**Protocol to Access   
Spectrum Characterization and Occupancy Sensing (SCOS) Network**

*San Antonio Plenary Team: Ken Baker, Mike Cotton, Roger Hislop, Nilesh, Jerome Kalke, Nilesh Khambekar, Apurva Mody, William Suriaputra*

Date: Nov 7 – 10, 2016

1. **Protocol Overview**

The purpose of the Spectrum Characterization and Occupancy Sensing (SCOS) network is to acquire and make available data from networks of sensors. It is intended to establish a platform that enables “spectrum sensing as a service” and collective measurement efforts.

**<Purpose of protocol>**

This standard specifies functional models, interfaces, protocols, and policies to allow for creative design and development of SCOS building blocks with minimal design constraint. It does not specify sensor hardware, signal processing, or data quality. It does, however, call for complete definition of those elements to permit the subjective user to assess data quality at reception.

Although database design and visualization are critical element to spectrum measurement, we limit the scope of this standards to details required to establish machine to machine communications between user, SSM and SSD, as well as the end data store to which the scan data is transmitted. User data management and visualization is left for users to meet business and/or organizational goals.

Figure 3 illustrates the basic SCOS interface structure.

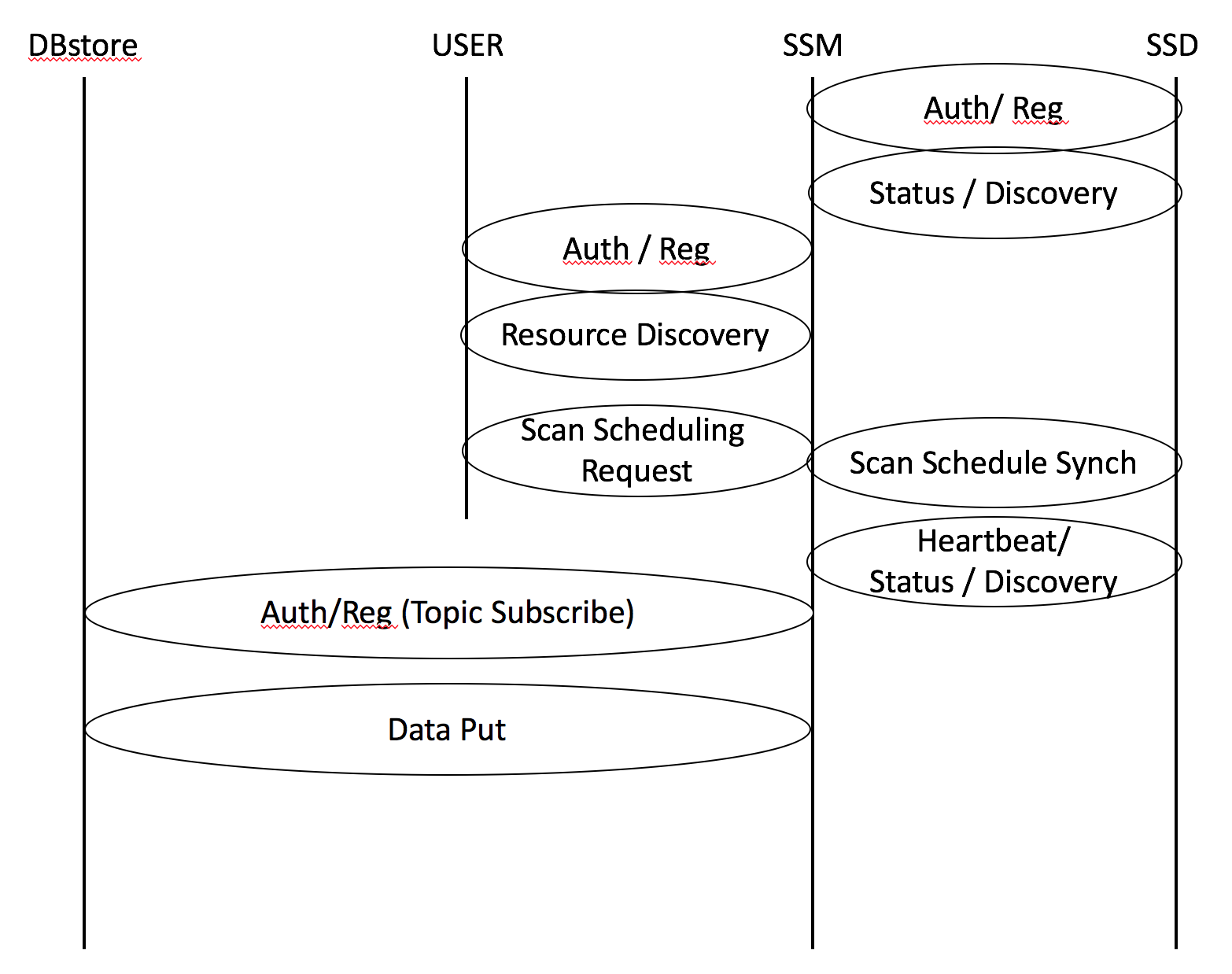


Figure . Basic Interface Structure

* 1. Entities

The entities shown in diagram are described as

* USER is the entity that initiates a spectrum monitoring request to one or more Spectrum Sensing Managers (SSM). Users can have various levels of privileges regarding what spectrum information collection can be initiated and what spectrum information can be accessed from a DBstore.
* DBstore is a data base for storing spectrum information collected from the sensing network. There can be multiple DBstores and these do not have to associated with a specific USER although a USER may have a private data base.
* SSM manages a collection of Spectrum Sensing Devices (SSD). Requests for spectrum measurements from USERs are inserted into a scan schedule on the SSM for all its attached SSDs, as far as possible under a set of slot availability rules. This schedule is synched to the appropriate SSDs associated with the SSM. Data from the SSDs are collected at the SSM for transmission to one or more DBstores for permanent storeage and processing.
* SSD is the sensing hardware that collects the spectrum data requested by the SSM on behalf of USERs. The SSDs 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.

1. **Interface Functions**

The communication between each of the entities defined above can be grouped and defined within the Interface Categories shown in Fig. 1 and described below.

