drip Working Group
Internet-Draft
Intended status: Standards Track
Expires: 5 April 2026

A. Wiethuechter
AX Enterprize, LLC
R. Moskowitz
HTT Consulting

2 October 2025

Registration & Usage of DRIP Entity Tags for Trustworthy Air Domain Awareness

draft-wiethuechter-drip-det-tada-01

Abstract

This document standardizes usage of Drone Remote Identification Protocol (DRIP) Entity Tags (DETs) as identifiers to enable trustworthy Air Domain Awareness (ADA) for Unmanned Aircraft Systems (UAS) in Remote Identification (RID), UAS Traffic Management (UTM) and Advanced Air Mobility (AAM). DETs can serve as Session IDs for privacy, Authentication Key IDs for accountability, and encryption key IDs for confidentiality.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 5 April 2026.

Copyright Notice

Table of Contents

1. Introduction

1.1. Purpose

This document provides to stakeholders in a National Airspace System (NAS), such as a Civil Aviation Authority (CAA) and its constituents, a point of entry into a set of relevant standards. These standards provide guidance to enable trustworthy Air Domain Awareness (ADA) primarily via cooperative technologies. Such technologies can include, but are not limited to, Unmanned Aircraft (UA) Systems (UAS) Remote Identification (RID), UAS Traffic Management (UTM) and Advanced Air Mobility (AAM).

This document specifies an interoperable manner of achieving the stated objectives in an international context to be used by CAAs in a relevant Means of Compliance (MOC). One such MOC is ASTM [F3586] for UAS RID to comply with the United States (US) CAA regulations.

A CAA MAY override any technical specification in this document but MUST, if claiming compliance with this document, provide the reason and an alternative specification satisfying the same objective.

1.2. Background

[RFC9153] provides comprehensive background on the UAS RID use-case and the UTM/AAM context.

Trustworthiness revolves around identification and authentication. [RFC9374] defines the Drone Remote Identification Protocol (DRIP) Entity Tag (DET), a trustable identifier backed by asymmetric cryptography. [RFC9575] standardizes authentication of DETs and associated information using strong cryptography suited for constrained mobile wireless links (typical of Broadcast RID). Broadcast and Network RID are defined by ASTM [F3411] which designates the International Civil Aviation Organization (ICAO) to maintain code-points for Specific Session ID Types and Specific Authentication Methods (SAMs) [ICAO-NUMBERS]. ICAO includes DETs as a Specific Session ID Type and lists the four DRIP SAMs.

DETs generated by registrants (typically a UAS Operator or UAS) are merely proposed until they are registered within the hierarchy consisting of at least two levels above the leaf DET: Registered Assigning Authority (RAA) and Hierarchial Host Identity Tag (HHIT) Domain Authority (HDA). The registry system, made up of DRIP Identity Management Entities (DIMEs) acting as either RAAs or HDAs, ensures uniqueness within the hierarchy, endorses inclusion through X.509 certificates, and provides access to public and private information associated with a DET registration. Public information is served using the Domain Name System (DNS) and is specified in [DET-DNS]. Private information, such as Personally Identifiable Information (PII), is served using the Registration Data Access Protocol (RDAP, [STD95]) and protected through policy based differentiated access.

This document specifies fundamental registration processes and interactions surrounding the Private Information Registry function defined in the DRIP Architecture ([RFC9434]).

1.3. Scope

Participating entities are assumed in this document to be in compliance with relevant existing UAS standards such as ASTM [F3411], [F3548] and [ICAO-ACCP].

This document governs use of a DET in the following cases:

1. An entity that uses a Session ID (some participating entities, e.g. HDAs, may not need Session IDs), MUST use a DET for that Session ID.

2. An entity that participates in DRIP interactions (even if not needing a Session ID), MUST use a DET for the DRIP public authentication key ID in those interactions.

3. An entity that requires a strongly cryptographically verifiable IP compatible identifier, MAY use a DET for any other legal purpose.

4. An entity SHOULD NOT use a DET as a locator in physical or logical space (e.g. as a globally routable IPv6 address outside of an overlay network), as deconflation is intended, see [RFC9063].

The archetypal case is a UAS for which the DET serves as both the UAS ID (first case above) and the Authentication Key ID (second case).

Author Note: narrow context for at least first bullet above.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.1. Additional Definitions

This document makes use of terms (PII, USS, etc.) defined in [RFC9153], [RFC9434] and [RFC9374]. For convenience of the reader, some of the major definitions are restated here.

DRIP Identity Management Entity (DIME): Originally defined in [RFC9434] and expanded technically in [DET-DNS]. An entity providing registrar and registry services specifically for DETs. They can act as either an Registered Assigning Authority (RAA) or Hierarchial Host Identity Tag (HHIT) Domain Authority (HDA).

Web Token: In the context of this specification; a Web Token can be either a Concise Binary Object Representation (CBOR, [STD94]) Web Token (CWT, [RFC8392]) or JavaScript Object Notation (JSON, [STD90]) Web Token (JWT, [RFC7519]).

3. Interactions & Responsibilities

This section outlines the various entity roles in the ecosystem. For each, it summarizes motivations for participating in that DRIP role, as well as the interactions and responsibilities they have in doing so. All entity roles, except as otherwise noted, MUST register a DET with a parent entity in the corresponding parental role (e.g. an HDA must register with an RAA), as per Section 4.2, and MUST use it subsequently in all DRIP interactions requiring an entity ID.

3.1. UAS Observers

An Observer is typically an individual who has a device capable of wirelessly receiving Broadcast RID whether it be an opportunistic passerby with a local-only (i.e. non-Internet connected) smartphone application or a widely deployed sensing system with various IP-based back-hauls. Each has their own motivations for detecting a UAS in their area and querying for additional information. Typically in DRIP, Observers are expected to use DNS (when able) to obtain at least the public keys of unknown DETs and use it to validate signatures found in ASTM [F3411] Authentication Messages. Some Observers may be granted, by cognizant authorities, authorization to perform more detailed queries of further information, that is considered private (as not explicitly classified as public, placed in DNS, and/or transmitted in cleartext in Broadcast RID).

> Author Note: Broadcast RID here can be dropped or substituted for some phrase comparing Observers to a sensor head, but distinct from a Display-only Observer. If that makes sense and if it is a good distinction to have? One could say Display Applications are in a way Observers (albeit very far removed from the field in most cases where the term is used). Unless we make them their own class of users that could use DETs (and for what?)

All Observers supporting DRIP:

* MUST adhere to relevant details of Section 6.4.2 of [RFC9575] with regards to receive processing of DRIP SAMs.
  - Processing of SAMs MAY be deferred or outsourced to a dedicated system.
* RECOMMEND display all DETs following [RFC5952] or as described in Appendix B.

Author Note: as justification for "processing MAY be deferred": Intention here was to allow for Observers not needing to fully process data if desired and push said processing to SDSPs. An example is a slim Finder concept that does no parsing of RID data, just extraction out of transport framing.

Observers authorized by cognizant authorities to access any private (in addition to all public) information recorded as part of the registration process also:

* MUST support RDAP authentication/authorization mechanisms allowed by the cognizant authority.
  - Those RECOMMENDED for global harmonization are identified in Section 4.3.2.4.
* MUST register to authentication systems as required by the cognizant authority.
  - A DIME MAY be used as part of an authentication system.

3.2. UAS Operators

UAS Operators is used herein as a catch-all term for several more specific UAS related roles: Remote Pilot, Pilot In Command (PIC), party (typically an organization) to whom or which the UAS is registered with the cognizant CAA, etc. DRIP does not attempt to clarify the definition of Operator to match any specific CAA, instead using the ambiguity to group these roles based on their similar DRIP interactions.

A UAS Operator’s desire to use DETs can be any number of the following operational factors:

* Confidentiality of flights through Session IDs.
* Privacy of Remote Pilots by encrypting their Ground Control Station (GCS) position.
  - This option may be constrained by their jurisdiction’s RID regulations.
* Reputation of their UAS based on observation and reports of behavior supported by Authentication.

To participate a UAS Operator SHOULD use a DET, registered either with an RAA or HDA, as part of any interactions in DRIP, such as registering a UAS Session ID for their UA. A UAS Operator MAY use

another cryptographic scheme for interactions but it MUST be asymmetric with their public key accessible to all domain participants via a standardized method for the cryptographic purpose of signature verification.

3.3. UAS Manufacturers

While UAS Manufactures do not have a direct requirement for DRIP, they typically wish to maintain good reputation of their brands and are often driven by market forces. Such forces include requirements and preferences of both UAS Observers and Operators.

