PAWS
Internet-Draft
Intended status: Standards Track
Expires: July 18, 2014

V. Chen, Ed.
Google
S. Das
Applied Communication Sciences
L. Zhu
Huawei
J. Malyar
iconectiv (formerly Telcordia
Interconnection Solutions)
P. McCann
Huawei

January 14, 2014

Protocol to Access White-Space (PAWS) Databases
draft-ietf-paws-protocol-08

Abstract

Portions of the radio spectrum that are allocated to licensees are available for non-interfering use. This available spectrum is called "White Space." Allowing secondary users access to available spectrum "unlocks" existing spectrum to maximize its utilization and to provide opportunities for innovation, resulting in greater overall spectrum utilization.

One approach to manage spectrum sharing uses databases to report spectrum availability to devices. To achieve interoperability among multiple devices and databases, a standardized protocol must be defined and implemented. This document defines such a protocol, the "Protocol to Access White Space (PAWS) Databases".

Status of this Memo

Table of Contents

1. Introduction

This section provides some high level introductory material. Readers are strongly encouraged to read Protocol to Access White-Space (PAWS) Databases: Use Cases and Requirements [RFC6953] for use cases, requirements, and additional background.

A geospatial database can track available spectrum (in accordance with the rules of one or more regulatory domains) and make this information available to devices. This approach shifts the complexity of spectrum-policy conformance out of the device and into the Database. This approach also simplifies adoption of policy changes, limiting updates to a handful of databases, rather than numerous devices. It opens the door for innovations in spectrum management that can incorporate a variety of parameters, including user location and time. In the future, it also can include other parameters, such as user priority, time, signal type and power, spectrum supply and demand, payment or micro-auction bidding, and more.

In providing this service, a database records and updates information necessary to protect primary users -- for example, this information may include parameters such as a fixed transmitter's call sign, its geo-location, antenna height, power, and periods of operation. The rules that the Database must follow, including its schedule for obtaining and updating protection information, protection rules, and information reported to devices, vary according to regulatory domain. Such variations, however, should be handled by each database, and exposure to the variations by devices should be minimized.

This specification defines an extensible protocol to obtain available spectrum from a geospatial database by a device with geo-location capability. It enables a device to operate in any regulatory domain that implements the same protocol and in which the device is authorized to operate. The document describes the use of HTTP/TLS as transport for the protocol.

2. Conventions and Terminology

2.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in Key words for use in RFCs to Indicate Requirement Levels [RFC2119].

2.2. Terminology

Database or Spectrum Database: A database is an entity that contains current information about available spectrum at a given location and time, as well as other types of information related to spectrum availability and usage.

Device ID An identifier for a device.

EIRP: Effective isotropically radiated power

ETSI: European Telecommunications Standards Institute

FCC: Federal Communications Commission

Listing server: A server that provides the URLs for one or more Spectrum Databases. A regulator, for example, may operate a Database Listing Server to publish the list of authorized Spectrum Databases for its regulatory domain.

Master Device: A device that queries the database, on its own behalf and/or on behalf of a slave device, to obtain available spectrum information.

Ruleset: A set of rules that governs operations of white space devices and Spectrum Databases within a regulatory domain. The same set of rules may be used by more than one regulatory domain.

Slave Device: A device that queries the database through a master device.

3. Protocol Overview

A Master Device uses the PAWS protocol to obtain a schedule of available spectrum at its location. The security necessary to ensure the accuracy, privacy, and confidentiality of the Device's location is described in the Security Considerations (Section 10). This document assumes that the Master Device and the Database are connected to the Internet.

A typical sequence of PAWS operations is outlined as follows. See Protocol Functionalities (Section 4) and Protocol Parameters (Section 5) for details:

1. The Master Device obtains (statically or dynamically) the URI for a Database appropriate for its location to send subsequent PAWS messages.
2. The Master Device establishes an HTTPS session with the Database.
3. The Master Device optionally sends an initialization message to the Database to exchange capabilities.
4. If the Database receives an initialization message, it responds with a message in the body of the HTTP response.

5. If required by regulatory domain, the Database registers the Master Device.
6. The Master Device sends an available-spectrum request message to the Database.
7. If required by the regulatory domain, the Master Device must verify with the Database that the Slave Device is valid.
8. The Database responds with an available-spectrum response message in the body of the HTTP response.
9. Depending on regulatory domain requirements and database implementation, the Master Device sends a spectrum-usage notification message to the Database.
10. If the Database receives a spectrum-usage notification message, it responds by sending the Master Device a spectrum-usage acknowledgement message.

3.1. Multi-ruleset Support

For a Master Device that supports multiple rulesets and operates with multiple databases in multiple regulatory domains, the PAWS protocol supports the following sequence of operations for each request by the Master Device:

1. The Master Device includes in its request its location and optionally includes the identifier of all the rulesets it supports and any parameter values it might need for the request
2. The Database uses the device location and also may use the ruleset list to determine its response, for example, to select the list of required parameters
3. If required parameters are missing from the request, the Database responds with a REQUIRED error and a list of names of the missing parameters
4. The Master Device makes the request again, adding the missing parameter values
5. The Database responds to the request, including the identifier of the applicable ruleset
6. The Master Device uses the indicated ruleset to determine how to interpret the Database response

NOTE: Regulatory rules contain many device-only requirements that govern device behavior, independent of any database rules. These requirements may be complex and involve device behavior that are not easily parameterized. The ruleset-id parameter provides a mechanism for the Database to inform the Master Device of the applicable ruleset without having to express device-side behavior within the protocol. The ruleset identifier is a string value that contains the name of regulatory body that established the rules and version information, such as "FccTvBandWhiteSpace-2010".

By separating the regulatory "authority" from the "ruleset-id", it allows the protocol to support multiple regulatory authorities that use the same device-side ruleset. It also allows support for a single authority to define multiple rulesets.

4. Protocol Functionalities

The PAWS protocol consists of several components:

- Database Discovery (Section 4.1) MUST be supported by the Master Device
- Initialization (Section 4.2) MAY be used by the Master Device and MUST be implemented by the Database.
- Device Registration (Section 4.3) MAY be used by the Master Device and MAY be implemented by the Database as a separate component or as part of the Available Spectrum Query (Section 4.4) component.
- Available Spectrum Query (Section 4.4) MUST be supported by Master Device and the Database.
- Spectrum Use Notify (Section 4.4.5) MAY be used by the Master Device and the Database. If the regulatory domain requires notification, it MUST be used by the Master Device and MUST be implemented by the Database.
- Device Validation (Section 4.5) MAY be used by the Master Device. If the regulatory domain requires device validation, it MUST be implemented by the Database and MUST be used by the Master Device.

This section describes the protocol components and their messages. Protocol Parameters (Section 5) contains a more thorough discussion of the parameters that comprise the PAWS request and response messages. Message Encoding (Section 6) provides details of the message encodings. HTTPS Binding (Section 7) describes the use of HTTPS (HTTP Over TLS [RFC2818]) for transporting PAWS messages and optional device authentication.

4.1. Database Discovery

Different regulators may have different requirements for the approval and operation of databases, such as:

- A regulator may only allow a limited number of certified databases to operate. It also may require the certification of each device-to-database pairing.
- A regulator may maintain a trusted website that lists all approved databases, known as the Listing Server. It also may mandate how devices use the listing server.

- A regulator may allow each database to define its own terms of use, so that, for example, an approved device may not be able to access all approved databases.

Prior to sending PAWS messages, a Device MUST determine the URI for a Database that provides service at its current location.

Preconfiguration

The Master Device MAY be provisioned statically with the URI of one or more Databases. For operation in regulatory domains that do not have a listing server, the device SHOULD be provisioned with the URI of all the databases for which it is certified or otherwise permitted to operate. The Device also SHOULD be provisioned with the URI of listing servers approved by regulators.

Configuration Update

To adapt to changes in the list of certified or approved databases, the Device SHOULD update its preconfigured list of databases. If the Master Device retrieves its preconfigured list of databases from a listing server, the device SHOULD check the listing server periodically to update its list.

A Database MAY indicate that its URI will be changing by including the URI of one or more alternate databases (See DbUpdateSpec (Section 5.7)) in its responses to a Device. Before a Database ceases operation, for example, it MUST include DbUpdateSpec in its responses to notify Devices. A Device MUST update its preconfigured list of databases to replace (only) its entry for the responding Database with the URLs of the alternate databases; the list of alternate databases MUST NOT affect any other entries.

Error Handling

The Device SHOULD select another database from its list of preconfigured databases if:

- The selected database is unreachable or does not respond.
- The selected database returns an UNSUPPORTED error (see Error Codes (Section 5.17)), which may indicate that the database does not support the regulatory domain where the device is located.

If a suitable database cannot be contacted, the Device MUST NOT operate under rules for database-managed spectrum. If the Device is already operating when it fails to contact a suitable database, and if the applicable regulatory domain provides a grace period, the Device may continue to operate during such period, but MUST cease use

of the spectrum under rules for database-managed spectrum at or before the expiration of the grace period. If a grace period is not provided by the applicable regulatory domain, an operating Device that fails to contact a suitable database MUST immediately cease use of the spectrum under rules for database-managed spectrum.

4.1.1. Listing Server

Within a regulatory domain that has a Database Listing Server, a Device MUST use it to determine the URLs of databases for the domain. The URI of the Listing Server for a regulatory domain MAY be preconfigured in the device. Where allowed by the regulator, the Device MAY save the database list and SHOULD contact the Database Listing Server periodically to update its list. The time between such updates MUST be no longer than one week, or any update interval required by the applicable regulatory domain, whichever is shorter.

If the Device is unable to contact the Database Listing Server to obtain the list of databases for the domain, the Device MUST NOT operate under rules for database-managed spectrum. If an operating Device attempting to update the available spectrum from a Database fails to contact the Database Listing Server to validate the Database that provides the available spectrum, the operating Device MUST immediately cease use of the spectrum under rules for database-managed spectrum.

The Database Listing request procedure is depicted in Figure 1.

o LISTING_REQ is the database-listing request message
o LISTING_RESP is the database-listing response message

```
               +----------------+    +--------------------+
               | Master Device |    |  Listing Server    |
               +----------------+    +--------------------+
                     |                      |
                     |    LISTING_REQ       |
                     |--------------------->|
                     |                      |
                     |    LISTING_RESP      |
                     |<---------------------|
                     |                      |
```

Figure 1

Specific message formats are defined by the regulators.

4.2. Initialization

A Master Device SHOULD use the initialization procedure to exchange capability information with the Database whenever the Master Device powers up or initiates communication with the Database. The initialization response informs the Master Device of specific regulatory-dependent parameterized-rule values, such as threshold distances and time periods beyond which the Device must update its available-spectrum data (see RuleSetInfo (Section 5.6)). The Master Device MAY manually configure these parameterized-rule values. The initialization message also represents extension points for database implementations or regulatory domains that require the extra handshake.

The Initialization request procedure is depicted in Figure 2.

- o INIT_REQ (Section 4.2.1) is the initialization request message
- o INIT_RESP (Section 4.2.2) is the initialization response message

```
               +---------------+    +-------------------+
               | Master Device |    | Spectrum Database |
               +---------------+    +-------------------+
                     |                     |
                     |    INIT_REQ         |
                     |-------------------->|
                     |                     |
                     |    INIT_RESP        |
                     |<--------------------|
                     |                     |
```

Figure 2

4.2.1. INIT_REQ

The initialization request message allows the Master Device to initiate exchange of capabilities with the Database.

```
+----------------------------------------+
|INIT_REQ                                |
+---------------------------+----------|
|deviceDesc:DeviceDescriptor | required |
|location:GeoLocation        | required |
|.......................................|
|*other:any                 |          |
+---------------------------+----------+
```

Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) for the Device is
   REQUIRED. If the Database does not support the device or any of
   the rulesets specified in the device descriptor, it MUST return an
   error with the UNSUPPORTED (Table 1) code in the error response.
   If the device descriptor does not contain any ruleset IDs, the
   Database SHOULD return a list of RulesetInfo (Section 5.6)
   parameters for each regulatory domain it supports at the specified
   location.

location: The GeoLocation (Section 5.1) for the Device is REQUIRED.

other: Depending on the regulatory domain or database
   implementation, the Master Device MAY specify additional handshake
   parameters in the INIT_REQ message. The Database MUST ignore all
   parameters it does not understand.

4.2.2. INIT_RESP

The initialization response message communicates database parameters to the requesting device.

```
+----------------------------------------+
|INIT_RESP                               |
+----------------------------+-----------+   1..* +-------------+
|rulesetInfos:list           | required  |------->| RulesetInfo |
|databaseChange:DbUpdateSpec | optional  |        +-------------+
|........................................|
|*other:any                  |           |
+----------------------------+-----------+
```

Parameters:

rulesetInfos: A list of RulesetInfo (Section 5.6) parameters MUST be
   included in the response. Each RulesetInfo parameter corresponds
   to a regulatory domain supported by the Database and is applicable
   to the location specified in the INIT_REQ (Section 4.2.1) message.
   If the Device included a list of ruleset IDs in the
   DeviceDescriptor parameter of its INIT_REQ message, each
   RulesetInfo parameter in the response MUST match one of the
   specified ruleset IDs.

databaseChange: The Database MAY include a DbUpdateSpec
   (Section 5.7) parameter to notify the Device of a change to the
   Database URI, providing one or more alternate database URLs. The
   Device SHOULD use the information to update its list of
   preconfigured databases to replace (only) its entry for the
   responding database with the list of alternate URLs.

other: Depending on the regulatory domain or database implementation, the Database MAY include additional handshake parameters in the INIT_RESP (Section 4.2.2) message. The Master Device MUST ignore all parameters it does not understand.

4.3. Device Registration

When a regulatory domain requires registration of a Master Device, the Device MUST send its registration information to the Database to establish certain operational parameters. FCC rules, for example, require that a 'Fixed Device' MUST register its owner and operator contact information, its device identifier, its location, and its antenna height.

The Database MAY support device registration as a separate Device Registration component, or as part of the Spectrum Availability component. If the Database does not support a separate Device Registration request, it MUST return an error with the UNIMPLEMENTED (Table 1) code in the error-response message.

The Device Registration request procedure is depicted in Figure 3.

- o REGISTRATION_REQ (Section 4.3.1) is the device-registration request message
- o REGISTRATION_RESP (Section 4.3.2) is the device-registration response message

```
      +---------------+        +--------------------+
      | Master Device |        | Spectrum Database |
      +---------------+        +--------------------+
            |                         |
            |    REGISTRATION_REQ     |
            |------------------------>|
            |                         |
            |    REGISTRATION_RESP    |
            |<------------------------|
            |                         |
```

Figure 3

4.3.1. REGISTRATION_REQ

The registration request message contains the required registration parameters.

```
+---------------------------------------------------------------+
|REGISTRATION_REQ                                               |
+-------------------------------+-------------------------------+
|deviceDesc:DeviceDescriptor    | required                      |
|location:GeoLocation           | required                      |
|deviceOwner:DeviceOwner        | required                      |
|antenna:AntennaCharacteristics | depends on regulatory domain  |
|...............................................................|
|*other:any                     |                               |
+-------------------------------+-------------------------------+
```

Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) for the Device is REQUIRED. The ruleset IDs included in the this parameter indicate the regulatory domains for which the Device wishes to register. If the registration information is unacceptable for all of the regulatory domains supported by the Database, the Database MUST return an error message with an appropriate error code. Otherwise, the Database MUST return, in its response, a list of RulesetInfo (Section 5.6) parameters for all regulatory domains for which device registration was accepted.

location: The GeoLocation (Section 5.1) for the Device is REQUIRED.

deviceOwner: The DeviceOwner (Section 5.5) information is REQUIRED.

antenna: Depending on the regulatory domain, the AntennaCharacteristics (Section 5.3) MAY be required.

other: Regulatory domains and database implementations MAY require additional registration parameters. To simplify its registration logic, the Device MAY send a union of the registration information required by all supported regulatory domains. The Database MUST ignore all parameters it does not understand. Consult the PAWS Parameters Registry (Section 9.1) for possible additional parameters.

4.3.2. REGISTRATION_RESP

The registration response message acknowledges successful registration by including a RulesetInfo message for each regulatory domain in which the registration is accepted. If the Database accepts the registration for none of the regulatory domains it supports, the Database MUST return an error.

```
+----------------------------------------+
|REGISTRATION_RESP                       |
+----------------------------+-----------+   1..* +-------------+
|rulesetInfos:list           | required  |------->| RulesetInfo |
|databaseChange:DbUpdateSpec | optional  |        +-------------+
|............................|...........|
|*other:any                  |           |
+----------------------------+-----------+
```

Parameters:

rulesetInfos: A list of RulesetInfo (Section 5.6) parameters MUST be included in the response, each of which corresponds to a regulatory domain for which the registration was accepted. The list MUST contain at least one entry.

databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) parameter to notify the Device of a change to the Database URI, providing one or more alternate database URLs. The Device SHOULD use the information to update its list of preconfigured databases to replace (only) its entry for the responding database with the list of alternate URLs.

other: Database implementations MAY return additional parameters in the registration response. Consult the PAWS Parameters Registry (Section 9.1) for possible additional parameters and requirements they place on the Device.

4.4. Available Spectrum Query

To obtain the available spectrum from the Database, a Master Device sends a request that contains its geo-location and any parameters required by the regulatory rules (such as device identifier, capabilities, and characteristics). The Database returns a response that describes which frequencies are available, at what permissible operating power levels, and a schedule of when they are available.

The Available Spectrum Query procedure is depicted in Figure 4.

