

FLEET OPERATIONS CONSOLE



A QUICK OVERVIEW

The operator console for the fleet · r2d2.office.ilab.zone

ORIGINATOR REF **R2D2-OV-001_B**
EDITION Edition A · Version 1
DATE May 2026

Originator: iLab / TaskForce-X. Releasable to coalition partners participating in TaskForce-X.

STANAG 4817 / AEP-105 · HIBW RC4

Contents

What R2-D2 is	3
A note on the names	3
Where to start	4
The areas	5
Node-RED Fleet — the automation backbone	5
Restreaming — camera feeds & video walls	5
Imp Intel — UXV calibration & evaluation (TaskForce-X)	5
The Force — telemetry & situational awareness	6
Temple Archives — the data layer	6
The Republic — nations & inventory	7
Settings — configuration for every area	7
Droidspeak — the handbook	7
The mission, in one line	8
One paragraph on sign-in	8
Lexicon	8

What R2-D2 is

R2-D2 is **one console for a whole fleet**. Behind it sit a swarm of Node-RED automation instances, a video restreamer, an intelligence-collection workflow, a relational/timeseries store, and a set of embedded map and telemetry apps. The point of R2-D2 is to put all of that behind a single URL — one place to see whether the fleet is healthy, push a flow change, pull a camera snapshot, work a collection requirement, watch a tactical map, or look up the registry of nations and equipment.

Open it at r2d2.office.ilab.zone. The left sidebar is the map of everything below; this page is the legend.

The data backbone is STANAG 4817 / AEP-105 — the CATL HIBW message set, release 0.3.0-rc4 (Draft 3 / RC4). Everything that crosses the fleet as operational data rides one envelope (a [header](#) + [body](#)) in seven message kinds: *NodeDescription* (who I am), *NodeStatus* (where I am / what I'm doing), *TaskAdmin* (assign/update/cancel a task), *TaskFeedback* (progress), *TaskResult* (outcome + data products), *DynamicUpdate* (world-model entities — contacts, tracks, routes), and *Chat*. These move over the MQTT/Zenoh mesh. That standard, not any one app, is what lets R2-D2 speak the same ISR data language a coalition system would. Node-RED is the *automation* backbone; STANAG 4817 RC4 is the *data* backbone every plane reads from and writes to. **Scope:** in **TaskForce-X** sessions we exercise this standard *passively* — for UXV testing, calibration, and evaluation — **not** to task nations or live sensor platforms. R2-D2 does not hold tasking authorization.

This is an intro, not a status board. It explains what each area *is* and where to start — it deliberately carries no live counts (instances online, feeds up, embeds ready). Those numbers live in the app and change minute to minute. For current status, open the **Overview** page.

A note on the names

The dashboard speaks Star Wars. The vocabulary is internal flavour, but it is *load-bearing* — route paths and table names use it, so it is worth learning. Every name maps to a plain concept. When in doubt, this is the decoder:

You'll see...	It means...
R2-D2	the console itself (this dashboard)
The Force	live telemetry & situational-awareness apps, embedded in-app
Holocron	a 4817 broker baseline viewer — verify the data before the COP
HoloNet	an MQTT topic browser (“the whispers of the system”)
Holoprojector	a video wall — arranges feeds into projection layouts
HoloVids	the camera-feed grid + live video wall
Rift Gates	the MQTT message brokers (TBMQ) that link fleets and field units
Probe	robot/machine introspection (ROS2 / Foxglove)
Force Sense	a read-only Kubernetes cluster + stack health view
Force Visions	replay past UXV history — recorded sessions and views played back
The Alliance	the unmanned-vehicle (UXV) fleet registry
Imp Intel	passive UXV calibration & evaluation (not live nation tasking)
Medusa (Hydroid)	the contact-fusion engine — fuses UXV readings for the COP (planned)
TaskForce-X	our test/eval sessions — UXV runs and capture, not live ops
Temple Archives	the data archive — feeds, reference catalogs, schemas
The Republic	member nations and their equipment inventory
ShopKeepers	the vendor / make-and-model catalog
Galaxy Map	a clickable network graph of every tool in R2-D2
Droidspeak	the in-app handbook — docs, install guides, live API reference
HIBW / 4817	STANAG 4817 / AEP-105 CATL HIBW — the ISR data wire standard (rc4)

Where to start

Three pages orient you before anything else:

- **Overview** (/) — cross-system KPIs and health for Node-RED, Restreaming, and the Force, on one screen. This is your “is everything OK?” page.
- **Galaxy Map** (/galaxy-map) — a live graph of every product and tool. Click a node to jump straight to it. Use it when you know *what* you want but not *where* it lives.
- **Droidspeak** (/droidspeak) — the in-app handbook: deep-dive docs, install guides, and a live API reference. New to the ecosystem? Skim it first, then come back to the areas below.

