Drone Remote ID Protocol
Internet-Draft
Intended status: Standards Track
Expires: 17 September 2026

A. Wiethuechter
AX Enterprize, LLC

16 March 2026

Hierarchical ORCHID Management Entity (HOME) Interfaces for Registration & Differential Access Query
draft-atw-home-interfaces-00

Abstract

This document standardizes the interfaces of an ORCHID Management Entity (OME) to allow clients to register and query with differential access an Overlay Routable Cryptographic Hash Identifier (ORCHID) such as a Hierarchial Host Identity Tag (HHIT). Existing technologies such as CBOR/JSON Object Signing & Encryption (COSE/JOSE) and Registration Data Access Protocol (RDAP) are selected to enable widespread interoperability.

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 17 September 2026.

Copyright Notice

Table of Contents

1. Introduction

1.1. Purpose

An ecosystem using Overlay Routable Cryptographic Hash IDentifiers (ORCHIDs, [RFC7343]) as its primary identifier can be a strong option for modern systems. The Hierarchical Host Identity Tags (HHITs, [RFC9374]) are used in the fast growing sector of Unmanned Aircraft Systems (UAS) to provide identity and authentication services using the Domain Name System (DNS).

With standardization, identity ecosystems can be created around the use of ORCHIDs to serve various use-cases. This provides widespread interoperability of the functions of registration, public query and private query using a selection of existing standards and methods. Being backed by cryptographic keys allow existing key infrastructure, certificates and policy to be leveraged with minimal overhead, modification or redeployment.

1.2. Background

HHITs are an evolution of the Host Identity Tags (HITs) in the Host Identity Protocol (HIP, [RFC9063]). All of these tags are categorized as an ORCHID which were standardized in [RFC7343] and updated by [RFC9374]. HHITs solve the problem of HITs being a flat namespace, which is hard to manage at a global scale, by adding hierarchy into the identifier. When the hierarchy is laid onto the DNS, interoperability of lookups becomes trivial.

[RFC9434] introduces the concept of the DRIP Identity Management Entity (DIME) that provides the critical function of registering and querying information around DETs. It further specifies logical entities of the Public Information Registry and Private Information Registry of which the former has been specified in [RFC9886] using the DNS.

The concept of the DIME can be backported to ORCHIDs in general with the addition of the term ORCHID Management Entity (HOME). In relation to key infrastructure policies and certificates the terms ORCHID Key Infrastructure (OKI) and ORCHID Key Infrastructure X.509 (OKIX) are introduced here to mirror but be distinct from the existing Public Key Infrastructure (PKI) and Public Key Infrastructure X.509 (PKIX).

1.3. Scope

This document provides specification of the interfaces and models for registration and differential access query of ORCHIDs to support identity services under an HOME. This is to provide specification for the final missing pieces of a DIME and is the exemplar use-case of using HHITs, such as the DET, as the core of an identity ecosystem.

The term HHIT is used when discussing the general technology while DET is used with discussion around UAS/aviation specific elements when possible. Additional terms reinforcing this delineation are found in Section 2. This document defines all data models in the Concise Data Definition Language (CDDL, [RFC8610]) with the full specification in Appendix B.

1.3.1. Disclaimer

The specifications in this document do NOT provide protection against incorrect (e.g. fraudulent) data entered during registration or asserted subsequently.

The selected technologies do 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 HOME, or the party on whose behalf the HOME is being operated, to validate the registration data. An HOME at a root in the hierarchy aligned with the scope of the data SHOULD provide services to obtain this goal, see Appendix A.

1.3.2. Targeted DRIP Requirements

The following requirements of [RFC9153] are satisfied when following this document: GEN-3, ID-4, REG-2, REG-4 and PRIV-2.

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 & Abbreviations

Hierarchical ORCHID Management Entity (HOME): A term to describe an entity providing various identity management functions (such as registration, query, etc.) to clients at a specific point in the hierarchy. HOME’s have four major functional components that can be co-located or distributed as a system of systems: Registrar, Registry, Authoritative Name Server and Certificate Authority. The DRIP Identity Management Entity (DIME) is a specialization of HOME in support of the UAS/aviation ecosystem.

ORCHID Key Infrastructure (OKI): A key infrastructure consisting of policies around entities using HHITs in a domain space. An example of an OKI is the DRIP Key Infrastructure (DKI) for UAS/ aviation.

ORCHID Key Infrastructure X.509 (OKIX): A set of X.509 certificate

profiles in compliance of Public Key Infrastructure X.509 (PKIX, [RFC5280]) that support an OKI. An example of an OKIX is the DRIP Key Infrastructure X.509 (DKIX) for UAS/aviation.

HOME Token: In the context of this specification; a HOME Token can be any of the Concise Binary Object Representation (CBOR; [STD94]) Object Signing & Encryption (COSE; [STD96]) or JavaScript Object Notation (JSON; [STD90]) Object Signing & Encryption (JOSE; [RFC7515], [RFC7516]) structure. See Section 4.1.

HRI Token: A HOME Token containing an HRI claim per Section 4.2.1.

HRR Token: A HOME Token containing an HRR claim per Section 4.2.2.

3. HOME Component Architecture & Functions

To properly provide the necessary services identified in both [RFC9153] and [RFC9434] a DIME, or more generally an HOME, requires four critical functions: _Registrar_, _Registry_, _Authoritative Name Server_ and _Certificate Authority_ as detailed below for this document and in Figure 1.

Registrar: This document covers the client facing interface of the Registrar function used by Registrants. Interactions done by the Registrar with other HOME functions are out of scope for this document.

Certificate Authority: This document does not cover specific details of this function of an HOME. However, it is expected to provide at least ORCHID Key Infrastructure X.509 (OKIX) certificates used in other parts of the HOME. These are [RFC5280] compliant certificates that support HHITs, see Section 6. Explicit policy for Certificate Authorities are left for more specific implementation and documents, such as a Concept of Operations.

Authoritative Name Server: The specific details of records hosted by the Authoritative Name Server as part of DNS is specified in [RFC9886] and are out of scope for this document. Interactions done by other HOME functions to manage an Authoritative Name Server are out of scope for this document. It should be noted that some artifacts generated during the registration process specified in this document end up in RRTypes hosted in DNS. This function serves as the Public Information Registry as defined by Section 4.1 of [RFC9434].

Registry: This document covers the client facing interface of the

Registry function used by Observers to query for additional details about the HHIT. Interactions done by the Registry with other HOME functions are out of scope for this document. It should be noted that artifacts generated during the registration process specified in this document MUST end up in the Registry for audit purposes. For this document the term Registry is short-hand for Private Information Registry as defined in Section 4.2 of [RFC9434].