- o AVAIL_SPECTRUM_REQ (Section 4.4.1) is the available-spectrum request message
- o AVAIL_SPECTRUM_RESP (Section 4.4.2) is the available-spectrum response message
- o AVAIL_SPECTRUM_BATCH_REQ (Section 4.4.3) is an OPTIONAL batch version of the available-spectrum request message that allows multiple locations to be specified in the request
- o AVAIL_SPECTRUM_BATCH_RESP (Section 4.4.4) is the response message for the batch version of the available-spectrum request that contains available spectrum for each location

- o SPECTRUM_USE_NOTIFY (Section 4.4.5) is the spectrum-usage notification message
- o SPECTRUM_USE_RESP (Section 4.4.6) is the spectrum-usage acknowledgment message

```
               +---------------+          +-------------------+
               | Master Device |          | Spectrum Database |
               +---------------+          +-------------------+
                     |                            |
                     |    AVAIL_SPECTRUM_REQ      |
                     | (AVAIL_SPECTRUM_BATCH_REQ) |
                     |--------------------------->|
                     |                            |
                     |    AVAIL_SPECTRUM_RESP     |
                     | (AVAIL_SPECTRUM_BATCH_RESP)|
                     |<---------------------------|
                     |                            |
                     |   (SPECTRUM_USE_NOTIFY)    |
                     |--------------------------->|
                     |                            |
                     |    (SPECTRUM_USE_RESP)     |
                     |<---------------------------|
                     |                            |
```

Figure 4

1. First, the Master Device sends an available-spectrum request message to the Database.
2. The Database MUST respond with an error using the NOT_REGISTERED (Table 1) code if:
   * registration information is required, and
   * the request does not include registration information, and
   * the Device has not previously registered with the Database
3. If the location specified in the request is outside the regulatory domain supported by the Database, the Database MUST respond with an OUTSIDE_COVERAGE (Table 1) error. If some locations within a batch request are outside the regulatory domain supported by the Database, the Database MAY return an OK response with available spectrum for only the valid locations; otherwise, if all locations within a batch request are outside the regulatory domain, the Database MUST respond with an OUTSIDE_COVERAGE error.
4. The Database MAY perform other validation of the request, (e.g., checking for missing required parameters, authorizations). It MUST return an error with appropriate error code (Table 1), if validation fails. If the request is missing required parameters, the Database MUST respond with a REQUIRED (Table 1) error and SHOULD include a list of the missing parameters.

5. If the request is valid, the Database responds with an available-spectrum response message. If the regulatory domain requires that devices must report anticipated spectrum usage, the Database MUST indicate so in the response message.
6. If the available-spectrum response indicates that the Master Device must send a spectrum-usage notification message, the Master Device MUST send the notification message to the Database.
7. If the Database receives a spectrum-usage notification message, it MUST send a spectrum-usage acknowledgment message to the Master Device.

The procedure for asking for available spectrum on behalf of a Slave Device is similar, except that the process is initiated by the Slave Device. The device identifier, capabilities, and characteristics communicated in the AVAIL_SPECTRUM_REQ message MUST be those of the Slave Device, but the location MUST be that of the Master Device. Although the communication and protocol between the Slave Device and Master Device is outside the scope of this document, the expected message sequence is shown in Figure 5.

```
      +------------+     +---------------+      +-------------------+
      |Slave Device|     | Master Device |      | Spectrum Database |
      +------------+     +---------------+      +-------------------+
          |                 |                           |
          | AVAIL_SPEC_REQ  |                           |
          |................>|                           |
          |                 |                           |
          |                 |    AVAIL_SPECTRUM_REQ     |
          |                 |-------------------------->|
          |                 |                           |
          |                 |    AVAIL_SPECTRUM_RESP    |
          |                 |<--------------------------|
          | AVAIL_SPEC_RESP |                           |
          |<................|                           |
          |                 |                           |
          | (SPECTRUM_USE)  |                           |
          |................>|   (SPECTRUM_USE_NOTIFY)   |
          |                 |-------------------------->|
          |                 |                           |
          |                 |    (SPECTRUM_USE_RESP)    |
          |                 |<--------------------------|
          |                 |                           |
```

Figure 5

4.4.1. AVAIL_SPECTRUM_REQ

The request message for the Available Spectrum Query protocol MUST include the Device's geo-location. If allowed by the regulatory domain, the location MAY be an anticipated location.

AVAIL_SPECTRUM_REQ	
deviceDesc:DeviceDescriptor	optional
location:GeoLocation	optional
owner:DeviceOwner	depends on regulatory domain
antenna:AntennaCharacteristics	depends on regulatory domain
capabilities:DeviceCapabilities	optional
masterDeviceDesc:DeviceDescriptor	optional
masterDeviceLocation:GeoLocation	optional
requestType:string	optional
....	....
*other:any	

Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) for the Device requesting available spectrum. When the request is made by a Master Device on its own behalf, the descriptor is that of the Master Device and it is REQUIRED. When the request is made on behalf of a Slave Device, the descriptor is that of the Slave Device, and it is REQUIRED if the "requestType" parameter is not specified. The deviceDesc parameter may be OPTIONAL for some values of requestType.

location: The GeoLocation (Section 5.1) for the Device requesting available spectrum. The location SHOULD be the current location of the Device, but more precisely, the location of the radiation center of the Device's antenna. When the request is made by the Master Device on its own behalf, the location is that of the Master Device and it is REQUIRED. When the request is made by the Master Device on behalf of a Slave Device, the location is that of the Slave Device and it is OPTIONAL (see also the masterDeviceLocation parameter). Depending on the regulatory domain, the location MAY be an anticipated position of the Device to support mobile devices. If the location specifies a region, rather than a point, the Database MAY return an error with the UNIMPLEMENTED (Table 1) code, if it does not support query by region.

owner: Depending on the device type and regulatory domain, the DeviceOwner (Section 5.5) information MAY be included to register the Device with the Database. This enables the Device to register and get spectrum-availability information in a single request.

antenna: Depending on the device type and regulatory domain, the AntennaCharacteristics (Section 5.3) MAY be required.

capabilities: The Master Device MAY include its DeviceCapabilities (Section 5.4) to limit the available-spectrum response to the spectrum that is compatible with its capabilities. The Database SHOULD NOT return spectrum that is not compatible with the specified capabilities.

masterDeviceDesc: Depending on regulatory rules and Database implementations, when the request is made by the Master Device on behalf of a Slave Device, the Master Device MAY be required to provide its own descriptor.

masterDeviceLocation: Depending on regulatory rules, when the request is made by the Master Device on behalf of a Slave Device, the GeoLocation (Section 5.1) of the Master Device MAY be required.

requestType: The request type is an OPTIONAL parameter that may be used to modify the request, but its use depends on applicable regulatory rules. The request type may be used, for example, to request generic Slave Device parameters without having to specify the device descriptor for a specific device. When the requestType parameter is missing, the request is for a specific device (Master or Slave), so the deviceDesc parameter is REQUIRED. See IANA Ruleset Registry, Initial Registry Contents (Section 9.2.2) for regulatory specifics.

other: Regulatory domains and database implementations MAY require additional request parameters. The Database MUST ignore all parameters it does not understand. Consult the PAWS Parameters Registry (Section 9.1) for possible additional parameters.

4.4.2. AVAIL_SPECTRUM_RESP

The response message for the Available Spectrum Query contains one or more SpectrumSpec (Section 5.9) elements, one for each regulatory domain supported at the location specified in the corresponding AVAIL_SPECTRUM_REQ (Section 4.4.1) request. Each SpectrumSpec element contains a list of one or more spectrum schedules, representing permissible power levels over time:

o Within each list of schedules, event-time intervals MUST be disjoint and SHOULD be sorted in increasing time

o A gap in the time schedule means no spectrum is available for that time interval

```
   +----------------------------------------+
   |AVAIL_SPECTRUM_RESP                     |
   +----------------------------+----------+
   |timestamp:string            | required |
   |deviceDesc:DeviceDescriptor | required |
   |spectrumSpecs:list          | required |-------+
   |............................|..........|       |
   |databaseChange:DbUpdateSpec | optional |       |
   |*other:any                  |          |       |
   +----------------------------+----------+       | 1..*
                                                   V
                            +---------------------------------+
                            |SpectrumSpec                     |
                            +-----------------------+----------+
                            |rulesetInfo:RulesetInfo | required |
                            |spectrumSchedules:list  | required |-+
                            |timeRange:EventTime     | optional | |
                            |frequencyRanges:list    | optional | |
                            |needsSpectrumReport:bool| optional | |
                            |maxTotalBwHz:float      | optional | |
                            |maxContiguousBwHz:float | optional | |
                            +-----------------------+----------+ |
                                             +--------------------+
                                             | 1..*
                                             V
                                +-------------------------------+
                                |SpectrumSchedule               |
                                +--------------------+----------+
                                |eventTime:EventTime | required |
                                |spectra:list        | required |
                                +--------------------+----------+
```

Parameters:

timestamp: Timestamp of the response of the form, YYYY-MM-DDThh:mm:ssZ, as defined by Date and Time on the Internet: Timestamps [RFC3339]. This SHOULD be used by the Device as a reference for the start and stop times in the spectrum schedules.

deviceDesc: The Database MUST include the DeviceDescriptor (Section 5.2) specified in the AVAIL_SPECTRUM_REQ message.

spectrumSpecs: The SpectrumSpec (Section 5.9) list is REQUIRED and MUST include at least one entry. Each entry contains the schedules of available spectrum for a regulatory domain. The Database MAY return more than one SpectrumSpec to represent available spectrum for multiple regulatory domains at the specified location.

databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) parameter to notify the Device of a change to the Database URI, providing one or more alternate database URLs. The Device SHOULD use the information to update its list of preconfigured databases to replace (only) its entry for the responding database with the list of alternate URLs.

other: Database implementations MAY return additional parameters in the response. Consult the PAWS Parameters Registry (Section 9.1) for possible additional parameters and requirements they place on the Device.

4.4.2.1. Update Requirements

When the stop time specified in the schedule has been reached, the Device:

- o MUST obtain a new spectrum-availability schedule, either by using the next one in the list (if provided) or making another Available Spectrum Query (Section 4.4)
- o If the new schedule indicates the in-use spectrum is no longer available, the Device MUST immediately cease use of that spectrum under rules for database-managed spectrum.
- o If the Device is unable to contact the Database to obtain a new schedule, the Device MUST immediately cease use of that spectrum under rules for database-managed spectrum.

When the Device moves beyond a threshold distance (established by regulatory rules) away from the actual location and all anticipated location(s) it reported in previous AVAIL_SPECTRUM_REQ or AVAIL_SPECTRUM_BATCH_REQ requests (see "maxLocationChange" in RulesetInfo (Section 5.6)), it:

- o MUST obtain a new spectrum-availability schedule by making another Available Spectrum Query (Section 4.4).
- o If the new response indicates the in-use spectrum is no longer available, the Device MUST stop operation immediately.
- o If the Device is unable to contact the Database to obtain a new schedule, depending on the regulatory domain, the Device MUST immediately cease use of the spectrum under rules of database-managed spectrum.

NOTE: Regulatory rules govern whether a device may request and use spectrum at anticipated locations beyond the threshold distance from its current location.

4.4.3. AVAIL_SPECTRUM_BATCH_REQ

The Database MAY support the batch request that allows multiple locations to be specified. This allows a portable Master Device to get available spectrum for a sequence of anticipated locations using

a single request. The Database MUST interpret each location in the batch request as if it were an independent request and MUST return results consistent with multiple individual AVAIL_SPECTRUM_REQ (Section 4.4.1) requests. The request message for the batch Available Spectrum Query protocol MUST include at least one GeoLocation (Section 5.1). If the Database does not support batch requests, it MUST return a UNIMPLEMENTED (Table 1) error.

```
 +------------------------------------------------------------------+
 |AVAIL_SPECTRUM_BATCH_REQ                                          |
 +----------------------------------+-------------------------------+
 |deviceDesc:DeviceDescriptor       | optional                      |
 |locations:list                    | required                      |--+
 |owner:DeviceOwner                 | depends on regulatory domain |  |
 |antenna:AntennaCharacteristics    | depends on regulatory domain |  |
 |capabilities:DeviceCapabilities   | optional                      |  |
 |masterDeviceDesc:DeviceDescriptor| optional                      |  |
 |masterDeviceLocation:GeoLocation  | optional                      |  |
 |requestType:string                | optional                      |  |
 +..................................+...............................+  |
 |*other:any                        |                               |  |
 +----------------------------------+-------------------------------+  |
                                                                       |
                                                                1..* V
                                                         +-------------+
                                                         | GeoLocation |
                                                         +-------------+
```

Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) for the Device requesting available spectrum. When the request is made by a Master Device on its own behalf, the descriptor is that of the Master Device and it is REQUIRED. When the request is made on behalf of a Slave Device, the descriptor is that of the Slave Device, and it is REQUIRED if the "requestType" parameter is not specified. The deviceDesc parameter may be OPTIONAL for some values of requestType.

locations: The GeoLocation (Section 5.1) list for the Device is REQUIRED. This allows the Device to specify its actual location plus additional anticipated locations, when allowed by the regulatory domain. At least one location MUST be included. This specification places no upper limit on the number of locations, but the Database MAY restrict the number of locations it supports by returning a response with fewer locations than specified in the request. If the locations specify regions, rather than points, the Database MAY return an error with the UNIMPLEMENTED (Table 1) code, if it does not support query by region. When the request is

made by a Master Device on its own behalf, the locations are those of the Master Device. When the request is made by the Master Device on behalf of a Slave Device, the locations are those of the Slave Device (see also the masterDeviceLocation parameter).

owner: Depending on the device type and regulatory domain, the DeviceOwner (Section 5.5) information MAY be included to register the Device with the Database. This enables the Device to register and get spectrum-availability information in a single request.

antenna: Depending on the device type and regulatory domain, the AntennaCharacteristics (Section 5.3) MAY be required.

capabilities: The Master Device MAY include its DeviceCapabilities (Section 5.4) to limit the available-spectrum response to the spectrum that is compatible with its capabilities. The Database SHOULD NOT return spectrum that is not compatible with the specified capabilities.

masterDeviceDesc: Depending on regulatory rules, when the request is made by the Master Device on behalf of a Slave Device, the Master Device MAY BE REQUIRED to provide its own descriptor.

masterDeviceLocation: Depending on regulatory rules and Database implementations, when the request is made by the Master Device on behalf of a Slave Device, the GeoLocation (Section 5.1) for the Master Device MAY be required.

requestType: The request type is an OPTIONAL parameter that may be used to modify the request, but its use depends on applicable regulatory rules. The request type may be used, for example, to request generic Slave Device parameters without having to specify the device descriptor for a specific device. When the requestType parameter is missing, the request is for a specific device (Master or Slave), so the deviceDesc parameter is REQUIRED. See IANA Ruleset Registry, Initial Registry Contents (Section 9.2.2) for regulatory specifics.

other: Regulatory domains and database implementations MAY require additional request parameters. The Database MUST ignore all parameters it does not understand. Consult the PAWS Parameters Registry (Section 9.1) for possible additional parameters.

4.4.4. AVAIL_SPECTRUM_BATCH_RESP

The response message for the batch Available Spectrum Query contains a schedule of available spectrum for the Device at multiple locations.

```
   +---------------------------------------+
   |AVAIL_SPECTRUM_BATCH_RESP              |
   +-----------------------------+---------+
   |timestamp:string             | required |
   |deviceDesc:DeviceDescriptor  | required |
   |geoSpectrumSpecs:list        | required |-------+
   |.............................|..........|       |
   |databaseChange:DbUpdateSpec  | optional |       |
   |*other:any                   |          |       |
   +-----------------------------+----------+       | 0..*
                                                    V
                             +-----------------------------------+
                             |GeoSpectrumSpec                    |
                             +---------------------+-------------+
                             |location:GeoLocation  | required |
                             |spectrumSpecs:list    | required |
                             +---------------------+-------------+
```

Parameters:

timestamp: Timestamp of the response of the form, YYYY-MM-DDThh:mm:ssZ, as defined by Date and Time on the Internet: Timestamps [RFC3339]. This SHOULD be used by the Device as a reference for the start and stop times in the spectrum schedules.

deviceDesc: The Database MUST include the DeviceDescriptor (Section 5.2) specified in the AVAIL_SPECTRUM_REQ message.

geoSpectrumSpecs: The geoSpectrumSpecs (Section 5.15) list is REQUIRED (though it MAY be empty if spectrum is unavailable). For each location, the Database MAY return one or more SpectrumSpec (Section 5.9) parameters to represent available spectrum for one or more regulatory domains. The Database MAY return available spectrum for fewer locations than requested. The Device MUST NOT make any assumptions on the order of the entries in the list and MUST use the location value in each GeoSpectrumSpec entry to match available spectrum to a location.

databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) parameter to notify the Device of a change to the Database URI, providing one or more alternate database URLs. The Device SHOULD use the information to update its list of preconfigured databases to replace (only) its entry for the responding database with the list of alternate URLs.

other: Database implementations MAY return additional parameters in the response. Consult the PAWS Parameters Registry (Section 9.1) for possible additional parameters and requirements they place on the Device.

See Update Requirements (Section 4.4.2.1) for when the Device must update its available spectrum data.

