IEEE Standard for Information Technology— Telecommunications and information exchange between systems Wireless Regional Area Networks (WRAN)— Specific requirements

Part 22.3: Spectrum Characterization and Occupancy Sensing

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved Date XX

IEEE-SA Standards Board

Abstract: This standard specifies the architecture, abstraction layers, interfaces and metadata requirements for Spectrum Characterization and Occupancy Sensing (SCOS) system, and defines performance parameters, units and measures. This SCOS system comprises one or more semi-autonomous Spectrum Sensing Devices which scan electromagnetic spectrum, digitize it and perform processing, transmitting the resultant data with appropriate metadata to a central storage and processing system, according to rules, policies or instructions imposed on the Spectrum Sensing Devices by a management system.

Keywords: radio spectrum sensing, spectrum monitoring, signal characterization, cognitive radio, IEEE 802.22.3, WRAN standards

1

[1] The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: **ISBN 978-0-7381-6723-7** **STD97146**
Print: **ISBN 978-0-7381-6724-4** **STDPD97146**

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.
The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

This standard specifies the functional elements, system architecture, abstraction layers, interfaces and metadata requirements for Spectrum Characterization and Occupancy Sensing (SCOS) system, with some limited definition of performance parameters, units and measures. It is intended to incorporate elements of existing standards and technology components to make it fast to implement using "off the shelf" hardware and software modules. The standard is intended to be flexible to make it forward-compatible as both radio sensing hardware and software technology develops, with an emphasis on using shared, virtualized, Internet-connected computing resources. The reference architecture describes one or more semi-autonomous Spectrum Sensing Devices which scan electromagnetic spectrum, digitize it and perform some level of processing, transmitting the resultant data with appropriate metadata to a Spectrum Sensing Management System. This command and control system manages scan requests from users, manages and advertises to users the scanning resources available to it from its connected Sensing Devices, and packages and forwards scan data to specified destinations according to rules, policies or instructions imposed by operator of the SCOS system.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association web site at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA web site at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. A patent holder or patent applicant has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses. Other Essential Patent Claims may exist for which a statement of assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this standard was submitted to the IEEE-SA for approval, the following voting members had participated in the IEEE P802.22.3 Task Group:

TBC

Major contributions to this standard were made by the following individuals:

TBC

The following members of the balloting committee voted on this **TBC**. Balloters may have voted for approval, disapproval, or abstention.

When the IEEE-SA Standards Board approved this on TBC, it had the following membership:

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Notes:

For collaboration, Google doc is being used.. Need to go back to Microsoft word IEEE standard draft for the IEEE formatting purposes. (In fact, we have already lost IEEE standard draft formatting after we converted the draft to Google doc.) *The main idea of Google doc is collaboration and accelerating the pace of creating more content.*

At this point, I am focusing on adding missing content as outlined in the above table. Once all content is added and we have high level agreement, I am planning to spend time on proofreading (figure, table numbering, grammar, formatting, etc). Please let me know your suggestions/comments. Thanks! - Nilesh

Suggestions for the Plenary meeting

Here are some of the key discussion threads that we need to close to finalize content

- Distinction between role and entity (Section 5.1)
  - for example SD is an entity and SD owner is a role that identifies organization/individual that deploys and has administrative and physical control over SD.
  - for example, DC is an entity and sensing data administrator is a role that identifies organization/individual that deploys and has administrative and physical control over the Data Clients consisting of data stores.
- SCOS Modes of operation
  - In the last call, we discussed a mode in which sensing manager is not performing the schedule. Should we add a mode in which SD is responsible for task scheduling?
- Discussion on the new content
  - Administration, security, policy (Section 7)
  - Entity descriptions (SD, SM, DM, TA, DC), model, and metadata
  - The 5 Interfaces (TA-SM, SM-SD, SM-DM, SD-DM, DM-DC) and associated messages (Section 6)
- Is the standard too heavy?
  - Is it possible to make the standard compact? How?
    - Are there any redundant parts?
- Here are some of the weak points in the technical content that need attention
  - SM Proxy
  - CR Mode

Link to Google doc:

Contents

IEEE Standard for Information Technology— Telecommunications and information exchange between systems Wireless Regional Area Networks (WRAN)— Specific requirements

Part 22.3: Standard for Spectrum Characterization and Occupancy Sensing

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

1.1 Scope

The purpose is to establish a high-level architecture that supports a diverse and evolving range of sensor technologies, data types, use cases, and installations. Further, the standard is developed to promote business models for users of the architecture (i.e., System Owner, Sensor Owner, and Data Client) with associated profit incentives, regulatory interoperability, and support for specialization (e.g., Internet services, radiofrequency engineering, big data analytics). The focus of this standard is on management and control of the sensor network (sensorNet) and data transmission to the Data Client. Attributes associated with Data Client (i.e., data management, analysis, and visualization) are decidedly not within the scope of this standard.

Recently, many national communication regulators have proposed and supported dynamic spectrum access methods with the aim of optimizing spectrum use in an efficient manner. To improve spectrum efficiency and spectrum sharing, without causing disturbances or interferences to licensed spectrum user, measurement systems able to provide spectrum occupancy information at a particular location and at a particular time are needed. This information can be used in different ways. For example, it can be used to provide spectrum occupancy information to spectrum sharing database services.
This standard realizes this need proposing the architecture of a Spectrum Characterization and Occupancy Sensing (SCOS) System.
The development of this standard has been pursued in order to improve spectrum use and support shared spectrum applications.

1.2 Purpose

The purpose of this standard is to define specifications for a Spectrum Characterization and Occupancy Sensing (SCOS) System. It describes operating characteristics and behaviors of the components of the SCOS system. Furthermore, it defines measurement parameters that must be evaluated. It includes protocols for reporting measurement information that enable coalescing the results from multiple such devices. It uses commonly used network transport mechanisms to achieve the control and management of the system. Interfaces and primitives are provided for conveying value added sensing information to various spectrum sharing database services.

1.3 Application

Various national regulators and government authorities are developing regulatory and policy frameworks to allow cooperative spectrum sharing approaches in order to optimize spectrum utilization. There is emphasis on greater spectrum efficiencies, spectrum sharing and spectrum utilization, which requires not only database-driven configuration of the radios, but systems that can provide spectrum occupancy at a particular location and at a particular time.

The IEEE 802.22.3 standard described in this document will help fulfil this need by creating a Spectrum Characterization and Occupancy Sensing (SCOS) system. This will improve knowledge of spectrum utilization and support shared spectrum applications, hence benefitting the regulators and users alike.

The Spectrum Occupancy Sensing (SCOS) System has many applications which include:
1. On-demand spectrum survey and report

2. Collaborative spectrum measurement and calibration
3. Labelling of systems using the spectrum
4. Spectrum planning
5. Spectrum mapping
6. Coverage analysis for wireless deployment
7. Terrain and topology - shadowing and fading analysis
8. Quantification of the available spectrum through spectrum observatories
9. Complement database access for spectrum sharing by adding in-situ awareness and faster decision making
10. Space-Time-Frequency spectrum hole identification and prediction where non-time-sensitive tasks can be performed at certain times and at certain locations, when the spectrum use is sparse or non-existent
11. Identification and geolocation of interference sources.

The Spectrum Characterization Occupancy Sensing (SCOS) systems may be deployed to characterize many bands such as VHF/UHF, L, S, C and X bands.

2. Normative References

Sections of the IEEE P1900.6 standard defining the M-SAPs.
To be completed…

3. Abbreviations and acronyms

CR	Cognitive Radio
DC	Data Client
DM	Data Manager
RF	Radio Frequency
SCOS	Spectrum Characterization and Occupancy Sensing
SD	Sensing Device
SM	Sensing Manager
TA	Tasking Agent

4. System Architecture

4.1 System Roles

The SCOS system identifies certain roles to meet the operational requirements and entities to meet the functional requirements. Roles are defined for individual/organizations and entities represent the software/hardware components in the system. The following roles have been identified based on operational requirements for the SCOS system.

Sensor Owner: The individual or organisation that deploys and has administrative and physical control over the sensing devices (SD). SDs are typically physical devices.

Sensing Data Administrator: The individual or organisation that deploys and has administrative and physical control over the Data Clients consisting of data stores or other consumers of spectrum sensing data delivered by the SCOS system.

SCOS Administrator: The individual or organisation that deploys and has administrative and physical control over the SCOS System, consisting of the Sensing Management System and Sensing Data Manager.

Tasking Agent: The individual or system that authenticates with the SCOS system and causes a scan activity to be scheduled.

4.2 System Architecture

Figure 1 illustrated the SCOS architecture comprised of the following key entities: Sensing Devices (SDs), Sensing Manager (SM), and Data Manager (DM).

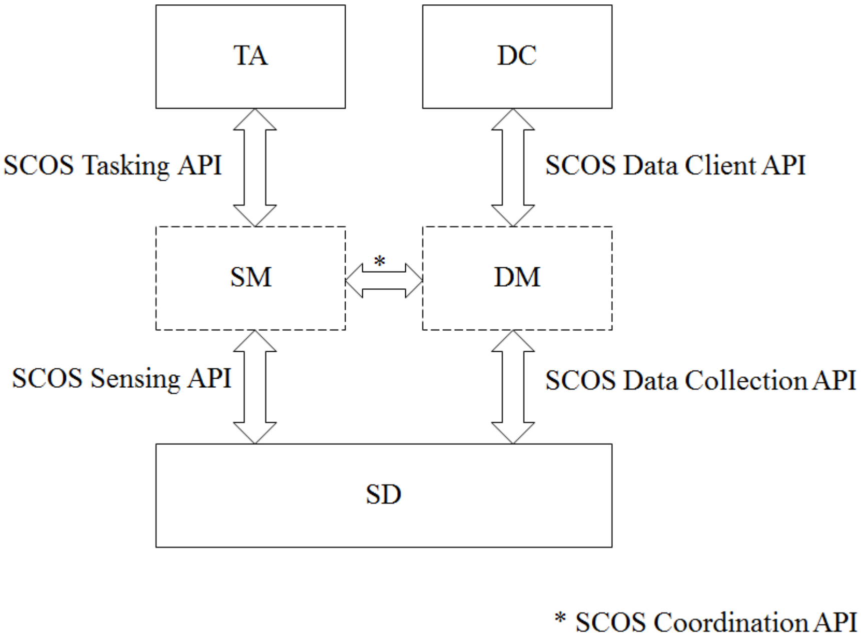


Figure 1: SCOS System Block Diagram.

The SCOS Platform is an aggregate entity composed of Sensing Manager and Data Manager. SCOS platform manages sensing tasks, sensing devices, and sensing data.

The SM handles sensing tasks from one or more TAs. The DM publishes sensing data to one or more DCs. Thus, the topology mapping for sensing tasks is hence N:1:N for TA:SM:SD. Similarly, the topology mapping for sensing data publishing is N:1:N for SD:DM:DC.

The SCOS Platform provides Tasking API to the Tasking Agents to initiate spectrum sensing tasks. The sensing tasks are scheduled by the SCOS platform on the sensing devices. The sensing devices send the sensing data to the SCOS platform. The SCOS platform publishes the sensing data to the Data Clients using SCOS Data Client API.

The SCOS Platform provides Sensing API and Data Collection API to the Sensing devices for the purpose of associating sensor devices with the platform, performing sensing operations, and collecting the sensing data.

The SCOS tasking API and SCOS sensing API together control the sensing functionality. This communication forms the *SCOS control plane*. The SCOS Data Collection API and SCOS Data Distribution API together are referred to as *SCOS data plane*.

The SCOS control-plane communication is synchronous; the data-plane communication is asynchronous.

It should be noted that within the SCOS system, the SDs shall not communicate with each other, or directly with the user of the SCOS system (TAs or DCs).

4.2.1 Entity Functions

- Tasking Agent is the entity that initiates a spectrum monitoring request to one or more Spectrum Sensing Managers (SM). Tasking Agents can be human or machine, and have various levels of privileges regarding what spectrum information collection can be initiated. Tasking Agents would determine where sensing data is to be transmitted, and authorization to access that data would rest with the owner of that data storage entity. and what spectrum information can be accessed from a Data Client.
    - An Tasking Agent (user of the SCOS system) and SM (Sensing Manager) communicate by REST API to ask for available resources, and request a scan.
- Data Client is a data store for storing spectrum information collected from the sensing network. There can be multiple DCs that sensing data is transmitted to by the Data Manager, and these can be, but not necessarily, associated with a specific Tasking Agent.
    - The Data Manager transmits data to the DC via a Message Queue, and the Tasking Agent interacts with the DC using their chosen mechanisms (out of scope of this standard)
- Spectrum Sensing Manager (SM or Sensing Manager) manages a collection of Spectrum Sensing Devices (SD). Requests for spectrum measurements from Tasking Agents are inserted into a scan schedule on the SM for all its attached SDs, as far as possible under a set of slot availability rules. This schedule is synched to the appropriate SDs associated with the SM. Data from the SDs are collected at the Data Manager for transmission to one or more DCs for long term storage and processing.
    - The SM is associated with SDs (Sensing Devices) through a synchronous interface, where the SM enumerates and holds a list of available resources for each SD.
    - The SM stores and manages a schedule of scans against the sensing resources, and synchronizes this schedule with all SDs both on a change being made and periodically to ensure correct state.
- Data Manager receives transmissions of packaged scan data from SDs, and retransmits it to one or more destinations, as defined by the policies associated with each Tasking Agent (the source of

scan requests)

  - The “Data Manager” applies any policies and then handles the Store & Forward to one or more DCs using a Message Queue or Streaming Mechanism
- The Sensing Manager and Data Manager together form the SCOS Manager, and each can be on the same platform or separate platforms.
- The Spectrum Sensing Device is the sensing hardware that collects the spectrum data requested by the SM on behalf of each Tasking Agent. The SDs may exist with various levels of sophistication. The less sophisticated might be capable of measuring only one band, at only one resolution with little on-board processing. Other sensors may incorporate sophisticated antenna techniques, multiple bands, calibration processes, on-board data processing and/or storage and/or be capable of mobile operation.
  - An SD performs the scans in the schedule, and transmits the data and associated metadata through an asynchronous interface (message queue, or real time stream) to a “Data Manager” that performs system data validation (i.e. that a transmission is received completely, partial scans are consolidated, etc).
- SM Proxy facilitates an SM talk to another, with the downstream SM appearing as if it were an SD with a set of resources it provides. This downstream SCOS system would need to be 802.22.3 compliant.
- SD Proxy enables an SM talk to any other proprietary sensing hardware, acting as a software translation mechanism that translates between commands/metrics/etc. It would need to be custom written for the particular device it talks to.

The flow of instructions and data is described in Figure 4: SCOS Functional Block Diagram.

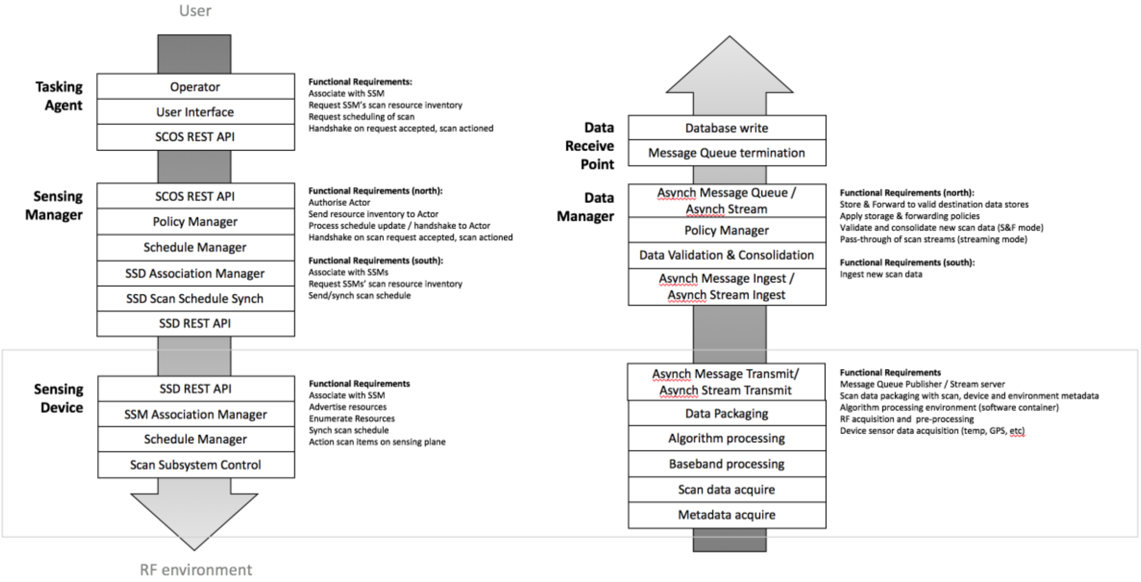

User
Tasking Agent
Operator
User Interface
SCOS REST API
Functional Requirements:
Associate with SSM
Request SSM's scan resource inventory
Request scheduling of scan
Handshake on request accepted, scan actioned
Sensing Manager
SCOS REST API
Policy Manager
Schedule Manager
SSD Association Manager
SSD Scan Schedule Synch
SSD REST API
Functional Requirements (north):
Authorise Actor
Send resource inventory to Actor
Process schedule update / handshake to Actor
Handshake on scan request accepted, scan actioned
Functional Requirements (south):
Associate with SSMs
Request SSMs' scan resource inventory
Send/synch scan schedule
Sensing Device
SSD REST API
SSM Association Manager
Schedule Manager
Scan Subsystem Control
Functional Requirements
Associate with SSM
Advertise resources
Enumerate Resources
Synch scan schedule
Action scan items on sensing plane
RF environment
Data Receive Point
Database write
Message Queue termination
Data Manager
Asynch Message Queue / Asynch Stream
Policy Manager
Data Validation & Consolidation
Asynch Message Ingest / Asynch Stream Ingest
Functional Requirements (north):
Store & Forward to valid destination data stores
Apply storage & forwarding policies
Validate and consolidate new scan data (S&F mode)
Pass-through of scan streams (streaming mode)
Functional Requirements (south):
Ingest new scan data
Asynch Message Transmit/ Asynch Stream Transmit
Data Packaging
Algorithm processing
Baseband processing
Scan data acquire
Metadata acquire
Functional Requirements
Message Queue Publisher / Stream server
Scan data packaging with scan, device and environment metadata
Algorithm processing environment (software container)
RF acquisition and pre-processing
Device sensor data acquisition (temp, GPS, etc)

Figure 2: SCOS Functional Block Diagram

4.3 System Workflow

The Tasking agents interact with Sensing Manager through the 802.22.3 defined Tasking API. The tasking API facilitates querying the available sensing resources and schedule sensing tasks as permitted by resource availability, authorisation level and system policies.

The sensing devices are associated with the SCOS system using the SCOS sensing API. The SM maintains an inventory of the sensing resources along with the SD capabilities and parameters described according to this 802.22.3 standard.

Following is the typical workflow within the SCOS system.

- The Tasking Agent submits a scanning task into a schedule managed by the SM using the SCOS Tasking API defined in this 802.22.3 standard
- The SM maintains the schedule of tasks and synchronises this schedule of tasks to the SD using the SCOS sensing API defined in this 802.22.3 standard
- Within the SD, radio energy shall be collected by an antenna and transferred through an interconnect to a signal conditioner. Conditioned signal will be then transferred to a signal processing system to produce a baseband signal that can be quantised and passed in digital form to be analysed through whichever sensing technique is offered by the SD
- The data from this analysis, and the associated metadata that includes the sensing technique and environment, hardware, software and configuration parameters as defined in this standard, can be temporarily stored on the local device before it shall be transmission to an end point.
- SD transmits the sensing data to the Data Manager using the SCOS Data Collection API.
- The DM establishes the endpoint of data package transmissions according to the Tasking Agent's nominated DCs, and in accordance with the policies defined. DM publishes the data to the DCs.
- SM and DM coordinate the task status.
- Finally, SM reports the success or otherwise for the sensing task back to the original Tasking Agent.

4.4 System Entity Models

4.4.1 Spectrum Sensing Device (SD)

SDs convert radiative electromagnetic energy into a voltage, which is then sampled. The samples can then be processed in various ways to provide information on the immediate RF environment, e.g., amplitude statistics versus frequency, amplitude and phase versus time at a given frequency, occupancy statistics, angle of arrival.

4.4.1.1 Hardware Model

A simplified hardware block diagram of a general SD model is depicted in 5. SD hardware designs are not required to have each component shown in the block diagram. Specifics for each component (e.g., presence, model, operational parameters), however, is required metadata when SD capabilities are queried by the SM. This SD definition metadata is also accompanied with the output data messages.