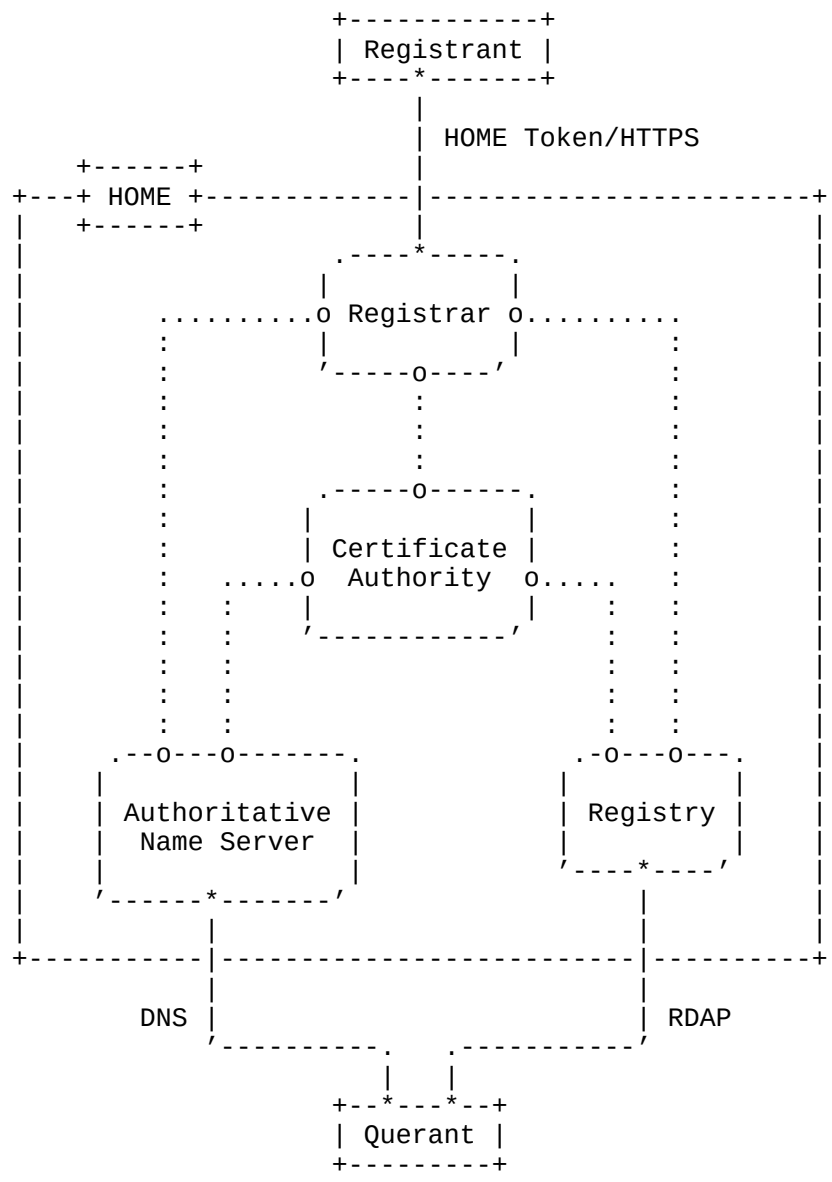


Figure 1: Simplified Diagram of HOME Registration/Query Components & Interactions

The specific interaction models and transports between the _Registrar_, _Certificate Authority_, _Registry_ and _Authoritative Name Server_ when not co-located (i.e. as a distributed system of systems) is out of scope for this document. Note that of the

duplicate links feeding _Authoritative Name Server_ and _Registry_ one set is chosen in an implementation for each. A number of these functions/interactions are lifted from the DNS and the existing relationships and protocols between such entities in that domain also can apply here. For example, between _Registrar_ and _Registry_ is the Extensible Provisioning Protocol (EPP, [STD69]) or equivalents such as RESTful Provisioning Protocol (RPP).

This document details the interactions between a registering client (known as _Registrant_) and the _Registrar_ through an HTTPS interface as described in Section 4 and the RDAP interaction by _Querants_ (such as _Observers_) and a _Registry_ as described in Section 5. These interfaces can be either used by machines as an API or provided with a UI for humans to interact with such as on a web browser.

The elements of any maps found in CDDL for this specification MUST be declared by an ecosystem. An initial key set for elements in aviation use cases are registered in Section 7.5. Context MAY be provided in-band in HTTP using a registered Media Type as part of the Content-Type header for what key set to use to avoid processing ambiguity.

4. Registration

This sections details the process of registration following Figure 2. This Z-diagram is the upper link of Figure 1 labeled with "HOME Token/HTTPS" between _Registrant_ and _Registrar_.

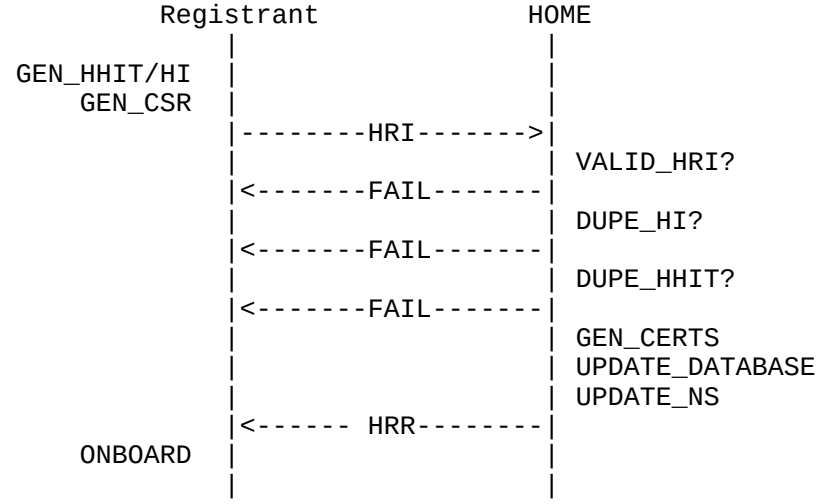


Figure 2: Registration Z-Diagram

The mechanism for registration is the use of COSE or JOSE over HTTPS. COSE MUST be supported and JOSE is RECOMMENDED for flexibility for non-constrained _Registrants_. The rest of this section specifies exact behavior for this interface of an HOME.

4.1. HOME Token

A HOME Token MUST be encoded in either COSE or JOSE. A token MAY contain multiple layers and SHOULD have the Content-Type parameter, as part of an unprotected header, set accordingly for each.

4.1.1. Signature Layer

Respective header parameters MUST be used to indicate the verification key. When the key is part of the payload (usually an HRI claim set, Section 4.2.1) a protected header parameter MUST be used (jwk for JOSE and kccs for COSE) to carry the verification key to avoid needing to parse the payload segment of the structure. For all other cases a KID parameter is used containing an HHIT, encoded per Appendix B.3, and placed in a unprotected header.

4.1.2. Encryption Layer

HOME Tokens SHOULD be encrypted when the payload may contain Personally Identifiable Information (PII). For example, a HOME would not encrypt their response token (Section 4.2.2) containing relevant processing errors when the original request token (Section 4.2.1) failed to decrypt.

Asymmetric algorithms MUST be supported and symmetric MAY be supported. The precise list of encryption algorithm selected and policies are left for the HOME to decide and MUST be provided to clients.

Encryption is done with Elliptic Curve Diffie-Hellman (ECDH) using Ephemeral-Static (ES) keying. The static key is provided by the HOME while a unique ephemeral key is provided by the client for each HOME Token and/or recipient. The transfer of this ephemeral key is covered by the relevant COSE/JOSE specifications for using ECDH with encryption objects. The HOME MUST provide its static keys through public methods such as the DNS or an HTTPS GET endpoint.

4.2. Claims Set

A HOME Token is at its core a set of claims using a CWT or JWT claim set. The claims defined in this section are registered in the respective registries in Section 7.2 and either or both MUST be included in the claim set for HOME Tokens. A HOME Token carrying either of these claims is considered as an HRI Token or HRR Token respectively.

The claim set SHOULD also contain the claims of nbf, exp, and iat. The set MAY contain the aud or iss claim with a targeted HHIT, encoded per Appendix B.3, if known. Additional claims are out of scope for this document.

The layer (Section 4.1.1 or Section 4.1.2) carrying a claim set is RECOMMENDED to set the Content-Type parameter, as part of the unprotected header, to application/home+cbor or application/home+json. This SHOULD include the ctx parameter to indicate the set of keys/parameters being used for any maps in the payload structure as they are application specific.

4.2.1. HOME Registration Inquiry (HRI)

The structure of an HRI claim is around registration of a set of entities, each uniquely identified by the _Registrant_ that contains keys (also each uniquely identified) and metadata for each.

```
inquiry = [
  entities: { &(entity-id) => entity },
  metadata: map / null
]
entity = [
  hhit_entity_type: uint,
  keys: { &(key-id): key },
  metadata: map / null
]
key = [
  csr: bytes / text .b64u bytes,
  metadata: map / null
]
```

Figure 3: CDDL: HOME Registration Inquiry

entities: A map containing elements using key (_entity-id_), value (_entity_) pairs for each _entity_ being registered in the inquiry.

entity-id: A string or integer key, as part of the _entities_ map,

set by the _Registrant_.

entity: An array containing the elements of _hhit_entity_type_, _keys_ and _metadata_.

_hhit_entity_type_: An HHIT Entity Type as defined in the registry found at [IANA.DRIP].

keys: A map containing the key (_key-id_), value (_key_) pairs for the keys being registered entity in the inquiry for a given _entity_.

key-id: A string or integer key, as part of the _keys_ map for each _entity_, set by the _Registrant_. Constrained in uniqueness to an individual HRI and the keys as part of an _entity_.

key: An array containing the elements of _csr_ and _metadata_.

csr: OKI(X)-compliant Certificate Signing Request (CSR) that is DER encoded. See Section 4.3.1 for considerations on the CSR.

metadata: A map containing elements (key, value pairs) and their associated values related to keys/entities in the inquiry. The contents are subject to policy of an HOME and MUST be provided to their clients. The specific key set to be used SHOULD be indicated using the optional parameter of the registered Media Type (Section 7.8) as part of a Content-Type parameter. An initial key set is registered in Section 7.5.

Each nesting in Figure 3 contains its own _metadata_. This allows a _Registrant_ to set elements at a higher nesting in the model to be shared among nested sections. When combining from each nesting level, elements found closer to the entities[entity-id].keys[key-id].metadata map are given priority.

