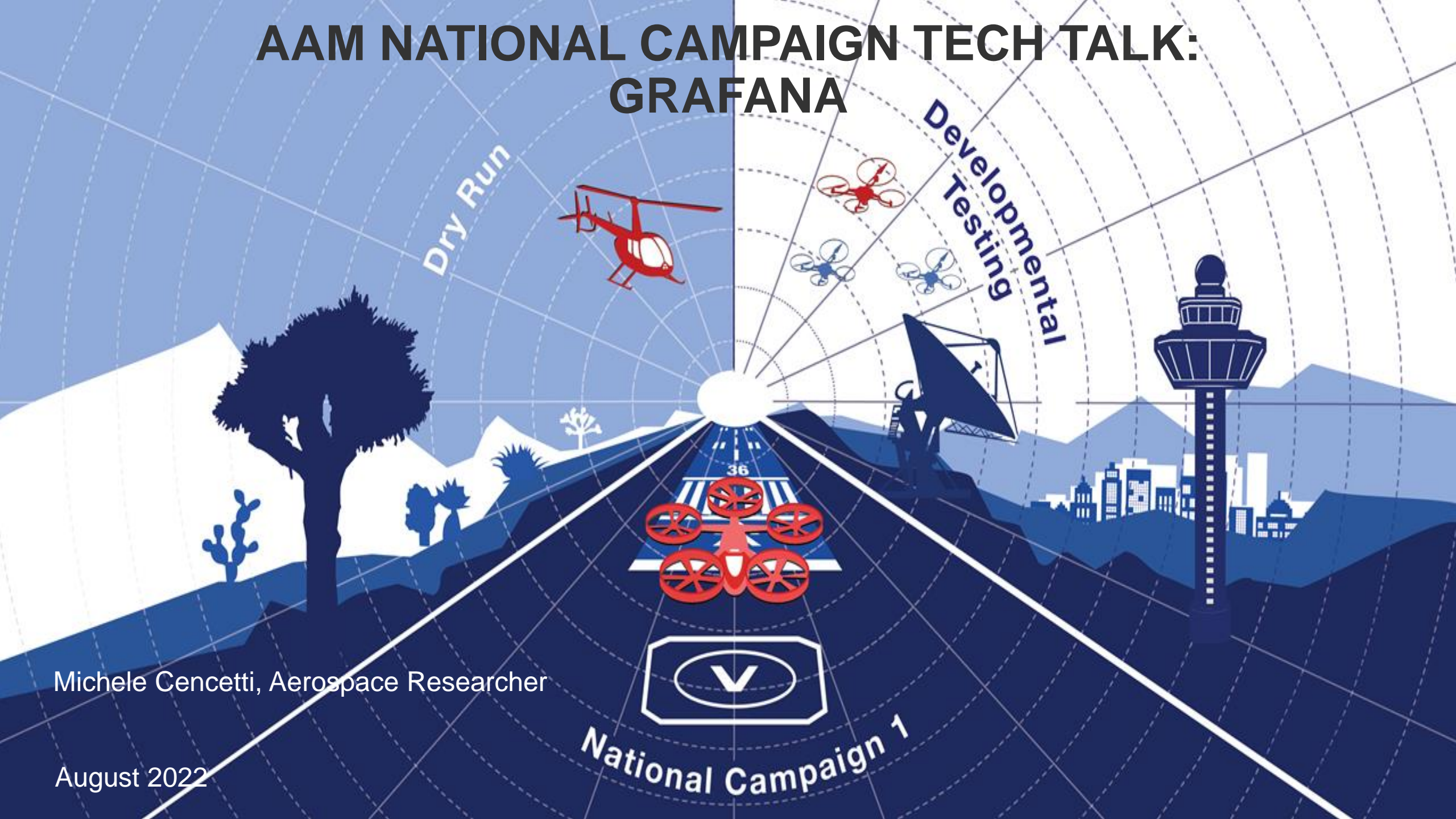


# AAM NATIONAL CAMPAIGN TECH TALK: GRAFANA



Michele Cencetti, Aerospace Researcher

August 2022

National Campaign 1



# Intro to Tech Talks

---

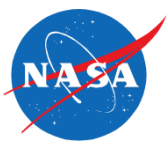
- The purpose of these talks is to engage with the community on types of technologies we are using and developing; there are many more planned
- Ground rules for talk
  - Answers to questions you have may be in upcoming slides
  - It's okay to ask an important question on a slide, but if it can wait then please do so
  - Mute your mike unless you need to talk
  - We'll keep an issues Parking Lot to keep the Tech Talk on point and on time
  - Remember the Tech Talk is being recorded for NASA and its Partners
- Recording. We are recording these Tech Talks and will post online, once approved for external release



# Summary

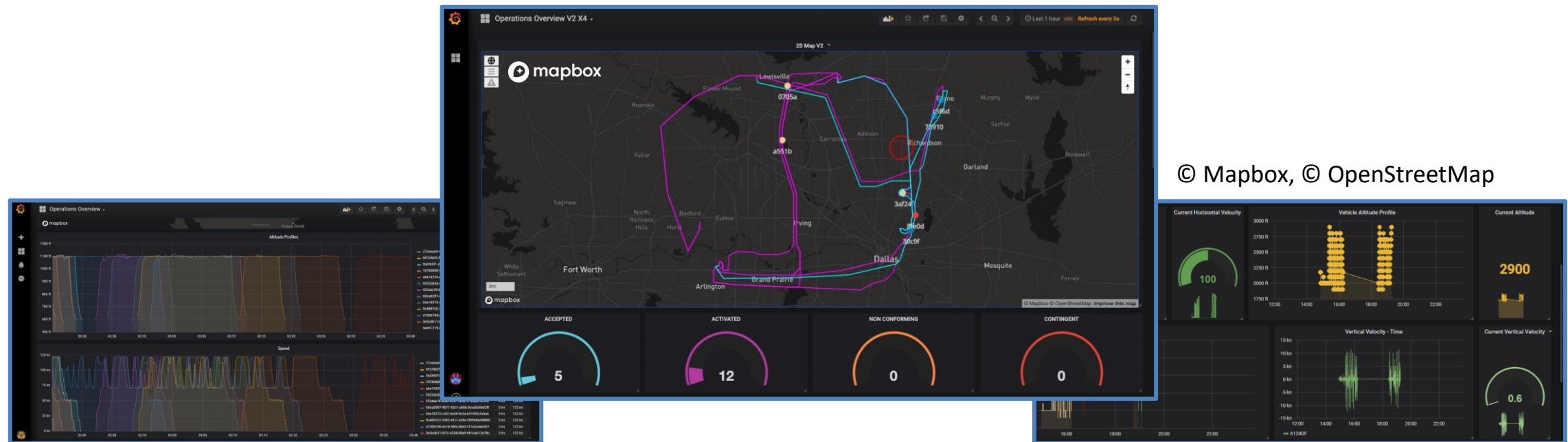
---

- What is Grafana?
  - Experience on Past Projects
  - Current Use Cases on National Campaign
- Grafana High Level Architecture
- Grafana Backend Layer
- Grafana Frontend Layer



# What is Grafana?

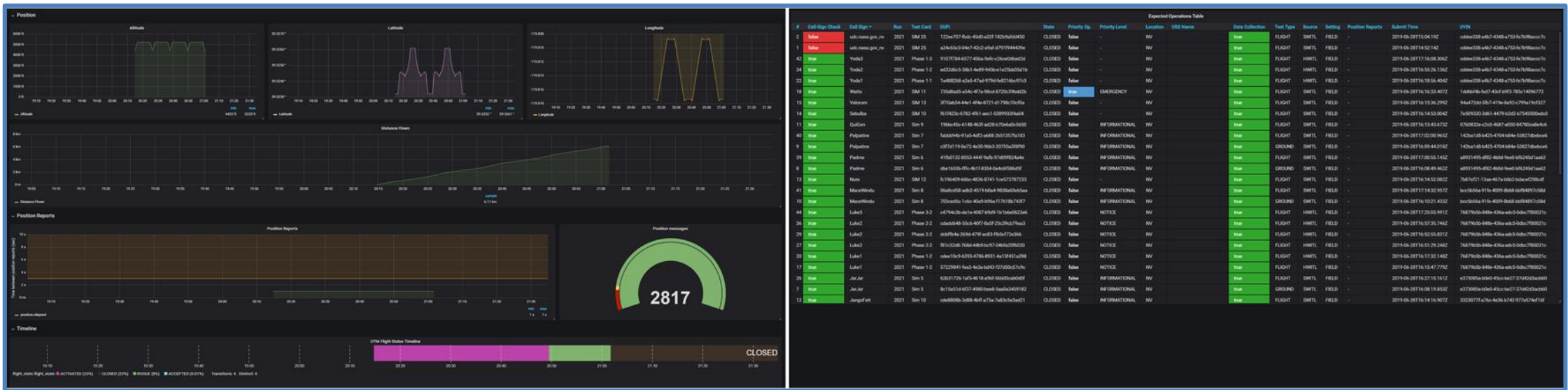
- Grafana® is a multi-platform web application designed to support **analytics** and **interactive visualization** (Microsoft Windows, Linux, macOS). Developed by Grafana Labs and headquartered in New York (<https://grafana.com/>)
- It provides charts, graphs, and alerts when connected to supported data sources





