

tangram, an open platform for modular, real-time air traffic management research

Xavier Olive¹, Junzi Sun², Xiaogang Huang³, and Michel Khalaf¹

¹ ONERA – DTIS, Université de Toulouse, Toulouse, France ² Delft University of Technology, Delft, The Netherlands ³ Shinetech Software, Beijing, China

DOI: [10.21105/joss.08662](https://doi.org/10.21105/joss.08662)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: Anastassia Vybornova

Reviewers:

- [@fredshone](#)
- [@abc8747](#)

Submitted: 04 June 2025

Published: 20 October 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

tangram is an open research framework for ADS-B and Mode S flight surveillance data. Initially developed for turbulence detection research, it has since evolved into a versatile tool applicable to real-time aviation research topics, including GNSS jamming detection, aviation weather monitoring, emission analysis, and airport performance monitoring.

The system comprises a JavaScript-based web application and a backend implemented in Python or Rust, depending on performance requirements. The web application is modular, enabling users to add custom plugins for data processing and analysis. The backend handles data collection, storage, and processing, while the frontend offers an interface for visualizing and interacting with the data.

The whole framework is designed to be extensible, allowing researchers to develop and integrate their plugins for specific research needs. This modularity enables the community to contribute to the platform.

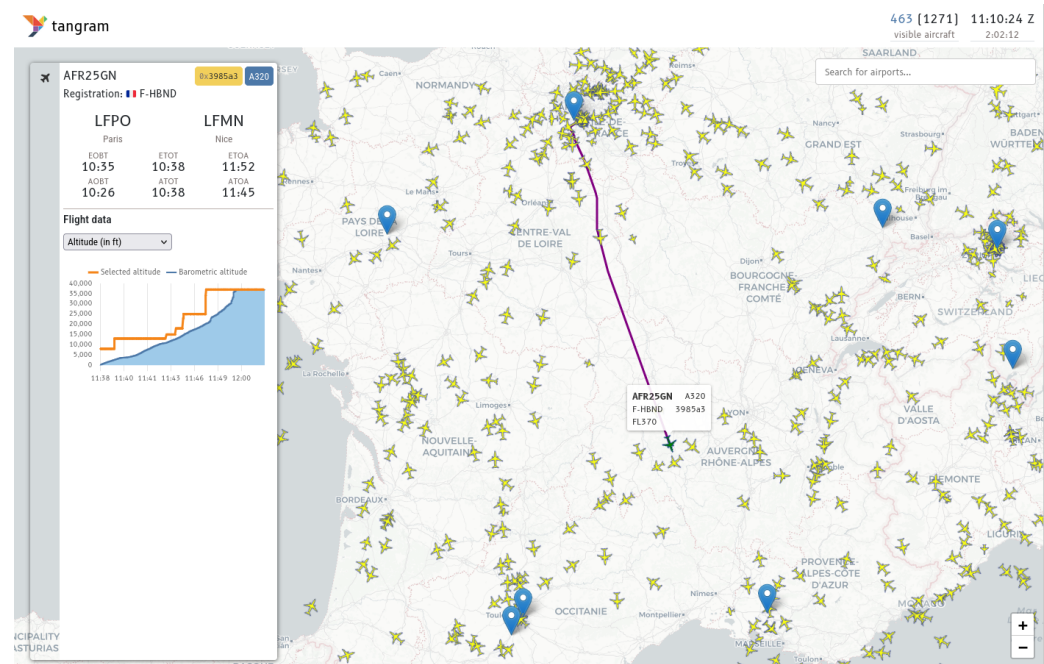


Figure 1: The tangram web application

Statement of need

The world of aviation involves many stakeholders producing a vast amount of data, but most of it is not publicly available for confidentiality and security reasons. Research in aviation is also often limited by the availability of data, but this has evolved in the past decade thanks to open data initiatives (EUROCONTROL, n.d.).