4.2.2. HOME Registration Response (HRR)

This claim is structured to use the provided _entity-id_s and _key-id_s from an HRI. It also can contain a list of failures, organized by those same identifiers.

```
response = [
  success: [
    entities: { &(entity-id): success-data },
    shared: map
  ] / null,
  failure: [ + error-data ] / null
]
success-data = [
  keys: { &(key-id): map },
  shared: map
]
error-data = [
  eid: &(entity-id) / null,
  kid: &(key-id) / null,
  cat: any,
  msg: any
]
```

Figure 4: CDDL: HOME Registration Response

success: An array containing two elements: _entities_ and _shared_. MUST be null when no successful registrations issued by the HOME.

entities: A map of elements with key (_entity-id_), value (_success-data_) pairs.

entity-id: A string or integer key, as part of the _entities_ map, provided in an HRI from a _Registrant_.

success-data: An array containing two elements: _keys_ and _shared_.

keys: A map of elements with key (_key-id_), value pairs. The value of each element is a map (keys from Section 7.5) containing artifacts of the registration associated with the specific key and MUST have an element containing the Canonical Registration Certificate.

key-id: A string or integer key, as part of the _keys_ map, provided in an HRI from a _Registrant_.

shared: Same as the _metadata_ field of HRI except contains artifact elements provided to the _Registrant_ for successful registration that is shared amongst either a set of _keys_ or set of _entities_.

failure: An array with at least one element of _error-data_ to

indicate failures, null otherwise.

error-data: An array of four elements encoding a validation or processing error of an HRI. When _eid_ or _kid_ are null it implies that all of that type are denoting the same failure explained via _cat_ and/or _msg_. When both _eid_ and _kid_ are null it indicates the processing of the HRI token itself or all keys across the HRI entities shared the same failure indicated by _cat_ and/or _msg_. See Section 4.4.1 for more details.

Similar to the _metadata_ in the HRI, Figure 4 has an _shared_ map at each level for data that is the identical across different keys/ entities. When combining from each nesting level, elements found closer to those in the success.entities[entity-id].keys[key-id] map are given priority.

4.3. Considerations

4.3.1. Keys & Certificate Signing Request

_Registrant_s obtains one or more Certificate Signing Requests (CSRs) that are OKI(X)-compliant by either generating their own key-pair(s), RECOMMENDED asymmetric, or obtaining them from a related party they are acting as a proxy for.

Cryptographic materials SHOULD be on the device intending to use said material in operation. If the _Registrant_ is acting as a proxy the methods of the generation and/or transfer of cryptographic materials to and from the device being proxied to is out of scope for this document.

If the _Registrant_ knows the HID of the _Registar_ and/or _Certificate Authority_ it wishes to be endorsed by, it SHOULD construct its HHIT accordingly and include it in the CSR per Section 6.2.1. The process of discovering a specific HID is out of scope for this document.

4.3.2. Registration Endpoint

The /.well-known/home/registry endpoint SHOULD be redirected to a fully qualified path by the HOME and MUST use 308 Permanent Redirect.

4.3.3. HRI Validation & Processing

HRI validation and processing is done across two functional components, which can be co-located or distributed, the _Registrar_ and _Certificate Authority_. The following text is based on the processing of an HRI Token. The HOME in question uses a single HHIT for both its _Registrar_ and _Certificate Authority_ roles.

The initial steps of decryption and signature validation are skipped as they are straightforward. Failure of either of these steps results in sending a response following Section 4.4.1. Assuming success the claim set is extracted from the HRI Token to be processed.

The nbf, ext, iat, iss and aud claims are validated, and then hri claim is processed recursively starting from the outside of the structure. Each entity merges their _metadata_ with the inquiry _metadata_ with entity elements taking precedence. Next each key merges its _metadata_ with the entity _metadata_ with key elements taking precedence. The HOME performs validation of the final combined _metadata_ map for each key using local validation and/or consultation of an "oracle" (see Appendix A).

Next each key CSR has its signature validated then the public key confirmed to be unique to the HOME’s scope. The HOME then constructs, using its Hierarchy HID and the CSR public key, a prospective HHIT for the key. If the CSR contains an HHIT as part of the Subject Alternative Name the two HHITs are checked for equivalence. The HOME finally checks that the HHIT is unique in its scope.

If everything above validates, the HOME endorses the new HHIT by generating and signing an OKI(x)-compliant Canonical Registration Certificate along with any other artifacts required for the entity registration. Any data for the _Registrant_ is then placed within the HRR success.entities[entity-id].keys[key-id] map using the same _entity-id_ and _key-id_ from the HRI. All artifacts of this endorsement are placed in the _Registry_ and artifacts that are considered public, such as those needed for [RFC9886], are placed into the _Authoritative Name Server_.

When a failure occurs at any point in the validation and/or processing of a entity/key the HRR _failure_ array is appended using the _entity-id_, _key-id_ and other elements to provide context for the issue. The HOME MAY compress the _failure_ array after processing. For example if all the keys for an _entity-id_ share the same failure the individual structures can be compressed to a single [<eid>, null, <cat>, <msg>] instead of one for each _key-id_.

The HOME upon handling all entities and keys a HOME constructs an HRR Token and responds to the _Registrant_ with it. The token SHOULD replicate the same layering as the original HRI Token. It MAY contain both the original HRI claim and the associated HRR claim.

4.4. Transport

This document defines the use of HTTPS as the transport for the HOME Tokens. The use of other transports are out of scope for this document.

For COSE, the resulting CBOR content MUST use a proper CBOR Tag to indicate what object is encoded. For JOSE content, the General or Flattened JWE JSON Serialization syntax MUST be used.

The "Content-Type" header of HTTPS SHOULD be used to indicate the content typing. For JOSE this is application/jose+json and for COSE is one of the following: application/cose; cose-type="cose-sign, application/cose; cose-type="cose-sign1, application/cose; cose-type="cose-encrypt, application/cose; cose-type="cose-encrypt0.

4.4.1. Response Codes

For HOME Tokens with a single entity/key object the HTTP Response Codes can be easily mapped. For example a key that fails processing due to a duplicate public key or HHIT can have the HOME Token returned using 409 Conflict. It is also simple for the first stages of an HOME Token carrying an HRI since it needs to successfully be decrypted and signature(s) before processing resulting in earlier failure conditions to respond to.

HOME Tokens with multiple entities/keys can be trivially mapped as well only when all entities/keys either pass or fail. Difficulty arises when a HOME Token has multiple entities/keys and not all of them fail or they all fail in different ways that can be attributed to either a 4xx or 5xx code.

For the above reasoning and to maintain interoperability HOMEs MUST follow the below guidelines for management of HTTP Response Codes and using the HRR claim’s _failure_ array and _error-data_ structure. Unless otherwise noted a responding HOME Token MUST use both encryption and signature layers.

422 Unprocessable Content: When a HOME Token fails to decrypt this

response code MUST be sent with only a signature layer (Section 4.1.1) signaling in the HRR claim the decryption issue(s) experienced. The _eid_, _kid_ and _cat_ fields MUST be set to null. The _msg_ SHOULD contains specific details about the failed decryption. More than one _error-data_ MAY be present for different decryption issues.

401 Unauthorized: Failed signature(s) of a HOME Token MUST use this response code with the HRR claim indicating the signature validation issue(s) experienced. The _eid_, _kid_ and _cat_ fields MUST be set to null. The _msg_ SHOULD contains specific details about what signature failed to verify. More than one _error-data_ MAY be present to separate signature(s) processed.

200 OK: This response code MUST be used to indicate the overall inquiry has been processed but that some entities and/or keys failed to be registered. The _cat_ field SHOULD be used to carry the respective 4xx or 5xx HTTP Response Code exhibited for the entities/keys to fail validation or processing. If _cat_ is not used, then clients MUST check contents of both the _success_ and _failure_ fields of the HRR.

201 Created: A response with this code MUST be used when all entities and keys in the original HRI were registered successfully and the _failure_ in HRR MUST be set to null.

4xx or 5xx: If all entities/keys fail with the same _cat_, used as in _200 OK_, then the HOME Token is sent using that HTTP Response Code. The _failure_ array in the HRR MUST be filled with _error-data_'s setting _eid_ and _kid_ as needed, _cat_ to null and _msg_ to any specific details for the given _eid_/_kid_ pair.

5. Differential Access Query

This section details the process of query for additional private information following the Z-diagram of Figure 5. It is the lower right link of Figure 1 labeled with "RDAP" between _Querant_ and _Registry_.

