European C-UAS Standards and Initiatives

Europe's main C-UAS standards are still maturing. The EUROCAE WG-115 has published ED-286/DO-389 (2021) defining the *Operational Services and Environment* (OSED) for C-UAS in controlled airspace, and ED-322/DO-403 (Dec 2023) covering **detection system performance** in airport settings [1] [2]. A future ED-286A update (2025) and a new "Interoperability Requirements for C-UAS" ED-xxx are in draft [2]. However, these documents remain high-level; ED-322 focuses on detection performance, not specific message protocols. Draft recommendations already urge use of **open standards** – for example Eurocontrol's WG-115 OSED draft advises C-UAS systems *"should support open standards to allow for scalability"* with new sensors, effectors, or third-party systems [3]. UN peacekeeping guidelines likewise emphasize that *"C-UAS systems should adhere to standard protocols for data exchange"* [4]. In short, Europe has begun defining **what** a C-UAS system must do, but **how** (protocols/architecture) is largely left open for users to choose.

UK SAPIENT Open Architecture

The UK MOD's **SAPIENT** standard provides a concrete example of an open C-UAS architecture. SAPIENT (BSI Flex 335) defines message schemas and APIs for heterogeneous sensors and decision nodes. The SAPIENT ICD is freely available (via BSI or GitHub) [5], and since 2019 MOD has **adopted it as its counter-drone standard** [6]. Critically, SAPIENT was proven in NATO trials: at NATO's TIE21 (2021) it enabled **70+ connections** between diverse C-UAS sensors and C2 systems, and at TIE22 (2022) it connected 31 sensor nodes from different vendors to 13 decision nodes [7]. SAPIENT uses Google's Protocol Buffers (protobuf) for lightweight messaging and even provides a middleware broker for routing/logging messages [5] [8]. In practice, adopting SAPIENT or a similar schema (or at least its design principles) gives a ready-made "alibi" of interoperability. SAPIENT is modular (plug-&-play), encourages AI fusion at the edge, and has test tools and sample data. Its success at NATO shows that building a system to SAPIENT's spec (or any published schema) will interoperate with many sensor/C2 systems. (The SAPIENT docs note usage of XML/JSON or protobuf – e.g. a switch to protobuf in v7 reduced bandwidth 60% [8] – so you could start with JSON over gRPC or similar if preferred.)

NATO and Other Standards

NATO provides some vehicle-control standards that impact C-UAS. **STANAG 4586** (fourth edition, 2017) defines interfaces between UAS ground control stations, air vehicles, and C4I nodes [9]. It specifies five "levels of interoperability" and which parts must comply. Complying with STANAG 4586 ensures your system can, e.g., control or receive telemetry from other NATO/U.S. drones. (In fact, NATO's upcoming STANAG 4817 will standardize multi-domain C2 links between a GCS and multiple uncrewed systems [10].) While STANAGs focus on flight control/data link, they illustrate the expectation of standardized messages. Other NATO work (e.g. the Maritime Unmanned Systems Initiative) points toward using common vocabularies (e.g. OGC sensor standards) for multi-domain ISR. In summary, aligning with NATO standards like STANAG 4586 (for UAS control links) and participating in NATO C-UAS exercises is advisable to ensure interoperability beyond just sensors.

U.S. Open Architectures and Protocols

U.S. C-UAS efforts strongly emphasize **open, modular systems** (the MOSA principle). For example, AeroVironment in 2025 announced integration of *OpenJAUS* into its AV_Halo command system, creating "a unified, open-standards framework for rapid UxS and control system integration" [11]. The press release notes that adding the JAUS standard makes the C2 "scalable, platform and sensor-agnostic" and accelerates multi-capability integration [12] [13]. (OpenJAUS is an open-source refinement of the DoD's JAUS messaging protocol, widely used for robotics/UAS interoperability.) In practice, using JAUS (or similar) means your system could plug into other U.S. unmanned systems with minimal glue code.