ADS-B and Mode S enable aircraft to broadcast their position and flight parameters, allowing real-time air traffic data collection. This data supports research on flight patterns (Olive et al., 2025), emissions (Seymour et al., 2020), and air traffic performance (Schultz et al., 2021). Since ADS-B is unencrypted and easily decoded with software-defined radios, many open-source tools, such as dump1090 (Sanfilippo, 2013), pyModeS (Sun et al., 2020), and platforms like The OpenSky Network (Schäfer et al., 2014), have emerged, providing researchers with free access to real-time and historical aviation data.

Most research initiatives and proofs of concept based on ADS-B data are limited to the use of historical data (Olive, 2019). However, the potential impact of such research depends on the possibility of adapting the algorithms to real-time data. The tangram framework aims to bridge this gap by providing a platform for real-time data collection and analysis.

Core structure of the framework

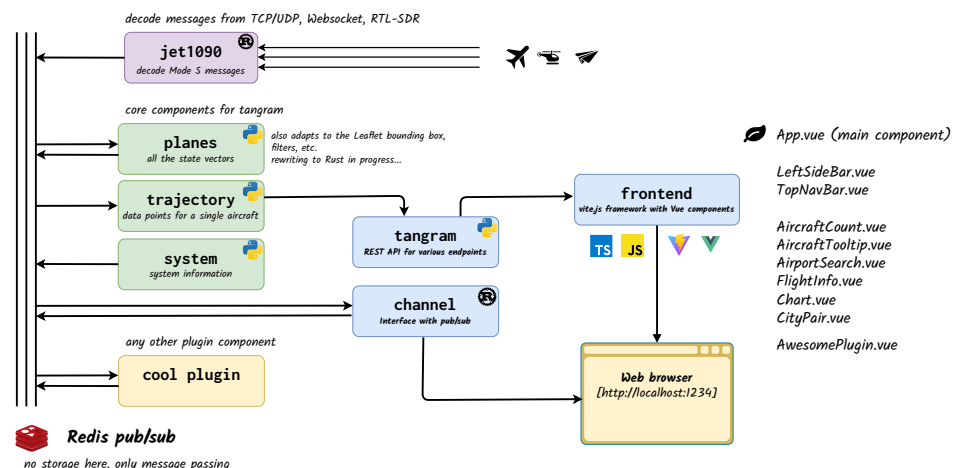
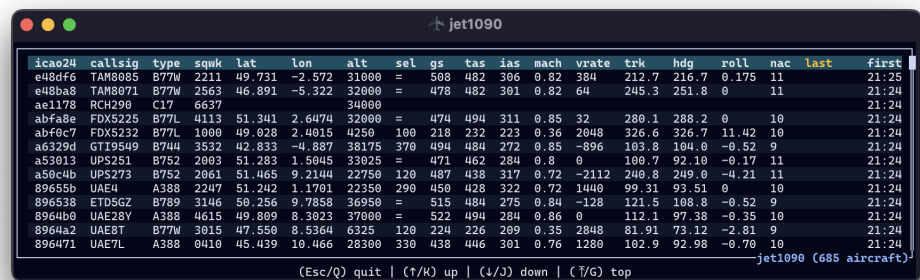


Figure 2: Core structure of the framework

The tangram framework consists of a suite of independent components that can be combined to create a powerful and flexible system (see Figure 2). The web application is built upon the Vite framework and consists of a series of Vue plugins. The interaction between the frontend and backend is based on REST APIs and WebSocket connections. The REST API (the tangram application) is used for data retrieval and management, while the WebSocket connection (through the channel executable) allows for real-time communication between the frontend and backend. The backend is responsible for data collection, storage, and processing. All the components communicate through a Redis pub/sub system, which allows for efficient data exchange between components and real-time updates.

jet1090

The jet1090 (Olive, 2025) tool is a Rust-based ADS-B decoder that can be used to decode ADS-B messages from a variety of sources, including software-defined radio devices. It is designed to be fast and efficient, making it suitable for real-time applications. jet1090 is comparable to the historical dump1090 decoder (Sanfilippo, 2013), with additional features such as the ability to decode Extended Mode S messages.