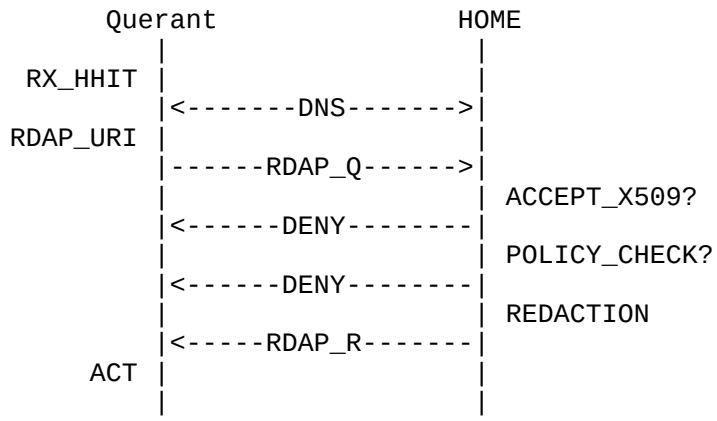


Figure 5: Query Z-Diagram

For HOME's the differential access mechanism known as Registration Data Access Protocol (RDAP, [STD95]) is selected as the interface for the _Registry_ and specified in the following sections. Public information is hosted by the _Authoritative Name Server_ and accessed using DNS methods as defined in [RFC9886]. Authentication of the HOME and _Querant_ is done through Mutual TLS using X.509 certificates.

As part of RDAP, the HOME MAY follow RFC9224 to be part of the RDAP bootstrap mechanism for discovery. This is not hard requirement as the same information can be obtained through the HHIT RRType from the _Authoritative Name Server_.

It is RECOMMENDED that eXtensible Access Control Markup Language (XACML) with Security Assertion Markup Language (SAML) is used to exchange policy information.

5.1. RDAP Query

A _Querent_ with an HHIT looking for private information is RECOMMENDED to perform a series of DNS queries to build a URI that is then used for an HTTPS GET query. The same URI MAY be obtained using the RDAP bootstrap process if the HOME followed RFC9224, but _Querants_ MUST NOT expect this.

The base URI needed is part of Subject Alternative Name extension of a Canonical Registration Certificate found in HHIT RRType defined in Section 5.1 of [RFC9886] and accessed by one or more reverse IPv6 DNS lookups. The exact certificate with the base URI for an HOME and its clients is determined by policy and where the URI is minimally duplicated across many certificates.

Once the base URI is found the _Querant_ concatenates the /.well-known/home/query/ string then the nibble-reversed HHIT to form the full query string.

The HTTPS GET method MUST be authenticated using one of the methods defined in Section 5.2.

5.1.1. Query Endpoint

The /.well-known/home/query/<IP address> endpoint SHOULD be redirected to a full qualified path conforming with Section 3.1.1 of [RFC9082] (i.e. ip/<IP address>) by the HOME. All other RDAP endpoints SHOULD NOT be implemented as specified in RFC9082. Other methods MAY be provided to enable differential access controlled queries of information pertaining to an HHIT but are out of scope for this document.

5.2. Differential Access Control

Per REG-2 and REG-4 of Section 4.4.1 of [RFC9153], RDAP queries to an HOME 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, this document standardizes the following RDAP behavior for authentication of clients and servers:

* MUST support HTTP Basic or Digest Authentication Scheme per Section 3.2 of [RFC7481] *AND*

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

* SHOULD use Mutual TLS but MAY do any RDAP compatible AAA for domestic queries

An HOME, when supporting Mutual TLS, SHOULD accept valid OKIX-compliant certificates and MAY accept other [RFC5280] (PKIX) compliant certificates. Standard TLS rejection methods MUST be followed to signal to the _Querant_ the rejection of the certificate and authentication.

The use of other RDAP compatible authentication mechanisms are out of scope for this document.

5.3. RDAP Extension & Response

A HOME MUST respond to a successful query using RFC9083. The HOME RDAP Extension shown in Figure 6 is placed under the key home as part of the main JSON map. The rdapConformance array MUST include the string literal of "home_v0" to signal conformance with this specification. Other RDAP specifications MAY be part of the response such as Section 5.4 of [RFC9083].

```
[
  hhit: [ ip6: text, hhit_entity_type: uint ],
  canon_x: text .b64u bytes,
  metadata: map / null
]
```

Figure 6: CDDL: HOME RDAP Extension

hhit: Array structure contain the HHIT Entity Type and the HHIT encoded per Appendix B.3.

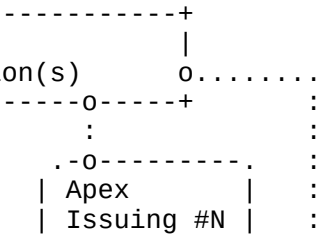
_canon_x_: Canonical Registration Certificate encoded in base64url.

metadata: Map that contains private information elements associated with the registration. Elements included are based on access policy and registration requirements of the HOME. Elements are redacted or removed using policy methods of the HOME and is out of scope for this document.

6. ORCHID Key Infrastructure (OKI)

The ORCHID Key Infrastructure (OKI) is focused on the delegation and management of the Hierarchy ID field of an HHIT and its levels.

There are two types of nodes in the hierarchy, Root and Inode, and relationships between nodes use tree terminology such as ancestor, descendant and degree. Additional, more administrative members, include the IPv6 Prefix and Apex. The relationship between these entities (and their Authorization and Issuing HHITs) are shown in Figure 7.




HID Root A Authorization(s)
HID Root B Authorization(s)
HID Root A Issuing #1
HID Root A Issuing #N
HID Root B Issuing #1
Operational #1
Operational #N
HID Inode A Authorization(s)
HID Inode B Authorization(s)
HID Inode A Issuing #1
HID Inode A Issuing #N
HID Inode B Issuing #1
Operational #1
Operational #N
HID Leaf A Authorization(s)
HID Leaf B Authorization(s)

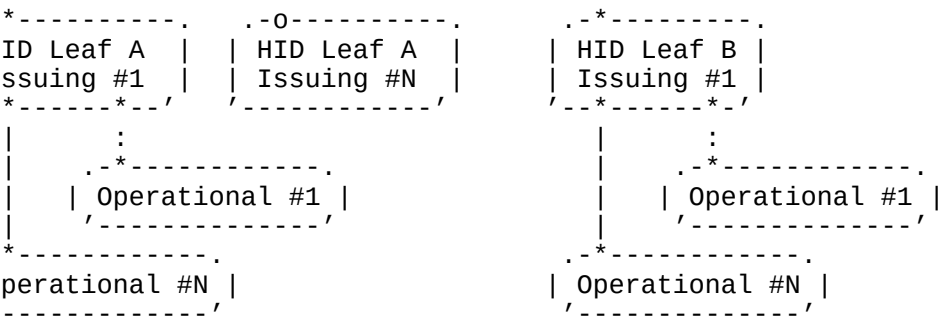


Figure 7: ORCHID Key Infrastructure (OKI) Hierarchy

In Figure 7 the solid connections are REQUIRED and dotted connections are OPTIONAL. A two-level hierarchy will consist of only a Root and Leaf level with no Inodes, shown in Figure 7 with a solid line but open connection point between its ancestor and descendant. Also note that Figure 7 shows two distinct paths in the hierarchy, to illustrate OPTIONAL cross-endorsement, for simplicity.

IPv6 Prefix: This level is not directly part of the OKI but performs the primary function of delegating reverse IPv6 zone(s) associated with Root Hierarchy ID values to respective Apexes or performing DNS delegations on an Apex's behalf.

Apex: An administrative owner of a set of Root Hierarchy ID values. The Apex performs assignment and/or allocation of these values and either delegates management of the reverse IPv6 zone(s) back to the IPv6 Prefix or manages it themselves. The Apex MAY have an Authorization HHIT, with a self-signed endorsement, as part of the function to endorse membership into a hierarchy.

Root: The first level of a Hierarchy ID, with descendant levels set to 0. It MUST have at least one Authorization HHIT to be used to obtain membership into the hierarchy and is either self-signed or endorsed by an Apex. A Root is the administrative entity that assigns/allocates the next level in the hierarchy to a descendant and manages the applicable reverse IPv6 zone(s). The endorsement of a descendant membership into the hierarchy MUST be performed using an Authorization HHIT endorsed by the direct ancestor. To support the endorsement of operational entities under the Root Hierarchy ID, the Root SHOULD use one or more Issuing HHITs that are endorsed by Root Authorization HHITs.