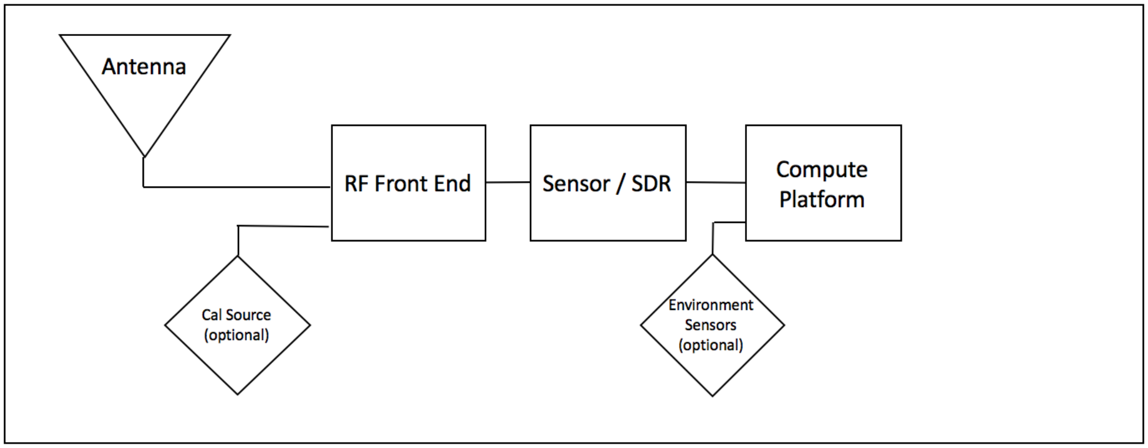


Figure 3: SD Simplified Hardware Model Block Diagram

The SD is composed of the following functional elements, as follows:

- Functional element 1 – Antenna: An antenna used to collect RF energy. This is fed to functional element 2 over a hardware interface (interconnect cable)
- Functional element 2 – Signal Conditioning Unit: An RF front end unit consisting of (all or some of) an RF switch (optional, with the ability to accept an optional calibration signal), filter, Low Noise Amplifier, mixer. This sends the conditioned signal to functional element 3 over an analogue hardware interface (interconnect cable/track).
- Functional element 3 – Signal Extraction Unit: Analogue Digital Converter, spectrum analyser or Software Defined Radio to act as a baseband processor, performing a demodulation of the conditioned signal and acquires the baseband signal. This sends a digitised signal over a digital interface (interconnect)
- Functional element 4 – Compute Platform: that provides
    - A signal processing function with a signal detection and/or classification algorithm. It sends detection/classification data to metadata consolidation and packaging function over a software interface.
    - A metadata consolidation and data packaging function that combines sensing data with environmental inputs (where implemented), hardware, operating and system-configured metadata. It sends data packages to the transmission system over a software interface.
    - A transmission unit that transmits scan data to the destination system over a best-effort IP connection.

The Compute Platform sends necessary command and control signals to Functional element 2 (Conditioning Unit) and Functional element 3 (Extraction Unit). It receives data from the Sensor/SDR, and polls any environment sensor input devices for necessary metadata items, such as GPS location. Interaction of the various elements is described in Figure 6: SD Functional Elements.

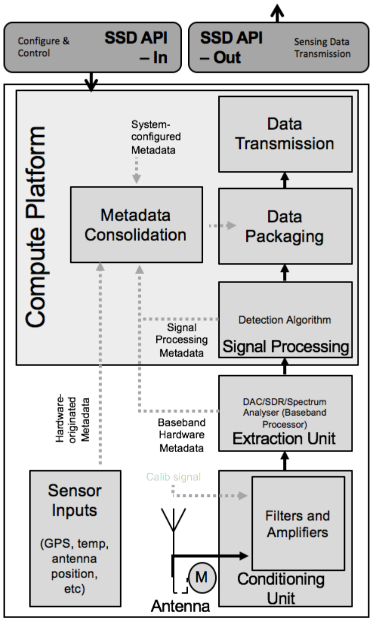


Figure 4: SD Functional Elements

This block diagram can be split into the hardware layer and the software processes that run alongside. These hardware blocks or software services generate metadata that is associated with each item.

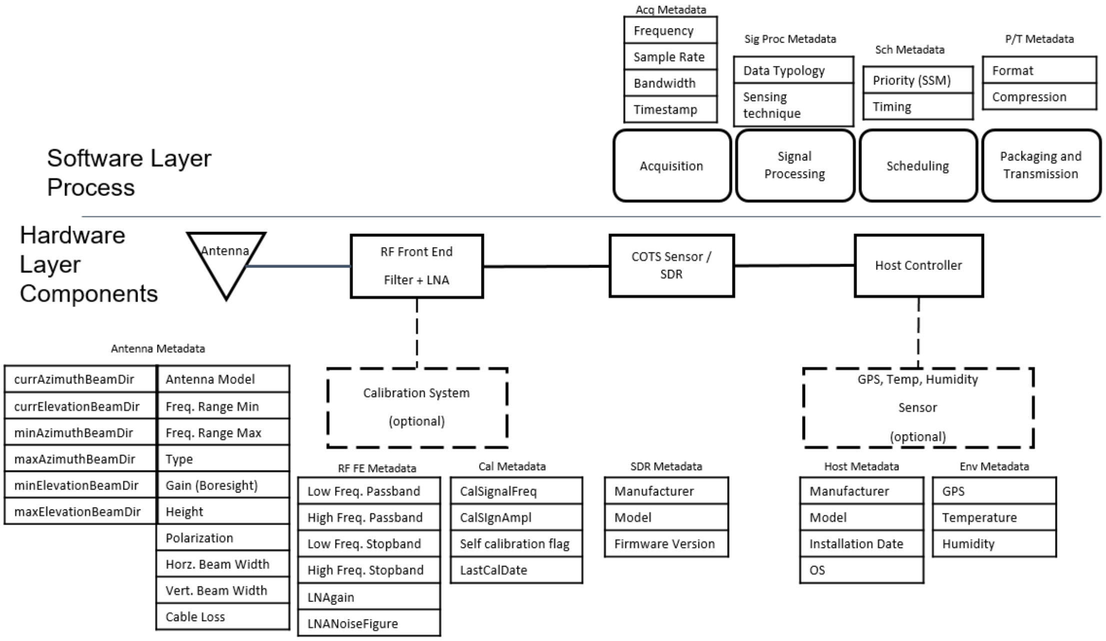


Figure 5: SD model - Hardware layer components and Software layer processes with relevant metadata

4.4.1.2 **SD Calibration Model**

A calibration can be done in the lab at build/commissioning time, and stored as a calibration file on the SD. Further, an SD with a self-calibration capability can be instructed through an administrative interface (not Tasking Agent request) to perform a calibration using a local calibration source.

Following table enumerates the parameter definition object for SD Calibration Model.

4.4.1.3 **SD Algorithm Model**

The algorithm model is described in terms of

- inputs into black box: the identity of the USER and SM requesting the scan, the measurement parameters, which algorithm is to be used; and
- outputs from the black box: the identity of the USER and SM requesting the scan, the requested scan parameters, the identification of the algorithm model, and the processed results.

Following table enumerates the parameter definition object for algorithmModel.

Parameter	R/O/C	Description
NAME: AlgorithmSet DATA TYPE: Array of String	Required	Names of algorithms supported by the SD. The maximum length of the ID string is 64 octets.

Following algorithms can be specified. At least once algorithm model needs to be supported by SD. Support for GenericEnergyDetection is normative.

Scan Algorithm	Description
GenericEnergyDetection	Normative.
CyclicFeatureDetection	
CustomScanAlgorithm	

The standard would allow development of advanced algorithms. For example, direction finding.

It is the responsibility of the SCOS Administrator to publish algorithm definitions externally. The implementation does not need to be publicly accessible.

[#Message]

Parameter	R/O/C	Description
NAME: TA ID DATA TYPE: String	Required	The TA ID. The maximum length of the ID string is 64 octets.
NAME: SM ID DATA TYPE: String	Required	The SM ID. The maximum length of the ID string is 64 octets.
NAME: SDScanParam DATA TYPE: Object of type SDScanParam	Required	The scan parameters object.
NAME: Algorithm DATA TYPE: String	Required	The SM ID. The maximum length of the ID string is 64 octets.

Metadata can be categorized into Classes, having different purposes:

- Class A (System Metadata) includes all pieces of data that are related to factory information and remain constant for the entire lifespan of the component (SSD);

- Class B (Current Status Metadata) includes data describing the actual configuration of the device, in terms of hardware (positioning, antenna configuration, battery level) and software (frequency settings, sampling rate, sensing algorithm, available local memory etc.);
- Class C (data related metadata), specifying parameters strictly related to performed sensing action (scanned time, timestamp, atmosphere conditions, amount of data to be read, estimated noise level);

Class A and Class C metadata are not subjected to any change since they are offered as a response to a specific query (in SSD association process and Sensing request, respectively).

Class B metadata are provided to SSM, after a specific user request, and can be subjected to modification and special settings by the Tasking Agent. They must be provided to the SSM before a scanning section starts, and they must be accompanied by an additional information bit, indicating their editing property (0, non-editable parameter; 1, editable parameter).

Each metadata must respect JSON message syntax and each message must contain the following fields:

- Name
  - This is a text field that contains the metadata name;
- Type
  - This field contain the data type [string|float|int|boolean];
- Editable
  - This field contain a boolean information. In particular it indicates the status of being editable of a specific piece of metadata (set to 0 for Class A and C, settable to 0 or 1 for Class B);
- Content
  - This field contain the content of the metadata.

4.4.1.4 SSD metadata specification

4.4.1.4.1 Top level hardware metadata

Parameter	Values	Description
Antenna	0	Number of antennas
Calibration source		Present/absent
RF switch		Present/absent
RFFilter		Present/absent
LNA		
Sensor		COTS/SDR

4.4.1.4.2 Antenna Metadata

Antenna metadata is reported in the table below. In the second column of the table the class of the metadata is specified.

Metadata Name	**Metadata class**
Antenna Model	Class A
Freq. Range Min	Class A
Freq. Range Max	Class A

Type	Class A
Gain (Boresight)	Class A
Polarization	Class A
Height	Class A
Horz. Beam Width	Class A
Vert. Beam Width	Class A
Min Azi. Beam Dir.	Class A
Max Azi. Beam Dir.	Class A
Min Elev. Beam Dir.	Class A
Max Elev. Beam Dir.	Class A
Curr. Azi. Beam Dir.	Class B
Curr. Elev. Beam Dir.	Class B
Cable loss	Class A

A detailed description of the field of each metadata is reported in the table below

Name	Type	Editable	Content
Antenna Model	string	“0”	It contains a string with the model of the antenna that is installed.
Freq. Range Min	float	“0”	Min frequency value expressed in Hz
Freq. Range Max	float	“0”	Max frequency value expressed in Hz
Type	string	“0”	Antenna type
Gain (Boresight)	float	“0”	Antenna gain expressed in dBi
Polarization	string	“0”	Antenna polarization [“VL”|“HL”|“LHC”|“RHC”|“Slant”]
Height	float	“0”	Antenna heigh in m.
Horz. Beam Width	float	“0”	Horizontal 3-dB beamwidth expressed in degrees
Vert. Beam Width	float	“0”	Vertical 3-dB beamwidth expressed in degrees
Min Azi. Beam Dir.	float	“0”	minimum direction of main beam in azimuthal plane expressed in degrees from N
Max Azi. Beam Dir.	float	“0”	maximum direction of main beam in azimuthal plane expressed in degrees from N
Min Elev. Beam Dir.	float	“0”	minimum direction of main beam in elevation plane expressed in degrees from horizontal plane
Max Elev. Beam Dir.	float	“0”	maximum direction of main beam in elevation plane expressed in degrees from horizontal plane
Curr. Azi. Beam Dir.	float	“0” if fixed antenna is used “1” if an antenna with beam steering capability is used.	Current direction of main beam in azimuthal plane expressed in degrees from N
Curr. Elev. Beam Dir.	float	“0” if fixed antenna is used “1” if an antenna with beam steering capability is used.	Current direction of main beam in elevation plane expressed in degrees from horizontal plane

Cable loss	float	“0”	Cable loss expressed in dB of the cable connecting the antenna with the RF front-end

4.4.1.4.3 RF Front-end metadata

RF Front-end metadata is reported in the table below. In the second column of the table the class of the metadata is specified.

Metadata Name	**Metadata class**
Low Freq Passband	Class A
High Freq Passband	Class A
Low Freq Stopband	Class A
High Freq Stopband	Class A
LNA Gain	Class A
LNA Noise Figure	Class A

A detailed description of the field of each metadata is reported in the table below

Name	Type	Editable	Content
Low Freq Passband	float	“0”	Low passband frequency evaluated at -1 dB and expressed in Hz
High Freq Passband	float	“0”	High passband frequency evaluated at -1 dB and expressed in Hz
Low Freq Stopband	float	“0”	Low stopband frequency evaluated at -60 dB and expressed in Hz
High Freq Stopband	string	“0”	High stopband frequency evaluated at -60 dB and expressed in Hz
LNA Gain	float	“0”	Low Noise Amplifier Gain expressed in dB
LNA Noise Figure	float	“0”	Noise Figure of LNA expressed in dB

4.4.1.4.4 Calibration Metadata

Calibration metadata is reported in the table below. In the second column of the table the class of the metadata is specified.

Metadata Name	**Metadata class**
Cal. Sig. Freq.	Class A
Cal. Sig. Ampl.	Class A
Self Calibration flag	Class A
Last Cal. Date	Class A

A detailed description of the field of each metadata is reported in the table below

Name	Type	Editable	Content
Cal. Sig. Freq.	float	“0”	Frequency of the internal calibration source expressed in Hz
Cal. Sig. Ampl.	float	“0”	Amplitude of the internal calibration source

			expressed in dB
Self Calibration flag	boolean	“0”	This is set to “1” if the sensor performs a periodical self calibration procedure. Otherwise it is set to “0” if the self calibration is performed after a user request
Last Cal. Date	string	“0”	The time stamp of the last calibration expressed as HH:MM:SS YYYY/MM/DD

4.4.1.4.5 SDR Metadata

SDR metadata is reported in the table below. In the second column of the table the class of the metadata is specified.

Metadata Name	**Metadata class**
SDR Manufacturer	Class A
SDR Model	Class A
Firmware version	Class A

A detailed description of the field of each metadata is reported in the table below

Name	Type	Editable	Content
SDR Manufacturer	string	“0”	Manufacturer of the sensor used
SDR Model	string	“0”	Model of the sensor used
Firmware version	string	“0”	Current firmware version

4.4.1.4.6 SSD Host Metadata

Host metadata is reported in the table below. In the second column of the table the class of the metadata is specified.

Metadata Name	**Metadata class**
Manufacturer	Class A
Model	Class A
Installation Date	Class A
OS	Class A

A detailed description of the field of each metadata is reported in the table below

Name	Type	Editable	Content
Manufacturer	string	“0”	Manufacturer of the host
Model	string	“0”	Model of the host
Installation Date	string	“0”	The date when SSD has been installed expressed as YYYY/MM/DD
OS	string	“0”	Operating System installed on the host

4.4.1.4.7 Environmental Metadata

Environment metadata is reported in the table below. In the second column of the table the class of the metadata is specified.

Metadata Name	**Metadata class**
GPS	Class C
Temperature	Class C
Humidity	Class C

A detailed description of the field of each metadata is reported in the table below

Name	Type	Editable	Content
GPS	Array of float	“0”	[Latitude expressed in decimal degrees (-90°-90°) Longitude expressed in decimal degrees (-180°-180°)
Temperature	float	“0”	Environment temperature expressed in K
Humidity	float	“0”	Environment relative humidity expressed in percentage

4.4.1.4.8 SSD Software configuration metadata

4.4.1.4.9 Acquisition Metadata

Acquisition metadata is reported in the table below. In the second column of the table the class of the metadata is specified.

Metadata Name	**Metadata class**
Frequency	Class B
Sample Rate	Class B
Bandwidth	Class B
Timestamp	Class C

A detailed description of the field of each metadata is reported in the table below

Name	Type	Editable	Content
Frequency	float	“1”	Center frequency of the channel where the SSD is currently tuned expressed in MHz
Sample Rate	float	“1”	Current sampling rate expressed in MS/s
Bandwidth	boolean	“1”	Bandwidth of the channel where the SSD is currently tuned expressed in MHz
Timestamp	string	“0”	The time information when the data has been acquired expressed as HH:MM:SS.mmm YYYY/MM/DD

4.4.1.4.10 Signal processing Metadata

Signal processing metadata is reported in the table below. In the second column of the table the class of the metadata is specified.

Metadata Name	**Metadata class**
Data Typology	Class B
Sensing Technique	Class B

A detailed description of the field of each metadata is reported in the table below

Name	**Type**	**Editabl e**	**Content**
Data Typology	String	"0"	Description of the received data domain. Possibilities are: ● I/Q capture ● Frequency transform ● Power spectral density
Sensing Technique	String	"0"	Sensing processing algorithm used by SDR. Possibilities are: ● Cyclostationarity ● Energy Detection ● Custom

4.4.1.4.11 Scheduling Metadata

Scheduling metadata is reported in the table below. In the second column of the table the class of the metadata is specified.

Metadata Name	**Metadata class**
Priority	Class B
Timing	Class B

A detailed description of the field of each metadata is reported in the table below

Name	Type	Editable	Content
Priority	String	"0"	Scheduling Scheme used for incoming request. Possibilities are: ● (FCFS) First Come First Served ● (RR) Round Robin ● Custom
Timing	String	"0"	Types of requests to be managed: ● On demand

			● Timed with periodicity

4.4.1.4.12 Packaging and transmission Metadata

Packaging and transmission metadata is reported in the table below. In the second column of the table the class of the metadata is specified.

Metadata Name	**Metadata class**
Format	Class C
Compression	Class C

A detailed description of the field of each metadata is reported in the table below

Name	Type	Editable	Content
Format	string	“0”	
Compression	string	“0”	

4.4.1.5 SSD Task Control metadata

4.4.1.5.1 Scheduler Specification

Algorithm	Value	Notes
Unspecified	0	
Host Controller	1	
Embedded Job Controller	2	
Multilevel	3	

4.4.1.5.2 SSD Output Specification

Algorithm	Value	Notes
Unspecified	0	Invalid
Time domain IQ	1	Default
Freq. domain IQ	2	
Time domain Amp, Phase	3	
Freq. domain Amp, Phase	4	

4.4.1.6 SCOS Association Metadata

Following table enumerates sensing device metadata for associating with SCOS.

Parameter	R/O/C	Description
NAME: SDName DATATYPE: string	Required	The name of the sensing device registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: SDMode DATA TYPE: Integer	Required	The mode in which sensing device operates. (1=online, 2=offline)
NAME: SDType DATA TYPE: Integer	Required	The type of the sensing device. (1=SDFull, 2=SDProxy)
NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: SDCertFile DATA TYPE: String	Required	The path of the SD certificate file. The maximum length of the ID string is 256 octets.
NAME: SDKeyFile DATA TYPE: String	Required	The name of the SD certificate file. The maximum length of the ID string is 256 octets.
NAME: SDCAFile DATA TYPE: String	Required	The name of the trusted certificate authority. The maximum length of the ID string is 256 octets.

4.4.2 Spectrum Sensing Manager (SM)

The SM is primarily responsible for sensing task management. The standard identifies following key functionalities within the SM entity

- Scan Scheduling
- Policy enforcement
- Communication with SDs, TAs, and DM
    - Tasking Interface with TAs
    - Sensing Interface with SDs
    - Sensing Coordination with DM

4.4.2.1 SM Task Scheduling

Scheduling requests from a TA are defined in terms of duration, time, repetition, etc, as well as a flag to indicate whether the desired scan slots are “Exact Time” slots or “Nearest Time” slots. The scheduler on the SM will use this to try meet the TA request (and either confirm the scan schedule is accepted or refused).

Following table enumerates the parameter definition object for scanTask.

Parameter	R/O/C	Description
NAME: TaskID DATA TYPE: String	Required	Unique ID for the Spectrum Scan. The maximum length of the ID string is 64 octets.
NAME: TaskDuration DATA TYPE: number	Required	Duration of scan in milliseconds.
NAME: TaskStartTime DATA TYPE: Time	Required	The start time for the task.
NAME: TaskRepeatInterval DATA TYPE: Number	Optional	The interval in seconds after which the task needs to be repeated.

NAME: TaskRepeatCount DATA TYPE: Number	Optional	The number of times the task needs to be repeated. The maximum length of the ID string is 64 octets.
NAME: TaskEndTime DATA TYPE: Time	Conditional	The end time for the task. If repeatInterval and repeatCount are specified, TaskEndTime is not required.
NAME: TaskAttributes DATA TYPE: Integer	Optional	Currently following task attributes can be specified. 0 = Exact time. 1 = Nearest time.
NAME: TaskOptions DATA TYPE: Integer	Optional	Custom options.

5.4.3.1 SM Simplified Model

Figure <#> shows a simplified model for SM.

- Sensor Management: SM maintains the information from sensors obtained during SD association message exchange. The sensor information is identified in Table <#>. This information is used by the SM during task scheduling for determining which set of sensors could be used for performing the scan.
- Sensing Task Management: SM maintains information about to-be-scheduled, scheduled, and on-going tasks. The information about to-be-scheduled tasks is used in task scheduling, the information about scheduled tasks is synchronized with the SDs. The information about the ongoing tasks is used in task query/coordination/notification with TAs, DM, and SDs.
- Policy Enforcement: A key part of SCOS administration is enforcing policies for spectrum sensing. The SM ensures that the TA issuing a scan request is authorized to perform the requested scan, the scanning parameters comply with the regulatory policy for the location, frequency, time, and resolution
- Task scheduling: Task scheduling involves enforcing policy for the requested scan, identifying a set of eligible and available sensors using the sensor information, associating chosen sensors to the task, assigning scanTaskID to the task, updating the task status in the task information.
- Sensing coordination: The SM needs to coordinate the status of tasks with DM for example, SM assigns a DM for each of the scheduledTasks. The scanTaskID assigned by the SM is used by SDs when SDs provide sensing data to the DM.
- SCOS Tasking API: The SM communicates with the TAs using the Tasking API, provides response to the methods under the Tasking APIs, provides notification to the TAs upon task completion or error events.
- SCOS Sensing API: The SM communicates with SDs using the Sensing API, receives the sensor capabilities, and provides a spectrum scan schedule to the sensors along with necessary information for sending sensing data to the DM.