The screenshot shows a terminal window titled 'jet1090' displaying a table of aircraft data. The table has columns for icao24, callsig, type, sqwk, lat, lon, alt, sel, gs, tas, ias, mach, vrte, trk, hdg, roll, nac, last, and first. The data is sorted by 'last' time. The bottom of the terminal shows '(Esc/Q) quit | (t/M) up | (i/J) down | (T/G) top' and 'jet1090 (685 aircraft)'.

icao24	callsig	type	sqwk	lat	lon	alt	sel	gs	tas	ias	mach	vrte	trk	hdg	roll	nac	last	first
e48df6	TAM8085	B77W	2211	49.731	-2.572	31000	=	508	482	306	0.82	384	212.7	216.7	0.175	11	21:25	
e48ba8	TAM8071	B77W	2563	46.891	-5.322	32000	=	478	482	301	0.82	64	245.3	251.8	0	11	21:24	
a61173	RCH290	C17	6637			34000											21:24	
abfa8e	FDX5225	B77L	4113	51.341	2.6474	32000	=	474	494	311	0.85	32	280.1	288.2	0	10	21:24	
abf8c7	FDX5232	B77L	1000	49.028	2.4015	4250	100	218	232	223	0.36	2048	326.6	326.7	11.42	10	21:24	
a6329d	GT19549	B744	3532	42.833	-4.887	38175	370	494	484	272	0.85	-896	103.8	104.0	-0.52	9	21:24	
a53013	UPS251	B752	2003	51.283	1.5045	33025	=	471	462	284	0.8	0	100.7	92.10	-0.17	11	21:24	
a50c4b	UPS273	B752	2061	51.465	9.2144	22750	120	487	438	317	0.72	-2112	240.8	249.0	-4.21	11	21:24	
89655b	UAE4	A388	2247	51.242	1.1701	22350	290	450	428	322	0.72	1440	99.31	93.51	0	10	21:24	
896538	ETD5GZ	B789	3146	50.256	9.7858	36950	=	515	484	275	0.84	-128	121.5	108.8	-0.52	9	21:24	
89640b	UAE20V	A388	4615	49.009	8.3023	37000	=	522	494	284	0.86	0	112.1	97.38	-0.35	10	21:24	
8964a2	UAE8T	B77W	3015	47.550	8.5364	6325	120	224	226	209	0.35	2848	81.91	73.12	-2.81	9	21:24	
896471	UAE7L	A388	0410	45.439	10.466	28300	330	438	446	301	0.76	1280	102.9	92.98	-0.70	10	21:24	

Figure 3: The table view of jet1090 in the terminal

planes

The planes module is a Rust-based component that maintains a real-time table of aircraft states. It updates aircraft positions and parameters using data from jet1090, ensuring the frontend has current information. Since different ADS-B messages provide different parameters (e.g., position, speed, identification), the state vector table aggregates the latest values, enabling accurate aircraft display on the map.

trajectory

The trajectory module is a Python-based component that provides the history of data for a given aircraft. It is responsible for storing and retrieving historical data about the aircraft's position and other parameters. The component uses the data stored by the Redis system and reformats it to be used by the frontend in a more standard JSON-like format.

tangram REST API

The core tangram component is a Python-based REST API that provides data retrieval and management capabilities. It is responsible for handling requests from the frontend and providing the necessary data for visualization and analysis. The API is designed to be modular and extensible, allowing the user to add their endpoints and functionality as needed. Basic endpoints provided by the API include the data from trajectory and planes. Since the component is based on FastAPI, it is also possible to dynamically add new endpoints to the API at the plugin level.

channel

The channel component (Huang, 2025) is a Rust-based WebSocket connection that makes the bridge between the frontend and the Redis pub/sub system. It is responsible for providing real-time updates from and to the frontend.

The plugin system in tangram

The tangram system is designed to be extensible. Users can easily implement particular functionalities by creating their own backend and frontend plugins.

A backend plugin is usually an executable or a Python module which could, for example:

- listen to a Redis channel;
- access private data or services requiring authentication;
- post-process and enrich data;
- share the resulting information on a Redis channel or REST endpoint.