4.4.5. SPECTRUM_USE_NOTIFY

The spectrum-use notification message MUST contain the geo-location of the Device and parameters required by the regulatory domain.

```
   +----------------------------------------------+
   |SPECTRUM_USE_NOTIFY                           |
   +----------------------------------+-----------+
   |deviceDesc:DeviceDescriptor       | required  |
   |location:GeoLocation              | required  |
   |masterDeviceLocation:GeoLocation  | optional  |
   |spectra:list                      | required  |-------+
   |..............................................|       |
   |*other:any                        |           |       |
   +----------------------------------+-----------+       | 0..*
                                                          V
                              +---------------------------------+
                              |Spectrum                         |
                              +---------------------+-----------+
                              |resolutionBwHz:float | required  |
                              |profiles:list        | required  |
                              +---------------------+-----------+
```

Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) for the Device is REQUIRED.

location: The GeoLocation (Section 5.1) for the Device. When the notification is made by a Master Device on its own behalf, the location is that of the Master Device and is REQUIRED. When the notification is made by a Master Device on behalf of a Slave Device, the location is that of the Slave Device and MAY be required, depending on regulatory rules.

spectra: The Spectrum (Section 5.11) list is REQUIRED, and specifies the spectrum anticipated to be used by the Device, which includes profiles of frequencies and power levels. The list MAY be empty, if the Device decides not to use any spectrum. For consistency, the resolution bandwidth value, "resolutionBwHz" SHOULD match that from one of the Spectrum (Section 5.11) elements in the corresponding AVAIL_SPECTRUM_RESP message, and the maximum power levels in the Spectrum element MUST be expressed as power over the specified "resolutionBwHz" value. The actual bandwidth to be used (as computed from the start and stop frequencies) MAY be different from the "resolutionBwHz" value. As an example, when regulatory rules express maximum power spectral density in terms of maximum power over any 100 kHz band, then the "resolutionBwHz" value should be set to 100 kHz, even though the actual bandwidth used can be 20 kHz.

masterDeviceLocation: Depending on regulatory rules, when the notification is made by the Master Device on behalf of a Slave Device, the GeoLocation (Section 5.1) for the Master Device MAY be required.

other: Depending on the regulatory domain, other parameters MAY be required. To simplify its logic, the Device MAY include the union of all parameters required by all supported regulatory domains. The Database MUST ignore all parameters it does not understand.

4.4.6. SPECTRUM_USE_RESP

The spectrum-use response message simply acknowledges receipt of the notification.

```
+--------------------------------------+
|SPECTRUM_USE_RESP                     |
+-----------------------------+--------+
|databaseChange:DbUpdateSpec | optional |
|......................................|
|*other:any                  |          |
+-----------------------------+--------+
```

Parameters:

databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) parameter to notify the Device of a change to the Database URI, providing one or more alternate database URLs. The Device SHOULD use the information to update its list of preconfigured databases to replace (only) its entry for the responding database with the list of alternate URLs.

other: Database implementations MAY return additional parameters in the response. Consult the PAWS Parameters Registry (Section 9.1) for possible additional parameters and requirements they place on the Device.

4.5. Device Validation

Typically, a Slave Device needs a Master Device to ask the Database on its behalf for available spectrum. Depending on the regulatory domain, the Master Device also must validate with the Database that the Slave Device is permitted to operate. When regulatory rules allow a Master Device to "cache" the available spectrum for a period of time, the Master Device MAY use the simpler Device Validation component, instead of the full Available Spectrum Query component, to validate a Slave Device.

When validating one or more Slave Devices, the Master Device sends the Database a request that includes the device identifier -- and any

other parameters required by the regulatory rules -- for each Slave Device. The Database MUST return a response that indicates whether each device is permitted to use the spectrum.

A typical sequence for using the Device Validation request is illustrated in Figure 6, where the Master Device already has a valid set of available spectrum for Slave Devices. Note that the communication and protocol between the Slave Device and Master Device is outside the scope of this document.

o DEV_VALID_REQ (Section 4.5.1) is the device-validation request message
o DEV_VALID_RESP (Section 4.5.2) is the device-validation response message

```
      +------------+     +---------------+      +-------------------+
      |Slave Device|     | Master Device |      | Spectrum Database |
      +------------+     +---------------+      +-------------------+
          |                 |                           |
          | AVAIL_SPEC_REQ  |                           |
          |................>|                           |
          |                 |                           |
          |                 |    DEV_VALID_REQ          |
          |                 |-------------------------->|
          |                 |                           |
          |                 |    DEV_VALID_RESP         |
          |                 |<--------------------------|
          | AVAIL_SPEC_RESP |                           |
          |<................|                           |
          |                 |                           |
          | (SPECTRUM_USE)  |                           |
          |................>|   (SPECTRUM_USE_NOTIFY)   |
          |                 |-------------------------->|
          |                 |                           |
          |                 |    (SPECTRUM_USE_RESP)    |
          |                 |<--------------------------|
```

Figure 6

4.5.1. DEV_VALID_REQ

```
   +-----------------------------------------------+
   |DEV_VALID_REQ                                  |
   +-----------------------------------+-----------+
   |deviceDescs:list                   | required  |---+
   |masterDeviceDesc:DeviceDescriptor  | optional  |   |
   +-----------------------------------+-----------+   |
                                                       V 1..*
                                    +-----------------------+
                                    |DeviceDescriptor       |
                                    +-----------------------+
```

Parameters:

deviceDescs: A DeviceDescriptor (Section 5.2) list is REQUIRED, which specifies the list of Slave Devices that to be validated.

masterDeviceDesc: Depending on regulatory rules, when the request is made by the Master Device on behalf of a Slave Device, the Master Device MAY be required to provide its own descriptor.

4.5.2. DEV_VALID_RESP

```
   +-----------------------------------------+
   |DEV_VALID_RESP                           |
   +-----------------------------+-----------+
   |deviceValidities:list        | required  |----
   |databaseChange:DbUpdateSpac  | optional  |    |
   +-----------------------------+-----------+    |
                                                  V 1..*
                            +-----------------------------------------+
                            |DeviceValidity                           |
                            +-----------------------------+-----------+
                            |deviceDesc:DeviceDescriptor  | required  |
                            |isValid:boolean              | required  |
                            +-----------------------------+-----------+
```

Parameters:

deviceValidities: A DeviceValidities (Section 5.16) list is REQUIRED to report the list of Slave Devices and whether each listed Device is valid. The number of entries MUST match the number of DeviceDescriptors (Section 5.2) listed in the DEV_VALID_REQ message.

databaseChange: The Database MAY include a DbUpdateSpec (Section 5.7) parameter to notify the Device of a change to the Database URI, providing one or more alternate database URLs. The Device SHOULD use the information to update its list of preconfigured databases to replace (only) its entry for the responding database with the list of alternate URLs.

5. Protocol Parameters

This section presents more details of the parameters that make up the PAWS request and response messages. It also includes a sub-section defining response codes.

5.1. GeoLocation

This parameter is used to specify the geo-location of the Device. It may be used to specify one of the following:

- o A single point with optional uncertainty
- o A region described by a polygon

These are represented using geometric shapes defined in Section 5 of GEOPRIV Presence Information Data Format Location Object [RFC5491], where:

- o A "point" with uncertainty is represented using the Ellipse shape
- o A region is represented using the Polygon shape

The coordinates are expressed using the WGS84 datum [WGS-84], and units are degrees or meters. The parameter MAY also include a confidence level, expressed as a percentage. The data model for GeoLocation is illustrated below:

```
+-------------------------------------------------+
|GeoLocation                                      |
+------------------+------------------------------+
|point:Ellipse     | optional                     |
|region:Polygon    | optional                     |
|confidence:int    | depends on regulatory domain |
+------------------+------------------------------+
```

Note: point and region are mutually exclusive, but one of them must be present.

```
   +----------------------------------------------------+
   |Ellipse                                             |
   +--------------------+-------------------------------+
   |center:Point        | required                      |--+
   |semiMajorAxis:float | depends on regulatory domain |  |
   |semiMinorAxis:float | depends on regulatory domain |  |
   |orientation:float   | optional                      |  |
   +--------------------+-------------------------------+  v
                                      +-----------------------------+
                                      |Point                        |
                                      +----------------+------------+
                                      |latitude:float  | required |
                                      |longitude:float | required |
                                      +----------------+------------+


   +--------------------------------+
   |Polygon                         |
   +-------------------+------------+  4..* +-----------------------------+
   |exterior:list      | required   |------>|Point                        |
   +-------------------+------------+       +----------------+------------+
                                            |latitude:float  | required |
                                            |longitude:float | required |
                                            +----------------+------------+
```

Parameters:

point: If present, it indicates that the GeoLocation represents a point. Paradoxically, a "point" is parameterized using an Ellipse, where the center represents the location of the point and the distances along the major and minor axes represent the uncertainty. The uncertainty values MAY be required, depending on the regulatory domain. Exactly one of "point" or "region" MUST be present.

region: If present, it indicates that the GeoLocation represents a region. Exactly one of "point" or "region" MUST be present.

center: The center refers to the location of a GeoLocation point and is represented as the center of an ellipse.

latitude, longitude: Floating-point numbers that express the latitude and longitude in degrees using the WGS84 datum [WGS-84].

semiMajorAxis, semiMinorAxis: If required by the regulatory domain, the location uncertainty, in meters, is parameterized using distances along the major and minor axes of the ellipse. When uncertainty is optional, the default value of each is 0.

orientation: This defines the orientation of the ellipse, expressed as the rotation, in degrees, of the semi-major axis from North towards the East. For example, when the uncertainty is greatest along the North-South direction, orientation is 0 degrees; conversely, if the uncertainty is greatest along the East-West direction, orientation is 90 degrees. When orientation is not present, the orientation MUST be interpreted as 0.

exterior: When GeoLocation describes a region, the "exterior" field refers to a list of latitude/longitude points that represents the vertices of a polygon. The first and last points MUST be the same. Thus, a minimum of 4 points is required. The following polygon restrictions from [RFC5491] apply:

* A connecting line MUST NOT cross another connecting line of the same polygon.
* The vertices MUST be defined in a counter-clockwise direction.
* The edges of a polygon are defined by the shortest path between two points in space (not a geodesic curve). Consequently, the length between two adjacent vertices SHOULD be restricted to a maximum of 130 km.
* All vertices are assumed to be at the same altitude.
* Polygon shapes SHOULD be restricted to a maximum of 15 vertices (16 points that includes the repeated vertex).

confidence: The location confidence level, as an integer percentage, MAY be required, depending on the regulatory domain. When the parameter is optional and not provided, its value MUST be interpreted as 95. Valid values range from 0 to 99, since, in practice, 100-percent confidence is not achievable. The confidence value is meaningful only when GeoLocation refers to a point with uncertainty.

5.2. DeviceDescriptor

The device descriptor contains parameters that identify the specific device, such as its manufacturer serial number, regulatory-specific ID (e.g., FCC ID), and any other device characteristics required by regulatory domains.

```
+--------------------------------+
|DeviceDescriptor                |
+---------------------+----------+
|serialNumber:string  | required |
|manufacturerId:string| optional |
|modelId:string       | optional |  1..*
|rulesetIds:list      | optional |------>string
|.....................|..........|
|*other:any           |          |
+---------------------+----------+
```

Parameters:

serialNumber: The manufacturer's device serial number is REQUIRED. The length of the value MUST NOT exceed 64 characters, conforming to the X.520 [ITUT.X520.2008] recommendations.

manufacturerId: The manufacturer's ID may be REQUIRED, depending on the regulatory domain. This SHOULD represent the name of the device manufacturer, SHOULD be consistent across all devices from the same manufacturer, and SHOULD be distinct from that of other manufacturers. The string value MUST NOT exceed 64 characters in length.

modelId: The device's model ID may be REQUIRED, depending on the regulatory domain. The string value MUST NOT exceed 64 characters in length.

rulesetIds: The list of identifiers for rulesets supported by the device (see Ruleset ID Registry (Section 9.2)). A Database MAY require that the device provides this list before servicing the device requests. If the Database does not support any of the rulesets specified in the list, the Database MAY refuse to service the device requests. See RulesetInfo (Section 5.6) for discussion on ruleset identifier. If present, the list MUST contain at least one entry.

other: Depending on the regulatory domain, other parameters may be required. The Database MUST ignore all parameters in the message it does not understand. See PAWS Parameters Registry (Section 9.1) for additional valid parameters and for the process for extending the message with more parameters. Additionally, see PAWS Ruleset ID Registry (Section 9.2) for the valid set of parameters for each ruleset.

5.3. AntennaCharacteristics

Antenna characteristics provide additional information, such as the antenna height, antenna type, etc. Whether antenna characteristics must be provided in a request depends on the device type and regulatory domain.

AntennaCharacteristics	
height:float	depends on regulatory domain
heightType:enum	optional
heightUncertainty:float	depends on regulatory domain
....	....
*characteristics: various	depends on regulatory domain

Parameters:

height: The antenna height in meters. Whether the antenna height is
   required depends on the device type and the regulatory domain.
   Note that the height may be negative.

heightType: If the height is required, then heightType is also
   REQUIRED. Valid values are:

AGL	Above ground level (default)
AMSL	Above mean sea level

heightUncertainty: The height uncertainty in meters. Whether this
   is required depends on the regulatory domain.

Depending on the regulatory authority, additional antenna
characteristics may be required, such as:

- antenna direction
- antenna radiation pattern
- antenna gain
- antenna polarization

5.4. DeviceCapabilities

Device capabilities provide additional information that MAY be used
by the Device to provide additional information to the Database that
may help it to determine available spectrum. If the Database does
not support device capabilities it MUST ignore the parameter
altogether.

```
   +------------------------------+
   |DeviceCapabilities            |
   +---------------------+--------+
   |frequencyRanges:list |optional |--+
   +---------------------+---------+  | 0..*
                                      V
                +------------------------------+
                |FrequencyRange                |
                +----------------------+---------+
                |startHz:float         |required |
                |stopHz:float          |required |
                +----------------------+---------+
```

Parameters:

frequencyRanges: Optional FrequencyRange (Section 5.13) list. Each
   FrequencyRange element MUST contain start and stop frequencies,
   and optionally, channel IDs, in which the Device can operate.
   When specified, the Database SHOULD NOT return available spectrum
   that falls outside these ranges (or channel IDs).

5.5. DeviceOwner

This parameter contains device-owner information required as part of device registration. Regulatory domains MAY require additional parameters.

DeviceOwner	
owner:vcard	required
operator:vcard	optional

Parameters:

owner: The vCard contact information for the individual or business that owns the Device is REQUIRED.

operator: The vCard contact information for the device operator is OPTIONAL, but may be required by specific regulatory domains

NOTE: Depending on the regulatory domain, the Database MAY be required to validate the device-owner information. In these cases, the Database MUST respond with an error if validation fails. See PAWS Ruleset ID Registry (Section 9.2) for regulatory-specific requirements on mandatory vCard properties.

All contact information MUST be expressed using the structure defined by the vCard Format Specification [RFC6350]. Note that the vCard specification defines maximum lengths for each field, conforming to X.520 [ITUT.X520.2008] recommendations.

5.6. RulesetInfo

This contains parameters for the ruleset of a regulatory domain that is communicated using the Initialization component (Section 4.2), Device Registration (Section 4.3), and Available Spectrum Query (Section 4.4) components.

RulesetInfo	
authority:string	required
rulesetId:string	required
maxLocationChange:float	optional
maxPollingSecs:int	optional
....	
*other:any	

Parameters:

authority: A string that indicates the regulatory domain to which the ruleset applies is REQUIRED. It MUST be a 2-letter country code defined by Country Codes - ISO 3166 [ISO3166-1]. The Device SHOULD use this to determine additional device behavior required by the associated regulatory domain.

rulesetId: The ID of a ruleset for the specified authority (see Ruleset ID Registry (Section 9.2)).

maxLocationChange: The maximum location change in meters is REQUIRED for Initialization Response (Section 4.2.2), but OPTIONAL otherwise. When the Device changes location by more than this specified distance, it MUST contact the Database to get the available spectrum for the new location. If the Device is using spectrum that is no longer available, it MUST immediately cease use of the spectrum under rules for database-managed spectrum. If this value is provided within the context of an Available Spectrum Response (Section 4.4.2), it takes precedence over the value within the Initialization Response.

maxPollingSecs: The maximum duration, in seconds, between requests for available spectrum is REQUIRED for the Initialization Response (Section 4.2.2), but OPTIONAL otherwise. The Device MUST contact the Database to get available spectrum no less frequently than this duration. If the new spectrum information indicates that the Device is using spectrum that is no longer available, it MUST immediately cease use of those frequencies under rules for database-managed spectrum. If this value is provided within the context of an Available Spectrum Response (Section 4.4.2), it takes precedence over the value within the Initialization Response.

other: This message is intended to be extensible with other regulatory-specific parameters. Devices MUST ignore all parameters in the message it does not understand.

5.7. DbUpdateSpec

This message is provided by the Database to notify devices of an upcoming change to the Database URL.

```
   +-------------------------------+
   |DbUpdateSpec                   |
   +---------------------+---------+       +--------------------------+
   |databases:list       |required |------>|DatabaseSpec              |
   +---------------------+---------+  1..* +---------------+----------+
                                           |name:string    | required |
                                           |uri:string     | required |
                                           +---------------+----------+
```

Parameters:

databases: List of one or more DatabaseSpec (Section 5.8) entries.
   A Device SHOULD update its preconfigured list of databases to
   replace (only) the database that provided the response with the
   specified entries.

5.8. DatabaseSpec

This message contains the name and URI of a database.

```
+--------------------------+
|DatabaseSpec              |
+---------------+----------+
|name:string    | required |
|uri:string     | required |
+---------------+----------+
```

Parameters:

name: The display name for a database.
uri: The corresponding URI of the database.