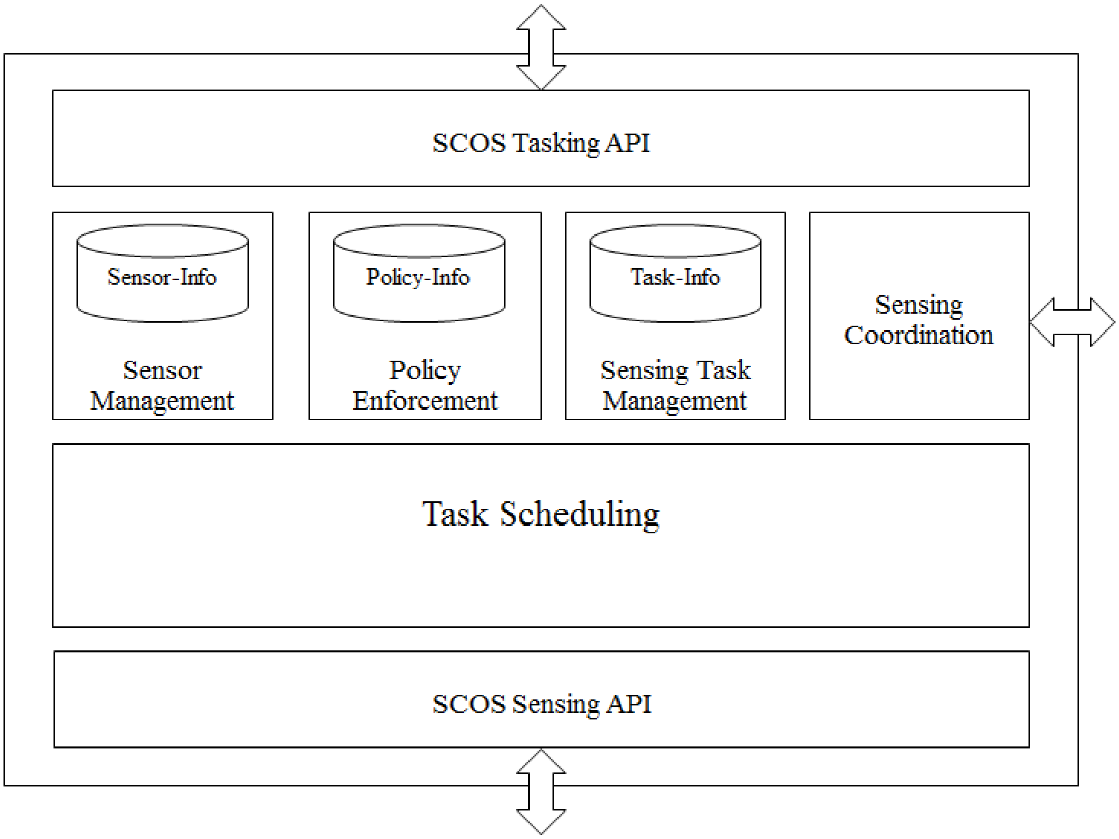


Figure 6: SMFunctional Block Diagram

5.4.3.2 SCOS Association Metadata

Following table enumerates sensing manager parameters toward associating with SCOS.

Parameter	R/O/C	Description
NAME: SMID DATA TYPE: String	Required	Unique ID for the Sensing Manager. The maximum length of the ID string is 64 octets.
NAME: SMSID DATA TYPE: String	Required	Unique ID for the SMS. The maximum length of the ID string is 64 octets.
NAME: SCOSOperator DATA TYPE: String	Required	The registered name of the SCOS operator. The maximum length of the ID string is 64 octets.
NAME: SMURL DATA TYPE: String	Required	The URL for reaching to the SM. The maximum length of the ID string is 256 octets.
NAME: SMCertFile DATA TYPE: String	Required	The path of the SM certificate file. The maximum length of the ID string is 256 octets.
NAME: SMKeyFile DATA TYPE: String	Required	The name of the SM certificate file. The maximum length of the ID string is 256 octets.
NAME: SMCAFile DATA TYPE: String	Required	The name of the trusted certificate authority. The maximum length of the ID string is 256

		octets.

4.4.3 Data Manager

The DM is primarily responsible for sensing data management. The standard identifies following key functionalities within the DM entity

- Collecting spectrum scan data
- Policy enforcement
- Communication with SDs, DCs, and SM
    - Data collection interface with SDs
    - Data distribution to the SDs
    - Sensing task coordination with the SM

4.4.3.1 Sensing Data Management

The SDs send spectrum measurements as requested by the scheduled scan. Table <#> describes the sensing data. The sensing data is identified by ScanTaskId, SDID, and timestamp. The SD also provides envInfo which includes environmental data including GPS, humidity, and temperature. The DM validates the received data, consolidates the data with other received data for the scanning task, stores it internally for further distribution to DCs, and provides acknowledgement to the receives sensing data to the SDs.

Parameter	R/O/C	Description
NAME: dataFormat DATA TYPE: Integer	Required	The format of the output data as specified in Table <#>.
NAME: sizeData DATA TYPE: Integer	Required	The number of measurements.
NAME: measData DATA TYPE: Array of Complex	Required	The complex measurement values. The size of the array is defined by sizeData.

4.4.3.2 DM Simplified Model

Figure <#> shows a simplified model for DM.

- Sensing data management: DM maintains data for the on-going tasks. The data received from the SDs needs to be stored internally for distribution to DCs. The DM may also need to have the capability to hold data for certain privileged tasks for a short duration identified by policy for the sensing data.
- Sensing data distribution: DM maintains the information regarding DCs and their associated scanning tasks. Sensing data distribution involves providing necessary reliability depending on the chosen transport..
- Policy enforcement: A key part of SCOS administration is enforcing policies on the sensing data. The DM ensures that the DCs issuing a subscription request for the sensing data are authorized to receive the data.
- Data validation and consolidation: The DM validates the data received from the SDs against the specified details from the task such as location, frequency, time, and measured data format. It

consolidates data based on scanning task requirements.

- Sensing coordination: The DM needs to coordinate the status of tasks with SM for example, the expected sensing data from specific SDs for specific sensing tasks; also, availability of DM resources for future scanning tasks which could be used SM in choosing the DM.
- SCOS Sensing Data Collection API: The DM communicates with the SDs using the Data Collection API. It receives data for specific sensing tasks from specified SDs as coordinated with the SM.
- SCOS Sensing Data Distribution API: The DM communicates with DCs, implements the API for distributing the sensing data to the authorized DCs.

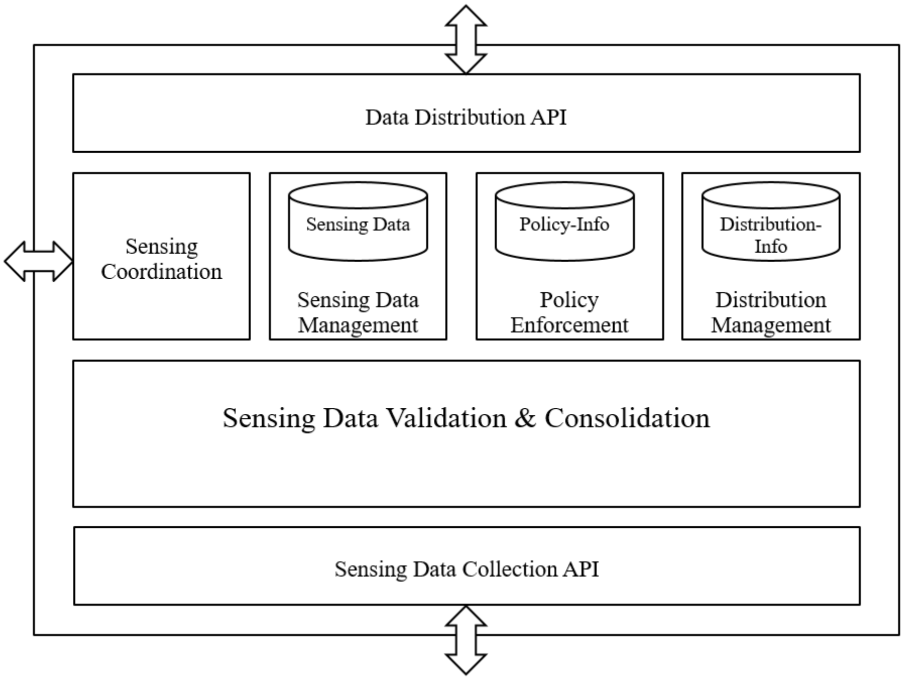


Figure 7: DM Functional Block Diagram

4.4.3.3 DM Metadata

The following table enumerates the parameter definition object for DataManager.

Parameter	R/O/C	Description
NAME: DMID DATA TYPE: String	Required	Unique ID for the Data Manager. The maximum length of the ID string is 64 octets.
NAME: SMSID DATA TYPE: String	Required	Unique ID for the SMS. The maximum length of the ID string is 64 octets.
NAME: SCOSOperator DATA TYPE: String	Required	The registered name of the SCOS operator. The maximum length of the ID string is 64 octets.
NAME: DMURL DATA TYPE: String	Required	The URL for reaching to the SM. The maximum length of the ID string is 256 octets.
NAME: DMCertFile DATA TYPE: String	Required	The path of the SM certificate file. The maximum length of the ID string is 256 octets.
NAME: DMKeyFile DATA TYPE: String	Required	The name of the SM certificate file. The maximum length of the ID string is 256

		octets.
NAME: DMCAFile DATA TYPE: String	Required	The name of the trusted certificate authority. The maximum length of the ID string is 256 octets.

4.4.4 **Tasking Agent**

The standard provides SCOS Tasking API to the TAs for performing spectrum scans. Using the API,

- TAs are associated with the SM of the SCOS platform
- TAs perform resource discovery
- TAs request spectrum scans at the desired time, frequency, and locations
- TAs specify the data clients that are authorized to receive the sensing data for the scans.
- TAs optionally perform query about the sensing tasks.

TAs communicate the information provided by the SM to the SCOS platform user. TAs may implement certain logic/policy to automate the spectrum scanning process. Figure <#> denotes a simplified model of a TA.

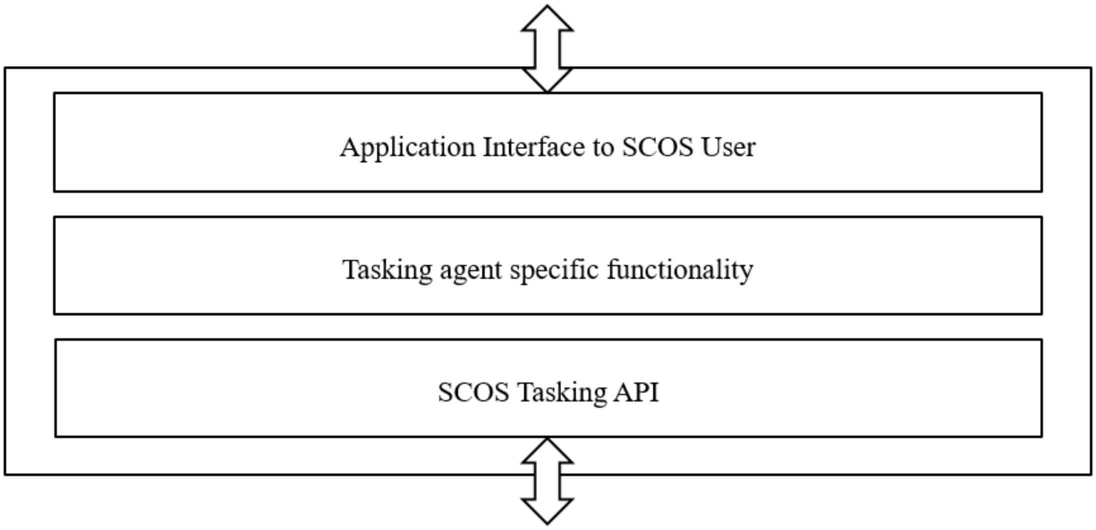

Application Interface to SCOS User
Tasking agent specific functionality
SCOS Tasking API

Figure 8: Tasking Agent Functional Block Diagram

4.4.4.1 **TA Metadata**

The following table enumerates the parameter definition object for Tasking Agent.

Parameter	R/O/C	Description
NAME: TAName DATA TYPE: string	Required	The name of the TA registered with the SCOS operator. The maximum length is 64 octets.
NAME: TAID DATA TYPE: String	Required	Unique ID for the TA. The maximum length of the ID string is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: TACertFile DATA TYPE: String	Required	The path of the TA certificate file. The maximum length of the ID string is 256 octets.
NAME: TAKeyFile DATA TYPE: String	Required	The name of the TA certificate file. The maximum length of the ID string is 256 octets.

NAME: TACAFile DATA TYPE: String	Required	The name of the trusted certificate authority. The maximum length of the ID string is 256 octets.

4.4.5 Data Client

The standard provides SCOS Data Distribution API to the DCs for requesting the spectrum sensing data. Using the API,

- DCs are associated with the DM of the SCOS platform
- DCs request sensing data for specific sensing tasks
- DCs receive the data streamed or forwarded by the DM

The DCs would typically ingest and store the received data as defined the SCOS user specific logic/policy. Figure <#> denotes a simplified model of a DC.

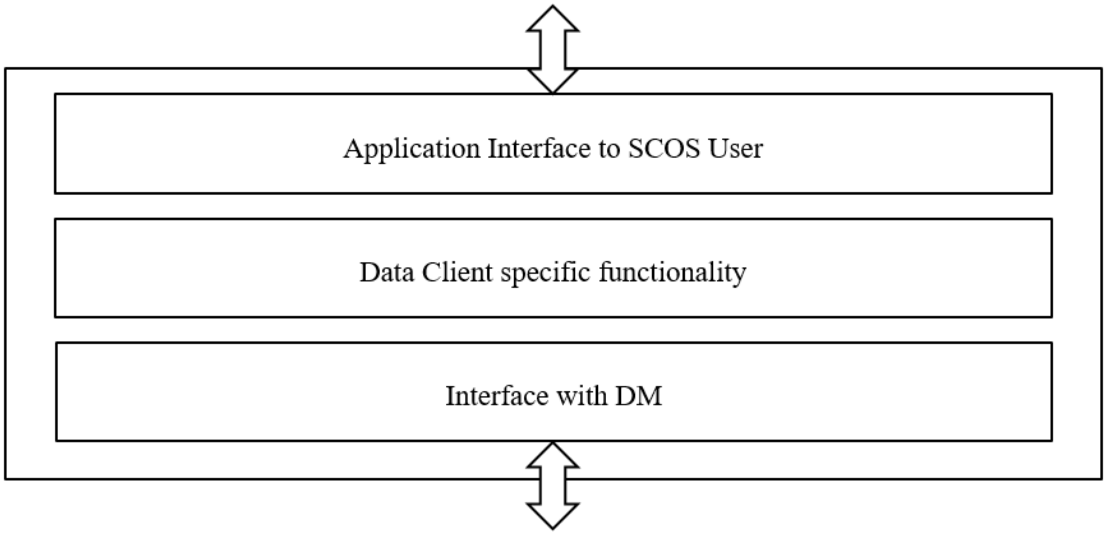

Application Interface to SCOS User
Data Client specific functionality
Interface with DM

Figure 9: Data Client Functional Block Diagram

4.4.5.1 DC Metadata

The following table enumerates the parameter definition object for a data client.

Parameter	R/O/C	Description
NAME: DCName DATA TYPE: string	Required	The name of the DC registered with the SCOS operator. The maximum length is 64 octets.
NAME: DCID DATA TYPE: String	Required	Unique ID for the DC. The maximum length of the ID string is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: DCCertFile DATA TYPE: String	Required	The path of the DC certificate file. The maximum length of the ID string is 256 octets.
NAME: DCKeyFile DATA TYPE: String	Required	The name of the DC certificate file. The maximum length of the ID string is 256 octets.

NAME: DCCAFile DATA TYPE: String	Required	The name of the trusted certificate authority. The maximum length of the ID string is 256 octets.

4.4.6 SD Proxy

[#Needs revision] The SD Proxy interfaces with SCOS on the behalf of proprietary SD. SD proxy interaction to SCOS is identical to that of the SD except that while registering it registers with SDType as SDProxy. Following diagram illustrates the SD-Proxy's functional architecture.

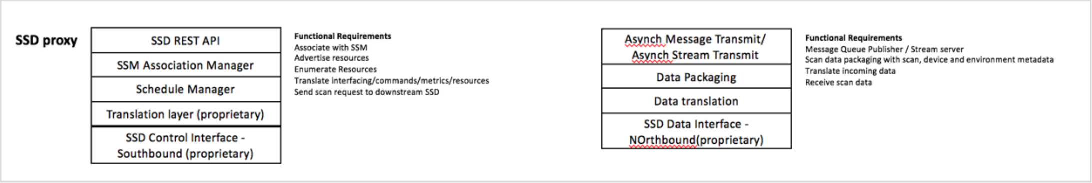

SSD proxy
SSD REST API
SSM Association Manager
Schedule Manager
Translation layer (proprietary)
SSD Control Interface - Southbound (proprietary)
Functional Requirements
Associate with SSM
Advertise resources
Enumerate Resources
Translate interfacing/commands/metrics/resources
Send scan request to downstream SSD
Asynch Message Transmit/ Asynch Stream Transmit
Data Packaging
Data translation
SSD Data Interface - NOrthbound(proprietary)
Functional Requirements
Message Queue Publisher / Stream server
Scan data packaging with scan, device and environment metadata
Translate incoming data
Receive scan data

Figure 10: SD Proxy Functional Block Diagram

4.4.7 SM Proxy

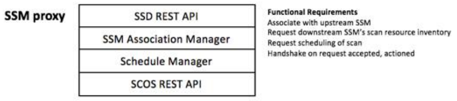

SSM proxy
SSD REST API
SSM Association Manager
Schedule Manager
SCOS REST API
Functional Requirements
Associate with upstream SSM
Request downstream SSM's scan resource inventory
Request scheduling of scan
Handshake on request accepted, actioned

Figure 11: SM Proxy Functional Block Diagram

[#Needs discussion (This demands SM-SM interface) and revision] The SM Proxy enables cascading of the SCOS system. There are two options for cascading

- SM Proxy acts as Proxy for all the underlying SDs and registers them with the SCOS system. Thus, all interactions with the actual underlying SDs are mediated by the SM-Proxy.
- SM-Proxy uses SM-SM interface and the two SMs interoperate with each other. This mode is planned to be introduced in the later version of the standard.

5. Interfaces, Messaging and Primitives

Figure 8 illustrates a simplified SCOS interactions model.

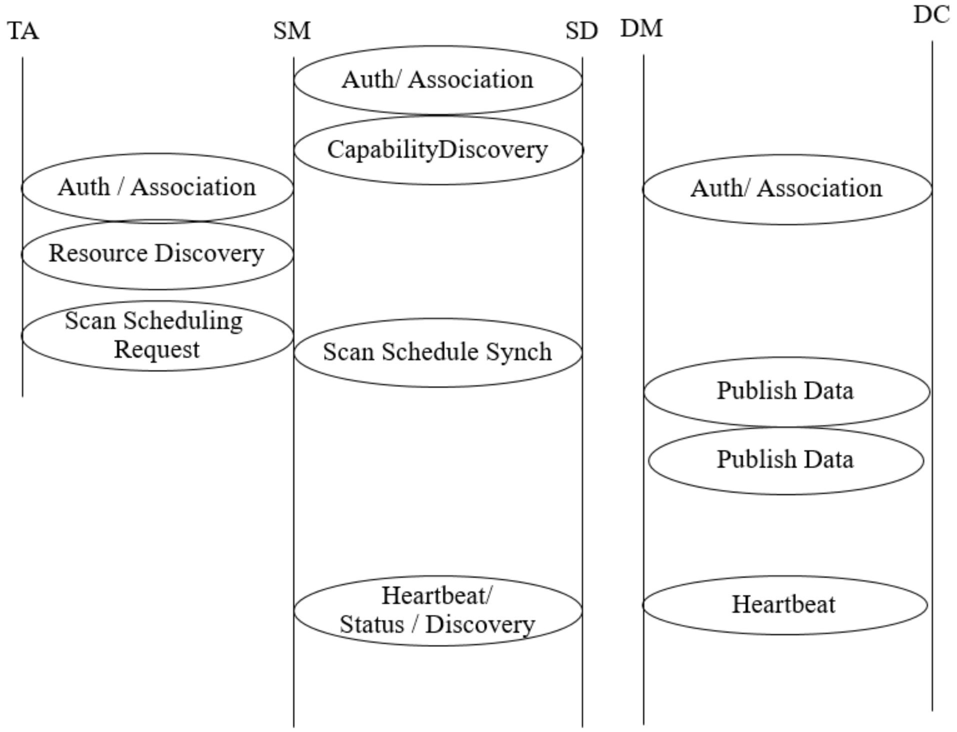


Figure 12. Simplified Interactions Model

5.1 SCOS Interfaces

5.1.1 SCOS communication interfaces

Following are the key SCOS communication interfaces.