This document assumes a UAS Manufacturer is already in compliance with ASTM [F3411]. Thus those that wish to support DRIP also:

* MUST provide a mechanism, typically through a GCS, to generate and register a DET for use as either a Session ID, Authentication Key ID or both.
  - MUST bind DET to both UAS Serial Number and Operator. Binding MAY be indirect if system can verify Serial Number already registered to Operator.
  - SHOULD be issued in a manner that keeps private key ONLY ever on UA, does not expose it to GCS etc., but MAY be done in manner that exposes it to GCS or Operator if necessary.
  - SHOULD allow user selection of a parent DIME (typically HDA) using a URI, Hierarchy ID (HID) value, or both
  - MAY support registering multiple DETs in a batch transaction
* SHOULD provide a mechanism to enable, during registration, subsequent encryption of selected fields (e.g. pilot/GCS location), only if the CAA allows it.
* MUST follow Section 6 of [RFC9575] as it relates to transmission of SAMs, when using DET as an Authentication Key ID
* MUST, in the [F3411] Basic ID Message, set the UAS ID Type to Specific Session ID, Specific Session ID Type to DRIP Entity ID, and encode the DET in network byte order, when using DET as a UAS Session ID
* MAY use a DET, following Section 4.2 of [RFC9374], as a UAS Serial Number

- It is expected that the UAS Manufacturer runs their own HDA, under which these DETs are registered

3.4. HDA Operators

In the UTM context, it is expected that the HDA function will be provided by a UAS Service Supplier (USS) to its clients (e.g. UAS Operators). This can be in-house or out-sourced to a service bureau more specialized in cryptographically verifiable identifiers usable to access data and systems on networks. Outside the UTM context, an HDA can be a stand-alone provider to any registrant desiring a DET, or more generally HHIT.

Being associated with a USS has some operational benefits:

* Ease for UAS Operators to use both Session IDs and UTM together

* Binding between the DET and Operational Intent

  - Enables safer use of encryption

  - Ability to keep DET private until just before an operation goes active

An HDA Operator:

* MUST register, with its parent RAA, one or more DETs, and use it or them in DRIP interactions

* MUST implement interfaces and functions described in Section 4

* MUST maintain DET registrations with relevant PII as required by their jurisdiction

* MUST fulfill the requirements of their jurisdiction

* SHOULD NOT provide to clients the ability to enable encryption of data elements considered PII, especially in Broadcast RID

As part of a USS an HDA Operator also:

* MAY provide encryption ID services, if allowed by jurisdiction

* MAY have its registration interfaces (Section 4.2) be integrated into relevant USS interfaces and functions

* SHOULD defer the publishing of a client DET in DNS when it is bound with an Operational Intent until it is "active"

- Methods to sync an Operational Intent in a USS state and DET state in HDA are out of scope for this document

3.5. RAA Operators

In the context of this document, RAA Operators are typically CAAs or other cognizant authorities of a Nation State. Each Nation State is allocated four RAA values that are used at their discretion. To obtain these allocations, a Nation State contacts the Apex operator for the UAS or other applicable hierarchy, currently IANA on behalf of ICAO, requesting their provisioning providing any details required by the Apex Operator to configure domain delegations.

Each RAA has the responsibility to delegate HDAs under them. Allocation methods of HDA values are at the discretion of the RAA Operator and their policies. Any requirements to operate under an RAA, beyond those as defined in this document to operate as an HDA, are out of scope for this document.

An RAA Operator:

* MUST obtain an RAA value(s) from the Apex Operator and provide content for NS RRType(s), see Section 5

* MUST register, with its parent (if any), one or more DETs, and use it or them in DRIP interactions

* MUST implement interfaces and functions described in Section 4

* MUST maintain DET registrations with relevant PII as required by their jurisdiction

* MUST provide requirements for prospective HDA Operators, and SHOULD do so publicly via an easily located web site

* MUST ensure HDA Operator compliance with this specification

  Author Note: subsection or appendix on a potential RAA scheme for CAA? i.e. 1x MIL, 1x GOV, 1x COMMERCIAL, 1x SPARE with some example guidance on HDA allocation?

3.6. Apex Operators

The Apex role is the administration root and the domain apex for a specific IPv6 prefix which are typically associated with an industry sector (i.e. DETs being for aviation). This translates to the technical tasking of delegating RAA values and maintaining the reverse DNS zone associated with the allocated IPv6 prefix which contain the DNS delegations to RAAs. An Apex in its capacity as the administration root MAY endorse the required RAA delegation in the hierarchy.

Currently IANA performs the functions of RAA delegation (with no endorsement) and DNS zone maintenance for the DET prefix. This is largely a custodial task as critical portions of the RAA range, with regards to aviation, are already reserved or allocated with the rest of the range unassigned.

In the context of aviation pre-allocated RAAs to CAAs are currently expected to mutually cross-certify as specified in [ICAO-ACCP].

4. Requirements for DIME Operation

The requirements below apply to both RAA and HDA Operators with a focus on how HDA Operators provide registration and related services for their clients (i.e. UAS Operators and their UAS). All data models are represented in this document using the Concise Data Definition Language (CDDL, [RFC8610]) using the prelude defined in Appendix E of [RFC8610] as well as an additional DRIP-specific prelude defined in Appendix A.

4.1. Common UAS RID Data Model

A common data model is specified by DRIP for various UAS RID elements typically transmitted over ASTM [F3411].

The field names and their general typing of Figure 1 are borrowed from the ASTM [F3411] data dictionary (Table 1 and Table 2). See that document for additional context and background information on aviation application-specific interpretation of the field semantics. The contents of this model are considered opaque to DRIP.

The model is an expansion upon the BRID RRType model, which only contained static data, defined in Section 5.2 of [DET-DNS].

```
   uas-datum = {
     ? timestamp => utc,
     ? uas_type => 0..15,
     ? uas_ids => [
       + [
         id_type: 0..15,
         uas_id: uas-id
       ]
     ],
     ? ua_status => 0..15,
     ? ua_position => [
       lla: position,
       barometric_altitude: number / null
     ],
     ? ua_bearing => uint,
     ? ua_speed => [
       vertical: number / null,
       horizontal: number / null
     ],
     ? ua_height => [
       agl: bool,
       height: number
     ],
     ? accuracy => [
       vertical: 0..15,
       horizontal: 0..15,
       altitude: 0..15,
       barometric: 0..15,
       timestamp: 0..15
     ],
     ? auth => [
       + [
         auth_type: 0..15,
         data: auth-data
       ]
     ],
     ? self_id => [
       desc_type: 0..255,
       desc: tstr .size 23
     ],
     ? operator_position => position,
     ? classification => [
       region: 0..8,
       category: 0..15,
       class: 0..15
     ],
     ? area => [
       count: 1..255,
```

```
      radius: number,
      floor: number,
      ceiling: number
    ],
    ? operator_id => [
      operator_type: 0..255,
      operator_id: tstr .size 20
    ],
    ? compliance => [+ uint]
  }
```

Figure 1: Common UAS RID CDDL

The mapping between JSON keys (as strings) and CBOR keys (as unsigned integers) for Figure 1 are defined in Section 5.5.

Author Note: do we specify the enumeration for compliance in this document? Is it an IANA registry?

4.2. Registration

This section defines the interaction model, data models and methods for registration of a DET into a DIME for global interoperability. Standardization of additional methods and data models to register to a DIME (DETs or otherwise) is out of scope for this document.

4.2.1. Disclaimer

DRIP does NOT provide protection against incorrect (e.g. fraudulent) data entered during registration or asserted subsequently. DRIP does protect against alteration (intentional or unintentional) of data subsequent to its assertion by the cryptographic signer. The signer might be the proximate sender (e.g. UA transmitting Broadcast RID) or might be an attestor far away and long ago (e.g. root Certificate Authority). It is the duty of the operator of each DIME, or the party on whose behalf the DIME is being operated, to validate the registration data. The RAA (e.g. CAA) SHOULD provide services to obtain this goal, see Section 4.2.4.1.

4.2.2. Interaction Model

Registration of DETs uses the interaction model shown in Figure 2.

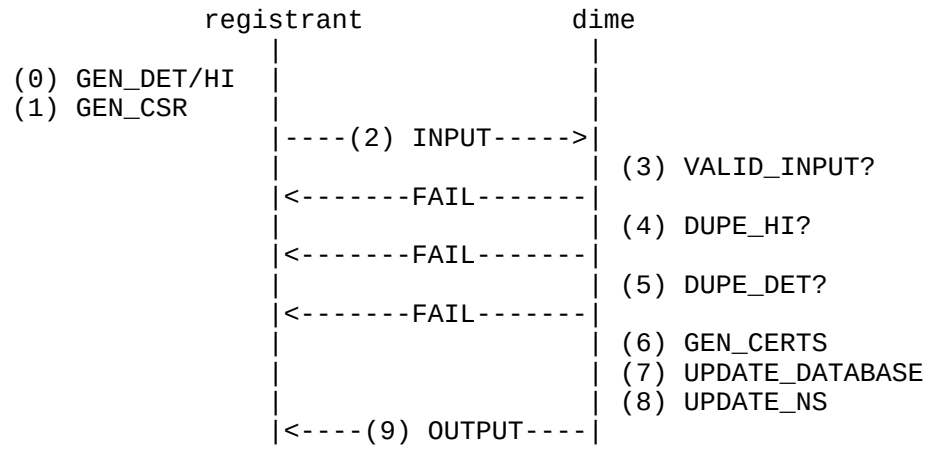

Figure 2: Simplified DET Registration Z-Diagram

(0) GEN_DET/HI: The registrant that plans to use the identity and its cryptographic properties SHOULD generate the asymmetric key-pair of their choosing and MUST protect the private key with Best Current Practices. The registrant MAY derive the corresponding DET. How the registrant obtains the desired HID for DET generation is out of scope for this document.