# What is Grafana?

- Experience on Past Projects
  - Grafana has been mostly used on **data monitoring** (consistency of submitted information) and flight tests **real-time visualization** (UTM)
  - **Custom dashboards** providing a real-time feedback in the AOL lab (Video Wall and Lab TV monitors)



Grafana Dashboard Examples: Single Flight Telemetry and Multiple Operations Table Data





# What is Grafana?

- Current Use Cases on National Campaign
  - Data visualization for both **real-time feed** and post-event analyses
  - **Back-end layer** has been extended to bridge the available data source pipelines with the required API endpoints for Grafana service
  - Custom plugins have been developed and added to the **front-end layer**, supporting the customization of dashboard layouts
- Customers
  - NASA Partners
  - ATMX (UAM SubProject)
  - Future NASA Projects



Credits: NASA



# What is Grafana?

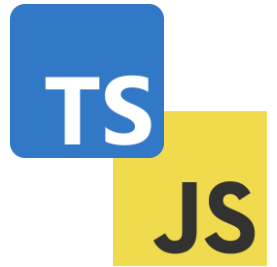
---

- Grafana framework is based on the definition of an **expandable** plug-in system
  - Built-in plugins (official release for database services, mostly integrating some of the most used data infrastructure)
  - Custom plugins (implementation of custom units based on the available resources or desired features)
- The main features of Grafana Framework are its **modularity** and **reusability**. The end users have the possibility to access a set of capabilities to **customize** dashboards layouts (visualization panels, interactive query builders, etc.)



# What is Grafana?

- It is an Opensource project (Apache License v2) and its available in two options:
  - Self-hosted installation or
  - An account on the Grafana Labs cloud service
- Grafana consists of two main parts: a front-end layer and a back-end layer
  - Front-end layer is written in **TypeScript, JavaScript**. It includes the plugins system that is used to visualize data and setup the data source connection and integration
  - Back-end layer is written in **Go**. It includes all the function needed to run the Grafana service and plugin management (framework layout, etc.)
- Plug-in system is divided into the following elements:
  - **Panel** plugins (frontend elements)
  - **Datasource** plugins (backend elements)
  - **App** plugins (bundle data source and panel plugins)

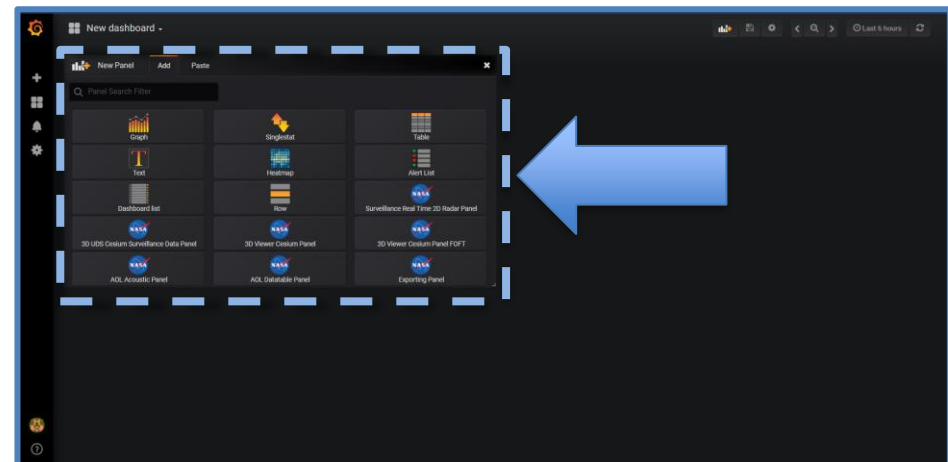






# What is Grafana?

- As the End User accesses Grafana service different features are available based on the corresponding role (admin, editor or viewer)
- A user management system is integrated with the admin role (user creation, teams' organization, dashboards permissions, etc.)
- Data sources and Dashboards creation
  - Panel **drag and drop** mechanism (one panel represents one instance of a plugin)
  - Panel **configuration** (options and data source linking)
  - Panels **resizing** and **positioning**