Inode: A middle level node that has both an ancestor and descendant

in the Hierarchy ID. It MUST have at least one Authorization HHIT to be used to obtain membership into the hierarchy and MUST be endorsed by a direct ancestor. An Inode is the administrative entity that assigns/allocates the next level in the hierarchy to a descendant and manages the applicable reverse IPv6 zone(s). To support the endorsement of operational entities under the Inode Hierarchy ID, the Inode SHOULD use one or more Issuing HHITs that are endorsed by Inode Authorization HHITs.

Leaf: The last level of a Hierarchy ID. It MUST have at least one Authorization HHIT to be used to obtain membership into the hierarchy and MUST be endorsed by a direct ancestor. A Leaf node is the administrative entity that registers operational entities and manages the applicable reverse IPv6 zone(s). To support the endorsement of operational entities under the Leaf Hierarchy ID, the Leaf MUST use one or more Issuing HHITs that are endorsed by Leaf Authorization HHITs.

This document allocates two new HHIT Entity Types in each level of hierarchy, one for Authorization and one for Issuing (Section 7.4).

6.1. Hierarchy Responsibilities

An HOME has a number of responsibilities depending on its place in the hierarchy. This section covers the technical responsibilities of each level of an HHIT’s hierarchy. Specific non-technical policies for hierarchy levels are out of scope for this document and are expected to be developed into OKI policy guidance or added to existing PKI policy guidance for each use-case.

Responsibility	Apex	Root	Inode	Leaf
provide using public methods any requirements, policy and/or guidance to prospective descendants	MUST	MUST	MUST	MUST
provide OKIX-compatible certificate(s) as endorsement for registration of	MAY _see *1+ Authorization HHIT*_	MUST	MUST	MUST

descendant(s)					
maintain registrations with relevant Personally Identifiable Information (PII) as required by their jurisdiction	MUST	MUST	MUST	MUST	
fulfill any jurisdiction requirements that are out of scope for this document	MUST	MUST	MUST	MUST	
support [RFC7401] _MAY support services using [RFC8003] such as [RFC8004]_	-	MAY	MAY	MAY	
1+ Authorization HHIT _see Table 2_	MAY	MUST	MUST	MUST	
1+ Issuing HHIT _MUST have same Hierarchy ID as an Authorization HHIT_ _MUST be locally registered using an Authorization HHIT_	MAY	MAY	SHOULD	MUST	
cross-endorse with siblings _MUST use an	-	MAY	MAY	MAY	

Authorization HHIT_					
maintain reverse IPv6 zone(s)	SHOULD _or delegate back to the IPv6 Prefix_	MUST	MUST	MUST	
assign/allocate values in the X level of hierarchy in the Hierarchy ID _see Table 3_	MUST _X=Root_	MUST _X=Inode_	MUST _X=Inode or Leaf_	-	
implement Section 4 and Section 5	MAY _or 501 Not Implemented_	SHOULD _or 501 Not Implemented_	SHOULD _or 501 Not Implemented_	MUST	

Table 1: Hierarchy Responsibilities

1+ Authorization HHIT	Apex	Root	Inode	Leaf
set Hierarchy ID to all zeros	MUST	MUST NOT	MUST NOT	MUST NOT
set Hierarchy ID to assignment/allocation from ancestor	MUST NOT	MUST	MUST	MUST
self register	MUST	if *Apex* MUST then MUST NOT	MUST NOT	MUST NOT
registered to direct ancestor	MUST NOT	if *Apex* MUST NOT then MUST	MUST	MUST

Table 2: Additional Responsibilities for 1+ Authorization HHIT

Assign/allocate values in Hierarchy ID	Apex	Root	Inode	Leaf
verify prospective descendant(s) for hierarchy membership	SHOULD	SHOULD	SHOULD	-
register descendant(s) hierarchy members using an Authorization HHIT	MAY	MUST	MUST	-
delegate corresponding reverse IPv6 zone(s) to descendant(s)	SHOULD _see *maintain reverse IPv6 zone(s)*_	MUST	MUST	-

Table 3: Additional Responsibilities for Assign/allocate values in Hierarchy ID

6.2. ORCHID Key Infrastructure X.509 (OKIX)

Existing Public Key Infrastructure X.509 (PKIX) certificate profiles can be augmented to support HHITs creating ORCHID Key Infrastructure X.509 (OKIX) profiles.

Other X.509 fields and OIDs MAY be required in a given jurisdiction. This is up to the original PKI(X) policy if applicable and is out of scope for this document.

6.2.1. Signing Request

The Certificate Signing Request is mandatory for all HHIT registrations. Depending on the entity being registered, there is specific content rules.

```
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: <subject to policy>
        Subject Public Key Info: <subject to policy>
        Attributes:
        Requested Extensions:
            X509v3 Subject Alternative Name: critical
                IP Address: <subject HHIT>
    Signature Algorithm: <subject to policy>
```

Figure 8: Certificate Signing Request Profile

Subject: As defined in Section 4.1.2.6 of [RFC5280]. This field is filled in based on the HHIT Entity Type being registered and subject to policy of the Certificate Authority.

Subject Alternative Name IP6: 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 HHIT and MUST mark the extension as critical. The HOME MUST check to ensure that the HHIT 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 HOME will use its HID for HHIT generation using the public key provided by this CSR.

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

6.2.2. Certificate: Lite Profile

OKIX-Lite is designed to fully encapsulate an OKI 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 9 and a field matrix found in Table 4.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: <1 byte in size>
        Signature Algorithm: <subject to policy>
        Issuer: CN = <issuer HHIT>
        Validity
            Not Before: <subject to policy>
            Not After : <subject to policy>
        Subject: <subject to policy>
        Subject Public Key Info: <subject to policy>
        X509v3 extensions:
            X509v3 Subject Alternative Name: critical
                IP Address: <subject HHIT>
    Signature Algorithm: <subject to policy>
```

Figure 9: OKIX-Lite Profile

Field	Authorization	Issuing	Operational	Comments
Serial Number	MUST	MUST	MUST	Section 6.2.4.1.1
Subject	MUST	MUST	MUST NOT	Section 6.2.4.2
Issuer	MUST	MUST	MUST	Section 6.2.4.3
Subject Alternative Name IP6	MUST	MUST	MUST	Section 6.2.4.4
Subject Alternative Name URI	MAY	MAY	MAY	Section 6.2.4.5
Basic Constraints	MUST	MUST	MUST NOT	CA=True, flagged as Critical
Subject Key Identifier	MUST NOT	MUST NOT	MUST NOT	-
Authority Key Identifier	MUST NOT	MUST NOT	MUST NOT	-
Key Usage	MAY	MAY	MAY	-
Certificate Authority Policy OIDs	MAY	MAY	MAY	-

Table 4: OKIX-Lite Field Matrix

6.2.3. Certificate: Full Profile

The X.509 certificates are minimalistic (less than 400 bytes for DER). The profile can be found in Figure 10 and a field matrix found in Table 5. This profile MUST be used as the Canonical Registration Certificate in the DNS.