5.9. SpectrumSpec

The SpectrumSpec element encapsulates the schedule of available
spectrum for a regulatory domain.

```
+---------------------------------------+
|SpectrumSpec                           |
+----------------------------+----------+
|rulesetInfo:RulesetInfo     | required |
|spectrumSchedules:list      | required |-----+
|timeRange:EventTime         | optional |     |
|frequencyRanges:list        | optional |     |
|needsSpectrumReport:boolean | optional |     |
|maxTotalBwHz:float          | optional |     |
|maxContiguousBwHz:float     | optional |     |
+----------------------------+----------+     |
                                              | 1..*
                                              V
                                   +--------------------------------+
                                   |SpectrumSchedule                |
                                   +---------------------+----------+
                                   |eventTime:EventTime  | required |
                                   |spectra:list         | required |
                                   +---------------------+----------+
```

Parameters:

rulesetInfo: The RulesetInfo (Section 5.6) is REQUIRED to identify the regulatory domain and ruleset for which the spectrum schedule applies (see Ruleset ID Registry (Section 9.2)). The Device MUST use the corresponding ruleset to interpret the response. Values provided within this parameter, such as maxLocationChange, take precedence over the values provided by the Initialization Procedure (Section 4.2).

spectrumSchedules: The SpectrumSchedule (Section 5.10) list is REQUIRED. At least one schedule MUST be included. More than one schedule MAY be included to represent future changes to the available spectrum. How far in advance a schedule may be provided depends on the regulatory domain. If more than one schedule is included, the eventTime intervals MUST be disjoint and SHOULD be sorted in increasing time. A gap in the time schedule indicates no available spectrum during that time-interval gap.

timeRange: The time range for which the specification is comprehensive is OPTIONAL. When specified, any gaps in time intervals within the "spectrumSchedules" element MUST be interpreted by the Device as time intervals in which there are no available spectrum.

frequencyRanges: The frequency ranges for which the specification is comprehensive is OPTIONAL. It is a list of FrequencyRange (Section 5.13) entries, and the ranges MUST be disjoint. When specified, it SHOULD correspond to the frequency ranges governed by the ruleset. A Device may combine this information with the available-spectrum specification within the "spectrumSchedules" element to distinguish between "unavailable spectrum" and "spectrum for which no information has been provided".

needsSpectrumReport: For regulatory domains that require a spectrum-usage report from devices, the Database MUST return true for this parameter if spectrumSchedules list is non-empty; otherwise, the Database MAY return false or omit this parameter altogether. If this parameter is present and its value is true, the Device MUST send a SPECTRUM_USE_NOTIFY (Section 4.4.5) message to the Database; otherwise, the Device SHOULD NOT send the SPECTRUM_USE_NOTIFY message.

maxTotalBwHz: The Database MAY return a constraint on the maximum total bandwidth (in Hertz) allowed, which may or may not be contiguous. A regulatory domain MAY require the Database to return this parameter. When present in the response, the Device MUST apply this constraint to its spectrum-selection logic to ensure total bandwidth does not exceed this value.

maxContiguousBwHz: The Database MAY return a constraint on the maximum contiguous bandwidth (in Hertz) allowed. A regulatory domain MAY require this Database to return this parameter. When present in the response, the Device MUST apply this constraint to its spectrum-selection logic to ensure no single block of spectrum has bandwidth that exceeds this value.

5.10. SpectrumSchedule

The SpectrumSchedule element combines EventTime (Section 5.14) with Spectrum (Section 5.11) to define a time period in which the spectrum is valid.

```
+--------------------------------+
|SpectrumSchedule                |
+--------------------+-----------+
|eventTime:EventTime | required |        +---------------------+
|spectra:list        | required |------->|Spectrum             |
+--------------------+-----------+   0..* +---------------------+
                                          |resolutionBwHz:float|
                                          |profiles:list        |
                                          +---------------------+
```

Parameters:

eventTime: The EventTime (Section 5.14) is REQUIRED to express "when" this specification is valid.
spectra: Spectrum (Section 5.11) list is REQUIRED to specify the available spectrum and permissible power levels, one per resolutionBwHz. The list MAY be empty when there is no available spectrum.

5.11. Spectrum

Available spectrum can be characterized by an ordered list of spectrum profiles that defines permissible power levels over a set of frequency ranges. Each Spectrum element defines permissible power levels as maximum power spectral densities over a specified resolution bandwidth, "resolutionBwHz". Note that the spectrum profiles represent the "availability mask", as defined by the governing rule set; they are not intended to encode device-level transmission-mask requirements.

- To support regulatory rules that define different "wide-band" and "narrow-band" power levels, PAWS allows multiple Spectrum elements to be included in the available-spectrum response, each with a different resolution bandwidth.
- When multiple Spectrum elements are included in the response, they represent a logical AND condition, such that the Device MUST satisfy all the conditions.
- Each Spectrum element MUST cover the range of frequencies governed by a ruleset, rather than splitting the frequencies across multiple Spectrum elements for the same resolution bandwidth.

- o Each spectrum profile represents the maximum permissible power spectral density over a contiguous range of frequencies.
- o When multiple spectrum profiles are included, they MUST be disjoint and SHOULD be ordered in non-decreasing frequency value.
- o Gaps in frequencies between consecutive spectrum profiles represent unavailability for those frequencies.

The following figure illustrates the Spectrum element and the SpectrumProfile list.

```
+------------------------------+
|Spectrum                      |
+---------------------+---------+
|resolutionBwHz:float |required |
|profiles:list        |required |---+
+---------------------+---------+   |  0..*
                                    V
              +------------------------------+
              |SpectrumProfile               |
              +--------------------+---------+
              |list                |required |
              +--------------------+---------+
                                    |
                                    V 2..*
               +----------------+---------+
               |hz:float        |required |
               |dbm:float       |required |
               +----------------+---------+
```

Parameters:

resolutionBwHz: This parameter is REQUIRED to define the resolution bandwidth (in Hertz) over which permissible power spectral density is defined. For example, FCC regulation would require one spectrum specification at a bandwidth of 6MHz, and ETSI regulation would require two specifications, at 0.1MHz and 8MHz. This parameter MAY be empty if there is no available spectrum.

profiles: A SpectrumProfile (Section 5.12) list is REQUIRED to specify permissible power levels over a set of frequency ranges. The list MAY be empty if there is no available spectrum.

Consider the following example with different permitted power spectral densities for the same set of frequencies over different resolution bandwidths (for illustrative purposes only):

```
[
  "spectrum": {
    "resolutionBwHz": 6e6,
    "profiles": [
      [
        {"hz": 5.18e8, "dbm": 30.0},
        {"hz": 5.24e8, "dbm": 30.0},
      ],
      ...
    ]
  },
  "spectrum": {
    "resolutionBwHz": 1e5,
    "profiles": [
      [
        {"hz": 5.18e8, "dbm": 27.0},
        {"hz": 5.24e8, "dbm": 27.0},
      ],
      ...
    ]
  }
]
```

This is interpreted as:

- Over any 6MHz within the frequency range, [518MHz, 524MHz), maximum permitted power is 30.0dBm (1000mW), and
- Over any 100 kHz within the frequency range, [518MHz, 524MHz), maximum permitted power is 27.0dBm (500mW)

This would allow, for example, operating two 100kHz sub-channels within the indicated 6MHz range at 500mW each, totaling 1000mW. Of course, many combinations are possible, as long as they satisfy both conditions.

The following example illustrates multiple spectrum profiles that has a gap from 530 MHz to 536 MHz:

```
   [
     "spectrum": {
       "resolutionBwHz": 6e6,
       "profiles": [
         [
           {"hz": 5.18e8, "dbm": 30.0},
           {"hz": 5.24e8, "dbm": 30.0},
           {"hz": 5.24e8, "dbm": 36.0},
           {"hz": 5.30e8, "dbm": 36.0},
         ],
         [
           {"hz": 5.36e8, "dbm": 30.0},
           {"hz": 5.42e8, "dbm": 30.0},
         ],
         ...
       ]
     },
     "spectrum": {
       "resolutionBwHz": 1e5,
       "profiles": [
         [
           {"hz": 5.18e8, "dbm": 27.0},
           {"hz": 5.24e8, "dbm": 27.0},
           {"hz": 5.24e8, "dbm": 30.0},
           {"hz": 5.30e8, "dbm": 30.0},
         ],
         [
           {"hz": 5.36e8, "dbm": 27.0},
           {"hz": 5.42e8, "dbm": 27.0},
         ],
         ...
       ]
     }
   ]
```

5.12. SpectrumProfile

A spectrum profile is characterized by an ordered list of (frequency, power) points that represents the shape of maximum permissible power levels over a range of frequencies.

- It MUST contain a minimum of two entries.
- The entries in the list MUST be ordered in non-decreasing frequency values.
- Two consecutive points MAY have the same frequency value to represent a "step function".

- o Three or more points MUST NOT share the same frequency value.
- o The first frequency is inclusive; the last frequency is exclusive.

The following figure defines the SpectrumProfile element.

```
+-------------------------------+
|SpectrumProfile                |
+---------------------+---------+
|list                 |required |---+
+---------------------+---------+   |  2..*
                                    V
              +----------------+---------+
              |hz:float        |required |
              |dbm:float       |required |
              +----------------+---------+
```

Parameters of each point in the profile:

hz: The frequency, in Hertz, at which the power level is defined.
dbm: The power level, expressed as dBm per resolution bandwidth, as defined by the "resolutionBwHz" element of the enclosing Spectrum (Section 5.11) element.

5.13. FrequencyRange

The FrequencyRange parameter specifies a frequency range.

```
+---------------------------------+
|FrequencyRange                   |
+----------------------+---------+
|startHz:float         |required |
|stopHz:float          |required |
+----------------------+---------+
```

Parameters:
startHz: The inclusive start of the frequency range (in Hertz) is REQUIRED.
stopHz: The exclusive end of the frequency range (in Hertz) is REQUIRED.

5.14. EventTime

The EventTime element specifies the start and stop times of an "event". This is used to indicate the time period for which a Spectrum (Section 5.11) is valid.

```
+---------------------------+
|EventTime                  |
+-----------------+---------+
|startTime:string |required |
|stopTime:string  |required |
+-----------------+---------+
```

Parameters:

startTime: The inclusive start of the event is REQUIRED.
stopTime: The exclusive end of the event is REQUIRED.

Both times are expressed using the format, YYYY-MM-DDThh:mm:ssZ, as defined by Date and Time on the Internet: Timestamps [RFC3339]. The times MUST be expressed using UTC.

A device that does not have access to the current date and time MUST use the timestamp at the top-level of the response message as a substitute for the current time (see Available Spectrum Response (Section 4.4.2) and Available Spectrum Batch Response (Section 4.4.4)). E.g.,

- (startTime - timestamp) gives the duration that a device must wait before the event becomes "active". If the value is zero or negative, the event is already active.
- If the event is already active, (stopTime - timestamp) is the duration that the event remains active. If the value is zero or negative, the event is no longer active and MUST be ignored.

5.15. GeoSpectrumSpec

The GeoSpectrumSpec element encapsulates the available spectrum for a location. It is returned within a AVAIL_SPECTRUM_BATCH_RESP (Section 4.4.4) batch response that contains one GeoSpectrumSpec parameter for each location in the request.

```
+------------------------------------+
|GeoSpectrumSpec                     |
+-----------------------+------------+
|location:GeoLocation   | required |
|spectrumSpecs:list     | required |-------+
+-----------------------+----------+       |
                                           | 1..*
                                           V
                                   +--------------+
                                   | SpectrumSpec |
                                   +--------------+
```

Parameters:

location: The GeoLocation (Section 5.1) is REQUIRED to identify the location at which the spectrum schedule applies.

spectrumSpecs: The SpectrumSpec (Section 5.9) list is REQUIRED. At least one entry MUST be included. Each entry represents schedules of available spectrum for a regulatory domain. More than one entry MAY be included to support multiple regulatory domains at a location.

5.16. DeviceValidity

The DeviceValidity element is used to indicate whether a device is valid. See Section 4.5.2.

DeviceValidity	
deviceDesc:DeviceDescriptor	required
isValid:boolean	required
reason:string	optional

Parameters:

deviceDesc: The DeviceDescriptor (Section 5.2) that was used to check for validity is REQUIRED.

isValid: A REQUIRED boolean value that indicates whether the Device is valid.

reason: If the device identifier is not valid, the Database MAY include a reason. The reason MAY be in any language. The length of the value MUST NOT exceed 128 characters.

5.17. Error Element

If the Database responds to a PAWS request message with an error, it MUST include an Error element.

Error	
code:int	required
message:string	optional
data:any	optional

Parameters:

code: An integer code that indicates the error type.

message: A description of the error. It MAY be in any language. The length of the value MUST NOT exceed 128 characters.

data: The Database MAY include additional data. For some errors, additional data may be required. The Device MUST ignore any data parameters it does not understand.

The following table defines valid error codes. They are loosely grouped into the following categories:

-100s: Indicates compatibility issues, e.g., version mismatch, unsupported or unimplemented features.

-200s: Indicates that the Device request contains an error that needs to be modified before making another request.

-300s: Indicates authorization-related issues.

Code	Name	Description
-100	(reserved)	
-101	VERSION	The Database does not support the specified version of the message.
-102	UNSUPPORTED	The Database does not support the Device. For example, it does not support the regulatory domain specified in the request.
-103	UNIMPLEMENTED	The Database does not implement the optional request or optional feature.
-104	OUTSIDE_COVERAGE	The specified geo-location is outside the coverage area of the Database. The Database MAY include a list of alternate databases that might be appropriate for the requested location. See OUTSIDE_COVERAGE Error (Section 5.17.1).
-105	DATABASE_CHANGE	The Database has changed its URI. The Database MAY include a DbUpdateSpec (Section 5.7) parameter in the error response to provide devices with one or more alternate database URLs. The Device SHOULD use the information to update its list of preconfigured databases to replace (only) its entry for the responding Database with the list of alternate database URLs. See DATABASE_CHANGE Error (Section 5.17.2).
-200	(reserved)	

-201 REQUIRED	A required parameter is missing. The Database MUST include a list of the required parameter names. The Database MAY include only names of parameters that are missing, but MAY include a full list. Including the full list of missing parameters may reduce the number of re-queries from the Device. See REQUIRED Error (Section 5.17.3).
-202 INVALID_VALUE	A parameter value is invalid in some way. The Database SHOULD include a message indicating which parameter and why its value is invalid.
-300 (reserved)	
-301 UNAUTHORIZED	The Device is not authorized to used the Database. Authorization may be determined by regulatory rules or be dependent on prior arrangement between the Device and Database.
-302 NOT_REGISTERED	Device registration required, but the Device is not registered.

Table 1: Error Codes

5.17.1. OUTSIDE_COVERAGE Error

When the error code is OUTSIDE_COVERAGE, the Database MAY include an ErrorData element within its Error response as the "data" field, and, if present, the ErrorData MAY include a list of DatabaseSpec (Section 5.8) entries that might be appropriate for the requested location.

```
+---------------------------+
|Error                      |
+----------------+----------+
|code:int        | required |
|message:string  | optional |    +-----------------------------+
|data:ErrorData  | optional |--->|ErrorData                    |
+----------------+----------+    +------------------+----------+
                                 |databases:list    | optional |
                                 +------------------+----------+
```

5.17.2. DATABASE_CHANGE Error

When the error code is DATABASE_CHANGE, the Database MAY include an ErrorData element within its Error response as the "data" field, and, if present, the ErrorData MUST include a DbUpdateSpec (Section 5.7) element that provides a list of alternate databases.

```
   +----------------------------+
   |Error                       |
   +----------------+-----------+
   |code:int        | required |
   |message:string  | optional |    +------------------------------+
   |data:ErrorData  | optional |--->|ErrorData                     |
   +----------------+-----------+    +------------------+-----------+
                                     |spec:DbUpdateSpec | required |
                                     +------------------+-----------+
```

5.17.3. REQUIRED Error

When the error code is REQUIRED, the Database MUST include an ErrorData element within its Error response as the "data" field, and the ErrorData element MUST include a list of the missing required parameters and MAY include the list of all required parameters.

```
   +----------------------------+
   |Error                       |
   +----------------+-----------+
   |code:int        | required |
   |message:string  | optional |    +----------------------------+
   |data:ErrorData  | required |--->|ErrorData                   |
   +----------------+-----------+    +----------------+-----------+ 1..*
                                     |parameters:list | required |--+
                                     +----------------+-----------+  |
                                                                     v
                                                                   string
```

Parameters:

parameters: List of one or more parameter names (strings). The name of a parameter SHOULD be expressed using dotted notation, when appropriate, e.g., "deviceDesc.serialNumber".

6. Message Encoding

The PAWS protocol is encoded using JSON-RPC [JSON-RPC] (see also JavaScript Object Notation (JSON) [RFC4627]). Each component described in Protocol Functionalities (Section 4) corresponds to one or more JSON-RPC methods. This section provides the JSON schema for each of the protocol messages and parameters defined in sections Protocol Functionalities (Section 4) and Protocol Parameters (Section 5). JSON schemas are presented in accordance with A JSON Media Type for Describing the Structure and Meaning of JSON Documents [I-D.zyp-json-schema].

NOTE: In general, all messages defined in this section are extensible

by adding additional properties to support regulatory-specific and database-specific requirements. In all cases, the Device or Database MUST ignore any parameter it does not understand.

6.1. JSON-RPC Binding

The JSON-RPC [JSON-RPC] protocol consists of two basic objects, Request and Response:

- o The JSON-RPC Request object encapsulates a PAWS functionality (operation) and the request message
- o The JSON-RPC Response object encapsulates a PAWS response message and Error element

The Database and Device MUST support JSON-RPC 2.0 encoding.