- TA and SM Interface
    - The interface between TA and SM is asynchronous.
    - The interactions on this interface are specified in the SCOS Tasking API.
- SM and SD Interface
    - The interface between SM and SD is required to be synchronous.
    - The interactions on this interface are specified in the SCOS Sensing API.
- SD and DM Interface
    - The interface between SD and DM is asynchronous.
    - The interactions on this interface are specified in the SCOS Data Collection API.

- DM and DC Interface
    - The interface between SD and DM is required to be asynchronous.
    - The interactions on this interface are specified in the SCOS Data Client API.

5.1.2 Tasking Agent to SM Interface

5.1.2.1 Authentication and Registration

These procedures define the association and authentication process for an SM and Tasking Agent entity to connect and communicate. They include facilities to prevent spoofing based on shared key exchange. Once an SM is authenticated and registered to a Tasking Agent, the Tasking Agent can then discover the capabilities of the SM and its associated SD's. The Tasking Agent may then define and make sensing requests to the SM, which include a designation of the Data Client(s) to which the data is to be sent. The SM will notify the Tasking Agent when measurements are successfully completed (or not) and available at the Data Client.

5.1.2.2 Resource Discovery

Resource Discovery is the process of informing the Tasking Agent of what capabilities that the SM has with regard to what types of measurements, what bands can be measured and associated measurement parameters that can be specified and controlled and over what locations.
This takes the form of a resource/capability message object and the current scan schedule per SD.

5.1.2.3 Scan Request

The Scan Request message from the Tasking Agent to the SM includes the parameters of the desired spectrum measurement to be made and any associated processing to be performed by either the SD or the SM. This scan request is wrapped in a scheduling task description, defining the time the scan is to be made, the repetition rate (if applicable), the locations, etc. When the scan parameters in their scheduling wrapper are received by the SM it will be validated as possible to be executed (i.e. the resources requested meet the SSMs schedule of resources available), and either acknowledged as being queue, or a refusal is returned to the Tasking Agent. If a scan schedule is updated for a particular SD, it is then replicated down to that SD.

5.1.3 SM to SD Interface

5.1.3.1 Authentication and Registration

These procedures define the association and authentication process for an SD and SM entity to connect and communicate. They include facilities to prevent spoofing based on shared key exchange. Once an SD is authenticated and registered to a SM, the SM can then discover the capabilities of the SD. An SM will have associated with it at least one SD. The SM may then assign sensing requests to the appropriate set of SDs in order to fulfil the sensing request of the Tasking Agent.

5.1.3.2 Status and Discovery

The Status and Discovery process serves two functions. The first is to inform the SM of what capabilities that the SD has with regard to what types of measurements, what bands can be measured and associated measurement facilities (such calibration, antenna control, mobility, storage, processing) that can be specified and controlled and over what locations. The SD will transmit a package describing its capabilities and available resources at time of authentication/discovery, and if there is any change in its configuration. The second function is to maintain association with the SM. It will transmit a heartbeat periodically to indicate it is still associated with the SM. If it is to disconnect, it will transmit a disassociation message (e.g. if it is rebooting or about to go into an offline mode).

5.1.3.3 Scan Request

The Scan Request message originating from the SM is sent to the appropriate SDs for execution as a scan schedule. It includes the parameters of the desired spectrum measurement to be made based on knowledge of the SD's capabilities. This request will include the time to make the measurement, the repetition rate (if applicable), the locations, etc. and the format of the measured data. In the case of a single, once-off scan, the schedule will indicate no repetition.

Message Parameters are captured in Table N.

5.1.4 Data Manager to Data Client Interface

5.1.4.1 Authentication and Registration

These procedures define the association and authentication process for a Data Client and DM entity to connect and communicate. They include facilities to prevent spoofing based on shared key exchange. Once a Data Client is authenticated and registered with a DM, the DM is then authorized to cause data to be delivered to the Data Client based on the privileges of the Data Client and the DM. The Data Clients can be grouped into Data Client Groups, where a transmission of data from the DM is delivered to multiple Data Clients.

5.1.4.2 Data Manager

These procedures define and enable the storage of data from the DM to the Data Client. The successful reception of this data initiates a notification of the initiating Tasking Agent that requested that data.

5.2 SCOS Messaging

The communication between each of the entities defined above can be grouped and defined within the Interface Categories shown in Figure 9. Message Sequence and described below.

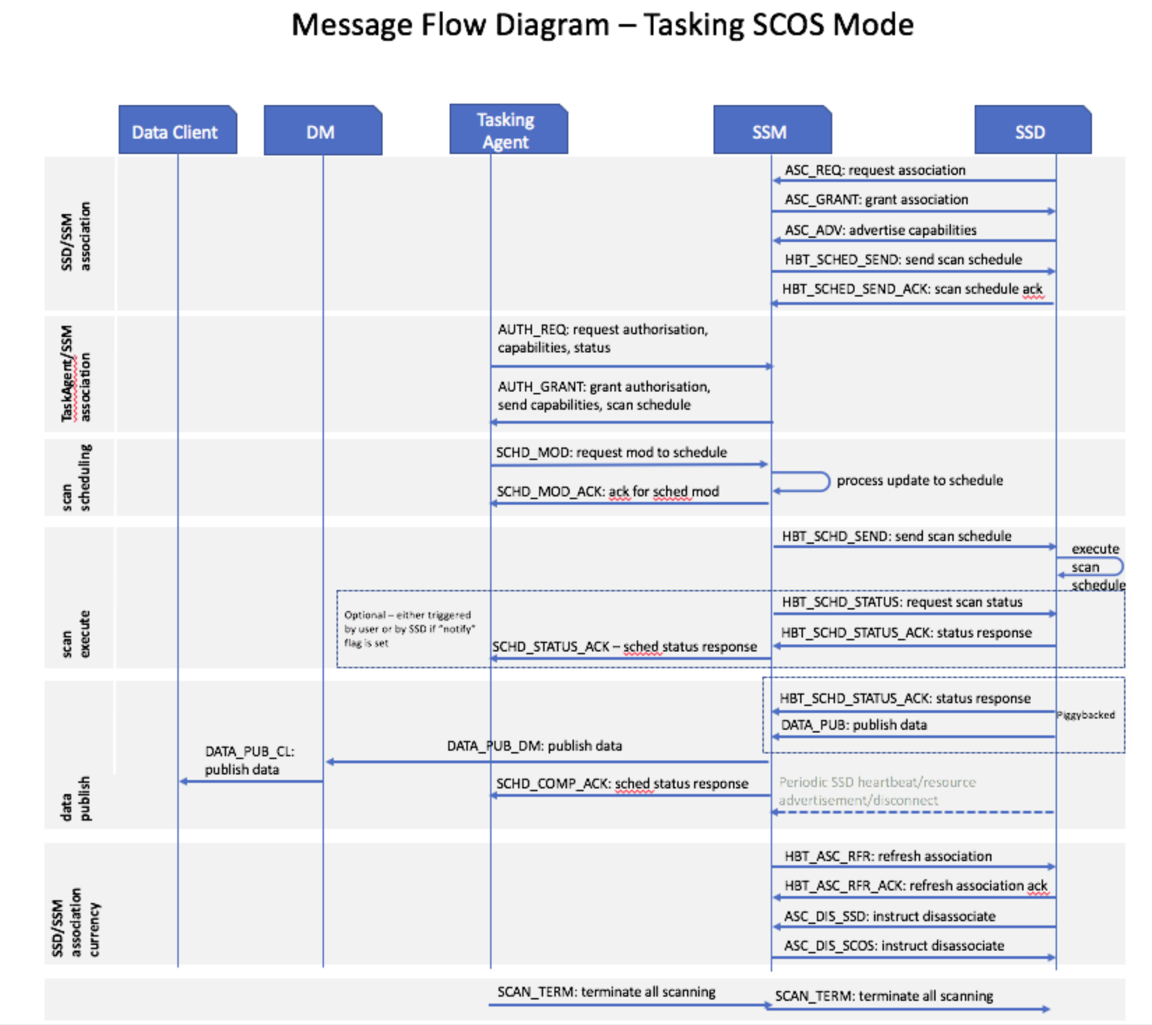

Message Flow Diagram – Tasking SCOS Mode
Data Client
DM
Tasking Agent
SSM
SSD
SSD/SSM association
ASC_REQ: request association
ASC_GRANT: grant association
ASC_ADV: advertise capabilities
HBT_SCHED_SEND: send scan schedule
HBT_SCHED_SEND_ACK: scan schedule ack
TaskAgent/SSM association
AUTH_REQ: request authorisation, capabilities, status
AUTH_GRANT: grant authorisation, send capabilities, scan schedule
scan scheduling
SCHD_MOD: request mod to schedule
process update to schedule
SCHD_MOD_ACK: ack for sched mod
scan execute
HBT_SCHD_SEND: send scan schedule
execute scan schedule
Optional – either triggered by user or by SSD if "notify" flag is set
HBT_SCHD_STATUS: request scan status
HBT_SCHD_STATUS_ACK: status response
SCHD_STATUS_ACK – sched status response
data publish
HBT_SCHD_STATUS_ACK: status response
Piggybacked
DATA_PUB: publish data
DATA_PUB_DM: publish data
DATA_PUB_CL: publish data
SCHD_COMP_ACK: sched status response
Periodic SSD heartbeat/resource advertisement/disconnect
SSD/SSM association currency
HBT_ASC_RFR: refresh association
HBT_ASC_RFR_ACK: refresh association ack
ASC_DIS_SSD: instruct disassociate
ASC_DIS_SCOS: instruct disassociate
SCAN_TERM: terminate all scanning
SCAN_TERM: terminate all scanning

Figure 13. SCOS Message Sequence

5.2.1 Message Encoding

SCOS messages are encoded in JavaScript Object Notation (JSON). JSON is a language-independent data-interchange format that is easy for humans to read and write. There are code and functions readily available in C, C++, C#, Java, JavaScript, MATLAB, Perl, and Python for parsing and generating JSON. It is a lightweight alternative to XML, commonly used to transmit data between server and browser application.

The first five fields are the same for all messages; they are:

1. Ver = Schema/data transfer version with the major.minor.revision syntax (string)
2. Type = Type of JSON message (string) {“Sys”, ”Loc”, or “Data”}
3. SensorID = Unique identifier of sensor (string of URL unreserved characters)
4. SensorKey = Authentication key given out by MSOD (integer)
5. t = Time [seconds since Jan 1, 1970 UTC] (long integer)

Each message begins with a header comprised of attribute-value pairs in ASCII characters. [#confirm]

If an attribute is not relevant to the sensor implementation, then the value is set to NaN or "NaN".

The following are specific formatting rules to be followed:

- All timestamps, i.e., t (defined above)and t1 (to be defined in Data message description) will be reported as seconds since 1/1/1970 midnight UTC in the UTC time zone.
- String values must only contain URL unreserved characters (i.e., uppercase and lowercase letters, decimal digits, hyphen, period, underscore, and tilde), and
- Field names cannot start with an underscore because that convention is reserved for internal implementation-specific uses.

5.2.2 Message Transport protocols

The SCOS standard is transport-agnostic. The standard defines requirements for the transport protocol. The implementers may choose appropriate transport protocol that meets these requirements and suits to the use-case. In Annex <#>, we illustrate how certain transport protocols could be applied toward implementing the SCOS interfaces.

5.3 Primitives

Each message (in general) will begin with a header as shown in the following table.

Parameter	R/O/C	Description
NAME: version DATA TYPE: string	Required	IEEE 802.22.3 SCOS protocol version. The maximum length is 64 octets.
NAME: scosmode DATA TYPE: Integer	Required	The mode for SCOS system.
NAME: scosmethod DATA TYPE: String	Required	The SCOS method in the context of the communication. The scaos methods are listed in the message descriptions. The maximum length is 64 octets.
NAME: msgtype DATA TYPE: Integer	Required	The valid message types are request and response. (1=Request, 2=Response 3=Notification 4=AdminCmd 5=AdminCmdResponse)
NAME: timestamp DATA TYPE: Time	Required	Timestamp associated with the communication..

The following are specific formatting rules to be followed to avoid problems when messages are ingested into MSOD: (1) All timestamps, i.e., t (defined above)and t1 (to be defined in Data message description) will be reported as seconds since 1/1/1970 midnight UTC in the UTC time zone. (2) String values must only contain URL unreserved characters (i.e., uppercase and lowercase letters, decimal digits, hyphen, period, underscore, and tilde), and (3) Field names cannot start with an underscore because that convention is reserved for MSOD internal use. <#!>

The data fields in the JSON message descriptions below are required fields. If an attribute is not relevant to *the sensor implementation, then the value is set to NaN or "NaN".*

5.3.1 SD<>SM Messages

scos_method_name	JSON Array Name of Request Message	JSON Array Name of Response Message
"sd_associate"	*sdAssociateRequest*	*sdAssociateResponse*
"sd_capability"	*sdCapabilityRequest*	*sdCapabilityResponse*
"sd_scan"	*sdScanRequest*	*sdScanResponse*
"sd_heartbeat"	*sdHeartbeatRequest*	*sdHeartbeatResponse*
"sd_disassociate"	*sdDisassociateRequest*	*sdDisassociateResponse*

5.3.1.1 SD-SM Association Message Exchange

Table <#Num> describes the sdAssociateRequest JSON object.

Parameter	R/O/C	Description
NAME: SDName DATA TYPE: string	Required	The name of the sensing device registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: SDMode DATA TYPE: Integer	Required	The mode in which sensing device operates. (1=online, 2=offline)
NAME: SDType DATA TYPE: Integer	Required	The type of the sensing device. (1=SDFull, 2=SDProxy)
NAME: SDID DATA TYPE: string	Conditional	The unique ID assigned to the sensing device. If ID is not pre-assigned, this is left empty. The maximum length is 64 octets.
NAME: SDCertFile DATA TYPE: String	Conditional	The path of the SD certificate file. The maximum length of the ID string is 256 octets.
NAME: SDKeyFile DATA TYPE: String	Conditional	The name of the SD certificate file. The maximum length of the ID string is 256 octets.
NAME: SDCAFile DATA TYPE: String	Conditional	The name of the trusted certificate authority. The maximum length of the ID string is 256 octets.

Table <#Num> describes the sdAssociateResponse JSON object.

Parameter	R/O/C	Description
NAME: SDName DATA TYPE: string	Required	The name of the sensing device registered with SCOS operator. The maximum length is 64 octets.
NAME: response DATA TYPE: string	Required	The response code for association.

NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: heartbeatInterval DATA TYPE: Integer	Required	Heartbeat interval in seconds.

5.3.1.2 SD-SM Capability Information Exchange

Table <#Num> describes the sdCapabilityRequest JSON object sent by the SM to SD.

Parameter	R/O/C	Description
NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: sendBaseCapability DATA TYPE: boolean	Conditional	True or False. If false, base capability information is not required.
NAME: freqIntervals DATA TYPE: Array of freqInterval	Conditional	Array of freqInterval objects. Each freqInterval object denotes a frequency range as defined in Table <#Num>.
NAME: timeIntervals DATA TYPE: Array of timeRange	Conditional	Array of timeInterval objects. Each timeInterval object denotes a time range as defined in Table <#Num>
NAME: scanPeriodicity DATA TYPE: Integer	Conditional	Supported scanPeriodicity interval. The periodicity interval is expressed in number of seconds.

Parameter	R/O/C	Description
NAME: lowFreq DATA TYPE: Integer	Required	The low frequency of a frequency interval.
NAME: highFreq DATA TYPE: Integer	Required	The high frequency of a frequency interval.

Parameter	R/O/C	Description
NAME: startTime DATA TYPE: Time	Required	The start of a time interval.
NAME: endTime DATA TYPE: Time	Required	The end of a time interval.

Parameter	R/O/C	Description
NAME: time DATA TYPE: String	Required	UTC time expressed in the format YYYY-MM-DDThh:mm:ssZ as defined by [#Ref]

Table <#Num> describes the sdCapabilityResponse JSON object sent by the SD to SM.

Parameter	R/O/C	Description
NAME: SDID DATA TYPE: string	Required	The name of the SD registered with SCOS operator. The maximum length is 64 octets.
NAME: SDCapabilityInfo DATA TYPE: sdCapabilityInfo	Conditional	Object describing SD capability (class B SD metadata) as described in Table <#%#>.

NAME: freqIntervals DATA TYPE: Array of freqInterval	Conditional	Array of freqInterval objects. Each freqInterval object denotes a frequency range as defined in Table <#Num>.
NAME: timeIntervals DATA TYPE: Array of timeRange	Conditional	Array of timeInterval objects. Each timeInterval object denotes a time range as defined in Table <#Num>
NAME: scanPeriodicity DATA TYPE: Integer	Conditional	Supported scanPeriodicity interval. The periodicity interval is expressed in number of seconds.

5.3.1.3 SD-SM Scan Message Exchange

Table <#Num> describes the sdScanRequest JSON object from SM to SD.

Parameter	R/O/C	Description
NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: TaskID DATA TYPE: String	Required	Unique ID for the Spectrum Scan. The maximum length of the ID string is 64 octets.
NAME: freqIntervals DATA TYPE: Array of freqInterval	Conditional	Array of freqInterval objects. Each freqInterval object denotes a frequency range as defined in Table <#Num>.
NAME: scanResolution DATA TYPE: Integer	Conditional	The suggested frequency resolution for the scan.
NAME: TaskDuration DATA TYPE: number	Required	Duration of scan in milliseconds.
NAME: TaskStartTime DATA TYPE: Time	Required	The start time for the task.
NAME: TaskRepeatInterval DATA TYPE: Number	Optional	The interval in seconds after which the task needs to be repeated.
NAME: TaskRepeatCount DATA TYPE: Number	Optional	The number of times the task needs to be repeated.
NAME: TaskEndTime DATA TYPE: Time	Optional	The end time for the task.

Table <#Num> describes the sdScanResponse JSON object from SD to SM.

Parameter	R/O/C	Description
NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: TaskID DATA TYPE: String	Required	Unique ID for the Spectrum Scan. The maximum length of the ID string is 64 octets.
NAME: scanStatus DATA TYPE: Array of Integer	Required	Array provides scan output status code for each of the freqIntervals. The status code is one of the response codes from Table. The freqIntervals should match with the freqIntervals from the request message.
NAME: timestamp DATA TYPE: Time	Required	Timestamp with the associated scanning output.
NAME: scanData	Required	Array of scanData objects. Each object represents

DATA TYPE: Array of scanData objects		SD measurements for the freqInterval. The scanData is defined in Table <#>
NAME: envInfo DATA TYPE: environMetadata	Required	The environmental data including GPS, temperature, and humidity as described in Table <#>

Parameter	R/O/C	Description
NAME: dataFormat DATA TYPE: Integer	Required	The format of the output data as specified in Table <#>.
NAME: sizeData DATA TYPE: Integer	Required	The number of measurements.
NAME: measData DATA TYPE: Array of Complex	Required	The complex measurement values. The size of the array is defined by sizeData.

5.3.1.4 SD-SM Heartbeat Message Exchange

Table <#Num> describes the sdHeartbeatRequest JSON object from SM to SD.

Parameter	R/O/C	Description
NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: calibrate DATA TYPE: boolean	Conditional	If true, SD is required to perform calibration.
NAME: calibrateTime DATA TYPE: Time	Conditional	If calibrate true, calibrateTime denotes the time for performing calibration.

Table <#Num> describes the sdHeartbeatResponse JSON object from SD to SM.

Parameter	R/O/C	Description
NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: calibrateStatus DATA TYPE: Integer	Conditional	The status code for the scheduled calibration.
NAME: envInfo DATA TYPE: envMetadata	Required	The type of TA. The maximum length is 64 octets.
NAME: healthInfo DATA TYPE: healthMetadata	Required	The SD health metadata as described in Table <#>.

Parameter	R/O/C	Description
NAME: batteryLevel DATA TYPE: Integer	Required	The battery level in percentage rounded to closest integer.

5.3.1.5 SD-SM Disassociation Message Exchange

Table <#Num> describes the sdDisassociateRequest JSON object from SD to SM.

Parameter	R/O/C	Description

NAME: SDID DATA TYPE: string	Required	The ID assigned to SD by the SCOS operator. The maximum length is 64 octets.
NAME: SDName DATA TYPE: string	Required	The name of the sensing device registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.

Table <#Num> describes the sdDisassociateResponse JSON object from SM to SD.

Parameter	R/O/C	Description
NAME: SDName DATA TYPE: string	Required	The name of the SD registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: status DATA TYPE: string	Required	The response code for dissociation request.
NAME: oldSDID DATA TYPE: string	Required	The SD ID that has been dissociated The maximum length is 64 octets.