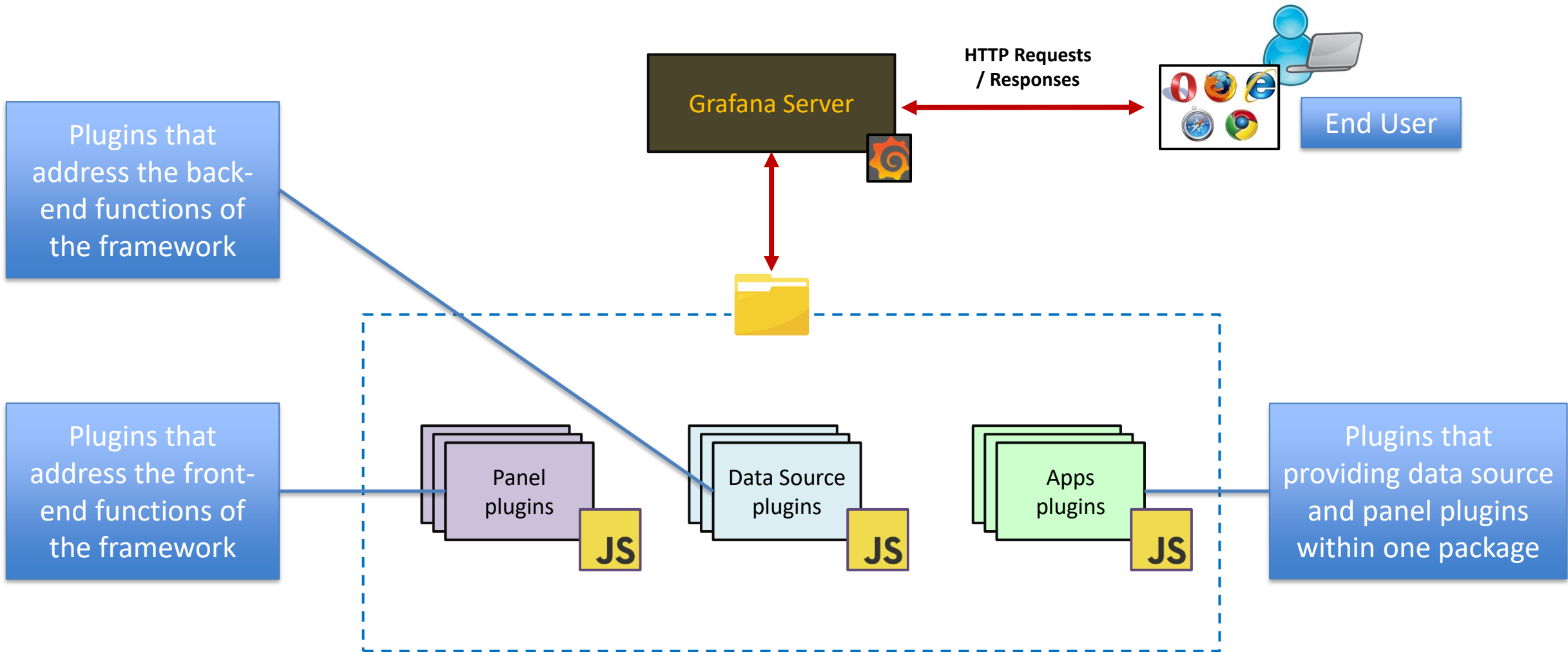


Grafana Dashboard: Panel Creation Example



# What is Grafana?

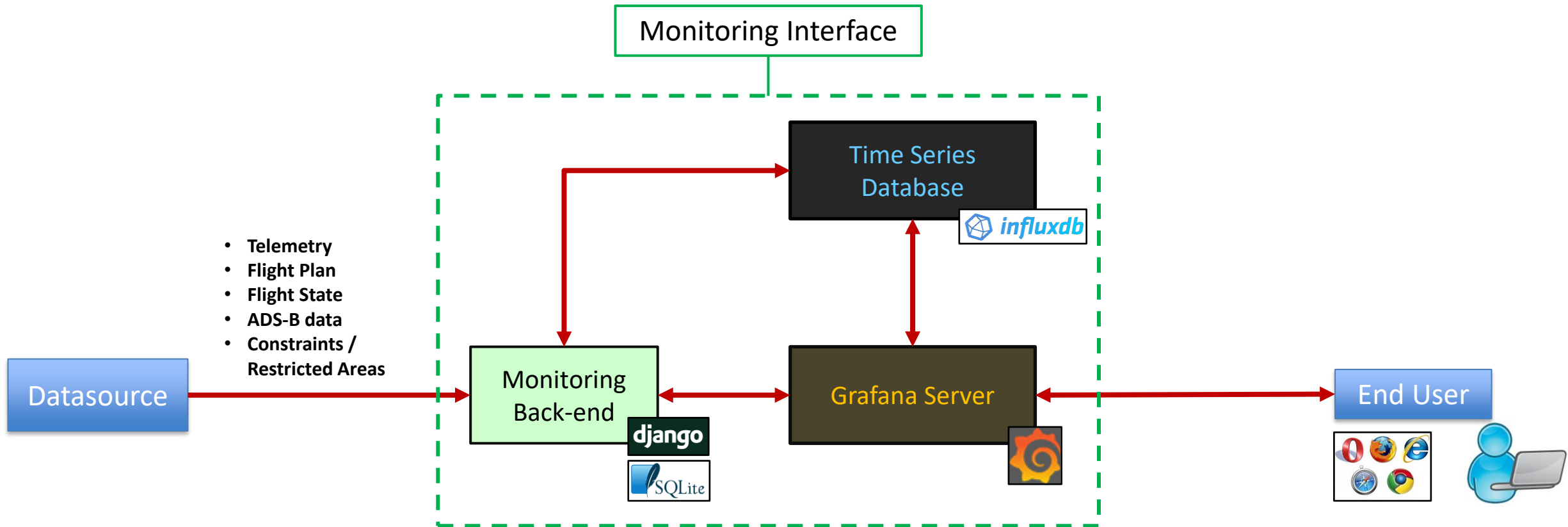
- Grafana Framework Overview

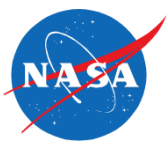




# AAM NC High Level Architecture – Grafana Backend Layer

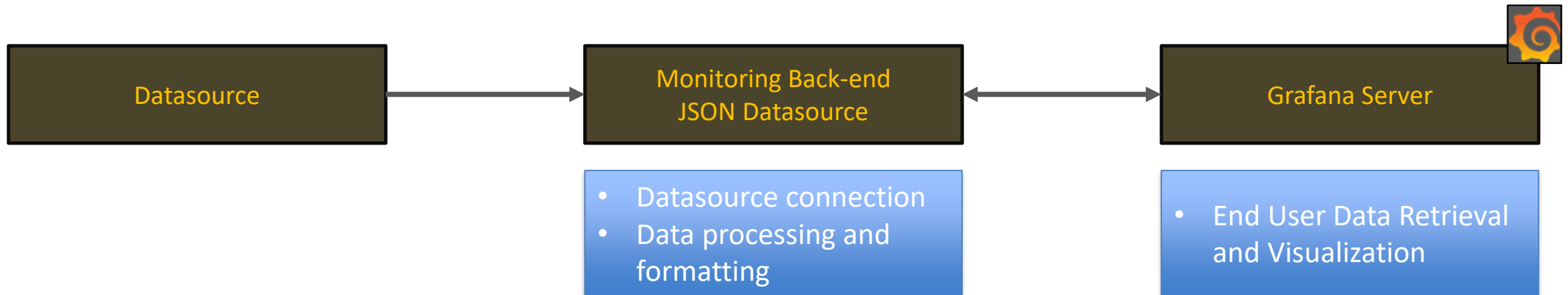
- Overview of the Monitoring Interface backend





# AAM NC High Level Architecture – Grafana Backend Layer

- Custom Data Sources
  - JSON Datasource - Web service with dedicated endpoints (Python Django, Java Spring Boot, C++ WebSocket server with Boost, etc.)
- Built-in Official Supported Data Sources:
  - Relational Databases: **MySQL**, Microsoft SQL Server (MSSQL), **PostgreSQL**, Azure Monitor, AWS CloudWatch, Google Cloud Monitoring
  - NoSQL and Timeseries Databases: Elasticsearch, Loki, **InfluxDB**, Graphite, OpenTSDB, Prometheus, Jaeger, Zipkin, Tempo





# AAM NC High Level Architecture – Grafana Backend Layer

- Data sources: WebSocket data service and Amazon Kinesis Stream (National Campaign AT1 Pipeline)
  - **Python/Django** AT1 Client (JSON Datasource plugin)
  - **AWS Kinesis Stream Consumer**, Java Client (Kinesis Client Library V2) which is used to connect to the real-time data stream

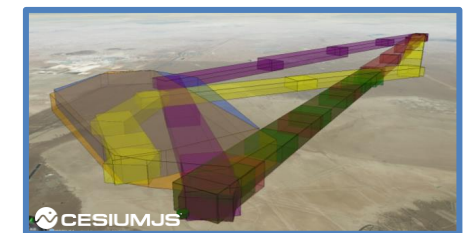
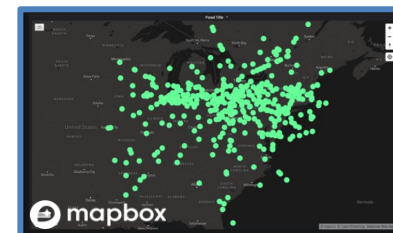
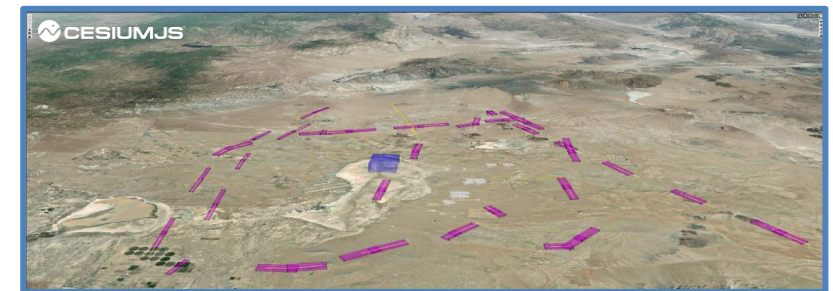
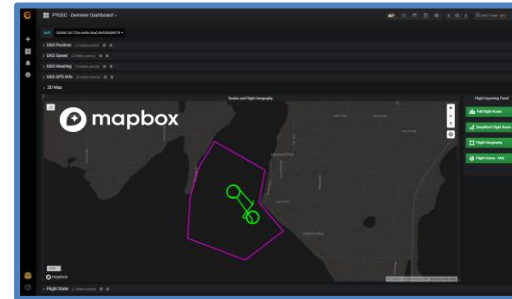


- Data Models
  - **Telemetry** Messages (position data, etc.)
  - **Operations** Messages (geometry of the flight plan/trajectory, flight state, etc.)
  - **ADS-B** Messages (position data, emitter type, etc.)
  - **Constraints/Restricted Volumes** Messages (geometry, etc.)



# AAM NC High Level Architecture – Grafana Frontend Layer

- Dashboards have been created using built-in plugins (single value panels, time series panels) and custom plugins (developing the desired features not available from the built-in elements)
- Custom Panel Plugins
  - Scatter Plots Panel
  - 2D Map Panel (Mapbox JavaScript, © Mapbox, © OpenStreetMap)
  - 3D Map Panel (Cesium JavaScript library)
  - Table Data Panel
  - Export/Data Control Panel
  - Radargram Panel (Plotly JavaScript library)