The JSON-RPC Request for PAWS has the following the following forms:

```
{
  "jsonrpc": "2.0",
  "method": string,
  "params": <PAWS_REQ>,
  "id": string
}
```

where "method" is the name of a PAWS functionality (operation), and <PAWS_REQ> represents one of the PAWS request objects associated with the method. Method names are defined with the prefix, "spectrum.paws.".

The non-error JSON-RPC Response for PAWS has the following forms:

```
{
  "jsonrpc": "2.0",
  "result": <PAWS_RESP>,
  "id": string
}
```

where <PAWS_RESP> represents one of the PAWS response objects associated with the method.

The error JSON-RPC Response for PAWS has the following form:

```
{
  "jsonrpc": "2.0",
  "error": {
    "code": integer,
    "message": string,
    "data": object,
  },
  "id": string
}
```

where the Error object and error codes are described by Error Element (Section 5.17). The Database SHOULD attempt to use the most specific applicable PAWS error code. When an accurate one is not available, it SHOULD fall back to standard JSONRPC error codes as defined in JSONRPC specification. For example, if the Database receives invalid JSON from the Device, it should respond with "-32700", signifying a parse error. As a last resort, the Database MAY send a suitable HTTP 5xx response.

Depending on prior arrangement between a Database and Device, the Request and Response objects MAY contain additional parameters. The Database or Device MUST ignore all parameters it does not understand.

6.2. init Method

This section describes the encoding for the JSON-RPC "spectrum.paws.init" method that represents the Initialization functionality (Section 4.2).

6.2.1. INIT_REQ Parameters

The JSON encoding of the Initialization request message INIT_REQ (Section 4.2.1) is described by the following schema:

```
   {
     "name": "INIT_REQ",
     "type": "object",
     "properties": {
       "type": "INIT_REQ",
       "version": {
         "type": "string",
         "required": true
       },
       "deviceDesc": {
         "type": "DeviceDescriptor",
         "required": true
       },
       "location": {
         "type": "GeoLocation",
         "description": "The location SHOULD be the current location
             of the Device's antenna. Depending on the regulatory
             domain, the location MAY be the anticipated position of
             the Device.",
         "required": true
       }
     }
   }
```

Example "init" JSON-RPC request:

```
   {
    "jsonrpc": "2.0",
    "method": "spectrum.paws.init",
    "params": {
     "type": "INIT_REQ",
     "version": "1.0",
     "deviceDesc": {
      "serialNumber": "XXX",
      "fccId": "YYY",
      ...
     },
     "location": {
      "point": {
       "center": {"latitude": 37.0, "longitude": -101.3}
      }
     }
    },
    "id": "xxxxxx"
   }
```

6.2.2. INIT_RESP Parameters

The JSON encoding of the Initialization response message INIT_RESP (Section 4.2.2) is described by the following schema:

```
{
  "name": "INIT_RESP",
  "type": "object",
  "properties": {
    "type": "INIT_RESP",
    "version": {
      "type": "string",
      "required": true
    },
    "rulesetInfos": {
      "type": "array",
      "description": "List of regulatory domains and associated
          attributes that govern the device at the location specified
          by the device. The list MUST have at least one entry.",
      "items": "RulesetInfo",
      "required": true
    },
    "databaseChange": {
      "type": "DbUpdateSpec",
      "description": "Indicates that the Database URI will be
          changing. Devices need to update their configurations",
      "required": false
    }
  }
}
```

Example "init" JSON-RPC response:

```
{
 "jsonrpc": "2.0",
 "result": {
  "type": "INIT_RESP",
  "version": "1.0",
  "rulesetInfos": [
    {
      "authority": "us",
      ...
    },
    ...
  ]
 },
 "id": "xxxxxx"
}
```

6.3. register Method

This section describes the encoding for the JSON-RPC "spectrum.paws.register" method that represents Device Registration functionality (Section 4.3).

6.3.1. REGISTRATION_REQ Parameters

The JSON encoding of the Registration request message REGISTRATION_REQ (Section 4.3.1) is described by the following schema:

```
{
  "name": "REGISTRATION_REQ",
  "type": "object",
  "properties": {
    "type": "REGISTRATION_REQ",
    "version": {
      "type": "string",
      "required": true
    },
    "deviceDesc": {
      "type": "DeviceDescriptor",
      "required": true
    },
    "location": {
      "type": "GeoLocation",
      "description": "The location SHOULD be the current location
          of the Device's antenna. Depending on the regulatory
          domain, the location MAY be the anticipated position of
          the Device.",
      "required": true
    },
    "deviceOwner": {
      "type": "DeviceOwner",
      "required": true
    },
    "antenna": {
      "type": "AntennaCharacteristics",
      "description": "Antenna characteristics, including its
          height and height type",
      "required": false
    }
  }
}
```

Example "register" JSON-RPC request:

```
   {
    "jsonrpc": "2.0",
    "method": "spectrum.paws.register",
    "params": {
     "type": "REGISTRATION_REQ",
     "version": "1.0",
     "deviceDesc": {
      "serialNumber": "XXX",
      "fccId": "YYY",
      ...
     },
     "deviceOwner": {
      "owner": [
       "vcard", [
        ["version", {}, "text", "4.0"],
        ["kind", {}, "text", "org"],
        ["fn", {}, "text", "Racafrax, Inc."]
       ]
      ],
      "operator": [
       "vcard", [
        ["version", {}, "text", "4.0"],
        ["fn", {}, "text", "John Frax"],
        ["adr", {}, "text",
         ["", "", "100 Main Street",
          "Summersville", "CA", "90034", "USA"
         ]
        ],
        ["tel", {}, "uri", "tel:+1-213-555-1212"],
        ["email", {}, "text", "j.frax@rackafrax.com"]
       ]
      ]
     },
     "location": {
      "point": {
       "center": {"latitude": 37.0, "longitude": -101.3}
      }
     },
     "antenna": {"height": 10.2, "heightType": "AGL"}
    },
    "id": "xxxxxx"
   }
```

6.3.2. REGISTRATION_RESP Parameters

The JSON encoding of the Registration response message REGISTRATION_RESP (Section 4.3.2) is described by the following schema:

```
   {
     "name": "REGISTRATION_RESP",
     "type": "object",
     "properties": {
       "type": "REGISTRATION_RESP",
       "version": {
         "type": "string",
         "required": true
       },
       "rulesetInfos": {
         "type": "array",
         "description": "List of regulatory domains for which device
             registration was successful. The list MUST have at
             least one entry.",
         "items": "RulesetInfo",
         "required": true
       },
       "databaseChange": {
         "type": "DbUpdateSpec",
         "description": "Indicates that the Database URI will be
             changing. Devices need to update their configurations",
         "required": false
       }
     }
   }
```

Example "register" JSON-RPC response:

```
   {
    "jsonrpc": "2.0",
    "result": {
     "type": "REGISTRATION_RESP",
     "version": "1.0"
     "rulesetInfos": [
       {
         "authority": "us",
         ...
       }
     ]
    },
    "id": "xxxxxx"
   }
```

6.4. getSpectrum Method

This section describes the encoding for the JSON-RPC "spectrum.paws.getSpectrum" method that represents the single-location query of the Available Spectrum Query functionality

(Section 4.4) that enables a Device to obtain a set of available spectrum from the Database.

6.4.1. AVAIL_SPECTRUM_REQ Parameters

The JSON encoding of the Available Spectrum request message AVAIL_SPECTRUM_REQ (Section 4.4.1) is described by the following schema:

```
{
  "name": "AVAIL_SPECTRUM_REQ",
  "type": "object",
  "properties": {
    "type": "AVAIL_SPECTRUM_REQ",
    "version": {
      "type": "string",
      "required": true
    },
    "deviceDesc": {
      "type": "DeviceDescriptor",
      "description": "Descriptor of the device for which to
          determine available spectrum.",
      "required": false
    },
    "location": {
      "type": "GeoLocation",
      "description": "The location SHOULD be the current location
          of the Device's antenna. Depending on the regulatory
          domain, the location MAY be the anticipated position of
          the Device. When request is made by a Master Device on
          behalf of a Slave Device, this is the location of the
          Slave Device.",
      "required": false
    },
    "owner": {
      "type": "DeviceOwner",
      "description": "May be required if the Device is not yet
          registered or if the DB does not implement a separate
          device-registration request. Also depends on device type
          and regulatory domain",
      "required": false
    },
    "antenna": {
      "type": "AntennaCharacteristics",
      "description": "Antenna characteristics, including its
          height and height type. May required depending on
          device type and regulatory domain",
      "required": false
```

```
       },
       "capabilities": {
         "type": "DeviceCapabilities",
         "description": "The Database SHOULD NOT return spectrum that
             is incompatible with the specified capabilities.",
         "required": false
       },
       "masterDeviceDesc": {
         "type": "DeviceDescriptor",
         "description": "When the request is made by a Master Device
             on behalf of a Slave Device, this is the descriptor of
             the Master Device.",
         "required": false
       },
       "masterDeviceLocation": {
         "type": "GeoLocation",
         "description": "When the request is made by a Master Device
             on behalf of a Slave Device, this is the location of
             the Master Device.",
         "required": false
       },
       "requestType": {
         "type": "string",
         "description": "Optional value that modifies the request.
             Valid values depends on regulatory domain.",
         "required": false
       }
     }
   }
```

Example "getSpectrum" JSON-RPC request:

```
   {
    "jsonrpc": "2.0",
    "method": "spectrum.paws.getSpectrum",
    "params": {
     "type": "AVAIL_SPECTRUM_REQ",
     "version": "1.0",
     "deviceDesc": {
      "serialNumber": "XXX",
      "fccId": "YYY",
      ...
     },
     "location": {
      "point": {
       "center": {"latitude": 37.0, "longitude": -101.3}
      }
     },
     "antenna": {"height": 10.2, "heightType": "AGL"}
    },
    "id": "xxxxxx"
   }
```

6.4.2. AVAIL_SPECTRUM_RESP Parameters

The JSON encoding of the Available Spectrum response message AVAIL_SPECTRUM_RESP (Section 4.4.2) is described by the following schema:

```
   {
     "name": "AVAIL_SPECTRUM_RESP",
     "type": "object",
     "properties": {
       "type": "AVAIL_SPECTRUM_RESP",
       "version": {
         "type": "string",
         "required": true
       },
       "timestamp": {
         "type": "string",
         "description": "Timestamp of the response, using
             YYYY-MM-DDThh:mm:ssZ RFC3339 format. This SHOULD be used
             by the Device as a reference for the start and stop times
             in the spectrum schedule",
         "format": "date-time",
         "required": true
       },
       "deviceDesc": {
         "type": "DeviceDescriptor",
         "required": true
       },
       "spectrumSpecs": {
         "type": "array",
         "description": "The Database MAY return more than one
             SpectrumSpec to represent available spectrum for multiple
             regulatory domains at the specified location.",
         "items": "SpectrumSpec",
         "required": true
       },
       "databaseChange": {
         "type": "DbUpdateSpec",
         "description": "Indicates that the Database URI will be
             changing. Devices need to update their configurations",
         "required": false
       }
     }
   }
```

Example "getSpectrum" JSON-RPC response:

```
   {
    "jsonrpc": "2.0",
    "result": {
     "type": "AVAIL_SPECTRUM_RESP",
     "version": "1.0",
     "timestamp": "2013-03-02T14:30:21Z",
     "deviceDesc": {
```

```
       "serialNumber": "XXX",
       "fccId": "YYY",
       ...
      },
      "spectrumSpecs": [
       {
        "rulesetInfo": {
          "authority": "us",
          ...
        },
        "needsSpectrumReport": false,
        "spectrumSchedules": [
         {
          "eventTime": {
           "startTime": "2013-03-02T14:30:21Z",
           "stopTime": "2013-03-02T20:00:00Z",
          },
          "spectra": [
            {
             "resolutionBwHz": 6e6,
             "profiles": [
               ...
               [
                {"hz":5.18e8, "dbm":30.0},
                {"hz":5.36e8, "dbm":30.0},
                {"hz":5.36e8, "dbm":36.0},
                {"hz":5.42e8, "dbm":36.0}
               ],
               [
                {"hz":6.20e8, "dbm":30.0},
                {"hz":6.26e8, "dbm":30.0},
               ],
               ...
             ]
            },
            {
             "resolutionBwHz": 1e5,
             "profiles": [
               ...
               [
                {"hz":5.18e8, "dbm":27.0},
                {"hz":5.36e8, "dbm":27.0},
                {"hz":5.36e8, "dbm":30.0},
                {"hz":5.42e8, "dbm":30.0}
               ],
               [
                {"hz":6.20e8, "dbm":27.0},
                {"hz":6.26e8, "dbm":27.0},
```

```
              ],
              ...
            ]
          }
        ]
      },
      {
       "eventTime": {
        "startTime": "2013-03-02T22:00:00Z",
        "stopTime": "2013-03-03T14:30:21Z",
       },
       "spectra": [
        ...
       ]
      }
     ],
    }
   ]
  },
  "id": "xxxxxx"
 }
```

6.5. getSpectrumBatch Method

This section describes the encoding for the JSON-RPC "spectrum.paws.getSpectrumBatch" method that represents the multiple-location query of the Available Spectrum Query functionality (Section 4.4) that enables a Device to obtain a set of available spectrum for multiple locations from the Database.

6.5.1. AVAIL_SPECTRUM_BATCH_REQ Parameters

The JSON encoding of the Batch Available Spectrum request AVAIL_SPECTRUM_BATCH_REQ (Section 4.4.3) is described by the following schema. This an OPTIONAL feature of the Database.

```
 {
   "name": "AVAIL_SPECTRUM_BATCH_REQ",
   "type": "object",
   "properties": {
     "type": "AVAIL_SPECTRUM_BATCH_REQ",
     "version": {
       "type": "string",
       "required": true
      },
     "deviceDesc": {
       "type": "DeviceDescriptor",
       "description": "Descriptor of the device for which to
```

```
            determine available spectrum.",
        "required": true
      },
      "locations": {
        "type": "array",
        "description": "At least one device location is required.
            Additional (anticipated) locations can also be included,
            as permitted by regulatory domain. When the request is
            made by a Master Device on behalf of a Slave Device, these
            are locations of the Slave Device.",
        "items": "GeoLocation",
        "required": true
      },
      "owner": {
        "type": "DeviceOwner",
        "description": "May be required if the Device is not yet
            registered or if the DB does not implement a separate
            device-registration request. Also depends on device type
            and regulatory domain",
        "required": false
      },
      "antenna": {
        "type": "AntennaCharacteristics",
        "description": "Antenna characteristics, including its
            height and height type. May required depending on
            device type and regulatory domain","AntennaCharacteristics",
        "required": false
      },
      "capabilities": {
        "type": "DeviceCapabilities",
        "description": "The Database SHOULD NOT return spectrum that
            is incompatible with the specified capabilities.",
        "required": false
      },
      "masterDeviceDesc": {
        "type": "DeviceDescriptor",
        "description": "When the request is made by a Master Device
            on behalf of a Slave Device, this is the descriptor of
            the Master Device.",
        "required": false
      },
      "masterDeviceLocation": {
        "type": "GeoLocation",
        "description": "When the request is made by a Master Device
            on behalf of a Slave Device, this is the location of
            the Master Device.",
        "required": false
      },
```

```
      "requestType": {
        "type": "string",
        "description": "Optional value that modifies the request.
            Valid values depends on regulatory domain.",
        "required": false
      }
    }
  }
```

Example "getSpectrumBatch" JSON-RPC request:

```
   {
    "jsonrpc": "2.0",
    "method": "spectrum.paws.getSpectrumBatch",
    "params": {
     "type": "AVAIL_SPECTRUM_BATCH_REQ",
     "version": "1.0",
     "deviceDesc": {
      "serialNumber": "XXX",
      "fccId": "YYY",
      ...
     },
     "locations": [
      {
       "point": {
        "center": {"latitude": 37.0, "longitude": -101.3}
       }
      },
      {
       "point": {
        "center": {"latitude": 37.0005, "longitude": -101.3005}
       }
      },
      ...
     ],
     "antenna": {"height": 10.2, "heightType": "AGL"}
    },
    "id": "xxxxxx"
   }
```

6.5.2. AVAIL_SPECTRUM_BATCH_RESP Parameters

The JSON encoding of the Batch Available Spectrum response AVAIL_SPECTRUM_BATCH_RESP (Section 4.4.4) is described by the following schema:

```
{
  "name": "AVAIL_SPECTRUM_BATCH_RESP",
  "type": "object",
  "properties": {
    "type": "AVAIL_SPECTRUM_BATCH_RESP",
    "version": {
      "type": "string",
      "required": true
     },
    "timestamp": {
      "type": "string",
      "description": "Timestamp of the response, using
          YYYY-MM-DDThh:mm:ssZ RFC3339 format. This SHOULD be used
          by the Device as a reference for the start and stop times
          in the spectrum schedule",
      "format": "date-time",
      "required": true
    },
    "deviceDesc": {
      "type": "DeviceDescriptor",
      "required": true
    },
    "geoSpectrumSpecs": {
      "type": "array",
      "description": "A list of locations and available spectrum at
          each location. For each location, there may be more than
          one SpectrumSpec element to support more than one ruleset
          at that location.",
      "items": "GeoSpectrumSpec",
      "required": true
    },
    "databaseChange": {
      "type": "DbUpdateSpec",
      "description": "Indicates that the Database URI will be
          changing. Devices need to update their configurations",
      "required": false
    }
  }
}
```

Example "getSpectrumBatch" JSON-RPC response:

```
{
 "jsonrpc": "2.0",
 "result": {
  "type": "AVAIL_SPECTRUM_BATCH_RESP",
  "version": "1.0",
  "timestamp": "2013-03-02T14:30:21Z",
```

```
     "deviceDesc": {
      "serialNumber": "XXX",
      "fccId": "YYY",
      ...
     },
     "geoSpectrumSpecs": [
      {
       "location": {
        "point": {
         "center": {"latitude": 37.0, "longitude": -101.3}
        }
       },
       "spectrumSpecs": [
        {
         "rulesetInfo": {
          "authority": "us",
          ...
         },
         "needsSpectrumReport": false,
         "spectrumSchedules": [
          {
           "eventTime": {
            "startTime": "2013-03-02T14:30:21Z",
            "stopTime": "2013-03-02T20:00:00Z",
           },
           "spectra": [
             {
              "resolutionBwHz": 6e6,
              "profiles": [
                ...
                [
                 {"hz":5.18e8, "dbm":30.0},
                 {"hz":5.36e8, "dbm":30.0},
                 {"hz":5.36e8, "dbm":36.0},
                 {"hz":5.42e8, "dbm":36.0}
                ],
                [
                 {"hz":6.20e8, "dbm":30.0},
                 {"hz":6.26e8, "dbm":30.0},
                ],
                ...
              ]
             },
             {
              "resolutionBwHz": 1e5,
              "profiles": [
                ...
                [
```

```
                 {"hz":5.18e8, "dbm":27.0},
                 {"hz":5.36e8, "dbm":27.0},
                 {"hz":5.36e8, "dbm":30.0},
                 {"hz":5.42e8, "dbm":30.0}
                ],
                [
                 {"hz":6.20e8, "dbm":27.0},
                 {"hz":6.26e8, "dbm":27.0},
                ],
                ...
              ]
             },
           ]
          },
          {
           "eventTime": {
            "startTime": "2013-03-02T22:00:00Z",
            "stopTime": "2013-03-03T14:30:21Z",
           },
           "spectra": [
            ...
           ]
          }
         ],
        }
       ]
      },
      {
       "location": {
        "point": {
         "center": {"latitude": 37.0005, "longitude": -101.3005}
        }
       },
       "spectrumSpecs": [
        ...
       ]
      }
     ],
    },
    "id": "xxxxxx"
   }
```

6.6. notifySpectrumUse Method

This section describes the encoding for the JSON-RPC "spectrum.paws.notifySpectrumUse" method that represents the Spectrum-usage notification of the Available Spectrum Query functionality (Section 4.4.5) that notifies the Database of

anticipated spectrum usage.