A frontend plugin will usually be a Vue component, which could, for example:

- add HTML elements to the web application (in the navigation bar or the sidebar);
- capture user interactions (e.g., mouse clicks, keyboard events);
- send requests to the backend (e.g., through the REST API or WebSocket);
- display data on the map (e.g., by adding markers, drawing lines, etc.)

Potential applications

The tangram framework has been thought to display ADS-B data on a map, similarly to the OpenSky Network or FlightRadar24, and to enrich it to include extra visualization features based on previously developed algorithms for turbulence detection (Olive & Sun, 2020), GNSS jamming detection (Felux et al., 2024), anomaly detection (Basora et al., 2019), airspace occupancy metrics or aircraft trajectory prediction (Jarry et al., 2025).

Acknowledgement

This project has been funded by the Dutch Research Council (NWO) Open Science Fund OSF 23.1.051 <https://www.nwo.nl/en/projects/osf231051>

References

- Basora, L., Olive, X., & Dubot, T. (2019). Recent Advances in Anomaly Detection Methods Applied to Aviation. *Aerospace*, 6(11), 117. <https://doi.org/10.3390/aerospace6110117>
- EUROCONTROL. (n.d.). *Aviation Data for Research*. <https://www.eurocontrol.int/dashboard/rnd-data-archive>
- Felux, M., Fol, P., Figuet, B., Waltert, M., & Olive, X. (2024). Impacts of Global Navigation Satellite System Jamming on Aviation. *Navigation*, 71(3). <https://doi.org/10.33012/navi.657>
- Huang, X. (2025). *WebSocket-based Real-Time Data Streaming Framework*. <https://github.com/emctoo/channel>
- Jarry, G., Dalmau, R., Olive, X., & Very, P. (2025). *A Neural ODE Approach to Aircraft Flight Dynamics Modelling* (No. arXiv:2509.23307). arXiv. <https://doi.org/10.48550/arXiv.2509.23307>
- Olive, X. (2019). traffic, a toolbox for processing and analysing air traffic data. *Journal of Open Source Software*, 4(39), 1518. <https://doi.org/10.21105/joss.01518>
- Olive, X. (2025). *jet1090: A Real-Time Mode S and ADS-B Decoder in Rust*. <https://mode-s.org/jet1090/>

- Olive, X., Basora, L., Sun, J., & Spinielli, E. (2025). Training a Machine Learning Model to Detect Holding Patterns in Aircraft Trajectories. *Journal of Open Aviation Science*, 2(2). <https://doi.org/10.59490/joas.2024.7943>
- Olive, X., & Sun, J. (2020). Detecting and Measuring Turbulence from Mode S Surveillance Downlink Data. *Proceedings of the 9th International Conference on Research in Air Transportation*. http://www.icrat.org/ICRAT/seminarContent/2020/papers/ICRAT2020_paper_3.pdf
- Sanfilippo, S. (2013). *dump1090: A simple Mode-S Decoder for RTLSDR devices*. <https://github.com/antirez/dump1090>
- Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., & Wilhelm, M. (2014). Bringing up OpenSky: A large-scale ADS-B sensor network for research. *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, 83–94. <https://doi.org/10.1109/ipsn.2014.6846743>
- Schultz, M., Reitmann, S., & Alam, S. (2021). Predictive classification and understanding of weather impact on airport performance through machine learning. *Transportation Research Part C: Emerging Technologies*, 131, 103119. <https://doi.org/10.1016/j.trc.2021.103119>
- Seymour, K., Held, M., Georges, G., & Boulouchos, K. (2020). Fuel Estimation in Air Transportation: Modeling global fuel consumption for commercial aviation. *Transportation Research Part D: Transport and Environment*, 88, 102528. <https://doi.org/10.1016/j.trd.2020.102528>
- Sun, J., Vu, H., Ellerbroek, J., & Hoekstra, J. M. (2020). pyModeS: Decoding Mode-S Surveillance Data for Open Air Transportation Research. *IEEE Transactions on Intelligent Transportation Systems*, 21(7), 2777–2786. <https://doi.org/10.1109/TITS.2019.2914770>