5.3.2 TA<>SM Messages

Following table describes the SCOS Tasking API methods

scos_method_name	JSON Array Name of Request Message	JSON Array Name of Response Message
“ta_associate”	*taAssociateRequest*	*taAssociateResponse*
“ta_resource_discovery”	*taResourceDiscoveryRequest*	*taResourceDiscoveryResponse*
“ta_schedule_scan”	*taScheduleScanRequest*	*taScheduleScanResponse*
“ta_scan_status”	*taScanStatusRequest*	*taScanStatusResponse*
“ta_scan_notify”	*taScanNotification*	*taScanNotificationResponse*
“ta_dissociate”	*taDissociateRequest*	*taDissociateResponse*

5.3.2.1 TA-SM Association Message Exchange

Table <#Num> describes the taAssociateRequest JSON object from TA to SM.

Parameter	R/O/C	Description
NAME: TAName DATA TYPE: string	Required	The name of the tasking agent registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: SMName DATA TYPE: string	Required	The name of the sensing manager to associate with.

		The maximum length is 64 octets.
NAME: TAType DATA TYPE: string	Required	The type of TA. Valid values include {“TATypeA”, “TATypeB”, “TATypeC”}
NAME: TAID DATA TYPE: string	Optional	The unique ID assigned to the tasking agent. The maximum length is 64 octets.
NAME: TACertFile DATA TYPE: String	Optional	The path of the TA certificate file. The maximum length of the ID string is 256 octets.
NAME: TAKeyFile DATA TYPE: String	Optional	The name of the TA certificate file. The maximum length of the ID string is 256 octets.
NAME: TACAFile DATA TYPE: String	Optional	The name of the trusted certificate authority. The maximum length of the ID string is 256 octets.
NAME: admPreferences DATA TYPE: AdmPreferences	Optional	A TA can optionally specify certain preferences related to how scanning task administration is performed.

Table <#Num> describes the AdmPreferences object

Parameter	R/O/C	Description
NAME: useNewTaskforRestart DATA TYPE: boolean	Optional	TA chooses to have new TaskID for a restarted task. In the case of restarting a sensing task, SM provides old TaskID as well in order to associate the two. More details in the Section 7 administration.
NAME: useNewTaskforMigration DATA TYPE: boolean	Optional	TA chooses to have new TaskID for a sensing task with change in SD resource. In the case of migrating a sensing task, SM provides old TaskID as well in order to associate the two. More details in the Section 7 administration.

Table <#Num> describes the taAssociateResponse JSON object from SM to TA

Parameter	R/O/C	Description
NAME: TAName DATA TYPE: string	Required	The name of the tasking agent registered with SCOS operator. The maximum length is 64 octets.
NAME: SMName DATA TYPE: string	Required	The name of the sensing manager to associate the TA with. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: status DATA TYPE: string	Required	The response code for the TA association request. The maximum length is 64 octets.
NAME: TAID DATA TYPE: string	Required	The unique ID assigned to the tasking agent. The maximum length is 64 octets.
NAME: SMID DATA TYPE: string	Required	The unique ID of the sensing manager. The maximum length is 64 octets.

5.3.2.2 TA-SM Resource Discovery Message Exchange

Table <#Num> describes the taResourceDiscoveryRequest JSON object from TA to SM.

Parameter	R/O/C	Description
NAME: TAID DATA TYPE: string	Required	The name of the tasking agent registered with SCOS operator. The maximum length is 64 octets.
NAME: perSDInfoRequest DATA TYPE: Boolean	Required	If True, per SD resource/capability information is requested.
NAME: freqIntervals DATA TYPE: Array of freqInterval	Optional	Array of freqInterval objects. Each freqInterval object denotes a frequency range as defined in Table <#Num>.
NAME: scanDataFormat DATA TYPE: Array of timeInterval	Optional	The format of the scan data as described in Table <#>. The maximum length is 64 octets.
NAME: scanResolution DATA TYPE: Integer	Optional	The minimum desired scan resolution.
NAME: locations DATA TYPE: Array of Location objects	Optional	Array of Location objects for the specified scan frequencies. Each Location object denotes desired coordinates as defined in Table <#Num>.
NAME: locationAccuracy DATA TYPE: Integer	Optional	Desired accuracy for location in terms of maximum distance in meters from the specified coordinate.

Parameter	R/O/C	Description
NAME: Latitude DATA TYPE: string	Required	Latitude is expressed in format DD°MM'SS'' N/S The maximum length is 64 octets.
NAME: Longitude DATA TYPE: string	Required	Longitude is expressed in format DD°MM'SS'' W/E The maximum length is 64 octets.

Table <#Num> describes the taResourceDiscoveryResponse JSON object from SM to TA.

Parameter	R/O/C	Description
NAME: SMID DATA TYPE: string	Required	The unique ID of the sensing manager. The maximum length is 64 octets.
NAME: TAID DATA TYPE: string	Required	The name of the tasking agent registered with SCOS operator. The maximum length is 64 octets.
NAME: sdResourceInfo DATA TYPE: Array of sdCapabilityInfo objectd	Conditional	If perSDInfoRequest is true in the request, array of sdCapabilityInfo (defined in Table #%#) objects is included.
NAME: statusFreqIntervals DATA TYPE: Array of Integer	Conditional	Status codes for each of the freqIntervals from the request message that meet the scanDataformat and scanResolution.
NAME: AccuracyLocation DATA TYPE: Array of Integer	Conditional	Accuracy for each of the Locations from the request message in terms of distance (measured in meter).

5.3.2.3 TA-SM Scan Scheduling Message Exchange

Table <#Num> describes the taScheduleScanRequest JSON object from TA to SM.

Parameter	R/O/C	Description
NAME: TAID DATA TYPE: string	Required	The name of the tasking agent registered with SCOS operator. The maximum length is 64 octets.
NAME: freqIntervals DATA TYPE: Array of freqInterval	Required	Array of freqInterval objects. Each freqInterval object denotes a frequency range as defined in Table <#Num>.
NAME: scanDataFormat DATA TYPE: Array of timeInterval	Required	The format of the scan data as described in Table <#>. The maximum length is 64 octets.
NAME: scanResolution DATA TYPE: Integer	Required	The minimum desired scan resolution.
NAME: locations DATA TYPE: Array of Location objects	Required	Array of Location objects. Each Location object denotes desired coordinates as defined in Table <#Num>.
NAME: locationAccuracy DATA TYPE: Integer	Required	Desired accuracy for location in terms of maximum distance in meters from the specified coordinate.

Table <#Num> describes the taScheduleScanResponse JSON object from SM to TA

Parameter	R/O/C	Description
NAME: SMID DATA TYPE: string	Required	The unique ID of the sensing manager. The maximum length is 64 octets.
NAME: TAID DATA TYPE: string	Required	The name of the tasking agent registered with SCOS operator. The maximum length is 64 octets.
NAME: TAScanID DATA TYPE: string	Required	The unique ID assigned for the scan scheduled for the TA. The maximum length is 64 octets.
NAME: status DATA TYPE: Array of Integer	Required	Status codes for each of the scan frequency ranges that support desired scan parameters and the desired locationAccuracy.

5.3.2.4 TA-SM Scan Status Inquiry Message Exchange

Table <#Num> describes the sdScanStatusRequest JSON object from TA to SM.

Parameter	R/O/C	Description
NAME: TAID DATA TYPE: string	Required	The name of the tasking agent registered with SCOS operator. The maximum length is 64 octets.
NAME: SMID DATA TYPE: string	Required	The unique ID of the sensing manager. The maximum length is 64 octets.

NAME: TAScanID DATA TYPE: string	Required	The unique ID assigned for the scan scheduled for the TA. The maximum length is 64 octets.

Table <#Num> describes the sdScanStatusResponse JSON object from SM to TA

Parameter	R/O/C	Description
NAME: TAID DATA TYPE: string	Required	The name of the tasking agent registered with SCOS operator. The maximum length is 64 octets.
NAME: SMID DATA TYPE: string	Required	The unique ID of the sensing manager. The maximum length is 64 octets.
NAME: TAScanID DATA TYPE: string	Required	The unique ID assigned for the scan scheduled for the TA. The maximum length is 64 octets.
NAME: status DATA TYPE: Array of Integer	Required	Status codes for each of the scan frequency ranges that support desired scan parameters and the desired locationAccuracy.

5.3.2.5 TA-SM Scan Notification Message Exchange

Upon completion of the scan or upon error event, SM notifies the TA of the status for the scan.

Table <#Num> describes the sdScanNotifyRequest JSON object from TA to SM.

Parameter	R/O/C	Description
NAME: TAID DATA TYPE: string	Required	The name of the tasking agent registered with SCOS operator. The maximum length is 64 octets.
NAME: SMID DATA TYPE: string	Required	The unique ID of the sensing manager. The maximum length is 64 octets.
NAME: TAScanID DATA TYPE: string	Required	The unique ID assigned for the scan scheduled for the TA. The maximum length is 64 octets.
NAME: status DATA TYPE: Array of Integer	Required	Status codes for each of the scan frequency ranges that support desired scan parameters and the desired locationAccuracy.

5.3.2.6 TA-SM Dissociation Message Exchange

Table <#Num> describes the taDissociateRequest JSON object from TA to SM.

Parameter	R/O/C	Description
NAME: TAID DATA TYPE: string	Required	The ID assigned to TA by the SCOS operator. The maximum length is 64 octets.
NAME: TAName DATA TYPE: string	Required	The name of the TA registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.

Table <#Num> describes the sdDissociateResponse JSON object from SM to TA

Parameter	R/O/C	Description
NAME: TAName DATA TYPE: string	Required	The name of the tasking agent registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: status DATA TYPE: string	Required	The response code for dissociation request.
NAME: oldTAID DATA TYPE: string	Required	The TA ID that has been dissociated The maximum length is 64 octets.

6.3.3 SD<>DM Messages

scos_method_name	JSON Array Name of Request Message	JSON Array Name of Response Message
“sd_dm_associate”	*sdAssociateRequest*	*sdAssociateResponse*
“sd_dm_publish”	*sdPublishRequest*	*sdPublishResponse*
“sd_dm_heartbeat”	*sdHeartbeatRequest*	*sdHeartbeatResponse*
“sd_dm_disassociate ”	*sdDisassociateRequest*	*sdDisassociateResponse*

6.3.3.1 SD-DM Association Message Exchange

Table <#Num> describes the sdAssociateRequest JSON object from SD to DM.

6.3.3.2 SD-DM Publish Message

Table <#Num> describes the sdPublishRequest JSON object from SD to DM.

Parameter	R/O/C	Description
NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.

NAME: TaskID DATA TYPE: String	Required	Unique ID for the Spectrum Scan. The maximum length of the ID string is 64 octets.
NAME: scanStatus DATA TYPE: Array of Integer	Required	Array provides scan output status code for each of the freqIntervals. The status code is one of the response codes from Table. The freqIntervals should match with the freqIntervals from the request message.
NAME: timestamp DATA TYPE: Time	Required	Timestamp with the associated scanning output.
NAME: scanData DATA TYPE: Array of scanData objects	Required	Array of scanData objects. Each object represents SD measurements for the freqInterval. The scanData is defined in Table <#>
NAME: envInfo DATA TYPE: environMetadata	Required	The environmental data including GPS, temperature, and humidity as described in Table <#>

Parameter	R/O/C	Description
NAME: dataFormat DATA TYPE: Integer	Required	The format of the output data as specified in Table <#>.
NAME: sizeData DATA TYPE: Integer	Required	The number of measurements.
NAME: measData DATA TYPE: Array of Complex	Required	The complex measurement values. The size of the array is defined by sizeData.

Table <#Num> describes the sdPublishResponse JSON object from SD to DM.

Parameter	R/O/C	Description
NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.

NAME: TaskID DATA TYPE: String	Required	Unique ID for the Spectrum Scan. The maximum length of the ID string is 64 octets.
NAME: status DATA TYPE: Array of Integer	Required	Each entry shows status for the publish request of the scanning data for each of the freqIntervals.
NAME: timestamp DATA TYPE: Time	Required	Timestamp for the associated scanning data that DM is acknowledging.

6.3.3.3 SD-DM Heartbeat Message

Parameter	R/O/C	Description
NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: Cmd DATA TYPE: Time	Optional	If nonzero, SD is given a command. Currently, two command codes are defined. If cmd is 1, DM is asking SD to list all topics for which the SD is publishing. If cmd is 2, DM is asking to stop publishing for a topic or all topics as suggested by next object in the message.
NAME: topic DATA TYPE: string	Conditional	If cmd is 2, this field denotes the topic for which DM is asking SD to stop publishing. If no topic is specified, DM is asking to stop publishing for all the topics.

Parameter	R/O/C	Description
NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: activeTopics DATA TYPE: Array of string	Required	Each entry in the list describes an active topic. The maximum length is 64 octets.

6.3.3.4 SD-DM Dissociation Message Exchange

Table <#Num> describes the sdDissociateRequest JSON object from SD to DM.

6.3.4 DC<>DM Messages

scos_method_name	JSON Array Name of Request Message	JSON Array Name of Response Message
“dc_associate”	*dcAssociateRequest*	*dcAssociateResponse*
“dc_subscribe”	*dcSubscribeRequest*	*dcSubscribeResponse*
“dc_topicdata”	*dcTopicData*	*dcTopicDataResponse*
“dc_unsubscribe”	*dcUnSubscribeRequest*	*dcUnSubscribeResponse*
“dc_heartbeat”	*dcHeartbeatRequest*	*dcHeartbeatResponse*
“dc_disassociate”	*dcDisassociateRequest*	*dcDisassociateResponse*

6.3.4.1 DC-DM Association Message Exchange

Table <#Num> describes the dcAssociateRequest JSON object from DC to DM.

Parameter	R/O/C	Description
NAME: DCName DATA TYPE: string	Required	The name of the data-client registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: DMName DATA TYPE: string	Required	The name of the data manager to associate with. The maximum length is 64 octets.

NAME: DCID DATA TYPE: string	Optional	The unique ID assigned to the data-client. The maximum length is 64 octets.
NAME: DCCertFile DATA TYPE: String	Optional	The path of the TA certificate file. The maximum length of the ID string is 256 octets.
NAME: DCKeyFile DATA TYPE: String	Optional	The name of the DC certificate file. The maximum length of the ID string is 256 octets.
NAME: DCCAFile DATA TYPE: String	Optional	The name of the trusted certificate authority. The maximum length of the ID string is 256 octets.

Table <#Num> describes the taAssociateResponse JSON object from SM to TA

Parameter	R/O/C	Description
NAME: DCName DATA TYPE: string	Required	The name of the dataclient registered with SCOS operator. The maximum length is 64 octets.
NAME: DMName DATA TYPE: string	Required	The name of the data manager to associate the DC with. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: status DATA TYPE: string	Required	The response code for the TA association request. The maximum length is 64 octets.
NAME: DCID DATA TYPE: string	Required	The unique ID assigned to the tasking agent. The maximum length is 64 octets.
NAME: DMID DATA TYPE: string	Required	The unique ID of the data manager. The maximum length is 64 octets.

6.3.4.2 DC-DM Subscribe Message Exchange

Table <#Num> describes the dcSubscribeRequest JSON object from DC to DM.

Parameter	R/O/C	Description
NAME: DCID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: TopicID DATA TYPE: String	Required	Unique TopicID for the spectrum sensing data to associate the DC with. The maximum length of the ID string is 128 octets.
NAME: TAID DATA TYPE: String	Required	ID of tasking agent associated with the scan. The maximum length of the ID string is 64 octets.

Table <#Num> describes the dcSubscribeResponse JSON object from DM to DC.

Parameter	R/O/C	Description
NAME: DCID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: TopicID DATA TYPE: String	Required	Unique TopicID for the spectrum sensing data to associate the DC with. The maximum length of the ID string is 128 octets.
NAME: status DATA TYPE: Integer	Required	Response code to subscribe request.

6.3.4.3 DC-DM TopicData Message Exchange

Table <#Num> describes the dcTopicData JSON object from DC to DM.

Parameter	R/O/C	Description

NAME: SDID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: TaskID DATA TYPE: String	Required	Unique ID for the Spectrum Scan. The maximum length of the ID string is 64 octets.
NAME: scanStatus DATA TYPE: Array of Integer	Required	Array provides scan output status code for each of the freqIntervals. The status code is one of the response codes from Table. The freqIntervals should match with the freqIntervals from the request message.
NAME: timestamp DATA TYPE: Time	Required	Timestamp with the associated scanning output.
NAME: scanData DATA TYPE: Array of scanData objects	Required	Array of scanData objects. Each object represents SD measurements for the freqInterval. The scanData is defined in Table <#>
NAME: envInfo DATA TYPE: environMetadata	Required	The environmental data including GPS, temperature, and humidity as described in Table <#>

Parameter	R/O/C	Description
NAME: dataFormat DATA TYPE: Integer	Required	The format of the output data as specified in Table <#>.
NAME: sizeData DATA TYPE: Integer	Required	The number of measurements.
NAME: measData DATA TYPE: Array of Complex	Required	The complex measurement values. The size of the array is defined by sizeData.

Table <#Num> describes the dcTopicDataResponse JSON object from DM to DC.

Parameter	R/O/C	Description

NAME: DCID DATA TYPE: string	Required	The unique ID assigned to the Data Client. The maximum length is 64 octets.
NAME: TopicID DATA TYPE: String	Required	Unique ID for the Topic. The maximum length of the ID string is 128 octets.
NAME: status DATA TYPE: Array of Integer	Required	Each entry shows status for the TopicData request of the scanning data for each of the freqIntervals.
NAME: timestamp DATA TYPE: Time	Required	Timestamp for the associated scanning data that DC is acknowledging.

6.3.4.4 DC-DM Unsubscribe Message Exchange

Table <#Num> describes the dcUnSubscribeRequest JSON object from DC to DM.

Parameter	R/O/C	Description
NAME: DCID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: TopicID DATA TYPE: String	Required	Unique TopicID for the spectrum sensing data the DC wants to unsubscribe. The maximum length of the ID string is 128 octets.
NAME: TAID DATA TYPE: String	Required	ID of tasking agent associated with the scan. The maximum length of the ID string is 64 octets.

Table <#Num> describes the dcUnSubscribeResponse JSON object from DM to DC.

Parameter	R/O/C	Description
NAME: DCID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.

NAME: TopicID DATA TYPE: String	Required	Unique TopicID for the spectrum sensing data the DC wants to unsubscribe. The maximum length of the ID string is 128 octets.
NAME: status DATA TYPE: Integer	Required	Response code to unsubscribe request.

6.3.4.5 DC-DM Heartbeat Message

Table <#Num> describes the dcHeartbeatRequest JSON object from DM to DC.

Parameter	R/O/C	Description
NAME: DCID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: Info DATA TYPE: Integer	Optional	If DM intends to notify DC certain information related to specific topic or DM/connectivity specific health information. Information codes: 0-15: DM/connectivity specific information >15: Topic specific information
NAME: topic DATA TYPE: string	Conditional	If information code > 15, this field denotes the topic for which DM is providing additional information.

Table <#Num> describes the dcHeartbeatResponse JSON object from DC to DM.

Parameter	R/O/C	Description
NAME: DCID DATA TYPE: string	Required	The unique ID assigned to the data-client. The maximum length is 64 octets.
NAME: topicsNeedAttention DATA TYPE: Array of string	Optional	Each entry in the list describes an active topic that needs attention. The maximum length of each topic entry is 128 octets.

6.3.4.6 DC-DM Dissociation Message Exchange

Table <#Num> describes the dcDissociateRequest JSON object from DC to DM.

Parameter	R/O/C	Description
NAME: DCID DATA TYPE: string	Required	The ID assigned to DC by the SCOS operator. The maximum length is 64 octets.
NAME: DCName DATA TYPE: string	Required	The name of the DC registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.

Table <#Num> describes the dcDissociateResponse JSON object from DM to DC

Parameter	R/O/C	Description
NAME: TAName DATA TYPE: string	Required	The name of the DC registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: status DATA TYPE: string	Required	The response code for dissociation request.
NAME: oldDCID DATA TYPE: string	Required	The DC ID that has been dissociated The maximum length is 64 octets.

6.3.5 SM<>DM Messages

scos_method_name	JSON Array Name of Request Message	JSON Array Name of Response Message
“sm_dm_associate”	*smAssociateRequest*	*smAssociateResponse*

“sm_task_coordinate”	*smTaskCoordinationRequest*	*smTaskCoordinationResponse*
“sm_task_moderation”	*smTaskModerationRequest*	*smTaskModerationResponse*
“sm_dm_heartbeat”	*smHeartbeatRequest*	*smHeartbeatResponse*
“sm_dm_disssociate”	*smDisassociateRequest*	*smDisassociateResponse*

6.3.5.1 SM-DM Association Message Exchange

Table <#Num> describes the dcAssociateRequest JSON object from DC to DM.

Parameter	R/O/C	Description
NAME: SMName DATA TYPE: string	Required	The name of the sensing manager registered with SCOS operator. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: DMName DATA TYPE: string	Required	The name of the data manager to associate with. The maximum length is 64 octets.
NAME: SMID DATA TYPE: string	Optional	The unique ID assigned to the sensing manager. The maximum length is 64 octets.
NAME: SMCertFile DATA TYPE: String	Optional	The path of the SM certificate file. The maximum length of the ID string is 256 octets.
NAME: SMKeyFile DATA TYPE: String	Optional	The name of the SM certificate file. The maximum length of the ID string is 256 octets.
NAME: SMCAFile DATA TYPE: String	Optional	The name of the trusted certificate authority. The maximum length of the ID string is 256 octets.

