Table of Contents

0SAS ESC API protocol

This document specifies the Application Programming Interface (API) for Interface between Spectrum Access System (SAS) and Environmental Sensing Capability (ESC).

Prerequisite operations

The following operations shall be done before SAS-ESC communications.

1. Exchange of Public Key Infrastructure (PKI) between SAS Administrator and ESC Operator
2. Exchange of Private Key for digital signature
3. Setting of communication timeout

Key Bridge expects the FCC to authorize more than one party to operate a ESC, and that the internal architecture and configuration of each ESC may differ. Key Bridge further expects that each ESC, or at least the ESC in this proposal, will be independent and autonomous to the one or more SASs that depend upon their respective ESC for incumbent detection. The same principal applies in reverse as well: a SASs internal architecture and configuration is also independent and autonomous to its ESC, and each ESC may pursue different strategies and implement different technologies for incumbent detection.

Regardless of their respective internal architecture, technology and detection strategies, a ESC and SAS must interface to share information about spectrum availability or un-availability. This essential concept is called peering and is shown in Illustration. Through peering a SAS and ESC may may exchange data in a neutral manner irrespective of their internal architecture and configuration.

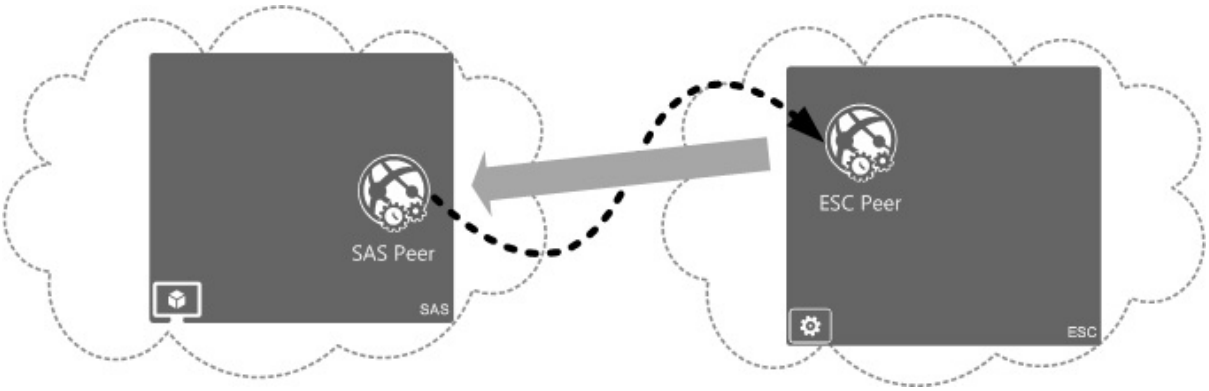

Illustration: SAS to ESC Peering masks internal architecture.

A SAS exterior peering protocol standard will place no constraint on the internal architecture and operation of the various entities that use the protocol. Referring again to Illustration 10, a generic peering process enables SAS of unknown constitution on the left peers with a Key Bridge ESC on the right.

SAS to ESC peering operational security policies and configurations are under development within a multi-stakeholder group in which Key Bridge is a participant.1 The actual protocol and data structures for SAS to ESC peering is not the subject of multi-stakeholder group consideration. Key Bridge has therefore invented proprietary and also proposed non-proprietary methods that a SAS may use when coordinating with a ESC to dynamically protect a non-informing incumbent spectrum user. The Key Bridge SAS Gateway Protocol is, at present, a proprietary application peering protocol that includes mechanisms to support cryptographically secure message exchange for both SAS to SAS and SAS to ESC peering, including all security prescriptions identified by the multi-stakeholder group.

A SAS and ESC Peering relationship using to the SAS Gateway Protocol is secure and neutral. A SAS Infrastructure may register with and receive information from any inter-operable ESC Infrastructure supporting a peering service. When using the Key Bridge SAS Gateway Protocol a compatible SAS may register with and receive information from a compatible ESC via a designated ESC Peering API.

In the Key Bridge architecture ESC Service Nodes are responsible for NIIU detection services and peer directly with a subscribing SAS via the SAS Gateway Protocol. This strategy is designed to limit the geographic extent of NIIU information conveyed to a SAS to that necessary for the SAS to administer CBSDs in a specific geographic region. However, the ESC places no constraint on SAS internal architecture or operation, and a SAS may peer with multiple ESC Service Nodes.

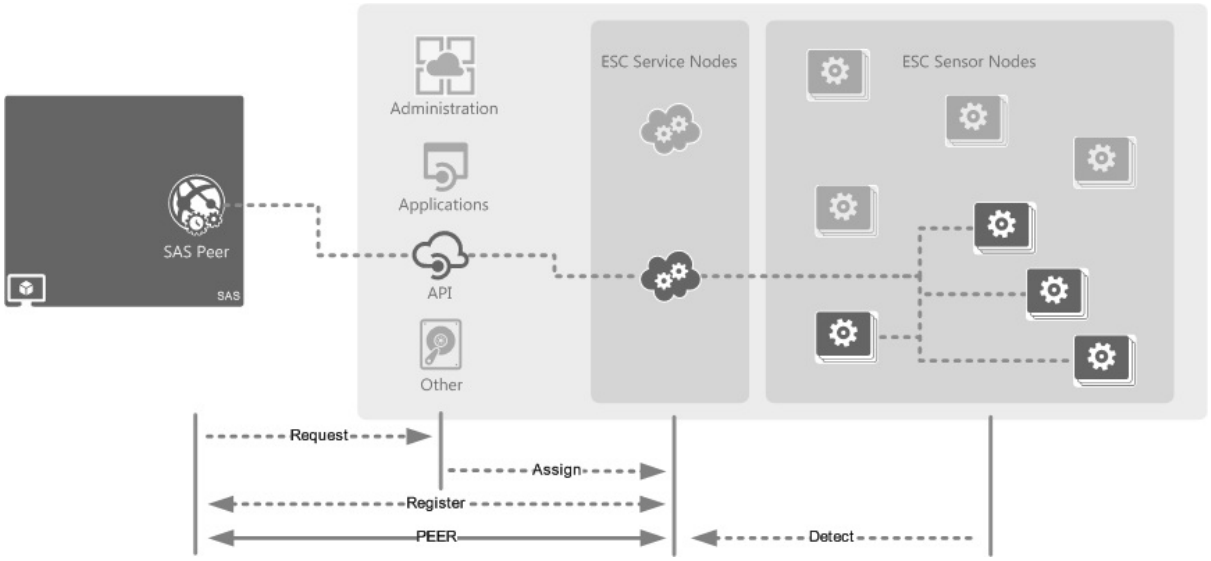

Illustration 11: ESC Service Nodes provide NIIU detection services to a SAS.

The process of a generic SAS peering with the Key Bridge ESC Infrastructure is detailed in Illustration 11, and proceeds according to the following general strategy.

- **Request**. A SAS first inquires about service availability to the ESC Infrastructure via a general peering API interface. The SAS service request includes the geographic area or areas where the SAS wishes to provide spectrum administration. If ESC services are not available in the requested geographic area the request is rejected (i.e. fails), otherwise the request is assigned to a matching ESC Service Node.
- **Assignment**. If NIIU detection capability is available in the SAS requested geographic area the ESC Infrastructure will assign the SAS registration request to a matching ESC Service Node.
- **Register**. After a geographic match has been established the respective ESC Service Node and the SAS exchange detailed information necessary to establish a direct and persistent peering relationship. The registration process includes conveying to the ESC Peer the specific SAS geographic regions of responsibility plus instructions describing how the ESC may query the SAS for additional information and also how to notify the SAS of ESC spectrum availability information, updates and instructions.
- **Peer**. Once established the ESC Service Node will respond to spectrum availability requests within its geographic area of service and also inform the SAS about any changes in spectrum availability.

An artifact of this CONOPS is that the Key Bridge ESC Service Nodes are assigned responsibility to determine the quantity and quality of protection for the incumbent user within their ESC Service Area, while responsibility to effect that protection is retained within a SAS. That is to say: AESC Service Nodeis the service providing component of the ESC Infrastructure, and it is the ESC Service Node that authorizes or de-authorizes CBSD operations within their responsible service area according to a *geographic partitioning strategy* within their respective ESC Service Area.

1Message exchange flow and message types

Message exchanges between SAS and ESC are shown in Figure 1 below.

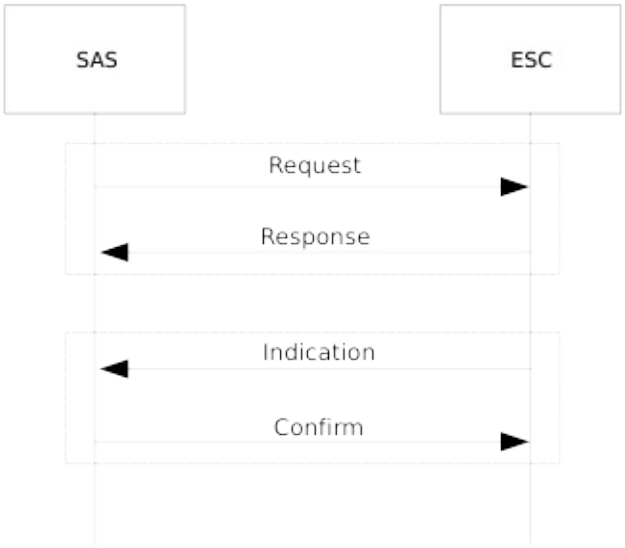

Figure 1: Message flow between SAS and ESC

As per this message flow, the following messages shall be exchanged between SAS and ESC.

- **SAS Registration Message**: The message for the SAS to register with the ESC. Request-Response flow in the figure 1 is used.
- **ESC Information Update Message**: The message for the ESC to indicate the update of the ESC information to the SAS. Indication-Confirm flow in the figure 1 is used.
- **DPA State Message**: The message for the ESC to indicate the DPA activation status to the SAS. Indication-Confirm flow in the figure 1 is used.
- **Keep Alive Message**: The message for the SAS to detect a failure with ESC. Request-Response
    - flow in the figure 1 is used.
    - **SAS Deregistration Message**: The message for the SAS to deregister from the ESC. Request-Response flow in the figure 1 is used.

2Message encoding and transport

2.1Message encoding

2.1.1JSON encode

The contents of SAS-ESC messages shall be generated by encoding the MessageContainer object specified in the section 7 of this document using JSON (JavaScript Object Notation) as defined in RFC 7159 [n.2]. Unicode characters shall be used and its default encoding shall be UTF-8.

2.2Message transport

HTTPS shall be used as the transport protocols for SAS-ESC message exchanges. The TLS protocol as specified in [n.3] and HTTP version 1.1 as specified in [n.4] shall be used.

The HTTP POST method shall be used for all requests from the ESC to the SAS and from the SAS to the ESC.

POST shall be sent to the URL provided by the SAS or the ESC. The URL shall be configured in accordance with the following format.

```
$BASE_URL/$RELEASE_NUMBER/$METHOD_NAME
```

$BASE_URL represents the base URL of the SAS or the ESC.

$RELEASE_NUMBER represents the CBRS release number corresponding to the ESC Requirements developed by WinnForum [n.1]. In this specification, $RELEASE_NUMBER shall be “v1.3”.

$METHOD_NAME represents method name of the API specified in this document and corresponding to the specific SAS-ESC message. The methods in the following table shall be used in the API.

Table 1: List of methods in API

SAS-ESC message name	$METHOD_NAME
SAS Registration Message	sasRegistration
ESC Information Update Message	escUpdate
DPA Activation Status Message	dpaStatusMessage
Keep Alive Message	keepAlive
SAS Deregistration Message	sasDeregistration

The error shall be indicated by using HTTP status code.

If there is no error in the message, HTTP status code 200 (SUCCESS) shall be returned.

If invalid or malformatted parameters are used in the message, HTTP status code 400 (BAD REQUEST) shall be returned.

If invalid or malformatted URL is used, HTTP status code 404 (NOT FOUND) shall be returned.

4SAS-ESC Messages

4.1Message Container

All the SAS-ESC Messages shall be generated by encoding the *MessageContainer* object using JSON. The table below defines the the *MessageContainer* object.

Table 2: MessageContainer object

Field	R/O/C	Descriptions
NAME: protectedHeader DATA TYPE: string	Required	The value of this parameter is the BASE64-encoded JOSE protected header. encoded as a JSON object equivalent to the JWT HS256 method described in Section 3 of RFC 7515 [n.7]. BASE64 encoding is per RFC 4648 [n.8]. Valid value is equivalent to the below JSON: `{ “typ”: “JWT”, “alg”: “HS256” }` Editor’s Note: The value specified here is just copied and pasted from the SAS-CBSD Protocol. If there is better value, it will be welcomed to reflect here.
NAME: encodedPayloadData DATA TYPE: string	Required	The value of this parameter is the encoded Payload Data to be signed by the Private Key.This parameter is calculated by taking the BASE64 encoding of a JSON describing Payload Data (section 7.1) according to the procedures in Section 3 of RFC 7515 [n.7]. BASE64 encoding is per RFC 4648 [n.8].
NAME: digitalSignature DATA TYPE: string	Required	This parameter contains the digital signature applied to the encodedPayloadData field. This parameter is calculated by taking the BASE64 encoding of the digital signature applied to the Payload Data, prepared according to the procedures in Section 3 of RFC 7515 [n.7], using the HMAC SHA-256 algorithm as declared in the protectedHeader field. BASE64 encoding is per RFC 4648 [n.8].

5.17.2 Payload Data

5.0.17.2.1 Payload Data for SAS Registration Message

5.0.1.17.2.1.1 SAS Registration Request

SAS Registration Request shall be sent by the SAS to register with the ESC. The SAS Registration Request may be sent also when the SAS information in the *SasRegistrationRequest* object in the following table is updated. The SAS Registration Request shall be generated by encoding the *MessageContainer* object in which JSON-encoded *SasRegistrationRequest* object is used to generate *encodedPayloadData* field.using JSON.

Table 3: SasRegistrationRequest object

Field	R/O/C	Descriptions
NAME: sasAdministratorId DATA TYPE: string	Required	This field shall be included to indicate which SAS Administrator manages the SAS. The format of this field shall be same as $ADMINISTRATOR_ID used in the SAS-SAS Protocol [n.5].
NAME: sasImplementationId DATA TYPE: string	Required	This field shall be included to indicate the identification of the SAS. The format of this field shall be same as $SAS_IMPLEMENTATION used in the SAS-SAS Protocol [n.5].
NAME: publicKey DATA TYPE: string	Required	This field shall be included to indicate the public key of SAS.
NAME: baseUrl DATA TYPE: string	Required	This field shall be included to indicate the base URL ($BASE_URL) of the SAS identified by the *sasImplementationId* field in this object.

5.0.1.27.1.1.2 SAS Registration Response

SAS Registration Response shall be sent by the ESC to the SAS for the response to the SAS Registration Request. The SAS Registration Response shall be generated by encoding the *MessageContainer* object in which JSON-encoded *SasRegistrationResponse* object is used to generate *encodedPayloadData* field. in the following table using JSON.

Table 4: SasRegistrationResponse object

Field	R/O/C	Descriptions
NAME: sasRegistrationId DATA TYPE: string	Required	This field shall be generated by the ESC and included to indicate the registration identifier for the SAS.
NAME: escOperatorId DATA TYPE: string	Required	This field shall be included to indicate the identification of ESC Operator managing ESC. The format of this field shall be FFS.
NAME: escInformation DATA TYPE: objectEscInformation	Required	This field shall be included to indicate the ESC information.

Table 5: EscInformation object

Field	R/O/C	Descriptions
NAME: escImplementationId DATA TYPE: string	Required	This field shall be included to indicate the identification of the ESC implementation. The format of this field shall be FFS.
NAME: escSensors		This field shall be included to indicate the information of ESC sensors

NAME: escSensors DATA TYPE: array of object:EscSensorData[5]	Required	deployed by the ESC Operator and managed by the ESC identified by the *escOperatorId* and *escImplementationId* fields in this object, respectively. See details in 7.1.2.1.

Editor's Note: Should the format of each ID be "ESC-CA certified unique ESC Operator identifier" similar to both SAS Administrator and Implementation ID?

Editor's Note: "escImplementationId" is included here for the similar purpose of "SAS Implementation" in SAS-SAS. In other words, one or more ESC instances might be operated. Decision to remove or keep this field strongly depends on Key Bridge for now.

5.0.1.2.17.1.1.2.1 Enhancements to EscSensorData object for SAS-ESC Interface

In this specification, theEscSensorDataobject specified in the SAS-SAS Protocol [n.5] shall be reused with enhancements. Enhanced definition of theEscSensorDataobject is described in the following table.

Table 6: Enhanced definition of EscSensorData object

Field	R/O/C	Descriptions
NAME: id DATA TYPE: string	N/A	This field shall not be included.
NAME: sensorId DATA TYPE: string	Required	This field shall be included to indicate a unique identifier of the ESC Sensor.
NAME: installationParam DATA TYPE: objectInstallationParam	N/A	This field shall not be included.
NAME: escInstallationParam DATA TYPE: objectEscInstallationParam	Required	This field shall be included to indicate the installation parameters of the ESC Sensor identified by thesensorIdfield in this object.
NAME: protectionLevel DATA TYPE: number	Required	This field shall be included to indicate the protection level to be applied to the ESC Sensor identified by thesensorIdfield in this object. The value of this field shall be in units of dBm/MHz with decimal point.

Table 7: EscInstallationParam object

Field	R/O/C	Descriptions
NAME: latitude DATA TYPE:number	Required	Latitude of the ESC Sensor location in degrees relative to the WGS 84 datum. 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.
NAME: longitude DATA TYPE:number	Required	Longitude of the ESC Sensor location in degrees relative to the WGS84 datum. 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.
NAME: height DATA TYPE:number	Required	The antenna height of ESC Sensor in meters. When theheightTypeparameter value is "AGL", the antenna height shall be given relative to ground level. When theheightTypeparameter value is "AMSL", it is given with respect to WGS84 datum.
NAME: heightType DATA TYPE: string	Required	The value shall be "AGL" or "AMSL".AGL height is measured relative to the ground level.AMSL height is measured relative to the mean sea level.
NAME: horizontalAccuracy DATA TYPE:number	Required	A positive number in meters to indicate accuracy oftheESC Sensorantennahorizontal location.

NAME: verticalAccuracy DATA TYPE:number	Required	A positive number in meters to indicate accuracy oftheESC Sensorvertical location.
NAME: antennaAzimuth DATA TYPE:number	Required	Boresight direction of the horizontal plane of the 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.
NAME: antennaDowntilt DATA TYPE:number	Required	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).
NAME: azimuthAntennaPattern DATA TYPE:array of object:AntennaPattern	Required	This parameter specifies anantenna pattern in any direction for the ESC Sensor antenna in the azimuthal plane.

Table 8: AntennaPattern object

Parameter	R/O/C	Description
NAME:angle DATA TYPE:number	Required	This is the angle.**In the azimuth plane:**the value is given in degrees relative to the boresight of the antenna. The value of this parameter is an integer between 0 and 360 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.
NAME:gain DATA TYPE:number	Required	The gain in dBi includes both antenna gain and beamforming gain. This parameter is an integer with a value between -127 and +128 (dBi). The gain provided is the gain in the direction of ‘angle’.

6.0.17.2.2 Payload Data for ESC Information Update Message

6.0.0.17.2.2.1 ESC Information Update Indication

ESC Information Update Indication shall be sent by the ESC to the SAS when the ESC information is updated. The ESC Information Update Indication shall be generated by encoding theMessageContainerobject in which JSON-encodedEscInformationUpdateSasRegistrationResponseobject is used to generateencodedPayloadDatafield.in the following table using JSON.

Table 9: EscInformationUpdate object

Field	R/O/C	Descriptions
NAME:escOperatorId DATA TYPE: string	Required	This field shall be included to indicate the identification of ESC Operator managing ESC. The format of this field shall be FFS.
NAME:escInformation DATA TYPE: objectEscInformation	Required	This field shall be included to indicate the ESC information.

6.0.0.27.2.2.2 ESC Information Update Confirm

ESC Information Update Confirm shall be sent by the ESC to the SAS for the response to the ESC Information Update Indication. The ESC Information Update Confirm shall be generated by encoding the *MessageContainer* object in which empty JSON object (i.e. “{}”) is used to generate *encodedPayloadData* field.

7.0.17.2.3 Payload Data for DPA Activation Status Message

7.0.17.2.3.1 DPA Activation Status Indication

DPA Activation Status Indication shall be sent by the ESC to the SAS. The DPA Activation Status Indication shall be generated by encoding the *MessageContainer* object in which the *DpaActivateStatusIndication* object is used to generate *encodedPayloadData* field in the following table using JSON.

Table 10: DpaActivateStatusIndication object

Field	R/O/C	Descriptions
NAME:dpaId DATA TYPE: string	Required	This field shall be included to indicate which a unique identifier of the DPA.
NAME:dpaActivationStatus DATA TYPE: object DpaActivationStatus	Required	This field shall be included to indicate the DPA activation status.

Table 11: DpaActivationStatus object

Field	R/O/C	Descriptions
NAME:dpaActivated DATA TYPE: boolean	Required	This field shall be included to indicate the DPA activation status of the frequency range indicated by the *frequencyRange* field in this object “true”: DPA is (has been) activated “false”: DPA is (has been) deactivated
NAME:frequencyRange DATA TYPE: object FrequencyRange[6]	Required	This field shall be included to indicate the frequency range.

Editor’s Note: Atomic Transaction = 1 DPA + 1 Channel

Editor’s Note: Open Issues.

1: ID Assignment,

2: Partial coverages (geometry and channel block).

7.0.0.17.2.3.2 DPA Activation Status Confirm

DPA Activation Status Confirm shall be sent by the ESC to the SAS for the response to the DPA Activation Status Indication. The DPA Activation Status Confirm shall be generated by encoding the *MessageContainer* object in which empty JSON object (i.e. “{}”) is used to generate *encodedPayloadData* field.

8.0.17.2.4 Payload Data for Keep Alive Message

8.0.0.17.2.4.1 Keep Alive Request

Keep Alive Indication shall be sent from the SAS to ESC. The Keep Alive Request shall be generated by encoding the MessageContainer object in which the *KeepAlive* object is used to generate encodedPayloadData field.in the following table using JSON.

Table 12: KeepAlive object

Field	R/O/C	Descriptions
NAME:sasRegistrationId DATA TYPE: string	Required	This field shall be included to indicate the registration identifier of the SAS.

8.0.0.27.2.4.2 Keep Alive Response

Keep Alive Confirm shall be sent from the ESC to the SAS for the response to the Keep Alive Request. The Keep Alive Response shall be generated by encoding theMessageContainerobject in which empty JSON object (i.e. “{}”) is used to generate encodedPayloadData field.

9.0.17.2.5 Payload Data for SAS Deregistration Message

9.0.0.17.2.5.1 SAS Deregistration Request

SAS Deregistration Request may be sent by the SAS to deregister from the ESC when the SAS wants to stop receiving indications from the ESC. The SAS Deregistration Request shall be generated by encoding the MessageContainer object in which the SasDeregistrationRequest object is used to generate encodedPayloadData field using JSON.

Table 13: SasDeregistrationRequest object

Field	R/O/C	Descriptions
NAME: sasAdministratorId DATA TYPE: string	Required	This field shall be included to indicate which SAS Administrator manages the SAS. The format of this field shall be same as $ADMINISTRATOR_ID used in the SAS-SAS Protocol [n.5].
NAME: sasRegistrationId DATA TYPE: string	Required	This field shall be generated by the ESC and included to indicate the registration identifier for the SAS.

9.0.0.27.2.5.2 SAS Deregistration Response

SAS Deregistration Response shall be sent by the ESC to the SAS for the response to the SAS Deregistration Request. The SAS Deregistration Response shall be generated by encoding the MessageContainer object in which empty JSON object (i.e. “{ }”) is used to generate encoded PayloadData field.

11Annex A (Normative) DPA State Machine

SAS and ESC employing the SAS-ESC Protocol specified in this present document shall consider the DPA State Machine. This DPA State Machine shall be considered in each channel per DPA. The figure below shows the DPA State Machine in a channel per DPA.

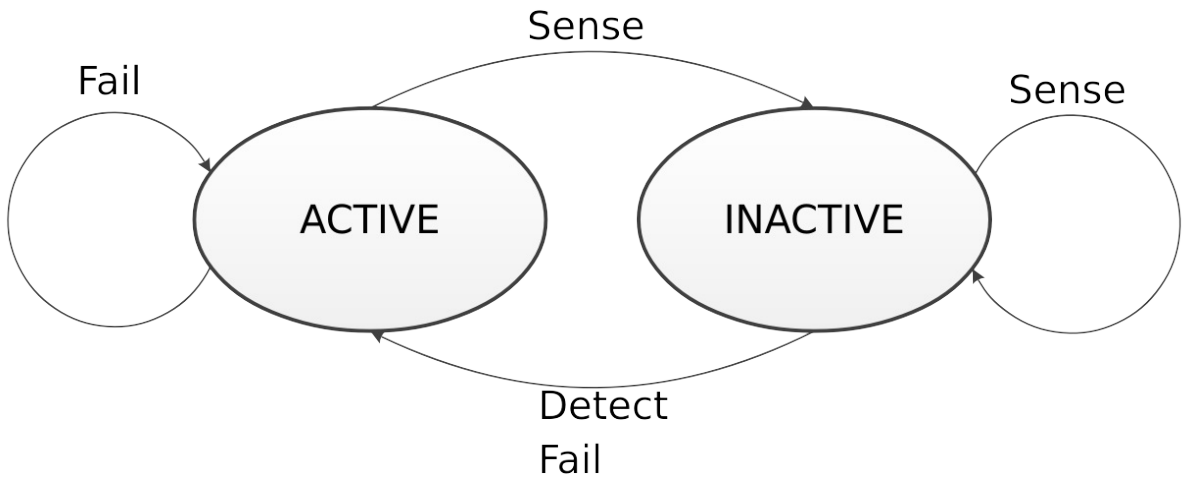

Figure 2: DPA State Machine in a channel per DPA

The DPA State shall be defined as follows:

- ACTIVE: “ACTIVE” refers to the state of a DPA in which the ESC detects the presence of the incumbent, in which two hours haves not passed after the ESC senses disappearance of the presence of any incumbents, or in which CBSDs are forbidden to use .a channel in its Neighborhood Area regardless of the presence of the incumbents. Both SAS and ESC shall consider “ACTIVE” the initial state of a DPA.
- INACTIVE: “INACTIVE” refers to the state of a DPA in which CBSDs are allowed to use. a channel in its Neighborhood Area, in which two hours have passed after the ESC sensed disappearance of the presence of any incumbents when the DPA was ACTIVE State, or in which the ESC senses no presence of the incumbents.