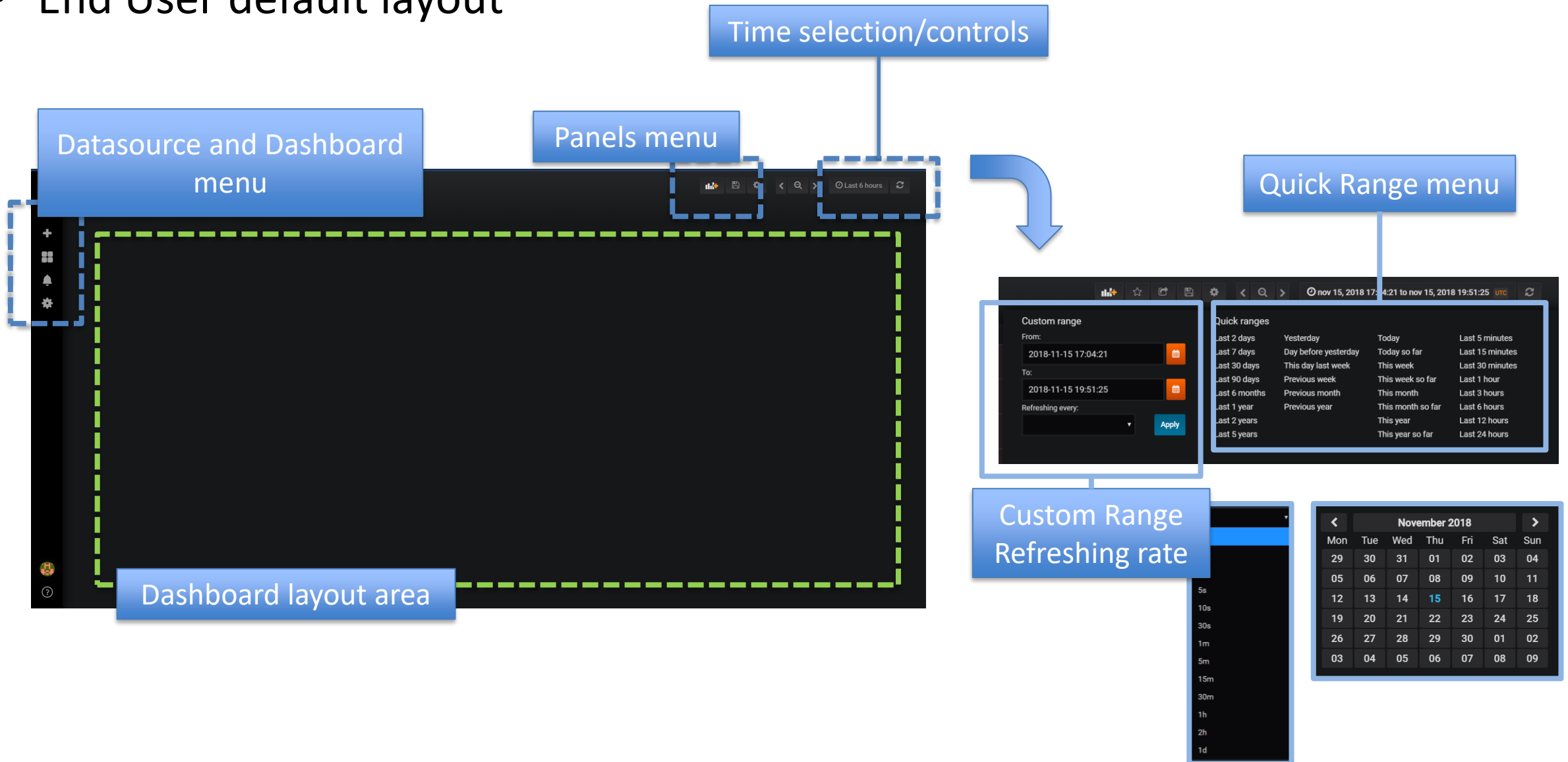


Grafana Dashboard Examples: 2D and 3D Map Panels Visualizing Flight Plans and Positions



# AAM NC High Level Architecture – Grafana Frontend Layer

- End User default layout

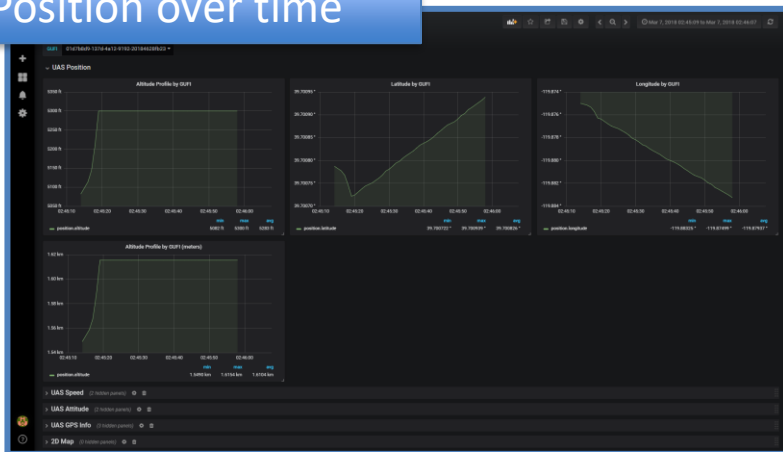




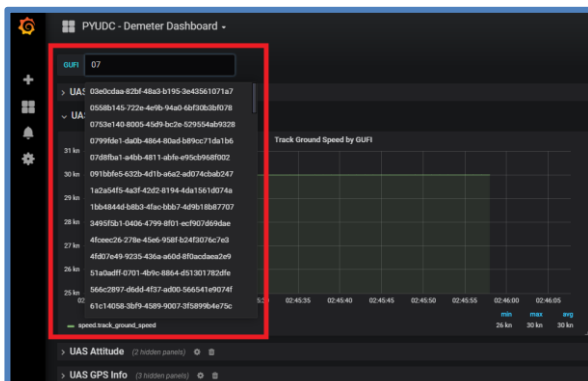
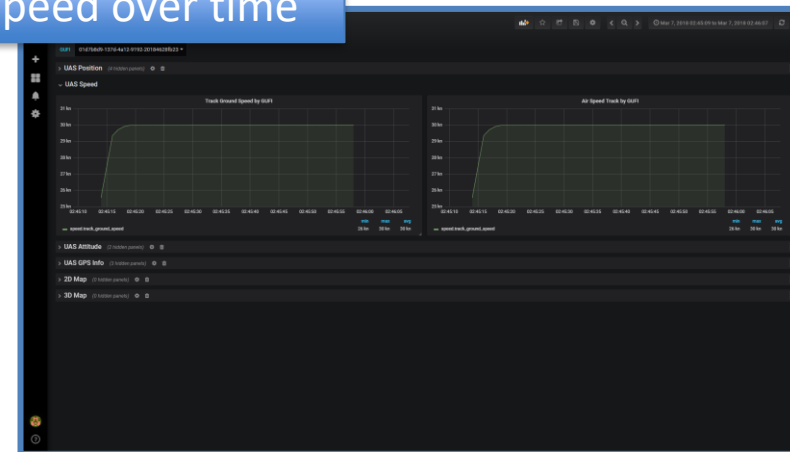
# AAM NC High Level Architecture – Grafana Frontend Layer

- Telemetry Data dashboard
  - Positions, speed, position update latency, etc.

Position over time



Speed over time



Flights filtered by Operation ID

```
Data Source: PYUDC
FROM default position WHERE gulf -- /Scur_gulf/ +
SELECT field (altitude) +
GROUP BY +
FORMAT AS Time series
ALIAS BY Naming pattern
Add Query
```

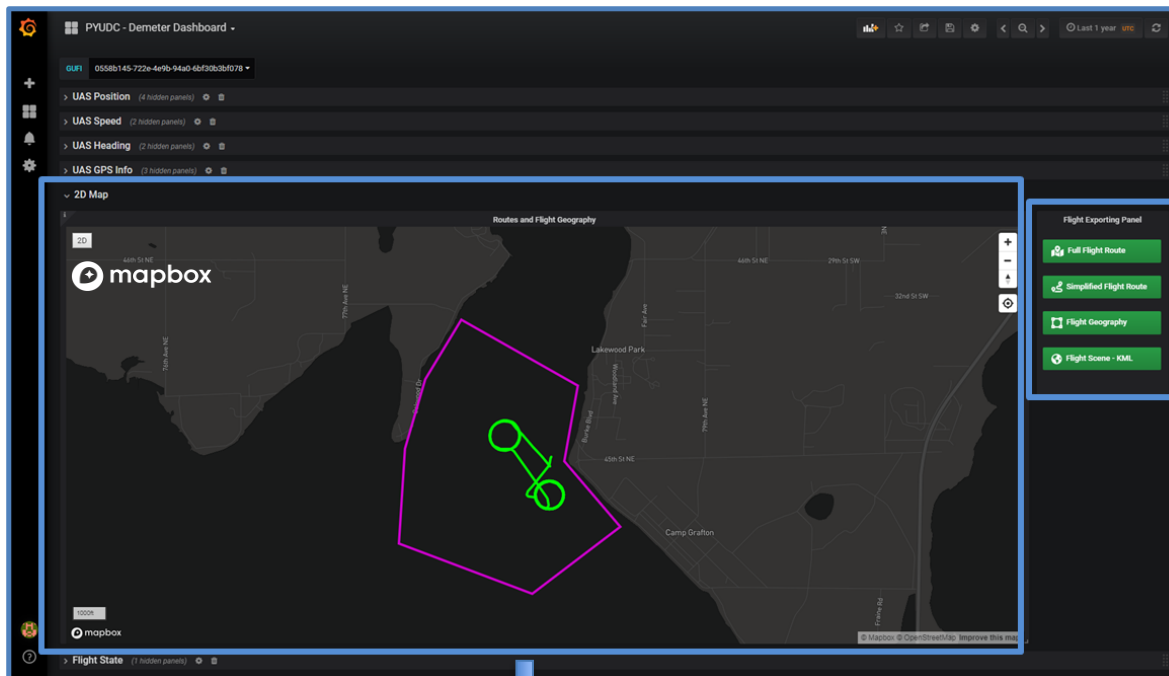
Database Queries Editor – (e.g., flight distance)





