

Wireless Innovation Forum Webinar Series

Webinar #20: CBRS CBSD Test and Verification

30 August 2017



Thank You to Our Sponsors

Google  **LEONARDO**

 **MOTOROLA SOLUTIONS**

THALES

Administrivia

Slides presented during this webinar are available in the handouts and will be posted here:

- <http://www.wirelessinnovation.org/webinars>

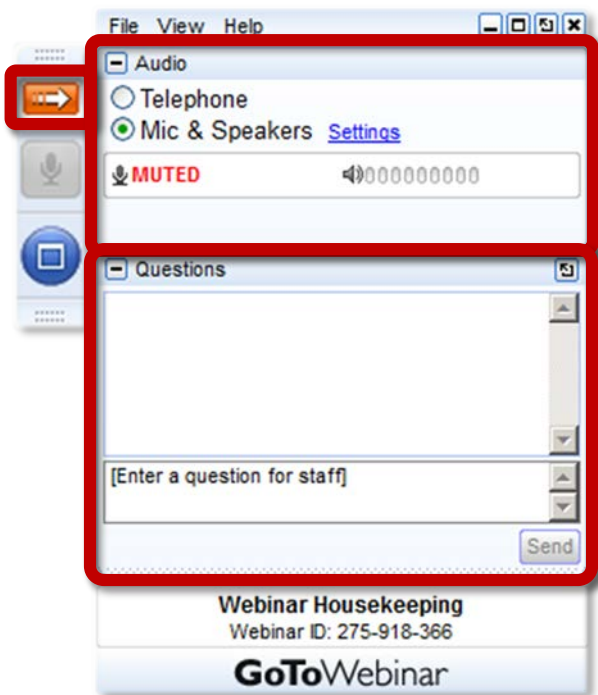
Recorded Webinar will be available on the Forum's You Tube Channel:

- <https://www.youtube.com/channel/UCYUeZvOuJTP27OzoKsys0w>

Email Lee.Pucker@wirelessinnovation.org if you need more information



Go To Webinar Interface



Your Participation

Open and close your control panel

Join audio:

- Choose **Mic & Speakers** to use VoIP
- Choose **Telephone** and dial using the information provided

Submit questions and comments via the Questions panel

Note: Today's presentation is being recorded and will be provided within 48 hours.

Today's Speakers

Awaiz Khan, Ruckus Wireless (Chair of the CBSD Test Task Group)

Doug Goedken, Nokia

Idan Raz, AirSpan

Chris Williams, Ericsson



Wireless Innovation Forum CBSD Test & Certification Webinar

WG4 CBSD
Ver 1.0.0
30-Aug-2017



Webinar Outline

- 1. Introduction: Overview of CBSD test strategy, in light of FCC Part 96 requirements**
 - FCC requirements & how CBSD testing meets those requirements
 - Overview of WInnForum development activity for CBSD certification: Test spec, Test Code, WInnForum Approved CBSD Test & Certification program
- 2. Test strategy:**
 - Description of certification strategy
 - In-scope / out-of-scope for WInnForum testing
 - Handling of Domain Proxy
 - Test case overview and test structure
 - Test code architecture and test code validation
- 3. Example test case from test specification**
- 4. Walk-through of test case with Mock-SAS code**
- 5. WInnForum Approved Test & Certification program details**



FCC Requirements

FCC Title 47 - Chapter I - Subchapter D - Part 96

Title 47: Telecommunication

PART 96—CITIZENS BROADBAND RADIO SERVICE

Contents

Subpart A—General Rules

- §96.1 Scope.
- §96.3 Definitions.
- §96.5 Eligibility.
- §96.7 Authorization required.
- §96.9 Regulatory status.
- §96.11 Frequencies.
- §96.13 Frequency assignments.

Subpart B—Incumbent Protection

- §96.15 Protection of federal incumbent users.
- §96.17 Protection of existing fixed satellite service (FSS) earth stations in the 3600-3700 MHz Band and 3700-4200 MHz Band.
- §96.19 Operation near Canadian and Mexican borders.
- §96.21 Protection of existing operators in the 3650-3700 MHz Band.

Subpart C—Priority Access

- §96.23 Authorization.
- §96.25 Priority access licenses.
- §96.27 Application window.
- §96.29 Competitive bidding procedures.
- §96.31 Aggregation of priority access licenses.
- §96.32 Priority access assignments of authorization, transfers of control, and leasing arrangements.

Subpart D—General Authorized Access

- §96.33 Authorization.
- §96.35 General authorized access use.

Subpart E—Technical Rules

- §96.39 Citizens Broadband Radio Service Device (CBSD) general requirements.
- §96.41 General radio requirements.
- §96.43 Additional requirements for category A CBSDs.
- §96.45 Additional requirements for category B CBSDs.
- §96.47 End user device additional requirements.
- §96.49 Equipment authorization.
- §96.51 RF safety.

Subpart F—Spectrum Access System

- §96.53 Spectrum access system purposes and functionality.
- §96.55 Information gathering and retention.
- §96.57 Registration, authentication, and authorization of Citizens Broadband Radio Service Devices.
- §96.59 Frequency assignment.
- §96.61 Security.
- §96.63 Spectrum access system administrators.
- §96.65 Spectrum access system administrator fees.
- §96.66 Spectrum access system responsibilities related to priority access spectrum manager leases.

Subpart G—Environmental Sensing Capability

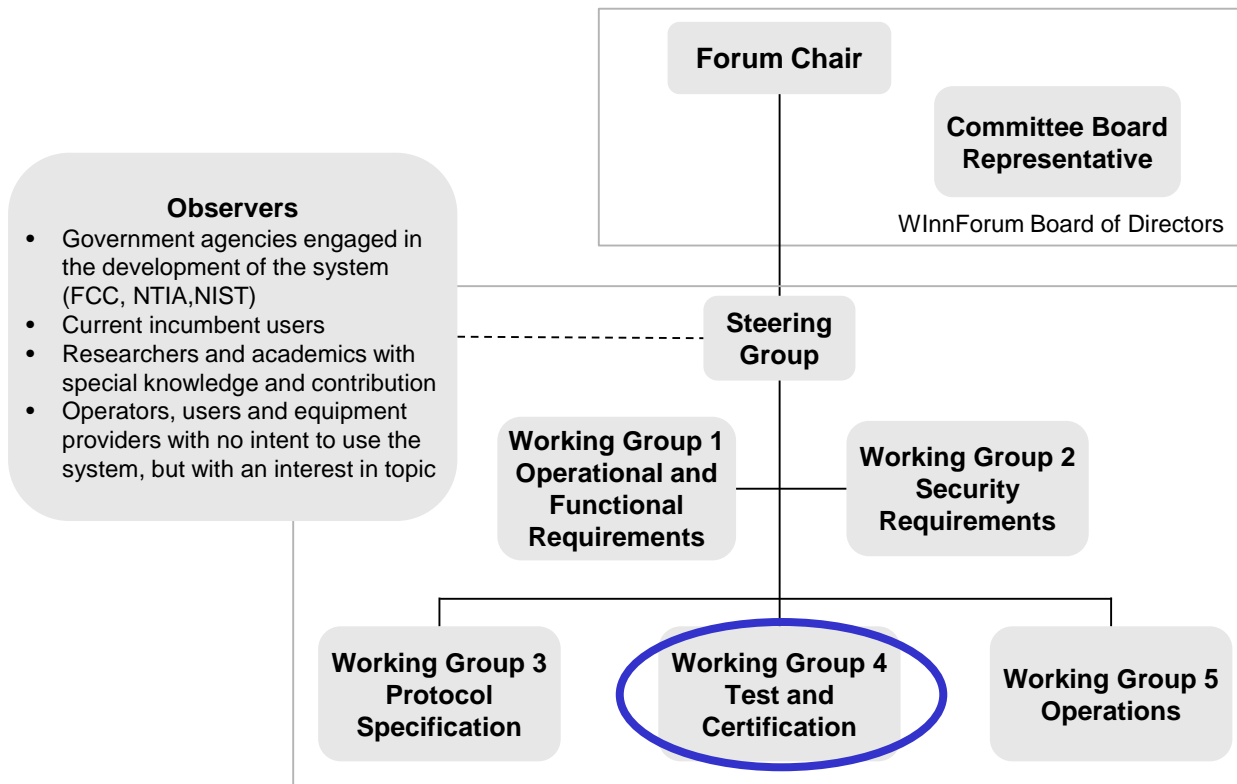
- §96.67 Environmental sensing capability.

AUTHORITY: 47 U.S.C. 154(i), 303, and 307.