The following trigger events shall be applied to the DPA State Transitions:

- Sense: “Sense” refers to an event in which the ESC senses no presence of the incumbents or in which two hours has passed after the ESC sensed disappearance of the presence of the incumbent. All the DPA States shall transition to “INACTIVE” after this trigger event.
- Detect: “Detect” refers to an event in which the ESC detects the presence of the incumbent in the DPA. All the DPA States shall transition to “ACTIVE” after this trigger event.

Fail: “Fail” refers to an event in which the SAS detects any failures in the Keep Alive Message exchange with the ESC. In particular, if the SAS doesn’t receive any Keep Alive Response for more than x seconds after sending a Keep Alive Message, it shall move to the “Fail” state.

- All the DPA States shall transition to “ACTIVE” after this trigger event.

12Abbreviations

For the purposes of the present document, the following abbreviations apply:

Abbreviation	Full Form
API	Application Programming Interface
CBRS	Citizens Broadband Radio Service
DPA	Dynamic Protection Area
EIRP	Effective Isotropic Radiated Power
ESC	Environmental Sensing Capability
FFS	For Further Study
FCC	Federal Communications Commission
JSON	JavaScript Object Notation
HTTP	Hyper Text Transfer Protocol
HRRPS	HTTP plus TLS Protocol
RAT	Radio Access Technology
SAS	Spectrum Access System
TLS	Transfer Layer Security
URL	Uniform Resource Locator
WinnForum	Wireless Innovation Forum

13Definitions

13.1Citizens Broadband Radio Service (CBRS)

Wireless operations authorized by the U.S. Federal Communications Commission (FCC) in the 3,550-3,700 MHz frequency band. The CBRS includes Priority Access and General Authorized Access tiers of service.

13.2Environmental Sensing Capability (ESC)

A system that detects and communicates the presence of a signal from an Incumbent User to an SAS to facilitate shared spectrum access consistent with 47 C.F.R. Part 96.

13.3ESC Operator

A legal entity that the FCC has authorized to operate an ESC.

13.4Incumbent User

A federal entity authorized to operate on a primary basis in accordance with the table of frequency allocations, a fixed satellite service operator, or a Grandfathered Wireless Broadband Licensee authorized to operate on a primary basis on frequencies designated in section 96.11.

13.5Protection

To avoid harmful interference from lower-tier user(s) to the upper tier user(s).

13.6SAS Administrator

A legal entity that the FCC has authorized to administer the operation of a SAS.

13.7Spectrum Access System (SAS)

A system that authorizes and manages the use of spectrum for the CBRS in accordance with subpart F of 47 C.F.R. Part 96.

14Related RFCs

14.1RFC 7159

14.2The JavaScript Object Notation (JSON) Data Interchange Format

https://tools.ietf.org/html/rfc7159

14.3RFC 7515

14.4JSON Web Signature (JWS)

https://tools.ietf.org/html/rfc7515

14.5RFC 4648

14.6The Base16, Base32, and Base64 Data Encodings

https://tools.ietf.org/html/rfc4648

14.7RFC 6749

14.8OAuth 2.0 for automated identification and token rotation

https://tools.ietf.org/html/rfc6749

14.9RFC 7516

14.10JSON web encryption
https://tools.ietf.org/html/rfc7516

14.11RFC 7517

14.12JSON web key
https://tools.ietf.org/html/rfc7517

14.13RFC 7518

14.14JSON web algorithm
https://tools.ietf.org/html/rfc7518

14.15RFC 7519

14.16JSON web token
https://tools.ietf.org/html/rfc7519

14.17RFC 7797

14.18JWS for unencoded payload
https://tools.ietf.org/html/rfc7797