(1) GEN_CSR: The registrant MUST generate a Certificate Signing Request (CSR) for their HI/DET. See Appendix D.1 for more details.

(2) INPUT: The registrant MUST format and send the required information for a given service registration as specified by the DIME using Section 4.2.3.

(3) VALID_INPUT?: The DIME, upon receipt of service registration inputs, MUST validate the input data. This may be simply performing syntax checks to ensure the data is properly formatted all the way to full semantic validation (see Section 4.2.4.1). Signed data MUST have their signatures verified before further processing. Failure of input validation SHOULD returns errors to the registrant.

(4) DUPE_HI?: The DIME MUST check that the proposed HI is not a already in use within the specified HID scope. The registrant SHOULD be notified with an error if duplication is detected.

(5) DUPE_DET?: The DIME MUST check that the proposed DET, derived

from the HI, is not a already in use within the specified HID scope. If the registrant proposed a DET with HID outside that of the DIME, it MUST be rejected. If the registrant only provided the HI, the DIME MUST generate the DET using the HID for which that DIME is authoritative. The registrant SHOULD be notified with an error if duplication is detected.

(6) GEN_CERTS: The DIME, after all the above validation, MUST generate a Broadcast Endorsement using the provided HI and DET from the registrant as the evidence. This Endorsement is used in [RFC9575]. The DIME MUST also generate a corresponding X.509 certificate as per Section 4.2.5.1, confirming that the DIME has accepted the registrant's DET/HI pair for registration. Other Certificates MAY be generated during registration but are out of scope for this document.

(7) UPDATE_DATABASE: The DIME MUST update the Private Information Registry with all registration data required by the jurisdiction or the DIME's own policy, typically including PII, all of which must be protected by AAA as per applicable regulation and policy. It MAY include additional metadata or public information. How the Private Information Registry is updated is out of scope for this document.

(8) UPDATE_NS: The DIME MUST update the Authoritative Name Server with any public information that was part of the registration. This information MUST be stored as described in [DET-DNS].

(9) OUTPUT: The DIME, upon successful updates of the Private Information Registry and Authoritative Name Server, responds according to Section 4.2.3.

4.2.3. Interface & Encoding

Registration request and response data MUST be encapsulated in a Web Token (either CWT [RFC8392] or JWT [RFC7519]). For Session IDs and Authentication Key IDs the registrant usually is the Operator (or a proxy for the Operator such as the GCS) but MAY be the UA directly if the UA has a long term cryptographic identity.

The method for a registrant to find a DIME is out of scope for this document. DIMEs MUST use [RFC8615] to provide DRIP-related interfaces and perform redirects to the relevant URIs. DIME registration endpoints MUST use POST expecting a Web Token from registrants. During a request a registrant MUST encrypt the Web Token in accordance with PRIV-2 of Section 4.3.1 of [RFC9153] as during a request it typically includes PII. An example is shown in Figure 3.

Author Note: /.well-known/ could be a single URI like register-hhit or more specific ones like uas-operator and uas-session. Do we have a preference?

```
POST /.well-known/register-hhit HTTP/1.1
Host: uss.example.com
Accept: application/drip+cwt, application/drip+jwt
Content-Type: application/drip+cwt
Content-Length: <CWT length>

[ CWT from registrant ]
```

Figure 3: Example REST Registration API (request)

The DIME responds, upon successful registration, with their own Web Token contain the data model defined in Section 4.2.5. The Web Token SHOULD be encrypted, if containing any PII. An example is shown in Figure 4.

```
HTTP/1.1 200 OK
Accept: application/drip+cwt, application/drip+jwt
Content-Type: application/drip+cwt

[ CWT from DIME ]
```

Figure 4: Example REST Registration API (response)

This document allocates application/drip+cwt, application/drip+jwt and application/drip to the Media Types registry. It also allocates register-hhit to the /.well-known/ registry.

4.2.3.1. Using CoAP for Registration

Support of the Constrained Application Protocol (CoAP, [RFC7252]) is OPTIONAL.

When using CoAP, DTLS SHOULD be used when possible and MUST send the data models (Section 4.2.4 and Section 4.2.5) encoded as CBOR. When DTLS is not possible, and CoAP is instead used with UDP, model data MUST be encapsulated, signed and encrypted in a CWT as above.

4.2.3.2. Guidance on Registration Errors

The following registration failure conditions and response are informative:

* Upon invalid input from the registrant, the DIME SHOULD respond with either 400 (Bad Request) or 403 (Forbidden) as appropriate.

* Upon a duplicate HI or DET, the DIME SHOULD respond with 409 (Conflict).

* Any other processing failure of registration by a DIME SHOULD be responded with an appropriate Server Error status code.

Care should be given when responding with 403 (Forbidden) to not expose any information that could be used to socially engineer a fake request. For example, when following Section 4.2.4.1, it would be unwise for the HDA to detail an exact reason for the rejection. A malicious registrant could spam the endpoint, farming information on what UAS Serial Numbers, Operator IDs and their potential bindings exist to exploit.

It should be noted that the above would be avoided with properly configured systems since signature validation and denial of service mitigations would be before the HDA even attempts to ascertain validation from the RAA.

4.2.4. Registration Model

The data sent to the DIME for a given registration is policy driven. For example a Session ID registration SHOULD include a CAA specific identifier of the Operator to enable the private binding of the Session ID to both the Serial Number (found in the CSR, Appendix D.1) but also the Operator themselves.

```
registration-content = {
  csr_list: [
    + csr: [
      idx: hash,
      entity_type: uint,
      csr: x509,
      vnb: utc / null,
      vna: utc / 1..15638400 / null,
      meta: { metadata }
    ]
  ],
  metadata
}
```

Figure 5: Registration Model CDDL

The data model defined in Figure 5 is what is minimally required for a DIME to be functional. Additional fields SHOULD be registered to IANA under Section 5.5 for global harmonization. RAAs and HDAs MAY use their respective keys and their associated maps to define additional keys to be included in a registration. A dedicated Private Use space is provided for RAAs and HDAs to define keys and their uses.

The uas map item uses the data model found in Section 4.1 to carry any static information expected to also be found over Broadcast RID. It MUST NOT include in the uas_ids key the UAS Serial Number when a Session ID is being registered. It also MUST NOT contain the operator_location key, as this information is usually considered PII and typically is dynamic or unknown at time of registration. The DIME MUST fill in the auth key with any SAMs related to a registration and place the data model into DNS with Section 5.2 of [DET-DNS] when the registrant entity is required to use Broadcast RID and expecting to use [RFC9575].

The registrant can select from a number of options to encode the values of valid not before (vnb) and valid not after (vna) for time of applicability of an individual CSR (csr-data). The use of the time or tdate types allow for absolute time references. The use of the uint type for vna specifies a positive offset in seconds from an absolute time. When null is used the registrant is informing the DIME that for vnb the DIME MUST use the current time as vnb during registration and for vna the DIME MUST set vna to a time such that the default DIME policy for maximum delta between vnb and vna is not exceeded.

> Implementation Note: due to not differentiating between integers and floats in its number type, implementations using JSON as the encoded format MUST check the value of vna to determine if its being used as an offset or absolute time.

As an example of the use of vnb and vna, a registrant may set vnb=null and vna=3600. The DIME in this instance would use the current absolute time at receipt of the registration request for vnb and then apply the offset specified by vna to obtain an absolute time for vna that is 3600 seconds after vnb. These absolute times of vnb and vna are then used in the certificate being created by the DIME to endorse the registration Section 4.2.5.1.

4.2.4.1. Validating Fields

In certain circumstances field contents may only be properly validated by other entities outside of DRIP. For example the binding between UAS Serial Number and Operator ID should already be known by the CAA, as typically this information is required out of band of DRIP directly to the CAA in some form. The CAA should provide or itself have access to an "oracle" that can validate these claimed bindings for registration to proceed. If such an "oracle" is not consulted there is a risk that information being registered is fraudulent and DRIP has no method or authority to confirm the claims. If such information is accepted, future information queries by authorities can result in bogus data being returned, with no binding to an actual UAS or Operator.