SOURCE: 80 FR 36222, June 23, 2015, unless otherwise noted.

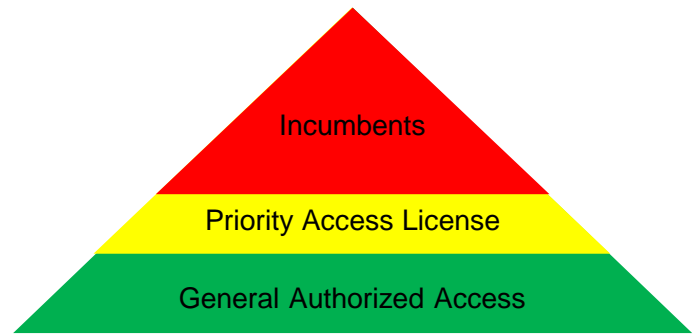
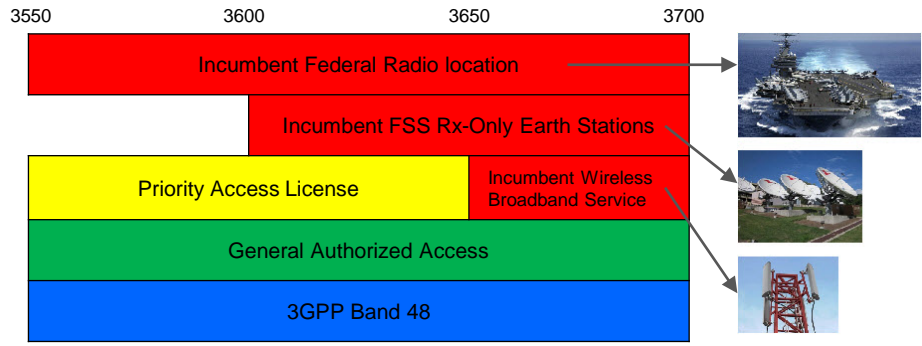


WinnForum Spectrum Sharing Committee Structure

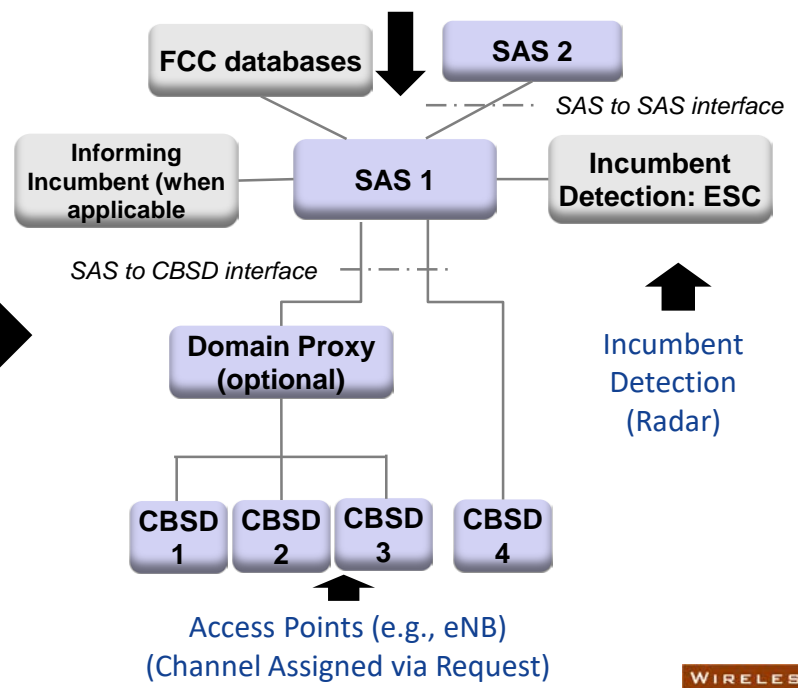


Spectrum Sharing multi-shareholder committee
<http://www.wirelessinnovation.org/ssc>

Architecture and Framework created to ensure Incumbent Protection



Incumbent Interference Determination CBRS-Channel Assignment



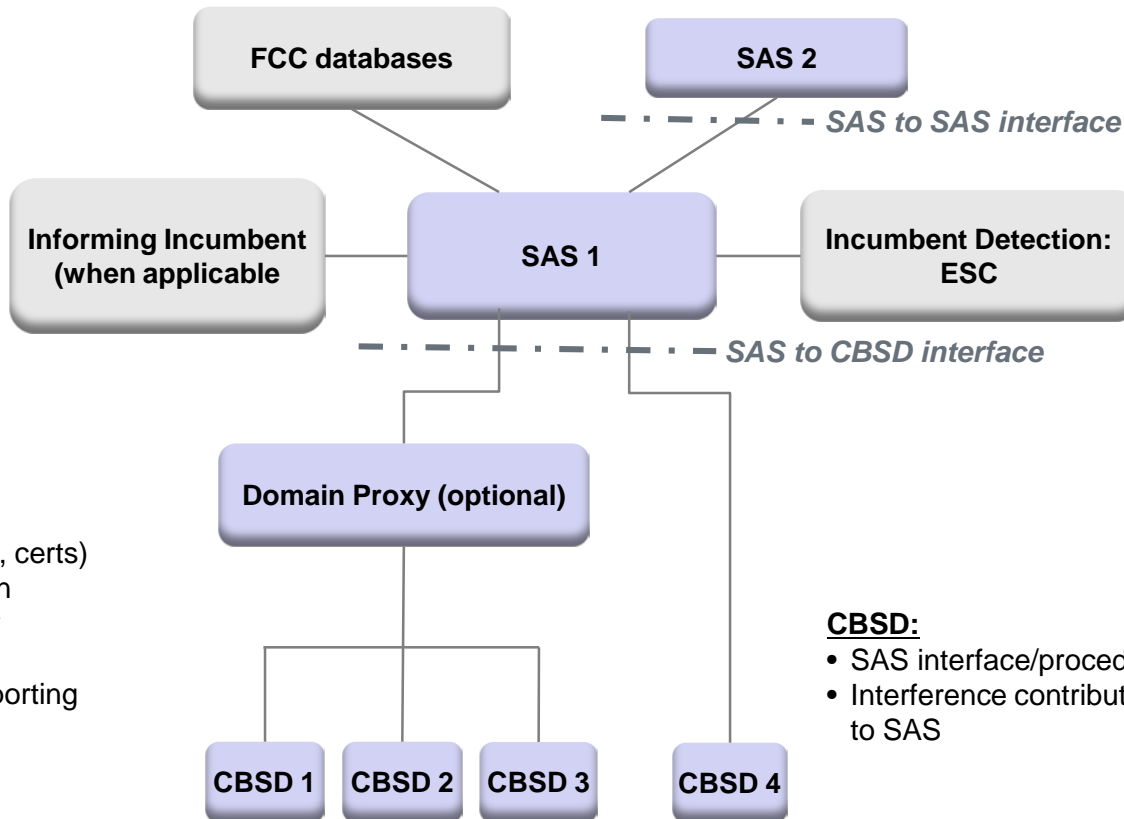
CBRS SAS and CBSD – Architecture

SAS:

- CBSD Registration
- Interference analysis
- Incumbent protection
- PAL license validation
- CBSD channel assignment
- CBSD power limits
- PAL protection
- SAS-to-SAS coordination

Domain Proxy:

- SAS interface security (TLS, certs)
- CBSD message aggregation
- CBSD interface security per operator requirements
- Interference contribution reporting to SAS

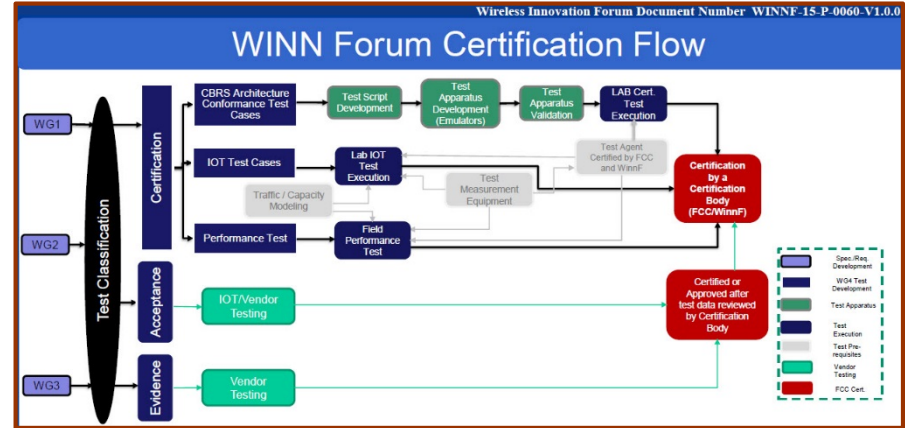


CBSD:

- SAS interface/procedure operation
- Interference contribution reporting to SAS

Documents pertaining to CBSD testing

WINNF-15-P-0060 SSC WG4 Certification Process



WINNF-17-RFI-00122-V1.0.0 CBRS CBSD Test Specification Request for Comment

- WG4 develops and publishes the test cases.
- Functional tests to address requirements from FCC and /or WG1, WG2, WG3
- Testing the protocol (messaging) exchanged between the CBSD/DP and SAS
- Test cases leveraged to create test scripts/code for facilitating execution of the test cases by the test labs.

Published documents available at: <https://workspace.winnforum.org/higherlogic/ws/public>

Standards that comprise Release 1

WINNF-TS-0112-V1.3.0 CBRS Operational and Functional Requirements (in ballot)
WINNF-TS-0065-V1.1.0 CBRS Communications Security Technical Specification
WINNF-TS-0071-V1.0.0 CBRS Operational Security Technical Specification
WINNF-TS-0016-V1.1.0 SAS to CBSD Protocol Specification
WINNF-TS-0096-V1.1.0 SAS to SAS Protocol Specification
WINNF-TS-0061-V1.1.0 SAS Test and Certification Specification (in finalization)
WINNF-TS-0122-V1.0.0 CBSD Test and Certification Specification (in ballot)
WINNF-TS-0245-V1.0.0 PAL Database Specification
WINNF-TS-0022-V1.0.0 CBRS PKI Certificate Policy
WINNF-17-SSC-0002-V2.0.1 WinnForum Recognized CBRS Air Interfaces and Measurements
WINNF-TR-0205 CBRS Protocols Technical Report

Published documents available at:

- <https://workspace.winnforum.org/higherlogic/ws/public>



http://www.wirelessinnovation.org/

WIRELESS INNOVATION FORUM

Home | Knowledge Center | Join | About Us | Events | News | Projects & Committees | Issue Submissions

Search our site...

Member Login

WIRELESS INNOVATION FORUM

PUBLIC HELP LOGIN SIGNUP

Public

Member Login

Username

Password

Login

I forgot my password

WIRELESS INNOVATION FORUM

Driving the future of radio communication and systems worldwide

PUBLIC HELP LOGIN SIGNUP

Groups

Home Documents Calendar Ballots

Public Documents

Public Groups Documents

Filters

Search by Name

Groups

All Groups

All Documents

Name	Rev #	State	Submitter	Group	Date	Actions
WINNF-TS-0096-V1.1.0 SAS...	1	Final	Pucker, Lee	(SSC) Sp...	82 2017-0...	
WINNF-TR-0205-V1.0.0 CBR...	-	Final	Pucker, Lee	(SSC) Sp...	45 2017-0...	
WINNF-TS-0022-v1.0.0 CBR...	1	Final	Pucker, Lee	(SSC) Sp...	145 2017-0...	

Filtering options available for finding specific file.

What is addressed?

CBSD, Domain Proxy and SAS

- Citizens Broadband radio Service Device (CBSD): entity communicating with the SAS and meets the general requirements applicable to all CBSDs and the specific requirements set forth in sections 96.41, 96.43 and 96.45
- Domain Proxy: entity communicating with the SAS on behalf of multiple individual CBSDs or networks of CBSDs.
- Spectrum Access System (SAS): authorizes and manages use of spectrum for the Citizens Broadband Radio Service in accordance with Title 47 CFR Part 96 subpart F.

CBSD Registration Processing

- How a CBSD registers with a SAS, including owner registration, professional installer registration and CBSD registration.

Spectrum Grant Request Processing

- How a CBSD requests and relinquish grants, and how grants are reassigned or terminated.

CBSD Measurement Reporting

- CBSD measurements of their local interference environment, and reporting data back to the SAS.

Certified Professional Installer Training Program Approval Standard

- Guidelines for adoption of uniform industry working standards and curriculum required to be consistent with the protection of spectrum, both licensed and GAA, for sharing in the 3550-3700 MHz band.

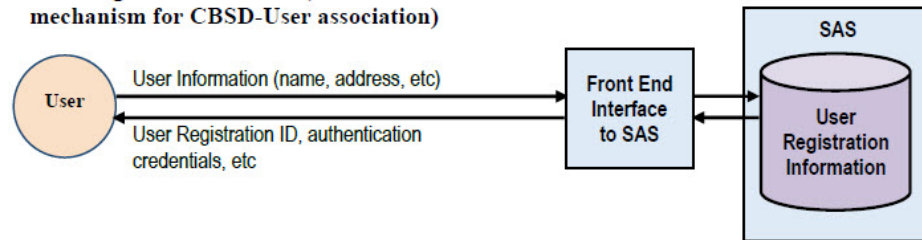
Communications Security

- The communications security policies governing SAS and CBSD communications interfaces.

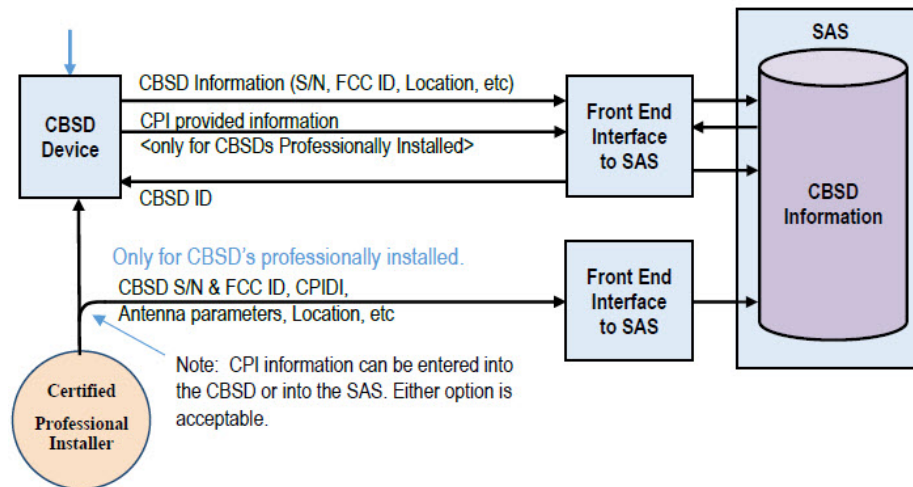
CBSD needs to be registered

- **Before a CBSD can begin channel allocation requests with the SAS, the CBSD must be registered with the SAS.**
- **User Registration**
 - CBSDs must be associated with a user
 - The user is required to pre-register (enroll) with the system
 - The user may be the owner, the deployer, or the manager of the CBSD(s).
- **CBSD Registration**
 - The CBSD must register with the SAS and provide installation details.
 - The CBSD is either user installed or installed by a Certified Professional Installer (CPI)

User Registration Process (includes a secure mechanism for CBSD-User association)



User Entered Data



Who can install the CBSD?

- **User installed CBSDs:**

- Category A CBSDs can be installed by a user or an authorized associate or employee of the user.
- To be user installed, the Category A CBSD must:
 - Be capable of automatically determining its location (GPS) and provide the location information as part of the CBSD registration process
 - Operate at the maximum power level, or lower, specified for a Category A device (30dBm/10MHz EIRP)
 - Have an antenna height <6 meters above average terrain, when installed outdoors

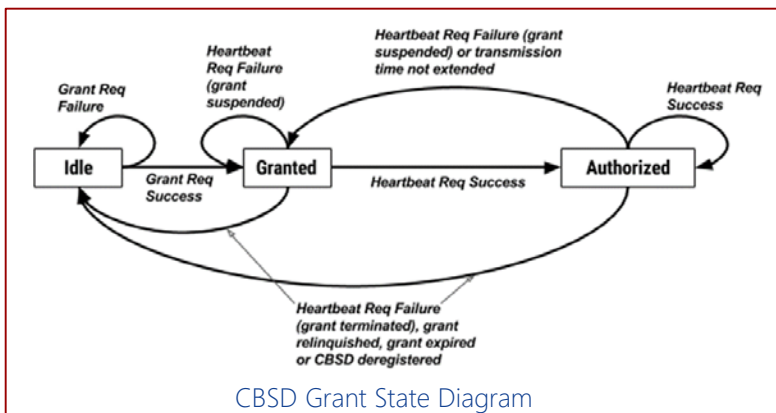
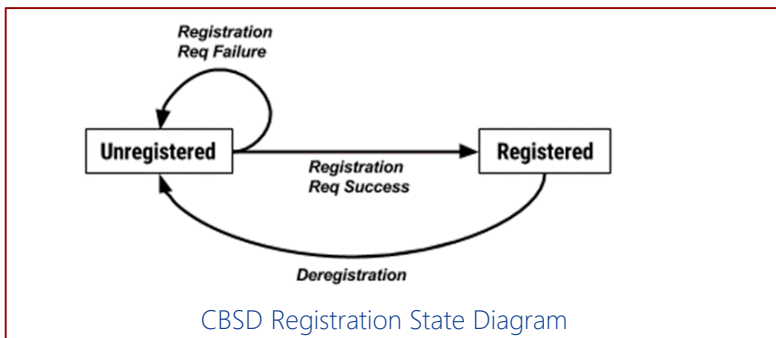
- **Professionally (CPI) installed CBSDs:**

- All Category B CBSDs.
- Category A CBSDs unable to automatically determine their location (GPS) accurately enough to meet the FCC Rules.
 - For instance, not capable of automatic determination of their location, either by design or due to disadvantaged placement.
- Category A device located outdoors with an antenna height >6 meters above average terrain
- Any Category A CBSD requiring antenna information (gain, pointing direction).

- CPIs must pass a certification program to be registered in a centralized database, accessible by all SASs. This will provide the installer with a system wide unique CPI Identifier and a method to authenticate the Installer when accessing the CPI account.

CBSD state transitions

From “WINNF-TS-0016-V1.1.0 SAS to CBSD Technical Specification”



- **SAS/CBSD operation is managed by two state machines**

- Both state machines reside in CBSD and SAS

- **CBSD Registration state machine**

- CBSD begins in an Unregistered state and enters the Registered state once all of the required registration parameters have been provided to the SAS (either directly or entered via an SAS interface).
- Only one instance of registration state machine
- CBSD must be in Registered state before requesting a frequency allocation from SAS
- CBSD may deregister itself or be deregistered by the SAS for a variety of reasons.

- **CBSD Grant state machine**

- Grant is the mechanism used by the SAS/CBSD to authorize a CBSD to operate over a certain frequency range at a certain power level for a certain period of time
- Multiple instances of grant state machine
- Enables multi-band/carrier CBSDs
- Grant State machine manages CBSD requests, grant allocations, grant maintenance, frequency reassignments, suspension/termination due to incumbent detection, grant relinquishment
- CBSD radio is only transmitting in the Authorized state

SAS-CBSD Procedures

1. **SAS Discovery Procedure**
2. **Authentication Procedure**
3. **CBSD Registration Procedure**
4. **CBSD Spectrum Inquiry Procedure**
5. **CBSD Grant Procedure**
6. **CBSD Heartbeat Procedure**
7. **CBSD Grant Relinquishment Procedure**
8. **CBSD Deregistration Procedure**

Each procedure consists of a request from the CBSD and a response from the SAS.

SAS-CBSD protocol is based on the HTTPS (HTTP over TLS)

SAS-CBSD Procedures (1 of 2)

SAS Discovery Procedure

- SAS provides services using IPv4, optionally IPv6
- SAS Administrators provide URLs for the CBSDs/Domain

Authentication Procedure

- TLS-v1.2 is used to perform authentication
- CBSD/Domain Proxy initiating the TLS connection authenticates with the SAS, and the SAS authenticates with the CBSD/Domain Proxy
- All message exchanges involving communication between a CBSD/Domain Proxy and the SAS are performed in an established TLS connection

CBSD Registration Procedure

- CBSD indicates to a SAS its intention to operate.
- CBSD provides a fixed location, unique identifiers (e.g., owner information, device information), Group membership, and radio-related capabilities.
- A successful registration procedure concludes with the SAS providing a unique identifier for that CBSD.
- Successful registration implies a validation by the SAS that the CBSD has been FCC certified and confers on the CBSD the right to be authorized by the SAS to operate in accordance with a Grant.

SAS-CBSD Procedures (2 of 2)

CBSD Spectrum Inquiry Procedure

- Optional procedure that the CBSD may initiate any time after receiving a cbsdId
- Allows Registered CBSDs to request information on available channels

CBSD Grant Procedure

- Authorization provided by a SAS to a CBSD, subject to a Heartbeat exchange, to transmit using specified operating parameters.
- Once issued, a Grant's operating parameters (e.g. EIRP, channel) are not changed; if new or modified operating parameters are required, a new Grant must be obtained.

CBSD Heartbeat Procedure

- CBSD informs the SAS its need to access the allocated spectrum for the specified grantId
- CBSD initiates this procedure any time prior to the expiration of the Heartbeat Interval timer.
- SAS is allowed to suspend or terminate the Grant

CBSD Grant Relinquishment Procedure

- The CBSD terminates radio operation associated with a specific Grant.

CBSD Deregistration Procedure

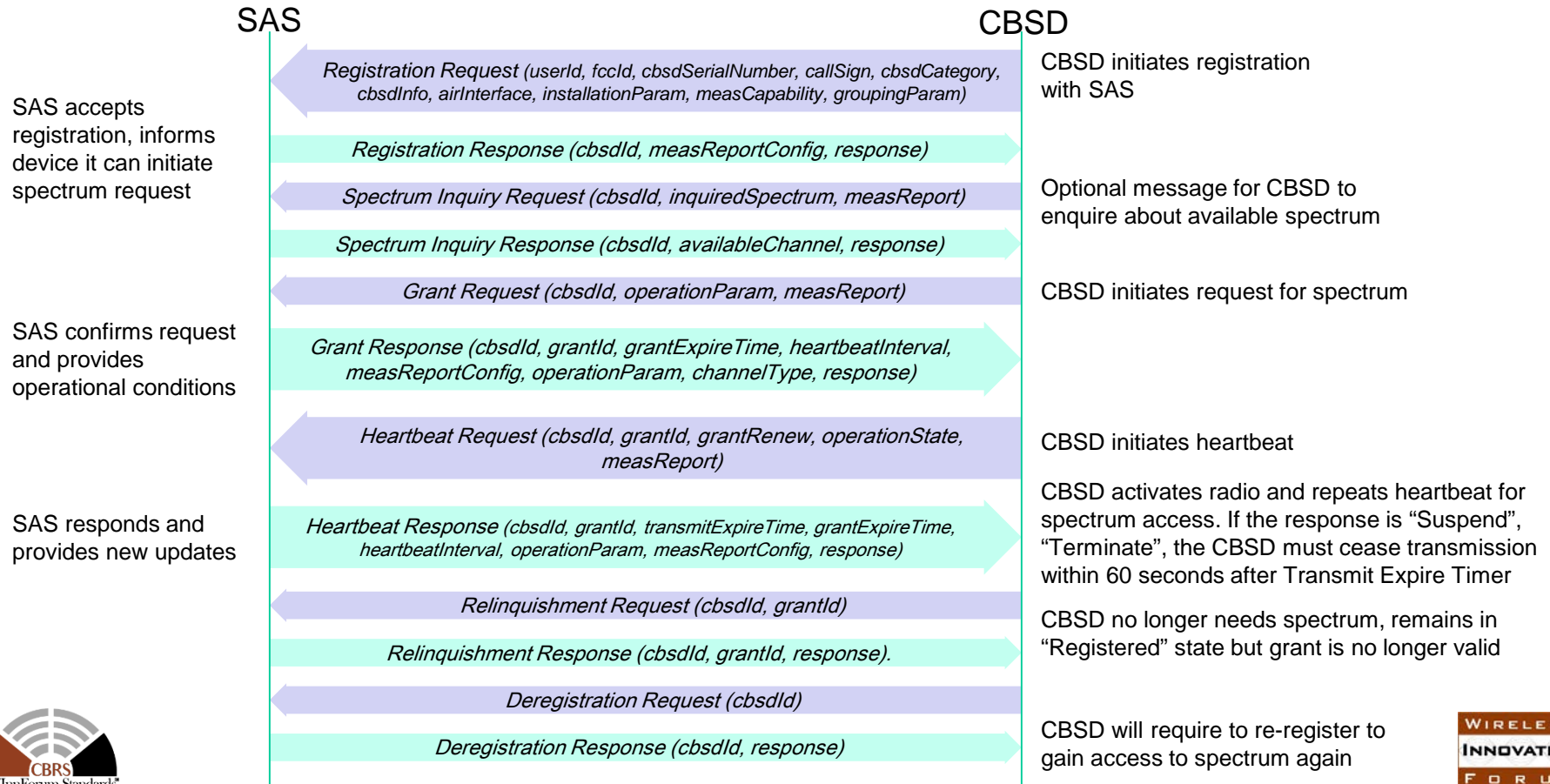
- When the CBSD determines it should deregister from the SAS, it shall cease transmission associated with any Grants and then send a DeregistrationRequest
- The SAS marks the CBSD as Unregistered, removes any existing Grants.

SAS-CBSD interface message summary

Messages apply to both CBSD and Domain Proxy

Procedure	Message	Description	Outcome
CBSD Authentication	TLS messages	CBSD establishes a TLS session with SAS	CBSD can initiate a registration
CBSD Registration	Registration Request Registration Response	CBSD provides information to the SAS for purpose of registering. CBSD must register prior to requesting access for spectrum	CBSD is registered with the SAS
CBSD Spectrum Inquiry	Optional Spectrum Inquiry Request Spectrum Inquiry Response	CBSD asks the SAS what spectrum may be available	CBSD obtains information that will assist it in requesting spectrum
CBSD Grant	Grant Request Grant Response	CBSD request access to an amount of spectrum and SAS will respond	Upon successful response, the CBSD can operate its radio
CBSD Heartbeat	Heartbeat Request Heartbeat Response	CBSD informs SAS it is still using the allocated spectrum. SAS can respond whether CBSD can continue, suspend operation or terminate operation	Depending on the response, the CBSD can continue operating or needs to stop transmitting on specified Grant
CBSD Spectrum Relinquishment	Relinquish Request Relinquish Response	CBSD informs SAS it no longer needs to access the spectrum associated with the given Grant	CBSD stops using the spectrum
CBSD Deregistration	Deregistration Request Deregistration Response	CBSD informs the SAS its request to have its registration deleted	CBSD is no longer registered with the SAS. It is not transmitting.

SAS-CBSD interface basic operation per WInnForum specifications



Response Codes

- 0: success
- 100 – 199: general errors related to the SAS-CBSD protocol
- 200 – 299: error events related to the CBSD Registration procedure
- 300 – 399: error events related to the Spectrum Inquiry procedure
- 400 – 499: error events related to the Grant procedure
- 500 – 599: error events related to the Heartbeat procedure

<i>responseCode</i> Value	Name
0	SUCCESS
100	VERSION
101	BLACKLISTED
102	MISSING_PARAM
103	INVALID_VALUE
104	CERT_ERROR
105	DEREGISTER
200	REG_PENDING
201	GROUP_ERROR
300	UNSUPPORTED_SPECTRUM
400	INTERFERENCE
401	GRANT_CONFLICT
500	TERMINATED_GRANT
501	SUSPENDED_GRANT
502	UNSYNC_OP_PARAM

Message Encoding and Transport

- HTTPS (HTTP+TLS) is used for communication between servers (SAS) and clients (CBSD)
- A SAS-CBSD message is encoded using JSON (JavaScript Object Notation)
- A parameter value can be one of the primitive JSON data types, i.e., string, number, boolean, array, or object.
- If a parameter is an object, a name for the object is given and a separate table describes parameters in the object.
- Each parameter is indicated as “Required”, “Optional” or “Conditional”:
 - “Required”: Parameter shall always be included in the message.
 - “Optional”: Parameter may be included in the message.
 - “Conditional”: Parameter shall be included in the message, if and only if the specified conditions are satisfied.
- A JSON array in a message can contain a single or multiple requests or responses associated with a CBSD or different CBSDs with a Domain Proxy

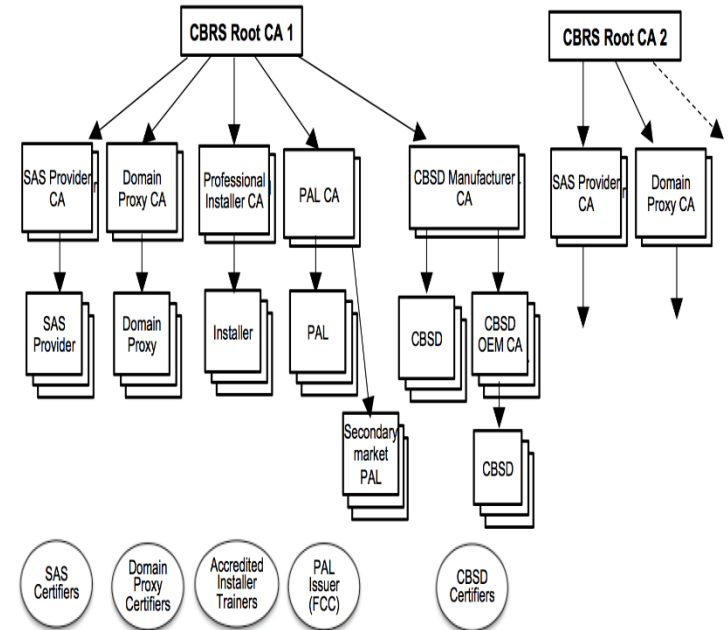
Example JSON-encoded SAS-CBSD message

```
{
  "registrationRequest": {
    "fccId": "abc123",
    "cbbsdCategory": "A",
    "callSign": "CB987",
    "userId": "John Doe",
    "airInterface": {
      "radioTechnology": "E_UTRA"
    },
    "cbbsdSerialNumber": "abcd1234",
    "measCapability": [
      "RECEIVED_POWER_WITHOUT_GRANT"
    ],
    "installationParam": {
      "latitude": 37.419735,
      "longitude": -122.072205,
      "height": 6,
      "heightType": "AGL",
      "indoorDeployment": true
    },
    "groupingParam": [
      { "groupId": "example-group-1",
        "groupType": "INTERFERENCE_COORDINATION" },
      ]
    }
  }
}
```

CBRS Communications Security per WInnForum requirements

From “WINNF-TS-0065-V1.1.0 CBRS Communications Security Technical Specification”

- The WInnForum COMSEC Technical Specifications defines all the communications security polices required to provide Authentication, Authorization and Encryption for messages exchanged within the whole end to end CBRS eco-system based on industry well known protocols and infrastructures (TLS and PKI).
- Our Current CBSD security design is based on the COMSEC TS.
- CBSDs use TLS1.2 to establish Authenticated and Encrypted communications channel with the SAS.
- This approach allow each CBSD to have its own encrypted channel to the SAS.
- This TLS connection secures only the SAS-CBSD interface. Independent security measures and techniques are needed to secure other interfaces like LTE S1 Control , User and management interfaces.
- CBSDs and SASs digital certificates are issued by dedicated Certificate Authorities certified by a CBRS Root CAs designated by the Industry.



CBSD Test Case / Code Development Milestones

Date	Deliverable
25 Aug	<ul style="list-style-type: none">WINNF-17-RFI-00122-V1.0.0 CBRS CBSD Test Specification Request for CommentWINNF-TS-0122-V1.0.0 CBSD Test and Certification Specification ballot start
30 Aug	Webinar: CBSD Testing - Overview, Strategy and Demonstration
14 Sep	FCC OET comments on RFI
15 Sep	Test Code Development Complete
22 Sep	WINNF-TS-0122-V1.0.0 CBSD Test and Certification Specification Ballot resolution begin
15 Oct	Test Code Verification Complete
16 Oct	Test Code delivery to OET/Test Labs
26 Oct	WINNF-TS-0122-V1.0.0 CBSD Test and Certification Specification Ballot resolution complete
16 Nov	Test Labs Certification Ready
16 Dec	First PAG TCB Enquiry to OET

CBSD Test Strategy



WinnForum CBSD Test Strategy

KDB 940660 D01 Part 96 CBSD v01 outlines two aspects required for certification under Part 96:

1. Demonstrating compliance to specific RF requirements and limits
2. Demonstrating compliance to rules governing the connection and interaction between the CBSD and SAS

WinnForum addresses second aspect only, verifying:

- Proper implementation of the SAS-CBSD protocol, according to WinnForum specification, and
- Proper CBSD behavior in interactions with the SAS

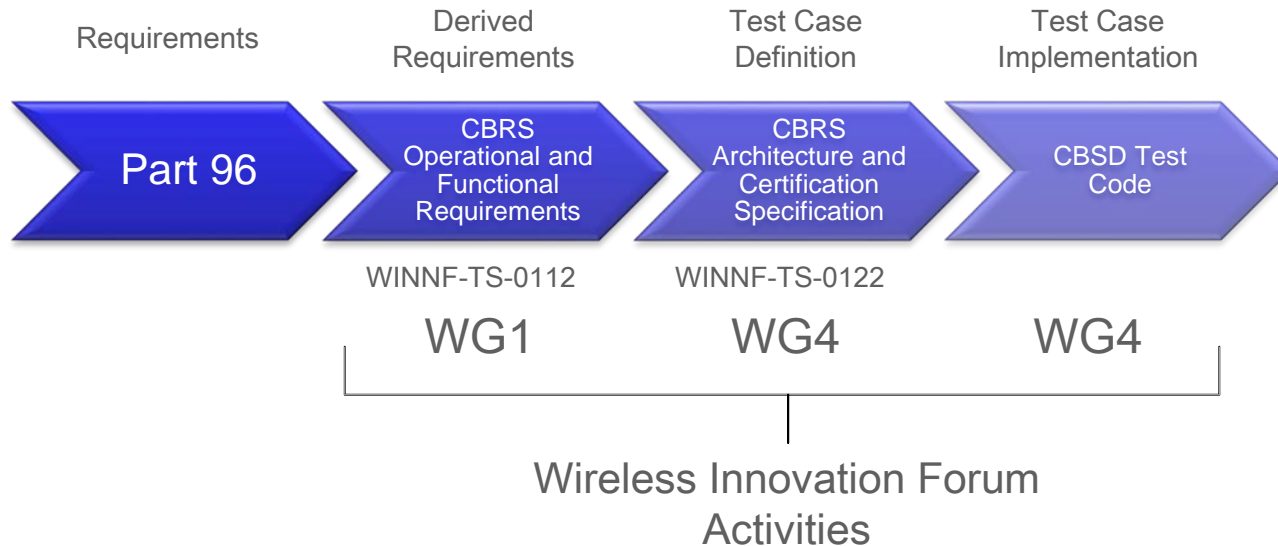
WinnForum test report demonstrates CBSD compliance to all rules governing the connection and interaction with SAS.

WinnForum SSC WG4 will provide the following deliverables to aid with FCC Certification of CBSDs:

1. Test specification to demonstrate compliance to SAS-CBSD protocol and CBSD behavior, as required by WinnForum specifications
2. Test code implementation of CBSD Test Harness, emulating SAS to complete test cases from the test specification

Test Case Development Flow

Test case & test code development is ultimately based upon demonstrating compliance to non-RF Part 96 requirements



Test Approach

WInnForum has taken a black-box approach to testing CBSDs:

- CBSD uses normal operational software (not test mode). Minimal additional functionality to support WInnForum test cases.
- Monitoring of external interfaces to verify protocol and behavior:
 - SAS-CBSD interface
 - RF interface
- Testing is air-interface technology agnostic
- RF conformance (EIRP, spectral mask, etc.) left to separate FCC OET procedures

WinnForum CBSD Test Scope

The following items are in-scope for WinnForum CBSD testing:

1. Testing of SAS-CBSD protocol, and CBSD behavior under SAS control
2. Stand-alone CBSD, or CBSD under supervision of Domain Proxy

The following items are out-of-scope for WinnForum CBSD testing:

1. Radio conformance to RF limits are left to normal FCC OET procedures.
2. Specification of particular RF test equipment used in WinnForum CBSD testing

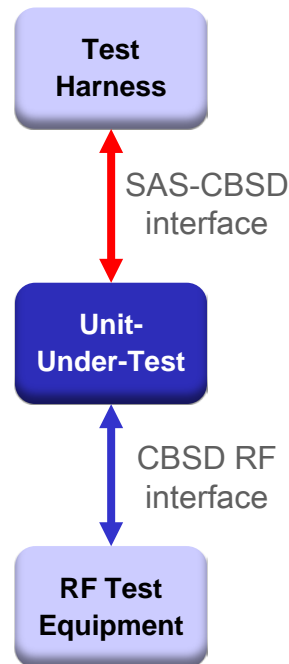
Test Harness and Setup

WinnForum test setup shown at right:

- UUT is CBSD or Domain-Proxy + CBSD
- Test Harness (“Mock-SAS”) simulates SAS to exercise SAS-CBSD interface and controls UUT for each test case
- Test Harness implemented by WinnForum to run all test cases

RF test equipment monitors CBSD RF interface to determine whether:

- UUT is transmitting within its grant
- UUT ceases transmission within requisite time upon certain events
- Does NOT perform calibrated RF measurements



Test Case Overview

- **Test specification provides test cases for each protocol message in the SAS-CBSD protocol:**
 - Registration & Deregistration processes
 - Spectrum Inquiry, Grant, and Relinquishment processes
 - Heartbeat process
 - Measurement Report procedure
- **For each process, success and failure test cases are included, such as:**
 - Suspension or Termination of in-service grant by SAS
 - SAS-CBSD communications failure
- **Specific test cases with multiple CBSD are provided for Domain Proxy, if present.**

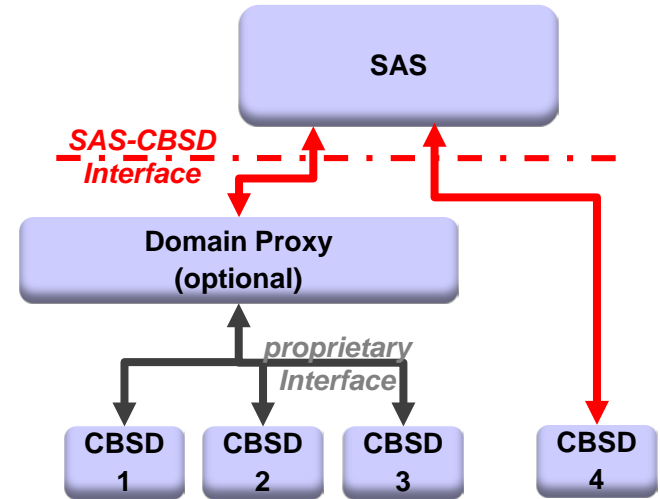
Domain Proxy

The Domain Proxy is a logical entity that can represent one or more CBSD to the SAS

Domain Proxy is an optional element.

If vendor implements:

- Domain Proxy + CBSD must meet Part 96 in same manner as stand-alone CBSD
- Domain Proxy + CBSD are certified together
- CBSD may only operate with Domain Proxy with which it has been certified



Testing With Domain Proxy

WinnForum testing with Domain Proxy:

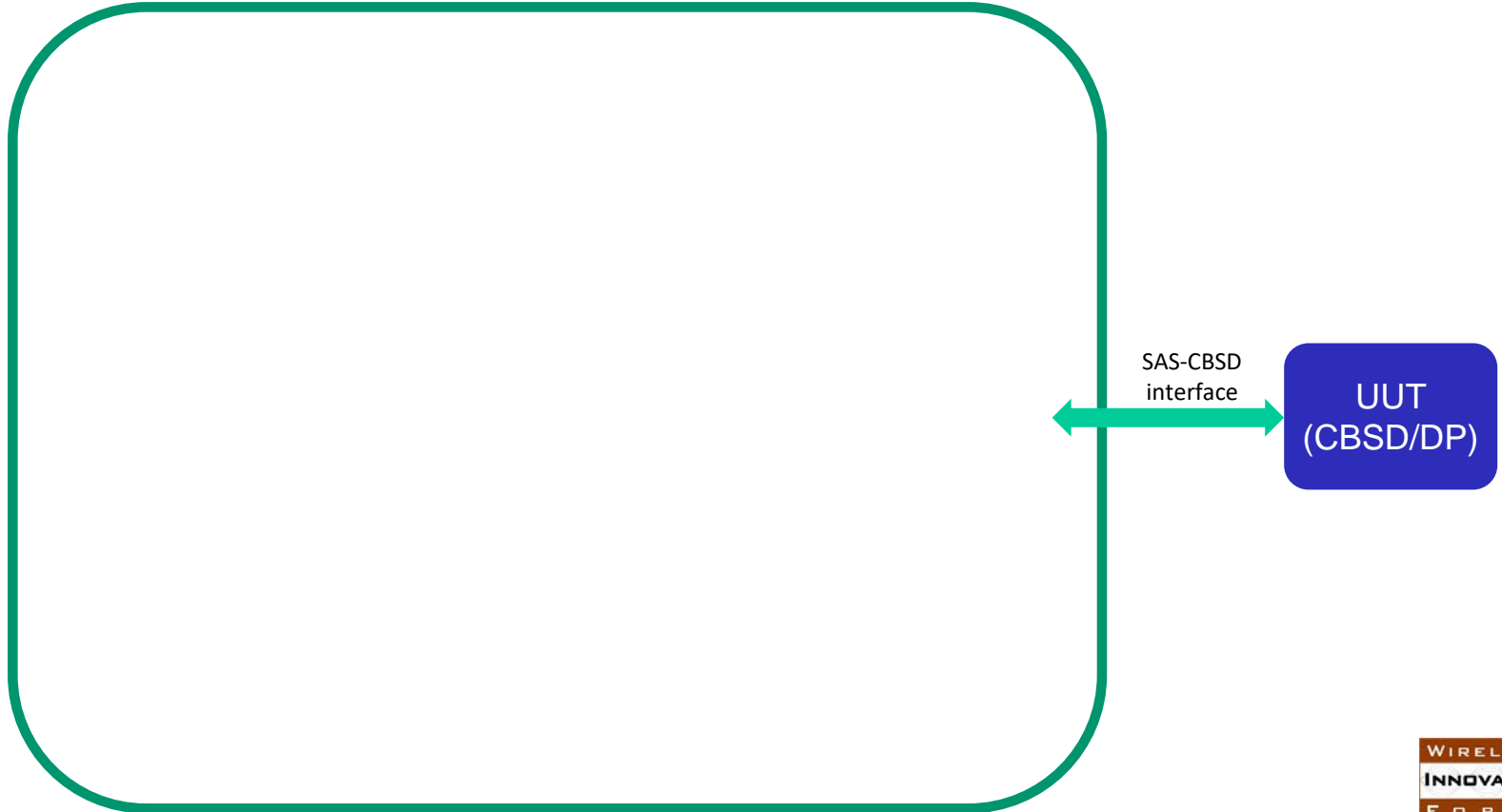
- Test cases for CBSD also apply to Domain Proxy + CBSD
- Some additional test cases specifically address Domain Proxy
- Test harness will support both stand-alone CBSD and CBSD with Domain Proxy
- Test cases for Domain Proxy include 2 CBSD