Other U.S. industry and DoD sources highlight open protocols. Booz Allen (a key C-UAS integrator) advertises that its C-UAS solutions use a *"modular open systems approach (MOSA)"* and *"open standards"* [14]. Their product documentation explicitly notes **TAK (Tactical Assault Kit)** compatibility and *"interoperable with existing commercial and government sensors, effectors and systems"* [15]. In their words: "Our systems connect the best technologies, regardless of source and without vendor lock-in" [16]. This means, for example, using common GIS formats (TAK uses Cursor on Target messages) and letting users view/command UAS tracks in standard maps. Northrop Grumman's M-ACE C-UAS also touts an "open-architecture software system" integrating hundreds of sensors and cameras [17]. The takeaway: commercial C-UAS solutions are designed to talk "standard languages" so you can mix-and-match sensors/effectors from different vendors.

Government reports mirror this philosophy. A 2024 MITRE study on homeland C-UAS recommends that OSTP/NIST "lead a coordinated effort to design, develop, implement C-UAS data and information exchange standards" to enhance interoperability and data quality [18]. In short, U.S. authorities acknowledge that interoperability needs explicit standards. For now, key examples include JAUS, STANAGs, and open schemas like SAPIENT's. (Note that the FAA/DHS guidance [16] is mostly legal – there's no mandated U.S. standard for C-UAS messaging yet.)

Integration Protocols and Architectures (Recommendations)

Given all this, the recommended **path forward** is to adopt an open, standards-based integration architecture from the start. Since you're building *all new* systems, you have no legacy to worry about and can freely choose best-of-breed protocols. Consider these actions and approaches:

- **Adopt a published C-UAS messaging schema.** For example, use the SAPIENT ICD (BSI Flex 335) or a similar interface as your "common language." This means defining all sensor reports (radar track, RF detection, visual ID, etc.) and effect commands as structured messages (e.g. protobuf or JSON). SAPIENT's GitHub provides proto definitions and test tools [5]. Even if not using SAPIENT itself, mimicking its architecture (edge-AI sensors that send semantic data, plus a decision-node fusion engine) ensures modularity. Alternatively, evaluate JAUS or the NATO STANAG interfaces if your system must integrate with existing UAS C2.

- **Use modern middleware/brokers.** Many C-UAS systems use a publish/subscribe data bus or broker to handle messages (e.g. SAPIENT Middleware, DDS, MQTT, or ZeroMQ). This decouples sensors and effectors. For example, SAPIENT's middleware routes node-generated messages and allows external queries [19]. Anduril's Lattice OS takes a similar "network-centric" view: it *"combines the phases of the C-UAS kill chain into an open architecture operating system"* that runs on a network and layers software for sensor fusion and targeting [20]. In practice, you could use an existing bus (ROS2/ROS, DDS/RTPS, or even Kafka) with agreed-upon message topics matching your chosen schema.

- **Ensure modularity and scalability.** Follow MOSA principles [14] [16]: design each function (detection, tracking, ID, shooter) as a plug-in module with a defined API. This way you can add new sensor types or AI algorithms without rewriting the whole system. For example, Dstl's SAPIENT was explicitly designed to let new AI-enabled sensor modules "plug-and-play" and fuse data [21] [22]. Likewise, your C2/UI should be modular (map servers, rule engines, etc.) and consume the common data bus.

- **Use geospatial and C2 standards.** For commands and control, leverage existing protocols where possible. For instance, use the Cursor-on-Target (CoT) or Link 16 equivalent for pointing map markers, as used by TAK and NATO (STANAG 4607 for surveillance tracks). Sensor data can be formatted in open geospatial formats (GeoJSON, SensorML) or mapped to NATO compliance. If using SAPIENT, it already has field definitions, but if not, at least use common field names for GPS, altitude, velocity, threat classification, etc.

- **Adopt an open data format.** Modern practice is to send structured messages in a neutral format (JSON, XML, or binary like protobuf). Europe's SAPIENT switched from XML to protobuf to cut bandwidth [8]. JAUS also has XML and binary profiles. If you build on something like SAPIENT or DDS, you'll naturally pick an efficient encoding. This ensures performance and lets you interoperate with any language/platform.

- **Leverage vendor ecosystems.** You're open to vendor standards, so pick one that has industry backing. Besides SAPIENT and JAUS, look at: OGC SensorThings or OGC SOS for sensor querying; Open Geospatial Consortium formats for data sharing; and networking standards (IPv6, DiffServ, TLS for security). Because you're all-new, you can choose IPv6 networks and modern cybersecurity by design. For UI, integrate with common C2 apps (TAK, C2IME) via their plug-ins/APIs.

- **Reference existing deployments.** As an "alibi," cite existing uses: e.g. "We will use the SAPIENT message model (open-source ICD) since it has been NATO-tested [7]." Or "We adopt JAUS message formats as per DoD guidance [11]." Or mention that Northrop's M-ACE, GA's pod solution, and Booz Allen's CUxS are built on similar open, multi-sensor frameworks [17] [14]. These citations show your design is industry-proven.

- **Stay engaged with standards bodies.** Watch EUROCAE ED-xxx (due 2025-26) for interoperability requirements [2]. Monitor NATO's Alliance Ground Surveillance and Joint Concept Group on UAS (JCG-UAS) for new STANAGs. Collaborate in any open trials (like NATO TIE or EU research projects) to validate your choices.

In summary: Do *not* build a proprietary silo. Use an open-architecture middleware with well-known protocols (e.g. SAPIENT-style messages or JAUS) so that each sensor, effecter, and control node speaks the same "language." This approach is explicitly recommended by industry and authorities [3] [4] [14]. By aligning with these practices – modular design, use of standard data formats, and open interfaces – you ensure your C-UAS system can integrate new components and interoperate with allies' systems. This meets the "build it like they did elsewhere" criterion: there **are** existing open protocols (SAPIENT, JAUS, STANAG data links, TAK/CoT) used in UK, NATO and US deployments, and you should design your system to employ them.

Sources: Official EUROCAE standards announcements [1] [2]; UK MOD/GOV SAPIENT docs [5] [7] [6]; NATO analysis [9]; U.S. industry and government releases [11] [14] [18]. These collectively validate the above guidance.

---

[1] ED-322 | System Performance and Interoperability Requirements for Non-Cooperative UAS Detection Systems - EUROCAE
https://www.eurocae.net/product/ed-322-system-performance-and-interoperability-requirements-for-non-cooperative-uas-detection-systems/

[2] COMMUNICATION & TRAINING MANAGER AT EUROCAE
https://www.eurocontrol.int/sites/default/files/2024-11/eurocontrol-2024-cuas-workshop-s3-milns.pdf

[3] Pilot Project VTOL
https://www.eurocontrol.int/sites/default/files/2020-11/tim-2020-day-4-industry-standards-el-malek.pdf

[4] resourcehub01.blob.core.windows.net
https://resourcehub01.blob.core.windows.net/$web/Policy%20and%20Guidance/corepeacekeepingguidance/Thematic%20Operational%20Activities/Military/2025.16%20Guidelines%20on%20Counter%20Unmanned%20Aircraft%20Systems.pdf

[5] [6] [7] [8] [19] [21] [22] SAPIENT autonomous sensor system - GOV.UK
https://www.gov.uk/guidance/sapient-autonomous-sensor-system

[9] [10] An Urgent Matter of Drones: Lessons for NATO from Ukraine - CEPA
https://cepa.org/comprehensive-reports/an-urgent-matter-of-drones/

[11] [12] [13] AV Announces Collaboration with OpenJAUS for Autonomous Uncrewed System 〈Uxs〉 Interoperability
https://www.avinc.com/resources/av-in-the-news/view/av-and-openjaus-announce-collaboration-for-autonomous-uncrewed-system-uxs-interoperability

[14] [15] [16] Counter-UAS Technologies
https://www.boozallen.com/markets/defense/counter-uas-technologies.html

[17] Counter Unmanned Aerial Systems (C-UAS) - Northrop Grumman | Northrop Grumman
https://www.northropgrumman.com/what-we-do/mission-solutions/counter-unmanned-aerial-systems-c-uas

[18] COUNTERING UNMANNED AIRCRAFT SYSTEMS SECURING THE HOMELAND AGAINST EVOLVING THREATS
https://www.mitre.org/sites/default/files/2025-01/PR-24-01820-27-Countering-Unmanned-Aircraft-Systems.pdf

[20] Anduril and Epirus integration designed to increase US Marine Corps counter-UAS capabilities – Unmanned airspace
https://www.unmannedairspace.info/counter-uas-systems-and-policies/anduril-and-epirus-integration-designed-to-increase-us-marine-corps-counter-uas-capabilities/