**Citizens Broadband Radio Service (CBRS);**

**Technical Specifications;**

**Application Programming Interface (API) for Interface between Spectrum Access System (SAS) and Environmental Sensing Capability (ESC)**

**(Release 1)**

*Editor’s Note: Message container definition and ESC State Machine required.*

# Contents

**[1. Scope 5](#_Toc494975846)**

**[2. References 5](#_Toc494975847)**

**[2.1 Normative references 5](#_Toc494975848)**

**[2.2 Informative references 5](#_Toc494975849)**

**[3. Definitions and abbreviations 5](#_Toc494975850)**

**[3.1 Definitions 5](#_Toc494975851)**

**[3.2 Abbreviations 6](#_Toc494975852)**

**[4. Prerequisite operations 6](#_Toc494975853)**

**[5. Message exchange flow and message types 7](#_Toc494975854)**

**[6. Message encoding and transport 8](#_Toc494975855)**

**[6.1 Message encoding 8](#_Toc494975856)**

**[6.1.1 JSON encode 8](#_Toc494975857)**

**[6.2 Message transport 8](#_Toc494975858)**

**[7. SAS-ESC Messages 9](#_Toc494975859)**

**[7.1 Message Container 9](#_Toc494975860)**

**[7.2 Payload Data 10](#_Toc494975861)**

**[7.2.1 Payload Data for SAS Registration Message 10](#_Toc494975862)**

**[7.2.1.1 SAS Registration Request 10](#_Toc494975863)**

**[7.2.1.2 SAS Registration Response 10](#_Toc494975864)**

**[7.2.1.2.1 Enhancements to](#_Toc494975865) *[EscSensorData](#_Toc494975865)* [object for SAS-ESC Interface 11](#_Toc494975865)**

**[7.2.2 Payload Data for ESC Information Update Message 14](#_Toc494975866)**

**[7.2.2.1 ESC Information Update Indication 14](#_Toc494975867)**

**[7.2.2.2 ESC Information Update Confirm 15](#_Toc494975868)**

**[7.2.3 Payload Data for DPA Activation Status Message 15](#_Toc494975869)**

**[7.2.3.1 DPA Activation Status Indication 15](#_Toc494975870)**

**[7.2.3.2 DPA Activation Status Confirm 16](#_Toc494975871)**

**[7.2.4 Payload Data for Keep Alive Message 16](#_Toc494975872)**

**[7.2.4.1 Keep Alive Request 16](#_Toc494975873)**

**[7.2.4.2 Keep Alive Response 16](#_Toc494975874)**

**[7.2.5 Payload Data for SAS Deregistration Message 17](#_Toc494975875)**

**[7.2.5.1 SAS Deregistration Request 17](#_Toc494975876)**

**[7.2.5.2 SAS Deregistration Response 17](#_Toc494975877)**

**[Annex A (Normative) DPA State Machine 18](#_Toc494975878)**

**[Document History 20](#_Toc494975879)**

# 1. Scope

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

# 2. References

## 2.1 Normative references

[n.1] “WINNF-TS-0112 Requirements for Commercial Operation in the U.S. 3550-3700 MHz Citizens Broadband Radio Service Band v1.3.0”, Wireless Innovation Forum, *To Be Published in [Date]*

[n.2] " RFC-7159 The JavaScript Object Notation (JSON) Data Interchange Format", IETF, March 2014.

[n.3] “WINNF-TS-0065 CBRS Communications Security Technical Specification v1.1.0”, Wireless Innovation Forum, 26 July 2017

[n.4] “RFC-2616 Hypertext Transfer Protocol -- HTTP/1.1”, Fielding, Gettys, Mogul, Frystyk, Masinter, Leach and Berners-Lee, June 1999.

[n.5] “WINNF-TS-0096 Signaling Protocols and Procedures for Citizens Broadband Radio Service (CBRS): Spectrum Access System (SAS) - SAS Interface Technical Specification v1.1.0”, Wireless Innovation Forum, 1 August 2017

[n.6] “WINNF-TS-0016 Signaling Protocols and Procedures for Citizens Broadband Radio Service (CBRS): Spectrum Access System (SAS) - Citizens Broadband Radio Service Device (CBSD) Interface Technical Specification v1.1.0”, Wireless Innovation Forum, 18 July 2017

[n.7] RFC-7515, JSON Web Signature (JWS), Jones, Bradley and Sakimura, May 2015.

[n.8] RFC-4648, The Base16, Base32, and Base64 Data Encodings, Josefsson, October 2006.

## 2.2 Informative references

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

*Editor’s Note: Not applicable for now*

# 3. Definitions and abbreviations

## 3.1 Definitions

**Citizens 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.

**Environmental 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.

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

**Incumbent 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.

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

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

**Spectrum 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.

## 3.2 Abbreviations

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

API Application Programming Interface

CBRS Citizens Broadband Radio Service

DPA Dynamic Protection Area

EIRP Effective Isotropic Radiated Power

ESC Environment Sensing Capability

FFS For Further Study

FCC Federal Communications Commission

JSON JavaScript Object Notation

HTTP Hyper Text Transfer Protocol

HTTPS HTTP plus TLS protocol

RAT Radio Access Technology

SAS Spectrum Access System

TLS Transfer Layer Security

URL Uniform Resource Locator

WInnForum Wireless Innovation Forum

# 4. Prerequisite operations

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

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

*Editor’s Note: New section as per discussion in 26th September 2017.*

# 5. Message 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.

# 6. Message encoding and transport

## 6.1 Message encoding

### 6.1.1 JSON 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.

## 6.2 Message 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.

# 7. SAS-ESC Messages

## 7.1 Message 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]. |

## 7.2 Payload Data

### 7.2.1 Payload Data for SAS Registration Message

#### 7.2.1.1 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.

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. |

#### 7.2.1.2 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.

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: object *EscInformation* | 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*  DATA TYPE: array of object: *EscSensorData* [5] | Required | This field shall be included to indicate the information of ESC sensors 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.*

##### 7.2.1.2.1 Enhancements to *EscSensorData* object for SAS-ESC Interface

In this specification, the *EscSensorData* object specified in the SAS-SAS Protocol [n.5] shall be reused with enhancements. Enhanced definition of the *EscSensorData* object 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: object *InstallationParam* | N/A | This field shall not be included. |
| NAME: *escInstallationParam*  DATA TYPE: object *EscInstallationParam* | Required | This field shall be included to indicate the installation parameters of the ESC Sensor identified by the *sensorId* field 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 the *sensorId* field 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 the *heightType* parameter value is “AGL”, the antenna height shall be given relative to ground level. When the *heightType* parameter 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 of the ESC Sensor antenna horizontal location. |
| NAME: *verticalAccuracy*  DATA TYPE: number | Required | A positive number in meters to indicate accuracy of the ESC Sensor vertical 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 an antenna 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’*. |

### 7.2.2 Payload Data for ESC Information Update Message

#### 7.2.2.1 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 the *MessageContainer* object in which JSON-encoded *EscInformationUpdate* object is used to generate *encodedPayloadData* field.

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: object *EscInformation* | Required | This field shall be included to indicate the ESC information. |

#### 7.2.2.2 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.

### 7.2.3 Payload Data for DPA Activation Status Message

### 7.2.3.1 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.

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).*

#### 7.2.3.2 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.

### 7.2.4 Payload Data for Keep Alive Message

#### 7.2.4.1 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.

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. |

#### 7.2.4.2 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 the *MessageContainer* object in which empty JSON object (i.e. “{}”) is used to generate *encodedPayloadData* field.

### 7.2.5 Payload Data for SAS Deregistration Message

#### 7.2.5.1 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.

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. |

#### 7.2.5.2 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 *encodedPayloadData* field.

# Annex 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 have 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.

*Editor’s note: shall we define “x” as a predefined value or shall we make it configurable?*

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

# Document History

|  |  |  |
| --- | --- | --- |
| **Document History** | | |
| v0.5.0 | 5th September 2017 | 1st Draft |
| v0.5.1 | 4th October 2017 | Updated based on the discussion in 26th September 2017.   * New section for prerequisite operations. * Added Message Container definition. * Normative Annex for DPA State Machine * Section renumbering |