The CAA SHOULD, as part of its capacity as an RAA onboarding an HDA, provide data models, communication framework and any authentication mechanisms to query the oracle directly if the CAA allows HDAs to perform the queries themselves. Otherwise the RAA MUST provide an HTTPS POST interface that consumes a Web Token generated by the HDA containing the registrant Web Token. If the RAA is not configured or does not wish to handle such requests it MUST respond with 501 (Not Implemented), HDAs should proceed without validation as if validation was successful.

If the RAA is configured to validate registration requests, it sends the data to the "oracle", via a mechanism out of scope for this document. When the "oracle" marks data as invalid the RAA MUST signal to the HDA that the registration is to be denied. Otherwise the RAA returns a Web Token to the HDA containing an X.509 Certificate with the Subject set to the same contents as the registrant CSR.

4.2.5. Response Model

```
response-content = [
  + [
    idx: hash,
    entity_type: uint,
    certs: {
      canonical_cert: x509,
      ? be_chain: [+ be: endorsement],
      * &(tstr , int) => any
    }
  ]
]
```

Figure 6: Response Model CDDL

The canonical_cert key MUST be the certificate defined in Section 4.2.5.1. The be_chain contains the Broadcast Endorsement structures defined in Section 4.1 of [RFC9575] and MAY not be sent by the DIME if the registrant is not expected to use Broadcast RID.

Additional items MAY be sent back to the registrant by the DIME. Their keys SHOULD be registered under IANA as part of Section 5.5 to support global harmonization.

4.2.5.1. Canonical Registration Certificate

Appendix D provides profiles for X.509 certificates adhering to the [ICAO-ACCP], developed by the Trust Framework Panel. At least one of the DRIP profiles MUST be used.

At least one certificate in the chain from an apex node to a leaf node MUST contain a URI, as part of the Subject Alternative Name Extension [RFC5280], which points to the relevant Private Information Registry containing associated PII registered. This certificate MUST be contained in the HHIT RRType defined in Section 5.1 of [DET-DNS]. The certificate MAY be anywhere on the path and SHOULD be as close to the apex in the path as possible for efficiency.

At the time of publication, the current best practice is for the HDA Issue certificate to contain the base URI to be used for all Private Information Registry requests under the HDA. When an HDA supports multiple Private Information Registry providers, it SHOULD set the appropriate URI in the Operational certificate. The HDA MAY either set all different URIs together in its HDA Issue certificate at cost of the querent being then required to perform queries at each provided URI until it receives a response or use different Issuer DETs for each different Private Information Registry provider.

When registering a DET to be used as a Session ID, the CSR Appendix D.1 MUST contain the UAS Serial Number as the Subject, and the certificate placed in the private registry also MUST contain the UAS Serial Number as the Subject, but the certificate placed in the public registry and used as the basis of the Broadcast Endorsement (or otherwise exposed publicly) MUST NOT contain the UAS Serial Number.

4.3. Query

A DIME has two query interfaces it MUST support. One interface is used for public information and the other (discoverable from public information) is for private information.

4.3.1. Public Information Registries

The information found in these registries has been designated (explicitly or implicitly) as public by cognizant authority and/or the information owner. Such information includes asymmetric cryptographic public keys needed for authentication in [RFC9575], static Broadcast RID data and trustworthy pointers to additional information.

These registries are Authoritative Name Servers in the DNS. See [DET-DNS] for operational requirements, query mechanism and response models.

4.3.2. Private Information Registries

The information found in these registries is considered Personally Identifiable Information (PII) and/or is stored for potential future audit of registration. Access to Private Information Registries MUST be performed using RDAP [STD95]. This section defines RDAP behavior for DRIP.

4.3.2.1. URI Path Segment

The URIs to Private Information resources MUST use the specification defined in Section 3.1.1 of [RFC9082]. Use of Section 3.1.3 of [RFC9082] is OPTIONAL.

4.3.2.2. Conformance Literal

The string literal drip_version_0 MUST be used in the rdapConformance section of the RDAP response to signal conformance with this specification.

4.3.2.3. Extension Model

Queries of DETs in RDAP SHOULD respond with the appropriate Object Class as defined in RFC9083 and MUST include the additional data element specified in Figure 7.

```
dime = {
  csr: x509,
  canonical_cert: x509,
  hhit: [
    entity_type: uint,
    ipv6: ip6
  ],
  be_chain: [+ be: endorsement]
  registrant: tstr,
  ? raa_approval: x509,
  metadata
}
```

Figure 7: RDAP Response CDDL

Additional keys SHOULD be registered to IANA under Section 5.5 for global harmonization. Specification of their usage and syntax in a NAS is provided by CAAs using a MOC that references this specification.

4.3.2.4. Differential Access

Per REG-2 and REG-4 of Section 4.4.1 of [RFC9153], RDAP queries to DRIP Private Information Registries MUST be protected using fine-grained AAA policies in a both human- & machine-readable form for automated enforcement. RDAP supports only HTTP based mechanisms for authentication as defined in Section 3.2 of [RFC7481]. A federated authentication mechanism, such as the examples in Section 3.2.1 of [RFC7481], is RECOMMENDED.

For international and/or global harmonization, DRIP standardizes the following RDAP behavior for authentication of clients and servers:

* MUST support HTTP Digest Authentication Scheme ([RFC7616]) *AND*

* SHOULD NOT support HTTP Basic Authentication Scheme ([RFC7617]) but MAY accept Basic if peer offers that and nothing stronger *AND*

* MUST support Mutual TLS Section 7.4.6 of [RFC5246] for global and/or international queries *AND*

* SHOULD use Mutual TLS Section 7.4.6 of [RFC5246] but MAY do any other RDAP compatible AAA for domestic queries

A DIME when supporting Mutual TLS MUST accept valid DKIX certificates (Appendix D) and MAY accept other certificates.

Author Note: the selection of Mutual TLS is an initial down selection from OpenID Connect for RDAP ([RFC9560]), or Mutual TLS (Section 7.4.6 of [RFC5246]) or eXtensible Access Control Markup Language (XACML) with Security Assertion Markup Language (SAML). This is still a discussion point.

5. IANA Considerations

5.1. RAA Delegation for CAAs

As part of [DET-DNS] IANA already has the pre-allocated mapping of RAA values for CAAs. Nations are to follow the guidance in Section 6.2.1.4 of [DET-DNS] to request delegation of their DNS zone under 3.0.0.1.0.0.2.ip6.arpa..

5.2. Well-Known URIs

IANA is requested to add the following entries in the "Well-Known URIs" registry [WELL-KNOWN].

URI Suffix	Change Controller	Reference	Status	Related Information
register-hhit	IETF	This RFC	permanent	N/A

Table 1: Additions to Well-Known URIs Registry

5.3. RDAP Extensions Registry

IANA is request to register the following value in the "RDAP Extensions" registry [RDAP-EXT].

Extension Identifier:
drip_version_0
Registry Operator:
Any
Specification:
This specification
Contact:
IETF <iesg@ietf.org>
Intended Usage:
This extension is used to convey private information under an HHIT registration.

5.4. Media Types

IANA is requested to add the following entries in the "Media Types" registry [MEDIA-TYPES].

Name	Template	Reference
DRIP CWT	application/drip+cwt	Section 5.4.1
DRIP JWT	application/drip+jwt	Section 5.4.2

Table 2: Additions to Media Types Registry

5.4.1. application/drip+cwt Registration

Type name: application

Subtype name: drip+cwt

Required parameters: N/A

Optional parameters: N/A

Encoding considerations: binary

Security considerations: TODO

Interoperability considerations: N/A

Published specification: This RFC

Applications that use this media type: Applications that access DIMEs requesting HHIT registration, as well as DIMEs responding to registration requests.

Fragment identifier considerations: N/A

Person & email address to contact for further information: DRIP WG mailing list (drip@ietf.org)

Intended usage: COMMON

Restrictions on usage: none

Author/Change controller: IETF

Provisional registration: no

5.4.2. application/drip+jwt Registration

Type name: application

Subtype name: drip+jwt

Required parameters: N/A

Optional parameters: N/A

Encoding considerations: binary, JWT values are encoded as a series of base64url-encoded values (with trailing '=' characters removed), some of which may be the empty string, separated by period ('.') characters.

Security considerations: TODO

Interoperability considerations: N/A

Published specification: This RFC

Applications that use this media type: Applications that access DIMEs requesting HHIT registration, as well as DIMEs responding to registration requests.

Fragment identifier considerations: N/A

Person & email address to contact for further information: DRIP WG mailing list (drip@ietf.org)

Intended usage: COMMON

Restrictions on usage: none

Author/Change controller: IETF

Provisional registration: no

5.5. DRIP CBOR/JSON Keys

This specification establishes a new sub-registry called "DRIP CBOR/JSON Keys" within the "Drone Remote ID Protocol" registry. This registry is to maintain a globally harmonized list of JSON keys (i.e. "Names") and CBOR map integer keys (i.e. "Labels") for use in DRIP. The following ranges are defined based on the CBOR Label values:

Range	Registration Policy	Range Use	Comments
from -2^64 to -25	Private Use (Section 4.1 of [RFC8126])	HID Private Keys	Private use for RAAs and HDAs
from -24 to 23	TODO: TBD	ASTM UAS Keys	Keys found in ASTM UAS Standards such as [F3411]
from 24 to 2^64-1	Specification Required (Section 4.6 of [RFC8126])	Global Harmonization	Reserved for future registrations to maintain global harmonization

Table 3: DRIP CBOR/JSON Key Registry Policies

Keys in the Private Use and Experimental ranges are RECOMMENDED to prepend a _ to the Name to avoid namespace collisions with registered Names.