Everything else is grouped in the sidebar by area. The rest of this page walks those areas in order.

The areas

Node-RED Fleet — the automation backbone

The original core of R2-D2: a fleet of Node-RED instances, each running deployed *flows* (the automation graphs Node-RED is built around).

- **Fleet** — grid and table views of every instance, with health at a glance.
- **Instances** — manage and configure a single instance.
- **Flows** — browse and deploy flows across instances.
- **Registry** — the source of truth (`registry.yaml`): serial numbers, hardware IDs, heartbeats, Helm deploy values, and per-instance metadata.

This is the “automation and orchestration” layer — when it’s green, the backbone the rest of the fleet rides on is healthy.

Restreaming — camera feeds & video walls

Pulls camera feeds (RTSP/RTMP/SRT/HLS and more) and routes them onto walls.

- **HoloVids** — a snapshot grid and live video wall of all feeds, with online/offline status in real time.
- **Holoprojector** — arranges feeds into saved *Views* (a layout plus its bound feeds) and broadcasts them to consoles and kiosks.

Settings for this area (FFmpeg defaults, grid layouts) live under **Settings** → **Restreaming / Layout Management**.

Imp Intel — UXV calibration & evaluation (TaskForce-X)

“Imperial Intelligence.” This is R2-D2’s **passive testing, calibration, and evaluation** workspace for unmanned vehicles (UXVs) during **TaskForce-X** sessions. It borrows the structure of a NATO ISR collection workflow — and the STANAG 4817 message set underneath it — but it is **not** a live operational tasking system.

Scope and authority. R2-D2 does **not** task nations, units, or live sensor platforms — we do not hold that authorization. Imp Intel is used *after* a session’s telemetry and video capture is complete, to replay and evaluate what each UXV recorded. Never to issue tasking.

What it is actually for: once a session’s data is in, these views help us **fine-tune each UXV’s baseline readings** — measuring each platform’s bias and variability against known-accuracy references, as a function of range — so its measurements can later be correlated and fused into the Common Operating Picture. The fusion itself is **Medusa**’s job (see *The Force*, below); Imp Intel is the calibration and bookkeeping front end that feeds it.

The collection-planning surfaces give that evaluation its shape:

- **Imp Intel** — the home view: the TCPED rolling window, collection requirements, rack-and-stack, and MOE/MOP effectiveness measures.
- **ICP** — the Intelligence Collection Plan; PIRs, SIRs, and EEIs (the layered information needs that frame a session).

- **IR Queue** — a parallel queue of standalone Information Requirements.
- **RFIs** — Requests For Information: outbound asks and inbound answers.
- **CRLs** — Collection Requirement Lists; bundle requirements for a session.
- **Scoring** — the rack-and-stack weights that prioritise requirements, with a live preview against the current queue.
- **Templates** — the import/export catalog mapping records across formats.

If the acronyms (PIR, EEI, CR, TCPED...) are unfamiliar, the app shows a plain tooltip on each — they trace back to NATO planning doctrine (AJP-5/AJP-2). Here they frame a **calibration exercise on our own UXVs**, not a live tasking order.

The Force — telemetry & situational awareness

Externally hosted operational apps, embedded directly inside R2-D2 so you never leave the console.

- **Force Sense** — read-only health of the Kubernetes cluster *and* the R2-D2 stack: controllers, workers, storage volumes, and pods.
- **Force Visions** — replay past UXV history: step back through recorded sessions, runs, and views to review what each platform did and saw.
- **Holocron** — a baseline viewer for the STANAG 4817 data riding the broker. Use it to confirm the **operational baseline** and to **validate** messages on the bus *before* they reach the COP — a pre-COP integration check that what's being published is correct, complete, and well-formed.
- **COP** — the Common Operating Picture (built on SitaWare): a unified view that aggregates tracks, layers, and sensor data.
- **Probe** — deep robot/machine introspection: ROS2 streams, Foxglove panels, real-time joint/frame data.
- **HoloNet** — an MQTT topic explorer; read and write messages on the bus.
- **The Alliance** — the registry of unmanned vehicles (ground, air, surface, subsurface), keyed by the call-sign aliases operators actually use.
- **Medusa** (Hydroid) — *planned; not yet built*. The contact-fusion engine. It will ingest the UXV measurements gathered and calibrated in Imp Intel, run **correlation and fusion** — weighting each reading by the variance learned during calibration — and publish **unified contacts** to the COP over the MQTT backbone. This page does not exist in the dashboard today.

The brokers that carry all this telemetry — **Rift Gates** (TBMQ) — are reached under **Settings** → **Rift Gates**.