6.6.1. SPECTRUM_USE_NOTIFY Parameters

The JSON encoding of the Spectrum Notification message SPECTRUM_USE_NOTIFY (Section 4.4.5) is described by the following schema:

```
{
  "name": "SPECTRUM_USE_NOTIFY",
  "type": "object",
  "properties": {
    "type": "SPECTRUM_USE_NOTIFY",
    "version": {
      "type": "string",
      "required": true
    },
    "deviceDesc": {
      "type": "DeviceDescriptor",
      "required": true
    },
    "location": {
      "type": "GeoLocation",
      "description": "The location of the Device. When the
          notification is made by a Master Device on behalf
          of a Slave Device, this is the location of the
          Slave Device."
      "required": false
    },
    "masterDeviceLocation": {
      "type": "GeoLocation",
      "description": "When the notification is made by the
          Master Device on behalf of a Slave Device, this is
          the location of the Master Device."
      "required": false
    },
    "spectra": {
      "type": "array",
      "description": "The spectrum anticipated to be used by
          the Device.",
      "items": "Spectrum",
      "required": true
    }
  }
}
```

Example "notifySpectrumUse" JSON-RPC notification:

```
{
 "jsonrpc": "2.0",
 "method": "spectrum.paws.notifySpectrumUse",
 "params": {
  "type": "SPECTRUM_USE_NOTIFY",
  "version": "1.0",
  "deviceDesc": {
   "serialNumber": "XXX",
   "fccId": "YYY",
   ...
  },
  "location": {
   "point": {
    "center": {"latitude": 37.0005, "longitude": -101.3005}
   }
  },
  "spectra": [
   {
    "resolutionBwHz": 6e6,
    "profiles": [
      [
        {"hz":5.18e8, "dbm":30.0},
        {"hz":5.24e8, "dbm":30.0}
      ]
    ]
   },
  ]
 },
 "id": "xxxxxx"
}
```

6.6.2. SPECTRUM_USE_RESP Parameters

The JSON encoding of the Spectrum-usage response SPECTRUM_USE_RESP (Section 4.4.6) is described by the following schema:

```
{
  "name": "SPECTRUM_USE_RESP",
  "type": "object",
  "properties": {
    "type": "SPECTRUM_USE_RESP",
    "version": {
      "type": "string",
      "required": true
    }
  }
}
```

Example "notifySpectrumUse" JSON-RPC response:

```
{
 "jsonrpc": "2.0",
 "result": {
  "type": "SPECTRUM_USE_RESP",
  "version": "1.0",
 },
 "id": "xxxxxx"
}
```

6.7. verifyDevice Method

This section describes the encoding for the JSON-RPC "spectrum.paws.verifyDevice" method that represents the Device Validation functionality (Section 4.5). This is used by a Master Device to validate Slave Devices.

6.7.1. DEV_VALID_REQ Parameters

The JSON encoding of the Device Validation request DEV_VALID_REQ (Section 4.5.1) is described by the following schema:

```
{
  "name": "DEV_VALID_REQ",
  "type": "object",
  "properties": {
    "type": "DEV_VALID_REQ",
    "version": {
      "type": "string",
      "required": true
    },
    "deviceDescs": {
      "type": "array",
      "description": "List of Slave Devices to be validated",
      "items": "DeviceDescriptor",
      "required": true
    }
  }
}
```

Example "verifyDevice" JSON-RPC request:

```
   {
    "jsonrpc": "2.0",
    "method": "spectrum.paws.verifyDevice",
    "params": {
     "type": "DEV_VALID_REQ",
     "version": "1.0",
     "deviceDescs": [
      {
       "serialNumber": "XXX",
       "fccId": "YYY",
       ...
      },
      {
       "serialNumber": "XXX3",
       "fccId": "YYY2",
       ...
      },
      ...
     ]
    },
    "id": "xxxxxx"
   }
```

6.7.2. DEV_VALID_RESP Parameters

The JSON encoding of the Device Validation response DEV_VALID_RESP (Section 4.5.2) is described by the following schema:

```
   {
     "name": "DEV_VALID_RESP",
     "type": "object",
     "properties": {
       "type": "DEV_VALID_RESP",
       "version": {
         "type": "string",
         "required": true
       },
       "deviceValidities": {
         "type": "array",
         "description": "List of DeviceValidity objects that shows the
             validity of each device included in the original Device
             Validity Request message.",
         "items": "DeviceValidity",
         "required": true
       },
       "databaseChange": {
         "type": "DbUpdateSpec",
         "description": "Indicates that the Database URI will be
             changing. Devices need to update their configurations",
         "required": false
       }
     }
   }
```

Example "verifyDevice" JSON-RPC response:

```
   {
    "jsonrpc": "2.0",
    "result": {
     "type": "DEV_VALID_RESP",
     "version": "1.0",
     "deviceValidities": [
      {
       "deviceDesc": {
         "serialNumber": "XXX",
         "fccId": "YYY",
         ...
       },
       "isValid": true
      },
      {
       "deviceDesc": {
         "serialNumber": "XXX3",
         "fccId": "YYY2",
         ...
       },
       "isValid": false,
       "reason": "Not authorized"
      }
     ]
    },
    "id": "xxxxxx"
   }
```

6.8. Sub-message Schemas

This section defines the schema for Protocol Parameters (Section 5) embedded in PAWS request and response messages.

6.8.1. GeoLocation

This parameter is used to specify the GeoLocation (Section 5.1) of the Device. The geometric shapes represent the JSON encoding shapes defined in GEOPRIV Presence Information Data Format Location Object [RFC5491].

```
     {
       "name": "GeoLocation",
       "type": "object",
       "properties": {
         "point": {
           "description": "A single location, with optional
               uncertainty measures. Exactly one of 'point' or 'region'
               must be present.",
```

```
           "type": "Ellipse",
           "required": false
         },
         "region": {
           "description": "A region described by a polygon. Exactly
               one of 'point' or 'region' must be present.",
           "type": "Polygon",
           "required": false
         },
         "confidence": {
           "description": "Confidence interval when location
               is a point with uncertainty. 0 to 100.",
           "type": "integer",
           "required": false,
           "default": 95
         }
       }
     }
     {
       "name": "Point",
       "type": "object",
       "properties": {
         "latitude": {
           "description": "Double-precision floating-point degrees.
               WGS84 datum.",
           "type": "number",
           "required": true
         },
         "longitude": {
           "type": "number",
           "description": "Double-precision floating-point degree.
               WGS84 datum.",
           "required": true
         }
       }
     }
     {
       "name": "Ellipse",
       "type": "object",
       "properties": {
         "center": {
           "type": "Point",
           "required": true
         },
         "semiMajorAxis": {
           "description": "Floating-point meters that describe
               location uncertainty along the major axis of
               the ellipse.",
```

```
          "type": "number",
          "required": false,
          "default": 0
        },
        "semiMinorAxis": {
          "description": "Floating-point meters that describe
              location uncertainty along the minor axis of
              the ellipse.",
          "type": "number",
          "required": false,
          "default": 0
        },
        "orientation": {
          "description": "Orientation of the ellipse, as rotation
              of the major axis from North towards East. Degrees.",
          "type": "number",
          "required": false,
          "default": 0
        }
      }
    }
    {
      "name": "Polygon",
      "type": "object",
      "properties": {
        "exterior": {
          "description": "List of Points in counter-clockwise
              order. They MUST form a loop with no edges that
              cross each other. First and last points MUST be
              the same. Minimum of 4 points.",
          "type": "array",
          "items": "Point",
          "required": true
        }
      }
    }
```

6.8.2. DeviceDescriptor

The DeviceDescriptor (Section 5.2) contains parameters that identify the specific device, such as its manufacturer serial number, regulatory-specific ID (e.g., FCC ID), and any other device characteristics required by regulatory domains, such as device-type classification. See Initial PAWS Parameters Registry Contents (Section 9.1.2) for additional valid parameters, e.g., "fccId", "etsiEnDeviceType", etc.

```
     {
       "name": "DeviceDescriptor",
       "type": "object",
       "properties": {
         "serialNumber": {
           "type": "string",
           "required": true
         },
         "manufacturerId": {
           "type": "string",
           "required": false
         },
         "modelId": {
           "type": "string",
           "required": false
         },
         "rulesetIds": {
           "type": "array",
           "description": "List of identifiers for rulesets supported
               by the device",
           "items": "string",
           "required": false
         }
       }
     }
```

6.8.3. AntennaCharacteristics

AntennaCharacteristics (Section 5.3) provide additional information, such as the antenna height, antenna type, etc.

```
    {
      "name": "AntennaCharacteristics",
      "type": "object",
      "properties": {
        "height": {
          "description": "Height of the antenna, in meters",
          "type": "number",
          "required": false
        },
        "heightType": {
          "description": "Reference type for height:
              Above Ground Level (AGL), or Above Mean Sea
              Level (AMSL).",
          "type": "string",
          "enum": "["AGL","AMSL"]",
          "default": "AGL",
          "required": false
        },
        "heightUncertainty": {
          "description": "Uncertainty of the height measurement,
              in meters.",
          "type": "number",
          "required": false
        }
      }
    }
```

6.8.4. DeviceCapabilities

Device capabilities (Section 5.4) provide additional information that MAY be used by the Device to provide additional information to the Database to help the Database determine available spectrum. If the Database does not support device capabilities, it MUST ignore the parameter.

```
    {
      "name": "DeviceCapabilities",
      "type": "object",
      "description": "Device capabilities to help DB determine
          available spectrum. The DB SHOULD NOT return available
          spectrum that falls outside the given frequency ranges.",
      "properties": {
        "frequencyRanges": {
          "type": "array",
          "items": "FrequencyRange",
          "required": false
        }
      }
```

```
     }
```

6.8.5. DeviceOwner

The DeviceOwner (Section 5.5) parameter contains device-owner information required as part of device registration. Regulatory domains MAY require additional parameters. JSON encoding of vCard is described in jCard: The JSON format for vCard [I-D.ietf-jcardcal-jcard].

```
     {
       "name": "DeviceOwner",
       "type": "object",
       "description": "Device-owner information required as part of
           Device registration. Regulatory domains MAY require
           additional parameters.",
       "properties": {
         "owner": {
           "type": "vCard",
           "description":"Contact information for the individual
               or business that owns the device.",
           "required": true
         },
         "operator": {
           "type": "vCard",
           "description":"Contact information for the device operator.",
           "required": false
         }
       }
     }
```

Example:

```
     {
       "deviceOwner": {
         "owner": [
           "vcard", [
             ["version", {}, "text", "4.0"],
             ["kind", {}, "text", "org"],
             ["fn", {}, "text", "Racafrax, Inc."]
           ]
         ],
         "operator": [
           "vcard", [
             ["version", {}, "text", "4.0"],
             ["fn", {}, "text", "John Frax"],
             ["adr", {}, "text",
               ["", "", "100 Main Street",
                "Summersville", "CA", "90034", "USA"
               ]
             ],
             ["tel", {}, "uri", "tel:+1-213-555-1212"],
             ["email", {}, "text", "j.frax@rackafrax.com"]
           ]
         ]
       }
     }
```

6.8.6. RulesetInfo

RulesetInfo (Section 5.6) contains parameters for the ruleset of a regulatory domain that is communicated using the Initialization component (Section 4.2).

```
     {
       "name": "RulesetInfo",
       "type": "object",
       "description": "The ruleset of a regulatory domain that is
           communicated to Devices in the Initialization Response
           message.",
       "properties": {
         "authority": {
           "type": "string",
           "description": "The regulatory domain at the specified
               location. It is a 2-letter country codes defined by
               ISO3166-1.",
           "required": true
         },
         "rulesetId": {
           "type": "string",
           "description": "The identifier of an applicable ruleset",
           "required": true
         },
         "maxLocationChange": {
           "type": "number",
           "description": "Maximum location change in meters.",
           "required": false
         },
         "maxPollingSecs": {
           "type": "integer",
           "description": "Maximum duration, in seconds, between
               requests for available spectrum.",
           "required": false
         }
       }
     }
```

6.8.7. DbUpdateSpec

DbUpdateSpec (Section 5.7) contains one or more database specifications. It is used to indicate a change to the Database URI.

```
     {
       "name": "DbUpdateSpec",
       "type": "object",
       "description": "Specification for updates to a Database URI",
       "properties": {
         "databases": {
           "type": "array",
           "description": "The specification of one or more databases",
           "item": "DatabaseSpec",
           "required": true
         }
       }
     }
```

6.8.8. DatabaseSpec

DatabaseSpec (Section 5.8) specifies the name and URI of a database.

```
     {
       "name": "DatabaseSpec",
       "type": "object",
       "description": "Specification for a database",
       "properties": {
         "name": {
           "type": "string",
           "description": "The display name of a databases",
           "required": true
         },
         "uri": {
           "type": "string",
           "description": "The URI of a databases",
           "required": true
         }
       }
     }
```

6.8.9. Spectrum

Available Spectrum (Section 5.11) can logically be characterized by a list of SpectrumProfiles, each defining the shape of the permissible power levels over a range of frequencies.

```
     {
       "name": "Spectrum",
       "type": "object",
       "description": "A per-bandwidth list of frequency ranges with
           permissible power levels. For example, In US, FCC
           requires only one spectrum specification at 6MHz
```

```
           bandwidth; ETSI requires two (at 0.1MHz and
           8MHz).",
       "properties": {
         "resolutionBwHz": {
           "type": "number",
           "description": "Resolution bandwidth (Hz) over which
               permissible power spectral densities are defined.",
           "required": true
         },
         "profiles": {
           "type": "array",
           "description": "List of SpectrumProfile objects to specify
               permissible power levels, over a set of frequency ranges,
               for a given resolutionBwHz. The list MAY be empty when
               there is no available spectrum.",
           "items": "SpectrumProfile",
           "required": true
         }
       }
     }

     {
       "name": "SpectrumProfile",
       "type": "array",
       "description": "A list of (frequency, power) points. A minimum of
           two entries is required.",
       "item": "SpectrumProfilePoint",
     }

     {
       "name": "SpectrumProfilePoint",
       "type": "object",
       "description": "A point defined by a frequency and power level
           at that frequency.",
       "properties": {
         "hz": {
           "type": "number",
           "description": "Frequency (Hz)",
           "required": true
         },
         "dbm": {
           "type": "number",
           "description": "Power (dBm) per resolution bandwidth as
               defined by enclosing resolutionBwHz.",
           "required": true
         }
       }
     }
```

```
   Example:
   {
     "resolutionBwHz": 6e6,
     "profiles": [
       [
         {"hz":5.18e8, "dbm":30.0},
         {"hz":5.36e8, "dbm":30.0},
         {"hz":5.36e8, "dbm":36.0},
         {"hz":5.42e8, "dbm":36.0}
       ],
       [
         {"hz":6.20e8, "dbm":30.0},
         {"hz":6.26e8, "dbm":30.0},
       ]
     ]
   }
```

