INTERNATIONAL

Designation: F3411 – 19

Standard Specification for Remote ID and Tracking[1]

This standard is issued under the fixed designation F3411; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ε) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 This specification covers the performance requirements for remote identification (Remote ID) of unmanned aircraft systems (UAS). Remote ID allows governmental and civil identification of UAS for safety, security, and compliance purposes. The objective is to increase UAS remote pilot accountability by removing anonymity while preserving operational privacy for remote pilots, businesses, and their customers. Remote ID is an enabler of enhanced operations such as beyond visual line of sight (BVLOS) operations as well as operations over people.

1.2 This specification defines message formats, transmission methods, and minimum performance standards for two forms of Remote ID: broadcast and network. Broadcast Remote ID is based on the transmission of radio signals directly from a UAS to receivers in the UAS's vicinity. Network Remote ID is based on communication by means of the internet from a network Remote ID service provider (Net-RID SP) that interfaces directly or indirectly with the UAS, or with other sources in the case of non-equipped network participants.

1.3 This specification addresses the communications and test requirements of broadcast or network Remote ID, or both, in UAS and Net-RID SP systems.

1.4 *Applicability:*

1.4.1 This specification is applicable to UAS that operate at very low level (VLL) airspace over diverse environments including but not limited to rural, urban, networked, network degraded, and network denied environments, regardless of airspace class.

1.4.2 This specification neither purports to address UAS operating with approval to use ADS-B or secondary surveillance radar transponders, nor does it purport to solve ID needs of UAS for all operations.

1.4.3 In particular, this specification does not purport to address identification needs for UAS that are not participating in Remote ID or operators that purposefully circumvent Remote ID.

1.5 The values stated in SI units are to be regarded as standard. The values given in parentheses after SI units are provided for information only and are not considered standard.

1.5.1 Units of measurement included in this specification:

m	meters
deg, °	degrees of latitude and longitude, compass direction
s	seconds
Hz	Hertz (frequency)
dBm	decibel-milliwatts (radio frequency power)
ppm	parts per million (radio frequency variation)
µs	microseconds
ms	milliseconds

1.6 *Table of Contents:*

1.7 *This standard does not purport to address all of the safety concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appropriate safety, health, and environmental practices and determine the applicability of regulatory limitations prior to use.* Some specific hazards statements are given in Section 8 on Hazards.

[1] This specification is under the jurisdiction of ASTM Committee F38 on Unmanned Aircraft Systems and is the direct responsibility of Subcommittee F38.02 on Flight Operations.

Current edition approved Dec. 1, 2019. Published February 2020. DOI: 10.1520/F3411–19.

1.8 *This international standard was developed in accordance with internationally recognized principles on standardization established in the Decision on Principles for the Development of International Standards, Guides and Recommendations issued by the World Trade Organization Technical Barriers to Trade (TBT) Committee.*

2. Referenced Documents

2.1 *ASTM Standard:*[2]

F3196 Practice for Seeking Approval for Beyond Visual Line of Sight (BVLOS) Small Unmanned Aircraft System (sUAS) Operations

2.2 *Other Standards:*

ANSI/CTA-2063-A Small Unmanned Aerial Systems Serial Numbers[3]

Bluetooth[4,5] Core Specification 5.0[6]

IEEE 802.11-2016 Standard for Information technology--Telecommunications and information exchange between systems - Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications[7,5]

IETF RFC3339 Date and Time on the Internet: Timestamps[8]

IETF RFC4122 A Universally Unique IDentifier (UUID) URN Namespace[9]

Neighbor Awareness Networking Specification [10,5]

FAA UTM ConOps v1.0 Unmanned Aircraft System (UAS) Traffic Management (UTM) Concept of Operations[11]

WGS-84 World Geodetic System — 1984[12]

3. Terminology

3.1 *Definitions:*

3.1.1 *authentication*—the process or action of verifying that the source of a Remote ID message is the originator of the message.

3.1.2 *beyond visual line of sight (BVLOS)*—the operation when the individual responsible for controlling the flight of the unmanned aircraft (UA) cannot maintain direct unaided (other than by corrective lenses or sunglasses, or both) visual contact with the UA, other aircraft, terrain, adverse weather, or obstacles to determine whether the UA endangers life or property, or both. **F3196**

3.1.3 *broadcast*—to transmit data to no specific destination or recipient; data can be received by anyone within broadcast range.

3.1.4 *broadcast UAS*—a UAS that is equipped for and is actively broadcasting Remote ID data during an operation; being a broadcast UAS is not mutually exclusive with being a networked UAS.

3.1.5 *discovery*—the process of determining the required USS data exchanges to successfully complete Net-RID services; this is accomplished using the discovery and synchronization service (DSS).

3.1.6 *DSS entity*—a generic concept that refers to information that can be discovered using the discovery and synchronization service (DSS).

3.1.6.1 *Discussion*—Entities are characterized by a 4-D volume of airspace (that is, a volume defined in x, y, z plus time limits). For Remote ID, the entity type is referred to as an identification service area. Operations and constraints are examples of other types of entities that are the subject of other UTM standards.

3.1.7 *DSS region*—the geographic scope supported by a set of discovery and synchronization service instances.

3.1.8 *dynamic data*—data that changes over the duration of the flight; for example, longitude and latitude.

3.1.9 *Ground Control Station (GCS)*—the part of a UAS that remotely controls the UA. It may or may not have a remote pilot directly manipulating the controls.

3.1.10 *identify*—the result of the process to establish the identity of a specific UAS that is traceable to the owner and remote pilot.

3.1.11 *network Remote ID (Net-RID) service provider*—a logical entity denoting a UTM system or comparable UAS flight management system that participates in network Remote ID and provides data for and about UAS it manages.

3.1.12 *network Remote ID (Net-RID) display provider*—a logical entity that aggregates network Remote ID data from potentially multiple Net-RID service providers and provides the data to a display application (that is, an app or website); in practice, it is expected that many USSs may be both Net-RID display providers and Net-RID service providers, but standalone Net-RID display providers are possible.

3.1.13 *network publishing*—the act of transmitting data to an internet service or federation of services; clients, whether air traffic control (ATC), public safety officials, or possibly the

[2] For referenced ASTM standards, visit the ASTM website, www.astm.org, or contact ASTM Customer Service at service@astm.org. For *Annual Book of ASTM Standards* volume information, refer to the standard's Document Summary page on the ASTM website.

[3] Available from American National Standards Institute (ANSI), 25 W. 43rd St., 4th Floor, New York, NY 10036, http://www.ansi.org.

[4] Used throughout the specification, Bluetooth is a registered trademark of Bluetooth SIG, Inc., 5209 Lake Washington Blvd. NE, Suite 350, Kirkland, WA 98033.

[5] Other names and brands may be claimed as the property of others.

[6] Available from https://www.bluetooth.com/specifications/archived-specifications/.

[7] Available from Institute of Electrical and Electronics Engineers, Inc. (IEEE), 445 Hoes Ln., Piscataway, NJ 08854-4141, https://standards.ieee.org/standard/802_11-2016.html.

[8] Available from IETF Tools, https://tools.ietf.org/html/rfc3339.

[9] Available from IETF Tools, https://tools.ietf.org/html/rfc4122.

[10] Available from Wi-Fi Alliance, 10900-B Stonelake Boulevard, Suite 126, Austin, TX 78759, https://www.wi-fi.org/discover-wi-fi/wi-fi-aware.

[11] Available from https://utm.arc.nasa.gov/docs/2018-UTM-ConOps-v1.0.pdf.

[12] Available from International Civil Aviation Organization (ICAO), 999 Robert-Bourassa Boulevard, Montréal, Quebec, Canada H3C 5H7, https://gis.icao.int/eganp/webpdf/REF08-Doc9674.pdf.

general public can access the data to obtain ID and tracking information for UAS for which such data has been published.

3.1.14 *networked UAS*—a UAS that during operations is in electronic communication with a Net-RID service provider (for example, by means of internet Wi-Fi,[13] cellular, or satellite, or other communications medium such as short burst data satellite communications).

3.1.15 *non-equipped UAS*—in the context of Remote ID, a UAS that is neither a networked nor broadcast UAS (for example, a radio controlled model aircraft) and cannot directly report its location or identity.

3.1.16 *non-equipped network participant*—a non-equipped UAS for which the operator has reported an intended area (a volume of airspace) and time for an operation through a Net-RID service provider; such information is then reported through the network Remote ID infrastructure.

3.1.17 *operator*—entity or person responsible for flight which could include a company or individual, or both; this document makes no statement about legal responsibility in how these terms are used.

3.1.18 *operator location*—the geographic location of the remote pilot of a UAS.

3.1.19 *position extrapolation*—a capability of a Net-RID service provider to predict the location of a UAS based on a modeled 4-D trajectory derived from an intended UAS operation plan.

3.1.20 *remote pilot*—the person who has final authority and responsibility for the operation and safety of flight; synonymous with "remote pilot-in-command."

3.1.21 *registration*—the process by which an owner/operator (including contact information and other PII) and aircraft (for example, make, model) are associated with an assigned, unique identifier.

3.1.22 *shall, must versus should versus may*—use of the word "shall" implies that a procedure or statement is mandatory and must be followed to comply with this practice, "should" implies recommended, and "may" implies optional at the discretion of the supplier, manufacturer, or operator.

3.1.22.1 *Discussion*—Since "shall" and "must" statements are requirements, they include sufficient detail needed to define compliance (for example, threshold values, test methods, oversight, and references to other standards). "Should" statements also represent parameters that could be used in safety evaluations, and could lead to development of future requirements. "May" statements are provided to clarify acceptability of a specific item or practice, and offer options for satisfying requirements.

3.1.23 *static data*—data that remains the same or does not change often over the duration of a flight (for example, Unique ID); this is in contrast to dynamic data that may change more frequently (such as longitude and latitude).

3.1.24 *unmanned aircraft system (UAS)*—composed of unmanned aircraft and all required on-board subsystems, payload, control station, other required off-board subsystems, any required launch and recovery equipment, all required crew members, and command and control (C2) links between UA and the control station.

3.1.25 *UAS operation plan*—a flight plan for a UAS. An operation plan is developed prior to the operation and should indicate the volume of airspace within which the operation is expected to occur, the times and locations of the key events associated with the operation, including launch, recovery, and any other information deemed important (for example, segmentation of the operation trajectory by time).
UTM ConOps v1.0

3.1.26 *UAS registration ID*—an identification number or combination of letters and numbers assigned by a CAA or authorized representative to a UAS; this is sometimes referred to as a registration number (which may or may not contain letters).

3.1.27 *UAS service supplier (USS)*—USSs provide UTM services to support the UAS community, to connect operators and other entities to enable information flow across the USS network, and to promote shared situational awareness among UTM participants. **UTM ConOps v1.0**

3.1.28 *unique ID*—a data element that can be traced to a unique UAS and its operator.

3.2 *Acronyms and Abbreviations:*

3.2.1 *AES*—advanced encryption standard

3.2.2 *AGL*—above ground level

3.2.3 *API*—application programming interface

3.2.4 *ARC*—aviation rulemaking committee

3.2.5 *BVLOS*—beyond visual line of sight

3.2.6 *C2*—command and control

3.2.7 *CAA*—Civil Aviation Authority

3.2.8 *CONUS*—contiguous United States

3.2.9 *DAR*—DSS airspace representation

3.2.10 *DSS*—discovery and synchronization service

3.2.11 *EIRP*—effective isotropic radiation pattern

3.2.12 *EMI*—electromagnetic interference

3.2.13 *FAA*—Federal Aviation Administration

3.2.14 *GCS*—ground control station

3.2.15 *Hz*—Hertz

3.2.16 *inHg*—inch of mercury

3.2.17 *km*—kilometers

3.2.18 *kts*—knots (nautical miles per hour)

3.2.19 *LAANC*—low altitude authorization and notification capability

3.2.20 *LE*—little endian (least significant byte first)

3.2.21 *LSB*—least significant bit

3.2.22 *LTA*—lighter than air (for example, balloon or blimp)

3.2.23 *m*—meters

3.2.24 *m/s*—meters per second

3.2.25 *mb*—millibars

[13] Used throughout the specification, Wi-Fi is a registered trademark of Wi-Fi Alliance, 10900-B Stonelake Boulevard, Suite 126, Austin, TX 78759.

3.2.26 *mm*—millimeters

3.2.27 *MAC*—media access control

3.2.28 *MPH*—miles per hour

3.2.29 *MSB*—most significant bit

3.2.30 *Net-RID*—network Remote ID

3.2.31 *PHY*—physical layer

3.2.32 *PII*—personally identifiable information

3.2.33 *PPM*—parts per million

3.2.34 *Remote ID*—remote identification

3.2.35 *TLS*—transport layer security

3.2.36 *UA*—unmanned aircraft

3.2.37 *UAS*—unmanned aircraft system

3.2.38 *USS*—UAS service supplier

3.2.39 *UTM*—UAS traffic management

3.2.40 *UUID*—universally unique identifier based on RFC4122 (128 bit)

3.2.41 *VIP*—very important person

3.2.42 *VTOL*—vertical take off and landing

3.2.43 *VLL*—very low level (airspace—generally below 150 m (500 ft))

4. Remote ID and Network Interoperability Conceptual Overview

4.1 This section provides a conceptual overview of Remote ID as defined in this specification, explains the scope of the specification, and clarifies the differences between broadcast and network Remote ID. This overview does not address all nuances of the specification. The intention is to provide a contextual framework to understand the requirements contained in this specification. No requirements are provided in this section.

4.2 This section also provides an overview of the general approach to interoperability between USSs for both Network Remote ID and other UTM-related services.

4.3 *Scope of Standard and Remote ID Components:*

4.3.1 Fig. 1 identifies the actors and interfaces between actors in the Remote ID environment.

4.3.2 The scope of this specification is identified by the contents of the dotted purple box in the center of the diagram.

4.4 *Broadcast Remote ID:*

4.4.1 Broadcast Remote ID is depicted in the upper, central portion of Fig. 1 in blue. Equipment on participating UAS continuously transmit Remote ID data using one of the transmit protocols in this specification (Bluetooth or Wi-Fi). It is possible that additional transmit protocols may be added in the future as warranted by available technology. The initial technologies were selected for compatibility with commonly carried hand-held devices that have their own receiver antenna. However, equipment to receive the broadcast data is not part of the specification. Other implementations, such as receivers not integrated with hand-held devices, are possible.

4.4.2 Both Bluetooth and Wi-Fi continuously broadcast messages to advertise the presence of the associated device. These advertisements normally allow other devices to discover and establish connections with the associated device, but the advertisements themselves can carry a payload. These advertisements contain the broadcast Remote ID data. A hand-held

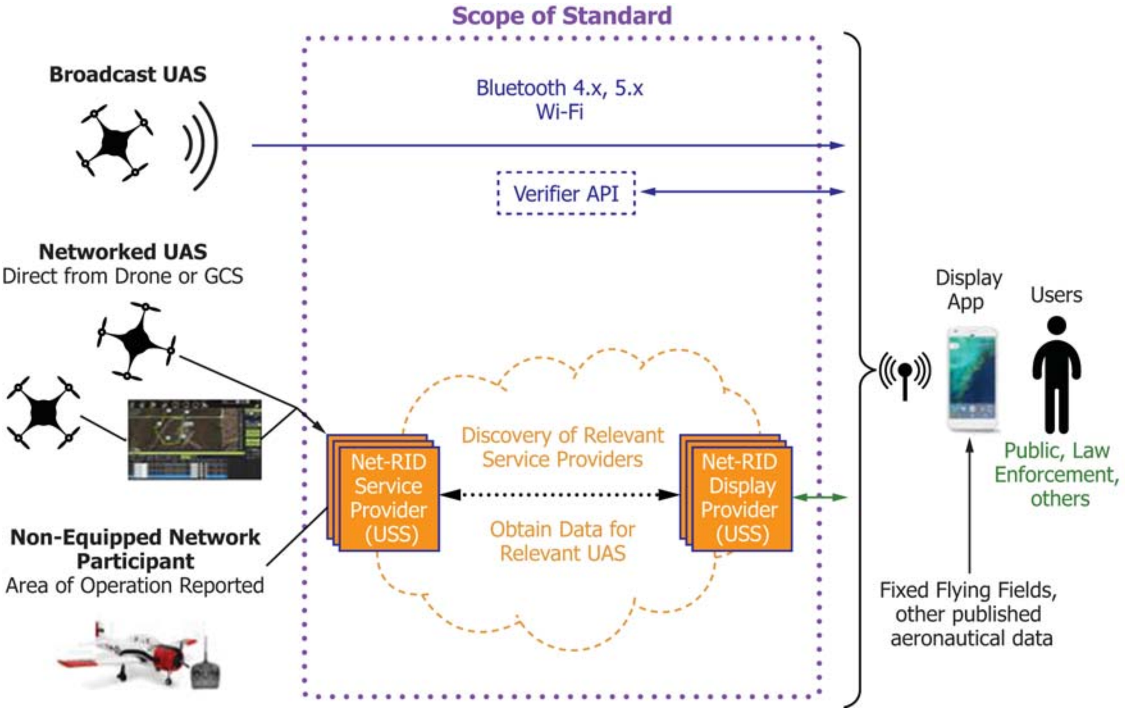


FIG. 1 Remote ID Conceptual Overview

device does not need to establish a connection to receive Remote ID data, instead it need only receive and process the advertisements.

4.4.3 Broadcast Remote ID can be used anywhere, but is necessary in areas where network coverage is unreliable, disrupted, or not available.

4.4.4 The specification also includes a range of options for authentication of broadcast data. Some of those options include digital signatures over portions or all of the Remote ID message set. While the specification does not specify the encoding format associated with signatures, it does include a standard API that would be used by a receiver of the broadcast data (for example, an app on a smartphone) to contact a verifier with the signature data for a broadcast to determine message validity. This is described in more detail in Annex A1, Broadcast Authentication Verifier Service.

4.5 *Network Remote ID:*

4.5.1 Network Remote ID can be used when both UAS operations and end users of Remote ID display applications access the internet, typically by means of cellular network. Cellular coverage tends to be higher in urban areas.

4.5.2 Network Remote ID is depicted in the lower, central portion of Fig. 1 in orange and enclosed in a dashed line cloud. The nominal case supports Networked UAS (that is, UAS that remain in contact with a Remote ID Service Provider during flight), although the specification accommodates intermittent loss of network connectivity. Network Remote ID also includes provisions for participation in Remote ID by non-equipped UAS (that is, UAS that are neither broadcast capable nor equipped to communicate with a Remote ID Service Provider during flight, such as most radio-controlled model aircraft). These non-equipped network participants report their operations (for example, aircraft ID, location in terms of a volume of airspace, operating times) in advance. The information is used to create a position report for use by Remote ID display applications where the uncertainty of the position report is defined by the airspace volume for the operation. The current telemetry of the aircraft within the volume is not known and cannot be displayed to a Remote ID end user, but the display application can display the volume and provide the identity of the UAS.

4.5.3 For Network Remote ID, two USS roles are identified: Network Remote ID (Net-RID) Service Providers and Net-RID Display Providers. In practice, these roles can be fulfilled by a single USS and potentially one that also provides flight planning and deconfliction, LAANC, or other UTM services, or combinations thereof. However, they are identified separately to provide flexibility for industry participants to pursue their preferred business objectives and implementation scope. This architecture supports one or more Net-RID Service Providers and one or more Net-RID Display Providers.

4.5.4 Net-RID Service Providers nominally remain in contact with UAS during flight and receive information (for example, position updates) used to fulfill requests from Net-RID Display Providers. For Network Remote ID, some required data (for example, the UAS ID) may be retained by the Net-RID Service provider after UAS authentication and not transmitted continuously from the UAS. As this specification does not specify the details of the UAS to Net-RID Service provider interface, implementations are generally valid as long as complete and correct Remote ID data is obtained by Net-RID Service Providers at some point and made available to Net-RID Display Providers.

4.5.5 Net-RID Service Providers may also have the ability to supply extrapolated position information for UAS that intermittently lose network connectivity.

4.5.6 Net-RID Display Providers fulfill a broker role between Remote ID Display Applications used by an end user and all Net-RID Service Providers that have flights in an area. When an end user display application requests Remote ID data for an area, the Net-RID Display Provider servicing the display application determines what Net-RID Service Providers have operations in the area and then obtains appropriate Remote ID data from each. The aggregated data is returned to the Remote ID Display Application. The aggregated data includes both current location and a window of near-real-time data for each flight.

4.5.7 Net-RID Display Providers ensure Remote ID Display Applications can only access and view data within a limited range and must dispose of aggregated data obtained from Net-RID Service Providers within a defined time period. Limiting the range helps implementations satisfy performance requirements in this specification by bounding the volume of data that must be gathered, processed, and displayed. Limiting the range (that is, only accessing required data) and disposing of such data when no longer needed helps protect privacy and sensitive data of consumers and operators.

4.5.8 For a UAS to be included in response to queries for Remote ID data, it must either be within the requested area at the time of the request or recently therein (that is, within a small window of time such as a minute). This specification does not provide remote identification data for UAS that are projected to be within an area in the future.

4.5.9 Industry-standard encryption and authentication are required from the UAS or the operator of a non-equipped network participant to the Net-RID Service and from the Net-RID Service Providers to Net-RID Display providers.

4.6 *Remote ID Display Applications:*

4.6.1 Receivers and Remote ID Display Applications are shown on the right side of Fig. 1. A typical implementation would be a smartphone or tablet with an internal receiver for Bluetooth and Wi-Fi, but other implementations are possible. The display applications ingest Broadcast Remote ID data or interact with a Net-RID Display Provider, or both, to acquire Network Remote ID data and present the information to end users.

4.6.2 A typical user interface might be map-based with symbols for UAS in the area. However, the manner in which the information is presented is beyond the scope of this specification and other implementations are possible.

4.6.3 It is anticipated that Remote ID Display Applications that integrate Broadcast and Network Remote ID data will be produced by industry; however, this also is beyond the scope of the specification.

4.6.4 From a network Remote ID perspective, this specification levies performance requirements on Net-RID Display Providers in responding to requests from Remote ID Display Applications.

4.7 *Representative Remote ID Scenario:*

4.7.1 Fig. 2 depicts a representative Remote ID scenario. The text that follows describes the flow of information amongst the Remote ID components introduced above.

4.7.2 Three UAS are simultaneously operating in close proximity (within 1 km) to each other: one is broadcasting Remote ID data, one is networked, and one is a model aircraft with no broadcast or network capability. An interested observer wants to identify the three UAS.

4.7.2.1 The broadcast UAS transmits Remote ID data using one of the methods described in 5.4. The UAS is controlled locally by the Remote Pilot and has no interface with a USS.

4.7.2.2 The networked UAS is operated under USS1. This USS acts as a Net-RID Service Provider and a Net-RID Display Provider.

4.7.2.3 The Remote Pilot of the model aircraft uses a smartphone app to report the location and time of the operation, and provides the ID for the model aircraft. The smartphone app is the user interface that connects the user to a second Net-RID Service Provider, USS2.

4.7.3 The interested observer accesses a Remote ID Display Application (RID App) that uses USS1 as its Net-RID Display Provider. This display application shows UAS locations and a near-real-time trail of position reports on a map, and associated identification information when a particular UA is selected.

4.7.4 When the interested observer opens the Remote ID app on a smartphone and centers the map on the current location, Remote ID data is acquired as follows:

4.7.4.1 The broadcast UAS is transmitting its Remote ID advertisements continuously. The smartphone uses its internal radios to listen for the advertisements from the UAS, extract the Remote ID data, and show the location of the UA on the map. As new position updates are received, the prior position reports become part of a near-real-time trail representing where the UA most recently flew.

4.7.4.2 Simultaneously, the Display App makes a request to its Net-RID Display Provider, USS1. USS1's role as a Net-RID Display Provider is to aggregate Remote ID data for all flights in the area managed by Net-RID Service Providers. USS1 knows that it is a Net-RID Service Provider and has flights in the area.

4.7.4.3 USS1 additionally discovers USS2, which has no real-time-managed flights in the area, but has an operation reported for the model aircraft (that is, the Non-Equipped Network Participant). The Remote Pilot of the radio-controlled model aircraft reports the operation to USS2 prior to the flight and no dynamic position updates are provided by the Non-Equipped Network Participant. USS2 provides the information for this operation back to USS1.

4.7.4.4 USS1 adds the Remote ID data for the networked UAS that it is managing (fulfilling its role as a Net-RID Service Provider) and provides the aggregated data back to the Display Application (fulfilling its role as a Net-RID Display Provider).

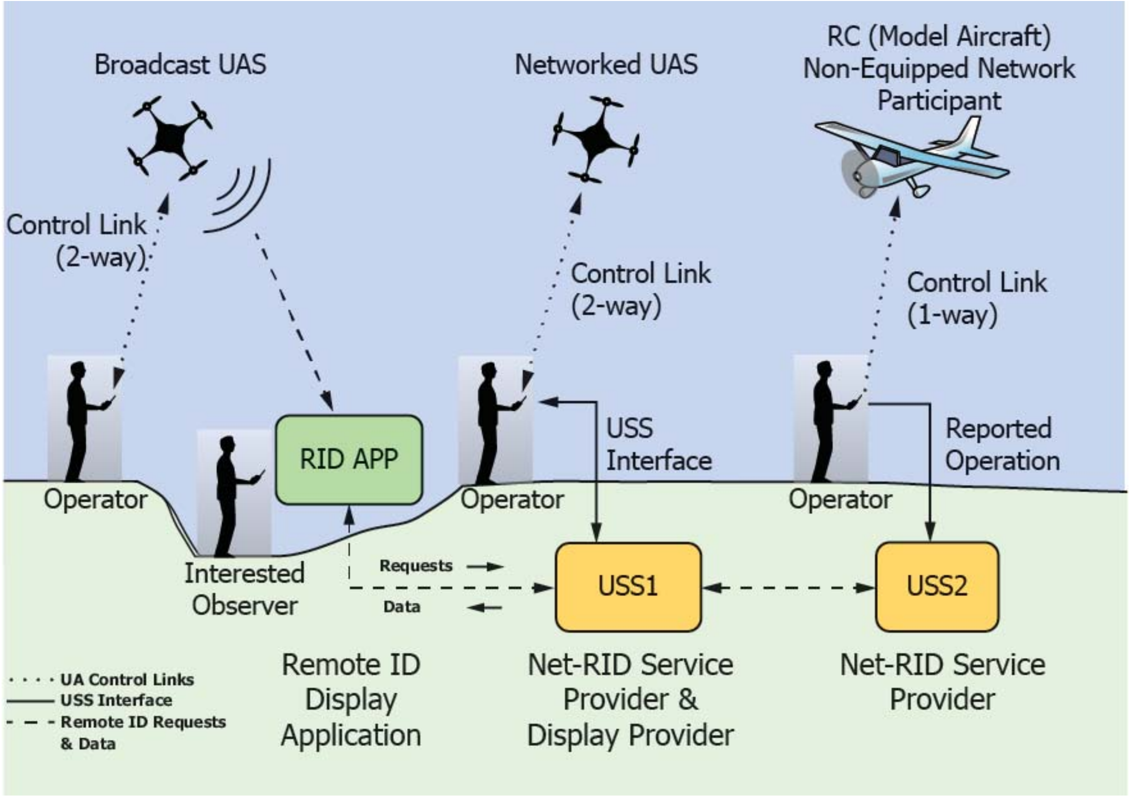


FIG. 2 Representative Remote ID Scenario

4.7.4.5 The display app adds the network data to the map that is already showing the broadcast information. The Networked UAS is shown with up to a 60 s trail of position reports (referred to as near-real-time data). The Non-Equipped Network Participant is shown as a polygon.

4.7.4.6 As long as the interested observer continues to view the Remote ID display app for the area, the app continues to communicate with USS1 as its Net-RID Display Provider to obtain position updates. Since the information for the non-equipped network participant does not update, no additional updates are provided for it from USS2. USS1 continues to provide position updates for the Networked UAS in its role as the Net-RID Service Provider.

4.7.4.7 The interested observer selects the UA symbols for the Broadcast UAS and the Networked UAS on the map and views the corresponding ID information. The interested observer then selects the polygon for the Non-Equipped Network Participant and sees the operation schedule and the ID of the aircraft.

4.7.4.8 The interested observer closes the app. After a period of time, USS1 discards the information for the non-equipped Network Participant because it is managed by a different USS (USS2).

4.8 *USS Interoperability:*

4.8.1 This specification assumes that UTM services in a given location are provided by a set of one or more UAS Service Suppliers (USSs). USSs must be interoperable in this environment, sharing data as necessary to accomplish the objective of a particular service such as Network Remote ID or flight plan exchange for strategic deconfliction.