Test Code Development and Maintenance



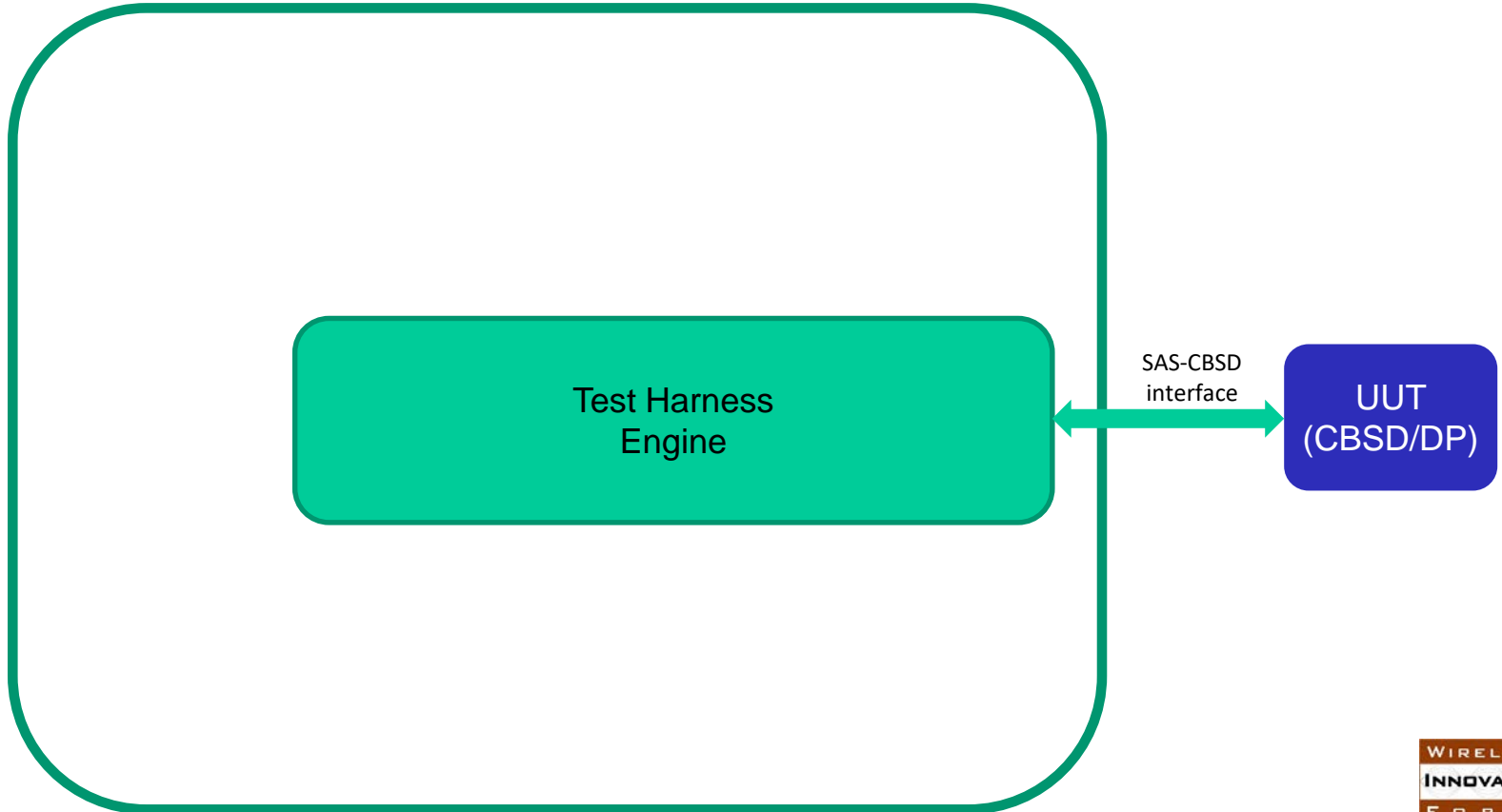
Overall Approach

Test Harness Software



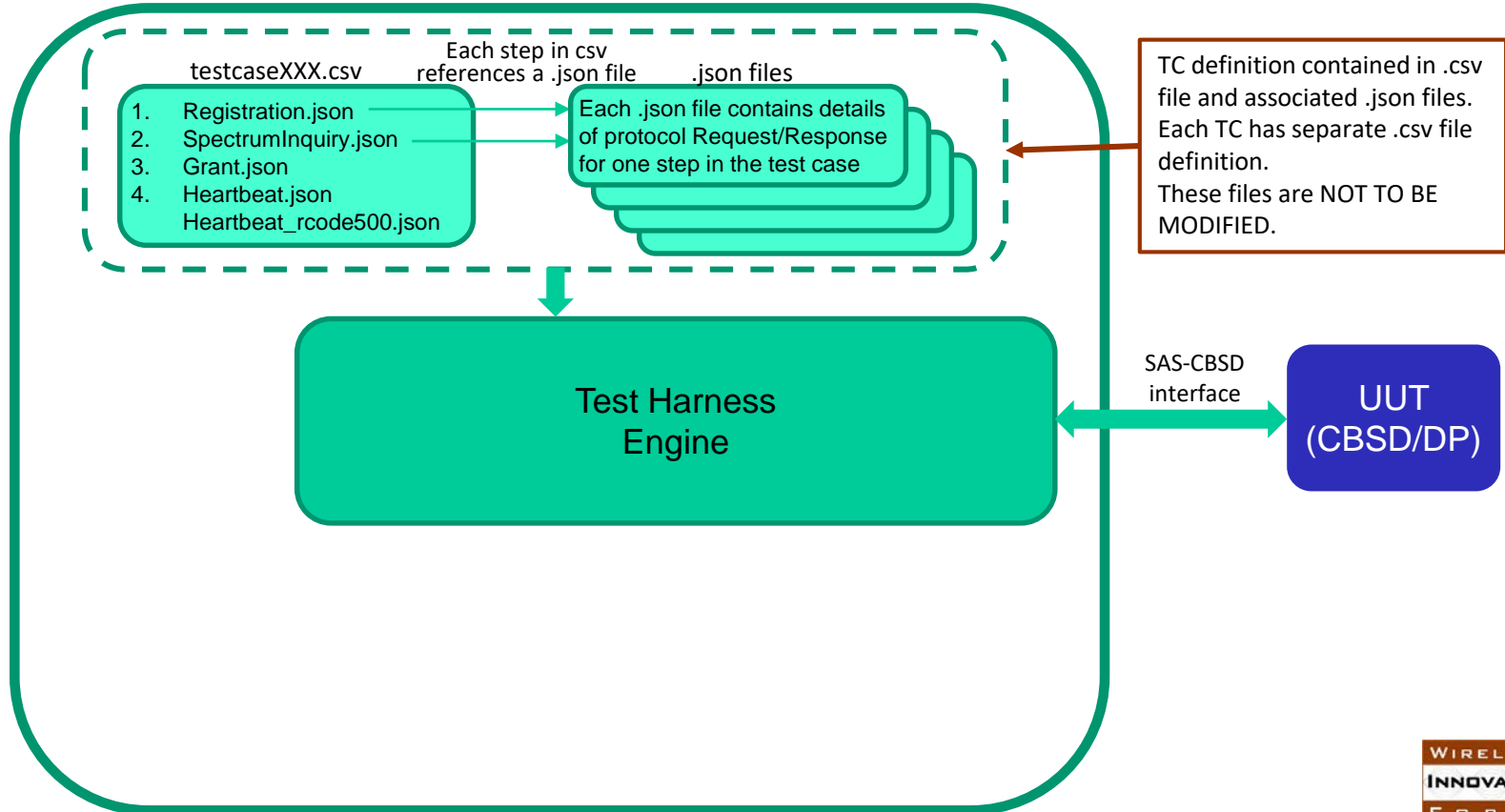
Test Harness Components (1)