The HID Private Keys range in this registry provide code-points (used in Section 4.2.4 and Section 4.3.2.3 as part of metadata) for each RAA and HDA to define their own subset of elements in both registration and RDAP responses. The spec value of this registry SHOULD be used to enable the DIME or client to provide a "hint" for the data elements included. Specification of the spec string semantics are left to individual RAAs and HDAs to define as they see fit. RAA and HDA Operators SHOULD provide specification for any Private Use or Experimental Use values, but this may not be public due to policy.

5.5.1. Review Criteria

For registrations, review should note if the Name field matches an existing entry, the request should be denied. A request should also be denied if the specification duplicates an existing entry both in syntax and semantics. For example, if an existing key called whee already exists with the specification of [+ uint] and a new request comes along to register whee_but_mine with specification [+ uint], then the latter should be rejected on the grounds of being able to use whee instead. This denial can be overruled if shown that the new registration has clear justification, as part of the specification citing technical reasoning, to exist.

5.5.2. CBOR/JSON Key Fields

Name (JSON Key Value): A string value, to be used as a key in the key: value pair of a JSON object. No specific rules apply for syntax beyond being a valid JSON string for a object key, but it is recommended to use all lower-case and snake-case with underscores. Care should be given to the length of a Name to reduce wire size of JSON encodings for constrained environments.

Label (CBOR Integer Key Value): A integer value, ranging from -2^64 to 2^64 - 1, to be used as a key in a CBOR map. Different ranges of values use different registration policies, see Table 3.

Description: A short description of the entry providing an overview of its semantic meaning and its expected use-cases.

Reference: A link to a permanent and readily available specification defining the value syntax and semantics to be used for this key.

5.5.3. Registration Form

Requested Range:
Name (JSON Key Value):
Description:
Specification:

Figure 8

5.5.4. Initial Values

Name (JSON Key Value)	Label (CBOR Integer Key Value)	Description	Reference
uas_type	0	Enumeration of UAS Type	This RFC, Section 4.1
uas_ids	1	UAS IDs	This RFC, Section 4.1
auth	2	Authentication Data	This RFC, Section 4.1
self_id	3	Self ID	This RFC,

			Section 4.1
classification	4	UA Category & Class	This RFC, Section 4.1
area	5	Area Count, Radius, etc.	This RFC, Section 4.1
operator_id	6	Operator ID	This RFC, Section 4.1
ua_status	7	Enumeration of Status	This RFC, Section 4.1
ua_position	8	UA Position	This RFC, Section 4.1
ua_bearing	9	Bearing / Track Direction	This RFC, Section 4.1
ua_speed	10	Vertical & Horizontal Speeds	This RFC, Section 4.1
ua_height	11	Height Above Ground or Take-off	This RFC, Section 4.1
accuracy	12	Enumerations of Accuracies	This RFC, Section 4.1
operator_position	13	Operator Position	This RFC, Section 4.1
timestamp	14	UTC Timestamp	This RFC, Section 4.1
compliance	15	Compliance	This RFC,

		Enumeration	Section 4.1
raa	23	RAA	This RFC
hda	24	HDA	This RFC
csr	25	DER Encoded X.509 CSR	This RFC
uas	26	UAS Datum Map	This RFC, Section 4.1
utm	27	UTM Operational Intent & Source	This RFC
spec	28	CAA Specification Code	This RFC
canonical_cert	39	DER Encoded X.509 Certificate	This RFC
be_chain	30	List of Broadcast Endorsements	This RFC
registrant	31	Registrant Information	This RFC
hhit	32	Hierarchial Host Identity Tag	This RFC
oracle_x509	33	Oracle X.509 Certificate	This RFC
uas_serial_number	34	ANSI/CTA 2063-A UAS Serial Number	This RFC
caa_assigned_id	35	CAA Assigned ID	This RFC
pilot_license_id	36	Pilot License Number	This RFC

Table 4: Pre-Allocated DRIP CBOR/JSON Keys

6. Security Considerations

The considerations discussed in [RFC9153], [RFC9374], [RFC9434], [RFC9575] and [DET-DNS] apply.

DIMEs use and provide various methods to protect data through: Attestation, Authentication, Authorization, Access Control, Attribution, Accounting, and Audit (AAA). All data, handled under DRIP, MUST be protected by AAA, as per applicable regulation and policy (which, in some cases, for public data, may impose minimal requirements). All private data MUST also be protected by strong encryption where permitted by applicable law etc. These requirements apply to data at rest and in transit in all phases of the process, i.e. registration and query.

Attestation may be mandated by CAAs for devices (such as UA). Remote ATtestation ProcedureS (RATS) [RFC9334] is recommended for DRIP. The specific attestation mechanisms in a given jurisdiction are out of scope for this document.

Author Note: RGM, what else should go here?

6.1. Cryptographic Materials

Best practices dictate that cryptographic materials that should only be available to selected parties SHOULD be generated by one or more of those parties and stored accessibly only on those parties devices. E.g. the asymmetric key-pair from which a DET will be derived SHOULD be generated by the entity identified by that DET and the corresponding private key should be stored only on that entity’s device. There may be scenarios where other parts of the UAS MAY generate the cryptographic materials and provision them as needed during an operation. Any such system MUST ensure security of the cryptographic material is guaranteed.

7. Privacy & Transparency Considerations

The considerations discussed in [RFC9153] and Section 10 of [RFC9434] apply.

Author Note: do we copy/paste Section 10 of [RFC9434] and update here or is below sufficient?

7.1. DETs as Session ID and Authentication Key ID

The properties of a DET allow it to be used as a Session ID and/or an Authentication Key ID. There may be scenarios in which Session IDs are desired for privacy but Authentication is not; although this may reduce transparency and security, a DET MAY be used exclusively as a Session ID in such. There are scenarios in which Authentication is desired but Session IDs are not (e.g. where a CAA does not allow Session IDs); a DET MAY be used exclusively as an Authentication Key ID in such. In the scenario expected to be most common, both Session IDs and Authentication are desired; the same DET SHOULD be used as both the Session ID and Authentication Key ID in such. Consequences and operational impact of using different DETs for the Session ID and Authentication Key ID of the same entity are unknown and not covered in this document.

7.2. Selective Encryption

Author Note: renamed section and/or add more explicit disclaimer?

Selective encryption may be allowed in some circumstances. An HHIT may be used in a private lookup to access decryption key material or obtain decryption as a service.

Selective encryption of UAS RID using DRIP is only allowed when the DET to be used as the Session ID is issued by an HDA associated with a USS and that DET is associated with the corresponding Operational Intent in UTM. This enables the encryption of selected data elements in Broadcast RID to provide a layer of privacy for operators (e.g. their position) without compromising transparency to entities (e.g. public safety / law enforcement personnel) that need to know.

Selective encryption typically requires network connectivity of the Observer to perform the private query to obtain the decryption service or key material. CAAs should consider the expected Observer environment and prohibit encryption wherever and whenever Observers cannot reasonably be expected to have connectivity. For example selective encryption, per CAA regulations, MAY be allowed in dense wireless IP connectivity areas (e.g. urban) but prohibited in poor wirelessly covered areas (e.g. rural).

The issue of Observer network connectivity MAY be mitigated with the use of a shared decryption key used by multiple Session IDs under a given USS/HDA over a period of time, that is preloaded onto the Observer device before connectivity is lost. For example Observers MAY query USS/HDAs for flight volumes in which they are interested to preload their decryption key(s) for the Operational Intents/Session IDs that day.

8. References

8.1. Normative References

[DET-DNS] Wiethuechter, A. and J. Reid, "DRIP Entity Tags in the Domain Name System", Work in Progress, Internet-Draft, draft-ietf-drip-registries-33, 19 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-drip-registries-33>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC9153] Card, S., Ed., Wiethuechter, A., Moskowitz, R., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Requirements and Terminology", RFC 9153, DOI 10.17487/RFC9153, February 2022, <https://www.rfc-editor.org/rfc/rfc9153>.

[RFC9374] Moskowitz, R., Card, S., Wiethuechter, A., and A. Gurtov, "DRIP Entity Tag (DET) for Unmanned Aircraft System Remote ID (UAS RID)", RFC 9374, DOI 10.17487/RFC9374, March 2023, <https://www.rfc-editor.org/rfc/rfc9374>.