# AAM NC High Level Architecture – Grafana Frontend Layer

- Operations dashboard (2D map)
  - Flight volume geometry, flight trajectory



Exporting resources:

- Full route - KML
- Simplified Route - KML
- Flight Geography - JSON
- Flight Scene - KML

2D Map flight  
geography and route

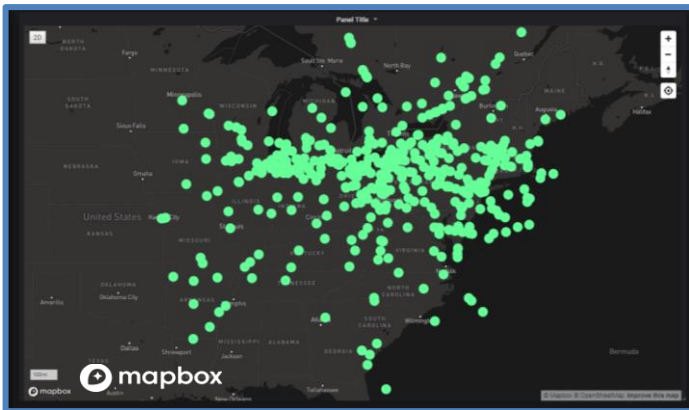
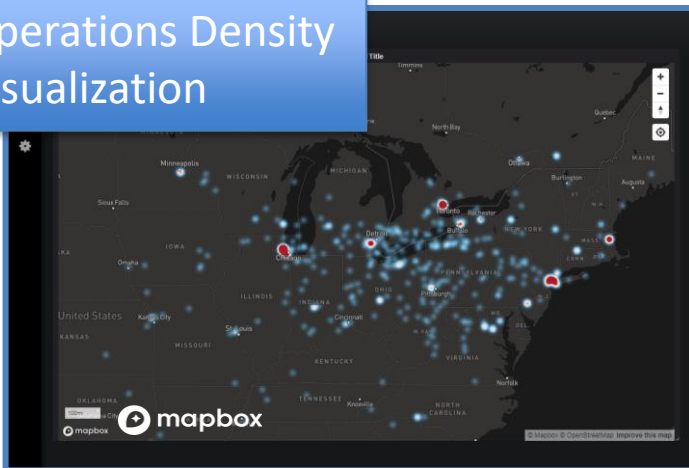
© Mapbox, © OpenStreetMap



# AAM NC High Level Architecture – Grafana Frontend Layer

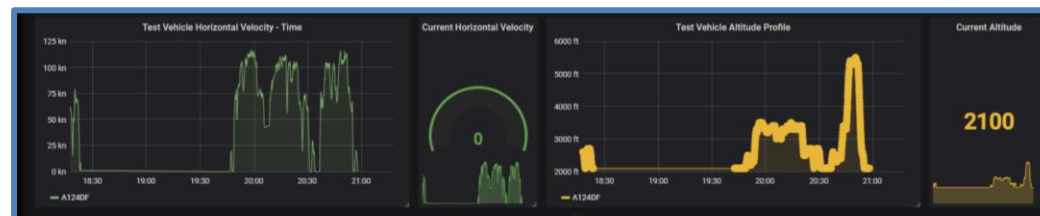
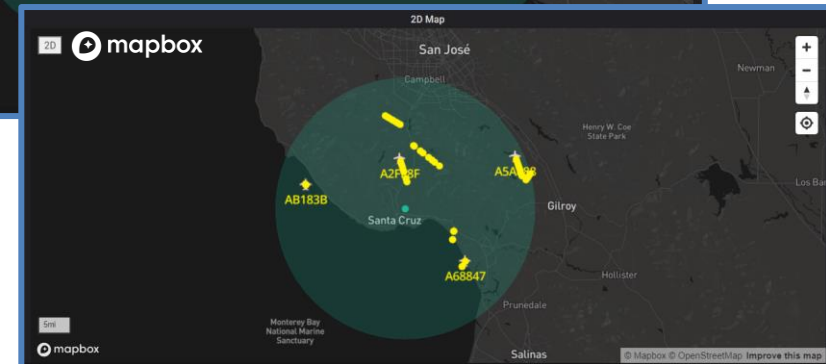
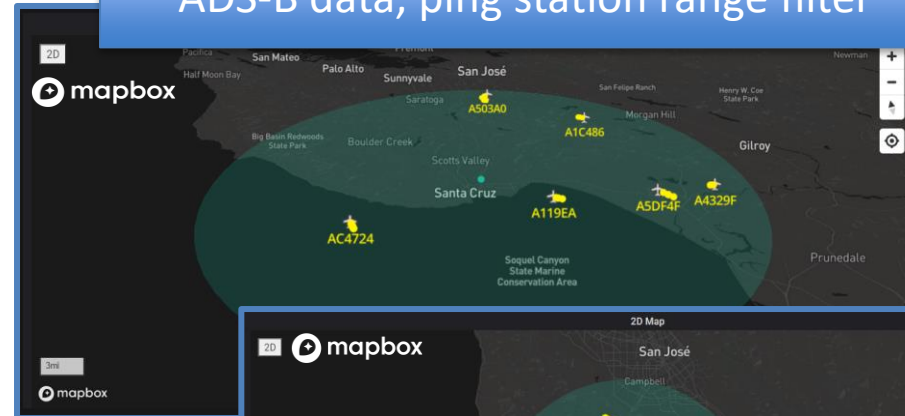
- Operations Dashboard (2D Map)
  - Heat-maps, vehicles position, ads-b data

Operations Density Visualization



© Mapbox, © OpenStreetMap

ADS-B data, ping station range filter





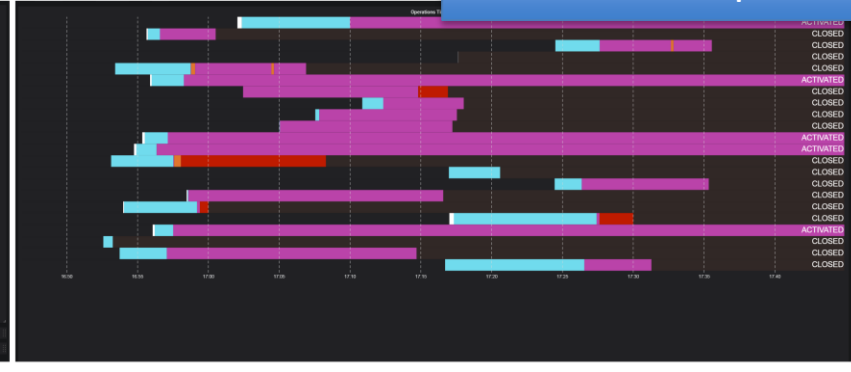
# AAM NC High Level Architecture – Grafana Frontend Layer

- Multiple operations dashboards
  - Flight states, altitude profiles, positions reports count, operations tables

Altitude profiles panel



Timeline panel

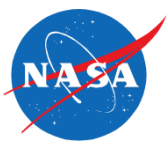


Single Operation selection



Operations table panel

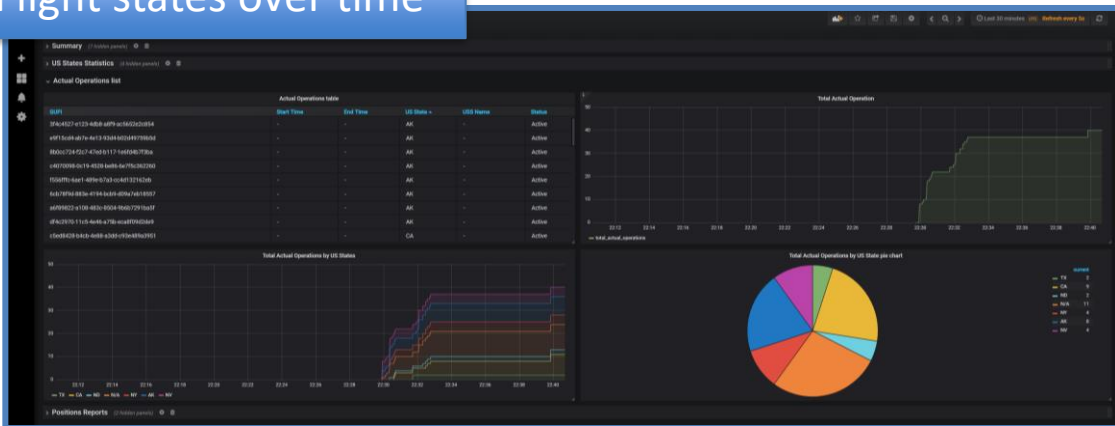
ID	Name	Status	Priority	Location
1	Star	AV	EMERGENCY	LA
2	Star	AV	AV	LA
3	Star	AV	AV	LA
4	Star	AV	AV	LA
5	Star	AV	AV	LA
6	Star	AV	AV	LA
7	Star	AV	AV	LA
8	Star	AV	AV	LA
9	Star	AV	AV	LA
10	Star	AV	AV	LA
11	Star	AV	AV	LA
12	Star	AV	AV	LA
13	Star	AV	AV	LA
14	Star	AV	AV	LA
15	Star	AV	AV	LA
16	Star	AV	AV	LA
17	Star	AV	AV	LA
18	Star	AV	AV	LA