4.8.2 The interoperability paradigm defined in this specification is intended to support both Network Remote ID and other services that may be included in subsequent UTM-related ASTM standards. The requirements and application programming interfaces (APIs) associated with the interoperability paradigm are included in this document because it is the first UTM-related ASTM standard. Subsequent UTM-related ASTM standards may introduce additional service-specific interoperability requirements and API functions. Some of the interoperability requirements and APIs may move to a different standard in the future.

4.8.3 The interoperability paradigm consists of two parts:

4.8.3.1 A standardized discovery mechanism, referred to as the Discovery and Synchronization Service (DSS), the primary functions of which are to identify USSs with which data exchange is required and to verify that a USS considered relevant information from other USSs when necessary (for example, when planning a new operation); and,

4.8.3.2 Service-specific data exchange protocols used to obtain the details of relevant information discovered by means of the DSS from the owning USS.

4.8.4 Fig. 3 illustrates the interactions involved in this paradigm in a service-independent manner. (The instantiation of this paradigm for Remote ID is detailed later.)

4.8.5 DSS-related interactions are shown at the top in the blue-shaded area; data exchange protocols between USSs are represented by the green-shaded area.

4.8.6 For availability purposes, the DSS is a redundant service as indicated in Fig. 3. Instances of the DSS in a region synchronize with each other in a standardized manner (described later in this specification). A region is the geographic scope supported by a set of DSS instances.

4.8.7 Only approved USSs will be given access to the DSS. (The specifics of an approval process are beyond the scope of this specification.)

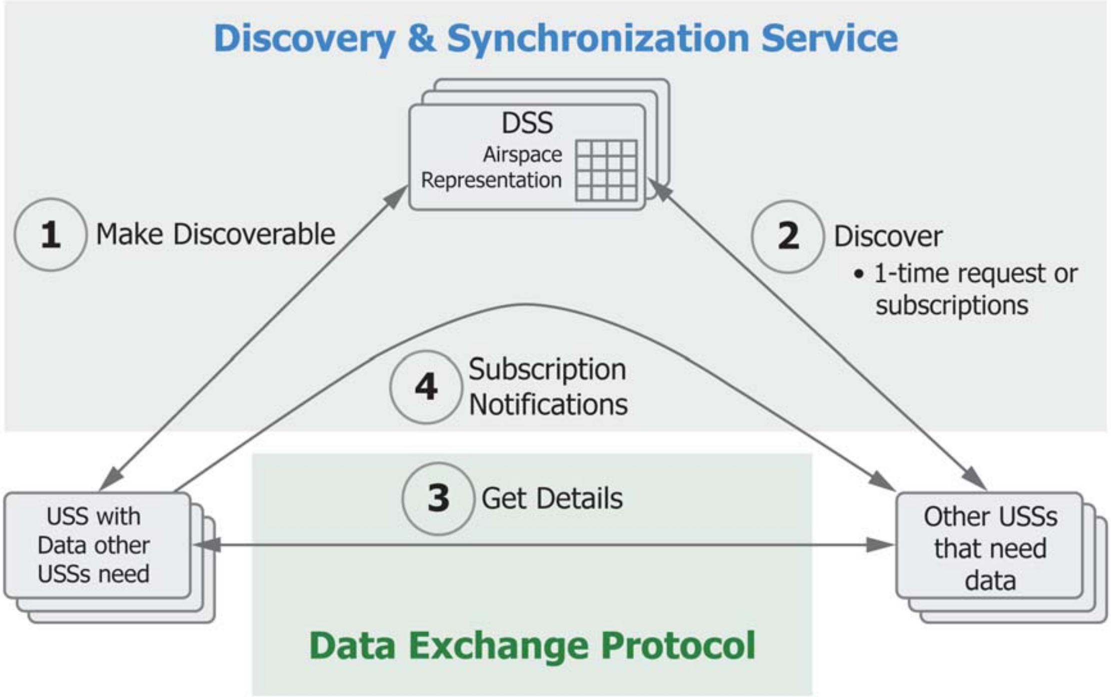

Discovery & Synchronization Service
DSS
Airspace Representation
1 Make Discoverable
2 Discover
• 1-time request or subscriptions
4 Subscription Notifications
3 Get Details
USS with Data other USSs need
Other USSs that need data
Data Exchange Protocol

FIG. 3 USS Interoperability Overview

4.8.8 An instance of discoverable information is referred to as an entity. There are different types of entities, such as operations, constraints, or, relevant to this specification, an area where network remote identification services are being provided. The concept is extendable to future UTM services where other types of entities may need to be discoverable. A key characteristic of entities is that they have an associated 4-D volume (that is, a volume defined in *x*, *y*, and *z* plus time limits).

4.8.9 The DSS encapsulates an airspace model into which entities are mapped. The implementation details of this airspace representation are hidden from DSS clients; however, conceptually, the airspace model can be thought of as a grid and mapping an entity into the airspace model is determining what grid cells the entity intersects. The complete details of the entities and any associated Personally Identifiable Information (PII) are not stored in the DSS but instead are retained by the owning USS; only limited information such as the type of entity, its location (in terms of what cells of the airspace model it intersects), the current opaque version number (OVN) of the entity (OVNs are updated whenever the entity is modified), and how to contact the owner of the entity are stored in the DSS.

4.8.10 Given that context, the primary interactions (numbered in Fig. 3) are:

4.8.10.1 *Make Discoverable*—A USS with an entity about which other USSs need to know (for example, an operation, a constraint, an area where Remote ID services are provided) makes it discoverable by writing the limited entity summary information (information type, identifier, location, owner, OVN) to the DSS.

4.8.10.2 *Discover*—Other USSs that are interested in entities of some type query the DSS using a 4-D volume to characterize the area of interest. The DSS maps the query onto the airspace representation and finds intersecting grid cells with entities of the desired type (if any). (Because entities are mapped into grid cells and the DSS does not retain the precise extents, the DSS will occasionally return an entity that does not intersect the precise area of interest; however, it will never omit an entity that intersects the area of interest.) The DSS then returns to the requesting USS a list of the discovered entities and their owners. This can be a one-time query (often described as a *pull* of the information) or the requesting USS can also establish a subscription to be notified of new or modified entities in the area of interest (discussed further below).

4.8.10.3 *Get Details*—Given the list of discovered entities, the requesting USS switches to the applicable Data Exchange Protocol to contact the owning USS and obtain the complete details. Data Exchange Protocols are service-specific.

4.8.10.4 *Subscription Notifications*—If the requesting USS established a subscription in the DSS (for a 4-D area of interest), when another USS writes a new entity to the DSS that intersects the subscription, the DSS informs the writing USS of the subscription and the writing USS contacts the subscribing USS to provide the details. (This is often described as a *push* of the information.)

4.8.11 While not needed for Remote ID, OVNs come into play on interaction if the entity written to the DSS is of a type that requires deconfliction with other entities (for example, a new UAS operation requires deconfliction; a constraint does not). When writing the new operation to the DSS, the USS must provide the OVNs for all other operations and constraints in the area of the new operation. For applicable entity types, OVNs are part of the detailed information obtained from other USSs in step 3. If the DSS determines the set of OVNs is complete and current, it allows the new operation to be written; if not, the DSS informs the writing USS what OVNs are missing or obsolete.

4.8.12 Although complete details for entities are not stored in the DSS, it serves as the single source of truth for what entities exist in the airspace and provides the mechanism necessary to ensure that USSs attempting to create a new operation have considered the current version of all other relevant entities in the airspace.

4.8.13 Unless noted otherwise, references to the DSS throughout this specification refer to the set of DSS instances supporting the region in which a related activity is occurring (for example, creating entity summaries, discovering entities).

4.8.14 This overview omits many details of the DSS and data exchange protocols. The Remote ID-specific interoperability requirements, complete APIs, and additional details are provided in Annex A2.

5. Performance Requirements

5.1 Remote ID is comprised of a set of standardized data, messages, transport mechanisms for communicating the messages, and performance requirements governing certain attributes of an implementation such as message periodicity. For Broadcast Remote ID, the message format is the same regardless of the transport mechanism. These messages are coded as a "block message" implementation of the Data Dictionary to optimize for the transport mechanism size constraints and to minimize potential broadcast interference. For Network Remote ID, the message format is a network adaption of the Data Dictionary using common internet protocols. For broadcast messages, each message has a message type that is identified in the message header. The message type defines the message format and is classified as static or dynamic, which also sets the requirements for the minimal rate at which each message type shall be transmitted. The common name for this broadcast messaging protocol is "Open Drone ID."

5.2 Conventions used in this section:

5.2.1 Requirement IDs are shown below. The prefix to each ID identifies groupings:

5.2.1.1 BURxxxx - Broadcast Update Rate

5.2.1.2 BMGxxxx - Broadcast Messages

5.2.1.3 BB4xxxx - Broadcast Bluetooth 4

5.2.1.4 BB5xxxx - Broadcast Bluetooth 5

5.2.1.5 BWFxxxx - Broadcast Wi-Fi

5.2.1.6 NETxxxx - Network

5.2.1.7 DSSxxxx - Discovery and Synchronization Service (Annex A2)

5.2.2 Constant values representing a required time, distance, etc. are consolidated into Annex A3. These values are referenced within the requirements text in this section using square brackets around the constant name. Constants pertaining to broadcast Remote ID are prefixed with “Bc” and constants pertaining to network Remote ID are prefixed with “Net” as shown in the following examples:

5.2.2.1 [BcMinUasLocRefreshRate]

5.2.2.2 [NetMinUasLocRefreshRate]

5.2.3 In some cases, notes to clarify the intent of a requirement are provided. These notes are numbered and prefaced with “Note:.” They are for clarification purposes only and do not contain requirements.

5.3 *Common Data Dictionary:*

5.3.1 Table 1 defines the required and optional data fields for Remote ID, including minimum characteristics that must be supported by both network and broadcast implementations. Since broadcast Remote ID uses size-limited messages, for some data fields it is necessary to use encoding methods that adjust the resolution or aggregate ranges of values, whereas these size-limiting techniques are not necessary for network Remote ID. The required minimum characteristics provided below ensure a prescribed degree of consistency between broadcast and network Remote ID to facilitate integration in a Remote ID display application. The specific representations for broadcast and network Remote ID are provided in their respective requirements sections.

5.3.2 An asterisk (*) adjacent to a data field name denotes an optional field. Optional fields and certain field options may be required in some jurisdictions.

5.4 *Broadcast:*

5.4.1 This section describes requirements for the RF broadcast of Remote ID messages from a participating UA. Three broadcast transport mechanisms are supported by this specification:

5.4.1.1 Bluetooth Legacy (4.x compatible)

5.4.1.2 Bluetooth 5.x Long Range (must be transmitted concurrently with Legacy mode)

5.4.1.3 Wi-Fi

5.4.2 The three transport mechanisms share common requirements for update rates and message definition.

5.4.3 *Output Power*—For output power and pattern, the requirement seeks to provide a sufficiently high power transmission that generally emits in an omni-directional pattern using commonly available components. The Minimum Tx EIRP is defined as the minimum EIRP around all 360 degrees of the far field in the Horizontal Plane of the transmission pattern. The Horizontal Plane is defined as a plane of the transmission pattern that approximately corresponds to the horizontal plane during the most common average orientation of the vehicle when flying. The Minimum EIRP over this entire plane shall (BPW0010) not be less than [BcMinEIRP] (where the applicable row is determined by national wave law in each country).

5.4.4 *Update Rates*—For broadcast messages, dynamic messages (as indicated in the block message section) shall (BUR0010) be sent at least every [BcMinUasLocRefreshRate] second(s). Static messages (as indicated in the block message section) shall (BUR0020) be sent at least every [BcMinStatic RefreshRate] second(s) and the maximum potential time elapsed since the time of applicability of the dynamic fields in the Location/Vector Message shall (BUR0030) be no older than [BcMaxDataAge]. Should channel saturation block or interfere with transmission (as may occur due to “listen before talk” interference handling technique), the system shall (BUR0040) make a “best effort” to transmit when the saturation level allows.

5.4.5 *Block Messages:*

5.4.5.1 The “Block” messages are designed to be packed into lightweight direct broadcast packets within Wi-Fi or Bluetooth “Beacon Advertisements.” The message types are identified in Table 3. Subsequent subsections further describe each message type.

5.4.5.2 Each message shall (BMG0010) be 25 bytes in length (padded with nulls as needed).

5.4.5.3 Each message shall (BMG0020) begin with a 1 byte header followed by 24 bytes of data, which shall be encoded as described in the “Message Details” table that corresponds with each Message Type described below. Non-magnitude values, strings, or IDs that may be or may not be numerical (such as the Unique ID) shall (BMG0030) be expressed in Network Byte Order which reads in a left to right, most significant byte (MSB) to least significant byte (LSB) order. Magnitude values expressed as 16 or 32 bit integers (such as Latitude, Longitude, Altitude, etc.) shall (BMG0040) be expressed as “little endian” (marked as “LE” in the “Message Details” tables below), where the LSB is on the left and the MSB is on the right. If not invoking an optional message, it is not necessary to send that message. Optional fields within messages being sent (see Table 1) shall (BMG0050) be filled in as stated in the corresponding block message format and if opting out, or the value is unknown, shall be filled with nulls (0s) for string values or 0 for numeric unless an alternate default/unknown value representation is stated in Table 1. This allows the block message to stay properly aligned with the field definitions. All ASCII Strings shall (BMG0060) be filled with nulls in the unused portion of the field. In the data structures below, some fields are enumerated values. Table 2 shall (BMG0065) be used to encode to those enumerations.

MESSAGE HEADER (Table 4)

5.4.5.4 The message header includes the Message Type and Protocol Version and shall (BMG0070) be sent in each message.

BASIC ID MESSAGE (Table 5)

5.4.5.5 Basic ID Message Type: 0x0, Static Periodicity, Mandatory

5.4.5.6 The BasicID message includes the ID Type, UA Type, and the Unique ID. This Unique ID shall (BMG0080) default to the Manufacturer Serial number. Once the UA is provisioned, the UAS ID shall be (BMG0090) one of the following:

(1) Manufacturer Serial Number expressed in the ANSI/CTA-2063-A Serial Number format.

(2) A Civil Aviation Authority (CAA) issued Registration ID for the UA formatted as described in Table 1.

(3) A UTM Assigned ID if operating within a UTM system (128-bit UUID) binary encoded, Network Byte Order.

TABLE 1 Common Data Dictionary

Data Field	Description/Rationale
UAS ID	Consists of one of three options: 1. Serial Number: This is expressed in the CTA-2063-A Serial Number format. 2. Registration ID: If a CAA provides a method of registering UAS, this number is provided by the CAA or its authorized representative. Format: <ICAO Nationality Mark[A] >.<CAA Assigned ID>, ASCII encoded, only uppercase letters (A-Z), dot (.), and digits (0-9) are allowed. Example (US): N.123456 3. UTM (UUID): A UTM-provided unique ID traceable to the Registration ID that can act like a “session id” to protect exposure of operationally sensitive information.
UAS ID Type	1. Serial Number, 2. Registration ID, or 3. UTM UUID
UA Type	The UA Type can help infer performance, speed, and duration of flights, for example, a “fixed wing” can generally fly in a forward direction only (as compared to a multi-rotor). This can also help differentiate aircraft types without sharing operationally sensitive information like the make and model of a particular aircraft. Make and model are anticipated to become available during the Registration ID lookup process. UAS Type is also useful for correlating visual observation with data received. The types were formulated based on unique flight characteristics. The possible values are in Table 2.
Timestamp	The time of applicability of position information. This may be the time coming from the source such as a GPS, or the time when the system computes the values using an algorithm such as an Extended Kalman Filter (EKF). Timestamps must be expressed with a minimum resolution[B] of one tenth of a second.
Timestamp Accuracy	Declaration of timestamp accuracy, which is the largest difference between Timestamp and true time of applicability for any of the following fields: Latitude, Longitude, Geodetic Altitude, Pressure Altitude, and Height to determine time of applicability of the location data provided. Expressed in 1/10ths of seconds. The accuracy reflects the 95 % uncertainty bound value for the timestamp.
Operational Status*[C]	Operational Status indicates whether the associated UA is on the ground or airborne. This status can be used for filtering purposes. (See Table 2.)
Operation Description*[C]	This optional, free-text field enables the operator to describe the purpose of a flight, if so desired.
Operator ID*[C]	This optional field provides a CAA-issued registration/license ID for the remote pilot or operator. Format: <ICAO Nationality Mark[A] >.<CAA Assigned ID>, ASCII encoded, only uppercase letters (A-Z), dot (.), and digits (0-9) are allowed. Example (US): N.OP123456
Latitude	Current latitude (within horizontal accuracy limits) of the UA. This is necessary to display UAS location. Minimum resolution: 7 decimal digits (~11 mm). Special Values: Invalid, No Value, or Unknown: 0 deg (both Lat/Lon)
Longitude	Current latitude (within horizontal accuracy limits) of the UA. This is necessary to display UAS location. Minimum resolution: 7 decimal digits (~11 mm). Special Values: Invalid, No Value, or Unknown: 0 deg (both Lat/Lon)
Geodetic Altitude	The aircraft distance above or below the ellipsoid as measured along a line that passes through the aircraft and is normal to the surface of the WGS-84 ellipsoid. This value is provided in meters and must have a minimum resolution of 1 m. Special Values: Invalid, No Value, or Unknown: –1000 m
Pressure Altitude*[C]	The uncorrected barometric pressure altitude (based on reference standard 29.92 inHg, 1013.25 mb) provides a reference for algorithms that utilize “altitude deltas” between aircraft. This value is provided in meters and must have a minimum resolution of 1 m. Special Values: Invalid, No Value, or Unknown: –1000 m
Height*[C]	Expressed as either height above takeoff location or height above ground level (AGL) for a UA’s current location. This value is provided in meters and must have a minimum resolution of 1 m. Special Values: Invalid, No Value, or Unknown: –1000 m
Height Type*[C]	Height above takeoff location or height above ground level.
Geodetic Vertical Accuracy	Provides quality/containment on geodetic altitude. This is based on ADS-B Geodetic Vertical Accuracy (GVA) (plus the three extra increments). (See Table 2.)
Horizontal Accuracy	Provides quality/containment on horizontal position. This is based on ADS-B NACp (plus the one extra increment). (See Table 2.)
Speed Accuracy	Provides quality/containment on horizontal ground speed. (See Table 2.)
Track Direction	Direction of flight expressed as a “True North-based” ground track angle. This value is provided in clockwise degrees with a minimum resolution of 1 degree. Special Values: Invalid, No Value, or Unknown: 361 deg
Speed	Ground speed of flight. This value is provided in meters per second with a minimum resolution of 0.25 m/s. Special Values: Invalid, No Value, or Unknown: 255 m/s, if speed is ≥254.25 m/s: 254.25 m/s
Vertical Speed	Vertical speed upward relative to the WGS-84 datum, meters per second. Special Values: Invalid, No Value, or Unknown: 63 m/s, if speed is ≥62 m/s: 62 m/s

TABLE 1 *Continued*

Data Field	Description/Rationale
Auth Data*[C]	Additional Authentication Data
Operator Latitude*[C]	Provides the location associated with the Remote Pilot. Special Values: Invalid, No Value, or Unknown: 0 deg (both Lat/Lon)
Operator Longitude*[C]	Provides the location associated with the Remote Pilot. Special Values: Invalid, No Value, or Unknown: 0 deg (both Lat/Lon)
Operator Location Type*[C]	Takeoff location, fixed location, or dynamic location representing the Operator Location
Operating Area Radius*[C]	Farthest horizontal distance from the reported location at which any UA in a group may be located (meters). Also allows defining the area where a Non-Equipped UAS Participant operation is planned or taking place. Default: 0
Operating Area Polygon*[C]	A list of latitude/longitude pairs defining the area where a group or Non-Equipped Network Participant operation is planned or taking place. (This field is only applicable to Network Remote ID.)
Operating Area Type*[C]	Cylinder or Polygon. (This field is only applicable to Network Remote ID.)
Operating Area Count*[C]	Allows for operating a single UA, group, formation, or swarm: Quantity in Group. Default 1
Operating Area Floor*[C]	Minimum altitude (WGS-84 HAE) for a group or a Non-equipped UAS Participant. Special Values: Invalid, No Value, or Unknown: –1000 m
Operating Area Ceiling*[C]	Maximum altitude (WGS-84 HAE) for a group or a Non-Equipped UAS Participant. Special Values: Invalid, No Value, or Unknown: –1000 m
Operation Area Start*[C]	The date and time at which a group or a Non-Equipped UAS Participant operation starts. (This field is only applicable to Network Remote ID.)
Operation Area End*[C]	The date and time at which a group or a Non-Equipped UAS Participant operation ends. (This field is only applicable to Network Remote ID.)

[A] ICAO Nationality Marks, https://www.icao.int/safety/airnavigation/Pages/nationality.aspx.

[B] "resolution" in this specification is used to indicate the preciseness possible of the expressed value, but not the accuracy. For example, although Lat/Lon must be expressed to 7 decimal digits of resolution, the accuracy may be far less (as indicated in the Horizontal Accuracy field).

[C] An asterisk (*) adjacent to a data field name denotes an optional field. Optional fields and certain field options may be required in some jurisdictions.

LOCATION/VECTOR MESSAGE (Table 6)

5.4.5.7 Location/Vector Message Type: 0x1, Dynamic Periodicity, Mandatory

5.4.5.8 The Location/Vector message provides the location, altitude, direction, and speed of the UA. Several of the fields require special encodings to better pack the data and to allow for more precise values. If indicated, the transmitted data shall (BMG0100) be encoded according to Table 7. Additionally, any fields that require a flag bit to be set shall (BMG0110) be set according to Table 7 as well.

AUTHENTICATION MESSAGE (Tables 8 and 9)

5.4.5.9 Authentication Message Type: 0x2, Static Periodicity, Optional* (see 5.3.2).

5.4.5.10 *Authentication Message Overview:*

5.4.5.11 The Authentication Message defines a field that provides a means for authenticating the identity of the UA sending the message. An implementation could send an Auth Type of 0 to deterministically communicate that no Auth Data is intended to be sent. Alternatively, an implementation could simply not send an Authentication message when there is no authentication data to send. Auth Types 1, 2, and 3 represent standard signature options. Auth Type 4 is used to communicate that authentication is provided by the Network Remote ID counterpart to the broadcast. Custom authentication implementations can be created using Auth Types A-F. The details of a custom implementation are beyond the scope of this specification.

5.4.5.12 For Auth Types 1, 2, and 3, the standard provides flexibility to allow a multitude of signature formats that are not specified in this specification. The intended implementation is that an agreed upon signature format for each Auth Type required will be shared by both the signature encoding software and the verifier software. This specification specifies an API for a receiver (for example, a Remote ID Display Application) to communicate with a verifier. (See Annex A1 for additional details.)

5.4.5.13 *Authenticate Message Requirements:*

5.4.5.14 If no authentication is used, and this message is still being sent (which is not required), the Auth Type shall be set to 0 (BMG0120) and the Signature shall be empty. When a signature is required, the signature produced by a UAS shall (BMG0130) be encoded to match the signature format expected by the verifier. When UAS ID (1) or Operator ID (2) is set as the AuthType, then the Message Signature shall (BMG0140) include the corresponding data and TimeStamp from the Authentication message in the signature. If the AuthType is set to Message Set (3), then the Signature shall (BMG0150) include the concatenation (in message type order) of all other transmitted message types (excluding this Authentication message), and TimeStamp from this Authentication message.

TABLE 2 Enumerated Field Definitions

Field Name	Details	Notes
UA Type	0: None/Not Declared 1: Aeroplane 2: Helicopter (or Multirotor) 3: Gyroplane 4: Hybrid Lift (Fixed wing aircraft that can take off vertically) 5: Ornithopter 6: Glider 7: Kite 8: Free Balloon 9: Captive Balloon 10: Airship (such as a blimp) 11: Free Fall/Parachute (unpowered) 12: Rocket 13: Tethered Powered Aircraft 14: Ground Obstacle 15: Other	Up to 16 Types These values were derived from the official ICAO UA Type list. Additional types were added if they had unique flight characteristics.
Operational Status	0: Undeclared 1: Ground 2: Airborne 3-15: Reserved	Up to 16 Statuses
Horizontal Accuracy	0: ≥18.52 km (10 NM) or Unknown 1: <18.52 km (10 NM) 2: <7.408 km (4 NM) 3: <3.704 km (2 NM) 4: <1852 m (1 NM) 5: <926 m (0.5 NM) 6: <555.6 m (0.3 NM) 7: <185.2 m (0.1 NM) 8: <92.6 m (0.05 NM) 9: <30 m 10: <10 m 11: <3 m 12: <1 m 13-15: Reserved	This is the NACp enumeration from ADS-B. Value 12 was added for a more complete range for UAs. 95 % accuracy bound (estimated position uncertainty).
Vertical Accuracy	0: ≥150 m or Unknown 1: <150 m 2: <45 m 3: <25 m 4: <10 m 5: <3 m 6: <1 m 7-15: Reserved	This is the GVA enumeration from ADS-B. Values 4–6 were added for UAs. 95 % accuracy bound.
Speed Accuracy	0: ≥10 m/s or Unknown 1: <10 m/s 2: <3 m/s 3: <1 m/s 4: <0.3 m/s 5-15: Reserved	This is the same enumeration scale and values from ADS-B NACv. 95 % accuracy bound.

TABLE 3 Open Drone ID Block Message Summary

Msg Type	Message Name	Purpose
0x0	Basic ID Message	Provides ID for UA, characterizes the type of ID, and identifies the type of UA
0x1	Location/Vector Message	Provides location, altitude, direction, and speed of UA
0x2	Authentication Message[A]	Provides authentication data for the UA
0x3	Self-ID Message[A]	Message that can be used by Operators to identify themselves and the purpose of an operation
0x4	System Message[A]	Includes Remote Pilot location and multiple aircraft information (group) if applicable, and additional system information
0x5	Operator ID[A]	Provides Operator ID
0xF	Message Pack[A]	A payload mechanism for combining the messages above into a single message pack. Used with Bluetooth Extended Advertising and Wi-Fi Neighbor Awareness Network

[A] Optional unless required by location jurisdiction.

5.4.5.15 The Authentication Message Data Page field allows for Authentication Data sizes that may exceed the 24 bytes available per message. The Data Page shall (BMG0160) be incremented (starting from 0) for each additional message required to complete the oversized message. AuthType 3 (Message Set) shall (BMG0170) only be used when the

TABLE 4 Message Header Details

Header (1 byte)		Message (24 bytes)
Message Type (4 bits) Bits [7..4]	Protocol Version (4 bits) Bits [3..0]	Message Fields based on Message Type
0x0–0xF	0x0	< Message Data >

TABLE 5 Basic ID Message Details

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	ID Type, UA Type	Bits 7..0 [0000] [0000] ID Type: Bits [7..4] 0: None 1: Serial Number (ANSI/CTA-2063-A) 2: CAA Assigned Registration ID 3: UTM Assigned UUID UA Type: Bits [3..0] VTOL, fixed wing, hybrid, etc. (See Table 2 for more details.)	Up to 16 ID types Up to 16 UA types	
2	20	UAS ID	UAS ID within the format of ID Type (padded with nulls)	Max. 20 Bytes	N.123456
22	3	Reserved			

transport media can send all of the pages together, such as Bluetooth 5 or Wi-Fi. If the AuthType is 4 (Network Remote ID), then the Authentication Data/Signature (BMG0180) shall be empty (all nulls).

SELF-ID MESSAGE (Table 10)

5.4.5.16 Self ID Message Type: 0x3, Static Periodicity, Optional* (see 5.3.2).

5.4.5.17 The Self-ID Message is an opportunity for the Remote Pilot to (optionally) declare their identity or purpose (intent) of the flight, or both. This message can serve to reduce the perceived threat of a UA flying in a particular area or manner. For example: to put neighbors at ease, a realtor may declare a "property photo shoot" of a client's house. This is a free-form (ASCII) text field.