[RFC9434] Card, S., Wiethuechter, A., Moskowitz, R., Zhao, S., Ed., and A. Gurtov, "Drone Remote Identification Protocol (DRIP) Architecture", RFC 9434, DOI 10.17487/RFC9434, July 2023, <https://www.rfc-editor.org/rfc/rfc9434>.

[RFC9575] Wiethuechter, A., Ed., Card, S., and R. Moskowitz, "DRIP Entity Tag (DET) Authentication Formats and Protocols for Broadcast Remote Identification (RID)", RFC 9575, DOI 10.17487/RFC9575, June 2024, <https://www.rfc-editor.org/rfc/rfc9575>.

[STD95] Internet Standard 95, <https://www.rfc-editor.org/info/std95>. At the time of writing, this STD comprises the following:

Newton, A., Ellacott, B., and N. Kong, "HTTP Usage in the Registration Data Access Protocol (RDAP)", STD 95, RFC 7480, DOI 10.17487/RFC7480, March 2015, <https://www.rfc-editor.org/info/rfc7480>.

Hollenbeck, S. and N. Kong, "Security Services for the Registration Data Access Protocol (RDAP)", STD 95, RFC 7481, DOI 10.17487/RFC7481, March 2015, <https://www.rfc-editor.org/info/rfc7481>.

Hollenbeck, S. and A. Newton, "Registration Data Access Protocol (RDAP) Query Format", STD 95, RFC 9082, DOI 10.17487/RFC9082, June 2021, <https://www.rfc-editor.org/info/rfc9082>.

Hollenbeck, S. and A. Newton, "JSON Responses for the Registration Data Access Protocol (RDAP)", STD 95, RFC 9083, DOI 10.17487/RFC9083, June 2021, <https://www.rfc-editor.org/info/rfc9083>.

Blanchet, M., "Finding the Authoritative Registration Data Access Protocol (RDAP) Service", STD 95, RFC 9224, DOI 10.17487/RFC9224, March 2022, <https://www.rfc-editor.org/info/rfc9224>.

8.2. Informative References

[F3411] ASTM International, "Standard Specification for Remote ID and Tracking", ASTM F3411-22A, DOI 10.1520/F3411-22A, July 2022, <https://www.astm.org/f3411-22a.html>.

[F3548] ASTM International, "Standard Specification for UAS Traffic Management (UTM) UAS Service Supplier (USS) Interoperability", ASTM F3548-21, DOI 10.1520/F3548-21, July 2025, <https://www.astm.org/f3548-21.html>.

[F3586] ASTM International, "Standard Practice for Remote ID Means of Compliance to Federal Aviation Administration Regulation 14 CFR Part 89", ASTM F3586-22, DOI 10.1520/F3586-22, July 2022, <https://www.astm.org/f3411-22a.html>.

[ICAO-ACCP] International Civil Aviation Organization, "ICAO Aviation Common Certificate Policy", July 2025.

[ICAO-NUMBERS]
International Civil Aviation Organization, "ICAO Remote ID Number Registry", July 2025, <https://www.icao.int/airnavigation/IATF/Pages/ASTM-Remote-ID.aspx>.

[MEDIA-TYPES]
IANA, "Media Types", July 2025, <https://www.iana.org/assignments/media-types/>.

[RDAP-EXT] IANA, "RDAP Extensions", July 2025, <https://www.iana.org/assignments/rdap-extensions/>.

[RFC4108] Housley, R., "Using Cryptographic Message Syntax (CMS) to Protect Firmware Packages", RFC 4108, DOI 10.17487/RFC4108, August 2005, <https://www.rfc-editor.org/rfc/rfc4108>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/rfc/rfc5246>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <https://www.rfc-editor.org/rfc/rfc5952>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/rfc/rfc7252>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/rfc/rfc7519>.

[RFC7616] Shekh-Yusef, R., Ed., Ahrens, D., and S. Bremer, "HTTP Digest Access Authentication", RFC 7616, DOI 10.17487/RFC7616, September 2015, <https://www.rfc-editor.org/rfc/rfc7616>.

[RFC7617] Reschke, J., "The 'Basic' HTTP Authentication Scheme", RFC 7617, DOI 10.17487/RFC7617, September 2015, <https://www.rfc-editor.org/rfc/rfc7617>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/rfc/rfc8392>.

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/rfc/rfc8610>.

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <https://www.rfc-editor.org/rfc/rfc8615>.

[RFC9063] Moskowitz, R., Ed. and M. Komu, "Host Identity Protocol Architecture", RFC 9063, DOI 10.17487/RFC9063, July 2021, <https://www.rfc-editor.org/rfc/rfc9063>.

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <https://www.rfc-editor.org/rfc/rfc9334>.

[RFC9560] Hollenbeck, S., "Federated Authentication for the Registration Data Access Protocol (RDAP) Using OpenID Connect", RFC 9560, DOI 10.17487/RFC9560, April 2024, <https://www.rfc-editor.org/rfc/rfc9560>.

[STD90] Internet Standard 90, <https://www.rfc-editor.org/info/std90>. At the time of writing, this STD comprises the following:

Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[STD94] Internet Standard 94, <https://www.rfc-editor.org/info/std94>.

At the time of writing, this STD comprises the following:

Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

[WELL-KNOWN] IANA, "Well-Known URIs", July 2025, <https://www.iana.org/assignments/well-known-uris/>.

[_802.1AR] IEEE, "IEEE Standard for Local and Metropolitan Area Networks - Secure Device Identity", DOI 10.1109/ieeestd.2018.8423794, July 2025, <https://doi.org/10.1109/ieeestd.2018.8423794>.

Appendix A. DRIP CDDL Prelude

This appendix is normative.

```
; DRIP Type Definitions
; (follow STD94 Section 6 for CBOR/JSON conversion)
metadata = (
  ? uas: uas-datum,
  ? utm: [
    operational_intent: uuid,
    uss_uri: uri
  ],
  ? raa: {
    ? spec: tstr,
    + &(tstr, int): any
  },
  ? hda: {
    ? spec: tstr,
    + &(tstr, int): any
  }
  * &(tstr, int): any
)
x509 = bstr ; DER encoded X.509 object (CSR or Certificate)
endorsement = bstr .size 136
hash = bstr .size 8 ; generated with cSHAKE
auth-data = bstr .size 1..362
uas-id = bstr .size 20
utc = tdate / time

; DRIP Keys (JSON key = CBOR label)
; private_use = -2^64 .. -25
; astm = -24 .. 23
```

```
   uas_type = 0
   uas_ids = 1
   auth = 2
   self_id = 3
   area = 4
   classification = 5
   operator_id = 6
   ua_status = 7
   ua_position = 8
   ua_bearing = 9
   ua_speed = 10
   ua_height = 11
   accuracy = 12
   operator_position = 13
   timestamp = 14
   compliance = 15
   raa = 23
   hda = 24
   csr = 25
   uas = 26
   utm = 27
   spec = 28
   canonical_cert = 29
   be_chain = 30
   registrant = 31 ; jcard or pointer (i.e. hhit of operator)?
   hhit = 32
   oracle_x509 = 33
   uas_serial_number = 34
   caa_assigned_id = 35
   pilot_license_id = 36

   ; External Tags
   uuid = #6.37
   ip6 = #6.54
   position = #6.103([
           latitude: number,
           longitude: number,
           altitude: number
   ])
```

Appendix B. HID Abbreviation

The DET MAY be abbreviated. This is useful for display applications with limited screen real-estate as the display of the entire 128-bit (32 characters in hexadecimal, even without recommended punctuation or spacing) IPv6 address can be costly. Abbreviations SHOULD follow the following format rules.

Hierarchy Level: Each hierarchy level (RAA/HDA) is represented by a maximum of six alphanumeric characters. This abbreviation SHOULD NOT be a single character in length, for obvious reasons of not being very useful. The decimal representation does fit into the 4 character restriction but is NOT RECOMMENDED. The RECOMMENDED abbreviation for the RAA level is to use the ISO3166-1 Alpha 2 code for nations. This abbreviation MAY be found in DNS under a HHIT RRType of the entity or its parents. If there is no abbreviation hint display devices SHOULD use a fixed size four character hexadecimal representation of the value. It is RECOMMENDED that display applications specify a default RAA value, and only display the RAA abbreviation explicitly when it does not match the default.

Entity Hash: The entity is represented by a fixed size four character hexadecimal string using the last four characters of the DET. If a collision (within the same HID space) on display occurs, the four characters SHOULD shift to the left by one hexadecimal character until the collision is resolved. This window MUST stay within the last sixteen hexadecimal characters of the DET. The : character found in an IPv6 address string is ignored.

Review(SWC): Do we need to insert a flag character such as "*" to indicate when a hash has been shifted? If the abbreviation is used only to avoid conflicts, no; but if anyone actually knows some DETs in their operation, if they get shifted, they probably won’t recognize them.

