Unite Aware

CommsNet / TRK_UHF

Release Notes

May 15, 2026

1. *Enhancements*

1.1. *List of Implemented Enhancements*

Reference ID	Description
Requirement 2239	Categorization of UNDOF devices
Requirement 2238	Add live monitoring of the crawlers
Requirement 2227	Detect GPS data spoofing in tracking
Requirement 2255	New endpoint to retrieve user permissions

- **Requirement 2239**

Change Request

UNDOF has provided us with the document categorizing its devices,We need to modify the Mototurbo crawler to incorporate this categorization.

Solution

We will modify the Mototrubo crawler code and its configuration to add UNDOF categorization when we receive their data

- **Requirement 2238**

Change Request

We want to be able to track the uptime and downtime for each crawler, so we need to change the way we store the status of each crawler

Solution

We will need to store a history of status changes for each module in a new collection (CommsNetMonitorLive)

Move the monitoring module to the common class and modify the code so that any change in the crawler's status is saved to the new live collection

- **Requirement 2227**

Change Request

In some missions, certain radio stations transmit a tracking point that is inaccurate and deviates from the actual data. An algorithm must be used to detect these points and flag them in the responses provided by the CommsNet API so that users of this data are aware of the falsified data.

Solution

In the Mototurbo crawler, for UNDOF, we analyze the incoming radio position and determine whether it is unusual. To do this, we compare the incoming value with the previous one, apply the Haverstein algorithm, and calculate whether the point is feasible in terms of time and distance. If it is incorrect, we mark it as UNUSUAL in the accuracy field; if it is correct, we leave it blank.

- **Requirement 2255**

Change Request

A new endpoint is needed to display, for each user, the Commsnet layers they can view based on their permissions.

Solution

We'll create a new endpoint, /user/[Mission], in the API that retrieves the groups to which the user belongs from the Azure token, intersects them with the groups allowed for the mission, and displays the results in JSON, indicating which layers the user has access to and which they do not.