Table <#Num> describes the taAssociateResponse JSON object from SM to TA

Parameter	R/O/C	Description
NAME: SDName DATA TYPE: string	Required	The name of the sensing manager registered with SCOS operator. The maximum length is 64 octets.
NAME: DMName DATA TYPE: string	Required	The name of the data manager to associate the SD with. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: status DATA TYPE: string	Required	The response code for the SM-DM association request. The maximum length is 64 octets.
NAME: DMID DATA TYPE: string	Required	The unique ID of the data manager. The maximum length is 64 octets.
NAME: SMID DATA TYPE: string	Required	The unique ID of the sensing manager. The maximum length is 64 octets.

6.3.5.2 SM-DM Sensing Task Coordination Message Exchange

Table <#Num> describes the dmTaskCoordinationRequest JSON object from SM to DM.

Parameter	R/O/C	Description
NAME: SMID DATA TYPE: string	Required	The unique ID of the sensing manager. The maximum length is 64 octets.
NAME: TaskID DATA TYPE: String	Required	Unique TaskID for the spectrum sensing data to associate the DM with. The maximum length of the ID string is 128 octets.

NAME: DMID DATA TYPE: String	Required	ID of data manager associated with the scan. The maximum length of the ID string is 64 octets.

Table <#Num> describes the dTaskCoordinationResponse JSON object from DM to SM.

Parameter	R/O/C	Description
NAME: DMID DATA TYPE: string	Required	The unique ID assigned of the DM handling the sensing data distribution for the sensing task. The maximum length is 64 octets.
NAME: TaskID DATA TYPE: String	Required	Unique TaskID for the spectrum sensing task. The maximum length of the ID string is 128 octets.
NAME: status DATA TYPE: Integer	Required	Response code to subscribe request.

6.3.5.3 SM-DM TopicData Message Exchange

Table <#Num> describes the dmTaskModeration JSON object from DM to SM.

Parameter	R/O/C	Description
NAME: SMID DATA TYPE: string	Required	The unique ID of the sensing manager. The maximum length is 64 octets.
NAME: DMID DATA TYPE: String	Required	ID of data manager associated with the scan. The maximum length of the ID string is 64 octets.
NAME: TaskID DATA TYPE: String	Required	Unique TaskID for the spectrum sensing data to associate the DM with. The maximum length of the ID string is 128 octets.

NAME: ModCmd DATA TYPE: Integer	Required	Sensing task moderation command code. Following are the defined command codes. 0: Invalid cmd 1: Data validation errors 2: Data rate mismatch 3: Switch the sensing task to another DM 4-31: reserved commands >31: custom command codes

Table <#Num> describes the dTaskModerationResponse JSON object from SM to DM.

Parameter	R/O/C	Description
NAME: DMID DATA TYPE: string	Required	The unique ID assigned of the DM handling the sensing data distribution for the sensing task. The maximum length is 64 octets.
NAME: TaskID DATA TYPE: String	Required	Unique TaskID for the spectrum sensing task. The maximum length of the ID string is 128 octets.
NAME: status DATA TYPE: Integer	Required	Response code for the task moderation request.

6.3.5.4 DC-DM Unsubscribe Message Exchange

Table <#Num> describes the dcUnSubscribeRequest JSON object from DC to DM.

Parameter	R/O/C	Description
NAME: DCID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: TopicID DATA TYPE: String	Required	Unique TopicID for the spectrum sensing data the DC wants to unsubscribe. The maximum length of the ID string is 128 octets.
NAME: TAID DATA TYPE: String	Required	ID of tasking agent associated with the scan. The maximum length of the ID string is 64 octets.

Table <#Num> describes the dcUnSubscribeResponse JSON object from DM to DC.

Parameter	R/O/C	Description
NAME: DCID DATA TYPE: string	Required	The unique ID assigned to the sensing device. The maximum length is 64 octets.
NAME: TopicID DATA TYPE: String	Required	Unique TopicID for the spectrum sensing data the DC wants to unsubscribe. The maximum length of the ID string is 128 octets.
NAME: status DATA TYPE: Integer	Required	Response code to unsubscribe request.

6.3.5.5 SM-DM Heartbeat message

Table <#Num> describes the smHeartbeatRequest JSON object from DM to DC.

Parameter	R/O/C	Description
NAME: SMID DATA TYPE: string	Required	The unique ID of the sensing manager. The maximum length is 64 octets.
NAME: DMID DATA TYPE: string	Required	The unique ID of the data manager. The maximum length is 64 octets.
NAME: Info DATA TYPE: Integer	Optional	If DM intends to notify DC certain information related to specific topic or SM/connectivity specific health information. Information codes: 0-15: SM/connectivity specific information >15: Topic specific information
NAME: topic DATA TYPE: string	Conditional	If information code > 15, this field denotes the topic for which SM is providing additional information.

Table <#Num> describes the smHeartbeatResponse JSON object from DC to DM.

Parameter	R/O/C	Description
NAME: DMID DATA TYPE: string	Required	The unique ID assigned to the data-client. The maximum length is 64 octets.
NAME: topicsNeedAttention DATA TYPE: Array of string	Optional	Each entry in the list describes an active topic that needs attention. The maximum length of each topic entry is 128 octets.

6.3.5.6 SM-DM Dissociation Message Exchange

Table <#Num> describes the dmDissociateRequest JSON object from DC to DM.

Parameter	R/O/C	Description
NAME: SMID DATA TYPE: string	Required	The ID of the sensing manager. The maximum length is 64 octets.
NAME: SMName DATA TYPE: string	Required	The name of the sensing manager. The maximum length is 64 octets.
NAME: DMID DATA TYPE: string	Required	The ID of the data manager. The maximum length is 64 octets.
NAME: DMName DATA TYPE: string	Required	The name of the data manager. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.

Table <#Num> describes the dmDissociateResponse JSON object from DM to DC

Parameter	R/O/C	Description
NAME: SMName DATA TYPE: string	Required	The name of the DC registered with SCOS operator. The maximum length is 64 octets.

NAME: DMName DATA TYPE: string	Required	The name of the data manager. The maximum length is 64 octets.
NAME: SCOSOperator DATA TYPE: string	Required	The name of the SCOS operator. The maximum length is 64 octets.
NAME: status DATA TYPE: string	Required	The response code for dissociation request.

6. System Administration and Security

6.1 Administration

Administrative functions on the SDs and SSMs are largely assumed to be implementer-specific, and out of scope for the standard, but recommendations are included in the informative annex.

This interface shall have a secure mechanism to administer the system, allowing:

- performing of SD calibrations
- updating firmware of SDs
- changing configuration of SD, and making associated changes to the SD configuration file
- triggering SD reboots and other SD hardware maintenance functions.

Additionally, administrator shall have a mechanism allowing

- pause or delete particular scanning task and corresponding message-queue/topic.
- flush a message-queue and restart task with the existing/new taskID
- dissociate a TA and all scheduled/ongoing tasks from the TA
- dissociate a SD
- migrate all sensing schedules for an SD to other SD(s).

The administration interface must be a secure interface with key exchange, and these keys must not be the same as keys used for TASKING AGENT<>SM<>SD authentication.

The standard suggests using special administration mode (AdminCmd and AdminCmdResponse) in the message header. There may be multiple response messages for a single command indicating the execution status of the command such as scheduled, in-progress, and done. Following table identifies administration commands for various interfaces.

6.1.1 SD Administration

SM-SD Interface Summary *(Actual messages need to be detailed upon high level agreement over the content.)*

Administration Command	SD Action	Administration Response	Details
NAME: admCalibrate DATA TYPE: String	SD performs CALif no time specified else schedules CAL.	NAME: admCalibrateStatus DATA TYPE: String	The time may be optionally specified with this command. The status from the SD could be SD_CAL_IN_PROGRESS, SD_CAL_SCHEDULED, SD_CAL_DONE, or SD_CAL_ERR
NAME: admFwUpdate DATA TYPE: String	SD updates firmware	NAME: admFwUpdateStatus DATA TYPE: String	The SM provides the firmware path, name, and update time. The SD status could be SD_FW_UPDATE_SCHEDULED, SD_FW_UPDATE_IN_PROGRESS,

			SD_FW_UPDATE_DONE, or SD_FW_UPDATE_ERR
NAME: admConfig DATA TYPE: String	SD applies suggested configuration	NAME: admConfigStatus DATA TYPE: String	The SM provides the config path, filename, and update time. The SD status could be SD_CONF_UPDATE_SCHEDULED, SD_CONF_UPDATE_IN_PROGRESS, SD_CONF_UPDATE_DONE, or SD_CONF_UPDATE_ERR
NAME: admPowerCycle DATA TYPE: String	SD performs power cycle	NAME: admPowerCycleStatus DATA TYPE: String	The SM optionally provides time. The SD status could be SD_POWER_CYCLE_SCHEDULED, SD_POWER_CYCLE_DONE, or SD_POWER_CYCLE_ERR

In addition to these commands, it is possible to define custom commands for administration. The standard may include more common use-cases in its next revision.

6.1.2 Sensing Task Administration

6.1.2.1 Pause/Resume a sensing task

It may be essential to pause a sensing task due to administrative or technical issue. In this case, a message-exchange is initiated from the SM to notify corresponding TA and SDs. SM coordinates this action with DM and a message exchange is initiated from DM to DCs. The SCOS administrator may need to resume the sensing task after the issue is resolved. Table <#> identifies associated interactions on the SM<->SD, SM <->TA, SM<->DM, and DM<-> DC interfaces.

6.1.2.2 Restart a sensing task

It may be essential to restart a sensing task due to administrative or technical issue. Here, in certain use-cases, TAs may prefer that a scanTaskID generated. TAs may communicate this preference during association (useNewTaskforRestart). If TA prefers a new task ID, old task ID is also communicated for associating the two scanning tasks.

A message-exchange is initiated from the SM to notify corresponding TA and SDs. SM coordinates this action with DM and a message exchange is initiated from DM to DCs. Table <#> identifies associated interactions on the SM<->SD, SM <->TA, SM<->DM, and DM<-> DC interfaces.

6.1.2.3 Migrate all the sensing tasks associated with an SD

It may be essential to migrate all sensing tasks to other SCOS resources (due to technical or security issue) . SM may assign a new task ID depending on TAs preferences.

A message-exchange is initiated from the SM to notify corresponding TA and SDs. SM coordinates this action with DM and a message exchange is initiated from DM to DCs. Table <#> identifies associated interactions on the SM<->SD, SM <->TA, SM<->DM, and DM<-> DC interfaces.

6.1.2.4 Delete a sensing task

It may be essential to delete a sensing task (due to technical, resource, or security issue) .

A message-exchange is initiated from the SM to notify corresponding TA and SDs. SM coordinates this action with DM and a message exchange is initiated from DM to DCs. Table <#> identifies associated interactions on the SM<->SD, SM <->TA, SM<->DM, and DM<-> DC interfaces.

6.1.2.5 Delete all sensing tasks from a TA

It may be essential to delete a sensing task (due to technical, resource, or security issue) .

A message-exchange is initiated from the SM to notify corresponding TA and SDs. SM coordinates this action with DM and a message exchange is initiated from DM to DCs. Table <#> identifies associated interactions on the SM<->SD, SM <->TA, SM<->DM, and DM<-> DC interfaces.

6.1.2.6 Summary of the message interactions for sensing task administration

SM-TA Interface messages summary *(Actual messages need to be detailed upon high level agreement over the content.)*

SM-SD Interface messages summary *(Actual messages need to be detailed upon high level agreement over the content.)*

SM-DM Interface messages summary *(Actual messages need to be detailed upon high level agreement over the content.)*

DM-DC Interface messages summary *(Actual messages need to be detailed upon high level agreement over the content.)*

6.1.3 SCOS Platform Behavior Specification

In addition to explicit administration, SCOS operator could define behavior of SCOS platform using policy policy construct. Following are a few examples of specifying SCOS platform behavior

- Allowed data distribution modes - streaming and/or store-forward
- Protocol choice for each of the interfaces - An SCOS operator may prefer MQTT for DM-DC interface and another operator may choose RESTful HTTP interface.
- Security mode for each of the interfaces - An SCOS operator may choose to enforce secure transport for all the interfaces (SD-DM, SD-SM, DM-SM, TA-SM, and DM-DC)

6.2 Security Systems

6.2.1 Scope

The SCOS standard includes security measures toward the maintaining the integrity and confidentiality of the sensing tasks and sensing data. Also, SCOS standard includes measures for ensuring authenticity of the messages. The standard makes provision for the security features and these are highly recommended however, it is upto the SCOS administrator to enforce the security mechanisms on most interfaces.For SM-DM interface, the SCOS platform must include security mechanisms for maintaining the integrity and confidentiality of the sensing tasks and sensing data.

Another part of platform security is authorization. Administrators need to ensure that only authorized users can issue sensing tasks, only authorized users have access to the sensing data published by the platform, and only authorized users can issue certain privileged scans. The standard in the current draft provides an approach to implementing administrative policies. Currently, in this version of the standard, The approach is not a mandatory approach and SCOS administrators may choose to implement policy using an alternate approach.

6.2.1.1 Out of scope

Following security aspects are out of scope of the standard

- Physical security of the infrastructure (SDs, SM, DM)
- Availability of the sensing data - The sensing tasks would typically generate enormous amount of sensing data and the SCOS standard does not require the SCOS platform implementation to make sensing data available past it has been received by the data clients. If data clients happen to lose the data, sensing needs to be performed again.
- Redundancy model for SM and DM - SM and DM are key entities and administrators may include a redundancy model for SM and DM to improve SCOS platform availability. The redundancy model is out of scope of the standard,

6.2.2 Authentication, Confidentiality, and Integrity

The standard requires implementing following procedure on SM-DM interface. On the remaining interfaces, it is highly recommended.

- TLS mutual authentication shall be performed per [n.1]
- EnityA (For example,TA) communicates with EntityB (For example, SM). 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. The EntityA (For example, TA) initiating the TLS connection shall authenticate the EntityB (For example, SM), and the EntityB (For example, SM) shall authenticate the EntityA (For example, TA).
- During the TLS message exchange, the EntityA (For example, TA) shall authenticate EntityB (For example, SM) according to the procedures defined in [n.4]. Server certificate validation shall be performed according to the procedures in [n.5].
- A EntityA (For example, TA) which is unable to successfully authenticate an EntityB (For example, SM) shall abort the TLS connection establishment procedure. It is implementation specific when the EntityA (For example, TA) should re-attempt the TLS connection establishment procedure.
- During the TLS message exchange, the EntityA (For example, TA) provides its client certificate to the EntityB (For example, SM). The EntityB (For example, SM)shall perform client certificate validation according to the procedures in [n.5]. The EntityB (For example, SM) which is unable to successfully authenticate a EntityA (For example, TA) shall abort the TLS connection establishment procedure.

6.2.3 Authorization

The standard suggests implementing authorization using policy construct for SCOS control plane and data plane.

6.2.3.1 Control Plane Authorization

Control plane authorization is implemented at the SM. It includes the

- ability to enforce a regulatory policy to determine if the location, time, frequency specified in the requested scan are compliant with regulations
- ability to check if the user is authorized to issue the requested sensing task
- ability to define priority for all scans from a specific user
- ability to define priority for a certain set of SD resources

6.2.3.2 Data Plane Authorization

Data plane authorization is implemented at the DM. It includes the

- ability to check if the data client is authorized to subscribe to the requested sensing data.
- rules for how data is distributed for various data clients. Only privileged data clients may be able

to request store and forward interface.

- ability to define max storage space in case of store and forward mode

The SCOS security mechanisms and certain administration mechanisms (as described in Section 7.1.3) are implemented using a policy file which is securely installed by the SCOS operator.

Annex A Informative: Reference Applications

A.1 White Space device radio operation

Either the network operator or device operator using spectrum sensing to identify primary or other secondary users of particular channels. Spectrum sensing would either built into the radio devices or in standalone sensing units.

The standard allows a “CR Mode” of operation that would make it suitable for use in radio systems to complement Geolocation Databases (such as a WSDB).

A.2 National spectrum regulation

National radio regulators would use a system comprising spectrum sensing devices to feed into a national spectrum utilization database for assignment management and planning purposes, and generating historical records for compliance monitoring and enforcement.

Devices deployed in various scenarios:

- Fixed devices at key locations and high sites
- Mobile devices on vehicles that travel widely and can create a sample set of spectrum utilization through snapshots at time or location intervals
- Devices either at fixed locations or periodically moved to create location-based spectrum utilization datasets
- Nationally deployed in a swarm of a given device density to create real-time national spectrum utilization maps and for validation of Spectrum Geolocation databases.

When spectrum monitoring is used for automated spectrum usage enforcement, data from a spectrum monitoring system has a critical role in the six basic steps for spectrum enforcement:
1. detecting, 2. identifying and classifying, 3. locating, 4. reporting, 5. mitigating, 6. remediating interference.

It is important to note that each of these six steps may, in general, require a different data type to be collected and stored; ranging from amplitude only information to raw IQ samples. It is possible for the sensing network to process spectrum data at the edge and only report the result of the processing, where conceivably a sensor or set of sensors can identify and locate an emitter without sending the raw spectrum measurements.

For example, the sensing network might report and store only the location information along with the signal classification information. This standard has been made flexible enough to define and enable both the collection of the various data types, along with associated meta-data, as well as the reporting and collection of the results of data analysis performed at the edge.

Spectrum management systems work to accomplish agency missions in geographic area(s) with limited and often shrinking frequency assignments. Monitoring can support spectrum managers in being more efficient by providing real-time and historical information about the RF environment on-base and at-boundary. Further, monitoring information can be used to mitigate and protect government wireless assets from intentional and unintentional interference.

A.3 Research programmes

Scientists using sensitive radio frequency systems (e.g. radio-telescopes) struggle with RF interference. SCOS devices can let them identify RFI and the location of their sources.

A.4 Law enforcement and public order

Law enforcement and other authorities are increasingly dealing with problems stemming from radio-controlled or radio-connected systems.

Illegal drone use: These include people flying radio-controlled unmanned aerial vehicles (drones) in prohibited places. SCOS systems can be used to detect characteristic transmissions of drone operation in areas such as in the airfield flight traffic area.

Detecting jamming devices: A problem area for security staff and law enforcement is the use of radio jammers to interfere with remote control devices like vehicle keyless entry systems or radio links for alarm systems. SCOS devices can be used to identify and locate jamming systems.

Detecting unauthorized mobile phone use: Controlled and high security areas such as prisons will frequently prohibit the use of cellular phones in certain areas, but may not jam operating frequencies because of other regulations. Identifying and locating transmissions allows direct action to be taken on equipment users.

A.5 Network Operator Applications

Radio planning for fixed radio deployment.
Spectrum forensics for identifying sources of interference.

Annex B Normative Functional Requirements

B.1 Tasking Agent Requirement

Tasking Agents accessing a distributed SCOS system will require access to one or more spectrum sensor devices situated at a remote location through an Internet-connected interface.

This interface must expose all of these functions to the SCOS system user through a defined, standardised interface

- The TA can discover the availability and capability of sensor devices connected to a Sensor Management System
- The TA can request the performance of a scan task according to chosen parameters, or in a more advanced system request a recurring scan to be scheduled that will be executed automatically by the SCOS system
- The TA can define where the data generated in the scan should be transmitted to once complete
- The TA must be given diagnostic or performance information relating to the execution of their scan task

The user must be able to access their scan data and associated metadata describing the scanner's environment, hardware and software configuration and scan settings.

B.2 Data Quality and Definition

Data acquired must be accompanied by adequate metadata information to allow for duplication of the measurement or assessment of data quality by a subject matter expert. Hardware specification and measurement parameters are to be included in the messaging metadata requirements. Information less amenable to metadata capture should be made available via appropriate and accessible documentation, e.g., algorithm described via written article with source code in Github repository.

B.3 Regulatory requirements

This standard should provide mechanisms to meet the regulatory requirements of national operators that have defined parameters or requirements for spectrum sensing in various applications. These regulatory requirements would take two forms: the first is technical requirements for sensitivity, resolution, etc. The second is limitations on how and where sensing might be done where there are sensitivities around privacy, military use and other national policies and regulations.

B.4 Policy Management and Enforcement Requirements

To allow for granularity in what the SCOS systems can do, but also ensure spectrum occupancy or utilization data is not exposed in contravention of national policy or regulation, it is proposed that the SM would be able to apply policies to allow or disallow certain functionality in the SDs, or disallow transmission of the data to third party systems.

These policies would be determined by the SM operator in accordance with their requirements and that of local authorities (e.g. a national regulator or network operator), and cascaded down to any connected SDs.

These policies would allow sensing only according to allowed metrics (e.g. no hi-resolution raw scans in military radar bands), and limit sensing data transmission to certain classes of third party systems.

B.5 Sensor Location-Fixing Requirements

The SCOS device can convey the location of the sensors to the aggregation entity such as the SM. The instruction to use available location capabilities on the SD (e.g. GPS location) will be part of the scan schedule instruction from the Tasking Agent requiring the scan. This feature allows the SM or the aggregation entity to localize the proximity of the signal source location allowing more efficient spectrum management. This location fixing capability will be implemented by the system operator to be in accordance with local regulatory requirements.