```
   Certificate:
       Data:
           Version: 3 (0x2)
           Serial Number: <20 bytes in size>
           Signature Algorithm: <subject to policy>
           Issuer: CN = <issuer HHIT>
           Validity
               Not Before: <subject to policy>
               Not After : <subject to policy>
           Subject: <subject to policy>
           Subject Public Key Info: <subject to policy>
           X509v3 extensions:
               X509v3 Subject Alternative Name: critical
                   IP Address: <subject HHIT>
                   URI: <subject to policy>
               X509v3 Subject Key Identifier: <subject HHIT>
               X509v3 Authority Key Identifier: <issuer HHIT>
               X509v3 Basic Constraints: critical
               X509v3 Key Usage: critical
       Signature Algorithm: <subject to policy>
```

Figure 10: OKIX-Full Profile

Field	Authorization	Issuing	Operational	Comments
Serial Number	MUST	MUST	MUST	Section 6.2.4.1.2
Subject	MUST	MUST	MUST NOT	Section 6.2.4.2
Issuer	MUST	MUST	MUST	Section 6.2.4.3
Subject Alternative Name IP6	MUST	MUST	MUST	Section 6.2.4.4
Subject Alternative Name URI	MAY	MAY	MAY	Section 6.2.4.5
Basic Constraints	MUST	MUST	MUST NOT	CA=True, flagged as Critical
Subject Key Identifier	MUST	MUST	MUST NOT	Section 6.2.4.6
Authority Key Identifier	MUST	MUST	MUST	Section 6.2.4.7
Key Usage	SHOULD	SHOULD	SHOULD	-
Certificate Authority Policy OIDs	RECOMMENDED	RECOMMENDED	RECOMMENDED	-

Table 5: OKIX-Full Field Matrix

6.2.4. Certificate Fields

The following sections contain clarifications on the usage of fields in the OKIX when they deviate from "standard" X.509 [RFC5280] practice.

6.2.4.1. Serial Number

6.2.4.1.1. Lite

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

However, _Certificate Authority_ 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 _Certificate Authority_s.

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

6.2.4.1.2. Full

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

6.2.4.2. Subject

The Subject field is only used in Authorization and Issuing certificates. For Operational certificates, the Subject MUST be empty and the HHIT will be in Subject Alternative Name (Section 6.2.4.4). In the Subject Alternative Name, the HHIT can be properly encoded as an IPv6 address. The contents of the Subject when used are determined by policy and is out of scope for this document.

6.2.4.3. Issuer

The Issuer MUST be the higher level's HHIT.

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

The Issuer content of its HHIT assists in finding this specific Issuer in the ip6.arpa. DNS tree and any additional information. The exact information that is provide is out of scope for this document.

6.2.4.4. Subject Alternative Name IP6

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

6.2.4.5. Subject Alternative Name URI

This field contains a pointer in the form of a URI where additional information on the _Certificate Authority_ and certificates under its control can be found. The contents MUST be a base URL containing at least the fields scheme, host and port to query for additional information of a HHIT.

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

The RECOMMENDED configuration with respect to Issuing and Operational certificates for a _Certificate Authority_ is to place the Subject Alternative Name URI in the Issuing certificate and exclude them in Operational certificates.

When multiple providers are used by the _Certificate Authority_ to handle additional information there SHOULD be a unique Issuing certificate with Subject Alternative Name URI for each provider, and Operational certificates MUST NOT contain a Subject Alternative Name URI.

Otherwise a _Certificate Authority_ with multiple providers MUST NOT have a Subject Alternative Name URI in the Issuing certificate and MUST set the Subject Alternative Name URI to the specific provider for each Operational certificate.

6.2.4.6. Subject Key Identifier

The Subject Key Identifier MUST be the HHIT. 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 Section 6.2.4.4.

6.2.4.7. Authority Key Identifier

The Authority Key Identifier MUST be the higher level's Subject Key Identifier (i.e. HHIT). 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).

7. IANA Considerations

7.1. Well-Known URIs

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

URI Suffix	Change Controller	Reference	Status	Related Information
home/register	IETF	This RFC	permanent	N/A
home/query	IETF	This RFC	permanent	N/A
home/oaas	IETF	This RFC	permanent	N/A

Table 6: Additions to Well-Known URIs Registry

Author Note: should we also add Well-Known URIs for update and delete methods? This would then cover all parts of CRUD.

7.2. CWT & JWT Claims

HOME Tokens contain a Claim Set compatible with those of CWT and JWT, so the CWT and JWT Claims registries, [IANA.CWT.Claims] and [IANA.JWT.Claims], are reused. No new IANA registry is created.

Per this specification, the following values are requested to be added to the "JSON Web Token Claims" registry established by [RFC7519] and the "CBOR Web Token (CWT) Claims" registry established by [RFC8392]. Each entry is requested to be added to both registries. The "Claim Description", "Change Controller", and "Reference" fields are common and equivalent for the JWT and CWT registries. The "Claim Key" and "Claim Value Type" fields are for

the CWT registry only. The "Claim Name" field is as defined for the CWT registry, not the JWT registry. The "JWT Claim Name" field is equivalent to the "Claim Name" field in the JWT registry.

Claim Name: HOME Registration Inquiry
Claim Description: HOME Registration Inquiry
JWT Claim Name: hri
Claim Key: TBD1
Claim Value Type: array
Change Controller: IETF
Reference: This RFC

Figure 11: HOME Registration Inquiry Claim: Registration Form

Claim Name: HOME Registration Response
Claim Description: HOME Registration Response
JWT Claim Name: hrr
Claim Key: TBD2
Claim Value Type: array
Change Controller: IETF
Reference: This RFC

Figure 12: HOME Registration Response Claim: Registration Form

7.3. RDAP Extensions Registry

IANA is requested to register the following value in the "RDAP Extensions" registry [IANA.RDAP.Extensions].

Extension Identifier:
  home_v0
Registry Operator:
  Any
Specification:
  This specification
Contact:
  IETF <iesg@ietf.org>
Intended Usage:
  This extension is used to convey private information
  stored under a HOME registration.

Figure 13: RDAP Extensions: Registration Form

7.4. HHIT Entity Types

IANA is requested to make the following changes to the "HHIT Entity Types" registry under [IANA.DRIP] to align with Table 7.

Value	HHIT Entity Type	Abbreviation	Operational (Y/N)	Reference
0	Undefined	UKN	N	[RFC9886]
1	Hierarchical ORCHID Management Entity	HOME	N	This RFC
2	Hierarchical ORCHID Management Entity: Authorization	HOME-A	N	This RFC
3	Hierarchical ORCHID Management Entity: Issuing	HOME-I	N	This RFC
4	HOME Apex	APEX	N	[RFC9886]
5	HOME Apex: Authorization	APEX-A	N	This RFC
6	HOME Apex: Issuing	APEX-I	N	This RFC
7	DRIP Identity Management Entity	DIME	N	[RFC9886]
8	DRIP Identity Management Entity: Authorization	DIME-A	N	This RFC
9	DRIP Identity Management Entity: Issuing	DIME-I	N	This RFC
10	Registered Assigning Authority	RAA	N	[RFC9886]
11	Registered	RAA-A	N	This RFC

	Assigning Authority: Authorization				
12	Registered Assigning Authority: Issuing	RAA-I	N	This RFC	
13	HHIT Domain Authority	HDA	N	[RFC9886]	
14	HHIT Domain Authority: Authorization	HDA-A	N	This RFC	
15	HHIT Domain Authority: Issuing	HDA-I	N	This RFC	
16	Unmanned Aircraft	UA	Y	[RFC9886]	
17	Ground Control Station	GCS	Y	[RFC9886]	
18	Unmanned Aircraft System	UAS	Y	[RFC9886]	
19	Remote Identification Module	RID	Y	[RFC9886]	
20	UAS Pilot	PILOT	Y	[RFC9886]	
21	UAS Operator	OP	Y	[RFC9886]	
22	Discovery & Synchronization Service	DSS	Y	[RFC9886]	
23	UAS Service Supplier	USS	Y	[RFC9886]	
24	Network RID Service Provider	DP	Y	[RFC9886]	

25	Network RID Display Provider	SP	Y	[RFC9886]
26	Supplemental Data Service Provider	SDSP	Y	[RFC9886]
27	Crowd Sourced RID Finder	FINDER	Y	[RFC9886]

Table 7: Updated HHIT Entity Type Registry

The above changes include two new fields for "Abbreviation" and "Operational (Y/N)", re-arrangement of the first 15 values and addition of entries for Authorization and Issuing for applicable entity types.

7.5. HOME Parameters

This document requests IANA create a new registry under [IANA.DRIP] called "HOME Parameters" with registry policies described in Table 8.

Range	Registration Policy
Integers less than 0	delegated to HOME Common Parameters registry
Integers greater than -1	Reserved

Table 8: HOME Parameters: Registry Policies

The positive integer values (including zero) MUST NOT be directly allocated. Instead new registries for specific use-case parameters/ keys are created, inherit the HOME Common Parameter registry (Section 7.6) for negative values, and setup new allocation schemes for the positive integers in their scope. See Section 7.7 for an example. Such a new registry SHOULD have an associated Media Type to allow participants a mechanism to explicitly provide context for the parameter set in use.

7.5.1. Registry Fields

Key: Title of the key.

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

Name: 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: A integer value to be used as a key in a CBOR map.

Value Type: CBOR/JSON typing for value under the Key.

Change Controller: TDB

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

7.5.2. Registration Form