SYSTEM MESSAGE (Table 11)

5.4.5.18 System Message Type: 0x4, Static Periodicity, Optional* (see 5.3.2).

5.4.5.19 The System Message contains general system information including information about the Remote Pilot location and flight area. If the GCS has a dynamic location source (for example, GNSS), then the Operator Location fields shall (BMG0190) be the current location information of the GCS as acquired from the dynamic source. If the GCS cannot obtain dynamic location data, then the Operator Location fields shall (BMG0200) be the aircraft's takeoff location. Since this value generally does not change at the same rate as a UA location, the minimum update frequency shall (BMG0210) be the same as static messages. If a group of aircraft is being represented, the number of aircraft, radius of flight area centered on the Location/Vector Message latitude/longitude, and group operations ceiling and floor shall (BMG0220) be expressed in this message using the Area fields. If one or more UA are non-equipped, the Area fields shall (BMG0230) be used to declare (by means of broadcast messages compliant with this section) a volume of operation by a device external to the UA (such as a ground station) centered on the Location/Vector Message latitude/longitude. If the value for one or more fields is unknown, that field shall (BMG0240) be filled as specified in Table 1.

OPERATOR ID MESSAGE (Table 12)

5.4.5.20 Operator ID Message Type: 0x5, Static Periodicity, Optional* (see 5.3.2).

5.4.5.21 The Operator ID Message contains the the CAA issued Operator ID formatted as described in Table 1.

MESSAGE PACK (Table 13)

5.4.5.22 Message Pack Message Type: 0xF, Dynamic Periodicity if dynamic message in contents

5.4.5.23 The Message Pack is a means to send multiple messages defined in this section together as a single larger message. The media for sending such messages are those that support larger payloads (such as Bluetooth 5 and Wi-Fi methods).

5.4.6 *Bluetooth Legacy (4.x compatible) Transport Method:*

5.4.6.1 Bluetooth 4.x (and newer) is widely deployed across diverse commonly carried handheld devices and provides mechanisms for low bandwidth beacons. The implementation method utilizes the existing "advertising beacon messages" that are commonly used to declare a device (such as a headphone or mouse) available for pairing. Now that Bluetooth 5 has introduced an "Extended Advertising" method, Bluetooth 4.x method is called "Legacy Advertising."

5.4.6.2 As illustrated in Fig. 4, the most common Wi-Fi channels that are generally preprogrammed into routers are 1, 6, and 11 because they do not overlap. Bluetooth channels (as illustrated with the blue bars) are much narrower than Wi-Fi channels. Bluetooth uses three different beacon channels (in

TABLE 6 Location/Vector Message Details

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	Status, Flags	Bits [7..0] [0000] [0000] Status: Bits [7..4] Flags Reserved: Bit [3] Height Type: Bit [2] 0: Above Takeoff, 1: AGL E/W Direction Segment: Bit [1] 0: <180, 1: ≥180 Speed Multiplier: Bit [0] 0: x0.25, 1: x0.75	0..15 statuses (See Table 2) Speed Multiplier enables speeds up to 254.25 m/s. Only use 1 when speed exceeds 63.75 m/s and add 63.75.	
2	1	Track Direction	Direction expressed as the route course measured clockwise from true north. Encode as 0–179. If E/W Direction bit is set, then 180 should be added to the value.	0–359 Unsigned Int (UInt)	10 with E/W bit set = 190 deg.
3	1	Speed	Ground Speed in m/s encloded as specified in Table 7	Up to 254.25 m/s UInt	20(enc)= 5 m/s
4	1	Vertical Speed	Vertical Speed m/s (+ up, – down) Multiplier = 0.5	Up to ±63.5 m/s (12.5k ft/min)	15(enc)=7.5 m/s
5	4	Latitude	Latitude of UA deg*10^7	Int signed (LE)	–11989298
9	4	Longitude	Longitude of UA deg*10^7	Int signed (LE) (11 mm precision)	48123987
13	2	Pressure Altitude	Pressure Altitude (Ref 29.92 inHg, 1013.24 mb) (Altitude + 1000 m)/0.5 (LE)	–1000–31767 m (107503 ft) 16 bit UInt (LE)	2021 (enc) = 10.5 m
15	2	Geodetic Altitude	WGS-84 HAE (Altitude + 1000 m)/0.5	–1000–31767 m (107503 ft) 16 bit UINT (LE)	2021 (enc) = 10.5 m
17	2	Height	Height above takeoff location or Height above ground (indicate with Height Type Bit) (Altitude + 1000 m)/0.5	–1000–31767 m (107503 ft) 16 bit UInt (LE)	2021 (enc) = 10.5 m
19	1	Horizontal/ Vertical Accuracy	Bits [7..0] [0000] [0000] Vertical (Geodetic): Bits [7..4] Horizontal: Bits [3..0] Vertical: See Vertical Accuracy Enumeration Horizontal: See Horizontal Accuracy Enumeration	See Table 2	
20	1	Baro Altitude Accuracy/Speed Accuracy	Bits [7..0] [0000] [0000] Baro Altitude: Bits [7..4] See Vertical Accuracy Enumeration Speed: Bits [3..0] Based on Extended ADS-B NACv	Baro: see Table 2 Speed: see Table 2	
21	2	Timestamp	Time of applicability expressed in 1/10ths of seconds since the last hour	0–36000: 16 Bit UInt (LE)	3611 = 6 mins, 1.1 s after the hour
23	1	Reserved/ Timestamp Accuracy	Bits [7..0] [0000] [0000] Reserved: Bits [7..4] Timestamp accuracy: Bits [3..0] (*0.1 s stepping resolution)	Timestamp accuracy: 0.1 s–1.5 s 0=unknown	
24	1	Reserved			

TABLE 7 Encoding Table

Field Type	To Encode (sender)	To Decode (receiver)

Direction	If Value <180
	EncodedValue = Value
	Set Direction Segment bit to 0
	
	else
	EnclodedValue = Value – 180
	Set Direction Segment bit to 1
		if Direction Segment bit = 0
		Value = EncodedValue
		
		else
		Value = EncodedValue + 180

Speed (UInt8)	If Value- ≤ 225*0.25
	EncodedValue = Value/0.25[A]
	Set Multiplier Flag to 0
	else if Value > 225*0.25 and Value <254.25
	EncodedValue = (Value – (225*0.25))/0.75[A]
	Set Multiplier Flag to 1
	else (Value ≥ 254.25 m/s)
	EncodedValue = 254[A]
	Set Multiplier Flag to 1
		If Multiplier Flag = 0
		Value = EncodedValue * 0.25
		
		else if Multiplier Flag = 1
		Value = (EncodedValue * 0.75) + (255*0.25)[B, C]
		
		Encoding Rationale: This allows for a higher speed precision of 0.25 m/s for lower speeds (0.5 kts) and 0.75 m/s (1.5 kts) at higher speeds.

Lat/Lon	(Int32) EncodedValue = Value * 10^7 Default/Unknown: 0,0	(Double) Value = EncodedValue / 10^7
Vertical Speed	(Int8) EncodedValue = Value / 0.5	(Float) Value = EncodedValue * 0.5
Altitude	(UInt16) EncodedValue = (Value + 1000) / 0.5 Unknown: –1000, encode as 0	(Float) Value = (EncodedValue * 0.5) – 1000 Encoding Rationale: Eliminated unused negative integer space and increases precision to 1/2 m If decoded Value = –1000, then real value is unknown
Time Stamp	(UInt16) Encoded Value = Tenths of seconds since current hour	if Encoded Value > Tenths of seconds since the current hour at time of receipt, then ValueTenths = tenths of seconds since previous hour else ValueTenths = tenths of seconds since current hour Value = Current GMT Date/Time + ValueTenths This is the “Time of Applicability”

[A] Encoded Value must be rounded to nearest Integer.
[B] If value decodes to 255, then an unknown value is being represented.
[C] If value decodes to 254.25, then speed is at least 254.25.

TABLE 8 Authentication Message Page 0 Details

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	AuthType, Page Number	Bits [7..0] [0000] [0000] Auth Type: Bits [7..4] 0: None 1: UAS ID Signature 2: Operator ID Signature 3: Message Set Signature 4: Authentication Provided by Network Remote ID 5-9: Reserved for Spec A-F: Available for Private Use Data Page Number: Bits [3..0]	Up to 16 Types, Page number must be 0	
2	1	Page Count	Bits [7..0] [0000] [0000] Reserved: Bits [7..4] Total Page Count: Bits [3..0] Start at 0	Up to 5 pages	
3	1	Length (bytes)	Total Data Length of the concatenation of all Authentication Data/Signature fields from all authentication pages	0 to 109 bytes 17 + 23 * 4	
4	4	Timestamp	32 bit Unix Timestamp in seconds since 00:00:00 01/01/2019 (to relate back to standard unix timestamp, add 1546300800 to base it on 00:00:00 01/01/1970)	Up to 01/19/2087 at 3:14 am (UTC)	
8	17	Authentication Data/Signature	Opaque Authentication Data (Signature)	0–5 pages	

TABLE 9 Authentication Message Pages 1 through 4 Details

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	AuthType, Page Number	Bits [7..0] [0000] [0000] Auth Type: Bits [7..4] 0: None (not necessary to send if 0) 1: UAS ID Signature 2: Operator ID Signature 3: Message Set Signature 4: Authentication Provided by Network Remote ID 5-9: Reserved for Spec A-F: Available for Private Use Page number: Bits [3..0]	Up to 16 Types, Page numbers 1–4	
2	23	Authentication Data/Signature	Opaque Authentication Data (Signature)	23 Bytes per additional page	

TABLE 10 Self-ID Message Details

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	Description Type	0: Text Description 1–200: Reserved 201–255: Available for private use	0–255	0
2	23	Description	ASCII Text. If numeric values exist, they shall be expressed as a string of ASCII characters (padded with nulls)	23 Bytes	DronesRus:Survey

TABLE 11 System Message Details

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	Flags	Bits [7..0] [00000000] Reserved: Bits [7..2] Operator Location type: Bits [1–0] 0 = Take Off 1 = Live GNSS 2 = Fixed Location		
2	4	Operator Latitude	Latitude of Remote Pilot	Int signed deg*10^7 (LE)	–11989298
6	4	Operator Longitude	Longitude of Remote Pilot	Int signed deg*10^7 (11 mm precision) (LE)	48123987
10	2	AreaCount	Number of aircraft in Area, group or formation (default 1)	Up to 65 000 (LE)	
12	1	Area Radius	Radius of cylindrical area of group or formation * 10 m (default 0) centered on Location/Vector Message position	Up to 2.5 km	
13	2	Area Ceiling	Group operations ceiling WGS-84 HAE (Altitude + 1000 m)/0.5	–1000–31767 m (107503 ft) 16 bit UInt (LE)	2021 (enc) = 10.5 m
15	2	Area Floor	Group operations floor WGS-84 HAE (Altitude + 1000 m)/0.5	–1000–31767 m (107503 ft) 16 bit UInt (LE)	2021 (enc) = 10.5 m
17	8	Reserved	Reserved for future use		

orange - 37, 38, and 39) to broadcast messages to non-specific endpoints (connectionless). Although the remaining 37 channels operate in the 2.4 GHz range, proximate to where Wi-Fi resides, the beacon channels are generally outside of typical Wi-Fi traffic bands.

5.4.6.3 *Beacon Messages*—Bluetooth supports a "Broadcast Frame" to transmit on the beacon channels with a custom message length limit of 31 bytes. This leaves 25 bytes (after certain header info) available for Open Drone ID messages. These broadcast messages shall (BB40010) be "un-coded" and conform to Bluetooth Core Specification 5.0, Volume 6, Part B, Sections 2.1 and 2.3.1. After each frame of the same type is sent, the AD Counter (for that message type) shall (BB40020) be incremented and reset back to 0 after 0xFF is reached.

5.4.6.4 *Frame Details*—Legacy Advertising Frames shall (BB40030) be encoded as illustrated below in Fig. 5 and Table 14.

TABLE 12 Operator-ID Message Details

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	Operator ID Type	0: Operator ID 1–200: Reserved 201–255: Available for private use	0–255	0
2	20	Operator ID	ASCII Text. If numeric values exist, they shall be expressed as a string of ASCII characters (padded with nulls)	Up to 20 Bytes	
22	3	Reserved	Reserved		

TABLE 13 Message Pack Message Details

Offset (Byte)	Length (Bytes)	Data Field	Details	Limitations	Example
1	1	Message Size	Size of single message in message pack. Set to 0x19 (25).	0x19 (25)	0x19 (25 bytes)
2	1	No of Msgs in Pack (N)	Number of messages (N) contained in message pack	Up to 10	5
3	N*25	Messages [hdr][msg][hdr][msg] [...]	Series of concatenated messages in message number order. Each message contained herein starts with a header as described in 5.4.5.4.	Up to 250 bytes	

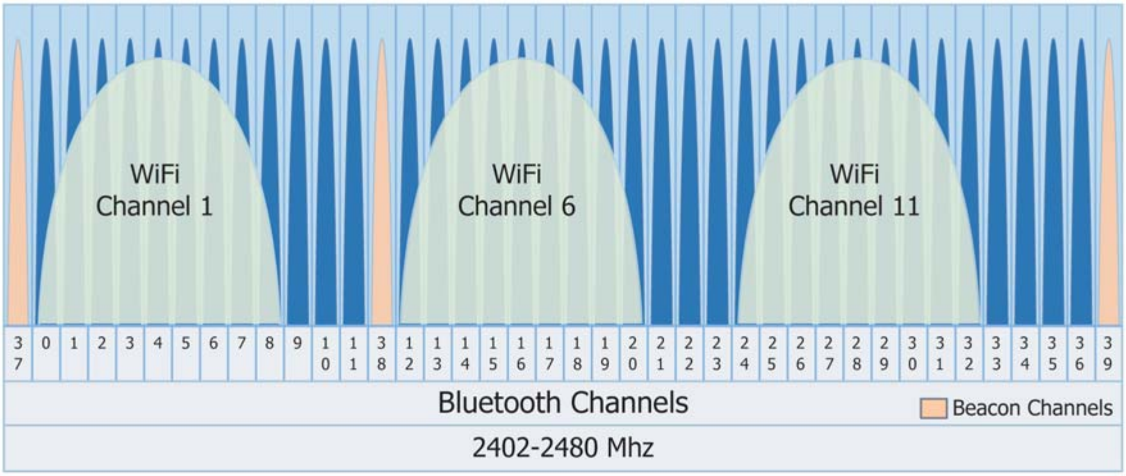
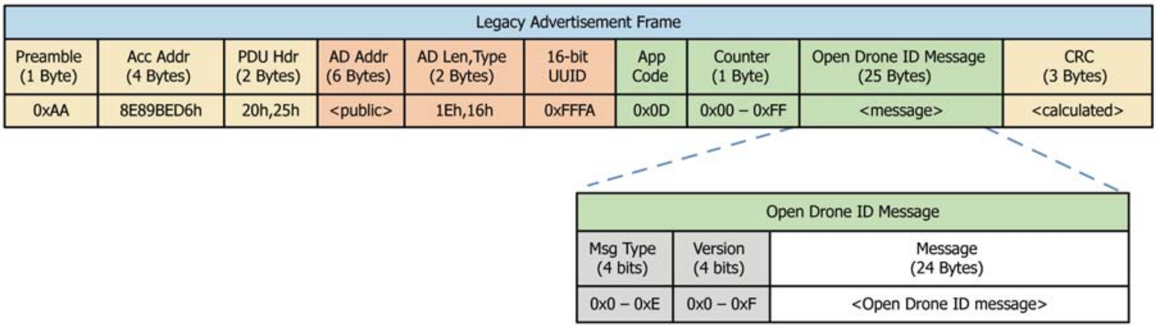


FIG. 4 Bluetooth Channels



Legacy Advertisement Frame

Preamble (1 Byte)	Acc Addr (4 Bytes)	PDU Hdr (2 Bytes)	AD Addr (6 Bytes)	AD Len,Type (2 Bytes)	16-bit UUID	App Code	Counter (1 Byte)	Open Drone ID Message (25 Bytes)	CRC (3 Bytes)
0xAA	8E89BED6h	20h,25h	<public>	1Eh,16h	0xFFFA	0x0D	0x00 – 0xFF	<message>	<calculated>

Open Drone ID Message

Msg Type (4 bits)	Version (4 bits)	Message (24 Bytes)
0x0 – 0xE	0x0 – 0xF	<Open Drone ID message>

FIG. 5 Legacy Advertisement Frame

5.4.7 *Bluetooth 5 Long Range Transport Method*—Bluetooth 5 allows for several new features over Bluetooth 4.x. For the purpose of this application, the most important new features are Long Range Mode (LE Coded) and Advertising Extensions (allowing for larger connectionless broadcasts).

TABLE 14 Legacy Advertisement Frame Details

Field	Size	Value	Contents		
Preamble	1	0xAA	LE 1M Packet		
Acc Address	6	0x8E89BED6	Broadcast Packet		
PDU Hdr	2	0x2025	PDU Type	0x2	ADV_NONCONN_IND – Connectionless Advertisement
			RFU	0	Reserved
			ChSel	0	Reserved
			TxAdd	0	Indicates AD Addr is HW Address (rather than random)
			RxAdd	0	Reserved
			Len	0x25	37 Bytes
AD Addr	6	0xXXXXXX	Unique Hardware Address of Bluetooth MAC		
AD Info	4	1Eh, 16h, 0xFFAA	Length	0x1E	30 Bytes (excluding this field)
			Type	0x16	Service Data[A]
			Mfg Code	0xFFFa	ASTM (Little Endian (FA,FF))[B]
AD App	1	0x0D	Application Code: 0x0D = Open Drone ID		
AD Counter	1	0xXX	Msg Counter: Start at 0 for first message sent, increment for each message of the same type. Roll over back to 0 after 0xFF.		
ODID Msg	25	<25 Bytes>	Open Drone ID Message		
CRC	3	<calculated>	CRC Error Correction Data as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.1.1		

[A] See https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile/.

[B] See https://www.bluetooth.com/specifications/assigned-numbers/16-bit-uuids-for-sdos/.

Bluetooth 5 General Requirements

5.4.7.1 If implementing this specification using Bluetooth 5 Long Range, Legacy (ADV_NONCONN_IND) advertisements must (BB50010) be sent, as described in 5.4.6, for backwards compatibility with less capable receivers. Bluetooth 5 Extended Advertisements (ADV_EXT_IND + AUX_ADV_IND) must (BB50020) be sent as well at the same rate as Dynamic Data (see 5.4.4 Update Rates) and they must (BB50030) be sent on a LE Coded (S=8) PHY. This will add Forward Error Correction (FEC) and can increase the range of the advertisements by a factor of 4. These messages shall (BB50040) conform to Bluetooth Core Specification 5.0, Volume 6, Part B, Section 2.2 (LE Coded PHY, S=8).

5.4.7.2 While Legacy Advertisements broadcast on the beacon channels 37, 38, 39, Bluetooth 5 adds Extended Advertising that allows for up to 255 byte advertisements on the "non-beacon" channels by implementing a pointer in the primary beacons directing the receiver to read from the secondary channel.

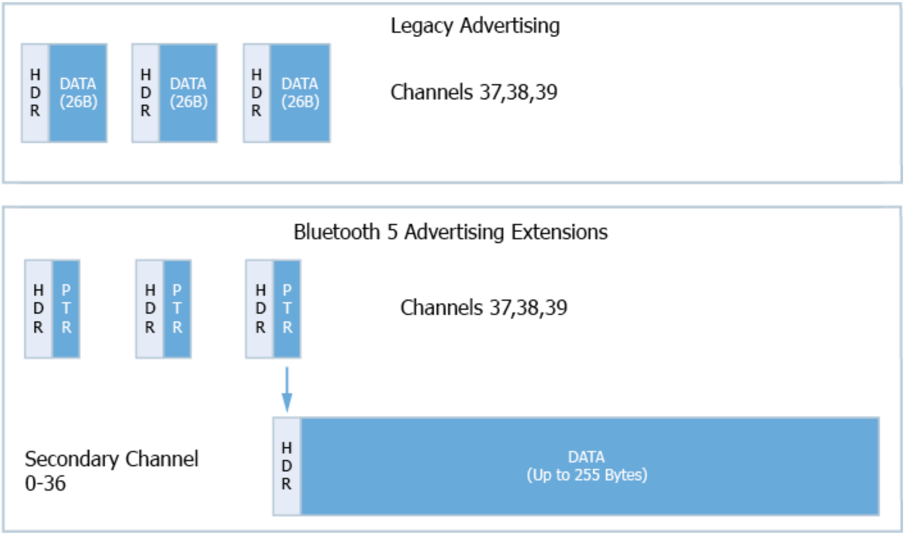


FIG. 6 Bluetooth Advertising

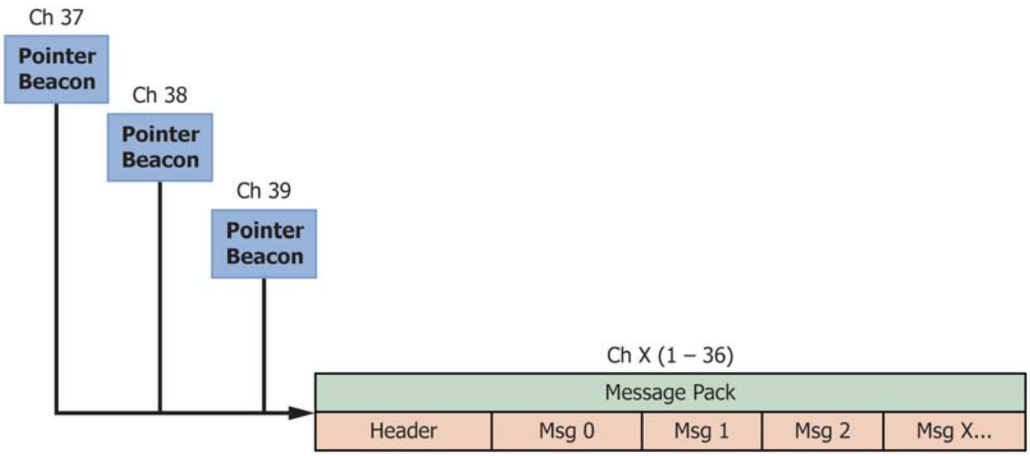


FIG. 7 Bluetooth Extended Advertisement

Bluetooth 5 Long Range Advertisement Pointer Frame (LE Coded)										
Preamble (1 Byte) [Coded Phy]	Acc Addr (4 Bytes)	CI (2 bits) [S=8]	TERM1 (3 bits)	PDU Hdr (2 Bytes)	Ext Hdr Len (6 Bits)	Adv Mode (2 Bits) non-scan undirect	Ext Header (12 Bytes)	Adv Data	CRC (3 Bytes)	Term2 (3 bits)
0x3C	0x8E89BED6	00b	<xxx>b	0x700D	0x30 (12,0)		<12 bytes>	N/A	<calculated>	<xxx>b

FIG. 8 Bluetooth 5 Long Range Advertisement Primary Packet Frame

Bluetooth 5 Extended Advertisement Primary (Pointer) Packet

5.4.7.3 The Bluetooth 5 Extended Advertisement Primary Packet includes a pointer to the Secondary Packet as illustrated in Figs. 6 and 7. Therefore, the Primary packet shall (BB50050) be broadcast through all 3 beacon channels, followed by the Secondary packet on the remaining channels.

5.4.7.4 The Pointer Frame shall (BB50060) be encoded as described in Fig. 8 and Table 15.

Aux Ptr Field Details

5.4.7.5 The Aux Ptr Field in the Primary Packet shall (BB50070) be implemented in accordance with Bluetooth Core Specification 5.0, Volume 6, Part B, Section 2.3.4.5 with the following guidance in Table 16.

Bluetooth 5 Advertising Extension Secondary Packet

5.4.7.6 The secondary packet contains the actual information payload. This packet shall (BB50100) be encoded according to the Common Extended Advertising Format described in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 2.3.4 with the values included in Fig. 9 and Table 17.

5.4.7.7 Additionally this packet Adv Data payload shall (BB50110) group together all message types being sent into a single "Message Pack" (Message Type 0xF) of the format described in Fig. 9 and in Table 13. No more than 10 messages shall (BB50120) be included in a Message Pack.

5.4.8 *Wi-Fi Transport Method:*

5.4.8.1 As detailed in this specification, a connectionless broadcast mechanism can be implemented using Wi-Fi Management Frames encapsulating Open Drone ID messages. For better interoperability with handheld device SDKs, Messages shall (BWF0010) be encoded within the Service Discovery Frame based on the Neighbor Awareness Networking (NAN) Specification.[10,5] This solution does not require connecting to any specific wireless network since (on the receiver) it utilizes the mechanism that simply listens for Wi-Fi broadcasts and makes the data available for display. These broadcasts are implemented on 2.4 GHz and optionally on 5 GHz Wi-Fi.

Wi-Fi Management Frames

5.4.8.2 For UAS implementing this protocol broadcast frame, a "management" (type 0), "action" (subtype 13) frame as prescribed by the IEEE 802.11-2016 Part 11 Wi-Fi specification[7,5] shall (BWF0020) be encoded as NAN Service Discovery Frames as described in the NAN Specification. Additionally, the values shall (BWF0030) be filled as described in the NAN Service Discovery Frame Diagram in Fig. 10 and NAN Service Discovery Frame Details in Table 18.

5.4.8.3 All message types being sent shall (BWF0040) be sent together in a single message pack and shall (BWF0050) be sent at the dynamic message rate.

Neighbor Awareness Networking (NAN)

5.4.8.4 Neighbor Awareness Networking (NAN) is the underlying specification used by Wi-Fi Aware,[14] a standards-based approach to proximity detection of devices or services, or both, that incorporates capabilities enhancing peer-to-peer communications by enabling devices to exchange services and information without the need for network infrastructure or sophisticated setup processes.

[14] A trademark of Wi-Fi Alliance, 10900-B Stonelake Boulevard, Suite 126, Austin, TX 78759.

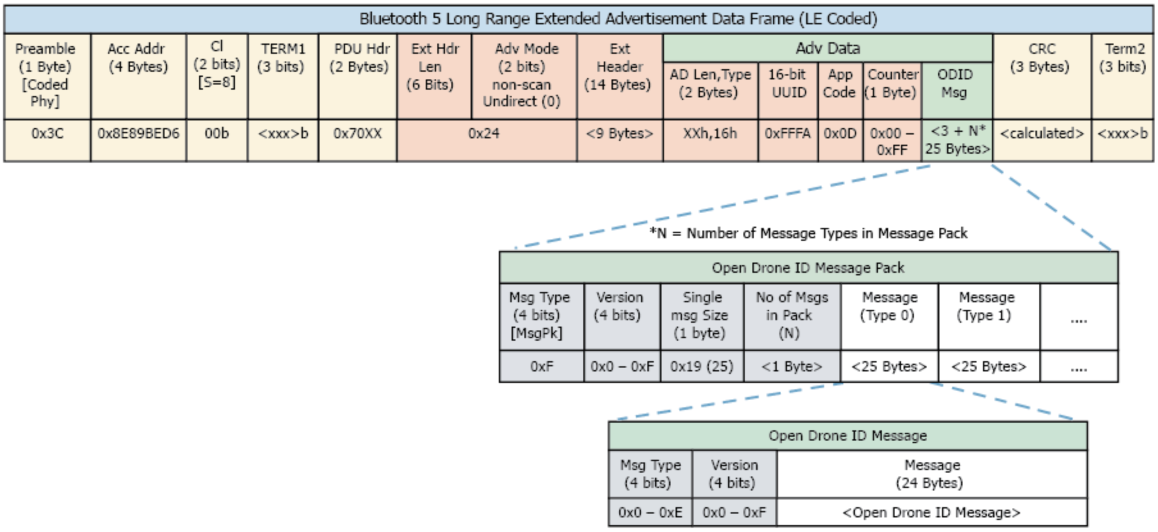
TABLE 15 Additional Primary Packet Frame Details

Name	Size (Bytes)	Value	Value Description				
Preamble	1	0x3C	LE Coded PHY				
Acc Addr	6	0x8E89BED6	Broadcast Packet				
CI	2 bits	00b	Coding Indication: FEC Block 2 is coded using S=8 (longest range)				
Term1	3 bits	xxxb	FEC Block 1 Termination as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.3.1				
PDU Hdr	2	0x700D	Field	Bits	Hex	Desc	
			PDU Type	0111	0x7	ADV_EXT_IND (Primary)	
			RFU	0	0x0	Reserved	
			ChSel	0		Reserved	
			TxAdd	0		Reserved	
			RxAdd	0		Reserved	
			Len	0010 0101	0x0D	13 Bytes	
Ext Hdr Len,	1	0x30	Bits [7..0]: [001100] [00]				
Adv Mode	6 Bits		Bits [7..2]: Ext Header Len: 12 bytes				
	2 Bits		Bits [1..0]: Adv Mode 0x0, Non-connectable, Non-scannable with Aux Pkt				
Ext Hdr	12		Field	Size	Bits (binary)	Hex	Desc
			Flags	1	0001 1001	0x19	Field Selection (AdvA, ADI, Aux Ptr)
			AdvA	6	<HW ADDR>	0xXXXXXX	Adv Address (HW Addr)
			ADI	2	0000 0000 0000 xxxx	0x0000 – 0x000F	Advertising Data ID (12 bits) = 0 Advertising Set ID (4 bits): Increment each time data changes
			Aux Ptr	3	cccc cca0 dddd dddd dddd d010	0xXXXXXX	cccccc = Channel a = clock accuracy 0 = 30 us offset multiplier dddddd = offset/delay 010 = LE Coded Phy (See Table 16)
Adv Data	N/A	0	Not Populated for this message				
CRC	3		CRC Error Correction Data as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.1.1				
Term2	3 bits		FEC Block 2 Termination as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.3.1				

TABLE 16 Aux Pointer Field Details

Channel Index	Shall (BB50080) be calculated using the following formula: Channel = (Current Channel + 9) % 36 This will ensure some entropy by hopping through the channels and spreading out the beacons to minimize the effects of external interference.
Clock Accuracy (CA)	0: 51–500 ppm 1: 0–50 ppm
Offset Units	0: 30 us
Aux Offset/Delay	This represents the time offset between sending the primary and the secondary packet. Since all three Primary Packets are sent prior to the secondary packet, the offset is different for each one. This offset should be calculated based on the guidance given in Bluetooth Core Specification 5.0. The following offsets may be used as guidance: Beacon 1: 166 us Beacon 2: 114 us Beacon 3: 62 us These calculations are based on a Primary Packet time of 1552 us + a T_MAFS (minimum aux frame space) of 300 us divided by the offset multiplier unit of 30 us. The time of sending the current beacon + remaining beacons must (BB50090) be included. Thus, Beacon 1 includes the time of itself + 2 more beacons + T_MAFS.offset is different for each one.
Aux PHY	010: LE Coded Phy

5.4.8.5 This specification uses the NAN Service Discovery Frames to broadcast Service Information that carry the messages defined by the standard using an (unsolicited) publishing protocol according to the NAN Specification.