Location can be fixed in three ways:

- At scan-time from lat/long co-ordinates from internal GPS, GLONASS or similar system
- Configured at startup-time from lat/long co-ordinates from internal GPS, GLONASS or similar system
- Configured on SD by authorised/certified installer at commissioning time with accurate lat/long coordinates
- Configured on SD by authorised/certified installer at commissioning time with street address/location

The type of location fix would be specified in device metadata (NOTE: ADD REFERENCE).

B.6 Service Level Agreement Requirements

To be completed.

B.7 Certification Requirements

To be completed.

B.8 Technical Requirements

B.8.1 Device classes and complexity

The following sensing device categories may be considered:

- **Energy Efficient Sensing Devices**: This standard should provide mechanisms of energy efficient operations, eg. solar powered or battery operated spectrum sensors for monitoring applications.
- **Small form factor devices**: Devices that can fit the spectrum sensing function within a small form factor (e. g. a USB dongle, cell phone etc.)
- **Advanced Spectrum Sensing Devices**: Advanced Spectrum Sensing Devices with capable Radio Frequency Front Ends (RFFE) and dedicated resources for spectrum sensing may be considered.
- **Non-dedicated Devices with Sensing Capability:** A number of consumer and professional radio devices contain radio receivers that can be used as sensing devices, including mobile phone handsets, Wi-Fi access points (from 802.11ac) and Dynamic Spectrum Access radio systems (including 802.22).

B.8.2 Number of devices

This standard shall support at least one Spectrum Sensing Device to cover a location or area, communicating with a back-end Spectrum Sensing Management System (SM), but will extend to describing an architecture and interfaces for multiple SDs potentially communicating with multiple SM instances.

B.8.3 Real-time applications

The sensing devices will be performing spectrum sensing functions according to its scheduler (which is managed by its SM), which can be updated in near-real time (dependant on speed of communication between Tasking Agent, SM and SD), or perform scans at scheduled intervals based on pre-configured schedules. However, the spectrum sensing reporting of data is out on a Best Effort basis, since the SCOS System uses the chosen available transport mechanism (e. g. 802.11, 802.22, Ethernet, Cable, Cellular etc.).

The SCOS system will benefit if sensing reports from various sensors are provided on a reasonable time-scales (e. g. minutes) so that the information is not stale. However, this is not a mandatory requirement. Also, the messaging format may be defined such that it does not produce excessive overhead penalty on the transport layer being used.

It is envisioned that real-time streaming will be provided for in future drafts of 802.22.3.

B.8.4 Channelization

This standard may specify a Spectrum Manager entity that can command various sensors to go and sense in certain bands, or it may even specify the spectrum sensors to ignore certain bands from sensing, and impose channelization maps for sensors to meet local regulations or technical requirements.

Specific channelization maps may be provided in future drafts of 802.22.3.

B.9 Security Requirements

The standard mandates secure authentication and authorisation between Tasking Agent and SM, SD and SM, SM and Data Manager, and Data Manager and DC. Traffic between these components must also be encrypted. The specific security technology to be used is not mandated in this standard, but recommended best practices are described in Annex B. Note that the security model does not extend past transmission to the DC. Responsibility for securing the spectrum data at destination remains the responsibility of the operator of that store.

The technology model is designed to ensure that data derived from SCOS devices and SM are not used as an attack vector against White Space Databases, regulator spectrum management databases, etc. It is also designed to ensure that only authorised Tasking Agents can make use of the SCOS resources, and that they are correctly identified to enable correct application of the relevant system policies.

In each case, the security model must address:
(1) Data categorization (i.e. sensitivity/confidentiality of scan data)
(2) Access control - authorization and authentication (of each element when interacting with another)
(3) Logging and auditing (of instructions, tasks, access control)
(4) Data encryption (within devices and in transmission)
(5) System and information integrity (validation of device configuration, storage system)

(NOTE: Section to be developed further)

B.9.1 Intra-device Layer Security (physical interfaces)

This standard defines security mechanisms to ensure integrity of sensing chain from antenna to DC.

- Antenna to amplifier/filters: physical security of device in terms of cable/connectors (tampering such as substituting antenna, physical such as connector corrosion)
- Amplifier to SDR: cable connectors or PCB connections
- SDR to processing unit: cable connectors or PCB connections
- Enclosure for active elements: Protection against moisture, dust ingress. Screening against RFI from external sources. Screening to protect antenna elements against RFI from active elements.

(NOTE – this section needs considerable attention)

B.9.2 Inter-Layer Security

B.9.2.1 Network Layer

Since this standard uses any available transport mechanism for data transmission, it will not recommend its own security mechanisms, but will use the existing security mechanisms of the transport mechanism being used (e.g. network 802.11 using Transport Layer Security,

B.9.2.2 Application Layer

Data transmissions should be secured on the application layer using mechanisms to guarantee the integrity and confidentiality of sensing and control data transmissions. This standard does not specify the technology used, but recommended implementation practices are noted in Annex B: Device and System Security Recommendations.

B.9.3 Security of sensed data

This Standard shall not support mechanisms that expose data of radio system users that are modulated onto signals that are examined by the 802.22.3 SCOS System. For example, any kind of demodulation of the signals that may interfere with the privacy of the radio system users shall not be not be supported. However, the SCOS system shall support sophisticated spectrum sensing methods such as cyclostationary processing that can detect signals and characterize their modulation type.

B.9.4 Security of analyzed or characterized spectrum data

This Standard shall support security mechanisms to ensure that spectrum characterization data is transmitted to the destination DC in a secure way. This Standard shall not specify the security mechanisms used to protect this data once received by the DC – this is implementation and system user specific.

Annex C SCOS Operational Modes

To allow great system flexibility with ability to meet multiple unknown use cases, but also allow a simplified task-specific operational use, two Operational Models are proposed:

- Tasking SCOS Mode: This is a full-featured mode suitable for wide application, where the SM acts as a management device to allow multiple different users ("Tasking Agents") to do different scans
- CR Mode: This mode is suitable for cognitive radio implementations, where a sensing device is used in a semi-fixed configuration, reporting channel occupancy to the radio management system over heartbeat messages for low overhead, with some capability to perform specific scans as a task to let a radio supervisor system request a detailed scan.

CR Mode is a subset of Tasking SCOS Mode, using the same interfaces, primitives and protocols.

Further, "Offline Mode" is proposed for further examination and inclusion in later versions of standard. This mode would enable sensing devices to be given a task schedule, and then operate offline from the SCOS management systems, and synchronize data and tasks later when re-associated to management systems.

Annex D SCOS Topology Examples

An SCOS system consists of *a single SM*, *a single DM* which communicate over any standard network transport with one or more SDs, TAs, and DCs.

Figure 2 illustrates the above described SCOS system topology.

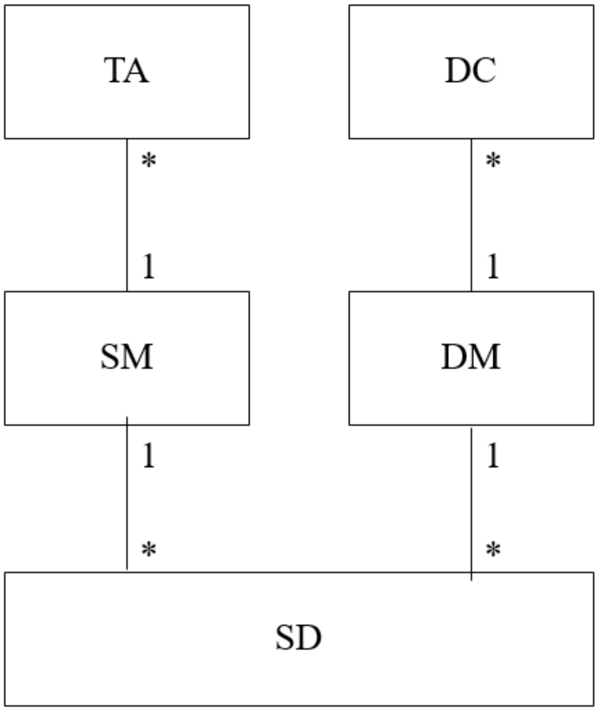


Figure 2: SCOS System Topology

1. Non 802.22.3 compliant SDs and SCOS Cascading

The 802.22.3 SCOS standard makes provision for proxying, that allows a non 802.22.3 compliant SD to associate with and be controlled by an SM, as well cascading of systems, where one 802.22.3 compliant SM to be associated with, and delegate tasks to, another 802.22.3 SCOS system.

SD Proxy facilitates an SM communicate with proprietary sensing hardware, acting as a software translation mechanism that translates between SCOS messages.

SM Proxy enables cascading of SCOS systems where an SM can communicate with other SMs as if they were associated SDs.

Figure 3 illustrates the extensions with an instance of system topology.

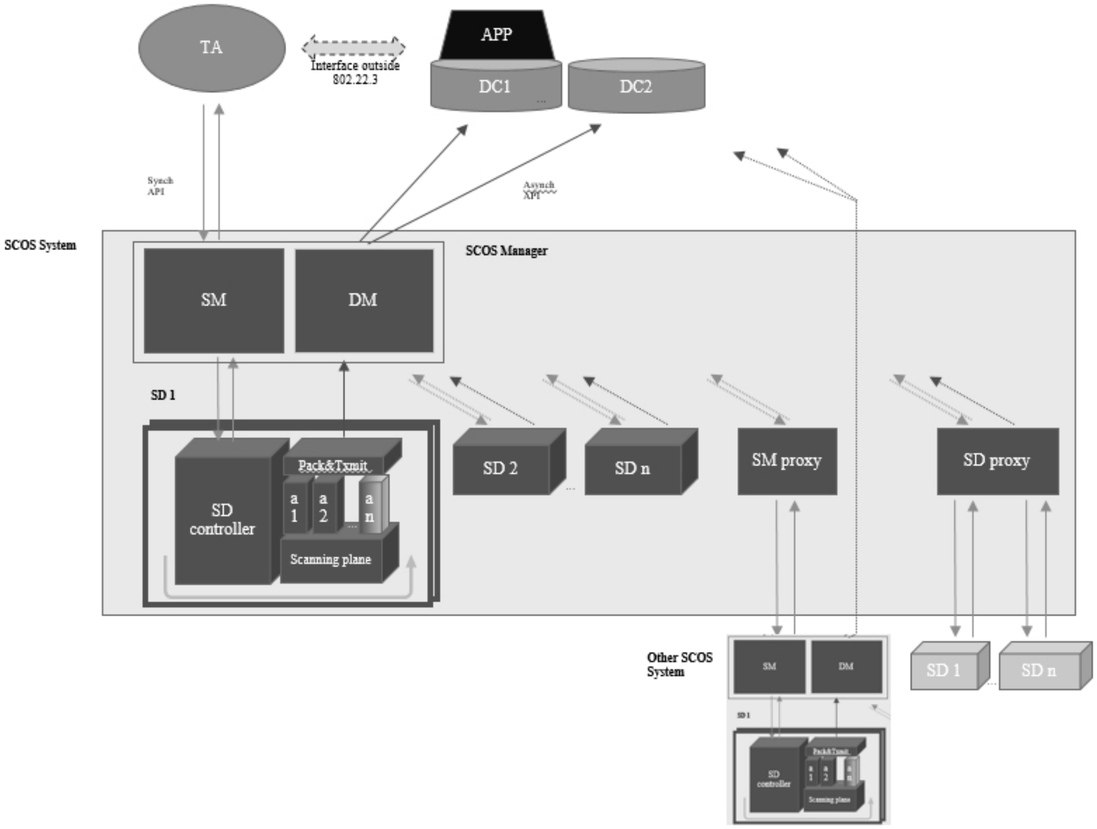


Figure 3. An SCOS System Instance with Extensions

Annex E System Policy Model

E.1 SCOS Policy

A policy layer in the SM at the northbound and southbound API will ensure that the SM is operated within requirements of a local authority (national regulator, law enforcement, military, etc).

The policy on the southbound interface will determine, based on the USER type, (e.g. how a central authority can define what kinds of sensing can be done in what bands, what data governance rules there are, etc
-- resource allocation – what kinds of users are authorized to request resources from the sensor network and in which priority (i.e. if a sensing network is resource constrained, who gets first dibs on the sensors)

Policy on the northbound interface (SM>DBstore) will have rules for how sensing data may be distibuted, and data storage policies. This would include which scan data takes priority if local storage in the store & forward buffer is running out due to a failed transmission link, and how long certain USER data is allowed to be stored in the local store & forward buffer.

Based on SCOS platform architecture, the SCOS policy is categorized based into following categories:

- Spectrum Sensing Manager (SM) Policy
- Spectrum Sensing Device (SD) Policy
- Sensing Data Manager (DM) Policy
- SCOS Platform Administration (SPA) Policy
- SCOS Platform User (SPU) Policy

The SCOS policy is expressed using JSON. Following subsection provides details about the schema of SCOS policy.

E.1.1 The SCOS policy schema:

Each SCOS policy is associated with a policy-name, policy-namespace, policy-category, policy-scope, optional policy-description, and one or more statement(s).

Following figure shows the structure of a SCOS policy.

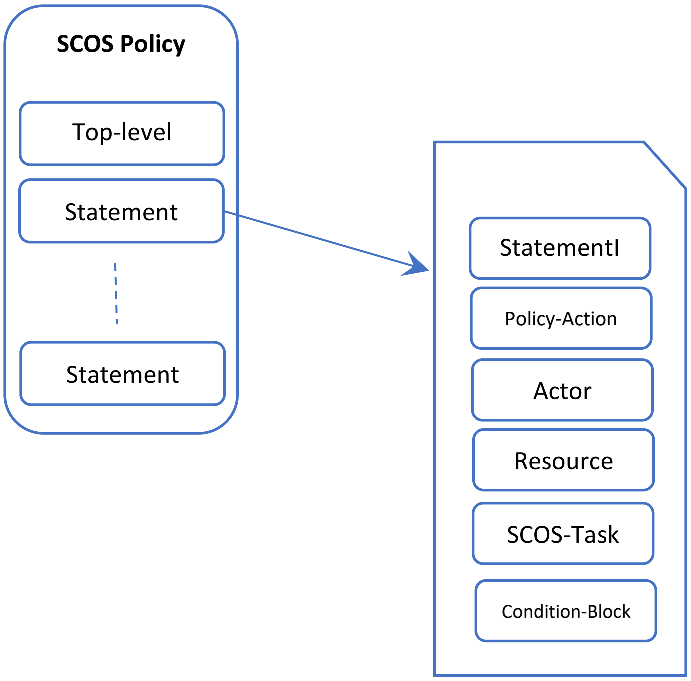


Figure XXX. High-level structure of the SCOS Policy

Example:

```
{
#  "Version": "2017-02-15",
  "Policy": {
    "namespace": "OperatorFoo",
    "name": "Calibration-access-control",
     "description": "This policy added by FooAdmin On this date."
    "Policy-type": "SPA-Policy",
    "Policy-Action": "permit",
   "Resource": "Foo:Sensors::*"
   "SCOS-Task": "Calibration-operation"
   "Scope": "Sensor-management:"
    "Condition": {
       "equals" : {
        "Actor" : "FooAdminRole"
      }
   }
}
```

The default namespace is global. Specific-Namespaces could be used to restrict the application of policy within a certain context. Namespaces avoids name collisions and enables to identify actors or resources uniquely (when the names have been reused across namespaces).

SCOS Policy Statement

Each statement specifies certain action. Different categories of SCOS-Policies are associated with different actions.

A statement may also have optional attributes that identify the context of applying the policy. These attributes allow specifying a fine-grained policy. Example context-attributes are Actor, Resource, SCOS-Task, time, frequency, and location.

A statement has an optional condition-block. The action is performed only when the condition(s) are matched.

Tasking Agent
A Tasking Agent is an entity that wishes to use the SCOS platform. A Taskin Agent could be specified in terms of role, user, or a user-group.

User
An user is an individual Tasking Agent with specified name and access-credentials.

User-group
A user-group is a logical collection of users. A user-group is specified with name and access-credentials.

Role
A role is specified with a name. The role could be associated with specific SCOS services/functionality. A role could also be associated with privilege of various users of SCOS platform.
An user (or a user-group) is associated a role.

Resource
SD and Sensing data are two prime resources within SCOS platform.

Resource-group
Multiple SDs could be grouped together to jointly specify policies for using the SDs. SDs could possibly be grouped based on various attributes such as location, SD-hardware-type, SD-software-type.

Namespace
Actors or resources are associated with a namespace. This avoids name collisions and enables to identify actors or resources uniquely (when the names have been reused across namespaces).

SCOS-Task
An SCOS-task represents a specific sensing-task within SCOS platform issued by a particular Actor on particular resources. Additionally, a policy may specify pre-defined SCOS operations in the SCOS-Task. These predefined SCOS-Tasks include: Scanning, Calibration, Storage, and Transmission.

Task-group
Multiple tasks could be grouped together for convenience in specifying policies. For example, various tasks that can be performed towards sensor-management for a particular SD operator could be grouped together and referred to in the SCOS policy. Similarly, sensing data management related tasks could be grouped together for precisely and conveniently specifying sensing-data-management related policies.

Conditions
A condition is specified with a triplet of field(key), conditional-operator, and value. Condition is optional within a statement.
A condition evaluates whether a field meets certain criteria. Following table identifies various conditional operators.

Conditional-operator Name	**Syntax**
equals	"equals" : "<value>"
Like	"like" : "<value>"
Contains	"contains" : "<value>"
In	"in" : ["<value1>","<value2>"]
Exists	"exists" : "<bool>"

LessThan	"lessthan" : "<value>"
GreaterThan	"greaterthan" : "<value>"
LessThanEquals	"lessthanequals" : "<value>"
GreaterThanEquals	"greaterthanequals" : "<value>"

Logical Operators
Logical operators enable to manipulate or combine multiple conditions. Following table specifies the logical operators.

Logical operator	**Syntax**
Not	“not”: {<condition>}
AllOf	"allOf" : [{<condition>},{<condition>}]
AnyOf	"anyOf" : [{<condition>},{<condition>}]

Aliases
Aliases add convenience. Using aliases, multiple users can be combined together or multiple resources can be combined together to be referred in the SCOS policy . Furthermore, multiple tasks can be combined using task-groups.

Furthermore, locations could be specified using aliases to capture latitude, longitude, and altitude. A group of frequencies could also be combined using aliases. A group of time-slots also could be combined using aliases.

E.1.1.1 SM Policy Schema

Each SM policy has following required fields: PolicyName, PolicyScope, PolicyType, and PolicyAction.
Optional fields include: Policy-Description, condition-block, Actor, Resource, and SCOS-Task.
With Policy-Action ‘set’, SM attribute and value(s) could be specified.

E.1.1.2 SD Policy Schema

Each SD policy has following required fields: PolicyName, PolicyScope, PolicyType, PolicyAction, and Resource.
Optional fields include: Policy-Description, condition-block, Actor and SCOS-Task.
Policy-Actions: Set, Permit, Deny, Calibrate, Scan,
With Policy-Action ‘set’, SD attribute and value(s) could be specified.

E.1.1.3 DM Policy Schema

Each DM policy has following required fields: PolicyName, PolicyScope, PolicyType, and PolicyAction, and SCOS-Task.
Optional fields include: Policy-Description, condition-block, Actor and Resource.
Policy-Actions: Set, Permit, Deny, Transmit-Sensing-Data, Store-Sensing-data, Discard-Sensing-data

With Policy-Action ‘set’, DM attribute and value(s) could be specified.

E.1.1.4 SPU Policy Schema

Each SPU policy has following required fields: PolicyName, PolicyScope, PolicyType, PolicyAction, and Actor.
Optional fields include: Policy-Description, condition-block, SCOS-Task, and Resource.
Policy-Actions: Set, Permit, Deny
With Policy-Action ‘set’, SPU attribute and value(s) could be specified.

E.1.1.5 SPA Policy Schema

Each SPA policy has following required fields: PolicyName, PolicyScope, PolicyType, and PolicyAction.
Optional fields include: Policy-Description, condition-block, SCOS-Task, Actor, and Resource.
Policy-Actions: Set, Permit, Deny,
With Policy-Action ‘set’, SPA attribute and value(s) could be specified.

E.1.1.6 Policy Specification

In the first version of this standard that the SM and DM store a policy file which is installed manually by the SCOS operator (through mechanism such as SSH and update pull, or remote SCP).

E.1.2 Policy Evaluation

Whenever an SCOS API needs to be executed, SM needs to confirm if the action is permitted by evaluating related policies.

There exist three scopes for SCOS policies: Sensing management scope, Sensing-data management scope, and Sensor-management scope. Depending on the API, policies in the appropriate scope are looked up.

The second step is ensure that the actor is authorized to perform tasks on the resource.   A specific accept policy or default-accept policy should be match for the user, user-group, or role.

The final step is ensure if the resource permits the intended task. A specific accept policy or default-accept policy should be match for the resource, or resource-group.

E.1.3 SCOS Policy Examples

E.1.3.1 SD Policy

Set sensitivity to -114 dBm task frequency UHFBand
PolicyID: <generated>
PolicyName: SCOSMinSensitivityRule
Policy-Category: SD-Policy
PolicyDescription: It applies to all SDs within the SCOS operational region.
Policy-Action: Set

Sensitivity: Value
Frequency: value

Discuss: Should frequency be within the context-block or condition-block?