Delimiter: Each section is delimited by a single character. This can be any whitespace character (except newline and tab) or any non-alphanumeric character (period, comma, semicolon, etc.). It is RECOMMENDED that the delimiter is consistent between components. The RECOMMENDED delimiter is the colon (:) character.

For example a DET with the values of RAA 16383 and HDA 1 without any abbreviation hint from DNS is represented by the string 3FFF 0001 xxxx with xxxx representing a entity hash. If an abbreviation for the RAA (such as DRIP01) and HDA (such as TEST) are found in DNS then the DET can be represented with the string of DRIP01 TEST xxxx.

For an example of the entity hash, let’s assume there are two DETs with the following hashes in the same HID: 0000:1111:2222:3333 and 0000:2222:1111:3333. At first both DETs are represented with the same abbreviation: RAA HDA 3333. One of these DETs is selected by the display application to shift the display hash one character to avoid the collision resulting in the following two abbreviations: RAA HDA 3333 and RAA HDA 1333.

Appendix C. Compliance Testing

It is the responsibility of each parent node in the tree that a child node pass functional interoperability testing prior to issuing a certificate for the child node.

Author: This section is a work in progress.

All interfaces MUST be tested on valid and invalid data (such as being malformed). When policy is required on an interface all essential permutations of the policy MUST be tested and all possible permutations SHOULD be tested. The policy engine MUST be invoked to validate proper decisions (including PERMIT and DENY or their equivalents) and actions are being made. All data returned from an interface MUST be tested to check that it conforms with specifications.

There is a range in the RAA space allocated for experimentation and testing purposes. Sub-ranges can be delegated to parties, for a limited period of time, at the behest of the RAA Designated Expert, to test DIME interoperability (e.g. HDA to RAA, and RAA to RAA interactions) in any of the below subsections. The IANA Considerations section of [DET-DNS] contains more information on how these delegations are to be handled.

Appendix C.4 provides a set of forms to be filled out and submitted as part of a CAA compliance process for using DRIP.

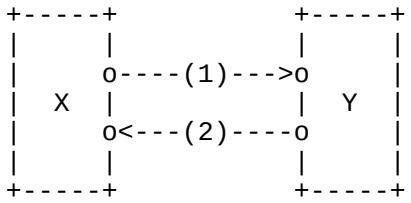

Figure 9: System Interface Test Diagram

C.1. Registration Interface

X, Y pairs: (registrant, HDA), (HDA, RAA)

1: Ensure that endpoints on Y can be accessed according to Y’s policy by X. Ensure that the endpoints conform to the normative requirements of Test that Y interface properly handles valid and malformed data. Test that Y securely generates proper artifacts and stores them.

2: Ensure that the interactions for (1) properly returns data to X from Y. This data MUST include the Broadcast Endorsements, X.509s and any other data elements required.

C.2. Public Query Interface

X, Y pairs: (UAS Observer, HDA), (UAS Observer, RAA), (HDA, RAA), (RAA, HDA), (HDA, HDA'), (RAA, RAA')

1: Ensure that Y has a properly configured and publicly accessible Authoritative Name Server for its delegated ip6.arpa domain.

2: Ensure that Y returns the valid RRTypes defined in [DET-DNS] to X based on an ip6.arpa query via standard DNS methods (i.e. using UDP on port 53). Ensure that the HHIT RRType contains the correct Certificate with an URI.

C.3. Private Query Interface

X, Y pairs: (UAS Observer, HDA), (UAS Observer, RAA), (HDA, RAA), (RAA, HDA), (HDA, HDA'), (RAA, RAA')

1: Ensure that the provide URI from the public query interface points to a valid URI. Ensure that the endpoint on Y has proper AAA mechanisms as defined in this document and enforces the proper policy options.

2: Ensure that the Y endpoint securely returns the data expected according to policy (matrix of authorized, valid request, unauthorized and invalid request) to X.

C.4. Compliance Submission Forms

Author Note: talk with AS on this section

Appendix D. DRIP Key Infrastructure X.509 (DKIX)

This appendix is normative.

The following sections define profiles and any specific field content for X.509 certificates that serve as a "shadow" of Endorsements that make up the DRIP Key Infrastructure (DKI). It is recommended that DKIX-Full (Appendix D.3) be deployed, as its CA certificates contain ICAO policy OIDs the reflect on the whole DKIX deployment.

A DKIX compliant Certificate Authority (CA) MUST implement generation of the DKI Endorsements. It SHOULD also generate both DKIX profiles (Appendix D.2 and Appendix D.3). In this section Endorsing Server and CA are interchangeable.

Author Note: At this point in defining the shadow PKIs, alternatives to a strict hierarchy is still an open work item.

D.1. Signing Request

The Certificate Signing Request (CSR) is a mandatory for all DET registrations. Depending on the entity being registered, there is specific content rules.

```
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject:
        Subject Public Key Info:
            Public Key Algorithm: ED25519
                ED25519 Public-Key:
                pub:
        Attributes:
        Requested Extensions:
            X509v3 Subject Alternative Name: critical
                IP Address:
    Signature Algorithm: ED25519
```

Subject: As defined in Section 4.1.2.6 of [RFC5280]. This field is filled in based on the type of entity being registered. When the attribute type is set to SERIAL_NUMBER, it MUST contain the ANSI/CTA 2063-A UAS Serial Number encoded as a string. The value of this field for other entity types are subject to policy and are out of scope for this document.

Subject Alternative Name: When the registrant knows which HID and/or Suite ID they want the CSR SHOULD contain the Subject Alternate Name extension as defined in Section 4.2.1.6 of [RFC5280] using ipAddress containing the fully formed DET and MUST mark the extension as critical. This DIME MUST check to ensure that the DET located in the extension is properly generated with the included public key of this CSR. If the registrant does not know or care the value of their HID and/or Suite ID, the Extensions field MUST NOT appear and the DIME will use its HID for DET generation using the public key provided by this CSR.

The use of other Extension fields are out of scope for this document.

D.2. Lite Profile

DKIX-Lite is designed to fully mirror the DKI in the smallest reasonable X.509 certificates (e.g. 240 bytes for DER), but still adhere to [RFC5280] MUST field usage. The profile can be found in Figure 10 and a field matrix found in Table 5.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
        Signature Algorithm: ED25519
        Issuer: CN =
        Validity
            Not Before:
            Not After :
        Subject:
        Subject Public Key Info:
            Public Key Algorithm: ED25519
                ED25519 Public-Key:
                pub:
        X509v3 extensions:
            X509v3 Subject Alternative Name: critical
                IP Address:
    Signature Algorithm: ED25519
    Signature Value:
```

Figure 10: DKIX-Lite Profile

Field	Authorization	Issuing	Operational	Comments
Serial Number	MUST	MUST	MUST	Appendix D.4.1.1
Subject	MUST	MUST	MUST NOT	Appendix D.4.2
Issuer	MUST	MUST	MUST	Appendix D.4.3
Subject Alternative Name (SAN) IP6	MUST	MUST	MUST	Appendix D.4.4
Subject Alternative Name (SAN) URI	MAY	MAY	MAY	Appendix D.4.5
Basic Constraints	MUST	MUST	MUST NOT	CA=True, flagged as Critical
Subject Key Identifier (SKI)	MUST NOT	MUST NOT	MUST NOT	-
Authority Key Identifier (AKI)	MUST NOT	MUST NOT	MUST NOT	-
Key Usage	MAY	MAY	MAY	-
CA Policy OIDs	MAY	MAY	MAY	Appendix D.4.8

Table 5: DKIX-Lite Field Matrix

D.3. Full Profile

The X.509 certificates are minimalistic (less than 400 bytes for DER). Any DRIP specific OIDs should come from the ICAO arc (see Appendix D.4.8). The profile can be found in Figure 11 and a field matrix found in Table 6.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
        Signature Algorithm: ED25519
        Issuer: CN =
        Validity
            Not Before:
            Not After :
        Subject:
        Subject Public Key Info:
            Public Key Algorithm: ED25519
                ED25519 Public-Key:
                pub:
        X509v3 extensions:
            X509v3 Subject Alternative Name: critical
                IP Address:
                URI:
            X509v3 Subject Key Identifier:
            X509v3 Authority Key Identifier:
            X509v3 Basic Constraints: critical
            X509v3 Key Usage: critical
    Signature Algorithm: ED25519
    Signature Value:
```

Figure 11: DKIX-Full Profile

Field	Authorization	Issuing	Operational	Comments
Serial Number	MUST	MUST	MUST	Appendix D.4.1.2
Subject	MUST	MUST	MUST NOT	Appendix D.4.2
Issuer	MUST	MUST	MUST	Appendix D.4.3
Subject Alternative Name (SAN) IP6	MUST	MUST	MUST	Appendix D.4.4
Subject Alternative Name (SAN) URI	MAY	MAY	MAY	Appendix D.4.5
Basic Constraints	MUST	MUST	MUST NOT	CA=True, flagged as Critical
Subject Key Identifier (SKI)	MUST	MUST	MUST NOT	Appendix D.4.6
Authority Key Identifier (AKI)	MUST	MUST	MUST	Appendix D.4.7
Key Usage	SHOULD	SHOULD	SHOULD	-
CA Policy OIDs	RECOMMENDED	RECOMMENDED	RECOMMENDED	Appendix D.4.8

Table 6: DKIX-Full Field Matrix

D.4. Certificate Fields

The following sections contain clarifications on the usage of fields in the DKIX when they deviate from "standard" X.509 [RFC5280] practice or have specific functions in the context of DRIP.