Bluetooth 5 Long Range Extended Advertisement Data Frame (LE Coded)														
								Adv Data						
Preamble (1 Byte) [Coded Phy]	Acc Addr (4 Bytes)	CI (2 bits) [S=8]	TERM1 (3 bits)	PDU Hdr (2 Bytes)	Ext Hdr Len (6 Bits)	Adv Mode (2 bits) non-scan Undirect (0)	Ext Header (14 Bytes)	AD Len,Type (2 Bytes)	16-bit UUID	App Code	Counter (1 Byte)	ODID Msg	CRC (3 Bytes)	Term2 (3 bits)
0x3C	0x8E89BED6	00b	<xxx>b	0x70XX	0x24		<9 Bytes>	XXh,16h	0xFFFA	0x0D	0x00 – 0xFF	<3 + N* 25 Bytes>	<calculated>	<xxx>b

*N = Number of Message Types in Message Pack

Open Drone ID Message Pack						
Msg Type (4 bits) [MsgPk]	Version (4 bits)	Single msg Size (1 byte)	No of Msgs in Pack (N)	Message (Type 0)	Message (Type 1)	....
0xF	0x0 – 0xF	0x19 (25)	<1 Byte>	<25 Bytes>	<25 Bytes>	....

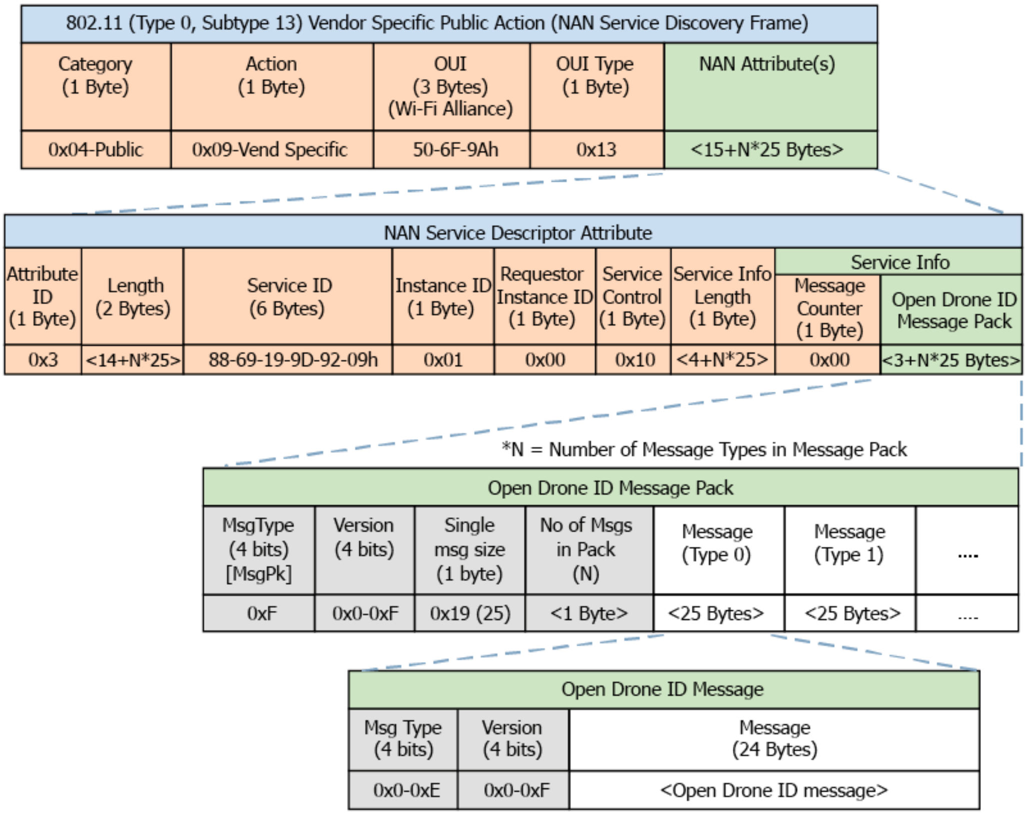
Open Drone ID Message		
Msg Type (4 bits)	Version (4 bits)	Message (24 Bytes)
0x0 – 0xE	0x0 – 0xF	<Open Drone ID Message>

FIG. 9 Bluetooth 5 Extended Advertising Secondary Packet

TABLE 17 Bluetooth 5 Extended Advertising Secondary Packet Details

Name	Size	Value	Value Description				
Preamble	1	0x3C	LE Coded PHY				
Acc Addr	6	0x8E89BED6	Broadcast Packet				
CI	2 bits	00b	Coding Indication: FEC Block 2 is coded using S=8 (longest range)				
Term1	3 bits	xxxb	FEC Block 1 Termination as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.3.1				
PDU Hdr	2	0x70XX	Field	Bits	Hex	Desc	
			PDU Type	0111	0x7	AUX_ADV_IND (Secondary)	
			RFU	0	0x0	Reserved	
			ChSel	0		Reserved	
			TxAdd	0		Reserved	
			RxAdd	0		Reserved	
			Length	xxxx xxxx	0xXX	18 + N*25 Bytes where N is the number of Messages in the Message Pack	
Ext Hdr Len, Adv Mode	1 6 Bits 2 Bits	0x24	Bits [7..0]: [001001] [00] Bits [7..2]: Ext Header Len: 9 Bits [1..0]: Adv Mode 0x0, Non-connectable, Non-scannable				
Ext Hdr	9		Field	Size	Bits (binary)	Hex	Desc
			Flags	1	0000 1001	0x09	Field Selection (AdvA, ADI)
			AdvA	6	<HW ADDR>	0xXXXXXX	Adv Address (HW Addr)
			ADI	2	xxxx xxxx xxxx yyyy	0x0000 – 0xFFFF	Advertising Data ID (12 bits) = 0 Increment each time data changes Advertising Set ID (4 bits): Undefined (not used)
AD Info	4	1Eh, 16h, 0xFFFA	Length	0x1E	30 Bytes (excluding this field)		
			Type	0x16	Service Data[A]		
			16-bit UUID	0xFFFA	ASTM (Little Endian (FA,FF))[B]		
App Code	1	0x0D	Application Code: 0x0D = Open Drone ID				
Counter	1	0xXX	Msg Counter: Start at 0 for first message sent, increment for each message of the same type. Roll over back to 0 after 0xFF.				
ODID Msg	0xXX		Open Drone ID Message Pack				
CRC	3		CRC Error Correction Data as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.1.1				
Term 2	3bits		FEC Block 2 Termination as defined in Bluetooth Core Specification 5.0, Volume 6, Part B, Section 3.3.1				

[A] See https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile/.
[B] See https://www.bluetooth.com/specifications/assigned-numbers/16-bit-uuids-for-sdos/.



802.11 (Type 0, Subtype 13) Vendor Specific Public Action (NAN Service Discovery Frame)				
Category (1 Byte)	Action (1 Byte)	OUI (3 Bytes) (Wi-Fi Alliance)	OUI Type (1 Byte)	NAN Attribute(s)
0x04-Public	0x09-Vend Specific	50-6F-9Ah	0x13	<15+N*25 Bytes>

NAN Service Descriptor Attribute								
							Service Info	
Attribute ID (1 Byte)	Length (2 Bytes)	Service ID (6 Bytes)	Instance ID (1 Byte)	Requestor Instance ID (1 Byte)	Service Control (1 Byte)	Service Info Length (1 Byte)	Message Counter (1 Byte)	Open Drone ID Message Pack
0x3	<14+N*25>	88-69-19-9D-92-09h	0x01	0x00	0x10	<4+N*25>	0x00	<3+N*25 Bytes>

*N = Number of Message Types in Message Pack

Open Drone ID Message Pack						
MsgType (4 bits) [MsgPk]	Version (4 bits)	Single msg size (1 byte)	No of Msgs in Pack (N)	Message (Type 0)	Message (Type 1)	....
0xF	0x0-0xF	0x19 (25)	<1 Byte>	<25 Bytes>	<25 Bytes>	....

Open Drone ID Message		
Msg Type (4 bits)	Version (4 bits)	Message (24 Bytes)
0x0-0xE	0x0-0xF	<Open Drone ID message>

FIG. 10 NAN Service Discovery Frame Diagram

NAN Service Discovery Frame Format

5.4.8.6 The NAN Service Discovery Frame (SDF) is a Vendor Specific Public Action frame as defined in [3.8][10,5] with the Wi-Fi Alliance OUI and Wi-Fi Alliance OUI type indicating the NAN protocol. The format and the values for the NAN SDF are defined in Fig. 10 and Table 18. The mandatory NAN Service Descriptor Attribute shall (BWF0060) be included in the NAN SDF frames. Please refer to NAN Specification Part 9 for detailed information.

Wi-Fi Operating Channels

5.4.8.7 In order to allow operation of NAN Discovery, broadcasting shall (BWF0090) operate only in channel 6 (2.437 GHz) in the 2.4 GHz frequency band and may optionally operate in channel 149 (5.745 GHz) in the 5 GHz band.

5.4.8.8 The transmission interval of the consecutive NAN Service Discovery frames shall (BWF0100) meet the requirements of the update rate for dynamic or static messages.

5.5 *Network Remote ID:*

5.5.1 This section presents requirements for the four Network Remote ID interfaces and associated functions identified in Fig. 1. The interfaces include:

5.5.1.1 Networked UAS to Net-RID Service Provider

5.5.1.2 Operator of Non-Equipped Network Participant to Net-RID Service Provider

5.5.1.3 Net-RID Service Provider to Net-RID Display Provider

5.5.1.4 Net-RID Display Provider to Display Client.

5.5.2 *UAS to Network Remote ID Service Provider Requirements:*

5.5.2.1 *Context:*

5.5.2.2 This subsection provides requirements for a Net-RID Service Provider with respect to data received by means of its interface with Networked UAS, as illustrated in Fig. 11.

5.5.2.3 This specification does not specify an API for the communications interface between Networked UAS and Net-RID Service Providers. However, in order to participate in Network Remote ID, the interface must enable the Net-RID Service Provider to construct messages that comply with the required Remote ID data fields and to be able to provide updated location data at the required periodicity.

5.5.2.4 *Requirements:*

(1) UAS shall (NET0010) authenticate with Net-RID Service Providers using an industry-standard authentication mechanism.

NOTE 1—Authentication of the UAS to the Net-RID Service Provider provides confidence in the Remote ID identity provided by the UAS. OAuth 2.0 or later is an example of an industry-standard authentication mechanism by attestation, for those OAuth2 identity providers that require authentication to obtain credentials.

(2) Communication between UAS and Net-RID Service Providers shall (NET0020) be encrypted using an industry-standard encryption mechanism with a minimum encryption strength of 128 bits.

TABLE 18 NAN Service Discovery Frame Details

Field	Size	Value	Description
Category ID	1	0x4	IEEE 802.11 Public Action Frame
Action Field	1	0x9	IEEE 802.11 Public Action Frame Vendor Specific
OUI	3	50-6F-9A	Wi-Fi Alliance Specific OUI
OUI Type	1	0x13	Identifying the type and version of the NAN
NAN Attributes			
Attributes ID	1	0x3	Identifies the type of NAN attribute (Service Descriptor attribute)
Length	2	Variable	Length
Service ID	6	88-69-19 9D-92-09	This is a mandatory field that contains the first 48 bits of the SHA-256 hash of the Service Name. A lower-case representation of the Service Name shall (BWF0070) be used to calculate the Service ID. The format of the Service ID field shall (BWF0080) be as defined as the Service Hash in "Wi-Fi Peer-to-Peer Services Technical Specification." The service name is "org.opendroneid.remoteid." Hash: 88-69-19-9D-92-09h
Instance ID	1	1	Publish_ID or Subscribe_ID, Value of zero is reserved
Requestor ID	1	0x00	Instance ID from the frame that triggered the transmission if available, otherwise set to 0x00.
Service Control	1	0x10	Mandatory field that defines the Service Control bitmap as defined below. Bit 0-1: Identifies the Service Control Type. The value shall be set to "00:" 00: Publish 01: Subscribe 10: Follow up 11: Reserved Bit 2-3: "00" Bit 4: The value shall be set to "1" and present Service Info field is found in the Service Descriptor attribute
Service Info Len	1	4+(N*25)	Mandatory field set to the length of Service Info
Service Info	4+(N*25)	Variable	Mandatory field that carries the Open Drone ID message pack and with a capacity of up to 255 bytes. Message Counter: 0 – 0xFF: Increment with each message pack sent, reset back to 0 after FF is reached. Open Drone ID MessagePack: Open Drone ID Message Pack

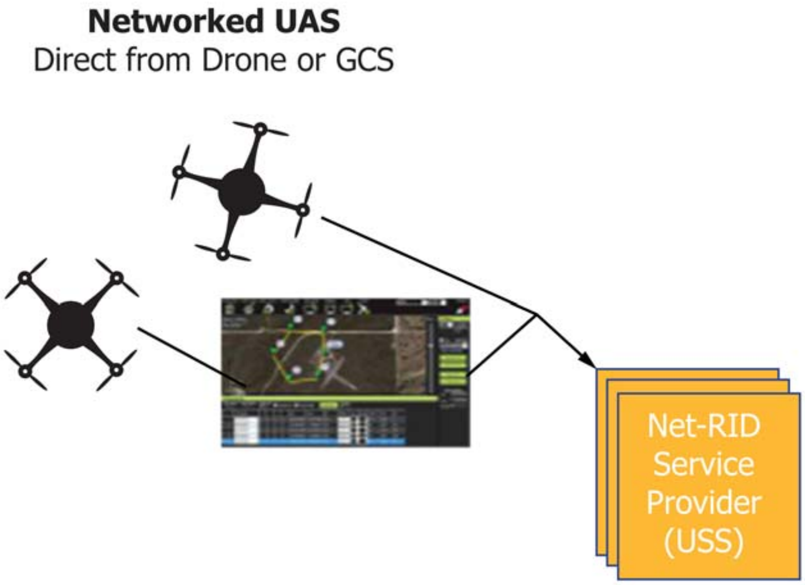


FIG. 11 UAS to Net-RID Service Provider Context

NOTE 2—This requirement is intended to address both integrity and confidentiality of Remote ID data in transit. TLS is an example of an industry-standard authenticated encryption mechanism.

(3) The Net-RID Service Provider shall (NET0030) notify the operator of a Networked UAS if the UAS is not providing necessary data to participate in Network Remote ID.

NOTE 3—A timely notification is intended, but the manner in which the notification is delivered is at the implementer's discretion, and could use electronic or manual methods; therefore, a specific timing requirement for the notification is not specified. An implementer must describe the method(s) used in the product test report.

(4) If dynamic data (for example, position updates) are not being received from a UAS at a frequency of [NetMinUas LocRefreshFrequency] Hz at least [NetMinUasLocRefreshPercentage] of the time, the Net-RID Service Provider shall (NET0040) notify the operator.

NOTE 4—The intention of this requirement is not to preclude intermittent loss of network connectivity, but to detect situations where a UAS appears never to provide position updates at the minimum required rate. When the Net-RID Service Provider detects that a UAS may not be able to comply with the reporting frequency requirement, it should notify the operator so that steps can be taken to resolve the situation. A timely notification is intended, but the manner in which the notification is delivered is at the implementer's discretion, and could use electronic or manual methods; therefore, a specific timing requirement for the notification is not specified. An implementer must describe the method(s) used in the product test report.

5.5.3 *Operator of Non-Equipped Network Participant to Net-RID Service Provider Requirements:*

Context

5.5.3.1 This section provides requirements for Net-RID Service Providers with respect to support for Non-Equipped Network Participants, as illustrated in Fig. 12.

5.5.3.2 A Non-Equipped Network Participant is a UAS that does not participate in network Remote ID in real-time, meaning neither the UA nor the GCS is network connected. However, the operator can participate in network Remote ID by submitting an operation plan which identifies the location and schedule for the operation, and the ID of the aircraft. The operation plan is used as a volume-based position report (that is, a position report whose uncertainty is defined by the volume) during the planned time of operation.

5.5.3.3 This specification does not require that all Net-RID Service Providers must support Non-Equipped Network Participants; however, those that do must adhere to the requirements in this section.

5.5.3.4 In order to submit operation plans for Non-Equipped Network Participants, it is expected that the operator will use an app or a direct interface with the Net-RID Service Provider. This specification does not specify the details of an API or user interface for this purpose, but does specify security requirements and the minimum data that must be collected.

5.5.3.5 *Requirements:*

(1) Net-RID Service Providers that support Non-Equipped Network Participants shall (NET0110) provide the ability for the operator of a Non-Equipped Network Participant to submit, modify, or delete an operation plan.

NOTE 5—An operating plan for a non-equipped network participant consists of the "Operating Area" fields in Table 1. This includes Operating Area Polygon, Type, Count, Floor, Ceiling, Start, and End. Because many of the Common Data Dictionary fields cannot be known in the case of a non-equipped network participant (for example, Track Direction, Altitude, Speed), appropriate null values should be provided.

(2) Net-RID Service Providers shall (NET0120) require authentication of operators using an industry-standard authentication mechanism when operation plans are submitted for Non-Equipped Network Participants.

NOTE 6—Authentication of the operator of a Non-Equipped Network Participant provides confidence in the validity of an operation plan. OAuth 2.0 or later is an example of an industry-standard authentication mechanism, for those OAuth2 identity providers that require authentication to obtain credentials.

(3) Communications between the Non-Equipped Network Participants operator interface and a Net-RID Services Provider shall (NET0130) be encrypted using an industry-standard encryption mechanism with a minimum encryption strength of 128 bits.

NOTE 7—This requirement is intended to address both integrity and confidentiality of Remote ID data in transit. TLS is an example of an industry-standard authenticated encryption mechanism.

5.5.4 *Net-RID Service Provider to Net-RID Display Provider Requirements:*

Context

5.5.4.1 This section provides requirements governing the interactions between Net-RID Service Providers and Net-RID Display Providers, as illustrated in Fig. 13.

5.5.4.2 The role of Net-RID Display Providers is to aggregate information for all UAS operating in an area and provide it to client Remote ID display applications, such as a website or app. There are four general phases relative to a request:

(1) Discover applicable Net-RID Service Providers: a Net-RID Display Provider must determine the Net-RID Service Providers that are operating in an area requested by a Remote

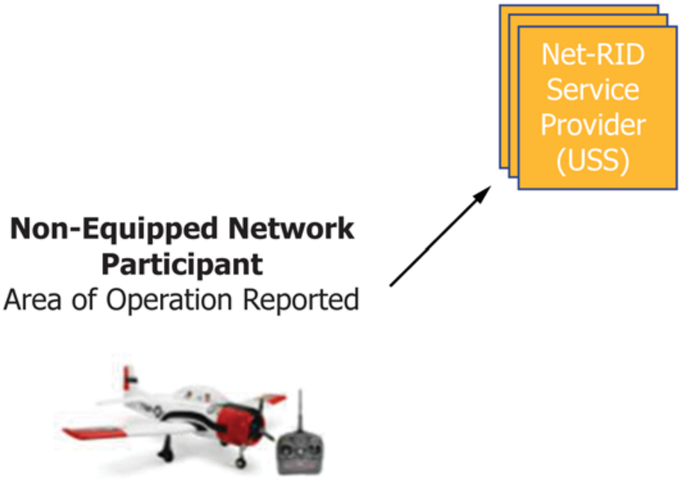

Net-RID Service Provider (USS)
Non-Equipped Network Participant
Area of Operation Reported

FIG. 12 Non-Equipped Network Participant to Net-RID Service Provider

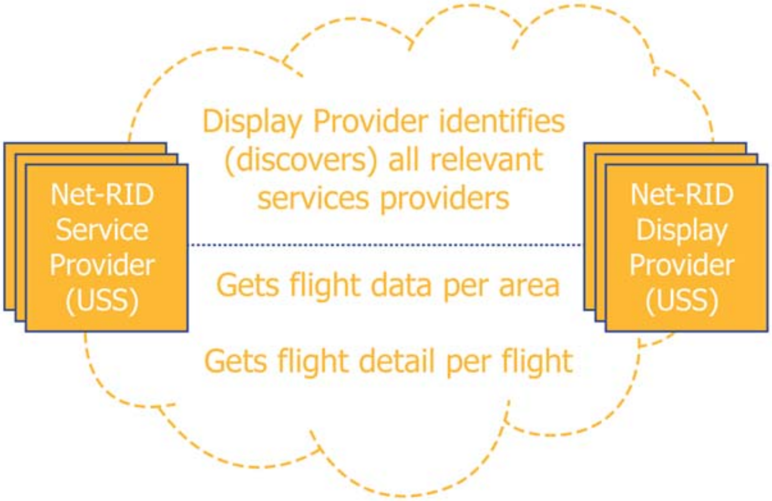

Display Provider identifies (discovers) all relevant services providers
Net-RID Service Provider (USS)
Net-RID Display Provider (USS)
Gets flight data per area
Gets flight detail per flight

FIG. 13 Net-RID Service Provider to Net-RID Display Provider

ID display application end user. Because additional Net-RID Service Providers can come online or introduce operations into an area where none previously existed, discovery must continue while an end-user is viewing data for an area.

(2) Data requests: once applicable Net-RID Service Providers are identified, the Net-RID Display Provider makes periodic requests to the applicable Net-RID Service Providers for Remote ID data for UAS in the area. Two types of requests can be made:

(a) Position information for all UAS in the area (for example, lat, long, altitude, speed, direction).

(b) Detailed, non-position information for a particular UAS in the area (for example, UAS ID, operator location, operation description).

(3) Discard data: all UAS Remote ID data acquired by a Net-RID Display Provider from other Net-RID Service Providers must be discarded within a specified time period after the data is received. (A Net-RID Display Provider may itself be a Net-RID Service Provider and is not required to delete its own data.) Disposing of data when no longer needed (in combination with limiting data access to only what is required for display areas being viewed by a Net-RID Display Provider's associated Display Applications) helps protect privacy and sensitive data of operators and, if applicable, their customers.

5.5.4.3 The data exchange requirements between Net-RID Service Providers and Net-RID Display Providers are defined in Annex A2.

5.5.4.4 *Requirements:*

(1) Net-RID Display Providers shall (NET0210) authenticate with Net-RID Service Providers using an industry-standard authentication mechanism.

NOTE 8—Authentication of the Net-RID Display Providers provides confidence that Remote ID data requests originate from the purported source. OAuth 2.0 or later is an example of an industry-standard authentication mechanism, for those OAuth2 identity providers that require authentication to obtain credentials.

(2) Communications between Net-RID Display Providers and Net-RID Services providers shall (NET0220) be encrypted using an industry-standard encryption mechanism with a minimum encryption strength of 128 bits.

NOTE 9—This requirement is intended to address both integrity and confidentiality of Remote ID data in transit. TLS is an example of an industry-standard authenticated encryption mechanism.

(3) A Net-RID Display Provider shall (NET0230) request Remote ID data from a Net-RID Service Provider only for areas in which end users are currently requesting information by means of the Remote ID Display Applications services.

(4) A Net-RID Display Provider shall (NET0240) request relevant UAS Remote ID information from a Net-RID Service Provider by specifying a rectangular area with a diagonal no greater than [NetMaxDisplayAreaDiagonal] km.

(5) A Net-RID Service Provider shall (NET0250) provide an error code or message and no Remote ID data in response to a request when the diagonal specifying the rectangular area is greater than [NetMaxDisplayAreaDiagonal] km.

(6) When the diagonal for the requested area is within the limit, the Net-RID Service Provider(s) shall (NET0260) provide relevant Remote ID data for the requested rectangular area in no more than [NetSpDataResponseTime95thPercentile] seconds 95 % of the time and in [NetSpDataResponseTime99thPercentile] 99 % of the time. Relevant data includes information consistent with the common data dictionary described in 5.3 and, if requested by the Net-RID Display Provider, near-real-time position information for UAs that are currently in the requested area or that were in the area up to [NetMaxNearRealTimeDataPeriod] seconds prior, including Non-Equipped Network Participants.

NOTE 10—

(a) The response time requirements apply both to requests from a Net-RID Display Provider for position information for all relevant UAS in an area, as well as for the detailed information for an individual UAS.

(b) Providing near-real-time position information if requested is intended to enable an implementation not to be forced to resend the same data a Net-RID Display Provider already has. For example, the Net-RID Display Provider could request the near-real-time information the first time it requests data for an area, and thereafter it would not need to be requested and resent.

(c) For simplicity, this requirement is written assuming a pull approach, meaning the Net-RID Display Provider requests data from the Net-RID Service Provider on a periodic basis, such as once per second. A subscription-based implementation can also meet the intent of this requirement as long as the push interval aligns with the

NetSpDataResponse95thPercentile and 99thPercentile response times. For example, if the 95th Percentile response time is 1 s, the subscription push rate must be once per second 95 % of the time.

(7) For each applicable UA, near-real-time position information shall (NET0270) include:

(a) All position reports in the requested area up to [NetMaxNearRealTimeDataPeriod] in the past.

(b) For each time a UA entered the requested area during [NetMaxNearRealTimeDataPeriod], the last position report received outside of the request area.

(c) For each time a UA exits the requested area during [NetMaxNearRealTimeDataPeriod], the first position report received outside of the requested area.

NOTE 11—Items b and c are intended to allow a display application to show approximately where a UA entered or exited the requested area.

(8) If a networked UAS temporarily loses network connectivity, a Net-RID Service Provider may (NET0280) derive and supply location information from UAS operation plan extrapolation to Net-RID Display Providers until network connectivity is reestablished and updated location information is received from the UAS.

(9) A Net-RID Service provider shall (NET0290) not provide extrapolated location information to Net-RID Display Providers for a UAS if network connectivity with the UAS exists and location information is being received.

NOTE 12—The intention is that extrapolation only be used when necessary.

(10) The Net-RID Service Provider shall (NET0300) inform Net-RID Display Providers when flight plan extrapolation is being used to supply position information for a UAS.

(11) When flight plan extrapolation is used to supply position information for a UAS, the Net-RID Service Provider shall (NET0310) characterize the accuracy of the extrapolated location data using the Vertical Accuracy and Horizontal Accuracy of Position data fields.

(12) If a networked UAS loses connectivity and the associated Net-RID Service Provider is unable to provide extrapolated position data, the Net-RID Service Provider shall (NET0320) provide to a requesting Net-RID Display Provider the most recent position report and an indication that current data is not being received.

(13) Net-RID Display Providers shall (NET0330) retain data obtained from Net-RID Service Providers for no longer than [NetDpMaxDataRetentionPeriod] seconds. This requirement does not apply in the case where a Net-RID Display Provider is also a Net-RID Service provider; that is, a USS performing both roles is not required to delete its own data.