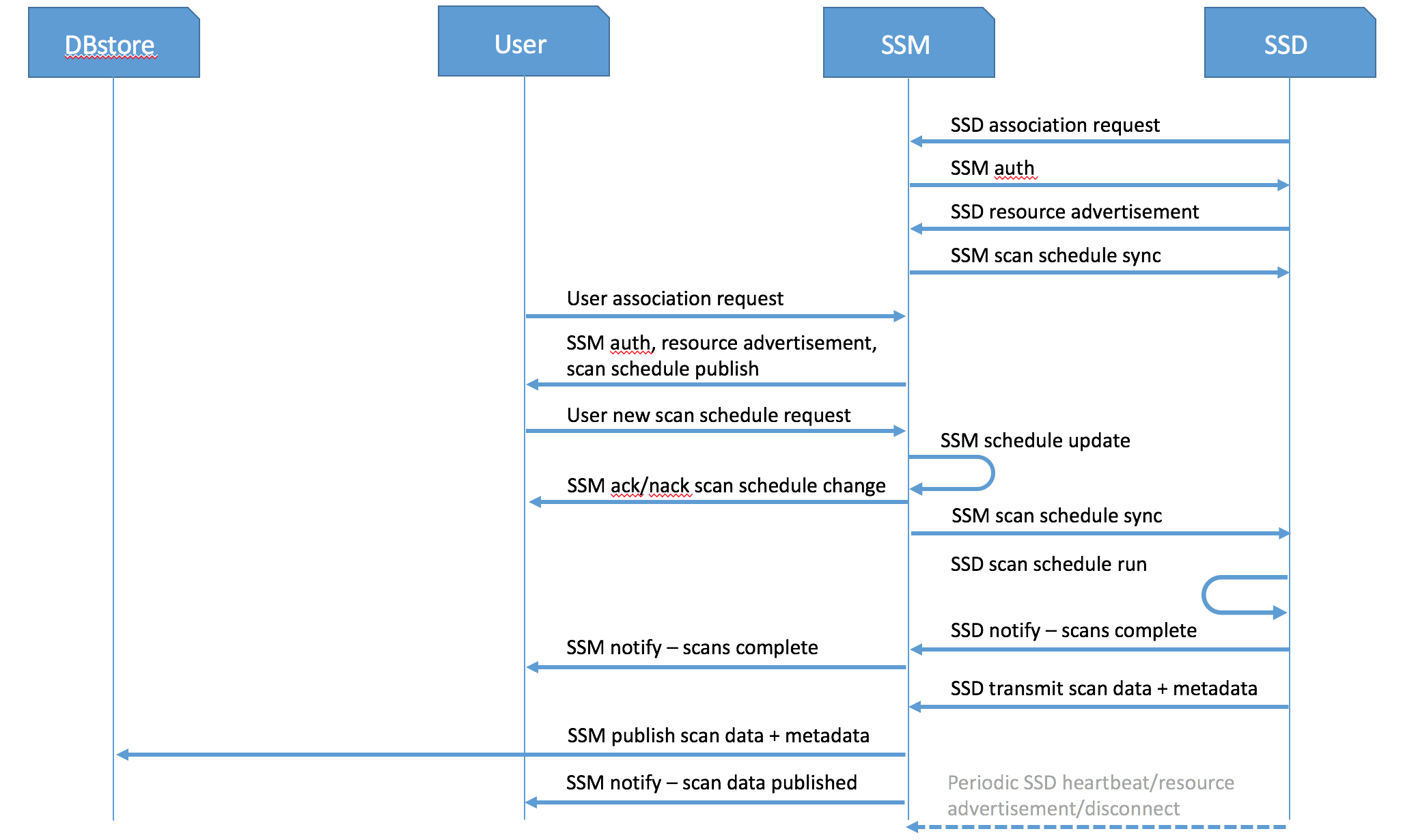


Figure 2. Message Sequence

* 1. **USER <−> SSM**
     1. Authentication and Registration

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

* + 1. Resource Discovery

Resource Discovery is the process of informing the USER of what capabilities that the SSM 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 descriptor and the current scan schedule per SSD.

* + 1. Scan Request

The Scan Request message from the USER to the SSM includes the parameters of the desired spectrum measurement to be made and any associated processing to be performed by either the SSD or the SSM. 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 SSM 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 USER. If a scan schedule is upated for a particular SSD, it is then replicated down to that SSD.

* 1. **SSM <−> SSD**
     1. Authentication and Registration

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

* + 1. Status and Discovery

The Status and Discovery process serves two functions. The first is to inform the SSM of what capabilities that the SSD 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 SSD 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 SSM. It will transmit a period heartbeat to indicate it is still associated with the SSM. 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).

* + 1. Scan Request

The Scan Request message originating from the SSM is sent to the appropriate SSDs for execution as a scan schedule. It includes the parameters of the desired spectrum measurement to be made based on knowledge of the SSD’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 repitition.

* 1. **DBstore <−> USER**
     1. Authentication and Registration

These procedures define the association and authentication process for a DBstore and USER entity to connect and communicate. They include facilities to prevent spoofing based on shared key exchange. Once a DBstore is authenticated and registered to a USER, the USER is then authorized to cause data to be delivered to the DBstore, and read that data.

* + 1. Resource Discovery

Resource Discovery is the process of informing the USER of what capabilities that the DBstore has with regard to what types of data can be stored, at what rate, at what access level, and in what quantity can be specified. It may also initiate that association of a particular DBstore with a specific SSM that will be providing the data.

* 1. **SSM <−> DBstore**
     1. Authentication and Registration

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

* + 1. Data PUT

These procedures define and enable the storage of data from the SSM to the DBstore. The successful reception of this data initiates a notification of the initiating USER that requested that data.

1. **Architecture**

Figure 1 illustrates the SCOS architecture comprised of a spectrum sensing devices (SSD) and a spectrum sensor manager (SSM). The general flow of actions is as follows:

* User queries the SCOS platform for a list capabilities and then requests measurements from that list to be performed on a schedule.
* SSM manages the sensor network resources to optimize execution of the user requests. It announces available capabilities, receives user requests, provides commands to sensors, collects data from the sensors, and disseminates data according to distribution policy.
* SSD registers its sensor capabilities with SSM upon registration, executes commands sent by the SSM, and returns measured data with required metadata.

The following subsections describe messaging, models, interfaces/protocols, and policies associated w the constituent parts of the SCOS architecture.