6.8.10. FrequencyRange

The FrequencyRange (Section 5.13) element describes a frequency range and permissible power level within the specified range.

```
     {
       "name": "FrequencyRange",
       "type": "object",
       "properties": {
         "startHz": {
           "type": "number",
           "description": "The inclusive start of the frequency range.",
           "required": true
         },
         "stopHz": {
           "type": "number",
           "description": "The exclusive end of the frequency range.",
           "required": true
         }
       }
     }
```

6.8.11. EventTime

The EventTime (Section 5.14) element specifies the start and stop times of an "event." It is used to indicate the time period for which a Spectrum (Section 5.11) is valid.

```
     {
       "name": "EventTime",
       "type": "object",
       "properties": {
         "startTime": {
           "type": "string",
           "description": "YYYY-MM-DDThh:mm:ssZ RFC3339 format.",
           "format": "date-time",
           "required": false
         },
         "stopTime": {
           "type": "string",
           "description": "YYYY-MM-DDThh:mm:ssZ RFC3339 format.",
           "format": "date-time",
           "required": false
         }
       }
     }
```

6.8.12. SpectrumSchedule

The SpectrumSchedule (Section 5.10) element combines EventTime with Spectrum to define a time period during which the spectrum is valid.

```
     {
       "name": "SpectrumSchedule",
       "type": "object",
       "description": "The SpectrumSchedule element combines EventTime
           with Spectrum to define a time period during which spectrum
           is valid.",
       "properties": {
         "eventTime": {
           "type": "EventTime",
           "description": "Period when the spectra is valid.",
           "required": true
         },
         "spectra": {
           "type": "array",
           "description": "List of available spectra and permissible
               power levels; one spectrum object per resolutionBwHz. The
               list MAY be empty when there is no available spectrum.",
           "items": "Spectrum",
           "required": true
         }
       }
     }
```

6.8.13. SpectrumSpec

The JSON encoding of the Available Spectrum response message AVAIL_SPECTRUM_RESP (Section 4.4.2) is described by the following schema:

```
{
  "name": "SpectrumSpec",
  "type": "object",
  "description": "The SpectrumSpec element represents schedules of
      available spectrum for a regulatory-domain ruleset.",
  "properties": {
    "rulesetInfo": {
      "type": "RulesetInfo",
      "description": "Indicates the active regulatory domain and
          attributes that define the applicable ruleset that
          govern the device",
      "required": false
    },
    "spectrumSchedules": {
      "type": "array",
      "description": "The Database MAY return more than one
          schedule to represent future changes to the available
          spectrum. This array MAY be empty if no spectrum is
          available. If more than one is present, the event-time
          intervals SHOULD be sorted and MUST be disjoint.",
      "items": "SpectrumSchedule",
      "required": true
    },
    "timeRange": {
      "type": "EventTime",
      "description": "The time range for which the spectrumSchedules
          is comprehensive",
      "required": false
    },
    "frequencyRanges": {
      "type": "array",
      "description": "The frequency ranges for which the
          spectrumSchedules is comprehensive",
      "items": "FrequencyRange",
      "required": false
    }
    "needsSpectrumReport": {
      "type": "boolean",
      "description": "For regulatory domains that require a
          spectrum-usage report from devices, the Database MUST
          return true for this parameter.",
      "default": false,
```

```
         "required": false
       },
       "maxTotalBwHz": {
         "type": "number",
         "description": "Constraint on total bandwidth allowed,
             summed across all blocks of spectrum.",
         "required": false
       },
       "maxContiguousBwHz": {
         "type": "number",
         "description": "Constraint on bandwidth allowed for
             any single block of spectrum.",
         "required": false
       }
     }
   }
```

6.8.14. GeoSpectrumSpec

The GeoSpectrumSpec (Section 5.15) element encapsulates the schedule of available spectrum at a location.

```
     {
       "name": "GeoSpectrumSpec",
       "type": "object",
       "description": "The GeoSpectrumSpec element encapsulates
           the schedule of available spectrum at a location.",
       "properties": {
         "location": {
           "type": "GeoLocation",
           "description": "The location at which the spectrum
               schedule applies.",
           "required": true
         },
         "spectrumSpecs": {
           "type": "array",
           "description": "At least one element MUST be included.
               More than one element MAY be included
               to represent available spectrum for more than one
               regulatory domain.",
           "items": "SpectrumSpec",
           "required": true
         }
       }
     }
```

6.8.15. DeviceValidity

The DeviceValidity (Section 5.16) element is used to indicate whether a device is valid. See Section 4.5.2.

```
  {
    "name": "DeviceValidity",
    "type": "object",
    "description": "The DeviceValidity element specifies whether
        the device is valid.",
    "properties": {
      "deviceDesc": {
        "type": "DeviceDescriptor",
        "required": true
      },
      "isValid": {
        "type": "boolean",
        "description": "Boolean that indicates if the Device is
            valid",
        "required": true
      },
      "reason": {
        "type": "string",
        "description": "If the device identifier is not valid,
            the Database MAY include a reason. The reason MAY be
            in any language.",
        "required": false
      }
    }
  }
```

6.8.16. Additional Properties

Note that A JSON Media Type for Describing the Structure and Meaning of JSON [I-D.zyp-json-schema] allows, as default behavior, the inclusion of additional properties by instances that are not explicitly defined in the JSON schema that the instance implements. The schema elaborated in this document adopts this default behavior. Hence, the instance MAY provide additional properties and associated values (which may be "any" JSON type) not explicitly listed in this schema. Further note that the Database and Device MUST ignore any such additional properties and their associated values that it does not understand.

7. HTTPS Binding

This section describes the use of HTTP over TLS (HTTPS) HTTP Over TLS

[RFC2818] as the transport mechanism for the PAWS protocol. TLS provides message integrity and confidentiality between the Master Device and the Database. The Master Device MUST implement server authentication, as described in Section 3.1 of HTTP Over TLS [RFC2818]. The Device uses the URI determined (either statically configured or dynamically discovered) to authenticate the server. The Device SHOULD fail a request if server authentication fails.

Depending on prior relationship between a database and device, the server MAY require client authentication, as described in the Transport Layer Security (TLS) Protocol [RFC5246], to authenticate the device.

To enable databases to handle large numbers of requests from large numbers of devices, the Database MAY support and Devices SHOULD support Stateless TLS Session Resumption [RFC5077].

A PAWS request message is carried in the body of an HTTP POST request. A PAWS response message is carried in the body of an HTTP response. A PAWS response SHOULD include a Content-Length header.

The POST method is the only method REQUIRED for PAWS. If a database chooses to support GET, it MUST be an escaped URL, but the encoding of the URL is outside the scope of this document. The database MAY refuse to support the GET request by returning an HTTP error code, such as 404 (not found).

The Database MAY redirect a PAWS request by returning a HTTP 3xx response (as defined by HTTP/1.1 [RFC2616]). The Database MUST provide the redirect URI in the Location header of the 3xx response, and the Device MUST handle redirects by using the Location header provided by the Database. When redirecting, the Device MUST observe the delay indicated by the Retry-After header. The Device MUST authenticate the Database that returns the redirect response before following the redirect. Also, the Device MUST authenticate the Database indicated in the redirect. Since the Device may communicate with a Database (which it authenticated) without user interaction, when the response code is 301 (Moved Permanently), the Device MAY redirect without asking a user for confirmation (note that this represents an exception to the HTTP/1.1 [RFC2616] requirements for HTTP POST methods).

The Database SHOULD use HTTP status code "307 Temporary Redirect" to indicate that the Device SHOULD resubmit the same request to an alternate URL. The Device MAY revert to the original URL for the very next request, or MAY continue to use the alternate URL for a period of time, e.g.,:

- For the remainder of its session, or
- For a fixed period of time, or
- Until power cycled, or
- Until it receives another redirect

However, it SHOULD NOT modify its stored list of URLs.

The Database SHOULD use HTTP status code "301 Moved Permanently" to indicate that the Device SHOULD resubmit this request, and all future requests, to an alternate URL. If the Device maintains a list of available URLs, it SHOULD replace only the current URL with the alternate URL.

8. Extensibility

8.1. Defining New Message Parameters

New request or response parameters for use with the PAWS protocol are defined and registered in the parameters registry following the procedure in Section 9.1.

Parameter names MUST conform to the param-name ABNF and parameter values syntax MUST be well-defined (e.g., using ABNF, or a reference to the syntax of an existing parameter).

```
param-name = 1*name-char
name-char = ALPHA / DIGIT / "_"
```

The parameter name SHOULD be lowerCamelCase. The name MUST NOT exceed 64 characters.

Unregistered vendor-specific parameter extensions that are not commonly applicable, and are specific to the implementation details of the Database where they are used SHOULD use a vendor-specific prefix that is not likely to conflict with other registered values (e.g., begin with 'companyname').

8.2. Defining Ruleset Identifiers

A ruleset represents a set of device-side requirements for which the device has been certified. It typically corresponds to, but is not limited to, a set of rules that govern a specific set of radio spectrum for a regulatory domain.

Ruleset identifiers are defined and registered in the Ruleset ID Registry following the procedure in Section 9.2. Ruleset ID values MUST conform to the ruleset-id ABNF. If the Ruleset ID requires additional parameters, they MUST be registered in the PAWS Parameters

Registry, as described by Section 9.1.

```
ruleset-id = 1*ruleset-char
ruleset-char = ALPHA / DIGIT / "_" / "."
```

The form of a Ruleset ID value SHOULD be guided by the following:

- o The value SHOULD describe the set of rules that allow a device to operate within one or more regulatory domains. For example, it MAY include the name of a regulatory body or a certification process
- o The value SHOULD include version information, such as a year and/or version number
- o The value MUST NOT exceed 64 characters

8.3. Defining Additional Error Codes

Additional error codes MAY be defined to extend the set listed in Section 5.17. Additional error codes MUST be registered, following the procedures in Section 9.3. If the error code requires additional response parameters, they MUST be registered in the PAWS Parameters Registry, as described by Section 9.1.

By convention, the error code SHOULD be a negative integer value, using one of the range of values defined in Error Codes (Section 5.17). If an appropriate category does not exist, it MAY use values in a different range.

9. IANA Considerations

9.1. PAWS Parameters Registry

This specification establishes the PAWS Parameters Registry.

Additional parameters for inclusion in the PAWS protocol requests, responses, or sub-messages are registered through the Specification Required [RFC5226] process, after a two-week review period on the [paws-iana-TBD]@ietf.org mailing list, on the advice of one or more Designated Experts. To allow for the allocation of values prior to publication, the Designated Expert(s) may approve registration once they are satisfied that such a specification will be published.

Registration requests must be sent to the [paws-iana-TBD]@ietf.org mailing list for review and comment, with an appropriate subject (e.g., "Request for parameter: example"). [[Editor's Note: The name of the mailing list should be determined in consultation with the IESG and IANA. Suggested name: paws-ext-review.]]

Within the review period, the Designated Expert(s) will either approve or deny the registration request, communicating this decision to the review list and IANA. Denials should include an explanation and, if applicable, suggestions as to how to make the request successful.

IANA must only accept registry updates from the Designated Expert(s), and should direct all requests for registration to the review mailing list.

9.1.1. Registration Template

Parameter name: The name of the parameter (e.g., "example").
Parameter usage location: The location(s) where the parameter can be used. The possible locations are the named requests, responses, and messages defined in Protocol Functionalities (Section 4) and Protocol Parameters (Section 5).
Specification document(s): Reference to the document that specifies the parameter, preferably including a URI that can be used to retrieve a copy of the document. An indication of the relevant sections also may be included, but is not required.

9.1.2. Initial Registry Contents

The PAWS Parameters Registry enables protocol extensibility to support any regulatory domain and ruleset. The initial contents of the registry, however, include only FCC-specific and ETSI-specific entries, because, as of this writing, it is the only regulatory domain that has finalized rules. There is no intent to restrict the protocol to FCC rules.

The PAWS Parameters Registry's initial contents are listed below.

FCC ID

Parameter name: fccId
Parameter usage location: DeviceDescriptor (Section 5.2)
Specification document(s): [[this document]] Specifies the device's FCC certification identifier. The value is an identifier string whose length MUST NOT exceed 32 characters. Note that, in practice, a valid FCC ID may be limited to 19 characters, as proposed in FCC Administration Topics Review [FCC-Review-2012-10].

FCC Device Type

Parameter name: fccTvbdDeviceType
Parameter usage location: DeviceDescriptor (Section 5.2)
Specification document(s): [[this document]] Specifies the TV Band White Space device type, as defined by the FCC. Valid values are "FIXED", "MODE_1", "MODE_2".

ETSI Device Type

Parameter name: etsiEnDeviceType
Parameter usage location: DeviceDescriptor (Section 5.2)
Specification document(s): Specifies the White Space Device type, as defined by the ETSI Harmonised Standard [ETSI-EN-301-598]. Valid values are single-letter strings, such as "A", "B", etc. Consult the documentation for details about the device types.

ETSI Device Emissions Class

Parameter name: etsiEnDeviceEmissionsClass
Parameter usage location: DeviceDescriptor (Section 5.2)
Specification document(s): Specifies the White Space Device emissions class, as defined by the ETSI Harmonised Standard [ETSI-EN-301-598], that characterises the out-of-block emissions of the device. The values are represented by numeric strings, such as "1", "2", "3", etc. Consult the documentation for details about emissions classes

ETSI Technology Identifier

Parameter name: etsiEnTechnologyId
Parameter usage location: DeviceDescriptor (Section 5.2)
Specification document(s): Specifies the White Space Device technology identifier, as defined by the ETSI Harmonised Standard [ETSI-EN-301-598]. The string value MUST NOT exceed 64 characters in length. Consult the documentation for valid values.

ETSI Device Category

Parameter name: etsiEnDeviceCategory
Parameter usage location: DeviceDescriptor (Section 5.2)
Specification document(s): Specifies the White Space Device category, as defined by the ETSI Harmonised Standard [ETSI-EN-301-598]. Valid values are the strings, "master" and "slave". It is case insensitive.

9.2. PAWS Ruleset ID Registry

This specification establishes the PAWS Ruleset ID Registry.

Ruleset type names for inclusion in the PAWS protocol messages are registered through the Specification Required [RFC5226] process, after a two-week review period on the [paws-iana-TBD]@ietf.org mailing list, on the advice of one or more Designated Experts. To allow for the allocation of values prior to publication, the Designated Expert(s) may approve registration once they are satisfied that such a specification will be published.

Registration requests must be sent to the [paws-iana-TBD]@ietf.org mailing list for review and comment, with an appropriate subject (e.g., "Request for parameter: example"). [[Editor's Note: The name of the mailing list should be determined in consultation with the IESG and IANA. Suggested name: paws-ext-review.]]

Within the review period, the Designated Expert(s) will either approve or deny the registration request, communicating this decision to the review list and IANA. Denials should include an explanation and, if applicable, suggestions as to how to make the request successful.

IANA must only accept registry updates from the Designated Expert(s), and should direct all requests for registration to the review mailing list.

9.2.1. Registration Template

Ruleset name: The name of the ruleset. The length of the string MUST NOT exceed 64 characters.

Additional message parameters: Additional parameters to associate with the rulesetId parameter. New parameters MUST be registered separately in the PAWS Parameters Registry, as described by Section 8.1.

Specification Document(s): Reference to the document that specifies the parameter, preferably including a URI that can be used to retrieve a copy of the document. An indication of the relevant sections also may be included, but is not required.

9.2.2. Initial Registry Contents

The PAWS Ruleset ID Registry enables protocol extensibility to support any regulatory domain and ruleset. The initial contents of the registry, however, include only FCC-specific and ETSI-specific entries, because, as of this writing, it is the only regulatory domain that has finalized rules. There is no intent to restrict the protocol to FCC rules.

The initial contents of the PAWS Ruleset ID Registry are listed below.

9.2.2.1. Federal Communications Commission (FCC)

For the additional parameters that start with the "fcc" prefix, see PAWS Parameters Registry Initial Contents (Section 9.1.2) for more information.

Ruleset name: FccTvBandWhiteSpace-2010
Additional message parameters:
- deviceDesc: The DeviceDescriptor (Section 5.2) parameter is REQUIRED for Available Spectrum Request (Section 4.4.1) and Available Spectrum Batch Request (Section 4.4.3).
- fccId: Specifies a device's FCC certification ID. It is a required parameter in DeviceDescriptor (Section 5.2).
- fccTvbdDeviceType: Specifies the type of TV-band White Space device, as defined by the FCC rules. It is a required parameter in DeviceDescriptor (Section 5.2).

Specification document(s): [[this document]] This ruleset refers to the FCC rules for TV-band White Space operations established in the Code of Federal Regulations (CFR), Title 47, Part 15, Subpart H [FCC-CFR47-15H].

Additional DeviceOwner (Section 5.5) requirements:

owner: The owner vCard [RFC6350] entry MUST include the formatted name of an individual or organization using the "fn" property. When the name is that of an organization, the entry also MUST include the "kind" property, with a value of "org".

operator: The operator vCard [RFC6350] entry MUST include the following properties:
- fn: Formatted name of a contact person responsible for the device's operation.
- adr: Address for the contact person.
- tel: Phone number for the contact person.
- email: E-mail address for the contact person.

9.2.2.2. European Telecommunications Standards Institute (ETSI)

For the additional parameters that start with the "etsi" prefix, see PAWS Parameters Registry Initial Contents (Section 9.1.2) for more information.