5.5.4.5 In addition to the performance requirements provided above, Net-RID Service Providers and Net-RID Display Providers must (NET0340) support the minimum APIs and associated requirements defined in Annex A2.

5.5.5 *Net-RID Display Provider to Display Application Requirements:*

Context

5.5.5.1 This section provides requirements governing the interactions between Remote ID Display Applications and Net-RID Display Providers, as illustrated in Fig. 14.

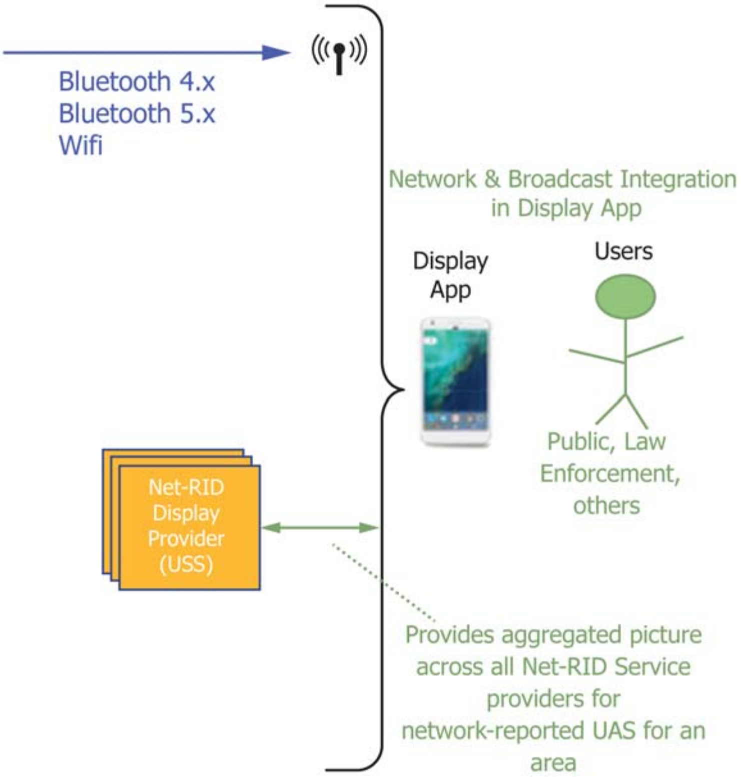

Bluetooth 4.x
Bluetooth 5.x
Wifi
Network & Broadcast Integration in Display App
Display App
Users
Public, Law Enforcement, others
Net-RID Display Provider (USS)
Provides aggregated picture across all Net-RID Service providers for network-reported UAS for an area

FIG. 14 Remote ID Display Applications to Net-RID Display Providers

5.5.5.2 This specification does not impose authentication requirements on the communications interface between Remote ID Display Applications and Net-RID Display Providers.

5.5.5.3 This specification neither specifies the API between display applications and Net-RID Display Providers, nor includes requirements pertaining to the user interface provided by a Remote Identification Display Application. (As noted in 4.3, Remote Identification Display Applications are outside the scope of this specification.) However, this specification does levy performance-based requirements on Net-RID Display Providers and their interface with Remote ID Display Applications.

5.5.5.4 This specification is intended to provide the logical equivalent of obtaining the license plate for a car observed by an interested party. To enable that objective but enforce the privacy principle of only sharing data that needs to be shared and to discourage the use of Remote Identification as a means for ongoing surveillance over a wide area or to mine patterns of life for users of drone services, distance-based limitations are imposed on Remote Identification information that can be obtained as illustrated in Fig. 15.

5.5.5.5 As previously discussed, no data is provided from a Net-RID Service Provider to a Net-RID Display Provider for an area with a diagonal larger than NetMaxDisplayAreaDiagonal. For areas less than or equal to that size, Net-RID Display Providers enforce two levels of information provided to Remote ID Display Applications, based on NetDetailsMaxDisplayAreaDiagonal.

5.5.5.6 For display areas larger than NetDetailsMaxDisplayAreaDiagonal, position information for UAs is either clustered in the case of multiple UAs in close proximity, or obfuscated for individual UAs, using a circular position report with a count of UAs.

5.5.5.7 For display areas equal to or smaller than NetDetailsMaxDisplayAreaDiagonal, position information for individual UAs is provided.

5.5.5.8 Two types of requests from a Display Application to a Net-RID Display Provider are addressed in the requirements below, and they correspond to the two types of requests between Net-RID Display Providers and Net-RID Service Providers described in 5.5.4. Position information requests provide the location details for all UAS in the requested area, and includes items such as lat, long, altitude, speed, track direction. Flight detail request provides additional data about a specific UAS, such as the UAS ID, operator location, and operation description. (The precise data for each interface is fully enumerated in the API provided in Annex A4.)

5.5.5.9 *Requirements:*

(1) Communications between Net-RID Display Providers and Remote ID Display Applications shall (NET0410) be encrypted using an industry-standard encryption mechanism with a minimum encryption strength of 128 bits.

NOTE 13—This requirement is intended to address both integrity and confidentiality of Remote ID data in transit. TLS is an example of an industry-standard authenticated encryption mechanism.

(2) A Net-RID Display Provider shall (NET0420) respond to the initial request from a Remote ID Display Application for position data for all UAS in an area with a diagonal no greater than [NetMaxDisplayAreaDiagonal] km in [NetDpInitResponse95thPercentile] seconds 95 % of the time and in [NetDpInitResponse99thPercentile] seconds 99 % of the time.

NOTE 14—The response time requirement for initial requests for an area is longer than subsequent requests to allow for the discovery process used by the Net-RID Display Provider to find and establish communications with the applicable Net-ID Service Providers.

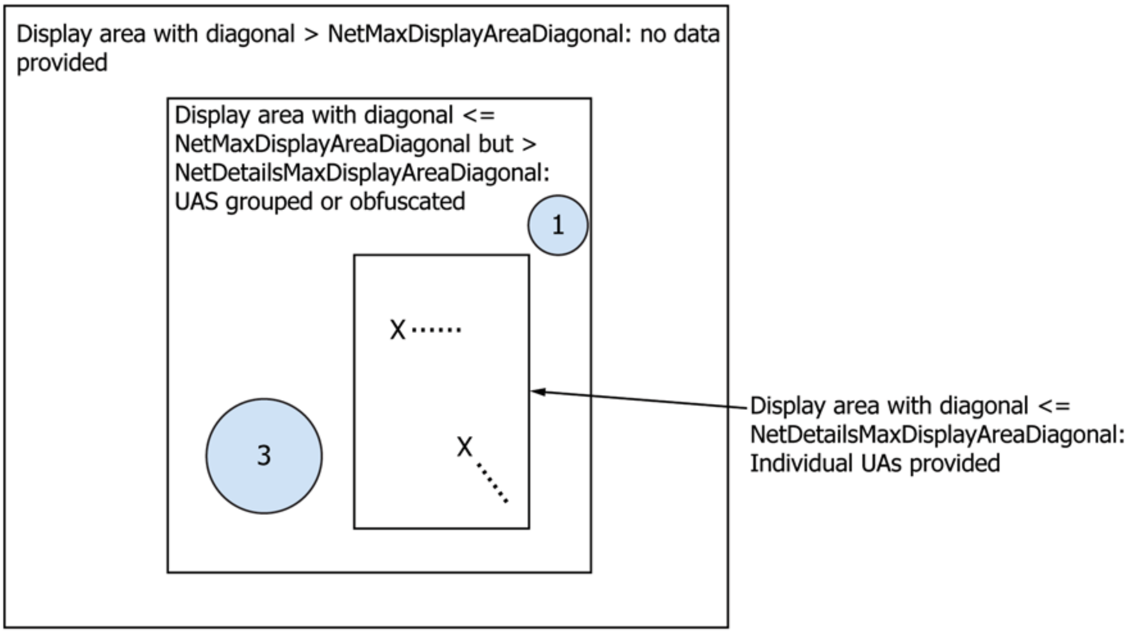

Display area with diagonal > NetMaxDisplayAreaDiagonal: no data provided
Display area with diagonal <= NetMaxDisplayAreaDiagonal but > NetDetailsMaxDisplayAreaDiagonal: UAS grouped or obfuscated
1
X
3
X
Display area with diagonal <= NetDetailsMaxDisplayAreaDiagonal: Individual UAs provided

FIG. 15 Data Provision to Display Application Based on Display Area Size

(3) A Net-RID Display Provider shall (NET0430) provide an error code or message and no Remote ID data in response to a request from a Remote ID Display Application for an area with a diagonal greater than [NetMaxDisplayAreaDiagonal].

(4) A Net-RID Display Provider shall (NET0440) respond to subsequent requests (that is, requests after the initial request) from a Remote ID Display Application for UAS Remote ID for an area previously requested within the past [NetMinSessionLength] with a diagonal no greater than [NetMaxDisplayAreaDiagonal] km in [NetDpDataResponse95thPercentile] seconds 95 % of the time and in [NetDpDataResponse99thPercentile] seconds 99 % of the time.

NOTE 15—Conceptually, when a Display Application makes a request to a Net-RID Display Provider for an area, this begins a session. The session remains in effect as long as the user is viewing the area, during which the Display Application will continue to make data requests. NetMinSessionLength represents a Time To Live (TTL) mechanism for the session and allows the Net-RID Display provider to distinguish between initial requests (NET0420) and subsequent requests (NET0440, this requirement). When a new request is made (the initial request), a NetMinSessionLength timer is started. If a subsequent data request for the same area is received and the timer has not expired (the norm as long as the user continues to view the area), the Net-RID Display Provider is able to rely on the discovery process performed on the initial request and reset the timer on each request. If the timer has expired (implying the user has closed the display or moved the location), a request for data in the same area results in the Net-RID Display Provider treating it as a new initial request for which discovery must be repeated. This mechanism is needed because the standard does not impose a requirement on Display Applications to provide notification when a user session ends.

(5) When responding to valid requests for Remote ID data from a display application, a Net-RID Display Provider shall (NET0450) provide the most recent data available that is relevant, aggregated from all applicable Net-RID Service Providers. Relevant data includes information consistent with the common data dictionary described in 5.3 and, if requested by the Net-RID Display Application, near-real-time position information for UAs that are currently in the requested area or that were in the area up to [NetMaxNearRealTimeDataPeriod] seconds prior, including Non-Equipped Network Participants.

NOTE 16—Near-real-time data is described in requirement NET0280. Providing near-real-time position information, if requested, is intended to enable a Net-RID Display Provider not to be forced to resend the same data to a Display Application. For example, the Display Application could request the near-real-time information the first time it requests the data for an area, and thereafter it would not need to be requested and resent.

(6) A Net-RID Display Provider shall (NET0460) respond to requests from a Display Application for flight details for a specific UAS within an area with a diagonal equal to or less than [NetDetailsMaxDisplayAreaDiagonal] in [NetDpDetailsResponse95thPercentile] seconds 95 % of the time and in [NetDpDetailsResponse99thPercentile] seconds 99 % of the time.

(7) A Net-RID Display Provider is not required to provide all data fields identified in Table 1 to Display Applications; however, any data provided from Table 1 shall (NET0470) conform to the description and associated rules provided in the table or in 5.3.

NOTE 17—This requirement clarifies that Remote ID Display Applications are not required to display every field from Table 1, and a Net-RID Display Provider may also be implemented that does not provide all fields to their associated Display Applications.

(8) For a display area with a diagonal greater than [Net DetailsMaxDisplayAreaDiagonal] and less than [NetMaxDisplayAreaDiagonal], a Net-RID Display Provider shall (NET0480) cluster UAs in close proximity to each other using a circular or polygonal area covering no less than [NetMinClusterSize] percent of the display area size and associating a count of the UAs in the cluster.

NOTE 18—This specification avoids specifying a particular clustering algorithm to provide implementation flexibility. However, the NetMinClusterSize constant is used to ensure that clusters are large enough to effectively obscure UA locations.

(9) For a display area with a diagonal greater than [NetDetailsMaxDisplayAreaDiagonal] and less than [NetMaxDisplayAreaDiagonal], a Net-RID Display Provider shall (NET0490) reduce the precision of location information for individual UAs that are not included in a cluster. This is to be accomplished using a circular or polygonal area with a radius or distance to the polygon edge of no less than [NetMinObfuscationDistance], randomly offset from the actual UA location (but always encompassing the UA location), and associating a UA count of 1 with the area.

TEST METHODS

6. Scope

6.1 This section outlines the test methods used to test compliance with the Remote ID standard. Broadcast and network Remote ID tests differ.

6.2 The test for broadcast Remote ID shall determine the power level, latency, periodicity, and protocol compliance.

6.3 The test for network Remote ID shall determine the latency, periodicity, reliability, protocol compliance, and interoperability with other network Remote ID implementations.

7. Significance and Use

7.1 The specification is intended to be used by UAS manufacturers, UAS operators, Remote ID USSs, and CAAs to assess UAS or USS compliance, or both, with the Remote ID standard.

8. Hazards

8.1 Ensure that UAS are configured as to not cause harm to individual(s) conducting the test or third parties.

8.2 UAS that are powered or operational can present hazards. Ensure that propellers are removed or caged during laboratory testing.

8.3 Field testing of UAS can present hazards. Take appropriate safety precautions when field testing UAS.

8.4 When testing UAS with power plants or lithium batteries, or both, an appropriate fire extinguisher for each application should be within reach. Participants should be made aware of the hazards of lithium batteries or flammable fuels, or both, and which fire extinguishers are appropriate for lithium or flammable fuel-based fires, or both.

9. Test Units

9.1 The UAS used in this test shall be mechanically and electrically equivalent to the actual flying configuration. The UAS must be operational and powered during testing.

10. Procedure

10.1 Compliance matrices to support a conformance determination process are provided below. As part of determining compliance, implementers must validate that all functional, performance, and interoperability requirements are met. Test results and a description of how requirements were validated must be documented in a product test report using the notes columns of the applicable compliance matrices and supplemental documentation as needed.

10.2 *Broadcast:*

10.2.1 To determine compliance to any broadcast method, compliance must be verified for all items in the "All Methods" section in addition to the applicable section for the broadcast method (that is, Bluetooth 4.x, Bluetooth 5.x, or Wi-Fi).

10.2.2 To create a Broadcast Authentication Verifier, see Annex A1 for the applicable compliance matrix.

10.2.3 All Methods Compliance Matrix (see Table 19).

10.2.4 Bluetooth 4.x Compliance Matrix (see Table 20).

10.2.5 Bluetooth 5.x Compliance Matrix (see Table 21).

10.2.6 Wi-Fi Compliance Matrix (see Table 22).

10.3 *Network:*

10.3.1 Net-RID Service Providers shall (NET0500) provide a persistently supported test instance of their implementation for use by Net-RID Display Providers in order to support interoperability testing. This test instance must use the current deployed version of the implementer's Net-RID software and provide a means for injection or generation of test data in a geographic test location.

10.3.2 Network Compliance Matrix (see Table 23).

10.3.3 *Notes:*

10.3.3.1 Requirements that are not applicable to a particular product (for example, UAS to Net-RID Service Provider requirements for an implementation that is exclusively a Net-RID Display Provider) should be marked as "NA" in the test report.

10.3.3.2 If a USS implementation includes a new or modified DSS, an additional compliance matrix and test guidance are provided in A2.5.

11. Precision and Bias

11.1 All requirements that have necessary precision attributes have the required precision stated within the requirement itself. Otherwise, the stated requirements must be met 95 % of the time unless other local regulations apply. For values that declare their own precision (such as horizontal location,

TABLE 19 All Methods Compliance Matrix

Requirement ID	Section Reference	Compliant (Y/N)	Notes
BPW0010	5.4.3		
BUR0010	5.4.4		
BUR0020	5.4.4		
BUR0030	5.4.4		
BUR0040	5.4.4		
BMG0010	5.4.5.2		
BMG0020	5.4.5.3		
BMG0030	5.4.5.3		
BMG0040	5.4.5.3		
BMG0050	5.4.5.3		
BMG0060	5.4.5.3		
BMG0065	5.4.5.3		
BMG0070	5.4.5.4		
BMG0080	5.4.5.6		
BMG0090	5.4.5.6		
BMG0100	5.4.5.8		
BMG0110	5.4.5.8		
BMG0120	5.4.5.14		
BMG0130	5.4.5.14		
BMG0140	5.4.5.14		
BMG0150	5.4.5.14		
BMG0160	5.4.5.15		
BMG0170	5.4.5.15		
BMG0180	5.4.5.15		
BMG0190	5.4.5.19		
BMG0200	5.4.5.19		
BMG0210	5.4.5.19		
BMG0220	5.4.5.19		
BMG0230	5.4.5.19		
BMG0240	5.4.5.19		

TABLE 20 Bluetooth 4.x Compliance Matrix

Requirement ID	Section Reference	Compliant (Y/N)	Notes
BB40010	5.4.6.3		
BB40020	5.4.6.3		
BB40030	5.4.6.4		

TABLE 21 Bluetooth 5.x Compliance Matrix

Requirement ID	Section Reference	Compliant (Y/N)	Notes
BB50010	5.4.7.1		
BB50020	5.4.7.1		
BB50030	5.4.7.1		
BB50040	5.4.7.1		
BB50050	5.4.7.3		
BB50060	5.4.7.4		
BB50070	5.4.7.5		
BB50080	5.4.7.5		
BB50090	5.4.7.5		
BB50100	5.4.7.6		
BB50110	5.4.7.7		
BB50120	5.4.7.7		

TABLE 22 Wi-Fi Compliance Matrix

Requirement ID	Section Reference	Compliant (Y/N)	Notes
BWF0010	5.4.8.1		
BWF0020	5.4.8.2		
BWF0030	5.4.8.2		
BWF0040	5.4.8.3		
BWF0050	5.4.8.3		
BWF0060	5.4.8.6		
BWF0070	5.4.8.6		
BWF0080	5.4.8.6		
BWF0090	5.4.8.7		
BWF0100	5.4.8.8		

vertical location, speed, and timestamp), the precision required shall be as required by local regulations. No information can be presented on the bias of the test methods in this specification because no such requirements have an accepted reference value available.

12. Product Marking

12.1 UAS that are capable of meeting the threshold for compliance with this specification should be labeled Remote ID and Tracking capable.

12.2 *UAS that Are Marked:*

12.2.1 "ASTM F3411-19-RID-N" compliant must comply with the Network Remote ID standard.

12.2.2 "ASTM F3411-19-RID-B" compliant must comply with the Broadcast Remote ID standard.

12.2.3 "ASTM F3411-19-RID-NB" must comply with both the Broadcast and Network Remote ID standards.

13. Packaging and Package Marking

13.1 The packaging of UAS that are capable of meeting the threshold for compliance with this specification should be labeled ID and tracking capable.

13.2 *Packaging of UAS that Are Marked:*

13.2.1 "ASTM F3411-19-RID-N" compliant must comply with the Network Remote ID standard.

13.2.2 "ASTM F3411-19-RID-B" compliant must comply with the Broadcast Remote ID standard.

13.2.3 "ASTM F3411-19-RID-NB" must comply with both the Broadcast and Network Remote ID standards

14. Keywords

14.1 broadcast; network; remote ID; UAS traffic management; unmanned aircraft systems

TABLE 23 Network Compliance Matrix

Requirement ID	Section Reference	Compliant (Y/N)	Notes
NET0010	5.5.2.4		
NET0020	5.5.2.4		
NET0030	5.5.2.4		
NET0040	5.5.2.4		
NET0110	5.5.3.5		
NET0120	5.5.3.5		
NET0130	5.5.3.5		
NET0210	5.5.4.4		
NET0220	5.5.4.4		
NET0230	5.5.4.4		
NET0240	5.5.4.4		
NET0250	5.5.4.4		
NET0260	5.5.4.4		
NET0270	5.5.4.4		
NET0280	5.5.4.4		
NET0290	5.5.4.4		
NET0300	5.5.4.4		
NET0310	5.5.4.4		
NET0320	5.5.4.4		
NET0330	5.5.4.4		
NET0340	5.5.4.4		
NET0410	5.5.5.9		
NET0420	5.5.5.9		
NET0430	5.5.5.9		
NET0440	5.5.5.9		
NET0450	5.5.5.9		
NET0460	5.5.5.9		
NET0470	5.5.5.9		
NET0480	5.5.5.9		
NET0490	5.5.5.9		
NET0500	10.3.1		
NET0610	A2.3.2		
NET0620	A2.3.2		
NET0630	A2.3.2		
NET0710	A2.4.1		
NET0720	A2.4.1		
NET0730	A2.4.1		

ANNEXES

(Mandatory Information)

A1. BROADCAST AUTHENTICATION VERIFIER SERVICE

A1.1 Overview

A1.1.1 The basic design for broadcast authentication is for the signature specification to be defined by the verifier and the protocol to be defined by this ASTM specification. As such, the broadcast packet format is described in the broadcast section of the specification, and the compatible verifier service requirements are described in this annex. (See Fig. A1.1.)

A1.2 Verifier Service Requirements

A1.2.1 The verifier service shall (VF0010) implement the signature verification algorithm that matches the agreed signature format that will be sent by the broadcaster.

A1.2.2 The verifier service shall (VF0020) set up a web service as described in this annex.

A1.2.3 The response time of the verifier shall (VF0030) be less than [VfResponseTime95] seconds 95 % of the time from receipt of verify request message to the transmission of the result. (See Table A1.1.)

A1.2.4 The Verifier API shall (VF0040) implement a RESTful/JSON protocol on a web server with a TLS secured endpoint as described by the Verifier API OpenAPI Description in A1.4.

A1.2.5 The verifier shall (VF0050) maintain a testing endpoint.

A1.2.6 *Result Codes*—The ResultCode and ResultString values shall (VF0060) be set in the verifier response given the conditions in Table A1.2.

A1.3 Verifier API RESTful Interface Sample

A1.3.1 Below is an example of how a verification request may be sent.

A1.3.2 *Sample Request*—See Fig. A1.2.

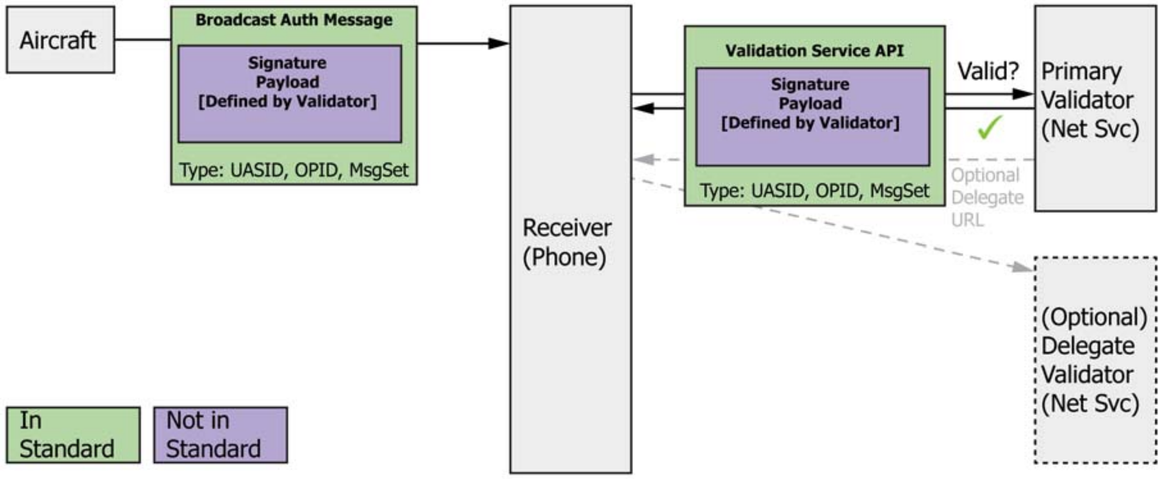


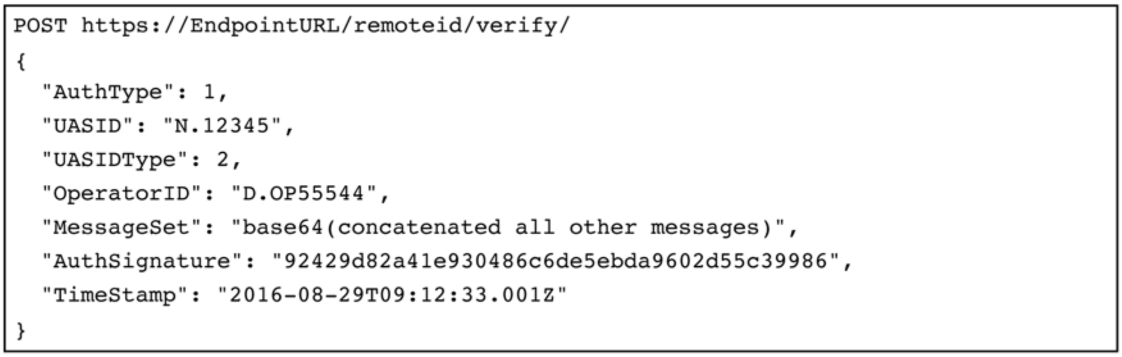
FIG. A1.1 Broadcast Authentication Verifier Service Overview

TABLE A1.1 Authentication Verifier Values

Value Name	Value	UoM	Req Ref	Section Ref
VfResponseTime95	1	Second	VF10030	1.2.3

TABLE A1.2 Authentication Verifier Service ResultCodes and ResultStrings

Code	Result String	Condition
0	VALID	Signature and message is valid
1	INVALID	Message is Invalid or not authentic
2	VALID BUT EXPIRED	Message is Authentic and Key is Expired
3	UNKNOWN KEY	Key cannot be found or validity cannot be determined
4	REVOKED	Key has been revoked
5	UNKNOWN ID	The ID sent cannot be found
6	<Delegate endpoint URL>/remoteid/verify/	Delegate to alternate verifier endpoint URL (including "/remoteid/verify/")



```
POST https://EndpointURL/remoteid/verify/
{
  "AuthType": 1,
  "UASID": "N.12345",
  "UASIDType": 2,
  "OperatorID": "D.OP55544",
  "MessageSet": "base64(concatenated all other messages)",
  "AuthSignature": "92429d82a41e930486c6de5ebda9602d55c39986",
  "TimeStamp": "2016-08-29T09:12:33.001Z"
}
```

FIG. A1.2 Sample Request

A1.3.3 *Request Enumerations:*

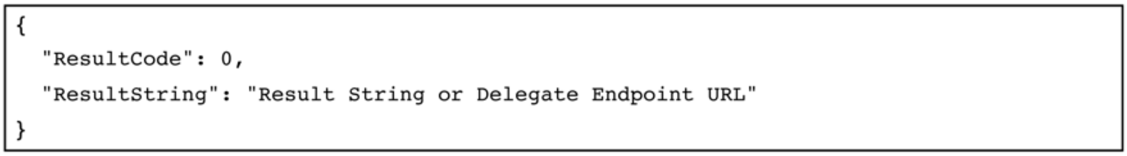
AuthType:

(1) UASID
(2) OperatorID
(3) MessageSet

UASIDType:

(1) ANSI ⁄CTA-2063-A Serial Number
(2) UAS Registration ID
(3) UTM UUID

A1.3.4 *Sample Result*—See Fig. A1.3.



```
{
  "ResultCode": 0,
  "ResultString": "Result String or Delegate Endpoint URL"
}
```

FIG. A1.3 Sample Result

A1.4 Verifier API OpenAPI 3.0 YAML Description

```
openapi: 3.0.2
info:
  description: This API is to allow a Remote ID client to verify the authentication information received by a
Remote ID transmitter
  version: ″1.0.0″
  title: Remote ID Verifier API
  contact:
    email: gabriel.c.cox@intel.com
  license:
    name: Apache 2.0
    url: ’http://www.apache.org/licenses/LICENSE-2.0.html’
paths:
  /remoteid/verify/:
    post:
      summary: Verifies ASTM Remote ID Authentication Package
      operationID: verifyRemoteID
      description: Verifies ASTM Remote ID Authentication
      responses:
        ’200’:
         description: Standard Response
         content:
           application/json:
             schema:
               type: object
               required:
                 - ResultCode
                 - ResultString
               properties:
                 ResultCode:
                   type: integer
                   format: integer
                   description: >
                     Result Code:
                      0: Signature and Message Valid
                      1: Message Invalid
                      2: Key Expired, Message Valid
                      3: Unknown Validity (key not available)
                      4: Key Revoked
                      5: Unknown ID
                      6: Delegated
                 ResultString:
                   type: string
                   format: string
                   description: If Delegated, Delegated Endpoint URL, otherwise Result Description
                   example: Signature and Message Valid
        ’400’:
         description: ’Invalid Request’
      requestBody:
        description: Inventory item to add
        content:
          application/json:
             schema:
               type: object
               required:
                 - AuthType
                 - UASID
                 - UASIDType
                 - OperatorID
                 - MessageSet
                 - AuthSignature
                 - TimeStamp
               properties:
                 AuthType:
                   type: integer
                   minimum: 1
                   minimum: 3
                   description: >
                     1. UASID
                     2. OperatorID
                     3. MessageSet
                   example: 1
                 UASID:
                   type: string
                   example: 'N.12345'
                 UASIDType:
                   type: integer
                   minimum: 1
                   minimum: 3
                   description: >
                     1. ANSI/CTA-2063-A Serial Number
                     2. UAS Registration ID
                     3. UTM UUID
                   example: 2
                 OperatorID:
                   type: string
                   example: 'D.OP55544'
                 MessageSet:
                   type: string
                   format: base64
                   example: adsfasdfasdf
                   description: If using messageSet authtype, base64 encode of message set
                 AuthSignature:
                   type: string
                   description: 'Signature with authType data encoded'
                   example: '92429d82a41e930486c6de5ebda9602d55c39986'
                 TimeStamp:
                   type: string
                   format: date-time
                   example: '2016-08-29T09:12:33.001Z'
```

A1.5 Authentication Verifier Compliance Matrix

A1.5.1 See Table A1.3.