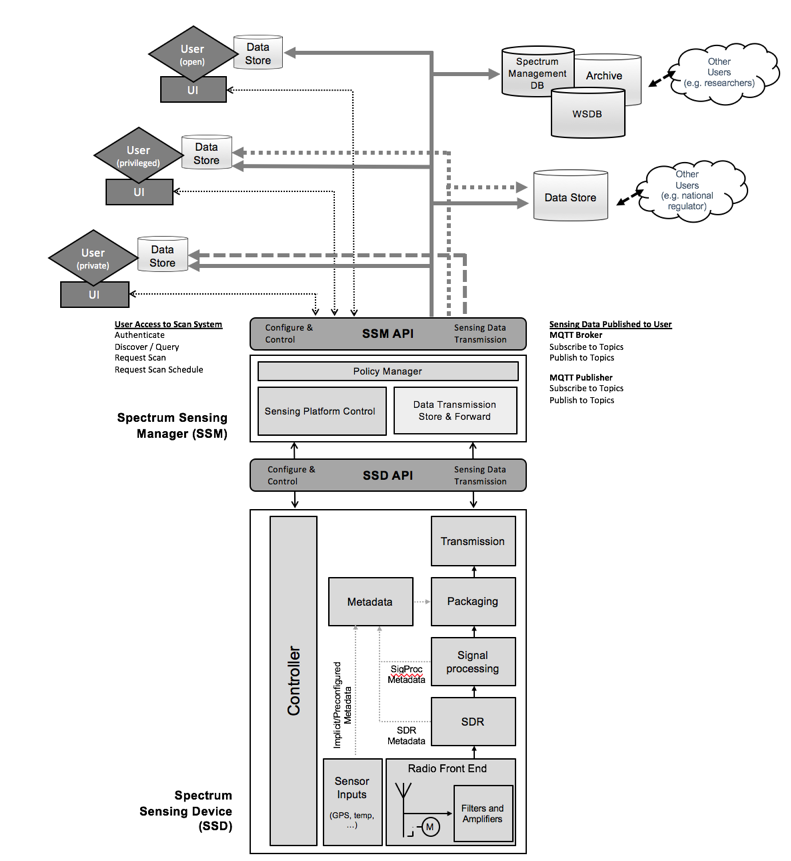


Figure 3. Architecture Block Diagram

* 1. Spectrum Sensing Device (SSD)

SSDs 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.

* + 1. Hardware Model

Figure 2 provides a hardware block diagram of a general SSD model. SSD 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 SSD capabilities are queried by the SSM. This SSD definition metadata will also accompany the output data messages.

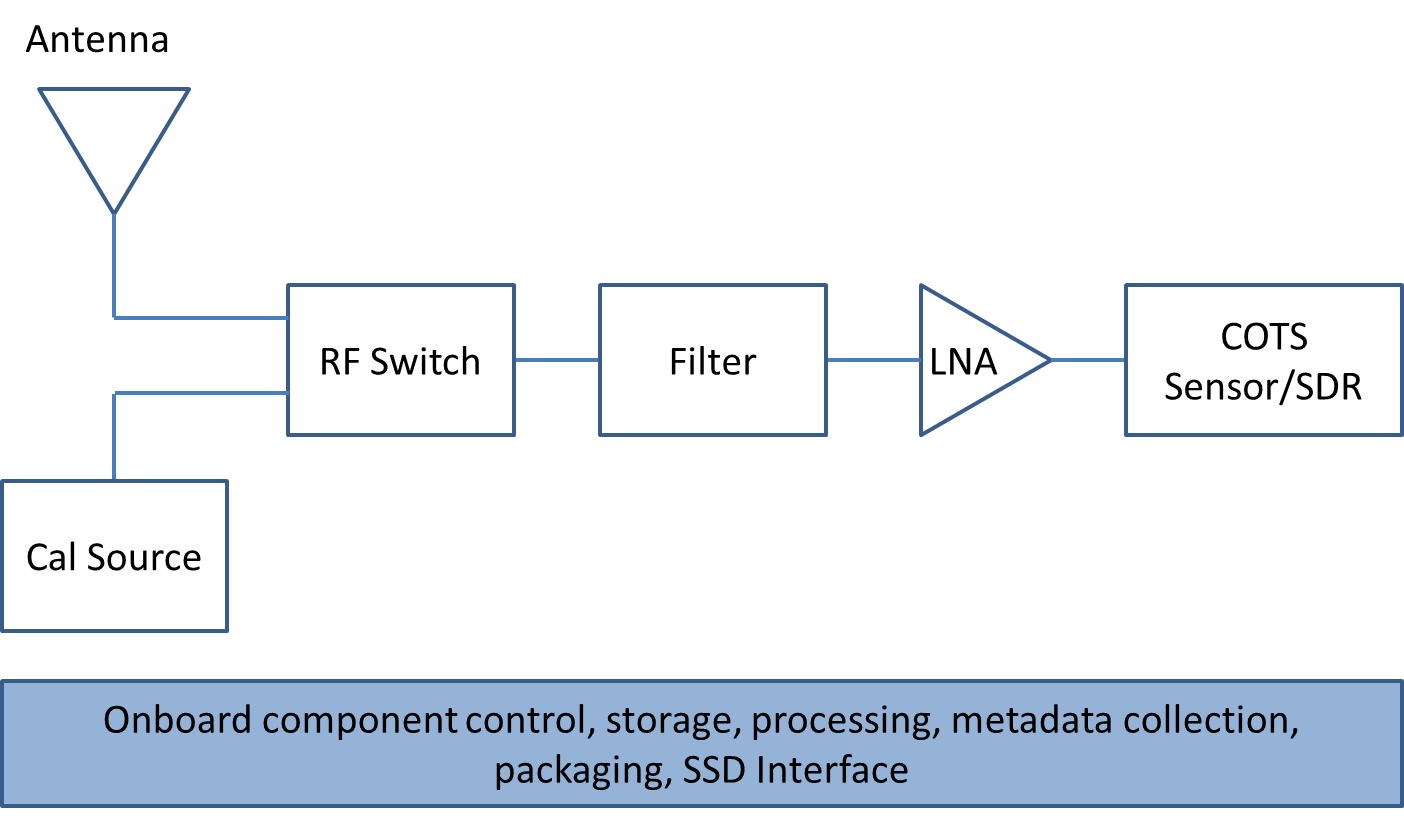


Figure 4. Hardware Model Block Diagram

* + 1. Calibration Model

<Mike to send calibration method details.>

A calibration can be done in the lab at build/commissioning time, and stored as a calibration file on the SSD.

Further, an SSD with a self-calibration capability can be instructed through an administrative interface (not USER request) to perform a calibration using a local calibration source.

* + 1. Algorithm Model

The Algorithm models shall be described in terms of inputs into black box: the identity of the USER and SSM requesting the scan, the measurement parameters, which algorithm is to be used; and outputs from the black box: the identity of the USER and SSM requesting the scan, the requested scan parameters, the identification of the algorithm model, and the processed results.

<Define use cases>

Normative model: general energy detection algortithm

At least one algorithm model is defined – a general energy detection algorithm.

<Trigger event algorithm>

<DF algorighm>

<Cyclostationary>

It is the responsibility of the SSM operator to publish algorithm definitions externally. Code does not need to be publicly accessible.

Allow for the development of advanced algorithms, e.g., DF.

* 1. Spectrum Sensor Manager (SSM)

<Roger’s input>

* + 1. Scheduling

<Mike’s input>

Scheduling is defined in terms of scan intervals that take up slots in a calendar schedule. These slots will include slots for long scans; and slots for very short scans to ensure fair allocation of scan resources.

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

* + 1. Data Distribution

<William’s input>

1. **Messaging**
   1. Format

SSD and SSM messages will be 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 applications.

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". Each message (in general) will begin with a header comprised of attribute-value pairs in ASCII characters. 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)