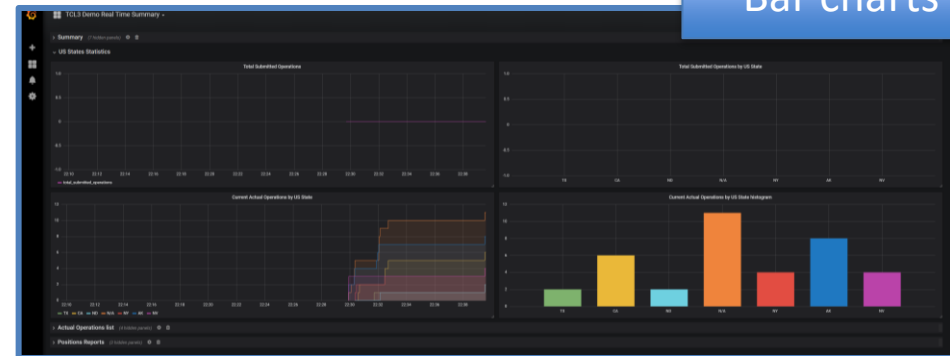
# AAM NC High Level Architecture – Grafana Frontend Layer

- Multiple operations dashboards
  - Flight states over time, bar charts, pie charts, 2D map bubble charts

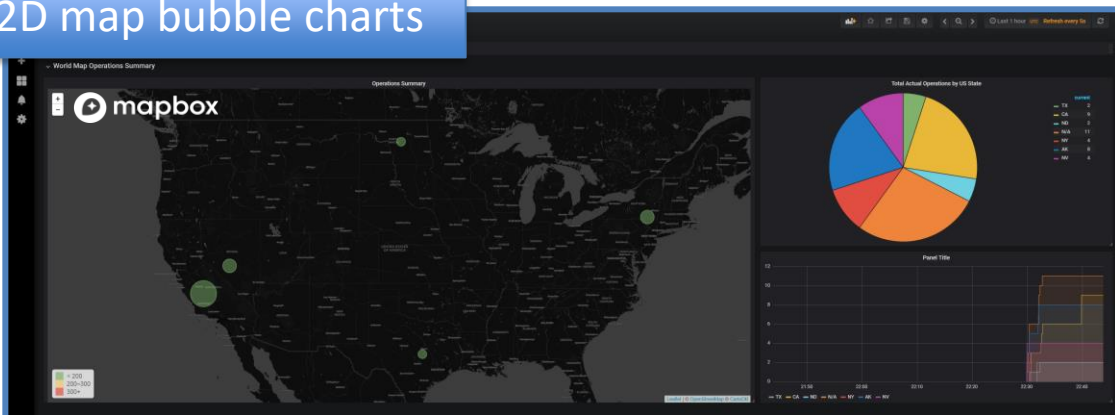
Flight states over time



Bar charts

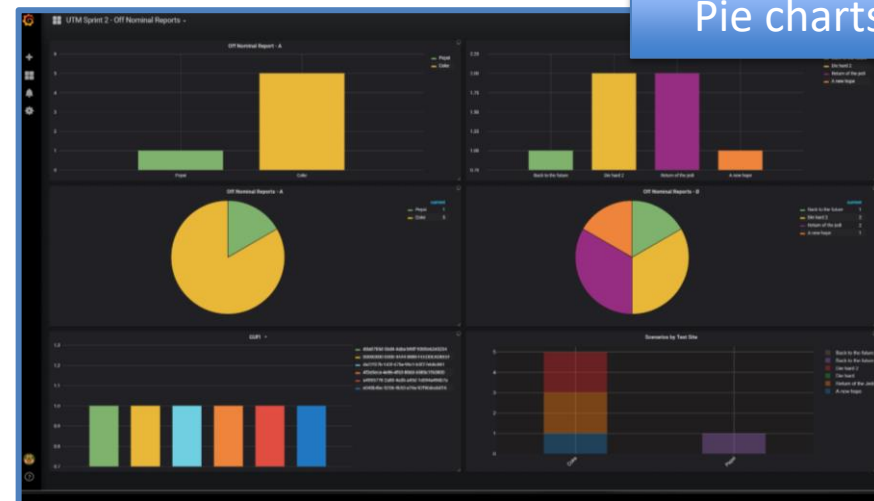


2D map bubble charts



© Mapbox, © OpenStreetMap

Pie charts

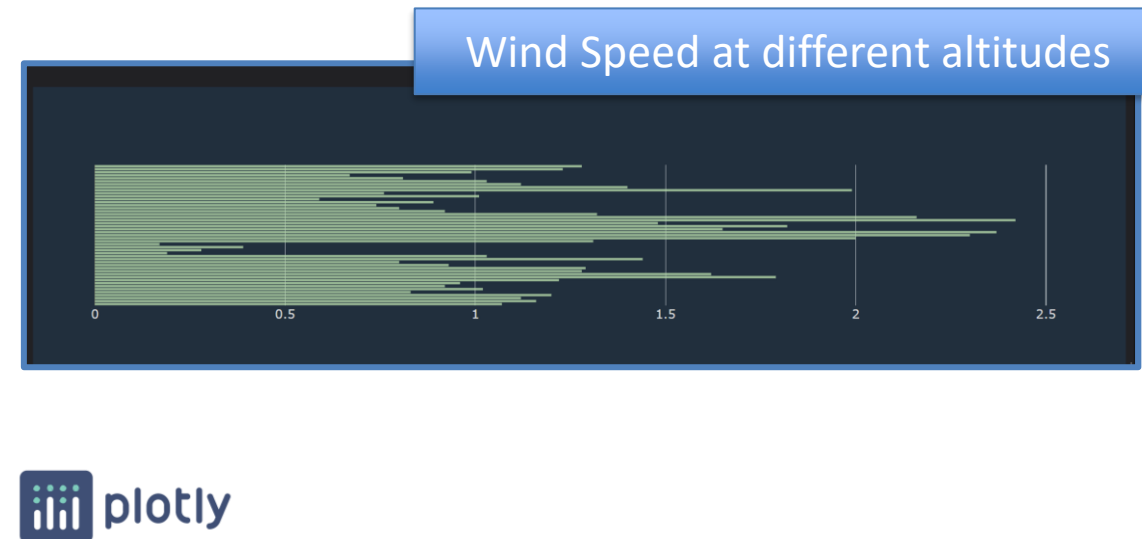
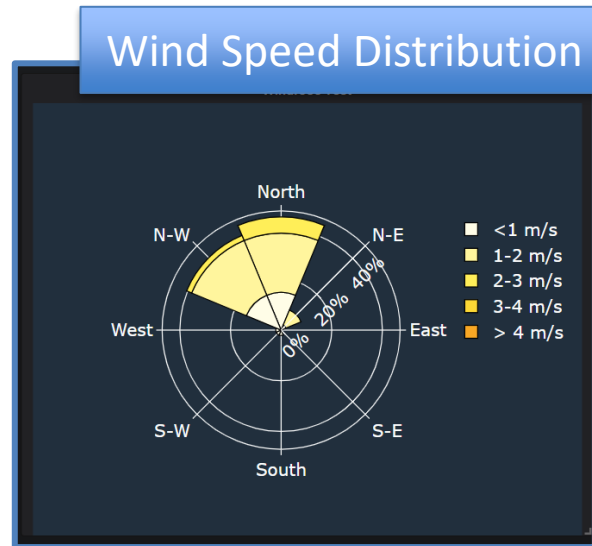
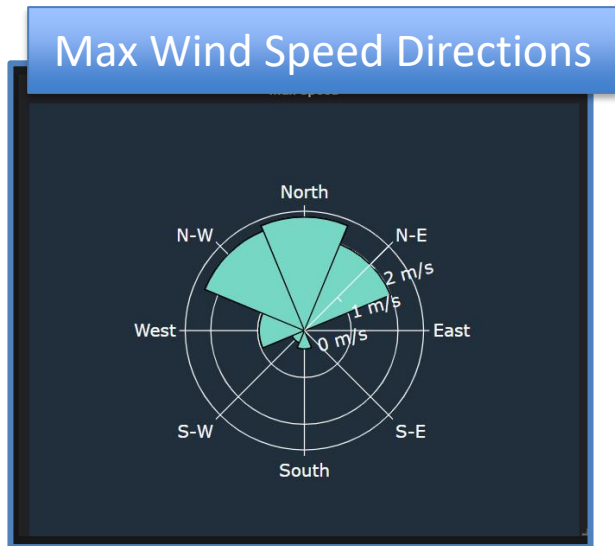




# AAM NC High Level Architecture – Grafana Frontend Layer

- Weather data visualization
  - Preliminary assessment of the data models that can be considered for weather visualization (wind data model, wind speed and direction, temperature, visibility, etc.)
  - Different options have been considered for the available data types (horizontal bar charts, wind-rose charts, polar charts, 3D panel, etc.)