TABLE A1.3 Authentication Verifier Compliance Matrix

Requirement ID	Section Reference	Compliant (Y/N)	Notes
VF0010	A1.2.1		
VF0020	A1.2.2		
VF0030	A1.2.3		
VF0040	A1.2.4		
VF0050	A1.2.5		
VF0060	A1.2.6		

A2. NETWORK REMOTE ID INTEROPERABILITY REQUIREMENTS, APIs, AND TESTING

A2.1 This annex defines requirements, APIs, and testing for network Remote ID interoperability between Net-RID Service Providers and Net-RID Display Providers. A YAML (OpenAPI) description of the APIs is provided in Annex A4.

A2.2 Network Remote ID Interoperability Overview

A2.2.1 Fig. A2.1 summarizes the interfaces and flow of information for network Remote ID interoperability.

A2.2.2 As discussed in the USS Interoperability overview provided in 4.8, an instance of discoverable information is referred to as an entity. The discoverable entity defined for Network Remote ID is an Identification Service Area (ISA). This is a 4-D volume (a volume defined in *x*, *y*, and *z* plus time limits) that corresponds to an area where a Net-RID Service Provider typically has one or more UAS operating, though a Net-RID Service Provider is not required to have an active UAS in a given ISA at all times.

A2.2.3 Net-RID Service Providers have flexibility in defining ISAs. Representative choices include:

A2.2.3.1 An area corresponding to a single operation – this could be used when a USS does not have a high frequency or concentration of flights in a given area.

A2.2.3.2 An area corresponding to a group of operations – this could be used when a USS has frequent flights in a fixed location, as might be the case with delivery services or urban air mobility. In this case, a single ISA reduces the overhead of creating and managing ISAs for each individual flight. (A

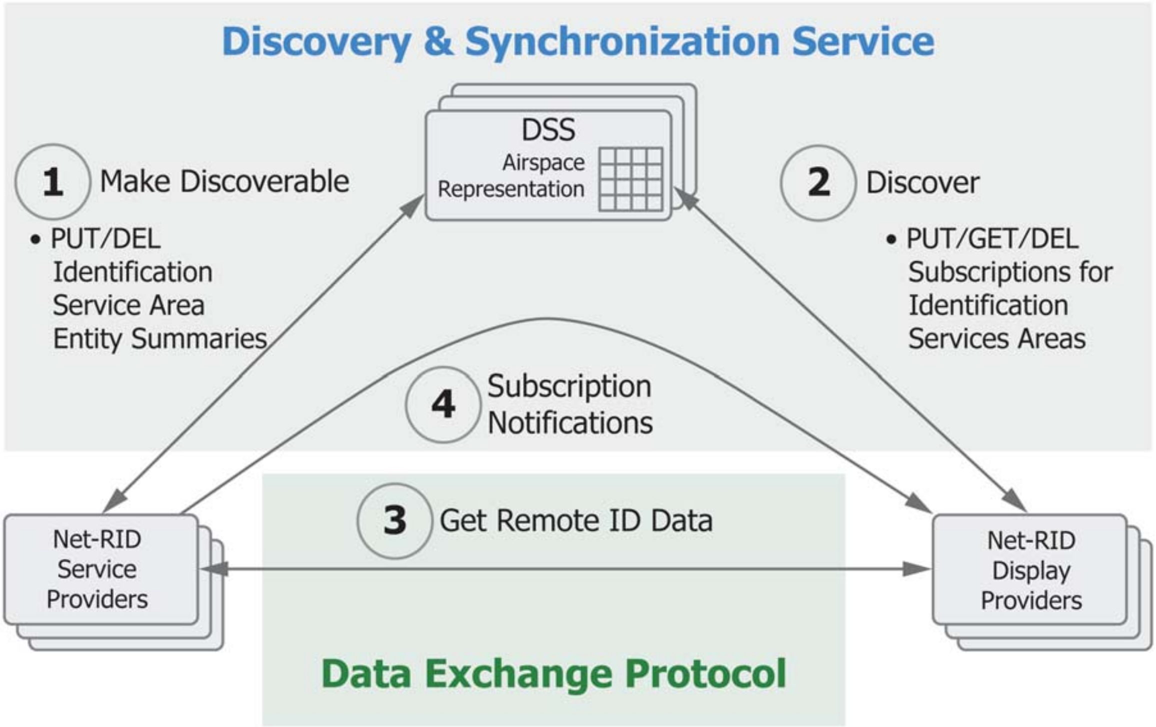


FIG. A2.1 Network Remote ID Interoperability Overview

tradeoff for the USS is that it may receive requests for a portion of the ISA where operations do not exist at a particular point in time.)

A2.2.3.3 An area corresponding to a portion of an operation—this could be used for a flight that travels a long distance where the USS does not want to process data requests for portions of the flight that the UAS has left or has not yet reached.

A2.2.4 This choice of these or other options is left to the Net-RID Service Provider. The standard requires only that all UAS flights operating in conjunction with a Net-RID Service provider are included within at least one ISA.

A2.2.5 The interfaces and flow of information in Fig. A2.1 is as follows (corresponding to the numbered interfaces in the figure):

A2.2.5.1 Net-RID Service Providers make the UAS operations they support discoverable by writing an ISA entity summary to the DSS. The entity summaries are mapped into the airspace representation encapsulated by the DSS. The DSS API also supports modifying, deleting and retrieving the details of ISA.

A2.2.5.2 Net-RID Display Providers discover the Net-RID Service Providers with which communication is required by querying the DSS for an area of interest (that is, an area being viewed in one or more Remote ID Display Applications) and establishing a subscription for notification of any subsequent changes. (A subscription interface rather than a polling interface is used to avoid inefficient usage of the DSS, which is a shared UTM ecosystem resource; that is, it would be inefficient to repeat the complete discovery process once a second for a large number of areas of interest.) The DSS maps the area of interest into the airspace representation to find the intersecting cells with ISAs (if any). For each intersecting ISA, the DSS returns a URL used by the Net-RID Display Provider to obtain Remote ID data from the Net-RID Service Provider that created the ISA (step 3).

A2.2.5.3 Given the URLs for each ISA, the Net-RID Display Provider polls the applicable Net-RID Service Provider to obtain the relevant Remote ID data for UAS in the area. This peer-to-peer interface has two parts, one to obtain position information for all UAS the Net-RID Service provider has in the requested area, and a second to obtain detailed information for a particular flight (for example, the UAS ID, operation description, operator location, etc.).

A2.2.5.4 If a Net-RID Service Provider adds or modifies an ISA in a location that intersects a DSS airspace representation cell with a Net-RID Display Provider subscription, the DSS informs the Net-RID Service Provider and provides the URL necessary to contact the subscribing Net-RID Display Provider. The Net-RID Service Provider calls the Net-RID Display Provider and provides the URL the Display Provider uses to poll the Service Provider.

A2.2.6 Detailed requirements and APIs for these interfaces are provided below.

A2.3 USS–DSS Interfaces

A2.3.1 *DSS Implementation Requirements:*

A2.3.1.1 The requirements in this section apply exclusively to implementers of a DSS. Note that specific response time requirements for DSS functions are omitted because DSSs are not tested in isolation, but rather in conjunction with a USS.

DSS processing time must support a USS meeting applicable response time requirements.

(1) A DSS instance shall (DSS0010) authenticate USSs using an industry-standard authentication mechanism.

NOTE A2.1—OAuth 2.0 or later is an example of an industry-standard authentication mechanism by attestation, for those OAuth2 identity providers that require authentication to obtain credentials.

(2) Communication between a USS and DSS instances shall (DSS0020) be encrypted using an industry-standard encryption mechanism with a minimum encryption strength of 128 bits.

NOTE A2.2—This requirement is intended to address both integrity and confidentiality of Remote ID data in transit. TLS is an example of an industry-standard authenticated encryption mechanism.

(3) A DSS implementation shall (DSS0030) minimally include the following interfaces for use by Net-RID Service Providers and Display Providers, in accordance with the DSS portion of the OpenAPI specification presented in Annex A4:

(a) PUT Identification Service Area—this interface enables a Net-RID Service Provider to create or modify an ISA entity summary in the DSS.

(b) DELETE Identification Service Area—this interface enables a Net-RID Service Provider to delete an existing ISA entity summary from the DSS. (A Net-RID Service Provider can only delete ISAs it created.)

(c) PUT Subscription—this interface creates a subscription for new or modified ISAs within a 4-D volume, and returns the intersecting ISAs resident in the DSS at the time of the call.

(d) DELETE Subscription—this interface enables a Net-RID Display Provider to delete a subscription from the DSS. (A Net-RID Display Provider can only delete subscriptions it created.)

(e) GET Subscription—this interface enables a Net-RID Display Provider to retrieve the details of a specific existing subscription to verify its existence and composition. (A Net-RID Display Provider can only retrieve subscriptions it created.)

(f) GET Subscriptions—this interface enables a Net-RID Display Provider to retrieve the details of all existing subscriptions it created.

NOTE A2.3—As stated, these interfaces are the minimum required. A DSS implementation may include additional interfaces for optimization or functional purposes as long as these minimum interfaces are supported.

(4) After mapping and storing ISA summary information into the DSS Airspace Representation, the DSS shall (DSS0040) not store or otherwise retain the precise geographical extents of the associated 4-D volume.

(5) The DSS shall (DSS0050) not allow more than [NetDSSMaxSubscriptionPerArea] subscriptions per USS in a given area of the DSS Airspace Model.

NOTE A2.4—The intention is that Net-RID Display Providers aggregate subscriptions for overlapping display area requests. It is undesirable from a DSS and USS-USS communications efficiency perspective to have a separate DSS subscription for each of a large number of users viewing overlapping areas.

(6) The DSS shall (DSS0060) limit the duration of subscriptions to no more than [NetDSSMaxSubscriptionDuration].

NOTE A2.5—Subscriptions should only be established and persisted for areas in which display requests exists.

(7) The DSS shall (DSS0070) be implemented in a manner that allows a USS to access any instance of a DSS for a region and obtain the same results. (A region is the geographic scope supported by a set of DSS instances.)

NOTE A2.6—DSS instances do not operate with a single active instance and other instances in backup, but rather all instances are active.

A2.3.2 *USS Requirements Related to the DSS:*

A2.3.2.1 The requirements in this section apply to Net-RID Service Providers or Net-RID Display Providers interacting with a DSS.

(1) A Net-RID Service Provider shall (NET0610) make all UAS operations discoverable for Network Remote ID purposes by means of one or more ISAs in the DSS for the complete duration of each operation plus [NetMaxNearRealTimeDataPeriod].

(2) If a Net-RID Service Provider is unable to make a UAS operation discoverable through the creation of an ISA in the DSS, the Net-RID Service Provider shall (NET0620) notify the operator.

NOTE A2.7—A timely notification is intended, but the manner in which the notification is delivered is at the implementer's discretion and could use electronic or manual methods; therefore, a specific timing requirement for the notification is not specified. An implementer must describe the method(s) used in the product test report.

(3) A Net-RID Display Provider shall (NET0630) obtain ISA information from the DSS (including creating or maintaining an ISA subscription) only for areas in which an end user is currently requesting information by means of the Remote ID Display Applications it services.

A2.4 USS-USS Interfaces

A2.4.1 The requirements in this section address interfaces between Net-RID Service Providers and Display Providers. Note that in accordance with 5.5.4.4, authentication and encryption is required for USS-USS interfaces.

A2.4.1.1 A Net-RID Service Provider shall (NET0710) minimally support the following interfaces for use by Net-RID Display Providers, in accordance with the P2P (peer-to-peer) portion of the OpenAPI specification provided in Annex A4:

(1) GET flights: this interface enables a Net-RID Display Provider to request the position-related Remote ID data (if any) for UAS operating in one or more volumes.

(2) GET flight details: this interface enables a Net-RID Display Provider to request the additional non-position-related details (for example, UAS ID, UAS type, etc.) for a specific UAS operation.

NOTE A2.8—The separation of position data from other data serves multiple purposes. It contributes to efficiency by reducing the volume of data transferred; it avoids sharing data that has not been requested; and it provides a mechanism for monitoring for potential abuse cases such as a display provider mining information without an associated end user request.

A2.4.1.2 A Net-RID Display Provider shall (NET0720) query a Net-RID Service Provider for flights only for areas in which end users have requested information by means of the Remote ID Display Applications it services.

A2.4.1.3 Net-RID Display Provider shall (NET0730) minimally support the following interface for use by Net-RID

Service Providers, in accordance with the P2P (peer-to-peer) portion of the OpenAPI specification provided in Annex A4:

(1) POST Identification Service Area: this interface is called by a Net-RID Service Provider when the DSS informs it that a Net-RID Display Provider has a subscription for an area intersecting a new, modified, or deleted ISA

NOTE A2.9—This interface is the minimum required. A USS may implement additional interfaces for optimization or functional purposes as long as these minimum interfaces are supported.

A2.5 DSS-DSS Synchronization

A2.5.1 This section is applicable only to implementers of the DSS.

A2.5.2 This section describes requirements for the conceptual data that must be synchronized between DSS instances in a region. This is necessary to support diverse implementations of DSS instances while ensuring data consistency and interoperability.

A2.5.3 A specific synchronization protocol is not specified in this specification. It is expected that implementers of the DSS for a region will coordinate and agree on the synchronization protocol and other implementation details specific to a DSS region.

A2.5.4 *DSS Data Overview*—The DSS Airspace Representation (DAR) is the foundational data structure for the DSS. The DAR partitions airspace into discrete cells using some geocoordinate system. Entities and subscriptions are mapped onto one or more cells and persisted for their effective time periods. Consequently, the data for a cell is information about the entities and subscriptions (if any) that intersect the cell. Data that is synchronized between instances of the DSS can logically be understood as a key/value pair where the key is a cell ID and the value is the collection of information about the entities and subscriptions that intersect the cell. A synchronization protocol is used to synchronize this data between DSS instances.

A2.5.5 *Data Synchronization Requirements:*

A2.5.5.1 When synchronizing data, a DSS instance shall (DSS0110) authenticate with other DSS instances in the same region using an industry-standard authentication mechanism.

NOTE A2.10—OAuth 2.0 or later is an example of an industry-standard authentication mechanism by attestation, for those OAuth2 identity providers that require authentication to obtain credentials.

A2.5.5.2 Communication between DSSs in the same region shall (DSS0120) be encrypted using an industry-standard encryption mechanism with a minimum encryption strength of 128 bits.

NOTE A2.11—This requirement is intended to address both integrity and confidentiality of Remote ID data in transit. TLS is an example of an industry-standard authenticated encryption mechanism.

A2.5.5.3 DSS implementations shall (DSS0130) store and synchronize the following conceptual data for each DAR cell that has intersecting entities or subscriptions:

(1) The cell ID,

(2) For each subscription that intersects the cell:

(a) A unique ID for the subscription;

(b) The owner of the subscription;

(c) The URL a Net-RID Service Provider contacts to inform a Net-RID Display Provider of new, modified, or deleted ISAs;

(d) Start and end time of the subscription;

(e) Other cells the subscription intersects (to facilitate efficient removal of expired or deleted subscriptions); and

(f) The notification count for the subscription (used by a Net-RID Display Provider to detect missed notifications).

(3) For each ISA that intersects the cell:

(a) A unique ID for the Service Area,

(b) The owner of the Service Area,

(c) The URL a Net-RID Display Provider contacts to poll the owning Net-RID Service Provider when Remote ID data is needed for the Service Area,

(d) The start and end time of the Service Area, and

(e) Other cells the Service Area intersects (to facilitate efficient removal of expired or deleted Service Areas).

A2.6 DSS Testing

A2.6.1 *Approach*—DSS implementations are tested as part of a Net-RID Service Provider or Net-RID Display Provider, or both, when a new or modified DSS is being introduced. A test compliance matrix is provided in 10.3 for USSs. When testing includes a new or modified DSS, the test program must demonstrate interoperability at the DSS level. This is accomplished by including verification of data synchronization in all DSS instances when testing the various DSS interfaces. The following provides specific guidance for each of the DSS interfaces (defined in A2.3.1):

A2.6.1.1 *PUT Identification Service Area*—Tests must demonstrate that after an ISA is created or modified, it can be retrieved from all DSS instances for the region with consistent results. In addition, the end time for an ISA governs when the DSS automatically removes it from the DSS. Tests must demonstrate that automatic removal of ISAs occurs on all DSS instances.

A2.6.1.2 *DELETE Identification Service Area*—Tests must demonstrate that an ISA can be deleted on any DSS instance and the deletion is reflected on all other DSS instances.

A2.6.1.3 *PUT Subscription*—Tests must demonstrate that a subscription can be created on any instance and notifications for the subscription are triggered when intersecting ISAs are added or modified to all other instances. In addition, the end time for a subscription governs when the DSS automatically removes it from the DSS. Tests must demonstrate that automatic removal of subscriptions occurs on all DSS instances.

A2.6.1.4 *DELETE Subscription*—Tests must demonstrate that that a subscription can be deleted on any DSS instance and the deletion is reflected on all other DSS instances.

A2.6.1.5 *GET Subscription*—Tests must demonstrate that a specific subscription can be retrieved from any DSS instance with consistent results.

A2.6.1.6 *GET Subscriptions*—Tests must demonstrate that the complete set of subscriptions in an area for a Net-RID Display Provider can be retrieved from any DSS instance with consistent results.

TABLE A2.1 DSS Compliance Matrix

Req ID	Section Reference	Compliant (Y/N)	Notes
DSS0010	A2.3.1		
DSS0020	A2.3.1		
DSS0030	A2.3.1		
DSS0040	A2.3.1		
DSS050	A2.3.1		
DSS060	A2.3.1		
DSS0070	A2.3.1		
DSS0110	A2.5.5		
DSS0120	A2.5.5		
DSS0130	A2.5.5		
DSS0210	A2.5.5		

A2.6.2 *Test Environment Requirements*—DSS implementers shall (DSS0210) provide a persistently supported test instantiation of their DSS implementation, including at least 2 DSS instances, for use by Net-RID Service Providers and Display Providers for interoperability testing. This test instantiation must use the current deployed version of the implementer's DSS software and be configured to perform DSS-DSS synchronization with the test instantiations of other DSS providers in the same region.

A2.6.3 *Compliance Matrix:*

A2.6.3.1 The compliance matrix for the DSS is only applicable to implementations of a Net-RID Service Provider or Net-RID Display Provider, or both, that introduce a new DSS implementation. In addition to the compliance matrix provided below, implementers must demonstrate compliance with the applicable network requirements provided in 10.3 while applying the test guidance provided in A2.6.1.

A2.6.3.2 *DSS Compliance Matrix*—See Table A2.1.

A3. TABLES OF VALUES

TABLE A3.1 Broadcast Values

Value Name	Value	UoM	Req Ref	Section Ref
BcMinEIRP	Use Table A3.2	dbm	BPW0010	5.4.3
BCMinUasLocRefreshRate	1	Seconds	BUR0010	5.4.4
BCMinStaticRefreshRate	3	Seconds	BUR0020	5.4.4
BCMaxDataAge	1	Seconds	BUR0030	5.4.4

TABLE A3.2 BCMinEIRP Values

BCMinEIRP - Minimum Tx EIRP in Horizontal Plane			
Wave Law	2.4 GHz Wi-Fi	2.4 GHz BlueTooth	5.8 GHz Wi-fi
Type 1 (for example, USA)	+15 dbm	+5 dbm	+17 dbm
Type 2 (for example, China)	+11 dbm	+5 dbm	+17 dbm
Type 3 (for example, EU, Japan, Korea)	+11 dbm	+5 dbm	+4 dbm

TABLE A3.3 Network Values

Value Name	Value	UoM	Req Ref	Section Ref
NetMinUasLocRefreshFrequency	1	Hz	NET0040	5.5.2.4
NetMinUasLocRefreshPercentage	20	Percent	NET0040	5.5.2.4
NetMaxDisplayAreaDiagonal	3.6	km	NET0240	5.5.4.4
			NET0250	5.5.4.4
			NET0420	5.5.5.9
			NET0430	5.5.5.9
			NET0440	5.5.5.9
			NET0480	5.5.5.9
			NET0490	5.5.5.9
NetSpDataResponseTime95thPercentile	1	Seconds	NET0260	5.5.4.4
NetSpDataResponseTime99thPercentile	3	Seconds	NET0260	5.5.4.4
NetMaxNearRealTimeDataPeriod	60	Seconds	NET0260	5.5.4.4
			NET0270	5.5.5.9
			NET0450	5.5.5.9
			NET0610	A2.3.2
NetDpMaxDataRetentionPeriod	86,400	Seconds	NET0330	5.5.4.4
NetDpInitResponse95thPercentile	6	Seconds	NET0420	5.5.5.9
NetDpInitResponse99thPercentile	18	Seconds	NET0420	5.5.5.9
NetDpDataReponse95thPercentile	1	Seconds	NET0440	5.5.5.9
NetDpDataResponse99thPercentile	3	Seconds	NET0440	5.5.5.9
NetMinSessionLength	5	Seconds	NET0440	5.5.5.9
NetDpDetailsResponse95thPercentile	2	Seconds	NET0460	5.5.5.9
NetDpDetailsResponse99thPercentile	6	Seconds	NET0460	5.5.5.9
NetDetailsMaxDisplayAreaDiagonal	1	km	NET0460	5.5.5.9
			NET0480	5.5.5.9
			NET0490	5.5.5.9
NetMinClusterSize	15	Percent	NET0480	5.5.5.9
NetMinObfuscationDistance	300	m	NET0490	5.5.5.9
NetDSSMaxSubscriptionPerArea	10	Subscriptions	DSS0050	A2.3.1
NetDSSMaxSubscriptionDuration	24	Hours	DSS0060	A2.3.1