How the data flows (the short version). UXV and sensor readings are published as STANAG 4817 messages onto the broker (**Rift Gates**). **Holocron** reads that bus to verify and validate the operational baseline *before* anything is trusted downstream. **Medusa** (planned) then correlates and fuses those validated readings into **unified contacts** and publishes them to the **COP**. In one line: *brokers* → *baseline check* → *fusion* → *operating picture*.

Temple Archives — the data layer

Durable storage and the shared reference catalogs the rest of the app reads from.

- **Farsight Feeds** — an OSINT intelligence API that provides data to over 75 datasources for COP usage (Disturbances, Events, CUI, vessel categorization, and related layers).

- **Feed Sync** — the worker that bridges REST feeds onto the MQTT bus.
- **4817 Link** — the STANAG 4817 endpoint for the 4817 docs, API, verification/validation, and integration.
- **Reference Data** — the cross-app catalogs (countries, regions, airfields, target categories, equipment, emitters) that Imp Intel and others build on.

The Republic — nations & inventory

The “federation of services” layer: who is in the coalition and what they field.

- **Member Nations** — allied nations with flags, country codes, domains, and contacts.
- **ShopKeepers** — the drone/vehicle vendor catalog (make and model).
- **Holdings** — each nation’s unmanned-vehicle inventory.
- **Platforms & Buildouts** — the platforms and sensor fit-outs (airframes, EO/IR, radar) behind those holdings.

Settings — configuration for every area

One place to tune every area, with both a friendly GUI and a raw-YAML “advanced” view for power users. The panels group by area:

- **Fleet** — Node-RED instance preferences and defaults.
- **Restreaming / FFmpeg** — encoder defaults, transport, and feed handling.
- **Layout Management** — the video-wall grid layouts HoloVids and Holoprojector draw from.
- **Holocron** — connection and display options for the 4817 baseline viewer.
- **COP** — SitaWare Common Operating Picture integration settings.
- **Ships** — UXV / platform display and call-sign options.
- **Temple Archives** — feed sources, sync cadence, and reference-data options.
- **The Republic** — member-nation, vendor, and inventory catalog settings.
- **Rift Gates** (TBMQ) — embedded broker console for the MQTT message brokers.
- **Zabbix** — embedded infrastructure-monitoring console.

Most areas pair a guided form with the underlying YAML, so you can start in the GUI and drop to raw config when you need to.

Droidspeak — the handbook

R2-D2’s own documentation, inside R2-D2.

- **Overview / Docs** — what the platform is, end to end, with a deep-dive page per area.
- **API Reference** — every endpoint the dashboard exposes, rendered as an interactive reference you can call live against the running fleet.
- **Install guides** — the production Kubernetes path and a single-node option.
- **Architecture** — runtime topology and the planned authentication migration.

Start here when you’re building on top of R2-D2 rather than operating it.

The mission, in one line

Keep the flows running, keep the cameras streaming, keep telemetry balanced, and keep the operating picture whole — all from one console.

One paragraph on sign-in

R2-D2 sits behind a single sign-in; once you're in, the authenticated areas (including the embedded Force apps) ride your session. If a page bounces you to a login, your session has expired — sign in again and return to it.

Lexicon

Two vocabularies run through R2-D2. The **dashboard's own names** (R2-D2, the Force, Holocron, Temple Archives, and so on) are decoded up front in *A note on the names* — start there for the Star Wars terms. The **abbreviations** below cover the ISR, doctrine, and infrastructure terms that appear across the areas.

Abbreviation	Expansion
AEP	Allied Engineering Publication
AJP	Allied Joint Publication
API	Application Programming Interface
APP	Allied Procedural Publication
COP	Common Operating Picture
CR	Collection Requirement
CRL	Collection Requirement List
CUI	Controlled Unclassified Information
EEI	Essential Element of Information
EO/IR	Electro-Optical / Infrared
HIBW	High-Information-Bandwidth
HLS	HTTP Live Streaming
ICP	Intelligence Collection Plan
IR	Information Requirement
ISR	Intelligence, Surveillance, and Reconnaissance
MOE / MOP	Measure of Effectiveness / Measure of Performance
MQTT	Message Queuing Telemetry Transport
OSINT	Open-Source Intelligence
PIR	Priority Intelligence Requirement
RFI	Request For Information
ROS2	Robot Operating System 2
RTMP / RTSP	Real-Time Messaging / Streaming Protocol
SIR	Specific Information Requirement
SRT	Secure Reliable Transport
STANAG	Standardization Agreement (NATO)
TCPED	Tasking, Collection, Processing, Exploitation, Dissemination
UXV	Unmanned Vehicle (any domain)
YAML	YAML Ain't Markup Language