Example panels of simulated data:

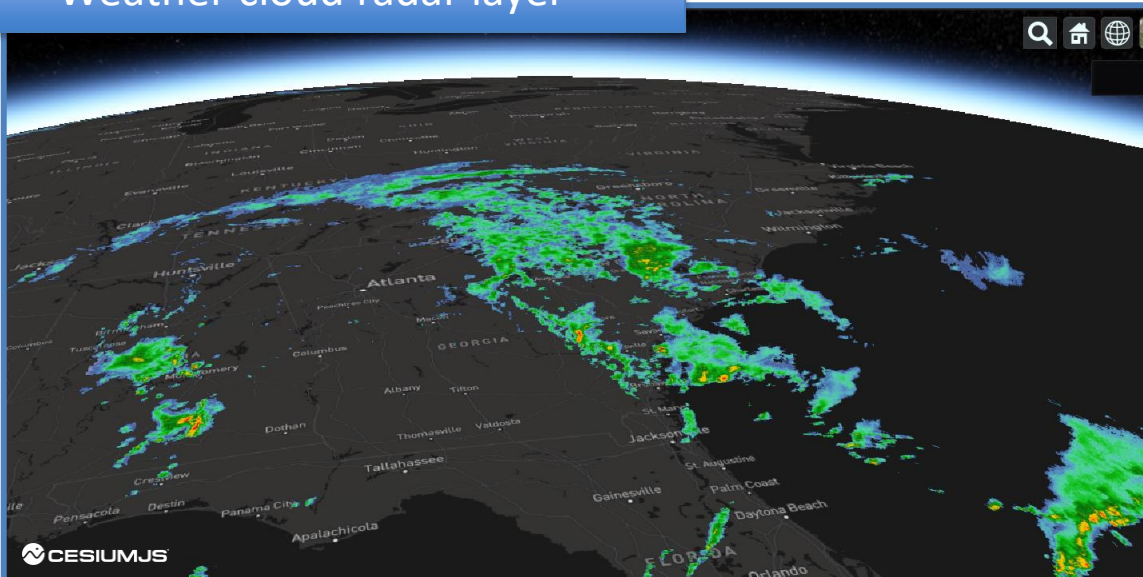




# AAM NC High Level Architecture – Grafana Frontend Layer

- Weather data visualization
  - 3D environment (wind speed and direction, cloud radar layer, etc.)

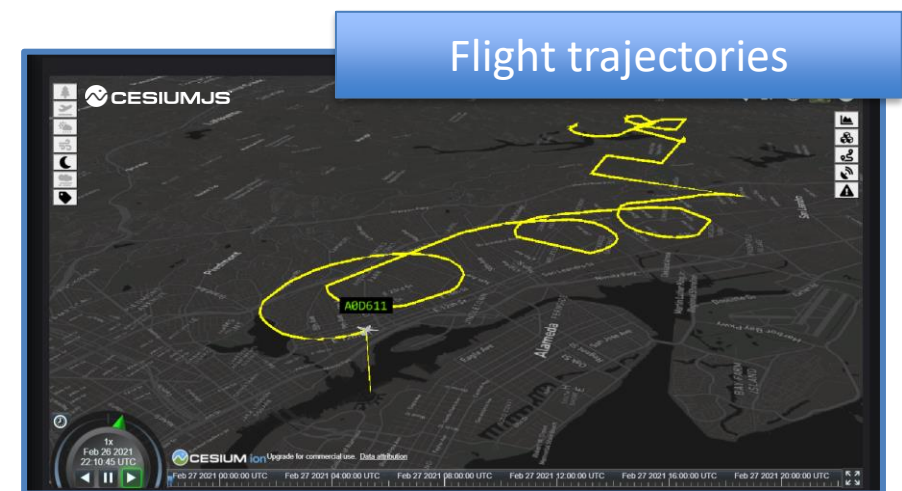
Weather cloud radar layer





# AAM NC High Level Architecture – Grafana Frontend Layer

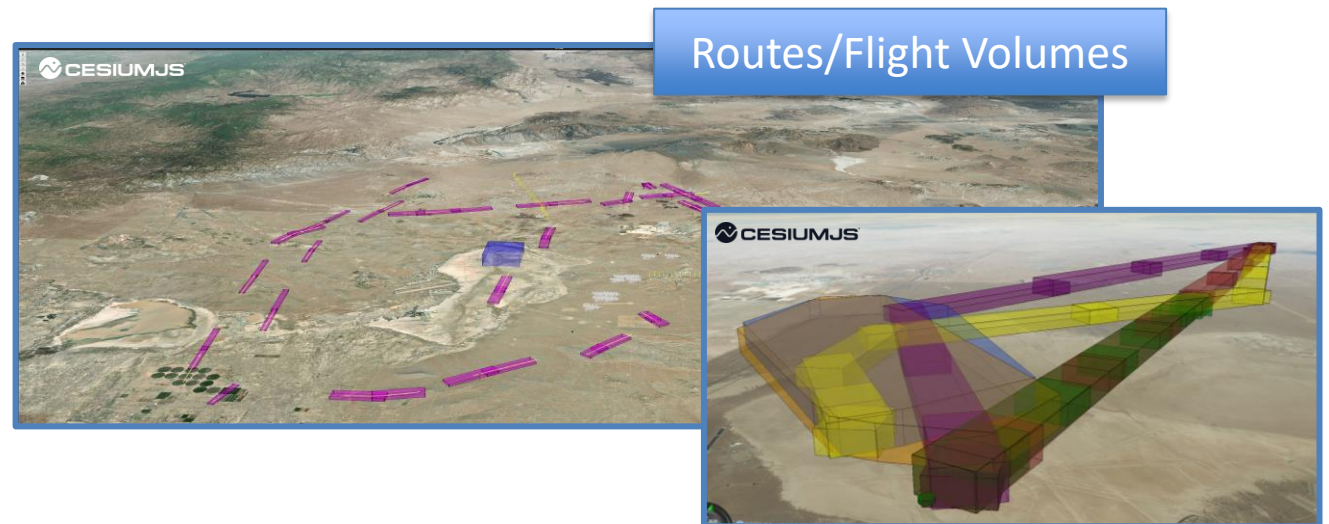
- 3D Map Panel (Cesium JavaScript library). A 3D Cesium scene has been developed as a panel plugin. The 3D scene is loaded inside a panel, but it can be reused in separate dashboards. The current implementation integrates the following graphical features:
  - **Lines/Polylines (static and dynamic)**: Vehicle trajectories, position projections on the terrain, etc.
  - **Surfaces (static and dynamic)**: Airspace boundaries projection, national parks terrain highlight, etc.





# AAM NC High Level Architecture – Grafana Frontend Layer

- **Volumes (static and dynamic)**: Flight volumes, routes corridors, volume reservations, etc.
- **Primitives/Geometric Shapes (static and dynamic)**: Simplified Aircraft model placeholder, surveillance data placeholder, etc.
- **3D GLTF Model (KML/KMZ data importer is also available as native feature with the library)**: 3D Aircraft model, simplified buildings models, etc.
- **Image Layers (static and dynamic)**: Weather information image layer, etc.