Ruleset name: ETSI-EN-301-598-1.0.0-draft
Additional message parameters:
- manufacturerId: Specifies a device's manufacturer's identifier. It is a REQUIRED parameter in DeviceDescriptor (Section 5.2).

```
      modelId:  Specifies a device's model identifier.  It is a REQUIRED
         parameter in DeviceDescriptor (Section 5.2).
      etsiEnDeviceType:  Specifies the device's ETSI device type.  It is
         a REQUIRED parameter in DeviceDescriptor (Section 5.2).
      etsiEnDeviceEmissionsClass:  Specifies the device's ETSI device
         emissions class.  It is a REQUIRED parameter in
         DeviceDescriptor (Section 5.2).
      etsiEnTechnologyId:  Specifies the device's ETSI technology ID.
         It is a REQUIRED parameter in DeviceDescriptor (Section 5.2).
      etsiEnDeviceCategory:  Specifies the device's ETSI device
         category.  It is a REQUIRED parameter in DeviceDescriptor
         (Section 5.2).
      requestType:  Modifies the available-spectrum request type.  It is
         an OPTIONAL parameter in the AVAIL_SPECTRUM_REQ (Section 4.4.1)
         and AVAIL_SPECTRUM_BATCH_REQ (Section 4.4.3) messages.  If
         specified, the only valid value is, "Generic Slave", and the
         Database MUST respond with generic operating parameters for any
         Slave Device.
      needsSpectrumReport  Depending on the regulatory domain, the
         Database MAY be required to set this value to true in the
         AVAIL_SPECTRUM_RESP (Section 4.4.2) and
         AVAIL_SPECTRUM_BATCH_RESP (Section 4.4.4) messages.
      maxTotalBwHz:  Specifies a constraint on total allowed bandwidth.
         It is a REQUIRED parameter in AVAIL_SPECTRUM_RESP
         (Section 4.4.2) and AVAIL_SPECTRUM_BATCH_RESP (Section 4.4.4).
      maxContiguousBwHz:  Specifies a constraint on total allowed
         contiguous bandwidth.  It is a REQUIRED parameter in
         AVAIL_SPECTRUM_RESP (Section 4.4.2) and
         AVAIL_SPECTRUM_BATCH_RESP (Section 4.4.4).
      maxLocationChange:  Specifies a constraint on maximum location
         changes.  It is a REQUIRED parameter in AVAIL_SPECTRUM_RESP
         (Section 4.4.2) and AVAIL_SPECTRUM_BATCH_RESP (Section 4.4.4).
   Specification document(s):  This ruleset refers to the ETSI
      Harmonised Standard [ETSI-EN-301-598] established by ETSI.
```

9.3. PAWS Error Code Registry

This specification establishes the PAWS Error Code Registry.

Additional error codes for inclusion in the PAWS protocol error message are registered through the Specification Required [RFC5226] process, after a two-week review period on the [paws-iana-TBD]@ ietf.org mailing list, on the advice of one or more Designated Experts. To allow for the allocation of values prior to publication, the Designated Expert(s) may approve registration once they are satisfied that such a specification will be published.

Registration requests must be sent to the [paws-iana-TBD]@ietf.org

mailing list for review and comment, with an appropriate subject (e.g., "Request for parameter: example"). [[Editor's Note: The name of the mailing list should be determined in consultation with the IESG and IANA. Suggested name: paws-ext-review.]]

Within the review period, the Designated Expert(s) will either approve or deny the registration request, communicating this decision to the review list and IANA. Denials should include an explanation and, if applicable, suggestions as to how to make the request successful.

IANA must only accept registry updates from the Designated Expert(s), and should direct all requests for registration to the review mailing list.

9.3.1. Registration Template

Code: Integer value of the error code.
Name: Name of the error.
Additional parameters: Additional parameters that are returned in the data portion of the error (See Section 5.17). New parameters MUST be registered separately in the PAWS Parameters Registry, as described by Section 9.1.
Description: Description of the error and its associated parameters, if any.

9.3.2. Initial Registry Contents

Initial registry contents are defined in the Table of Error Codes (Table 1).

10. Security Considerations

PAWS is a protocol whereby a Master Device requests a schedule of available spectrum at its location (or location of its Slave Devices) before it (they) can operate using those frequencies. Whereas the information provided by the Database must be accurate and conform to applicable regulatory rules, the Database cannot enforce, through the protocol, that a client device uses only the spectrum it provided. In other words, devices can put energy in the air and cause interference without asking the Database. Hence, PAWS security considerations do not include protection against malicious use of the White Space spectrum. For more detailed information on specific requirements and security considerations associated with PAWS, see Protocol to Access White Space database: PAWS Use Cases and Requirements [RFC6953].

By using the PAWS protocol, the Master Device and the Database expose themselves to the following risks:

- o Accuracy: The Master Device receives incorrect spectrum-availability information.
- o Privacy: An unauthorized entity intercepts identifying data for the Master Device or its Slave Devices, such as serial number and location.

Protection from these risks depends on the success of the following steps:

1. The Master Device must determine the address of a proper database.
2. The Master Device must connect to the proper database.
3. The Database must determine or compute accurate spectrum-availability information.
4. PAWS messages must be transmitted unmodified between the Database and the Master Device.
5. PAWS messages must be encrypted between the Database and the Master Device to prevent exposing private information.
6. For a Slave Device, the spectrum-availability information also must be transmitted unmodified and secure between the Master Device and the Slave Device.

Of these, only steps 1, 2, 4, and 5 are within the scope of this document. This document addresses Step 1 by allowing static provisioning of one or more trusted Databases; dynamic provisioning is out of scope. Step 3 dependent on specific database implementations and regulatory rules and is outside the scope of this document. Step 6 requires a protocol between master and slave devices and is thus outside the scope of this document.

10.1. Assurance of Proper Database

This document assumes that the Database is contacted using a domain name or an IP address. Using HTTP over TLS HTTP Over TLS [RFC2818], the Database authenticates its identity, either as a domain name or IP address, to the Master Device by presenting a certificate containing that identifier as a "subjectAltName" (i.e., as a dNSName or IP address). If the Master Device has external information as to the expected identity or credentials of the proper database (e.g., a certificate fingerprint), these checks MAY be omitted. Note that in order for the presented certificate to be valid at the client, the client must be able to validate the certificate. In particular, the validation path of the certificate must end in one of the client's trust anchors, even if that trust anchor is the Database certificate itself. A Master Device should allow for the fact that a Database can change its certificate authorities (CAs) over time.

10.2. Protection Against Modification

To prevent a PAWS response message from being modified en route, messages must be transmitted over an integrity-protected channel. Using HTTP over TLS, the channel will be protected by appropriate cypher suites.

10.3. Protection Against Eavesdropping

Using HTTP over TLS, messages protected by appropriate cypher suites are also protected from eavesdropping or otherwise access by unauthorized parties en route.

10.4. Client Authentication Considerations

Although the Database can inform a device of available spectrum it can use, the Database cannot enforce that the Master Device uses any/only those frequencies. Indeed, a malicious device can operate without ever contacting a database. Consequently, client authentication is not required for the core PAWS protocol (although it may be required by specific regulators). Depending on a prior relationship between a Database and Master Device, the Database MAY require client authentication. TLS provides client authentication, but there are some considerations:

- As indicated in Section 3.2 of HTTP Over TLS [RFC2818], the TLS client authentication procedure only determines that the device has a certificate chain rooted in an appropriate CA (or a self-signed certificate). The database would not know what the client identity ought to be, unless it has some external source of information. Distribution and management of such information, including revocation lists, are outside the scope of this document.
- Authentication schemes are secure only to the extent that secrets or certificates are kept secure. When there are a vast number of deployed devices using PAWS, the possibility that device keys will not leak becomes small. Implementations should consider how to manage the system in the eventuality that there is a leak.

11. Contributors

This document draws heavily from the following Internet Draft documents, [I-D.das-paws-protocol] and [I-D.wei-paws-framework]. The editor would like to specifically call out and thank the contributing authors of these two documents.

Donald Joslyn
Spectrum Bridge Inc.
1064 Greenwood Blvd.
Lake Mary, FL 32746
U.S.A.
Email: d.joslyn at spectrumbridge dot com

Xinpeng Wei
Huawei
Phone: +86 13436822355
Email: weixinpeng@huawei.com

12. Acknowledgments

The authors gratefully acknowledge the contributions of: Gabor Bajko, Ray Bellis, Teco Boot, Nancy Bravin, Rex Buddenberg, Gerald Chouinard, Stephen Farrell, Michael Fitch, Joel M. Halpern, Daniel Harasty, Michael Head, Jussi Kahtava, Warren Kumari, Kalle Kulsmanen, Paul Lambert, Andy Lee, Anthony Mancuso, Basavaraj Patil, Scott Probasco, Brian Rosen, Andy Sago, Peter Stanforth, John Stine, and Juan Carlos Zuniga.

13. References

13.1. Normative References

[FCC-CFR47-15H]
U. S. Government, "Electronic Code of Federal Regulations, Title 47, Part 15, Subpart H: Television Band Devices", December 2010, <http://www.ecfr.gov/cgi-bin/text-idx?rgn=div6&view=text&node=47:1.0.1.1.16.8>.

[I-D.ietf-jcardcal-jcard]
Kewisch, P., "jCard: The JSON format for vCard", draft-ietf-jcardcal-jcard-07 (work in progress), October 2013.

[ITUT.X520.2008]
International Telecommunication Union, "ITU-T Recommendation X.520: Information technology - Open Systems Interconnection - The Directory: Selected attribute types", November 2008, <http://www.itu.int/rec/T-REC-X.520-200811-I>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, January 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5491] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV Presence Information Data Format Location Object (PIDF-LO) Usage Clarification, Considerations, and Recommendations", RFC 5491, March 2009.

[RFC6350] Perreault, S., "vCard Format Specification", RFC 6350, August 2011.

[WGS-84] National Imagery and Mapping Agency, "Department of Defense World Geodetic System 1984, Its Definition and Relationships with Local Geodetic Systems, NIMA TR8350.2 Third Edition Amendment 1", January 2000, <http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html>.

13.2. Informative References

[ETSI-EN-301-598]
European Telecommunication Standards Institute (ETSI), "Draft ETSI EN 301 598 (V1.0.0): White Space Devices (WSD); Wireless Access Systems operating in the 470 MHz to 790 MHz frequency band; Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive", July 2013, <http://www.etsi.org/deliver/etsi_en/301500_301599/301598/01.00.00_20/en_301598v010000a.pdf>.

[FCC-Review-2012-10]
Federal Communications Commission, "Administration Topics Review", October 2012, <http://transition.fcc.gov/bureaus/oet/ea/presentations/files/oct12/2b-TCB-Admin-Issues-Oct-2012-GT.pdf>.

[I-D.das-paws-protocol]
Das, S., Malyar, J., and D. Joslyn, "Device to Database Protocol for White Space", draft-das-paws-protocol-02 (work in progress), July 2012.

[I-D.wei-paws-framework]
Wei, X., Zhu, L., and P. McCann, "PAWS Framework", draft-wei-paws-framework-00 (work in progress), July 2012.

[I-D.zyp-json-schema]
Galiegue, F., Zyp, K., and G. Court, "JSON Schema: core definitions and terminology", draft-zyp-json-schema-04 (work in progress), January 2013.

[ISO3166-1]
"Country Codes", <http://www.iso.org/iso/country_codes.htm>.

[JSON-RPC]
"JSON-RPC 2.0 Specification", <http://www.jsonrpc.org/specification>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

[RFC6953] Mancuso, A., Probasco, S., and B. Patil, "Protocol to Access White-Space (PAWS) Databases: Use Cases and Requirements", RFC 6953, May 2013.

Appendix A. Changes / Author Notes.

Changes from 07:

o Propose ruleset ID name for ETSI: ETSI-EN-301-598-1.0.0-draft

o Change TBD email address to paws-iana-TBD@ietf.org for proposing changes to the PAWS IANA registries

o Moved discussion of required vCard properties to regulatory-specific sections

- Fixed vCard examples for organization names: Use "fn" property, but set "kind" to "org".
- Shorten parameter names:
  * freqHz -> hz
  * powerDbmPerBw -> dbm

Changes from 06:

- Multi-ruleset support:
  * Changed RulesetInfo to have single ruleset ID
  * Changed INIT_RESP to return a list of RulesetInfo parameters, rather than a single one
  * Changed REGISTRATION_RESP to return a list of RulesetInfo parameters to indicate the regulatory domains for which registration was accepted
  * Added SpectrumSpec (Section 5.9) parameter to represent available-spectrum specification for one regulatory domain, allowing AVAIL_SPECTRUM_RESP and AVAIL_SPECTRUM_BATCH_RESP to include answers for multiple regulatory domains
  * Changed GeoSpectrumSchedule to GeoSpectrumSpec (Section 5.15) for supporting batch responses to represent available spectrum for multiple regulatory domains at a location
- To avoid ambiguity or redundant information, clarified that:
  * Event-time intervals within a single set of schedules MUST be disjoint
  * A single Spectrum element MUST cover the entire range of frequencies governed by a ruleset, rather than splitting them to present a "channelized" view
- Add "ruleset" to Terminology section
- Sync Terminology section with Use Case document
- Add "masterDeviceDesc" to Device Validate request
- Add "masterDeviceLocation" to the AVAIL_SPECTRUM requests and the SPECTRUM_NOTIFY message. Change "location" to be the location of the Slave Device, if the request is made by a Master Device on behalf of a Slave Device
- Update vCard reference and example
- Add jsonrpc 2.0 to all sample messages
- Clarify that Listing Servers may be preconfigured in a device
- Clarify meaning of maximum power levels vs bandwidth, including renaming parameter names:
-
  * maxPowerDBm -> powerDbmPerBw
  * bandwidth -> resolutionBwHz
- Explicitly allowed generic JSON-RPC error codes as possible codes.
- Replace SHALL with MUST for consistency
- Replace URI with URL for consistency
- Reduce clutter in JSON encoding examples by removing string-concatenation characters

- o Changed "depends" to "depends on regulatory rules" in several places

Changes from 05:

- o Remove requirement for JSON-RPC 1.0
- o More typo fixes and clarifications

Changes from 04:

- o Add "masterDeviceDesc" parameter to the available-spectrum requests to allow both Master and Slave device descriptors when the Master is making the request on behalf of a Slave.
- o Add "requestType" parameter to the available-spectrum requests to support requesting generic operating parameters for any Slave Device.
- o Add DbUpdateSpec as optional parameter to all response messages and to the error response to allow a Device to detect a database change at any stage of the control flow.
- o For the OUTSIDE_COVERAGE error, added ability to return a list of alternate databases
- o Explicitly allow JSON-RPC v2.0 and v1.0 encodings
- o Relaxed language that state, "MUST stop operation" to "MUST cease use of spectrum under rules for database-managed spectrum". I.e., the device may have other fallback strategies allowed by regulators.

Changes from 03:

- o Expanded the Database Discovery mechanism to describe in more detail pre-configuration with URLs of databases and database-listing servers, including mechanisms for updating the configurations when things change
  - * Add database-change field to Available Spectrum Response (Section 4.4.2)
- o Added fields that are anticipated to be needed by the ETSI harmonized standard for White Space Devices:
  - * Added bandwidth constraints to the Available Spectrum Response (Section 4.4.2)
  - * Updated Available Spectrum Response to return RulesetInfo, rather than just a ruleset identifier
  - * Added optional device-manufacturer and device-model IDs to the DeviceDescriptor (Section 5.2) message. Also moved fccId from this message to the IANA section.
  - * Expanded IANA (Section 9) sections
- o Clarified restrictions on the specification of the vertices of a Polygon.

- Changed default confidence level to 95% for a point with uncertainty
- Clarified how devices without absolute time source can use the timestamps in the response messages
- Change method names to start with "spectrum.paws." prefix
- Added maximum string lengths
- Updated author contact info
- More typo fixes

Changes from 02:

- Added timestamp to the AVAIL_SPECTRUM_RESP (Section 4.4.2) and AVAIL_SPECTRUM_BATCH_RESP (Section 4.4.4) data models to serve as a reference for the event times in the response. This was accidentally omitted (but was specified in their JSON encodings (Section 6)).
- Fixed typos throughout the JSON encoding (Section 6) sections, typically adding missing commas.

Changed from 01:

- Added a description of message sequences to support multiple rulesets and multiple jurisdictions Section 3.1.
- Modified DeviceDescriptor (Section 5.2) to add rulesetIds parameter
- Modified RulesetInfo (Section 5.6), AvailableSpectrumResponse (Section 4.4.2) to add rulesetId parameter.
- Add Extensibility (Section 8) section.
- Filled in IANA (Section 9) section.
- Removed blank Example Messages section

Changes from 00:
- Add JSON encoding
- Adopt RFC5491 for GeoLocation
- Adopt vCard for contact information
- Add Response Code section and update text referencing the defined response codes
- Change DeviceIdentifier to be DeviceDescriptor, allowing identifiers and device-characteristic fields to be included.

Authors' Addresses

Vincent Chen (editor)
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: vchen@google.com

Subir Das
Applied Communication Sciences
150 Mount Airy Road
Basking Ridge, NJ 07920
U.S.A.

Phone:
Fax:
Email: sdas at appcomsci dot com
URI:

Lei Zhu
Huawei

Phone: +86 13910157020
Fax:
Email: lei.zhu@huawei.com
URI:

John Malyar
iconectiv (formerly Telcordia Interconnection Solutions)
444 Hoes Lane/RRC 4E1106
Piscataway, NJ 08854
U.S.A.

Phone:
Fax:
Email: jmalyar at iconectiv dot com
URI:

Peter J. McCann
Huawei
400 Crossing Blvd, 2nd Floor
Bridgewater, NJ 08807
USA

Phone: +1 908 541 3563
Fax:
Email: peter.mccann@huawei.com
URI: