Signaling Protocols and Procedures for Citizens Broadband Radio Service (CBRS): Spectrum Access System (SAS) - SAS Interface Technical Specification

Document WINNF-TS-0096

Version 1.3.1

29 January 2019

TERMS, CONDITIONS & NOTICES

This document has been prepared by the SSC Work Group 3 to assist The Software Defined Radio Forum Inc. (or its successors or assigns, hereafter "the Forum"). It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the SSC Work Group 3.

Contributors to this document that have submitted copyrighted materials (the Submission) to the Forum for use in this document retain copyright ownership of their original work, while at the same time granting the Forum a non-exclusive, irrevocable, worldwide, perpetual, royalty-free license under the Submitter's copyrights in the Submission to reproduce, distribute, publish, display, perform, and create derivative works of the Submission based on that original work for the purpose of developing this document under the Forum's own copyright.

Permission is granted to the Forum's participants to copy any portion of this document for legitimate purposes of the Forum. Copying for monetary gain or for other non-Forum related purposes is prohibited.

THIS DOCUMENT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS DOCUMENT.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

This document was developed following the Forum's policy on restricted or controlled information (Policy 009) to ensure that that the document can be shared openly with other member organizations around the world. Additional Information on this policy can be found here: http://www.wirelessinnovation.org/page/Policies_and_Procedures.

Although this document contains no restricted or controlled information, the specific implementation of concepts contain herein may be controlled under the laws of the country of origin for that implementation. Readers are encouraged, therefore, to consult with a cognizant authority prior to any further development.

Wireless Innovation Forum ™ and SDR Forum ™ are trademarks of the Software Defined Radio Forum Inc.

Table of Contents

List of Figures

List of Tables

Contributors

The following individuals made significant contributions to this document:

Editors: James Ni (Federated Wireless), Greg Billock (Google), Yi Hsuan (Google), Sho Furuichi (Sony)

Other Member Representatives:

- AT&T: Neeti Tandon
- Ericsson: Kumar Balachandran
- Federated Wireless: Masoud Olfat
- Google: Greg Billock
- Key Bridge Global: Jesse Caulfield
- Motorola Solutions: David Gurney
- Nokia: Mike Dolan, Al Hirsbrunner, Steve Magee, Prakash Moorut
- Qualcomm: Doug Knisely
- Sony Corporation: Naotaka Sato, Sho Furuichi
- Verizon: Max Solondz, Naseem Khan

Signaling Protocols and Procedures for Citizens Broadband Radio Service (CBRS): Spectrum Access System (SAS) - SAS Interface Technical Specification

1 Scope

This document is a Technical Specification of a signaling protocol and procedures for the SAS-SAS interface (See [i.1]).

The key words "required", "shall", "shall not", "should", "should not", "recommended", "may", and "optional" in this document are to be interpreted as described in RFC 2119 [n.2]. In addition, the key word “conditional” shall be interpreted to mean that the definition is an absolute requirement of this specification only if the stated condition is met.

2 References

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[n.1] CBRS Communications Security Technical Specification, WINNF-TS-0065, Version V1.1.0

[n.2] RFC-2119, “Key words for use in RFCs to Indicate Requirement Levels”, March 1997

[n.3] RFC-5246, “The Transport Layer Security (TLS) Protocol Version 1.2”, Dierks and Rescorla, August 2008

[n.4] RFC-2818, HTTP Over TLS, Rescorla, May 2000

[n.5] RFC-5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Cooper, Santesson, Farrell, Boeyen, Housley & Polk, May 2008

[n.6] RFC-2616, Hypertext Transfer Protocol -- HTTP/1.1, Fielding, Gettys, Mogul, Frystyk, Masinter, Leach and Berners-Lee, June 1999

[n.7] RFC-7159, "The JavaScript Object Notation (JSON) Data Interchange Format", March 2014

[n.8] “Requirements for Commercial Operation in the U.S. 3550-3700 MHz Citizens Broadband Radio Service Band”, Wireless Innovation Forum WINNF-TS-0112

[n.9] RFC-3339, "Date and Time on the Internet: Timestamps", July 2002

[n.10] “Signaling Protocols and Procedures for Citizens Broadband Radio Service (CBRS): Spectrum Access System (SAS) - Citizens Broadband Radio Service Device (CBSD) Interface Technical Specification”, WINNF-TS-0016.

[n.11] Electronic Code of Federal Regulations, Title 47, Chapter I, Subchapter D, Part 96, http://www.ecfr.gov/cgi-bin/retrieveECFR?gp=&SID=0076fe7586178336d9db4c5146da8797&mc=true&n=pt47.5.96&r=PART&ty=HTML.

[n.12] SAS-SAS Information Sharing Framework, Wireless Innovation Forum WINNF-16-I-0093

[n.13] RFC-7946, The GeoJSON Format, Butler et. al., August 2016.

[n.14] "CBRS PKI Certificate Policy", WINNF-TS-0022

[n.15] "Operations for Citizens Broadband Radio Service (CBRS): Priority Access Licensee (PAL) Database Technical Specification", WINNF-TS-0245

[n.16] NGA.STND.0036_1.0.0_WGS84 (Version 1.0.0 - July 8, 2014): Department of Defense (DoD) World Geodetic System (WGS) 1984.

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] "SAS Functional Architecture", Wireless Innovation Forum Spectrum Sharing Committee, 7 Sep 2015, WINNF-15-P-0047-V1.0.0.

[i.2] VOID

[i.3] VOID

[i.4] 47 CFR §2.969 (see http://www.ecfr.gov/cgi-bin/text-idx?node=se47.1.2_1926)

[i.5] RFC-4627, The application/json Media Type for JavaScript Object Notation (JSON), Crockford, July 2006

[i.6] JSON Schema, http://json-schema.org/

[i.7] Report and Order and Second Further Notice of Proposed Rulemaking, Amendment of the Commission's Rules with Regard to Commercial Operations in the 3550-3650 MHz Band, GN Docket No. 12-354, Federal Communications Commission, April 21, 2015.

[i.8] Order on Reconsideration and Second Report and Order, Amendment of the Commission's Rules with Regard to Commercial Operations in the 3550-3650 MHz Band, GN Docket No. 12-354, Federal Communications Commission, May 2, 2016.

3 Definitions and abbreviations

CA	Certificate Authority
CBSD	Citizens Broadband Radio Service Device

ESC	Environmental Sensing Capability
FCC	Federal Communications Commission
FRN	FCC Registration Number
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP (e.g. with TLS)
ID	Identifier
JSON	JavaScript Object Notation
PAL	Priority Access License
PPA	PAL Protection Area
SAS	Spectrum Access System
TLS	Transport Layer Security
URL	Universal Resource Locator
UTC	Coordinated Universal Time

4 Description of SAS-SAS Prerequisites

This section provides a high-level view of the prerequisites to SAS-SAS message exchange. Note: this section is informative.

4.1 Pre-requisite Procedures

Before commencement of SAS-SAS communications, several procedures need to be implemented and performed. Details of these procedures are not within the scope of this document. Purposes and high-level functions of these procedures are described below.

1. Communication security. A security framework is followed so that SASs can verify each other's identity and trust the information exchanged through the SAS-SAS interface. [n.1]

2. Data use restrictions agreements. Because the SAS-SAS exchange carries SAS-Essential Data as defined in [n.8, n.12], a SAS Administrator which enables its SAS to exchange information with another SAS must establish a use restriction agreement with the Administrator of the peer SAS governing uses of the data exchanged. Note: A sample use restriction agreement can be located through the Wireless Innovation Forum. This sample agreement is neither endorsed nor approved by the Wireless Innovation Forum.

4.2 Peer SAS Discovery

SAS Administrators can register SAS implementations which form an initial peer set for SAS-SAS exchange. This process includes the URL endpoints for their SASs, allowing a particular SAS implementation to contact the SAS-SAS interface provided by peer SASs.

1. Peer SAS discovery involves the methods which a SAS Administrator uses to configure a SAS implementation with parameters needed to contact peer SASs.

2. Both static and dynamic methods may be supported.

3. Dynamic provisioning may leverage existing protocols like Domain Name System (DNS)/ Dynamic Host Control Protocol (DHCP) to determine the SAS connection information.

5 SAS-SAS Procedures

5.1 SAS Mutual Authentication and Communications Security

SAS mutual authentication and communications security shall conform to the Wireless Innovation Forum Communications Security specification [n.1].

The authentication procedure is initiated by a SAS attempting to communicate with another SAS. TLS-v1.2 as specified in [n.3] shall be used to perform authentication. Previous versions of TLS (e.g., TLS-v1.1 per RFC-4346, TLS-v1.0 per RFC-2246 or SSL-v3.0) shall not be used. During the TLS exchange, mutual authentication shall be performed.

Server certificate validation shall be performed according to the procedures in [n.3]. A SAS which is unable to successfully authenticate a peer SAS shall abort the TLS connection establishment procedure. It is implementation specific whether a SAS is required to re-attempt communications.

During the TLS message exchange, the SAS provides its client certificate to the peer SAS. Both SASs shall perform certificate validation according to the procedures in [n.3]. A SAS which is unable to successfully authenticate a peer shall abort the TLS connection establishment procedure.

5.1.1 TLS Encryption

Subsequent to successful authentication, the SAS shall negotiate a ciphersuite to use for encrypting all communications between the two entities. The ciphersuite shall be selected from the following list (ref. [n.1]):

- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

A SAS which is unable to successfully setup such an encrypted connection with a peer SAS shall abort the TLS connection establishment procedure. Procedures to be followed on failure of authentication, including further attempts, are left to the discretion of the SAS administrator.

This authentication procedure shall be followed before any message exchange between two SASs.

5.2 Record Exchanges

The SAS-SAS Protocol is built upon the exchange of data recorded by one SAS and communicated to another SAS. Such communication may happen on a state change in a particular entity (e.g., a Citizens Broadband Radio Service Device (CBSD) given a frequency grant, an enforcement action taken by the FCC, or a change in incumbent activity learned by a SAS), or upon request by an authorized peer SAS (e.g., when a new SAS enters the peer group, or recovers from a service outage and requests incremental information).

The format of the SAS-SAS protocol imposes no constraints on the data storage of any SAS implementation: it is strictly an exchange format for metadata related to particular entities known to the SAS and about which information is exchanged to achieve the functional objectives required to be performed by the SAS and the SAS administrator.

As subjects of information exchange, each record is referred to by a globally unique record ID within the group of peer SASs conforming to this specification. Such record IDs will be constructed of a sequential list of tokens, allowing for the SAS namespace to include presently-existing namespaces when possible for maximum interoperability with existing naming schemes.

These token sequences are represented using a '/' separator character. So for example, a particular CBSD record ID is represented as "cbsd/$CBSD_REFERENCE_ID". In this document, the symbol "$" before any token refers to a token chosen by the entity issuing the token.

5.2.1 SAS-SAS exchange entities and IDs

Table 1: SAS-SAS Protocol exchange entities

CBRS Entity	Description
SAS Administrators	Manage specific SAS implementations. There may be many implementations maintained by a single SAS administrator. IDs will be of the form "sas_admin/$ADMINISTRATOR_ID" where the second token is chosen by the administrator and verified to be unique in the Certificate Authority (SAS-CA) issuance process for the SAS (See [n.14]).

CBRS Entity	Description
SAS Implementations	A particular SAS implementation. IDs will be of the form "sas_impl/$ADMINISTRATOR_ID/$SAS_IMPLEMENTATION" where the second token is the ID of the administrator and the third token is chosen by the SAS administrator and verified uniquely by the SAS-CA issuance process.
CBSDs	Specific devices which will operate in the CBRS band and gain spectrum use authorizations from the SAS. IDs will be of the form "cbsd/$CBSD_REFERENCE_ID" where the second token is a reference token unique to that CBSD. (See 8.3)
ESC Sensors	Specific Environmental Sensing Capability (ESC) Sensors which need protection as part of their function of informing the presence of federal incumbents. IDs will be of the form "esc_sensor/$ADMINISTRATOR_ID/$SENSOR_ID" where the second token is the ID of the SAS administrator and the third token is chosen by the administrator.
Zones	Geographical areas with various meanings within the SAS. For example, census tracts (Priority Access License (PAL) license areas), Grandfathered Part 90 protection areas, ad hoc protection zones, etc. Such zones will have IDs of the form "zone/$CREATOR/$ZONE_ID" where the second token takes on values representing the SAS administrator responsible for the creation of the zone, or a predefined token related to static government-sourced information defining geographical zones. The third token takes on values assigned by the creator or assigned using a pre-existing namespace.
Coordination events	Part 96 rules require formal exchange of information regarding a wide range of events, such as GAA/FSS arrangements, PAL-to-PAL arrangements, enforcement actions by FCC, etc. IDs for particular coordination events will be assigned by the SAS administrator in which the coordination event is registered and follow the format "coordination/$ADMINISTRATOR_ID/$EVENT_ID" where the second token is equivalent to a valid SAS administrator ID and the third token is chosen uniquely by that SAS administrator within this namespace.

These entities compose the full range of entities about which the SAS is required to exchange information. For each entity, the SAS-SAS protocol defines a set of information which the SAS

possesses about such an entity and which is the full set required to be exchanged in order to fulfill its function.

5.3 Message flow

The message exchanges between two SASs are based on client-server request and response flows. SASs symmetrically issue requests to their respective peer SASs independently and the peer SASs respond with either success or error responses (Pull Type). A SAS may initiate exchanging data with its respective peer SASs, without a request from the peer SASs (Push Type).

Message exchanges between two SASs are shown in Figure 1 below.

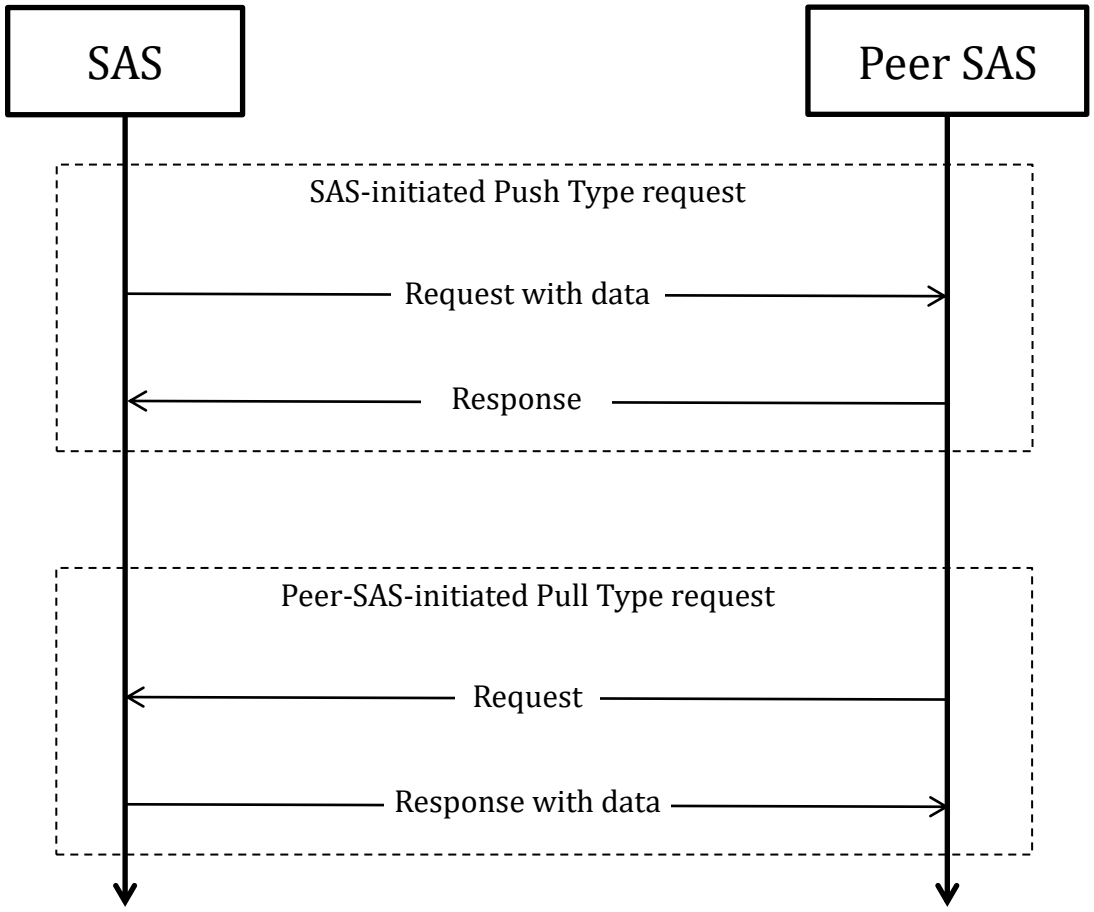

Figure 1: SAS to SAS Exchange Flow, Push and Pull types

6 SAS-SAS Synchronization

6.1 Time-range request support requirements

A SAS employing this protocol may periodically make time-range record requests to every peer SAS for the following record types as necessary for its operations:

- CBSD
- Zone
- Coordination Event

SASs shall respond to such requests using their own system time basis for response. For example, a SAS receiving a request for a time range from 4pm to 5pm will reply with all qualifying records modified (by the SAS receiving the request) in that time range.

SASs responding to time-range requests and for which the following constraints are met shall reply with a complete set of records meeting the request parameters [See Table 2]:

- the “start_time” and “end_time” parameters are within 25 hours of each other;
- the “start_time” parameter represents a value no older than 30 days previous to the present in the time reference of the SAS responding to the request;
- The “end_time” parameter represents a time which is at least 60 seconds in the past in the time reference of the SAS responding to the request.
- All time values shall be in UTC.

If the SAS responding to the request is unable to supply the corresponding data, it shall reply with the appropriate HTTP status code indicating temporary (503) or permanent (500) inability to comply with such a request.

If request parameters are received which are outside the acceptable ranges and the responding SAS does not supply the corresponding data, it shall reply with a 400 HTTP status code.

If the requested parameter range results in a data transfer larger than 50MB, it should return a 416 HTTP status code to signify that the requesting SAS should create requests with smaller time ranges.

> NOTE: These requirements enable each SAS to maintain a “high-water mark” of exchanged records with its peer SASs for these record types. Consequently, each SAS can maintain a representation of the CBSDs, qualifying exchanged zones, and coordination records which each peer SAS has modified.

If a record returned by the responding SAS has changed state subsequent to the time range in which a requesting SAS asks for it, the responding SAS shall return the most current (that is, last known state) of the record or the state of the record as of the end_time request parameter. If there are no records qualifying to be returned, the SAS shall return a *MessageAggregation* object containing an empty *recordData* list of records [See Section 7.3]. The requesting SAS shall use the last version of the record it receives in response to the greatest timestamp range request as the current state of the record.

INFORMATIVE NOTE:

The exchange record behavior description is intended to support a high degree of variability in SAS implementation while producing eventually consistent synchronization results. That is, it should support concepts like log exchange, where change records are kept and exchanged based on their timestamps, as well as implementations that use smaller timestamp-based change records linked to the most up-to-date records stored in the SAS, or queries against the database based on change information stored there, or other varied mechanisms for providing the required data. The end result is that after fetching incremental updates from a point in the past up to the near-real-time present, a peer SAS will have a correctly updated representation of each relevant record. See Annex A for examples of such exchanges.

6.1.1 Qualification tests for inclusion in time-range responses

When responding to time-range requests, a SAS shall include every record that meets the following qualification tests:

- For CBSD records, every record corresponding to a CBSD with at least one active or pending grant (In this context, a pending grant is defined to be a requested grant for which radio resources have not yet been granted by the SAS, but for which the grant request parameters are required for purposes of margin allocation calculations. See [n.8] R2-SGN-16) whose parameters changed between the start and end time indicated in the request parameters.

- For zone records, every PAL Protection Area (PPA) which is qualified for protection, and every ad-hoc protection zone as indicated to SAS administrators according to the requirements in 47 CFR 96.15(a)(6) which is qualified for protection and for which the parameters change between the start and end time indicated in the request parameters.

- For coordination records, every coordination event as required for exchange by the coordination type and by the CBRS Operational Requirements [n.8] whose triggering time is between the start and end time indicated in the request parameters.

- A creation or update timestamp for qualifying records that is greater than or equal to the start parameter and less than or equal to the end parameter.

6.2 By-ID request support

If supported, the SAS shall return the most current details corresponding to the requested record when requested by a peer SAS for the following record types:

- SAS Administrator
- SAS Implementation
- CBSD
- ESC Sensors
- Zone (for zone types corresponding to PPA and ad-hoc protection zones – other types are optional for support)
- Coordination events

See Annex A for examples of such exchanges.

6.3 Push support

A SAS shall respond to data pushes for the following record types:

- CBSD
- Zone
- Coordination

The SAS shall respond to both by-ID and time-range push of these record types. See Annex A for examples of such exchanges.

A SAS is not required to provide pushes of these record types to other SASs, but if it does provide such pushes, it shall provide them to all peer SASs.

The SAS receiving a push request shall return a 200 HTTP status code in response to a push request if the request has triggered the appropriate response of the SAS which satisfies its implementation. The SAS shall return a non-200 HTTP status code appropriate to its error state if the push has failed to trigger an appropriate response (e.g. a 50x HTTP status code shall indicate a temporary (503) or permanent (500) failure of the receiving SAS to process the push request). In the case of incomplete or otherwise inaccurate data in such a request, the receiving SAS may return a 422 HTTP status code.

6.4 Full record dump

At least every seven days, the SAS shall prepare a full record dump including at least qualifying CBSD, Zone, and ESC sensor records and provide access to it to other SASs. Qualifying records are:

- CBSDs which have at least one grant active [or pending (for purposes of margin allocation)];
- zones (e.g. PPA zones, ad hoc exclusion zones) which require protection;
- ESC sensors which are affiliated with the SAS.

Other records (e.g. records of other types) may be included in such a dump.

The SAS shall make information about the dump available upon request to peer SASs. The dump data itself shall also be made available for at least 14 days.

> INFORMATIVE NOTE:
>
> This means that the SAS creates a full activity record dump every week. At least two such dumps must be available for retrieval at any given time, but the SAS is free to discard old activity dumps after that. The /dump endpoint references the most recent dump. This means that if a SAS is retrieving a dump and needs to restart it won't accidentally be retrieving data that is removed by the source SAS unless it takes over a week for such a retrieval.
>
> The requirement that the SAS make time-range data available for 30 days means that a full synchronization can be accomplished by a SAS by requesting the most recent full dump data from peers and then requesting hour-long segments of updates to that data until it has obtained the full data set.
>
> There are no requirements on any SAS in the protocol for any particular timing of requests or of providing data push. A SAS is free to use implementation-specific mechanisms to manage its own synchronization schedule such that it maintains the ability to satisfy its operational goals and the protection requirements of Part 96 [n.11].

7 Message Encoding and Transport

7.1 Message Encoding

SAS-SAS messages shall be encoded using JSON (JavaScript Object Notation) as defined in RFC-7159 [n.7]. Encoded message examples are shown in Annex A.

7.2 Message Transport

HTTPS (HTTP plus TLS) shall be used as the transport protocols for SAS-SAS message exchanges. The TLS protocol as specified in section 5.1 and HTTP version 1.1 as specified in [n.6] shall be used. An example HTTP request message header follows:

```
GET /v1.3/sas_admin/xyz HTTP/1.1
Host: www.sasadministratorapi.com
Date: Wed, 24 May 2017 02:23:17 GMT
Content-type: application/json
```

The SAS shall include its system time in the Date HTTP header field in all SAS-SAS messages (ref. [n.6]). Requesting SASs will ensure time-synchronization with the responding SAS to within 60 seconds.

The HTTP GET and POST methods shall be used for all SAS-SAS requests. The URL endpoints for SAS-SAS requests are described in Table 2. Requests are sent to the SAS URL path which includes the string "/{sas_version_number}/{sas_record_type}"[1] to indicate the SAS protocol version and the SAS record type describing the message. Each SAS administrator chooses the base URL of its SAS service, which shall include the sas_version_number. The SAS version number shall be in the form "v" + x + "." + y, where '+' is string concatenation with stripped off whitespace, and the operands x and y refer to the major and minor release numbers respectively. The sas_version_number of the SAS-SAS Protocol defined in this version of this technical specification is "v1.3".

A SAS method corresponds to a pair of request and response messages defined in Section 8. SAS methods and their corresponding URL constructions are listed below in Table 2: URL constructions and return types for SAS-SAS methods

Two exchange types, push and pull, are allowed for SAS to SAS information exchange.

- "Push": used by one SAS to push information to be shared to peer SASs.
- "Pull": used by one SAS to pull wanted information from peer SASs.

"Push" and "Pull" methods are directly mapped to the 'POST" and "GET" methods respectively as defined in the HTTP protocol.

[1] The curly braces indicate that the SAS should substitute the appropriate string value for the enclosed parameter.

Table 2: URL constructions and return types for SAS-SAS methods

Information Element Type	URL construction
SAS Administrators	Individual Records Pull: `GET $BASE_URL/sas_admin/$URLENCODED_ID` `sas_record_type: sas_admin` Return type: *SasAdministrator* object (See 8.1)
SAS Implementations	Individual Records Pull: `GET $BASE_URL/sas_impl/$URLENCODED_ID` `sas_record_type: sas_impl` Return type: *SasImplementation* object (See 8.2)
CBSDs	Individual Records Pull: `GET $BASE_URL/cbsd/$URLENCODED_ID` Push: `POST $BASE_URL/cbsd/$URLENCODED_ID` `sas_record_type: cbsd` Exchange data type: *CbsdData* object (See 8.3) Time-range records Pull: `GET $BASE_URL/cbsd:searchByTime?start_time=$T1&end_time=$T2` Push: `POST $BASE_URL/cbsd:searchByTime?start_time=$T1&end_time=$T2` Exchange data type: *MessageAggregation* object (See 7.3), *recordData* content type is *CbsdData* object (8.3)
ESC Sensors	Individual Records Pull: `GET $BASE_URL/esc_sensor/$URLENCODED_ID` `sas_record_type: esc_sensor` Exchange data type: *EscSensorData* object (See 8.4)

Information Element Type	URL construction
Zones	Individual Records Pull: `GET $BASE_URL/zone/$URLENCODED_ID` Push: `POST $BASE_URL/zone/$URLENCODED_ID` `sas_record_type: zone` Exchange data type: *ZoneData* object (See 8.5) Time-range records Pull: `GET $BASE_URL/zone:searchByTime?start_time=$T1&end_time=$T2` Push: `POST $BASE_URL/zone:searchByTime?start_time=$T1&end_time=$T2` Exchange data type: *MessageAggregation* object (See 7.3), *recordData* content type is *ZoneData* object (8.5)
Coordination events	Individual Records Pull: `GET $BASE_URL/coordination/$URLENCODED_ID` Push: `POST $BASE_URL/coordination/$URLENCODED_ID` `sas_record_type: coordination` Exchange data type: *CoordinationEvent* object (See 8.6) Time-range records Pull: `GET $BASE_URL/coordination:searchByTime?start_time=$T1&end_time=$T2` Push: `POST $BASE_URL/coordination:searchByTime?start_time=$T1&end_time=$T2` Exchange data type: *MessageAggregation* object (See 7.3), *recordData* content type is *CoordinationEvent* object (8.6)

Information Element Type	URL construction
Full activity dump	Pull: `GET $BASE_URL/dump` Push: `POST $BASE_URL/dump` `sas_record_type: dump` Return type: *FullActivityDump* object (See 8.7)

URL endpoints for data exchange shall be based on a base URL which is supplied by a SAS implementation, and which defines the resource to be exchanged (requested or supplied) following this table. (This is indicated in the *url* field of the *SasImplementation* object.)

The construction uses the following format convention:

`$BASE_URL/$RECORD_TYPE/$URLENCODED_ID` for individual record exchange. $RECORD_TYPE is the `sas_record_type` of the type of records to be exchanged. Note that the first path token of the `$ID` is also the `sas_record_type` of the record to be exchanged. The $ID is urlencoded ($URLENCODED_ID) such that UrlDecode($ENCODED_ID) = $ID. See Annex C for examples.

`$BASE_URL/$RECORD_TYPE:searchByTime?start_time=$START&end_time=$END` for time-range requests. The $START and $END parameters are url-escaped ISO 8601 time codes defining time limits for the records exchanged (See [n.9]). All times shall be in UTC.

If a malformatted $START or $END parameter is presented, or if $START is equal to or greater than $END, the SAS shall respond with the appropriate HTTP status code (400).

7.3 Message Contents Aggregation

Multiple required data elements may be placed into a single request for a push exchange, and similarly, in a pull exchange, the response message may contain aggregated data elements.

When using the individual record GET or POST methods described in Table 2, the SAS shall encode message payloads as a JSON object. When responding to the time-range record GET or POST methods, the SAS shall encode message payloads as a *MessageAggregation* object of the type in Table 3. This payload includes an array of JSON objects. The elements in such an array will be objects of the requested (or provided) message type.

In the case of error conditions in the SAS-SAS requests, the SAS shall use the appropriate HTTP status codes and an empty response. For example, an error in constructing the appropriate URL or a URL unsupported by the target SAS should be answered by a 404 status code. A syntactically correct request for which the SAS has no data shall produce the response of an empty JSON object (equivalent to "{}").

Table 3: *MessageAggregation* Object

Field	Data Type	Field Definition
startTime	string	• **Format**: string describing time and date. It is expressed using the format, YYYY-MM-DDThh:mm:ssZ, as defined by "Date and Time on the Internet: Timestamps" [n.9]. Indicates the beginning timestamp of the records included in the response (inclusive).
endTime	string	• **Format:** string describing time and date. It is expressed using the format, YYYY-MM-DDThh:mm:ssZ, as defined by "Date and Time on the Internet: Timestamps" [n.9]. Indicates the ending timestamp of the records included in the response (inclusive).
recordData	array of Object	The array of records in the response. Zero or more of a single type of object as defined in Section 8. Type depends on the sas_record_type as described in Table 2, or by SAS Administrator construction as used in the full record dump (8.7).

8 Parameters of SAS-SAS Messages

In this section, parameters of SAS-SAS messages are described in more detail. A parameter value can be one of the primitive JSON data types (string, number, boolean, array, or object). If a parameter is an object, a name for the object is given and a separate table describes parameters in the object.

All messages in the protocol are extensible using JSON extension mechanisms.

In every message and object, all fields are optional unless specifically described as required.

The JSON objects specified in the following subsections are conformant with RFC-4627 [i.5]. Note that this means that Unicode characters are used and have a default encoding of UTF-8.

8.1 SAS Administrator Message

Table 4: *SasAdministrator* object

Field	Data Type	Field Definition
id	string	• **Format**: sas_admin/$ADMINISTRATOR_ID • **$ADMINISTRATOR_ID**: SAS-CA certified unique SAS administrator identifier
name	string	Human-readable local significant string. The name of the SAS Administrator.
contactInformation	array of object: ContactInformation	Contains various contact information for the SAS Administrator.

8.1.1 ContactInformation object

Table 5: *ContactInformation* object

Field	Data Type	Field Definition
contactType	string	Human-readable string describing the type of contact information (e.g. “emergency”, “operations”)
name	string	Human-readable local significant string. The name of the contact point.
phoneNumber	array of string	Human-readable phone numbers at which this entity can be reached. Format: should follow the E.123 ITU-T recommendation.
email	array of string	Human-readable email contact information for this entity. Format: should follow the E.123 ITU-T recommendation.
address	array of string	Human-readable address for this entity
note	array of string	Any additional human-readable information for this entity. (e.g. hours of operation, preferred contact method, escalation procedures)

8.2 SAS Implementation Message

Table 6: *SasImplementation* object

Field	Data Type	Field Definition
id	string	• **Format**: sas_impl/$ADMINISTRATOR_ID/$SAS_IMPLEMENTATION • **$ADMINISTRATOR_ID**: SAS-CA certified unique SAS administrator identifier • **$SAS_IMPLEMENTATION**: SAS-CA certified unique SAS implementation instance identifier
name	string	Human-readable local significant string, the name of this SAS implementation.
administratorId	string	• **Reference**: ID of a *SasAdministrator* object
contactInformation	array of object: ContactInformation	Contains various contact information
publicKey	string	X.509 key (ref. [n.5])
fccInformation	object: FCCInformation	Contains the FCC certification information for this SAS implementation.
url	string	Publicly addressable URL for this SAS

8.2.1 FCCInformation object

Table 7: *FCCInformation* object

Field	Data Type	Field Definition
certificationId	string	FCC-issued certification ID, if any.
certificationDate	string	Date of certification, in format YYYY-MM-DD.
certificationExpiration	string	Date of certification expiration, in format YYYY-MM-DD
certificationConditions	string	Human-readable string or reference annotating the certification
frn	string	The FRN of the certified entity, if applicable.

Field	Data Type	Field Definition
sasPhase	string	Defines the Phase ("1" or "2") of certification.

8.3 CBSD Data Message

Table 8: *CbsdData* object

Field	Data Type	Field Definition
id	string	• **Format**: cbsd/$CBSD_REFERENCE_ID **$CBSD_REFERENCE_ID** is defined as $FCC_ID + "/" + sha1($SERIAL_NUMBER), the SHA-1 hash of the device manufacturer serial number that is unique within the FCC ID namespace scope. This creates a persistent, unique mapping from specific device parameters to the ID. $FCC_ID and $SERIAL_NUMBER are the unescaped *fccId* and *cbsdSerialNumber* strings registered by the CBSD in the *RegistrationRequest* JSON object [n.10] SHA-1 is to be applied to the string with no additional line termination characters. Reference implementation: the Python hashlib.sha1() implementation.
registration	object: RegistrationRequest (see [n.10])	Contains device installation parameters. All physical installation and device characterization parameters known to the source SAS shall be included as they were registered by the CBSD. Any group parameters related to interference protection shall be included as they were registered by the CBSD.
grants	array of object: GrantData	Contains [all] active [and pending (for purposes of margin allocation)] grants of the CBSD. All transmission-related parameters of [such] grants shall be included for all grants as they were returned to the CBSD.

Table 9: *GrantData* object

Field	Data Type	Field Definition
id	string	A grant identifier unique to this grant and CBSD allowing peer SASs to identify the grant.
terminated	boolean	Indicates whether the grant is currently terminated or not. “Terminated” in this context means a grant relinquished by a CBSD, terminated by the SAS, or suspended for a lengthy interval (longer than 7 days).
operationParam	object: OperationParam (see [n.10])	This data object includes operation parameters associated with the approved Grant. This field shall be included.
requestedOperationParam	object: OperationParam (see [n.10])	This data object includes requested operation parameters. This field shall be included.
channelType	string	“PAL”: the channel is a PAL channel. “GAA”: the frequency range is for GAA use.
grantExpireTime	string	Indicates the UTC time when the grant expires. It is expressed using the format, YYYY-MM-DDThh:mm:ssZ, as defined by [n.9].

8.3.1 Required and Optional Registration Data

The fields in Tables 8 and 9 shall be exchanged as registered for a CBSD (see the *RegistrationRequest* object in [n.10]; names and semantics of these fields are identical to those described in that specification).

The following parameters of the *RegistrationRequest* object included in the *CbsdData* object shall be exchanged as they are registered.

> *fccId*, *cbsdCategory*, *airInterface*, *installationParam* (see below), *measCapability*, *groupingParam* (see below)

These parameters (and any others) are optional:

userId, cbsdSerialNumber, cbsdInfo, callSign

The following parameters of the *InstallationParam* object included in the *CbsdData* object shall be exchanged. Any other parameters are optional.

latitude, longitude, height, heightType, indoorDeployment, antennaAzimuth, antennaGain, antennaBeamwidth

The following parameters of the *GroupParam* object shall be exchanged as they are registered when the *groupType* parameter of that object is equal to "INTERFERENCE_COORDINATION" or any other group type and accompanying group information identified as SAS-Essential data. Otherwise, the *GroupParam* objects are not required to be exchanged.

groupType, groupId, other accompanying data

Other fields from the *RegistrationRequest* object may be optionally included in this message as registered. Fields not required to be exchanged in this protocol, but required by syntactic constraints of the SAS-CBSD protocol [n.10] to be present or carry a particular format may be populated using an empty placeholder or a dummy value.

The following parameters of the *GrantData* objects included in the *CbsdData* object shall be exchanged as they are allocated for use by the CBSD (that is, a successful Grant response has been returned for that CBSD in response to a Grant procedure containing these *OperationParam* data elements), and the Grant has not subsequently been terminated.

grantExpireTime, operationParam (see below), *channelType*

In addition, *requestedOperationParam* shall also be exchanged to facilitate operations described in [n.8] R2-SGN-16. The following parameters of the *OperationParam* objects included in the *CbsdData* object within the *grants* parameter shall be exchanged as they are allocated.

maxEirp, operationFrequencyRange (including both *lowFrequency* and *highFrequency* data elements)

8.3.2 Signaling Grant Termination

When a Grant is terminated by a SAS, it shall exchange a CBSD registration record when requested for qualifying time range requests containing the record of the CBSD Registration Data with the terminated Grant marked by using a value of *true* for the *terminated* field.

8.4 ESC Sensor Message

Table 10: *EscSensorData* object

Field	Data Type	Field Definition
id	string	• **Format**: esc_sensor/$ADMINISTRATOR_ID/$SENSOR_ID • **$ADMINISTRATOR_ID**: the SAS administrator requesting ESC sensor protection • **$SENSOR_ID**: a unique identifier for the referenced ESC sensor created by the ESC administrator
installationParam	object: EscInstallationParam	Contains ESC Sensor installation parameters
protectionLevel	number	The protection level to be applied to this ESC sensor in units of dBm/MHz. If not present, the level should be interpreted as equal to that indicated in (n.8, R2-ESC-07)

Table 11: *EscInstallationParam* object

Field	Data Type	Field Definition
latitude	number	Latitude of the ESC antenna location in degrees relative to the WGS 84 datum [n.16]. The allowed range is from -90.000000 to +90.000000. Positive values represent latitudes north of the equator; negative values south of the equator. Values are specified using 6 digits to the right of the decimal point.
longitude	number	Longitude of the ESC antenna location in degrees relative to the WGS84 datum [n.16]. The allowed range is from -180.000000 to +180.000000. Positive values represent longitudes east of the prime meridian; negative values west of the prime meridian. Values are specified using 6 digits to the right of the decimal point.

Field	Data Type	Field Definition
height	number	This parameter contains the ESC antenna height in meters that may be expressed as an integer or as a numeric value including a decimal point. When the *heightType* parameter value is "AGL", the antenna height should be given relative to ground level. When the *heightType* parameter value is "AMSL", it is given with respect to WGS84 datum.
heightType	string	The value should be "AGL" or "AMSL". AGL height is measured relative to the ground level. AMSL height is measured relative to the mean sea level.
antennaAzimuth	number	This parameter contains the boresight direction of the horizontal plane of the ESC antenna in degrees with respect to true north. The value of this parameter is an integer with a value between 0 and 359 inclusive. A value of 0 degrees means true north; a value of 90 degrees means east.
antennaDowntilt	number	If present, this parameter contains the ESC antenna down tilt in degrees and is an integer with a value between -90 and +90 inclusive; a negative value means the antenna is tilted up (above horizontal).
azimuthRadiationPattern	array of object: RadiationPattern	This parameter specifies an ESC antenna radiation pattern or an effective ESC antenna radiation pattern in any direction in the azimuthal plane, specified at 1 degree increments referenced to the antenna boresight direction.
elevationRadiationPattern	array of object: RadiationPattern	If present, this parameter specifies an ESC antenna radiation pattern or an effective ESC antenna radiation pattern in any direction in the elevation plane (orthogonal to the azimuthal plane).

Table 12: *RadiationPattern* object

Field	Data Type	Field Definition
angle	number	This is the radiation angle. **In the azimuth plane:** the value is given in degrees relative to the antenna boresight direction. The value of this parameter is an integer, increasing in the clockwise direction as viewed from above, between 0 and 359 inclusive. **In the elevation plane**: the angle is given in degrees relative to the horizon. The value of this parameter is an integer between -180 and 180 inclusive. Radiation below the horizon has a positive elevation angle and radiation above the horizon has a negative elevation angle.
gain	number	The radiation gain in dBi. This parameter is an integer with a value between -127 and +128 (dBi). The gain provided is the gain in the direction of '*angle'*.

8.5 Zone Definition Message

Table 13: *ZoneData* object

Field	Data Type	Field Definition
id	string	• **Format**: zone/$CREATOR/$ZONE_ID • **$CREATOR**: SAS Administrator ID or static government zone definition source ID • **$ZONE_ID**: the identification of the referenced zone defined by the $CREATOR When usage is equal to "PPA" the format of the $CREATOR string is "ppa/$ADMINISTRATOR_ID" and the $ZONE_ID is equal to the PPA-ID string. When usage is equal to "CENSUS_TRACT" the format of the $CREATOR string is "census_tract/census/$YEAR" and the $ZONE_ID is equal to the FIPS code of the census tract. $YEAR is equal to the census year in which the census tract was defined. (Note: this zone type exchange is optional.) When usage is equal to "EXCLUSION_ZONE" the format of the $CREATOR string is "exclusion_zone/ntia/$DATE", and $DATE is a unique "YYYY_MM_DD" string describing the date on which NTIA issued the definition of the exclusion zone. $ZONE_ID is a unique reference to an exclusion zone.
name	string	Human-readable local significant string. The name of this zone.

creator	string	• **Format**: Human-readable string, one of the following:
		  • SAS Administrator record ID
		  • Static government zone definition source ID

Field	Data Type	Field Definition

usage	string	• **Format:** Enumeration describing the usage of the zone. One of the values:
		  • "CENSUS_TRACT"
		  • "PPA"
		  • "EXCLUSION_ZONE"

terminated	boolean	Indicates whether the zone (e.g. PPA) is currently terminated or not.
ppaInfo	object: *PPAInformation*	For zones where the *usage* parameter is equal to "PPA" this field shall be included.
zone	object: GeoJSON ([n.13]) and Annex B	Self-contained geometry description of the addressed zone.

8.5.1 PPA Information

When SAS Administrators exchange the Zone Data record for a PAL Protection Area (PPA), the *ppaInfo* field is included in the *ZoneData* object.

Table 14: *PPAInformation* object

Field	Data Type	Field Definition
palId	array of string	List of one or more PAL Database record IDs (ref: [n.15]) upon which the PPA is based. All PALs indicated in the list shall have a single PAL Holder.
cbsdReferenceId	array of string	List of one or more CBSD Reference IDs in the cluster list of the PPA.
ppaBeginDate	string	Date of the start of the PPA protection period. It is expressed in UTC using the format, YYYY-MM-DDThh:mm:ssZ, as defined by [N.6].
ppaExpirationDate	string	This field represents the PPA expiration date. It is expressed in UTC using the format, YYYY-MM-DDThh:mm:ssZ, as defined by [N.6].

ppaRegionType	string	This field describes the region type of the PPA to be used in calculating the path loss for PPA protection. The field shall be set to one of the following values: • "URBAN" • "SUBURBAN" • "RURAL"

8.6 Coordination Event Message

Table 15: *CoordinationEvent* object

Field	Data Type	Field Definition
id	string	• **Format**: coordination/$ADMINISTRATOR_ID/$EVENT_ID • **$ADMINISTRATOR_ID**: SAS Administrator ID • **$EVENT_ID**: event record ID created by the originating SAS Administrator
name	string	Human-readable local unique reference to the event
creator	string	Human-readable string identifying the creator of the coordination event.
creationDate	string	• **Format**: string describing time and date. It is expressed using the format, YYYY-MM-DDThh:mm:ssZ, as defined by "Date and Time on the Internet: Timestamps" [n.9].
expirationDate	string	• **Format**: string describing time and date. It is expressed using the format, YYYY-MM-DDThh:mm:ssZ, as defined by "Date and Time on the Internet: Timestamps" [n.9].
description	string	Human-readable description of the coordination event.
coordinationType	string	• **Format**: Enumerated value indicating the type of event. One of the values: "INTERFERENCE_REPORT", "AD_HOC_EXCLUSION_ZONE", "ENFORCEMENT_ACTION", "ESC_SENSOR_DEPLOYMENT"

Field	Data Type	Field Definition
coordinationDevice	array of string	• **Reference**: ID(s) of the involved device (e.g. a *CbsdData* ID or an *EscSensorData* ID). May be empty. May also refer to a specific incumbent device or area.
coordinationZone	array of string	• **Reference**: Array of IDs of the involved *ZoneData* objects. May be empty.
coordinationData	object: type is dependent upon the CoordinationType field	Structured object describing the coordination data • Event specific • Extensible anchor for any other metadata needed for automated handling of particular coordination events.

8.7 Full Activity Dump Message

Table 16: *FullActivityDump* object

Field	Data Type	Field Description
files	array of ActivityDumpFile	Array of one or more objects corresponding to files comprising the full activity dump.
generationDateTime	string	The date and time at which the activity dump was generated. The dump is guaranteed to include the effects of all activities pertinent to the current state of records qualified for exchange by the criteria of (section 6.4) up to and including the *generationDateTime*. It is expressed using the format, YYYY-MM-DDThh:mm:ssZ, as defined by "Date and Time on the Internet: Timestamps" [n.9].
description	string	Any additional human-readable description the source SAS may wish to attach.

8.7.1 ActivityDumpFile object

Table 17: *ActivityDumpFile* object

Field	Data Type	Field Description
url	string	The retrieval URLs at which the peer can retrieve the activity dump file. Retrieval of the resources at these URLs shall support byte-range requests using the standard HTTP Content-Range mechanisms. Retrieval of these URLs shall occur only within the security context of section 5.1.
checksum	string	The SHA-1 checksum of the contents of the activity dump file referred to by *url*.
size	number	The size of the activity dump file in bytes.
version	string	The version of the SAS-SAS protocol used for generating this file. Format should follow that of section 7.2. Example: “v1.0”
recordType	string	The type of records contained in the activity dump file. Corresponds to one of the ID sas_record_type prefix values as defined in Table 2. Examples: “zone”, “cbsd”, “coordination”

The format of the resources retrieved from the indicated URLs are JSON objects corresponding to the schema in Table 2 used for time-range responses, and containing *recordData* objects corresponding to the messages defined in this section. Records in a single file shall be all of one record type. The combination of the records in all the indicated URLs shall contain the full activity dump for the source SAS up to and including the timestamp indicated in *generationDateTime*.

9 History

Document history		
V1.0.0	29 November 2016	Version 1.0.0 released by Forum Chair
V1.1.0	1 August 2017	Approved technical revision released by the Forum Chair
V1.2.0	20 October 2017	Approved technical revision released by the Forum Chair
V1.3.0	24 April 2018	Approved technical revision released by the Forum Chair
V1.3.1	29 January 2019	Approved editorial revision released by SSC Steering Group

Annex A: Message Exchange Examples

This Annex includes examples of CBSD message exchanges. Other data types are exchanged similarly.

1 Pull request of specific CBSD record

The requesting SAS constructs the URL using the BASE_URL of the peer SAS to which the request will be issued, given the known ID of the CBSD record:

```
$BASE_URL/cbsd/cbsd%2Fabc123%2Ff00268bfa5c402163dfd7d2d82ff537018e55c6b
```

Within the appropriate security and prerequisite contexts, it performs a GET HTTP request for this URL. In response, the peer SAS returns a 200 HTTP status code upon success, with an HTTP payload consisting of a JSON object following the schema described in Section 8.3 for the CBSD data message:

```
{
  "id": "cbsd/abc123/f00268bfa5c402163dfd7d2d82ff537018e55c6b"
  "registration": {
      "fccId": "abc123",
      "cbsdCategory": "A",
      "callSign": "CB987",
      "userId": "John Doe",
      "airInterface": {
        "radioTechnology": "E-UTRA"
      },
      "measCapability": [
        "RECEIVED_POWER_WITHOUT_GRANT"
      ],
      "installationParam": {
        "latitude": 37.419735,
        "longitude": -122.072205,
        "height": 6,
        "heightType": "AGL",
        "indoorDeployment": true
      },
      "groupingParam": [
        { "groupId": "exampleGroup",
          "groupType": "INTERFERENCE_COORDINATION" }
      ]
  },
  "grants": [ {
    "id": "SAMPLE_ID_12345",
    "operationParam": {
      "maxEirp": 30,
```

```
      "operationFrequencyRange": {
        "lowFrequency": 3550000000,
        "highFrequency": 3570000000
      }
    },
    "requestedOperationParam": {
      "maxEirp": 30,
      "operationFrequencyRange": {
        "lowFrequency": 3550000000,
        "highFrequency": 3570000000
      }
    },
    "channelType": "GAA",
    "grantExpireTime": "2017-09-08T04:30:00Z",
    "terminated": false
  } ]
}
```

Note that for a push of an update to this record, the SAS would construct the same URL, then issue a POST request to the peer SAS, passing this same record as the HTTP payload, to which the peer SAS would be expected to return a 200 HTTP success code as acknowledgement, with no HTTP body necessary in the response.

2 Time-range request for CBSD records

Ordinarily, a requesting SAS will not know the ID of CBSDs whose records may have changed, and so requests updates from peers concerning the most recent status of any records which may have changed within a certain time period. In ordinary operation, the SAS will send such requests periodically to peers, maintaining a high-water-mark of last-synced state. It will construct a new request URL using the BASE_URL of the peer SAS to which the request will be issued, given the time range extents of the period of interest:

```
$BASE_URL/cbsd:searchByTime?start_time=2017-04-
01T11%3A12%3A13Z&end_time=2017-04-01T11%3A12%3A23Z
```

Within the appropriate security and prerequisite contexts, it performs a GET HTTP request for this URL. In response, the peer SAS returns a 200 HTTP status code upon success, with an HTTP payload consisting of a JSON object following the schema described for the aggregation response defined in Section 7.3 and containing records corresponding to objects following the schema of the CBSD data message. Note that the time parameters here are URL-encoded (and so escape the ':' character), and represent an interval of 10 seconds. The peer SAS interprets these times in its own reference time frame, and returns all records which changed state within this interval. If the CBSD used as an example in Section 1 changed state by relinquishing its previous grant and getting a new one, the resulting HTTP message might look like this:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
Date: Mon, 03 Oct 2016 11:07:33 GMT

{
  "startTime": "2017-04-01T11:12:11Z",
  "endTime": "2017-04-01T11:12:18Z",
  "recordData": [
{
  "id": "cbsd/abc123/f00268bfa5c402163dfd7d2d82ff537018e55c6b"
  "registration": {
      "fccId": "abc123",
      "cbsdCategory": "A",
      "callSign": "CB987",
      "userId": "John Doe",
      "airInterface": {
        "radioTechnology": "E-UTRA"
      },
      "measCapability": [
        "EutraCarrierRssiNoTx"
      ],
      "installationParam": {
        "latitude": 37.419735,
        "longitude": -122.072205,
        "height": 6,
        "heightType": "AGL",
        "indoorDeployment": true
      },
      "groupingParam": [
        { "groupId": "exampleGroup",
          "groupType": "INTERFERENCE_COORDINATION" }
      ]

  },
  "grants": [ {
    "id": "SAMPLE_ID_12345",
    "operationParam": {
      "maxEirp": 30,
      "operationFrequencyRange": {
        "lowFrequency": 3580000000,
        "highFrequency": 3600000000
      }
    },
    "requestedOperationParam": {
      "maxEirp": 30,
      "operationFrequencyRange": {
        "lowFrequency": 3580000000,
        "highFrequency": 3600000000
      }
    },
```

```
    "channelType": "GAA",
    "grantExpireTime": "2017-11-08T04:30:00Z",
    "terminated": false
  } ]
}
]
}
```

Note that because this request spanned both state changes, only the latest entry for this CBSD in the result set is returned. If the state changes had happened in different time ranges, the peer SAS would have the option to return either the most current status of the CBSD record in question in response to both requests, or the last status of the CBSD record in question within the queried time range.

Note also that the timestamps returned by the peer SAS record its own high-water mark for changes returned as a result of the query. By reflecting a different high water mark than the query, it lets the requesting SAS know any data after the *endTime* in the response message is not included, so the requesting SAS should use the returned *endTime* for its new high-water-mark, not the value from the end_time parameter in its request. This allows the peer SAS to ensure that the requesting SAS can keep its time basis aligned with the ability of the peer SAS to provide complete data in its own time reference.

Normative Annex B: GeoJSON format used by SAS

This Annex specifies a subset of the GeoJSON format [n.13] which shall be used by SASs for geospatial data interchange.

1. GeoJSON format specifications for SAS

The following defines a subset of the GeoJSON format which shall be used by SAS for geospatial data interchange. Each GeoJSON string shall have the following properties:

1. A single "FeatureCollection" root node, comprising an array "features".
2. Each item in the "features" array is an object of "type" "Feature" and with a "geometry" field.
3. The "geometry" field is an object of "type" "polygon", which represents a single polygon, with or without holes (see [n.13], section 3.1.6). Polygons pertaining to different "Feature" should have empty intersection.
4. The "geometry" object contains a "coordinates" array, where each item is an array of point coordinates. The first array of coordinates represents the outer boundary of the polygon and should have a counterclockwise orientation, all the other arrays of coordinates, if present, represent the boundaries of the holes and should have a clockwise orientation. The point coordinates will be in **longitude-latitude** order (see [n.13], appendix A.1). The last point in each "coordinates" array must be equal to the first (by GeoJSON specification of a LineString, see [n.13] section 3.1.4).

2. Example

The example below comprises a GeoJSON string in the format used by SAS, representing two rectangular polygons: the first polygon with a rectangular hole, the second polygon without any holes.

```
{
  "type": "FeatureCollection"
  "features": [{
    "type": "Feature"
    "geometry": {
      "type": "Polygon"
      "coordinates": [
        [ [10.0, 3.0], [15.0, 3.0], [15.0, 6.0], [10.0, 6.0], [10.0, 3.0] ],
        [ [10.1, 3.1], [10.1, 5.9], [14.9, 5.9], [14.9, 3.1], [10.1, 3.1] ]
      ]
    }, {
    "type": "Feature"
    "geometry": {
      "type": "Polygon"
      "coordinates": [
        [ [2.0, 1.5], [7.0, 1.5], [7.0, 4.5], [2.0, 4.5], [2.0, 1.5] ]
      ]
    }
```

```
    } ]
}
```

Annex C: URL and ID Examples

This Annex includes examples of URLs and IDs.

1. CBSD record

For a record with FCC ID "example_fcc_id" and serial number "example_serial_number", the ID used in the CbsdData object is

cbsd/$CBSD_REFERENCE_ID

= cbsd/example_fcc_id/a61ca59761d21c89d2c952dfccc0ee1495a822d7

A pull request for this record would use the following URL:

$BASE_URL/cbsd/UrlEscape(cbsd/example_fcc_id/a61ca59761d21c89d2c952dfccc0ee1495a822d7)

which is equivalent to

$BASE_URL/cbsd/cbsd%2Fexample_fcc_id%2Fa61ca59761d21c89d2c952dfccc0ee1495a822d7

The full URL in this case would look like:

http://www.sasadministratorapi.com/v1.3/cbsd/cbsd%2Fexample_fcc_id%2F
a61ca59761d21c89d2c952dfccc0ee1495a822d7

2. SAS administrator record

For a record from the SAS administrator "example_sas_admin", the ID used in the SasAdministrator object is

sas_admin/$ADMINISTRATOR_ID = sas_admin/example_sas_admin

A pull request for this record would use the following URL:

$BASE_URL/sas_admin/UrlEscape(sas_admin/example_sas_admin)

= $BASE_URL/sas_admin/sas_admin%2Fexample_sas_admin

3. SAS implementation record

For a record from the SAS administrator "example_sas_admin" with unique implementation ID "example_impl_id", the ID used in the SasImplementation object is

sas_impl/$ADMINISTRATOR_ID/$IMPLEMENTATION_ID

= sas_impl/example_sas_admin/example_impl_id

A pull request for this record would use the following URL:

$BASE_URL/sas_impl/UrlEscape(sas_impl/example_sas_admin/example_impl_id)

= $BASE_URL/sas_impl/sas_impl%2Fexample_sas_admin%2Fexample_impl_id