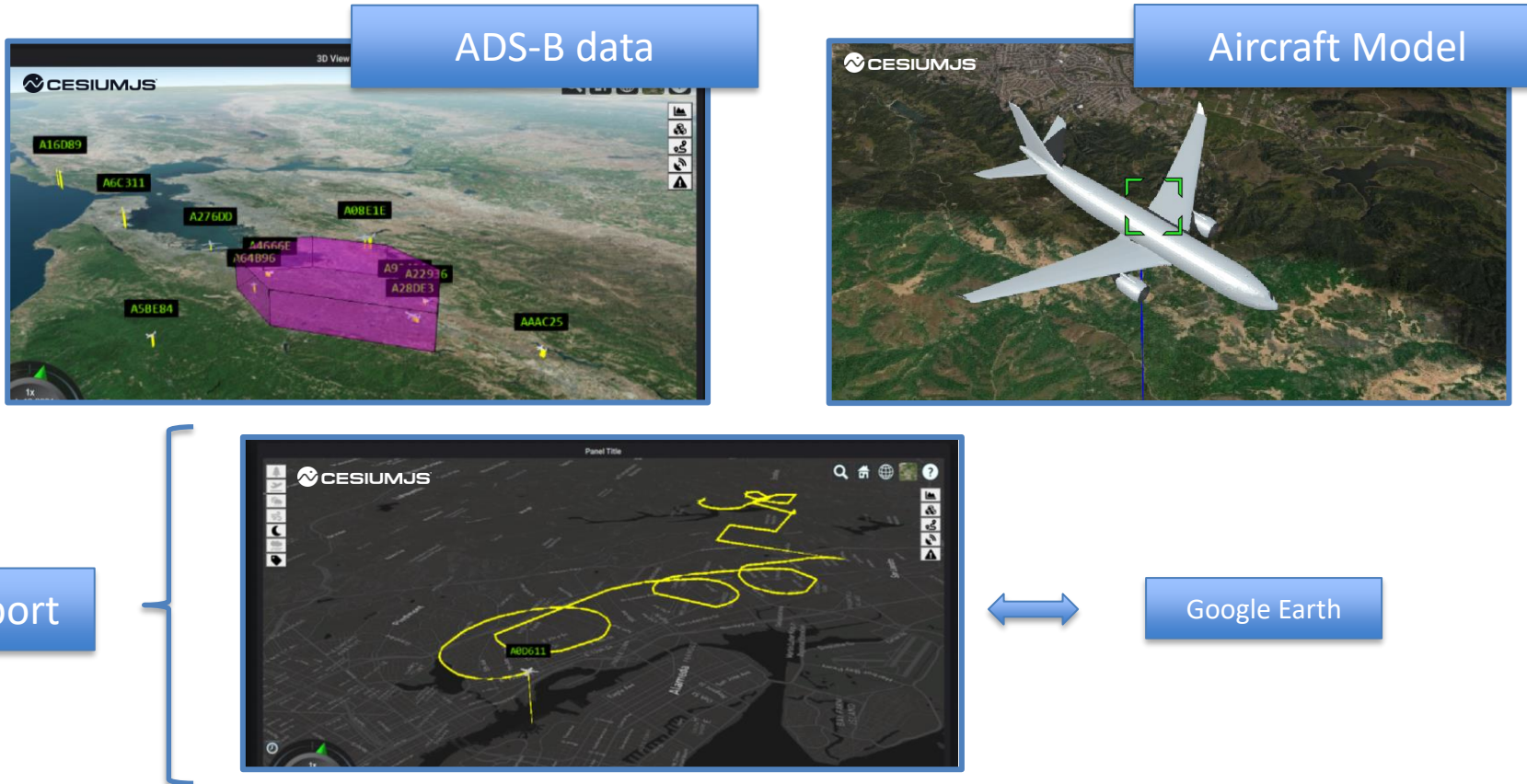






# AAM NC High Level Architecture – Grafana Frontend Layer

- 3D Map Panel (Cesium JavaScript library)
  - ADS-B data, vehicles 3D models
  - Data export (flight geometry, KML/text format)





# AAM NC High Level Architecture – Grafana Frontend Layer

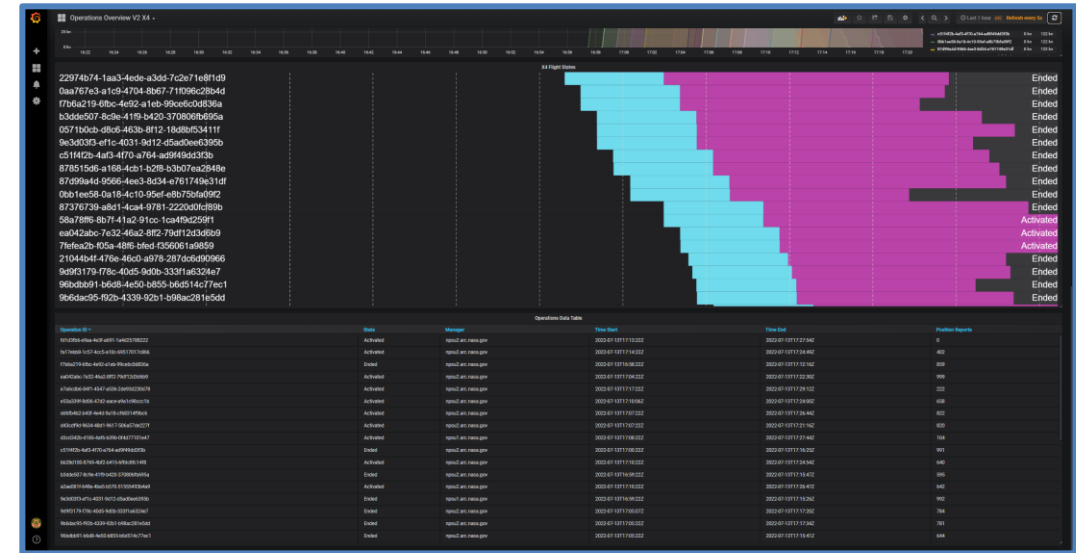
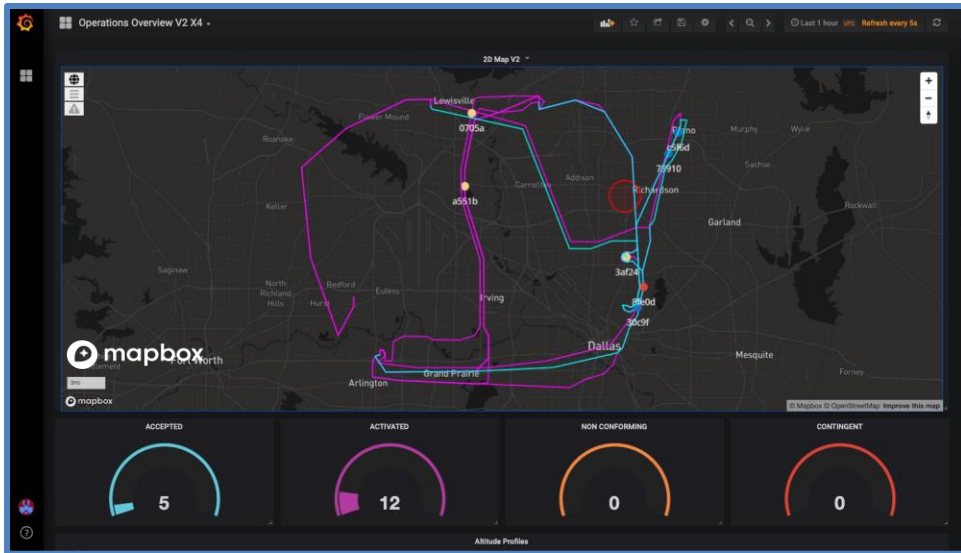
- Dashboard examples (3D flight plans/volumes, altitude profiles, flight states counts)



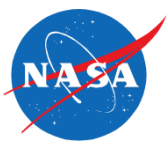


# AAM NC High Level Architecture – Grafana Frontend Layer

- Dashboard examples (2D flight plans, flight states timeline, altitude profiles)



© Mapbox, © OpenStreetMap



# Security in the Data Pipeline

- Most powerful feature is the ability to funnel data from hundreds of different data sources (i.e., services or databases) into a single dashboard without migrating the data from where it lives
- Grafana provides different ways to authenticate users (Basic Authentication, OAuth login, LDAP, SAML). A set of supported authentication providers is also available (Google OAuth, GitLab OAuth, GitHub OAuth, etc.)
  - The current Grafana service we are using implements the Basic Authentication. The service can be accessed only through the NASA network with a username and password and it's not public exposed
- All users are assigned one of four standard roles: the **Grafana Admin, Admin, Editor, and Viewer**
  - The main difference between the Editor and Viewer role is represented by the capability to modify the graphical layout of dashboards as well as the configuration of each single panel
- Encryption
  - The communication between the end users and the Grafana uses the HTTP protocol
  - The communication between Grafana and its data sources is not currently encrypted. Data are provided from the ATI pipeline and the communication is established through the AWS Kinesis Stream authentication mechanism (AWS Access Keys for programmatically access as AWS Kinesis Consumer)

# Questions



# APPENDIX



# References

---

- <https://grafana.com/blog/2022/02/22/how-secure-is-your-grafana-instance-what-you-need-to-know/> .
- 2D maps use data from © Mapbox and © OpenStreetMap and their data sources. To learn more, visit <https://www.mapbox.com/about/maps/> and <http://www.openstreetmap.org/copyright>.
- For more information about the Cesium platform (© Cesium GS) visit <https://cesium.com/>.
- For more information about Google Earth (Freeware) visit <https://earth.google.com/web/>.
- For more information about the Plotly JavaScript library (© Plotly) visit <https://plotly.com/javascript/>.