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.

* 1. SSD Messages

<Apurva’s input>

* 1. SSM Messages

<Roger’s input>

1. **USER definition**

There are at least three main classes of consumers of spectrum data:

Type C Consumer: those that want to read spectrum information from a data store that is already populated with their required data. These users are not contemplated in this standard.

Type B Consumers: have a requirement to keep spectrum information up to date (e.g. spectrum occupancy database operators), and will need to request periodic scans to achieve this.

Type A Consumers: are specifically looking for current sensing information, and request specific scans to obtain specific data (e.g. law enforcement).

In general it should be assumed that a Type A consumer would have higher priority in terms of scan resources, as governed by a Policy expanded on below.

1. **SSM Policy**

<Nilesh input>

A policy layer in the SSM at the northbound and southbound API will ensure that the SSM 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 (SSM>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.

* 1. Policy file: It is envisioned in the first version of this standard that the SSM stores a policy file which is installed manually by Sensing Operator (through mechanism such as SSH and update pull, or remote SCP).
  2. Security Considerations
  3. <Nilesh input>

1. **SSM Administration**

Administrative functions on the SSDs and SSMs are lagely assumed to be implementor-specific, and out of scope for the standard.

Requirements in an Informative Annex would include that there is:

* 1. Administrator interface:

An administrator interface that has a secure mechanism to administer the system, allowing:

* performing calibrations
* updating firmware
* changing configuration of SSD, and making associated changes to the SSD configuration file
* triggering reboots

The administration interface must be a secure interface with key exchange, and these keys must not be the same as keys used for USER<>SSM<>SSD authentication.