```
Key:
Description:
Name:
Label:
Value Type:
Change Controller:
Reference:
```

Figure 14: HOME Parameters: Registration Form

7.5.2.1. Review Criteria

For a Key the specified value of a new registration should not duplicate the syntax and/or semantics of an existing registration. For the Name parameter of the Key, a new registration MUST NOT duplicate an existing registration Name.

Value Type MUST be a valid CBOR type or from [IANA.CBOR.Tags]. Their typing in JSON is assumed to follow the translation from CBOR to JSON following Section 6 of [RFC8949].

7.6. HOME Common Parameters

This document requests IANA create a new registry under [IANA.DRIP] called "HOME Common Parameters" with registry policies described in Table 9. Initial entries for this registry are in Figure 15 and uses the registration form of Section 7.5.2.

Range	Registration Policy
Integers less than -65536	Private Use (Section 4.2 of [RFC8126])
Integers in the range -257 to -65536	Specification Required (Section 4.7 of [RFC8126])
Integers in the range -1 to -256	Standard Action (Section 4.9 of [RFC8126])

Table 9: HOME Common Parameters: Registry Policies

Key: HHIT
Description: Hierarchial Host Identity Tag & Entity Type
Name: hhit
Label: -1
Value Type: array
Change Controller: IETF
Reference: This RFC

Key: VNB
Description: Valid Not Before
Name: vnb
Label: -2
Value Type: tdate / time
Change Controller: IETF
Reference: This RFC

Key: VNA
Description: Valid Not After
Name: vna
Label: -3
Value Type: tdate / time / number
Change Controller: IETF
Reference: This RFC

Key: Publish
Description: Boolean to publish to DNS

```
   Name: publish
   Label: -4
   Value Type: bool
   Change Controller: IETF
   Reference: This RFC

   Key: Certificate Signing Request (CSR)
   Description: DER Encoded X.509 CSR
   Name: csr
   Label: -5
   Value Type: bytes
   Change Controller: IETF
   Reference: This RFC

   Key: Oracle Certificate
   Description: DER Encoded Oracle X.509 Certificate
   Name: oracle_x
   Label: -6
   Value Type: bytes
   Change Controller: IETF
   Reference: This RFC

   Key: Canonical Certificate
   Description: DER Encoded X.509 Certificate
   Name: canon_x
   Label: -7
   Value Type: bytes
   Change Controller: IETF
   Reference: This RFC

   Key: Canonical Certificate Chain
   Description: List of DER Encoded X.509 Certificates
   Name: canon_chain
   Label: -8
   Value Type: array
   Change Controller: IETF
   Reference: This RFC

   Key: Contact
   Description: Base64 Encoded jCard Contact Information
   Name: contact
   Label: -9
   Value Type: text
   Change Controller: IETF
   Reference: This RFC
```

Figure 15: Initial HOME Common Parameters

7.7. HOME Aviation Parameters

This document requests IANA create a new registry under [IANA.DRIP] called "HOME Aviation Parameters" with registry policies described in Table 10. Initial entries for this registry are in Figure 16 and uses the registration form of Section 7.5.2.

When using this registry the "ctx" parameter of Section 7.8 is set to "aviation".

Range	Registration Policy
Integers less than 0	delegated to HOME Common Parameters registry
Integers in the range 0 to 65535	Specification Required (Section 4.7 of [RFC8126])
Integers greater than 65535	First Come First Served (Section 4.3 of [RFC8126])

Table 10: HOME Aviation Parameters: Registry Policies

Key: UAS Type
Description: ASTM F3411 UAS Type Enumeration
Name: uas_type
Label: 0
Value Type: uint
Change Controller: IETF
Reference: Section 5.2 of RFC9886

Key: UAS Identifiers
Description: UAS ID Type & Value Pairs
Name: uas_ids
Label: 1
Value Type: array
Change Controller: IETF
Reference: Section 5.2 of RFC9886

Key: Authentication Data
Description: Authentication Type & Data Pairs
Name: auths
Label: 2
Value Type: array
Change Controller: IETF
Reference: Section 5.2 of RFC9886

```
Key: Self ID
Description: Self Description Type & Value
Name: self_id
Label: 3
Value Type: array
Change Controller: IETF
Reference: Section 5.2 of RFC9886

Key: UAS Classification
Description: UAS Category & Class
Name: classification
Label: 4
Value Type: array
Change Controller: IETF
Reference: Section 5.2 of RFC9886

Key: Area
Description: Count, Radius, Floor & Ceiling
Name: area
Label: 5
Value Type: array
Change Controller: IETF
Reference: Section 5.2 of RFC9886

Key: Operator ID
Description: UAS Pilot Operator ID
Name: operator_id
Label: 6
Value Type: array
Change Controller: IETF
Reference: Section 5.2 of RFC9886
```

Figure 16: Initial HOME Aviation Parameters

7.8. Media Types

IANA is requested to add the entries of Table 11 to the "Media Types" registry [IANA.MediaTypes].

Name	Template	Reference
home+cbor	application/home+cbor	This RFC, Section 7.8.1
home+json	application/home+json	This RFC, Section 7.8.2

Table 11: New Media Types

7.8.1. application/home+cbor

```
Type name:
  application
Subtype name:
  home+cbor
Required parameters:
  N/A
Optional parameters:
  "ctx" (Context as a string for map keys. Case insensitive.)
Encoding considerations:
  binary
Security considerations:
  N/A
Interoperability considerations:
  N/A
Published specification:
  This RFC
Applications that use this media type:
  Entities that exchange CBOR/COSE data as part of HOME interactions.
Fragment identifier considerations:
  N/A
Person & email address to contact for further information:
  DRIP WG mailing list (tmrid@ietf.org)
Intended usage:
  Common
Restrictions on usage:
  None
Author/Change controller:
  IETF
Provisional registration:
  No
```

Figure 17: home+cbor Media Type Registration

7.8.2. application/home+json