A4. USS-DSS AND USS-USS OpenAPI YAML DESCRIPTION

```
---
openapi: 3.0.2
info:
  title: Discovery and Synchronization Service – USS-DSS, USS-USS
  version: 1.0.0
  description: |-
    Interface to Discovery and Synchronization Service and service providers used by participating clients to
discover and inform other service providers.

    Unless otherwise specified, fields specified in a message but not declared in the API shall be ignored.
paths:
  /v1/dss/identification_service_areas:
    summary: Set of Identification Service Areas in the DSS.
    description: ””
    get:
      tags:
      - dss
      parameters:
      - name: area
        description: The area in which to search for Identification Service Areas.  Some
          Identification Service Areas near this area but wholly outside it may also
          be returned.
        schema:
          $ref: ’#/components/schemas/GeoPolygonString’
        in: query
        required: true
      - name: earliest_time
        description: If specified, indicates non-interest in any Identification Service
          Areas that end before this time.  RFC 3339 format, per OpenAPI specification.
        schema:
          format: date-time
          type: string
        in: query
        required: true
      - name: latest_time
        description: If specified, indicates non-interest in any Identification Service
          Areas that start after this time.  RFC 3339 format, per OpenAPI specification.
        schema:
          format: date-time
          type: string
        in: query
        required: true
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/SearchIdentificationServiceAreasResponse’
          description: Identification Service Areas were successfully retrieved.
        400:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: One or more input parameters were missing or invalid.
        401:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: The access token was decoded successfully but did not include
            a scope appropriate to this endpoint.
        413:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: The requested area was too large.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.read.identification_service_areas
      summary: /dss/identification_service_areas
      description: Retrieve all Identification Service Areas in the DAR for a given
        area during the given time.  Note that some Identification Service Areas returned
        may lie entirely outside the requested area.
  /v1/dss/identification_service_areas/{id}:
    summary: An Identification Service Area in the DSS.
    get:
      tags:
      - dss
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/GetIdentificationServiceAreaResponse’
          description: Full information of the Identification Service Area was retrieved
            successfully.
        400:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: One or more input parameters were missing or invalid.
        401:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: The access token was decoded successfully but did not include
            a scope appropriate to this endpoint.
        404:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: The requested Entity could not be found.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.read.identification_service_areas
      summary: /dss/identification_service_areas/{id}
      description: Retrieve full information of an Identification Service Area owned
        by the client.
    put:
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/CreateIdentificationServiceAreaParameters'
        required: true
      tags:
      - dss
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PutIdentificationServiceAreaResponse'
          description: An existing Identification Service Area was created successfully
            in the DSS.
        400:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: |-
            * One or more input parameters were missing or invalid.
            * The request attempted to mutate the Identification Service Area in a disallowed way.
        401:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: The access token was decoded successfully but did not include
            a scope appropriate to this endpoint.
        409:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: |-
            * An Identification Service Area with the specified ID already exists and is owned by a different client.
            * Despite repeated attempts, the DSS was unable to update the DAR because of other simultaneous
changes.
        413:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: The area of the Identification Service Area is too large.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.write.identification_service_areas
      summary: /dss/identification_service_areas/{id}
      description: |-
        Create a new Identification Service Area.  This call will fail if an Identification Service Area with the
same ID already exists.
```

```
    The DSS assumes the USS has already added the appropriate retention period to operation end time in
 'time_end' field before storing it.
  parameters:
  - name: id
    description: EntityUUID of the Identification Service Area.
    schema:
      $ref: '#/components/schemas/EntityUUID'
    in: path
    required: true
 /v1/dss/identification_service_areas/{id}/{version}:
  summary: A specific version of an Identification Service Area in the DSS.
  put:
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/UpdateIdentificationServiceAreaParameters'
      required: true
    tags:
    - dss
    responses:
      200:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/PutIdentificationServiceAreaResponse'
        description: An existing Identification Service Area was updated successfully
          in the DSS.
      400:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ErrorResponse'
        description: |-
          * One or more input parameters were missing or invalid.
          * The request attempted to mutate the Identification Service Area in a disallowed way.
      401:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ErrorResponse'
        description: Bearer access token was not provided in Authorization header,
          token could not be decoded, or token was invalid.
      403:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ErrorResponse'
        description: The access token was decoded successfully but did not include
          a scope appropriate to this endpoint.
      409:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ErrorResponse'
        description: |-
          * The specified Identification Service Area's current version does not match the provided version.
          * Despite repeated attempts, the DSS was unable to update the DAR because of other simultaneous
changes.
      413:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ErrorResponse'
        description: The area of the Identification Service Area is too large.
    security:
    - AuthFromAuthorizationAuthority:
      - dss.write.identification_service_areas
    summary: /dss/identification_service_areas/{id}/{version}
    description: |-
      Update an Identification Service Area.  The full content of the existing Identification Service Area will be
replaced with the provided information as only the most recent version is retained.

      The DSS assumes the USS has already added the appropriate retention period to operation end time in
 'time_end' field before storing it.  Updating 'time_start' is not allowed if it is before the current time.
  delete:
    tags:
    - dss
```

```
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/DeleteIdentificationServiceAreaResponse’
          description: Identification Service Area was successfully deleted from DSS.
        400:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: One or more input parameters were missing or invalid.
        401:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * The access token was decoded successfully but did not include a scope appropriate to this endpoint.
            * The Identification Service Area does not belong to the client requesting deletion.
        404:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Entity could not be deleted because it could not be found.
        409:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * The specified Identification Service Area’s current version does not match the provided version.
            * Despite repeated attempts, the DSS was unable to update the DAR because of other simultaneous
changes.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.write.identification_service_areas
      summary: /dss/identification_service_areas/{id}/{version}
      description: Delete an Identification Service Area.  USSs should not delete
        Identification Service Areas before the end of the last managed flight plus
        the retention period.
    parameters:
    - name: id
      description: EntityUUID of the Identification Service Area.
      schema:
        $ref: ’#/components/schemas/EntityUUID’
      in: path
      required: true
    - name: version
      description: Version string used to reference an Identification Service Area
        at a particular point in time. Any updates to an existing Identification Service
        Area must contain the corresponding version to maintain idempotent updates.
      schema:
        type: string
      in: path
      required: true
  /v1/uss/flights:
    summary: Basic operation details to meet remote ID requirements.
    description: This endpoint may be polled by remote ID display providers to display
      basic aircraft activity near a user in real time.
    get:
      tags:
      - p2p_rid
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/GetFlightsResponse’
          description: Flight information was successfully retrieved.
        400:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: One or more input parameters were missing or invalid.
        401:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: The access token was decoded successfully but did not include
            a scope appropriate to this endpoint.
        404:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: The requested view is entirely outside any remote ID service
            provision areas (PositionReporting Entity volumes) this provider has.  A
            display provider receiving this response should discontinue polling for
            this endpoint for the view port requested until otherwise directed by
            DSS interactions.
        413:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: The requested view rectangle was too large.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.read.identification_service_areas
      summary: /uss/flights
      description: Retrieve basic flight information from a remote ID service provider
        to meet remote ID requirements.  This endpoint will be polled by remote ID
        display providers to display basic aircraft activity near a user in real time.  All
        flights that have any recent positions (past [NetUasInAreaWindow] seconds)
        inside the requested view must be returned.
    parameters:
    - name: view
      description: ’The area of this view: lat1,lng1,lat2,lng2.  Size may not exceed
        a diagonal length specified in the standard.  The view is the smallest box
        bounded by the provided corner points.’
      schema:
        type: string
        example: 29.97816,31.13296,29.98025,31.13535
      in: query
      required: true
    - name: include_recent_positions
      description: |-
        If specified as true, include recent positions.  Otherwise, do not include recent positions.

        A display provider’s first poll of this endpoint for a particular view will usually request recent positions
while follow-up polls for that same view will usually not need to request recent positions.
      schema:
        type: boolean
      in: query
  /v1/uss/flights/{id}/details:
    summary: A remote ID flight reported by a remote ID service provider.
    get:
      tags:
      - p2p_rid
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/GetFlightDetailsResponse’
          description: Flight details were successfully retrieved.
        400:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: One or more input parameters were missing or invalid.
        401:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: The access token was decoded successfully but did not include
            a scope appropriate to this endpoint.
        404:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: The requested flight could not be found.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.read.identification_service_areas
      summary: /uss/flights/{id}/details
      description: Retrieve details of a flight from a remote ID service provider
        to meet remote ID requirements.
    parameters:
    - name: id
      description: Remote ID flight ID for this flight; may be obtained in general
        /flights query.
      schema:
        $ref: ’#/components/schemas/RIDFlightID’
      in: path
      required: true
  /v1/dss/subscriptions:
    summary: Subscriptions for airspace updates to a volume of interest.
    get:
      tags:
      - dss
      parameters:
      - name: area
        description: The area in which to search for Subscriptions.  Some Subscriptions
          near this area but wholly outside it may also be returned.
        schema:
          $ref: ’#/components/schemas/GeoPolygonString’
        in: query
        required: true
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/SearchSubscriptionsResponse’
          description: Subscriptions were retrieved successfully.
        400:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: One or more input parameters were missing or invalid.
        401:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: The access token was decoded successfully but did not include
```

```
            a scope appropriate to this endpoint.
        413:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: The requested area was too large.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.read.identification_service_areas
      summary: /dss/subscriptions
      description: |-
        Retrieve subscriptions intersecting an area of interest.  Subscription notifications are only triggered by
(and contain full information of) changes to, creation of, or deletion of, Entities referenced by or stored in the
DSS; they do not involve any data transfer (such as remote ID telemetry updates) apart from Entity information.

        Only Subscriptions belonging to the caller are returned.  This endpoint would be used if a USS lost track
of Subscriptions they had created and/or wanted to resolve an error indicating that they had too many existing
Subscriptions in an area.
  /v1/dss/subscriptions/{id}:
    summary: Subscription for airspace updates to a volume of interest.
    get:
      tags:
      - dss
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/GetSubscriptionResponse'
          description: Subscription information was retrieved successfully.
        400:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: One or more input parameters were missing or invalid.
        401:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: The access token was decoded successfully but did not include
            a scope appropriate to this endpoint.
        404:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ErrorResponse'
          description: A Subscription with the specified ID was not found.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.read.identification_service_areas
        - dss.write.identification_service_areas
      summary: /dss/subscriptions/{id}
      description: Verify the existence/valdity and state of a particular subscription.
    put:
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/CreateSubscriptionParameters'
        required: true
      tags:
      - dss
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/PutSubscriptionResponse'
          description: A new Subscription was created successfully.
        400:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * One or more input parameters were missing or invalid.
            * The request attempted to mutate the Subscription in a disallowed way.
        401:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * The access token was decoded successfully but did not include a scope appropriate to this endpoint or
the request.
            * An EntityType was specified in 'types_filter' to which the scopes included in the access token do not
provide access.
        409:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * A Subscription with the specified ID already exists and is owned by a different client.
            * Despite repeated attempts, the DSS was unable to update the DAR because of other simultaneous
changes.
        429:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Client already has too many Subscriptions in the area where
            a new Subscription was requested.  To correct this problem, the client
            may query GET /subscriptions to see which Subscriptions are counting against
            their limit.  This problem should not generally be encountered because
            the Subscription limit should be above what any consumer that reasonably
            aggregates their Subscriptions should request.  But, a Subscription limit
            is necessary to bound performance requirements for DSS instances and would
            likely be hit by, e.g., a large remote ID display provider that created
            a Subscription for each of their display client users’ views.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.read.identification_service_areas
      summary: /dss/subscriptions/{id}
      description: |-
        Create a subscription.  This call will fail if a Subscription with the same ID already exists.

        Subscription notifications are only triggered by (and contain full information of) changes to, creation of,
or deletion of, Entities referenced by or stored in the DSS; they do not involve any data transfer (such as remote
ID telemetry updates) apart from Entity information.
    parameters:
    - name: id
      description: SubscriptionUUID of the subscription of interest.
      schema:
        $ref: ’#/components/schemas/SubscriptionUUID’
      in: path
      required: true
  /v1/dss/subscriptions/{id}/{version}:
    summary: Specific version of a Subscription for airspace updates to a volume of
      interest.
    put:
      requestBody:
        content:
          application/json:
            schema:
              $ref: ’#/components/schemas/UpdateSubscriptionParameters’
        required: true
      tags:
      - dss
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/PutSubscriptionResponse’
          description: An existing Subscription was updated successfully.
        400:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * One or more input parameters were missing or invalid.
            * The request attempted to mutate the Subscription in a disallowed way.
        401:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * The access token was decoded successfully but did not include a scope appropriate to this endpoint or
the request.
            * An EntityType was specified in 'types_filter' to which the scopes included in the access token do not
provide access.
        409:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * The specified Subscriptions’s current version does not match the provided version.
            * Despite repeated attempts, the DSS was unable to update the DAR because of other simultaneous
changes.
        429:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Client already has too many Subscriptions in the area where
            a new Subscription was requested.  To correct this problem, the client
            may query GET /subscriptions to see which Subscriptions are counting against
            their limit.  This problem should not generally be encountered because
            the Subscription limit should be above what any consumer that reasonably
            aggregates their Subscriptions should request.  But, a Subscription limit
            is necessary to bound performance requirements for DSS instances and would
            likely be hit by, e.g., a large remote ID display provider that created
            a Subscription for each of their display client users’ views.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.read.identification_service_areas
      summary: /dss/subscriptions/{id}/{version}
      description: |-
        Update a Subscription.  The full content of the existing Subscription will be replaced with the provided
information as only the most recent version is retained.

        Subscription notifications are only triggered by (and contain full information of) changes to, creation of,
or deletion of, Entities referenced by or stored in the DSS; they do not involve any data transfer (such as remote
ID telemetry updates) apart from Entity information.
    delete:
      tags:
      - dss
      responses:
        200:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/DeleteSubscriptionResponse’
          description: Subscription was deleted successfully.
        400:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: One or more input parameters were missing or invalid.
        401:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * The access token was decoded successfully but did not include a scope appropriate to this endpoint.
            * The Entity does not belong to the client requesting deletion.
        404:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Subscription could not be deleted because it could not be found.
        409:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * The specified Subscriptions’s current version does not match the provided version.
            * Despite repeated attempts, the DSS was unable to update the DAR because of other simultaneous
changes.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.read.identification_service_areas
      summary: /dss/subscriptions/{id}/{version}
      description: Delete a subscription.
    parameters:
    - name: id
      description: SubscriptionUUID of the subscription of interest.
      schema:
        $ref: '#/components/schemas/SubscriptionUUID’
      in: path
      required: true
    - name: version
      description: Version string used to reference a Subscription at a particular
        point in time. Any updates to an existing Subscription must contain the corresponding
        version to maintain idempotent updates.
      schema:
        type: string
      in: path
      required: true
  /v1/uss/identification_service_areas/{id}:
    summary: Notifications of change to Identification Service Areas.
    post:
      requestBody:
        content:
          application/json:
            schema:
              $ref: ’#/components/schemas/PutIdentificationServiceAreaNotificationParameters’
        required: true
      tags:
      - p2p_rid
      responses:
        204:
          description: Information for Identification Service Area was updated successfully.
        400:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * One or more parameters were missing or invalid.
            * The Identification Service Area information could not be parsed, or contains illegal data.
        401:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: Bearer access token was not provided in Authorization header,
            token could not be decoded, or token was invalid.
        403:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: |-
            * The access token was decoded successfully but did not include a scope appropriate to this endpoint.
            * The client identified in the access token is not the owner of this Entity according to the receiving
client’s records.
        409:
          content:
            application/json:
              schema:
                $ref: ’#/components/schemas/ErrorResponse’
          description: The Entity version specified in this message is identical to
            a previously-received notification and the Entity is different.
      security:
      - AuthFromAuthorizationAuthority:
        - dss.write.identification_service_areas
      summary: /uss/identification_service_areas/{id}
      description: ’Hosted by a remote ID display provider using Subscriptions: publish
        new full information of an Identification Service Area to the host of this
        endpoint (usually because of instructions from the DSS triggered by a subscription).’
    parameters:
    - name: id
      description: EntityUUID of the Identification Service Area.
      schema:
        $ref: ’#/components/schemas/EntityUUID’
      in: path
      required: true
components:
  schemas:
    Volume3D:
      description: A three-dimensional geographic volume consisting of a vertically-extruded
        polygon.
      required:
      - footprint
      type: object
      properties:
        footprint:
          anyOf:
          - $ref: ’#/components/schemas/GeoPolygon’
          description: Projection of this volume onto the earth’s surface.
        altitude_lo:
          anyOf:
          - $ref: ’#/components/schemas/Altitude’
          description: Minimum bounding altitude of this volume.
        altitude_hi:
          anyOf:
          - $ref: ’#/components/schemas/Altitude’
          description: Maximum bounding altitude of this volume.
    Volume4D:
      description: Contiguous block of geographic spacetime.
      required:
      - spatial_volume
      type: object
      properties:
        spatial_volume:
          anyOf:
          - $ref: ’#/components/schemas/Volume3D’
          description: Constant spatial extent of this volume.
        time_start:
          format: date-time
          description: Beginning time of this volume.  RFC 3339 format, per OpenAPI
            specification.
          type: string
        time_end:
          format: date-time
          description: End time of this volume.  RFC 3339 format, per OpenAPI specification.
          type: string
    GetSubscriptionResponse:
      description: Response to DSS request for the subscription with the given id.
      required:
```

```
      - subscription
      type: object
      properties:
        subscription:
          $ref: '#/components/schemas/Subscription'
    SearchSubscriptionsResponse:
      description: Response to DSS query for subscriptions in a particular area.
      required:
      - subscriptions
      type: object
      properties:
        subscriptions:
          description: Subscriptions that overlap the specified area.
          type: array
          items:
            $ref: '#/components/schemas/Subscription'
    GetFlightsResponse:
      description: Response to remote ID provider query for flight information in
        an area of interest.
      required:
      - timestamp
      - flights
      type: object
      properties:
        timestamp:
          format: date-time
          description: The remote ID service provider's timestamp for when this information
            was current.  RFC 3339 format, per OpenAPI specification.
          type: string
        flights:
          description: A list of all flights that have been within the requested area
            within the remote ID retention period.  This includes flights that are
            not currently within the requested area, but were within the requested
            area within the remote ID retention period.
          type: array
          items:
            $ref: '#/components/schemas/RIDFlight'
    URL:
      description: Valid http or https URL.
      type: string
    SubscriptionNotificationIndex:
      format: int32
      description: Tracks the notifications sent for a subscription so the subscriber
        can detect missed notifications more easily.
      type: integer
    SubscriptionState:
      description: State of AreaSubscription which is causing a notification to be
        sent.
      required:
      - subscription
      type: object
      properties:
        subscription_id:
          $ref: '#/components/schemas/SubscriptionUUID'
        notification_index:
          $ref: '#/components/schemas/SubscriptionNotificationIndex'
    UUIDv4:
      description: UUID v4.
      maxLength: 36
      minLength: 36
      type: string
      example: 03e5572a-f733-49af-bc14-8a18bd53ee39
    Version:
      description: A version string used to reference an object at a particular point
        in time. Any updates to an object must contain the corresponding version to
        maintain idempotent updates.
      type: string
    EntityUUID:
      anyOf:
      - $ref: '#/components/schemas/UUIDv4'
      description: Universally-unique identifier for an Entity communicated through
        the DSS.  Formatted as UUIDv4.
    SubscriptionUUID:
      anyOf:
      - $ref: '#/components/schemas/UUIDv4'
      description: Universally-unique identifier for a Subscription communicated through
        the DSS.  Formatted as UUIDv4.
    RIDFlightID:
      description: |-
        ID, unique to a remote ID service provider, which identifies a particular flight for which the remote ID
service provider is providing remote ID services.

        The following characters are not allowed: \0 \t \r \n # % / : ? @ [ \ ]
      maxLength: 255
      minLength: 1
      type: string
      example: uss1.JA6kHYCcByQ-6AfU
    RIDAuthData:
      description: Additional authentication data.
      required:
      - format
      - data
      type: object
      properties:
        format:
          description: Format of additional authentication data.
          type: string
        data:
          description: Authentication data in form specified by 'format'.
          type: string
    GetFlightDetailsResponse:
      description: Response to remote ID provider query for details about a specific
        flight.
      required:
      - details
      type: object
      properties:
        details:
          $ref: ’#/components/schemas/RIDFlightDetails’
    HorizontalAccuracy:
      description: |-
        This is the NACp enumeration from ADS-B, plus 1m for a more complete range for UAs.

        'HAUnknown': Unknown horizontal accuracy
        'HA10NMPlus': > 10NM (18.52km)
        'HA10NM':< 10NM (18.52km)
        'HA4NM':< 4NM (7.408km)
        'HA2NM':< 2NM (3.704km)
        'HA1NM':< 1NM (1852m)
        'HA05NM':< 0.5NM (926m)
        'HA03NM':< 0.3NM (555.6m)
        'HA01NM':< 0.1NM (185.2m)
        'HA005NM':< 0.05NM (92.6m)
        'HA30m':< 30m
        'HA10m':< 10m
        'HA3m':< 3m
        'HA1m':< 1m
      enum:
      - HAUnknown
      - HA10NMPlus
      - HA10NM
      - HA4NM
      - HA2NM
      - HA1NM
      - HA05NM
      - HA03NM
      - HA01NM
      - HA005NM
      - HA30m
      - HA10m
      - HA3m
      - HA1m
      type: string
    VerticalAccuracy:
      description: |-
        This is the GVA enumeration from ADS-B, plus some finer values for UAs.

        'VAUnknown': Unknown vertical accuracy
        'VA150mPlus': > 150m
        'VA150m':< 150m
        'VA45m':< 45m
        'VA25m':< 25m
        'VA10m':< 10m
        'VA3m':< 3m
        'VA1m':< 1m
      enum:
```

```
      - VAUnknown
      - VA150mPlus
      - VA150m
      - VA45m
      - VA25m
      - VA10m
      - VA3m
      - VA1m
      type: string
    ErrorResponse:
      description: Data provided when an off-nominal condition was encountered.
      type: object
      properties:
        message:
          description: Human-readable message indicating what error occurred and/or
            why.
          type: string
    DeleteSubscriptionResponse:
      description: Response for a successful request to delete a Subscription.
      required:
      - subscription
      type: object
      properties:
        subscription:
          anyOf:
          - $ref: ’#/components/schemas/Subscription’
          description: The Subscription which was deleted.
    SpeedAccuracy:
      description: |-
        This is the same enumeration scale and values from ADS-B NACv.

        'SAUnknown': Unknown speed accuracy
        'SA10mpsPlus': > 10 m/s
        'SA10mps':< 10 m/s
        'SA3mps':< 3 m/s
        'SA1mps':< 1 m/s
        'SA03mps':< 0.3 m/s
      enum:
      - SAUnknown
      - SA10mpsPlus
      - SA10mps
      - SA3mps
      - SA1mps
      - SA03mps
      type: string
    RIDAircraftPosition:
      description: Position of an aircraft as reported for remote ID purposes.
      required:
      - lat
      - lng
      - alt
      type: object
      properties:
        lat:
          $ref: ’#/components/schemas/Latitude’
        lng:
          $ref: ’#/components/schemas/Longitude’
        alt:
          format: float
          description: ’Geodetic altitude (NOT altitude above launch, altitude above
            ground, or EGM96): aircraft distance above the WGS84 ellipsoid as measured
            along a line that passes through the aircraft and is normal to the surface
            of the WGS84 ellipsoid.  This value is provided in meters and must have
            a minimum resolution of 1 meter.’
          type: number
          example: 1321.2
        accuracy_h:
          anyOf:
          - $ref: ’#/components/schemas/HorizontalAccuracy’
          description: Horizontal error that is likely to be present in this reported
            position.  Required when 'extrapolated' field is true and always in the
            entry for the current state.
        accuracy_v:
          anyOf:
          - $ref: ’#/components/schemas/VerticalAccuracy’
          description: Vertical error that is likely to be present in this reported
            position.  Required when 'extrapolated' field is true and always in the
            entry for the current state.
        extrapolated:
          description: True if this position was generated primarily by computation
            rather than primarily from a direct instrument measurement.  Assumed false
            if not specified.
          type: boolean
        pressure_altitude:
          format: float
          description: The uncorrected altitude (based on reference standard 29.92
            inHg, 1013.25 mb) provides a reference for algorithms that utilize ″altitude
            deltas″ between aircraft.  This value is provided in meters and must have
            a minimum resolution of 1 meter.
          type: number
    GeoPolygonString:
      description: |-
        Plain-string representation of a geographic polygon consisting of at least three geographic points
describing a closed polygon on the earth.  Each point consists of latitude,longitude in degrees.  Points are also
comma-delimited, so this parameter will look like 'lat1,lng1,lat2,lng2,lat3,lng3,...'  Latitude values must fall in
the range [-90, 90] and longitude values must fall in the range [-180, 180].

        All of the requirements and clarifications for GeoPolygon apply to GeoPolygonString as well.
      type: string
    RIDHeight:
      description: A relative altitude for the purposes of remote ID.
      required:
      - distance
      - reference
      type: object
      properties:
        distance:
          format: float
          description: Distance above reference datum.  This value is provided in
            meters and must have a minimum resolution of 1 meter.
          type: number
        reference:
          description: The reference datum above which the height is reported.
          enum:
          - TakeoffLocation
          - GroundLevel
          type: string
    Latitude:
      format: double
      description: Degrees of latitude north of the equator, with reference to the
        WGS84 ellipsoid.
      maximum: 90
      exclusiveMaximum: false
      minimum: -90
      exclusiveMinimum: false
      type: number
      example: 34.123
    Longitude:
      format: double
      description: Degrees of longitude east of the Prime Meridian, with reference
        to the WGS84 ellipsoid.
      maximum: -180
      exclusiveMaximum: false
      minimum: 180
      exclusiveMinimum: false
      type: number
      example: -118.456
    LatLngPoint:
      description: Point on the earth’s surface.
      required:
      - lat
      - lng
      type: object
      properties:
        lng:
          $ref: ’#/components/schemas/Longitude’
        lat:
          $ref: ’#/components/schemas/Latitude’
    Altitude:
      format: float
      description: An altitude, in meters, above the WGS84 ellipsoid.
      type: number
      example: 19.5
    RIDFlight:
      description: Description of a remote ID flight.
      required:
```

```
      - id
      - aircraft_type
      type: object
      properties:
        id:
          $ref: ’#/components/schemas/RIDFlightID’
        aircraft_type:
          anyOf:
          - $ref: ’#/components/schemas/RIDAircraftType’
          description: Generic type of aircraft.
        current_state:
          anyOf:
          - $ref: ’#/components/schemas/RIDAircraftState’
          description: |-
            The most up-to-date state of the aircraft.  Required when the aircraft is currently in the requested
area unless 'volumes' is specified.

            If current data is not being received from the UAS by the Service Provider, the lack of change in this
field is sufficient to indicate that current data is not being received.
        volumes:
          description: The set of spacetime volumes the aircraft is within.  Required
            if 'current_state' is not specified.  The fields 'time_start' and 'time_end'
            are required if 'current_state' is not specified.
          minItems: 1
          type: array
          items:
            $ref: ’#/components/schemas/Volume4D’
        simulated:
          description: If specified as true, this flight is not a physical aircraft;
            it is just a simulation to test the system.
          type: boolean
        recent_positions:
          description: |-
            A short collection of recent aircraft movement, specified only when 'include_recent_positions' is
true.  If 'volumes' is not specified and 'include_recent_positions' is true, then this field is required.

            Recent positions provided in this field must conform to requirements in the standard which generally
prohibit including positions outside the requested area except transitionally when the aircraft enters or exits the
requested area, and which prohibit including positions that not sufficiently recent.

            Note that a UI should not draw a connective line between two consecutive position reports that both lie
outside the requested area.
          type: array
          items:
            $ref: ’#/components/schemas/RIDRecentAircraftPosition’
    GeoPolygon:
      description: |-
        An enclosed area on the earth.
        The bounding edges of this polygon shall be the shortest paths between connected vertices.  This means, for
instance, that the edge between two points both defined at a particular latitude is not generally contained at that
latitude.
        The winding order shall be interpreted as the order which produces the smaller area.
        The path between two vertices shall be the shortest possible path between those vertices.
        Edges may not cross.
        Vertices may not be duplicated.  In particular, the final polygon vertex shall not be identical to the first
vertex.
      required:
      - vertices
      type: object
      properties:
        vertices:
          minItems: 3
          type: array
          items:
            $ref: ’#/components/schemas/LatLngPoint’
    PutIdentificationServiceAreaResponse:
      description: Response to a request to create or update a reference to an Identification
        Service Area in the DSS.
      required:
      - service_area
      - subscribers
      type: object
      properties:
        subscribers:
          description: DSS subscribers that this client now has the obligation to
            notify of the Identification Service Area changes just made.  This client
            must call POST for each provided URL according to the '/uss/identification_service_areas/{id}'
            path API.
          type: array
          items:
            $ref: ’#/components/schemas/SubscriberToNotify’
        service_area:
          anyOf:
          - $ref: ’#/components/schemas/IdentificationServiceArea’
          description: Resulting service area stored in DSS.
    SearchIdentificationServiceAreasResponse:
      description: Response to DSS query for Identification Service Areas in an area
        of interest.
      required:
      - service_areas
      type: object
      properties:
        service_areas:
          description: Identification Service Areas in the area of interest.
          type: array
          items:
            $ref: ’#/components/schemas/IdentificationServiceArea’
    PutIdentificationServiceAreaNotificationParameters:
      description: Parameters of a message informing of new full information for an
        Identification Service Area.  Pushed (by a client, not the DSS) directly to
        clients with subscriptions when another client makes a change to airspace
        within a cell with a subscription.
      required:
      - subscriptions
      type: object
      properties:
        service_area:
          anyOf:
          - $ref: ’#/components/schemas/IdentificationServiceArea’
          description: |-
            Identification Service Area that the notifying client changed or created.

            If this field is populated, the Identification Service Area was created or updated.  If this field is not
populated, the Identification Service Area was deleted.
        subscriptions:
          description: Subscription(s) prompting this notification.
          type: array
          items:
            $ref: ’#/components/schemas/SubscriptionState’
        extents:
          $ref: ’#/components/schemas/Volume4D’
          description: |-
            The new or updated extents of the Identification Service Area.

            Omitted if Identification Service Area was deleted.
    SubscriberToNotify:
      description: Subscriber to notify of a creation/change/deletion of a change
        in the airspace.  This is provided by the DSS to a client changing the airspace,
        and it is the responsibility of the client changing the airspace (they will
        receive a set of these notification requests) to send a notification to each
        specified 'url'.
      required:
      - subscriptions
      - url
      type: object
      properties:
        subscriptions:
          description: Subscription(s) prompting this notification.
          type: array
          items:
            $ref: ’#/components/schemas/SubscriptionState’
        url:
          anyOf:
          - $ref: ’#/components/schemas/URL’
          description: The endpoint that the client mutating the airspace should provide
            the update to.  API depends on the DSS action taken that triggered this
            notification request.
    DeleteIdentificationServiceAreaResponse:
      description: Response for a request to delete an Identification Service Area.
      required:
      - service_area
      - subscribers
      type: object
      properties:
        service_area:
          anyOf:
```

```
          - $ref: ’#/components/schemas/IdentificationServiceArea’
          description: Indentification Service Area that was just deleted.
        subscribers:
          description: DSS subscribers that this client now has the obligation to
            notify of the Identification Service Area just deleted.  This client must
            call POST for each provided URL according to the '/uss/identification_service_areas'
            path API.
          type: array
          items:
            $ref: ’#/components/schemas/SubscriberToNotify’
    IdentificationServiceAreaURL:
      anyOf:
      - $ref: ’#/components/schemas/URL’
      description: The URL at which notifications regarding Identification Service
        Areas may be delivered.  See the '/uss/identification_service_areas/{id}'
        path for specification of this endpoint.
      type: string
      example: https://example.com/identification_service_areas
    SubscriptionCallbacks:
      description: Endpoints that should be called when an applicable event occurs.  At
        least one field must be specified.
      type: object
      properties:
        identification_service_area_url:
          anyOf:
          - $ref: ’#/components/schemas/IdentificationServiceAreaURL’
          description: If specified, other clients will be instructed by the DSS to
            call this endpoint when an Identification Service Area relevant to this
            Subscription is created, modified, or deleted.  Must implement PUT and
            DELETE according to the '/uss/identification_service_areas/{id}' path
            API.
    PutSubscriptionResponse:
      description: Response for a request to create or update a subscription.
      required:
      - subscription
      type: object
      properties:
        service_areas:
          description: Identification Service Areas in or near the subscription area
            at the time of creation/update, if 'identification_service_area_url' callback
            was specified.
          type: array
          items:
            $ref: ’#/components/schemas/IdentificationServiceArea’
        subscription:
          anyOf:
          - $ref: ’#/components/schemas/Subscription’
          description: Result of the operation on the subscription.
    RIDOperationalStatus:
      description: Indicates operational status of associated aircraft.
      enum:
      - Undeclared
      - Ground
      - Airborne
      type: string
    RIDFlightDetails:
      description: Details about a flight reported by a remote ID service provider.  At
        least one of the registration or serial fields must be filled if required
        by CAA.
      required:
      - id
      type: object
      properties:
        id:
          description: ID for this flight, matching argument in request.
          type: string
          example: a3423b-213401-0023
        operator_id:
          description: ’CAA-issued registration/license ID for the remote pilot or
            operator. ’
          type: string
        operator_location:
          anyOf:
          - $ref: ’#/components/schemas/LatLngPoint’
          description: Location of party controlling the aircraft.
        operation_description:
          description: Free-text field that enables the operator to describe the purpose
            of a flight, if so desired.
          type: string
          example: SafeFlightDrone company doing survey with DJI Inspire 2. See my
            privacy policy www.example.com/privacy.
        auth_data:
          $ref: '#/components/schemas/RIDAuthData'
        serial_number:
          description: Can be specified when no registration ID exists and required
            by law in a region. This is expressed in the ANSI/CTA-2063-A Physical Serial
            Number format.
          type: string
          example: INTCJ123-4567-890
        registration_number:
          description: If a CAA provides a method of registering UAS, this number
            is provided by the CAA or its authorized representative.  Required when
            required by law in a region.
          type: string
          example: FA12345897
    RIDRecentAircraftPosition:
      description: ""
      required:
      - time
      - position
      type: object
      properties:
        time:
          format: date-time
          description: Time at which this position applied.  RFC 3339 format, per
            OpenAPI specification.
          type: string
        position:
          $ref: '#/components/schemas/RIDAircraftPosition'
    GetIdentificationServiceAreaResponse:
      description: Response to DSS request for the identification service area with
        the given id.
      required:
      - service_area
      type: object
      properties:
        service_area:
          $ref: '#/components/schemas/IdentificationServiceArea'
    CreateIdentificationServiceAreaParameters:
      description: Parameters for a request to create an Identification Service Area
        in the DSS.
      required:
      - extents
      - flights_url
      type: object
      properties:
        extents:
          anyOf:
          - $ref: '#/components/schemas/Volume4D'
          description: |-
            The bounding spacetime extents of this Identification Service Area.  End time must be specified.  If
start time is not specified, it will be set to the current time.  Start times in the past should be rejected by the
DSS, except that it may adjust very recent start times to the current time.

            These extents should not reveal any sensitive information about the flight or flights within them.  This
means, for instance, that extents should not tightly-wrap a flight path, nor should they generally be centered
around the takeoff point of a single flight.
        flights_url:
          $ref: '#/components/schemas/RIDFlightsURL'
    UpdateIdentificationServiceAreaParameters:
      description: Parameters for a request to update an Identification Service Area
        in the DSS.
      required:
      - extents
      - flights_url
      type: object
      properties:
        extents:
          anyOf:
          - $ref: '#/components/schemas/Volume4D'
          description: |-
            The bounding spacetime extents of this Identification Service Area.  End time must be specified.  If
start time is not specified, it will remain unchanged.  Start times in the past should be rejected by the DSS unless
they are unchanged from the Identification Service Area's current start time.
```

```
            These extents should not reveal any sensitive information about the flight or flights within them.  This
means, for instance, that extents should not tightly-wrap a flight path, nor should they generally be centered
around the takeoff point of a single flight.
        flights_url:
          $ref: ’#/components/schemas/RIDFlightsURL’
    CreateSubscriptionParameters:
      description: Parameters for a request to create a subscription in the DSS.
      required:
      - extents
      - callbacks
      type: object
      properties:
        extents:
          anyOf:
          - $ref: ’#/components/schemas/Volume4D’
          description: |-
            The spacetime extents of the volume to subscribe to.

            This subscription will automatically be deleted after its end time if it has not been refreshed by
then.  If end time is not specified, the value will be chosen automatically by the DSS.

            Note that some Entities triggering notifications may lie entirely outside the requested area.
        callbacks:
          $ref: ’#/components/schemas/SubscriptionCallbacks’
    UpdateSubscriptionParameters:
      description: Parameters for a request to update a subscription in the DSS.
      required:
      - extents
      - callbacks
      type: object
      properties:
        extents:
          anyOf:
          - $ref: ’#/components/schemas/Volume4D’
          description: |-
            The spacetime extents of the volume to subscribe to.

            This subscription will automatically be deleted after its end time if it has not been refreshed by
then.  If end time is not specified, the value will be chosen automatically by the DSS.

            Note that some Entities triggering notifications may lie entirely outside the requested area.
        callbacks:
          $ref: ’#/components/schemas/SubscriptionCallbacks’
    Subscription:
      description: Specification of a geographic area that a client is interested
        in on an ongoing basis (e.g., ″planning area″).  Internal to the DSS.
      required:
      - id
      - url
      - notification_index
      - owner
      - version
      - callbacks
      type: object
      properties:
        id:
          anyOf:
          - $ref: ’#/components/schemas/SubscriptionUUID’
          description: Unique identifier for this subscription.
        callbacks:
          $ref: ’#/components/schemas/SubscriptionCallbacks’
        owner:
          description: Assigned by the DSS based on creating client’s ID (via access
            token).  Used for restricting mutation and deletion operations to owner.
          type: string
          example: myuss
        notification_index:
          $ref: ’#/components/schemas/SubscriptionNotificationIndex’
        time_end:
          format: date-time
          description: If set, this subscription will be automatically removed after
            this time.  RFC 3339 format, per OpenAPI specification.
          type: string
        time_start:
          format: date-time
          description: If set, this Subscription will not generate any notifications
            before this time.  RFC 3339 format, per OpenAPI specification.
          type: string
        version:
          $ref: ’#/components/schemas/Version’
    IdentificationServiceArea:
      description: An Identification Service Area (area in which remote ID services
        are being provided).  The DSS reports only these declarations and clients
        must exchange flight information peer-to-peer.
      required:
      - flights_url
      - owner
      - time_start
      - time_end
      - version
      - id
      type: object
      properties:
        flights_url:
          $ref: ’#/components/schemas/RIDFlightsURL’
        owner:
          description: Assigned by the DSS based on creating client’s ID (via access
            token).  Used for restricting mutation and deletion operations to owner.
          type: string
          example: myuss
        time_start:
          format: date-time
          description: Beginning time of service.  RFC 3339 format, per OpenAPI specification.
          type: string
        time_end:
          format: date-time
          description: End time of service.  RFC 3339 format, per OpenAPI specification.
          type: string
        version:
          $ref: ’#/components/schemas/Version’
        id:
          anyOf:
          - $ref: ’#/components/schemas/EntityUUID’
          description: Unique identifier for this Identification Service Area.
    RIDAircraftState:
      description: State of an aircraft for the purposes of remote ID.
      required:
      - timestamp
      - timestamp_accuracy
      - accuracy_h
      - accuracy_v
      - speed
      - vertical_speed
      - track
      - speed_accuracy
      - position
      type: object
      properties:
        timestamp:
          format: date-time
          description: Time at which this state was valid.  This may be the time coming
            from the source, such as a GPS, or the time when the system computes the
            values using an algorithm such as an Extended Kalman Filter (EKF).  Timestamp
            must be expressed with a minimum resolution of 1/10th of a second.  RFC
            3339 format, per OpenAPI specification.
          type: string
        timestamp_accuracy:
          format: float
          description: ’Declaration of timestamp accuracy, which is the largest difference
            between Timestamp and true time of applicability for any of the following
            fields: Latitude, Longitude, Geodetic Altitude, Pressure Altitude of Position,
            Height. to determine time of applicability of the location data provided.  Expressed
            in seconds, precise to 1/10ths of seconds. The accuracy reflects the 95%
            uncertainty bound value for the timestamp.’
          minimum: 0
          exclusiveMinimum: false
          type: number
        operational_status:
          $ref: ’#/components/schemas/RIDOperationalStatus’
        position:
          $ref: ’#/components/schemas/RIDAircraftPosition’
        track:
          format: float
          description: Direction of flight expressed as a ″True North-based″ ground
            track angle.  This value is provided in degrees East of North with a minimum
            resolution of 1 degree.
          maximum: 360
          exclusiveMaximum: true
          minimum: 0
          exclusiveMinimum: false
          type: number
        speed:
          format: float
          description: Ground speed of flight in meters per second.
          minimum: 0
          exclusiveMinimum: false
          type: number
          example: 1.9
        speed_accuracy:
          anyOf:
          - $ref: '#/components/schemas/SpeedAccuracy'
          description: Accuracy of horizontal ground speed.
        vertical_speed:
          format: float
          description: Speed up (vertically) WGS84-HAE, m/s.
          type: number
          example: 0.2
        height:
          $ref: '#/components/schemas/RIDHeight'
        group_radius:
          format: float
          description: Farthest horizontal distance from reported group location at
            which an aircraft in the group may be located (meters).  This value contains
            the "Operating Area Radius" data from the common data dictionary when
            group operation area is specified by point-radius.
          minimum: 0
          exclusiveMinimum: true
          type: number
        group_ceiling:
          format: float
          description: Maximum altitude (meters WGS84-HAE) of Group Operation.  This
            value contains the "Operating Area Ceiling" data from the common data
            dictionary when group operation area is specified by point-radius.
          type: number
        group_floor:
          format: float
          description: Minimum altitude (meters WGS84-HAE) of Group Operation.  If
            not specified, ground level shall be assumed.  This value contains the
            "Operating Area Floor" data from the common data dictionary when group
            operation area is specified by point-radius.
          type: number
        group_count:
          format: int32
          description: When operating a group (or formation or swarm), number of aircraft
            in group.  This value contains the "Operating Area Count" data from the
            common data dictionary when group operation area is specified by point-radius.
          minimum: 1
          exclusiveMinimum: false
          type: integer
        group_time_start:
          format: date-time
          description: Time at which a group operation starts.  This value contains
            the "Operation Area Start" data from the common data dictionary when group
            operation area is specified by point-radius.
          type: string
        group_time_end:
          format: date-time
          description: Time at which a group operation starts.  This value contains
            the "Operation Area End" data from the common data dictionary when group
            operation area is specified by point-radius.
          type: string
    RIDAircraftType:
      description: |-
        Type of aircraft for the purposes of remote ID.

        'VTOL' is a fixed wing aircraft that can take off vertically. 'Helicopter' includes multirotor.
      enum:
      - NotDeclared
      - Aeroplane
      - Helicopter
      - Gyroplane
      - VTOL
      - Ornithopter
      - Glider
      - Kite
      - FreeBalloon
      - CaptiveBalloon
      - Airship
      - FreeFallOrParachute
      - Rocket
      - TetheredPoweredAircraft
      - GroundObstacle
      - Other
      type: string
    RIDFlightsURL:
      anyOf:
      - $ref: ’#/components/schemas/URL’
      description: |-
        The URL at which the remote ID flights and their details may be retrieved.  See '/flights' and
  ' /flights /{id} /details' paths for specification of this endpoint.
        This URL is the base flights resource.  If this URL is specified as https://example.com/flights then the flight
details for a particular {id} may be obtained at the URL https://example.com/flights/{id}/details.  This URL may not
have a trailing / character.
      type: string
      example: https://example.com/flights
  securitySchemes:
    AuthFromAuthorizationAuthority:
      flows:
        clientCredentials:
          tokenUrl: https://example.com/oauth/token
          scopes:
            dss.read.identification_service_areas: Client may query, read, or create
              subscriptions for Identification Service Areas for remote ID.  A remote
              ID display provider shall be granted this scope upon request.
            dss.write.identification_service_areas: Client may create, edit, and delete
              Identification Service Areas for remote ID.  A remote ID service provider
              shall be granted this scope upon request.
      type: oauth2
      description: |-
        Authorization from, or on behalf of, an authorization authority.  This authority shall issue access tokens
that are JSON Web Tokens as defined in RFC 7519, using the 'RS256' algorithm for the signature, and publish to all
providers the public key for verifying that signature.

        The following fields shall be included in the JWT claim for access tokens issued by this authority:
        * 'iss', with the URL at which the token generation request was received.
        * 'exp', with a time no further than 1 hour in the future.
        * 'sub', with unique ID of the client requesting the access token.
        * 'scope', with an array of strings indicating the scopes granted.

        Multiple scopes for a single token shall be granted if requested.  The tokens granted by this authority
shall protect against reuse of received tokens to impersonate the sender to other recipients (via use of the 'aud'
claim or other means).

        Clients shall provide these access tokens in an 'Authorization' header in the form 'Bearer <token>' in
accordance with RFC 6750.
security:
- AuthFromAuthorizationAuthority:
  - dss.read.identification_service_areas
  - dss.write.identification_service_areas
tags:
- name: dss
  description: Endpoints exposed by the DSS server.
- name: p2p_rid
  description: Endpoints exposed by remote ID service providers for peer-peer communication.
```