D.4.1. Serial Number

D.4.1.1. Lite

The Serial Number is a MUST field, but it has no usage in this DRIP-Lite. It is 1-byte in size and thus duplicates are guaranteed. To drop this field could make many X.509 parsing libraries fail.

However, CA certificate’s Serial Number MAY be the common 20 bytes. This is to avoid possible issues with general software expecting this size Serial Numbers for CAs.

If Serial Numbers are incrementally assigned, 31 bits are sufficient for an Issuing CA with 2B DETs active. A CA should determine its best-used Serial Number field size to limit the impact of this field on the certificate size.

D.4.1.2. Full

The certificates will contain a 20-byte randomly generated Serial Number, compliant with CABForum recommendations. Serial Numbers are included for CRL functionality.

D.4.2. Subject

The Subject field is only used in Authorization and Issuing Certificates. For Operational Certificates, the Subject is Empty and the DET will be in Subject Alternative Name (SAN, Appendix D.4.4). In the SAN, the DET can be properly encoded as an IPv6 address.

The format defined in Figure 12 MUST be used. The CA Subject Name serves a dual purpose: foremost, to place the CA within the DKI tree, but secondly for outside of DRIP usage to tag that this CA’s function is to serve DRIP Entities.

```
DRIP-{APEX|RAA|HDA}-{A|I}[-RAA#][-HDA#]
```

Figure 12: CA Subject Format

As examples; an Authorization DET for RAA 16376 would be: DRIP-RAA-A-16376 and an Issuing DET for HDA 16376 under the same RAA would be: DRIP-HDA-I-16376-16376.

D.4.3. Issuer

The Issuer MUST be the higher level's DET.

The Issuer for the Apex Authorization certificate MUST be its DET (indicating self-signed). If the RAA Authorization certificate is self-signed (i.e., no Apex), its Issuer is its DET.

The Issuer content of its DET assists in finding this specific Issuer in the DRIP ip6.arpa. DNS tree and any additional information.

D.4.4. Subject Alternative Name IP6

Subject Alternative Name is only used in Operational certificates. It is used to provide the DET as an IP address with an Empty Subject (SAN MUST be flagged as Critical).

The Subject Alternative Name is also used in Manufacturer DET certificates. These may contain the hardwareModuleName as described in [_802.1AR] that references [RFC4108].

Per [RFC5280] and [_802.1AR], Manufacturer DET certificates with hardwareModuleName MUST have the notAfter date as 99991231235959Z.

D.4.5. Subject Alternative Name URI

This field contains a pointer in the form of a URI where additional information on the CA and certificates under its control can be found. For example, an authorized authority may access end-entity PII like an Operator phone number through this URI link using the method specified in Section 4.3.2.

Authorization certificates MAY have a SAN URI and MUST when it is self-signed. Issuing certificates SHOULD have a SAN URI. Operational certificates MAY have a SAN URI.

The RECOMMENDED configuration with respect to Issuing and Operational certificates for a CA is to place the SAN URI in the Issuing certificate and exclude them in Operational certificates. When multiple providers are used by the CA to handle additional information there SHOULD be a unique Issuing certificate with SAN URI for each provider, and Operational certificates MUST NOT contain a SAN URI. Otherwise a CA with multiple providers MUST NOT have a SAN URI in the Issuing certificate and MUST set the SAN URI to the specific provider for each Operational certificate.

The contents of the SAN URI are the base URI of the host to query for additional information of a DET.

Author Note: do we want to create a ./well-known/hhit-query? Then base URIs in the certificates can become something like: https://hda.example.com, then the clients append /.well-known/hhit-query/<ip6> (<ip6> here is a specific DET). Path preservation can then be used by the server to replace /.well-known/hhit-query/ with whatever is required on the path for the specific server such as /my/bloated/path/to/rdap/ip/.

D.4.6. Subject Key Identifier

The Subject Key Identifier MUST be the DET. This is a major deviation from "standard" X.509 certificates that hash (normally with SHA2) the Public Key to fill the Subject Key Identifier.

The Subject Key Identifier is NOT included in Operational certificates. If needed by some application, it is identical with SAN.

D.4.7. Authority Key Identifier

The Authority Key Identifier MUST be the higher level's Subject Key Identifier (i.e. DET). This partially follows standard practice to chain up the Authority Key Identifier from the Subject Key Identifier, except for how the Subject Key Identifiers are populated.

The Authority Key Identifier for the Apex Authorization (or self-signed RAA Authorization) certificate MUST be the Subject Key Identifier (indicating self-signed).

D.4.8. CA Policy OIDs

It is recommended that a CA have policy OIDs defining rules for EEs within its domain. The OIDs used here will tend to be within ICAO's arc of 1.3.27.16.

If the CA certificate has policy OIDs, it MUST include an ICAO LOA OID (arc of 1.3.27.16.1.1.0) defining "the confidence in the security measures that protect the private key and manage the certificate lifecycle".

OID	Description	Applicability
1	Low	This level is relevant to environments where Risks and consequences of data Compromise are low. Subscriber Private Keys shall be stored in software at this

		Identity Assurance Level (IAL).
2	LowDevice	This policy is identical to that defined for the Low Assurance policy (above) with the exception of identity proofing, re-key, and Activation Data.
3	Low-TSP Mediated Signature	This policy is identical to that defined for the Low Assurance policy (above) with the exception that the Private key is not in the possession of the user, but rather by a Trust Service Provider.
4	Medium	This level is relevant to environments where Risks and consequences of data Compromise are moderate. This may include transactions having substantial monetary value or Risk of fraud or involving access to private information where the likelihood of malicious access is substantial. Subscriber Private Keys shall be stored in software at this IAL.
5	MediumDevice	This policy is identical to that defined for the Medium Assurance policy (above) with the exception of identity proofing, re-key, and Activation Data.
6	Medium-TSP Mediated Signature	This policy is identical to that defined for the Medium Assurance policy (above) with the exception that the Private key is not in the possession of the user, but rather by a Trust Service Provider.
7	MediumHardware	This policy is identical to that defined for the Medium Assurance policy (above) with the exception of Subscriber Cryptographic Module requirements described in [ICAO-ACCP].
8	MediumDeviceHardware	This policy is identical to that

			defined for the Medium Hardware Assurance policy (above) with the exception of identity proofing, re-key, and Activation Data.
9	High		This level is relevant to environments where Risks and consequences of data Compromise are high. Certificates issued at the High-CardAuth IAL shall only be issued for Card Authentication, as defined by NIST SP 800-73 or equivalent standard. Further, this policy is identical to that defined for the identical to the MediumHardware IAL except where specifically noted in [ICAO-ACCP].
10	High-CardAuth		This level is relevant to environments where Risks and consequences of data Compromise are high. Certificates issued at the High-cardAuth IAL shall only be issued for Card Authentication, as defined by NIST SP 800-73 or equivalent standard.
11	High-ContentSigning		This level is relevant to environments where Risks and consequences of data Compromise are High. This may include transactions having substantial monetary value or Risk of fraud or involving access to private information where the likelihood of malicious access is substantial. Certificates issued at the High IAL shall only be issued to human Subscribers.Certificates issued at the High-contentSigning IAL shall only be issued to the CMS for signing the HIGH card security objects (e.g. Certificates, CRLs, OCSP responses). Further, this policy is identical to that defined for the identical to the MediumHardware IAL except where specifically noted in [ICAO-ACCP].

Table 7: ICAO LOA OID Assignments under 1.3.27.16.1.1.0

In this document, the term “Device” is defined as a Non-Person Entity, i.e., an entity with a digital identity that acts in cyberspace but is not a human actor. This can include Organizations, hardware devices (e.g. a UA), software applications, and information artifacts.

Operational Certificates issued to Devices shall assert policies mapped to LowDevice, MediumDevice, MediumDeviceHardware, or High Content Signing policies. All other policies defined here should be reserved for human Subscribers when used in Operational Certificates. Thus it is recommended that Issuing CAs/HDAs should be segregated by device and human subscribers so policy can be stated at the CA level rather in individual certificates.

Authors’ Addresses

Adam Wiethuechter
AX Enterprize, LLC
4947 Commercial Drive
Yorkville, NY 13495
United States of America
Email: adam.wiethuechter@axenterprize.com

Robert Moskowitz
HTT Consulting
Oak Park, MI 48237
United States of America
Email: rgm@labs.htt-consult.com