E.1.3.2 SM Policy

Set scheduling minimum slot duration
PolicyID: <generated>
PolicyName: SSMMinSensingSlotDuration
Policy-Category: SM-Policy
PolicyDescription: It specifies the minimum slot duration for sensing task. The value is in seconds.
Policy-Action: Set
Min-sensing-slot-duration: Value (seconds)

Note: Fine-grained policy could be specified for a particular resource (SDs) or sensing-tasks.

Set sensing behavior for prioritized scan
PolicyID: <generated>
PolicyName: SM-Prioritized-Scan-Behavior
Policy-Category: SM-Policy
PolicyDescription: It specifies whether existing scan should be suspended if a prioritized scan-request is received.
Policy-Action: Set
Prioritized-scan-enabled: true
Condition-block: if wait-time greaterthan value (in seconds)
Note: Condition-block is optional. Condition-block can be used to specify a condition when existing scans can be suspended.

E.1.3.3 DM Policy

Set max-data-storage-duration at DM
PolicyID: <generated>
PolicyName: SDMMaxStorageConfig
Policy-Category: DM-Policy
PolicyDescription: It specifies how long DM can hold the sensing data.
Policy-Action: Set
Max-data-storage-duration: Value (seconds)

Note: Optionally specify task or SDs or SDS. The value is in seconds.

Discard sensing-data for <scan-task-L-band-User-Jim> if sensing-data is unqualified.
PolicyID: <generated>
PolicyName: User-Jim-L-Band-Discard-Data-Policy
PolicyDescription: If sensing data does not meet the criteria specified in the sensing task, discard the data.
Policy-Category: DM-Policy
Policy-Action: discard-data
SCOS-Task: scan-task-L-band-User-Jim
Condition-block: sensing-data-quality is ‘unqualified’.

Note: The condition-block identifies when the operation is performed. Here, sensing-data has attribute sensing-data-quality. The condition is satisfied when the attribute's value is unqualified. The sensing-task is identified pre-defined using name-alias. Optionally, the policy could be made more specific for certain time, and location attributes.

E.1.3.4 SPA Policy

Enable SM-Proxy device usage in SCOS system.
PolicyID: <generated>
PolicyName: Enable-SD-Proxy-Config
Policy-Category: SPA-Policy
PolicyDescription: Enable SM-Proxy devices in the SCOS platform.
Policy-Action: Set
SM-Proxy-Enabled: Boolean-Value

Note: Optionally, the policy could be made more specific for certain frequency, time, and location attributes.

E.1.3.5 SPU Policy

Deny scan operation for User-Foo in the military bands
PolicyID: <generated>
PolicyName: MilitaryBandScanRestrictionPolicy
Policy-Category: SPU-Policy
PolicyDescription: Deny certain users/roles/groups to scan in certain bands.
SCOS-Task: scan
Actor: User-Foo
Policy-Action: Deny
frequency: X-band

Note: The actors could be specified with user/role/user-group. The frequency bands could be pre-defined using name-aliases. Optionally, the policy could be made more specific for certain time, and location attributes.

E.1.3.6 SDS Policy

Send sensing-data for <scan-task-L-band-User-Jim> to data-store <FooStore3>
PolicyID: <generated>
PolicyName: User-Jim-L-Band-DataStorePolicy
Policy-Category: SDS-Policy
PolicyDescription: Configure data store for a scan request.
Policy-Action: Transmit-sensing-data
SCOS-Task: scan-task-L-band-User-Jim
Resource: FooStore3

Note: The data-store is specified with resource on which operation is done. The sensing-task is identified pre-defined using name-alias. Optionally, the policy could be made more specific for certain time, and location attributes.

Annex F Informative: Latency Requirements for Scans

The latency requirement for performing a sensing task and transmitting metadata from SD to DM is a critical metric for certain use cases.

The maximum allowed latency depends on signal type and frequencies that need to be scanned, and are determined by the application. As such, a number of reference applications are given, with recommended latencies. In each case, to meet the specified latency, the SCOS system design and particular implementation would need to be capable.

Maximum Latency would be the sum of:

- Task Request Latency: Time from scan request to scan start
- Scan Time Latency: Time from scan start to scan complete
- Spectrum Characterisation Latency: Time from scan complete to algorithmic processing
- Sensing Data Delivery Latency: Time from processing to delivery to Data Client

Reference Application	**Band Swept**	**Maximum Latency**
TVWS Base Station Device	460-760MHz	
CBRS Base Station Device	3.5GHz-3.8GHz	
TVWS CPE Device	460-760MHz	
CBRS CPE Device	3.5GHz-3.8GHz	
Etc	460-760MHz	

Annex G Informative: Regulatory Technical requirements

Various countries will have differing requirements here, but a few countries already have definitions in place that should be observed. For example, in the FCC rules for the VHF/UHF TV bands, the FCC requires a spectrum sensing detection accuracy as specified by the Table 1: FCC Sensing sensitivity requirements.

Table 1: FCC Sensing sensitivity requirements

Regulatory domain	**Type of signal**	**Sensing detection threshold (in dBm)**	**Data fusion rule for distributed sensing**[a]	**Monitoring requirements**
USA	ATSC	–114 (averaged over 6 MHz)	“OR” rule	Detection threshold referenced to an omni-directional receive antenna with a gain of 0 dBi
USA	NTSC	–114 (averaged over 100 kHz)	“OR” rule	Detection threshold referenced to an omni-directional receive antenna with a gain of 0 dBi
USA	Wireless microphone	–107 (averaged over 200 kHz)	“OR” rule	Detection threshold referenced to an omni-directional receive antenna with a gain of 0 dBi

[a]The value “1” indicates detection.

Other requirements for the 2.7 GHz to 3.7 GHz band shall be defined based on the evolving regulations. For example, the spectrum sensing devices in the 2.7 GHz to 3.7 GHz can sense for Radar Signals and provide that information to the Spectrum Access System (SAS) that is being defined in these bands.

Annex H Device and System Security Recommendations

* Remote access to SD hardware through remote secure shell (SSH) and similar technologies must not use the same keys as SM/SD interface keys

* Devices' physical characteristics must be evaluated and enumerated at build validation and testing, with hardware and configuration parameters written to file /DEVICEHARDWAREPARAMETERS.CONFIGFILE (placeholder) and stored in non-writable file

* Any changes to hardware configuration (e.g. change of antenna) must be recorded in DEVICEHARDWARECONFIGCHANGES.LOGILE and changes made to relevant parameters in D..H...P...CONFIGFILE, either through manual editing of config file or through a secure remote update mechanism (e.g. scripted SCP file revision).

Annex I Radio performance requirements

I.1 Sensitivity and Noise

Annex J (informative)

Sensing

This annex contains descriptions of a number of sensing techniques. A sensing technique is an implementation of the spectrum sensing function.

J.1 References

Annex K Implementation Guidelines/Notes

<Currently this section contains miscellaneous notes>

Tasking API, Mission API and Data Request API: The Mission API would be the equivalent of the SCOS API (i.e. where the Tasking Agent requests a scan schedule); the Task API would be the SD API (i.e. the SM sending an schedule update to a specific SD). The Data Request API would not be included in the current design, as the retrieval of scan data would be between the Tasking Agent and the Data Client, which is implementation dependent.

K.1 Management Reference Architecture

K.1.1 Spectrum Sensing Platform – Sensing Service Control

K.1.1.1 Spectrum sensing API

The spectrum sensing platform provides spectrum sensing as a service using Spectrum sensing API. This is the northbound interface from the block diagram.
We identify following four types of API

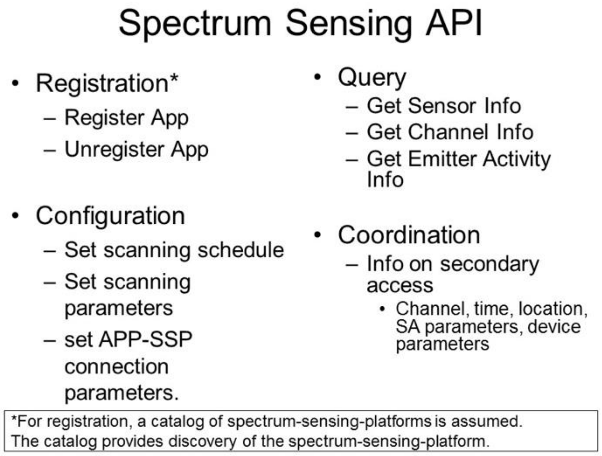
1. Registration (ID, key exchange, authorisation)
2. Query (sensor model, signal processing capability (occupancy, characterisation, calibration, df), health, availability, location)
3. Configuration (sensing config, scheduling config, calibration, [operational])

4. Notification of Change (reverse Query)
5. Notification of Busy (TBC?)

With the Registration API, an SSA can enable/disable usage of the API. Configuration API enables an SSA to configure the SSP for desired purpose. Using Query API, an SSA can request real-time data or past data. (*Inference regarding secondary spectrum-access is purposefully excluded from the SSP API. For example, Is it safe to transmit? This spectrum-access inference logic is considered to be in the apps that are using the spectrum-sensing platform*.)

The coordination API is optional. It can be used in circumstances wherein the Apps wants to provide information about secondary spectrum-access. For example, an SSA may use the real-time sensing data and infer feasibility of secondary spectrum-access. This SSA would grant spectrum-access parameters to secondary user radios and use the coordination API to notify the secondary spectrum access to SSP.

Following diagram captures the high-level summary of the SSP API.



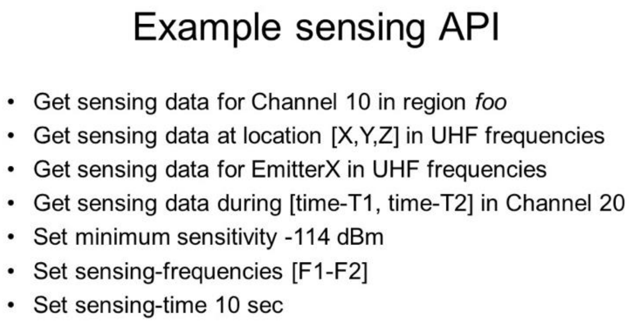
Spectrum Sensing API

- Registration*
  - Register App
  - Unregister App
- Configuration
  - Set scanning schedule
  - Set scanning parameters
  - set APP-SSP connection parameters.
- Query
  - Get Sensor Info
  - Get Channel Info
  - Get Emitter Activity Info
- Coordination
  - Info on secondary access
    - Channel, time, location, SA parameters, device parameters

*For registration, a catalog of spectrum-sensing-platforms is assumed.
The catalog provides discovery of the spectrum-sensing-platform.

K.1.1.2 Example spectrum sensing API design

The spectrum sensing requirements vary significantly in terms of geographies (different countries have different regulations) and they have been evolving over time. The requirements also vary depending on frequency-bands. Thus, there is a need for configurability and extensibility for SSP API. In this regard, policy-based interface is very much appealing. Furthermore, we may consider developing semantics for sensing-data and ontology-driven sensing policy (OWL). Following diagram shows some examples of possible SSP API.

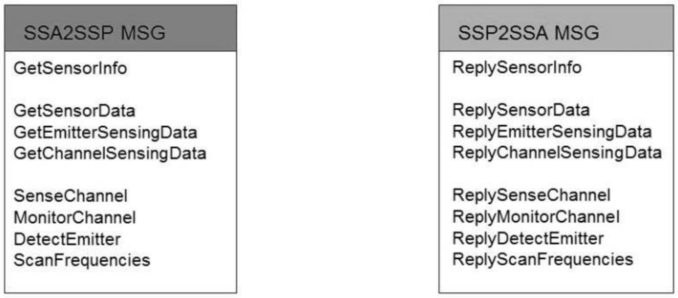


Example sensing API

- Get sensing data for Channel 10 in region *foo*
- Get sensing data at location [X,Y,Z] in UHF frequencies
- Get sensing data for EmitterX in UHF frequencies
- Get sensing data during [time-T1, time-T2] in Channel 20
- Set minimum sensitivity -114 dBm
- Set sensing-frequencies [F1-F2]
- Set sensing-time 10 sec

K.1.1.3 Message Exchange

SSA API requests and SSP API response are encapsulated in messages. Each message has a message-ID, message-Type, and the message body. Following diagram identifies various message types.



SSA2SSP MSG

GetSensorInfo

GetSensorData
GetEmitterSensingData
GetChannelSensingData

SenseChannel
MonitorChannel
DetectEmitter
ScanFrequencies

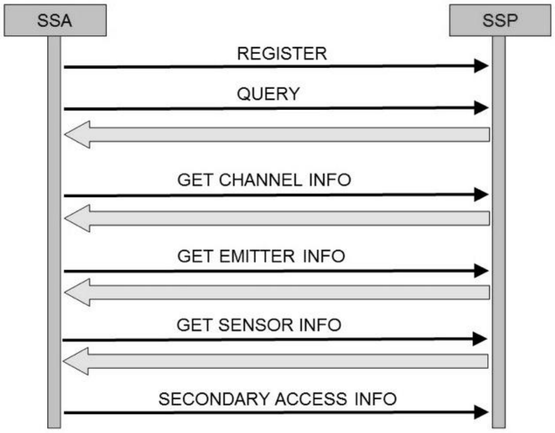
SSP2SSA MSG

ReplySensorInfo

ReplySensorData
ReplyEmitterSensingData
ReplyChannelSensingData

ReplySenseChannel
ReplyMonitorChannel
ReplyDetectEmitter
ReplyScanFrequencies

Following sequence diagram illustrates message exchange between SSA and SSP.



K.1.2 Spectrum Sensing Control

The spectrum sensing platform provides spectrum sensing service by controlling the spectrum sensing devices (SD) with southbound interface. There are following 3 types of API

1. Registration: Allows to add/remove an SD to SSP
2. Control: Controlling the sensing function and schedule of an SD
3. Query: Requesting sensing data from an SD

K.1.2.1 Sensing Functions

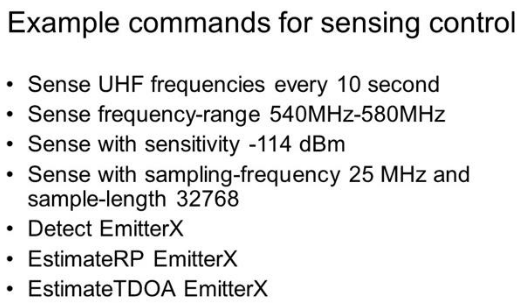
There exist multiple sensing techniques/algorithms from energy detection to exploiting cyclostationarity and signal statistics. Some sensors may be able to report occupancy in terms of aggregate RF-power received at the sensor location while higher end sensors may be able to estimate location and received power (RP) in the presence of cochannel interface and noise.

K.1.2.2 Sensing Schedule

The SSP may need to scan a wide range of frequencies at a specific periodicity. Thus, SSP may in tur define a sensing schedule for each of SDs. The schedule may be adapted in response to certain events or policies from the SSAs.

Examples

Following are a few examples of the interface between the SSP and SD.

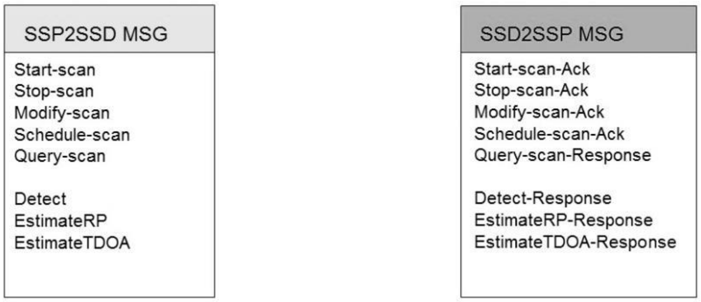


Example commands for sensing control

- Sense UHF frequencies every 10 second
- Sense frequency-range 540MHz-580MHz
- Sense with sensitivity -114 dBm
- Sense with sampling-frequency 25 MHz and sample-length 32768
- Detect EmitterX
- EstimateRP EmitterX
- EstimateTDOA EmitterX

K.1.2.3 Message Exchange

The message from SSP to SD is formatted in the similar way (has message-ID, message-type, and actual message). Following diagram shows some of the message types.



SSP2SSD MSG

Start-scan
Stop-scan
Modify-scan
Schedule-scan
Query-scan

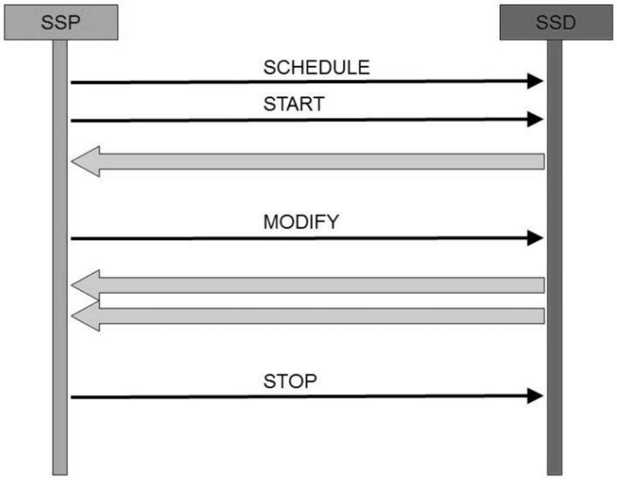
Detect
EstimateRP
EstimateTDOA

SSD2SSP MSG

Start-scan-Ack
Stop-scan-Ack
Modify-scan-Ack
Schedule-scan-Ack
Query-scan-Response

Detect-Response
EstimateRP-Response
EstimateTDOA-Response

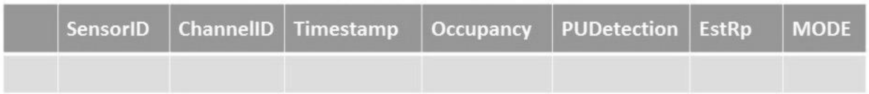
Following sequence diagram illustrates the message exchange between SSP and SD.



K.1.2.4 Data Client

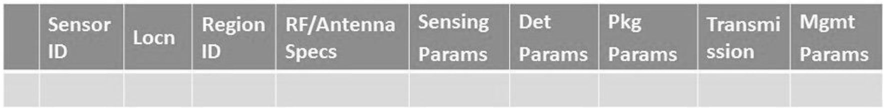
SSP collects and stores the sensing data to provide the services defined under the SSP APi. One of the popular approaches is to use relational database. Following diagram illustrates records for (a) sensing measurement, (b) SD (c) SSA.

Sensing measurement Record



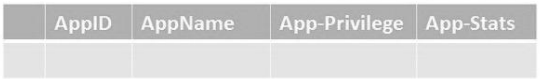
	SensorID	ChannelID	Timestamp	Occupancy	PUDetection	EstRp	MODE

Sensor Record



	Sensor ID	Locn	Region ID	RF/Antenna Specs	Sensing Params	Det Params	Pkg Params	Transmission	Mgmt Params

Spectrum-sensing App Record



	AppID	AppName	App-Privilege	App-Stats

Alternate approach could be to develop spectrum sensing semantics based data-store.

K.1.2.5 Management and Maintenance

The back-end management system exchanges control information with SDs
Manage them

- Device health reports – power, temperature, location, GPS health, OS/environment health, network health, storage health, scheduled and by query
- Manage by device, group, Class of device

Validate their operation

- Run test scans against known data points (e.g. from WSDB)

Verify integrity of information chain

- Software tools to validate process chain

Perform maintenance

- Push updates to devices (software, OS, firmware, certifications)
- Perform remote reboots, resets
- Shell into device to do diagnostics

Annex L (normative)

IEEE 802.22 regulatory domains and regulatory classes requirements

This annex describes the various technical parameters and specifications required by the various regulatory domains for operation of the IEEE Std 802.22 in the TV bands.

A.1 Regulatory domains, regulatory classes, and professional installation

Table I.1 specifies the regulatory domains and licensing regime where the IEEE 802.22 systems are planned to be authorized to operate in the TV bands.

Table I.1—Regulatory domains

Geographic area	Regulatory domain ISO 3166 (3 Bytes)	Licensing regime	Approval authority
United States	USA	Unlicensed	FCC
Canada	CAN	Licensed	IC
United Kingdom	GBR	—	OFCOM
—	—	—	—

Table I.2 specifies the authorized regulatory classes under their respective regulatory domains.

Table I.2—Regulatory classes

Regulatory domain	Regulatory class and profile	
	Fixed	Personal portable
USA	Stationary fixed	Mode I & II[a]
CAN	Stationary fixed	N/A
—	—	—

[a] The behavioral limits sets for Modes I and II are defined in the FCC Report and Order. However, IEEE Std 802.22 will only operate in portable nomadic Mode II.

Table I.3 specifies the requirement for professional installation of the WRAN BS and CPEs.

Table I.3—Professional installation requirement

Regulatory domain	Type of terminal		Definition of professional installer
	BS	CPE	
USA	Professionally installed	Professionally installed	A professional installer is a competent individual or team of individuals with experience in installing radio communications equipment and who normally provides service on a fee basis—such an individual or team can generally be expected to be capable of ascertaining the geographic coordinates of a site and entering them into the device for communication to a database.
CAN	Professionally installed	N/A	Same as for USA.
—	—	—	—

Annex M (informative)

Bibliography

At the time of publication, the editions indicated were valid. All standards and specifications are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the references listed below.

[B1]