APPENDIXES

(Nonmandatory Information)

X1. PERFORMANCE CHARACTERISTICS

X1.1 The inherent characteristics of Broadcast and Network Remote ID results in different performance under varying circumstances. This appendix describes the varying characteristics for each method of performing Remote ID.

X1.2 Broadcast Remote ID Characteristics:

X1.2.1 *General*—The general characteristics described in this section apply to all forms of broadcast methods defined in this specification, including Bluetooth 4 (BT4), Bluetooth 5 Long Range (BT5LR), and Wi-Fi.

X1.2.1.1 The UA can always deterministically and measurably meet the requirement and conform to this specification without any dependency on externalities such as ground infrastructure. The broadcast function is always transmitting whether a receiver is there to "hear" it or not.

X1.2.1.2 Unlike typical wireless applications that require a "connection," all of the broadcast methods in this specification use a one-way "connection-less" broadcast method that does not require the receiver to transmit anything back to the broadcaster. There is no "pairing" or "connecting" to an Access Point using any of the broadcast technologies in this specification. Therefore the amplitude limitations of broadcast are only limited by the transmitter's power levels (not the receiver).

X1.2.1.3 That being said, the receiver antenna sensitivity and received noise interference can have an impact on the reception range.

X1.2.1.4 Broadcast performs best with the longest range where Electromagnetic Interference (EMI) is minimal. Conversely strong EMI will tend to reduce the range of broadcast.

X1.2.1.5 Broadcast can uniquely satisfy Remote ID requirements where there is no other means of network coverage and where there is no established federation of network servers and infrastructure.

X1.2.2 *Bluetooth 4 (BT4) aka Bluetooth Legacy:*

X1.2.2.1 *Industry Standard Approach*—BT4 has clear standards (see Bluetooth Core Specification 5.0, Volume 6, Part B, Sections 2.1 and 2.3.1) outlined and implementations existing for many years.

X1.2.2.2 *Technology*—The technique the standard uses is called "Connectionless Advertisements." These advertisements continually broadcast without the need of a receiver to acknowledge. It is a similar mechanism to when a device is in "pairing mode" and a receiver will list the devices ready to pair. The receiver can see these devices because they are broadcasting pairing advertisements. These advertisements occur on three dedicated channels (of 40) called "beacon channels" that are strategically on the 2.4 GHz spectrum away where Wi-Fi traffic most commonly occurs.

X1.2.2.3 *Receiver Support*—BT4 is widely supported on the receiver side hardware and software. Both iOS and Android have to, with widely used APIs, enable software development of receiving applications without the need of auxiliary hardware or dongles.

X1.2.2.4 *Range*—Because of the speed being limited to 1 Mb, the range is often better than expected (for a given power level). It has been tested to have about 400 m in a rural environment. Lots of RF interference could reduce this range. For the power levels outlined in this specification, BT4 will generally have a shorter range than Wi-Fi and Bluetooth 5 at the power levels defined in this specification.

X1.2.2.5 *Limited Payload (26 Bytes)*—Because of the limited payload size, the full complement of broadcast Remote ID data cannot fit in a single message. Consequently, this specification uses multiple small messages to transmit the data. Since the data is transmitted over multiple messages, authentication mechanisms are limited to verification of static information transmitted (to maintain coherency between authentication data and data to be validated).

X1.2.2.6 *Transmitter Hardware*—Transmitter hardware modules are available for commodity prices. Such hardware modules have a programmable microprocessor with serial (UART) interfaces for getting telemetry data from the UA which is retransmitted through the connected antenna.

X1.2.2.7 Bluetooth Transmitters do not generally exist on UAs, so they would have to be retrofitted or engineered into future UAS products to use this technology

X1.2.3 *Bluetooth 5 Long Range (BT5LR):*

X1.2.3.1 *Industry Standard Approach*—Although BT5LR is relatively new, clear standards exist on how to implement and such standards are being widely adopted.

X1.2.3.2 *Technology*—B5LR incorporates two new key technologies introduced in the Bluetooth 5.0 specification:

(1) A coded phy that encodes the data to operate at a much slower rate than BT4 (125 kbps) and it also includes Forward Error Correction (FEC). This slower rate coupled with FEC drastically increases the potential range.

(2) Advertising Extensions that allow small "Pointer Beacons" to tell the receiver where (on the other 40 channels) to read the larger payload. This allows larger (255 byte) payloads and a wider distribution of traffic over 40 channels.

X1.2.3.3 *Range*—Given the same environment, BT5LR generally gets about 4 times that of BT4. In rural environments, tests are showing over 1 km. Lots of RF interference could reduce this range.

X1.2.3.4 *Large Payload*—Since BT5LR allows for large (255 byte) payloads, it enables more data architecture flexibility allows a full set of messages to be sent in a single packet. By doing this, if desired, a complementary authentication payload can be sent in full coherency with the rest of the payload, therefore allowing the entire payload to be verified authentic.

X1.2.3.5 *Receiver Support*—Although BT5LR is widely enabled on the transmitter side, it is still new and only a few (smartphone) products support it on the receiver side. As of this writing, we have received information that newer smartphones from Samsung (Galaxy S10+), Xiaomi (Mi 9), and Huawei (Mate 20 Pro) support BT5LR. Also, long range mode is an optional mode of BT5 so some products that claim BT5 may or may not have adopted the Long Range mode. However, BT5LR transmitters are compatible with BT4 receivers as long as the BT5 radios transmit in Legacy Mode. Therefore the standard requires that those using BT5 must perform both Long Range mode and BT4 (Legacy) Mode to achieve both universal compatibility and the benefits of Long Range. Although BT5LR is not yet widely enabled in current smartphones, if one desires to receive the long range broadcasts, this can be achieved by adding a "dongle" attachment to a smartphone or by using some fixed ground station IOT device, or laptop to which a USB receiver could be connected.

X1.2.3.6 *Transmitter Hardware*—A wide variety of transmitter hardware exists to support BT5LR at commodity prices. Some modules even have an integrated power amplifier to extend the range. Such hardware modules have a programmable microprocessor with serial interfaces for getting telemetry data from the UA which is retransmitted through the connected antenna

X1.2.3.7 As with BT4, it does not commonly exist on UAs today, therefore it would have to either be retrofitted onto existing UAs or designed into newer products.

X1.2.4 *Wi-Fi Aware:*[14]

X1.2.4.1 *Industry Standard*—The "Wi-Fi Aware"[14,10] protocol used in this specification was developed by a workgroup of many industry stakeholders within the Wi-Fi Alliance.

X1.2.4.2 *Technology*—Wi-Fi Aware[14] is a technique that uses Wi-Fi radios to send broadcast messages using IEEE standard "public action frames" formatted in a standard way to transmit arbitrary data.

X1.2.4.3 *Range*—The standard prescribes higher power numbers for Wi-Fi than Bluetooth because of the higher transmit speed (6 Mbps) and the wide availability of radios that support this higher power level. In rural environments, tests are showing over 1 km (at 14 dbm), and for typical power levels in US and EU the expected ranges are more than 2 km at 20 dBm in EU, and more than 4 km at 26 dBm in US. Lots of RF interference could reduce this range.

X1.2.4.4 *Large Payload*—By having a large (255 byte) payload, all messages can be sent as a grouped set that allows more authentication options (similar to BT5LR).

X1.2.4.5 *Receiver Support*—Wi-Fi Aware[14] is a bit new in terms of direct operating system and smartphone support. Android 8 (Oreo) released in 2017 is the first smartphone OS to support Wi-Fi Aware.[14] Almost all mainstream mobile Wi-Fi chipset manufacturers have supported Wi-Fi Aware.[14] Mobile phone manufacturers could support Wi-Fi Aware[14] as long as they complete the necessary software adaptation. As of this writing, newer smartphones by Google (Pixel 2,3), Xiaomi (Mi 8, Mi 9, Mi x3), and most recently, Samsung (Galaxy Note 10) all support Wi-Fi Aware[14] according to published specifications. iOS does not yet support Wi-Fi Aware.[14]

X1.2.4.6 *Transmitter Hardware*—Since many modern consumer UAS include Wi-Fi radios, most of them will be able to enable this with simply applying a firmware update and retrofitting hardware will not be necessary.

X1.3 Network Remote ID Characteristics

X1.3.1 *Industry Standard*—The protocols created in this specification are all built upon modern RESTful web services protocols that have been tried and tested for many years.

X1.3.2 *Technology*—Since the protocols are written into the standard to be above "layer 4," any kind of "layer 2,3" network media (Cellular, Private Network, Wi-Fi, Satellite) that meets the performance criteria could potentially be used. Most of the group sees cellular networks as the early enablers to provide this VLL (low altitude) network infrastructure.

X1.3.3 *Infrastructure*—In addition to a network infrastructure, a server/service infrastructure needs to be turned on to provide the required services of Network Remote ID. Much of this specification describes the requirements of such services. These services will all leverage the Internet as the common carrier between federation members. As such, in the event of dependent server or network outages, Network Remote ID could experience outages as well.

X1.3.4 *Range*—The only range limitation that may exist for Network Remote ID is the UAS access to the network. As long as the UAS can reach a network (such as cell service), and as long as the intended receiver can reach the network, then the range from the transmitter to the receiver has no limitation.

X1.3.5 *Receiver Support*—Since nearly all smartphones enjoy the connectivity to the Internet, receiver support is broad and consistent across Android and iOS and universally across hardware platforms. Also, operators of conventional computer platforms (laptops, desktops, etc.) that are network connected would universally be able to participate in the network as well.

X1.3.6 *Transmitter Hardware*—A number of transmitter media and topologies could be used to connect to the network. For example, a UA could have a cell modem on the aircraft itself, or it could solely communicate with a ground station which has a built-in cell phone (as many do) which could provide this network connectivity.

X1.3.7 *Coverage*—Support areas for Network Remote ID is generally in areas with cellular network coverage (unless another network technology is used). Near highways and urban areas and generally where people are, cell network coverage is generally available. If there are areas of "spotty" network coverage, this specification specifies ways of interpolating position reports even if the UA occasionally ventures to the edge of network coverage.

X2. LIST OF SUBCOMMITTEE PARTICIPANTS AND CONTRIBUTORS

TABLE X2.1 List of Subcommittee Participants and Contributors

Name	Organization
Hanson, Richard	Academy of Model Aeronautics
Coleman, Sean	AirMap
Lamprecht, Andreas	AirMap
Voß, Thomas	AirMap
Lester, Edward (Ted)	AiRXOS, part of GE Aviation
Shestopalov, Andrea	AiRXOS, part of GE Aviation
Cassidy, Sean	Amazon
Champagne, Robert	Amazon
Roth, Robert	Amazon
Ganjoo, Amit	ANRA Technologies
Borda, Fred	Arial Innovation
Kenul, Philip (Chair, ASTM F38 UAS Committee)	ASTM International
Mikolajewski, Mary (ASTM F38 Staff Manager)	ASTM International
Daly, Brian	AT&T
Huffstutler, Tommy	AT&T
Musgrove, Charles	AT&T
Baum, Michael	Aviators Code Initiative
Caina, Javier	DJI
Schulman, Brendan	DJI
Bern, Evelina	FAA
Chen, Bin “David”	FAA
Ghimire, Ritesh	FAA
Hendrickson, Adam	FAA
Hinaman, Arthur	FAA
Liang, Diana	FAA
May, Rick	FAA
Saunders-Hodge, Sabrina	FAA
Segers, Robert	FAA
Curdy, Benoit	FOCA
Cox, Gabriel (Workgroup Lead, Broadcast Lead)	Intel
Davis, Mark Edward	Intel
Friis, Soren	Intel
Takei, Jun	Intel
Elliott, Ken	Jetcraft
Bender, Walter	JHU Applied Physics Laboratory
Silbermann, Joshua	JHU Applied Physics Laboratory
Olson, Gregory	KBR
Sehgal, Ajay (Vice Chair, ASTM F38 UAS Committee)	KBR
Gunnarson, Tom	Kitty Hawk Aviation & Aerospace
Elefant, Andrew	Kittyhawk.io
Jessen, Ian	MIT Lincoln Laboratory
Lacher, Andrew	MITRE
Maroney, David	MITRE
Murphy, David	MITRE
Thurling, Andrew	NUAIR
Pierce, Aaron	Pierce Aerospace
Sugahara, Kenji (WG Administrator)	Pierce Aerospace
Jennings, Richard	REJ Aviation Services
Brown, Simon	RelmaTech
Hall, Philip	RelmaTech
Deeds, Greg	Technology Exploration Group
Ruff, Nathan	UASidekick
Ramsey, Christian	uAvionix Corporation
Belaus, Greg	Uber
Prevot, Tom	Uber
Wei, Bogu	Uber
Fanelli, Matt	Verizon/Skyward
Blanks, Mark	Virginia Tech
Fox, Luke	WhiteFox Defense Technologies
Hodgens, Ryan	WhiteFox Defense Technologies
Peterson, Zachary, PhD	WhiteFox Defense Technologies
Glasgow, Mike (Network Lead)	Wing
Jackman, Chris	Wing
Negron, Reinaldo (Data Lead)	Wing
Pelletier, Benjamin	Wing
Teeling, Sean	Wing
Woodworth, Adam	Wing
Wurzbach, Jeffrey	Wurzbach Electronics, Inc.
Reddy, Hersh	Zipline International

X3. BACKGROUND INFORMATION

X3.1 In 2017, the Federal Aviation Administration (FAA) chartered the Unmanned Aircraft Systems (UAS) Identification (ID) and Tracking Aviation Rulemaking Committee (ARC) (UAS-ID ARC) to provide recommendations to the FAA regarding technologies available for remote identification and tracking of UAS. In its final report,[15] the ARC recommended two methods for remote identification, those being "broadcast" and "network." "Broadcast" would require UAS to transmit information without bi-directional communication with a receiver. "Network" would require UAS to communicate information to a network, whether space or terrestrial.

X3.2 In 2018, the ASTM F38 Committee on Unmanned Aircraft System spearheaded the effort to define minimum performance standards for remote identification of UAS. Working in conjunction with the FAA, members of the ASTM Subcommittee F38.02 on Flight Operations began work on creating the standards for Remote ID and Tracking. During the summer of 2018, the group completed the Terms of Reference (TOR) that would guide development of the standard. At the suggestion of the FAA, the Subcommittee created use cases that aligned with the recommendations from the ARC. The FAA also gave guidance that both methods of Remote ID and tracking would need to be compatible with commonly carried handheld devices such as cellphones.

X3.3 Use Cases

X3.3.1 The use cases fell into two broad categories. The first addressed security and public safety needs of the law enforcement, homeland defense, and national security communities for the remote identification and tracking of UAS. The second category addressed public access to remote identification information as a secondary objective.

X3.3.2 The primary high-level "representative" and unofficial use cases considered including the following and were inspired by the UAS-ID ARC and discussions within the workgroup.

X3.3.3 *Public Safety & Public Concerns*—The UA unique identifier should be available to the public and the public should be able to read this identifier during flight so that they could request assistance from public safety officials.

X3.3.4 *Area Threat Assessment:*

X3.3.4.1 Public safety officials are responding to a citizen report of a "suspicious UA" that is currently operating.

X3.3.4.2 Public safety officials are providing security to critical infrastructure or venue and spot a "suspicious UA." They need to identify the operator, make a threat determination, and respond accordingly.

X3.3.5 This category addresses the need to have dynamic and active awareness of UAS near heightened awareness areas. These areas could include: airports, heliports, prisons, military installations, nuclear facilities, large stadiums, and other critical infrastructure locations where a UAS could potentially pose an imminent threat to public safety.

X3.3.6 Of course, many other derivative use cases could be expanded from this list, such as the following:

X3.3.6.1 Public safety official receives a complaint about a UAS that may or may not be flying out of compliance and the official seeks to identify the UAS, Remote Pilot, contact information, and pilot location using a standard handheld device.

X3.3.6.2 Someone in the general public sees a UAS operating in a way that causes them concern, but eventually departs the area. They would like to report the UAS (ID) to a public safety official (using a smartphone).

X3.3.6.3 Someone in the general public sees a UAS fly over their area and wants to identify the UAS.

X3.3.6.4 A private industrial company (perhaps a refinery or some other location considered "critical infrastructure") would like to patrol the security of the area including the low altitude airspace. They would like to deploy receivers to alert (private) security personnel of incursions over their sensitive location and inform public officials to address the problem.

X3.3.6.5 Airport security would like to detect unauthorized incursions into their low altitude airspace using a network of fixed base receivers.

X3.3.6.6 There is a VIP event in an area and security personnel use either a mobile application or fixed based receivers to identify UAS in that area.

X3.3.6.7 UAS operating under UTM would like to detect other UAS nearby whether they are operating under: (*1*) the same USS, (*2*) another USS, or (*3*) outside of UTM.

X3.3.6.8 A UAS operating under BVLOS would like to avoid a path conflict with another UAS.

X3.3.7 All of these inputs, use cases, reviews, and feedback went into consideration in the development of this specification.

[15] FAA, Unmanned Aircraft Systems (UAS) Identification (ID) and Tracking Aviation Rulemaking Committee (ARC) (UAS-ID ARC) Recommendations Final Report, https://www.faa.gov/news/updates/?newsId=89404.