```
Type name:
  application
Subtype name:
  home+json
Required parameters:
  N/A
Optional parameters:
  "ctx" (Context as a string for map keys. Case insensitive.)
Encoding considerations:
  same as STD90
Security considerations:
  N/A
Interoperability considerations:
  N/A
Published specification:
  This RFC
Applications that use this media type:
  Entities that exchange JOSE/JSON data as part of HOME interactions.
Fragment identifier considerations:
  N/A
Person & email address to contact for further information:
  DRIP WG mailing list (tmrid@ietf.org)
Intended usage:
  Common
Restrictions on usage:
  None
Author/Change controller:
  IETF
Provisional registration:
  No
```

Figure 18: home+json Media Type Registration

7.9. CoAP Content-Format

IANA is requested to add the entries of Table 12 to the "CoAP Content-Formats" registry within the "Constrained RESTful Environments (CoRE) Parameters" registry group [IANA.CORE].

Content Type	Content Coding	ID	Reference
application/home+cbor; ctx=aviation	-	TBD	This RFC
application/home+json; ctx=aviation	-	TBD	This RFC

Table 12: New CoAP Content-Formats

8. Security Considerations

The considerations discussed in [RFC9153], [RFC9374], [RFC9434] and [RFC9886] apply.

8.1. AAA

OMEs 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 a jurisdiction for devices. Remote ATtestation ProcedureS (RATS, [RFC9334]) is recommended for such mandates. The specific attestation mechanisms are out of scope for this document.

8.2. 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 an HHIT will be derived SHOULD be generated by the entity identified by that HHIT and the corresponding private key should be stored only on that entity’s device. There may be scenarios where other parts of a system, such as a UAS, generates the cryptographic materials and provision them as needed during an operation. Any such system MUST ensure security of the cryptographic material is guaranteed.

9. Privacy & Transparency Considerations

The use of COSE or JOSE encryption provides privacy for _Registrant_ data when communicating with an HOME. The considerations of [RFC9153] apply.

10. References

10.1. Normative References

[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>.

[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>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/rfc/rfc7515>.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <https://www.rfc-editor.org/rfc/rfc7516>.

[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>.

[RFC9164] Richardson, M. and C. Bormann, "Concise Binary Object Representation (CBOR) Tags for IPv4 and IPv6 Addresses and Prefixes", RFC 9164, DOI 10.17487/RFC9164, December 2021, <https://www.rfc-editor.org/rfc/rfc9164>.

[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>.

[RFC9886] Wiethuechter, A., Ed. and J. Reid, "DRIP Entity Tags (DETs) in the Domain Name System", RFC 9886, DOI 10.17487/RFC9886, December 2025, <https://www.rfc-editor.org/rfc/rfc9886>.

[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>.

[STD96] Internet Standard 96, <https://www.rfc-editor.org/info/std96>. At the time of writing, this STD comprises the following:

Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/info/rfc9052>.

Schaad, J., "CBOR Object Signing and Encryption (COSE): Countersignatures", STD 96, RFC 9338, DOI 10.17487/RFC9338, December 2022, <https://www.rfc-editor.org/info/rfc9338>.

10.2. Informative References

[IANA.CBOR.Tags]
IANA, "CBOR Tags", November 2025, <https://www.iana.org/assignments/cbor-tags/cbor-tags/>.

[IANA.CORE]
IANA, "CoAP Content-Formats", July 2025, <https://www.iana.org/assignments/core-parameters>.

[IANA.CWT.Claims]
IANA, "CBOR Web Token (CWT) Claims", November 2025, <https://www.iana.org/assignments/cwt/>.

[IANA.DRIP]
IANA, "Drone Remote ID Protocol", October 2025, <https://www.iana.org/assignments/drip/>.

[IANA.JWT.Claims]
IANA, "JSON Web Token (JWT) Claims", November 2025, <https://www.iana.org/assignments/jwt/>.

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

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

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

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/rfc/rfc4291>.

[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>.

[RFC7343] Laganier, J. and F. Dupont, "An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers Version 2 (ORCHIDv2)", RFC 7343, DOI 10.17487/RFC7343, September 2014, <https://www.rfc-editor.org/rfc/rfc7343>.

[RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <https://www.rfc-editor.org/rfc/rfc7401>.

[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>.

[RFC8003] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Registration Extension", RFC 8003, DOI 10.17487/RFC8003, October 2016, <https://www.rfc-editor.org/rfc/rfc8003>.

[RFC8004] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", RFC 8004, DOI 10.17487/RFC8004, October 2016, <https://www.rfc-editor.org/rfc/rfc8004>.

[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>.

[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>.

[STD69] Internet Standard 69, <https://www.rfc-editor.org/info/std69>.

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

Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <https://www.rfc-editor.org/info/rfc5730>.

Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <https://www.rfc-editor.org/info/rfc5731>.

Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Host Mapping", STD 69, RFC 5732, DOI 10.17487/RFC5732, August 2009, <https://www.rfc-editor.org/info/rfc5732>.

Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, DOI 10.17487/RFC5733, August 2009, <https://www.rfc-editor.org/info/rfc5733>.

Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Transport over TCP", STD 69, RFC 5734, DOI 10.17487/RFC5734, August 2009, <https://www.rfc-editor.org/info/rfc5734>.

[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>.

Appendix A. Oracle As A Service

This appendix is normative.

In certain circumstances field contents may only be properly validated by other entities outside of DRIP. A so called "oracle" can be provided by an RAA or designated 3rd party as a service (i.e. "oracle-as-a-service") or mandated by an RAA to be implemented directly by their HDAs. 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 could result in bogus data being returned.

The rest of this section (and subsections) only applies for an "oracle-as-a-service" model.

A.1. Use Case Example

An an 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. When an "oracle" is not consulted there is a risk that the a recorded binding between UAS and Operator is non-existent, resulting in faulty CAA enforcement capabilities.

A.2. Expected HOME Behavior

HOME’s when presented with additional information that requires an "oracle", MUST forward without modification the Registrants HOME Token for processing to their RAAs "oracle-as-a-service" endpoint. HOME’s acting as an RAA under this specification MUST always provide the "oracle-as-a-service" endpoint (/.well-known/home/oaas) and respond as follows:

* return an applicable HTTP Response error when the "oracle" does not exist within a given jurisdiction
  - OMEs receiving such a response SHOULD proceed as if the "oracle" successfully validated the request
* perform a redirect to appropriate system when "oracle" is being provided as a 3rd party service that the HDAs are expected to independently consult
* process and respond following Appendix A.3

OMEs process non-error HTTP codes by including the returned "oracle" certificate in artifacts in the Private Information Registry and continuing the registration process. An error HTTP code received by a HOME MUST be mimicked back to the _Registrant_.

A.3. Expected Oracle Behavior

The "oracle" MUST validate the HOME Token signature before processing any of the claim data it has authority over. Additional authentication and/or authorization on the endpoint MAY be required by a jurisdiction and its selection, implementation and policy are out of scope for this document.

If an "oracle" accepts the data presented it MUST respond with a valid X.509 certificate, RECOMMENDED to be OKIX compliant, with the Subject set to the same contents as the _Registrant_ CSR. When a "oracle" denies any data presented it MUST respond with an appropriate HTTP Response code and MUST NOT include a reason. This is to avoid clients data mining information to forge malicious requests. Once the "oracle" has completed its validations on fields it has authority over and responded accordingly the HOME Token MUST be deleted from the "oracle".

The policy and other inner mechanics of the "oracle" are out of scope for this document.

Appendix B. CDDL & Encoding Rules

This appendix is normative.

B.1. Prelude

The HOME CDDL Prelude of Figure 19 is built upon the prelude defined in Appendix E of [RFC8610] to allow interoperability with JSON encodings. The conversion of fields to/from CBOR and JSON MUST use the advice in Section 6 of [RFC8949].

```
entity-id = int / text
key-id = int / text
map = { &(int, text) => any }
```

Figure 19: CDDL: HOME Prelude

B.2. HRI & HRR Claims

The full claim set for HRI and HRR, including prelude, are shown in Figure 20.

```
rdap = [
  hhit: [ ip6: text, hhit_entity_type: uint ],
  canon_x: text .b64u bytes,
  metadata: map / null
]

response = [
  success: { &(entity-id): success-data } / null
  failure: [ + error-data ] / null
  shared: map / null
]
success-data = [
  keys: { &(key-id): map },
  shared: map / null
]
error-data = [
  eid: &(entity-id) / null,
  kid: &(key-id) / null,
  cat: any,
  msg: any
]

inquiry = [
  entities: { &(entity-id) => entity },
  metadata: map / null
]
entity = [
  hhit_entity_type: uint,
  keys: { &(key-id): key },
  metadata: map / null
]
key = [
  csr: bytes / text .b64u bytes,
  metadata: map / null
]

entity-id = int / text
key-id = int / text
map = { &(int, text) => any }
```

Figure 20: CDDL: HRI & HRR Claims

B.3. IPv6 Handling

IPv6 addresses MUST be tagged using [RFC9164] when encoded in CBOR. For JSON, IPv6 addresses MUST be an address string as defined in [RFC4291], Section 2.2.

B.4. HOME Common Parameters

```
hhit = [
  ip6: #6.54(bytes .size(16)) / text, ; ip6 addr tstr
  hhit_entity_type: uint
]
vnb = tdate / time
vna = tdate / time / number
publish = bool
csr = bytes / text .b64u bytes ; DER encoded
oracle_x = bytes / text .b64u bytes ; DER encoded
canon_x = bytes / text .b64u bytes ; DER encoded
canon_chain = [ bytes / text .b64u bytes ] ; DER encoded
contact = text ; b64 encoded jCard object
```

Figure 21: CDDL: HOME Common Parameters

B.4.1. VNB & VNA Handling

The inclusion of the valid not before (VNB) and valid not after (VNA) parameters to the metadata map of Section 4.2.1 allows a _Registrant_ to set very specific validity times for their registration.

VNB when present is always an absolute time reference per its typing. VNA is either an absolute time reference (like VNB) or a positive offset in seconds (to be added to the VNB absolute time). Negative values of VNA MUST be considered an encoding error.

> 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.

When VNB is not present in the metadata map the HOME MUST use the current absolute date/time as VNB. The absence of VNA is a signal for the HOME to use a default offset, determined by policy and out of scope for this document, to add to VNB.

As an example of the use of VNB and VNA, a _Registrant_ may set VNB to null and VNA to 3600. The HOME 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.

B.5. HOME Aviation Parameters

These parameters are carried over from Section 5.2 of [RFC9886]. The content in Figure 22 carries the same semantics from [RFC9886] but is more well-defined in syntax of CDDL.

```
uas_type = (uint .le 15) .default 0
uas_ids = [ + [
  id_type: uint .le 15,
  uas_id: bytes .size(20)
] ]
auth = [ + [
  type: uint .le 15,
  data: bytes .size(1..362)
] ]
self_id = [
  type: uint .size(1),
  description: text .size(23)
]
area = [
  count: (uint .size(1)) .default 1,
  radius: number,
  floor: number,
  ceiling: number
]
classification = [
  type: (uint .le 8) .default 0,
  class: (uint .le 15) .default 0,
  category: (uint .le 15) .default 0
]
operator_id = [
  type: uint .size(1),
  id: bytes .size(20)
]
```

Figure 22: CDDL: HOME Aviation Parameters

Contributors

Robert Moskowitz
HTT Consulting
Email: rgm@htt-consulting.com

Robert Moskowitz provided the concept and general structure for the OKI and the specific inclusions for X.509 profiles to support HHITs.

Author’s Address

Adam Wiethuechter
AX Enterprize, LLC
Email: adam.wiethuechter@axenterprize.com