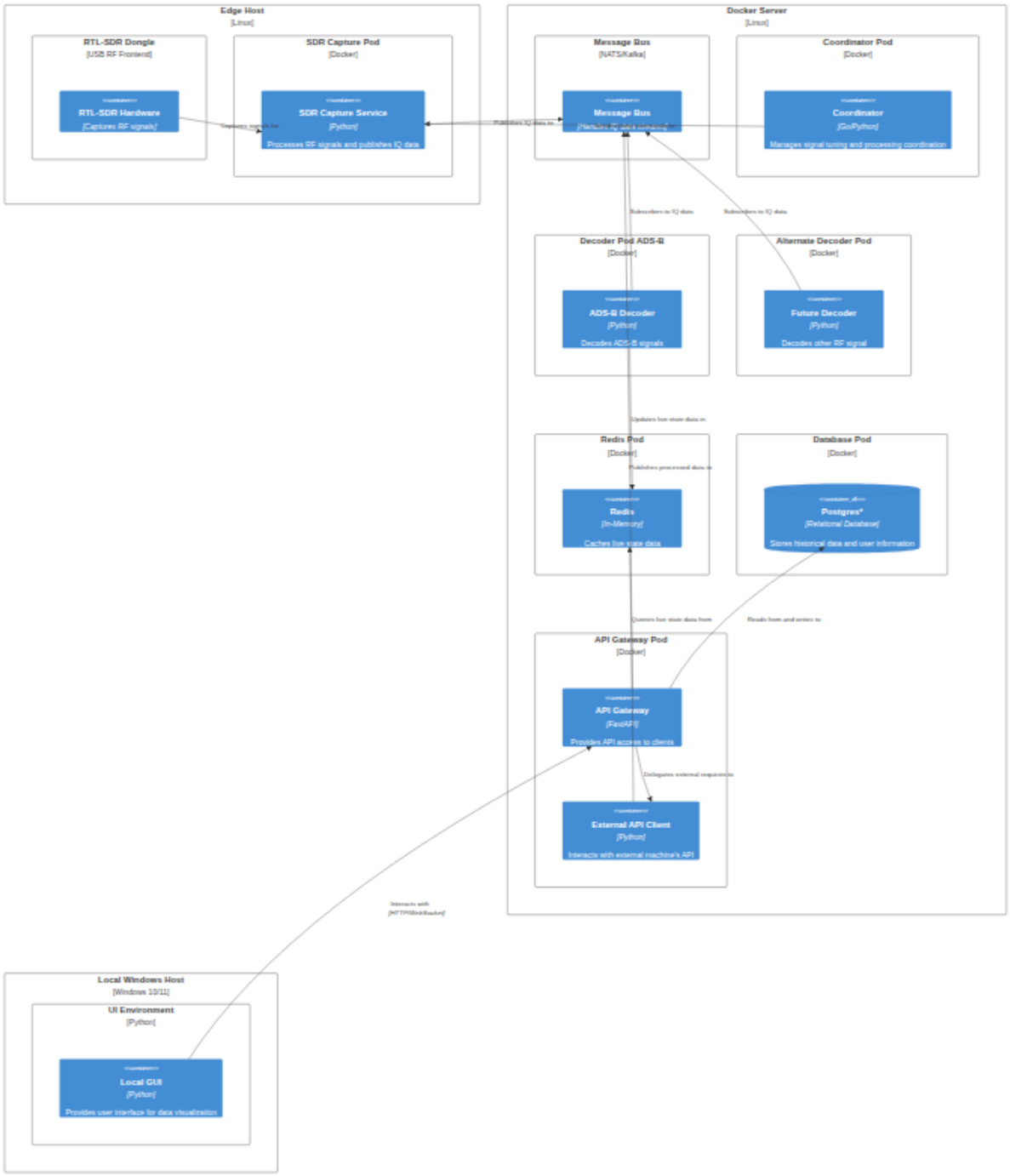


Figure 4: WaveTap Deployment Diagram

Project Implementation Plan

Platform & Tools

This project is planned to utilize python and open-source packages. Packages include:

- PyRTLSDR
- pyModeS
- FastAPI

This project will also utilize hardware such as a USB RTL-SDR device, and optionally a Raspberry Pi to host the SDR service.

Containerization is being used to implement the microservice architecture, Docker is being utilized for this since this application is not safety or security critical enough to drive using Podman.

Deployment

Deployment initially (for this project) will be conducted using Docker for local deployment. Future scalability could include things such as Kubernetes to manage multiple SDR capture services. The docker containers utilized will use Docker networking to communicate between components of the application.

Timeline

The Gantt chart below shows the anticipated timeline for this project and its completion.