Test Harness Software



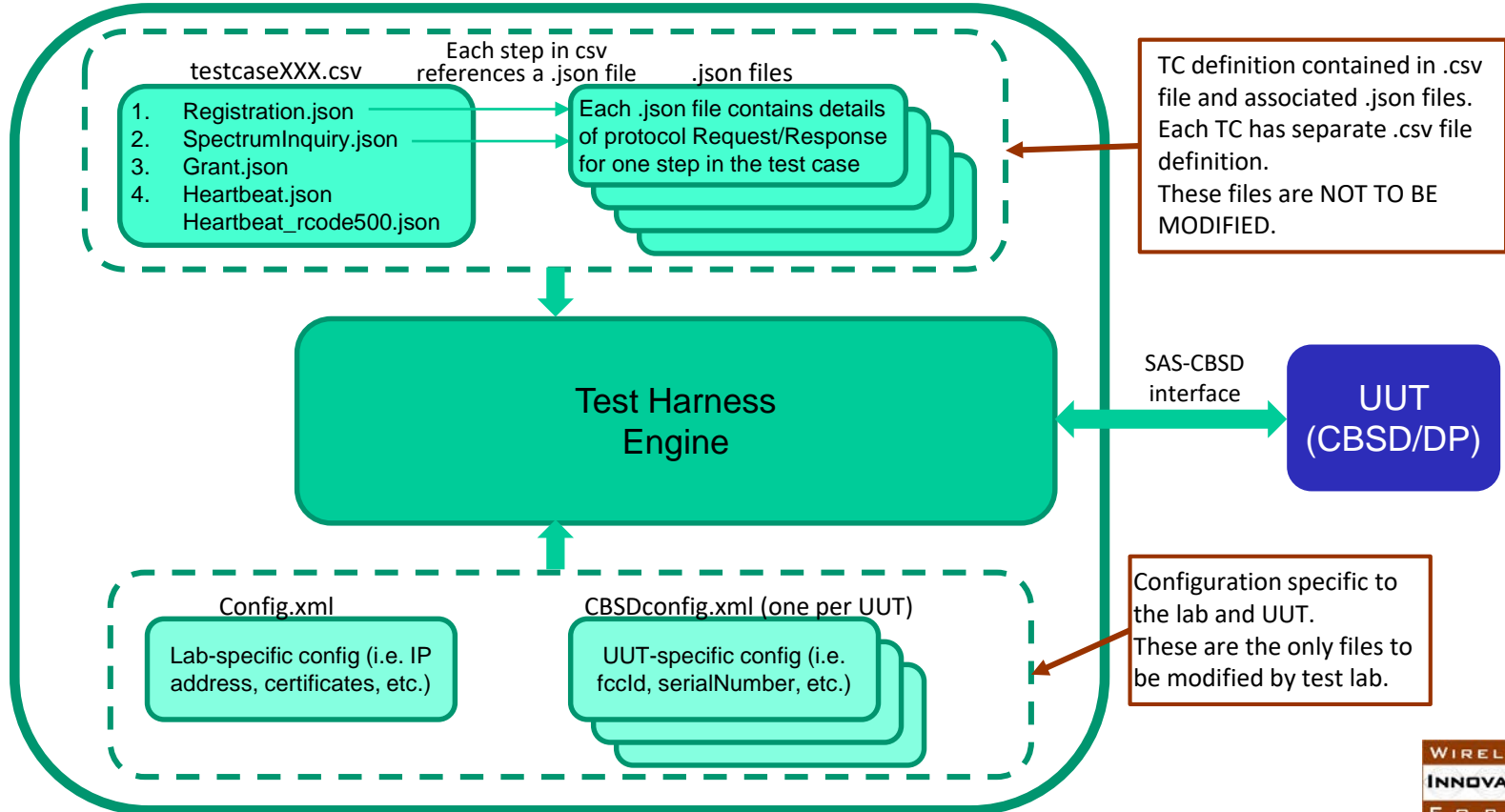
Test Harness Components (2)

Test Harness Software



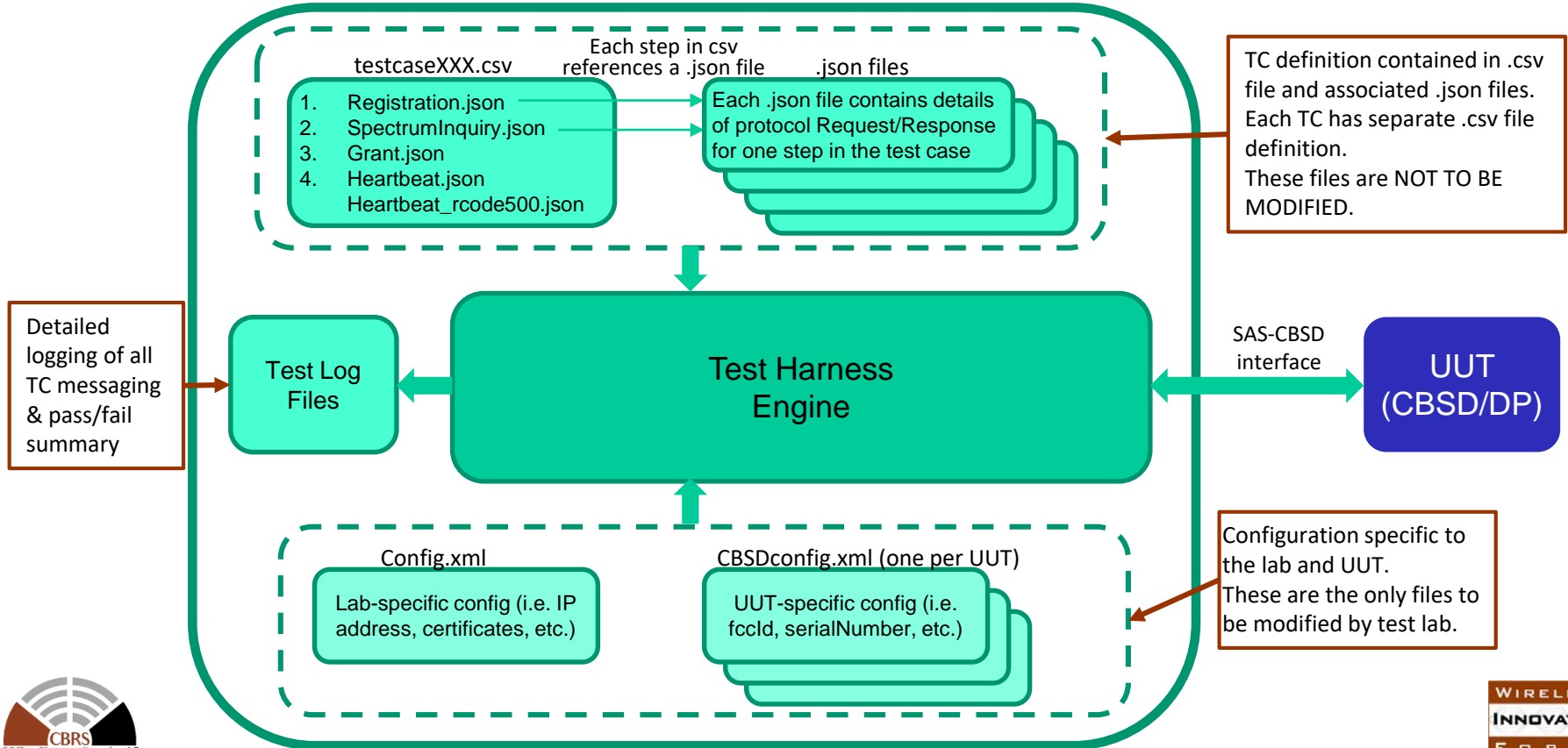
Test Harness Components (3)

Test Harness Software



Test Harness Components (4)

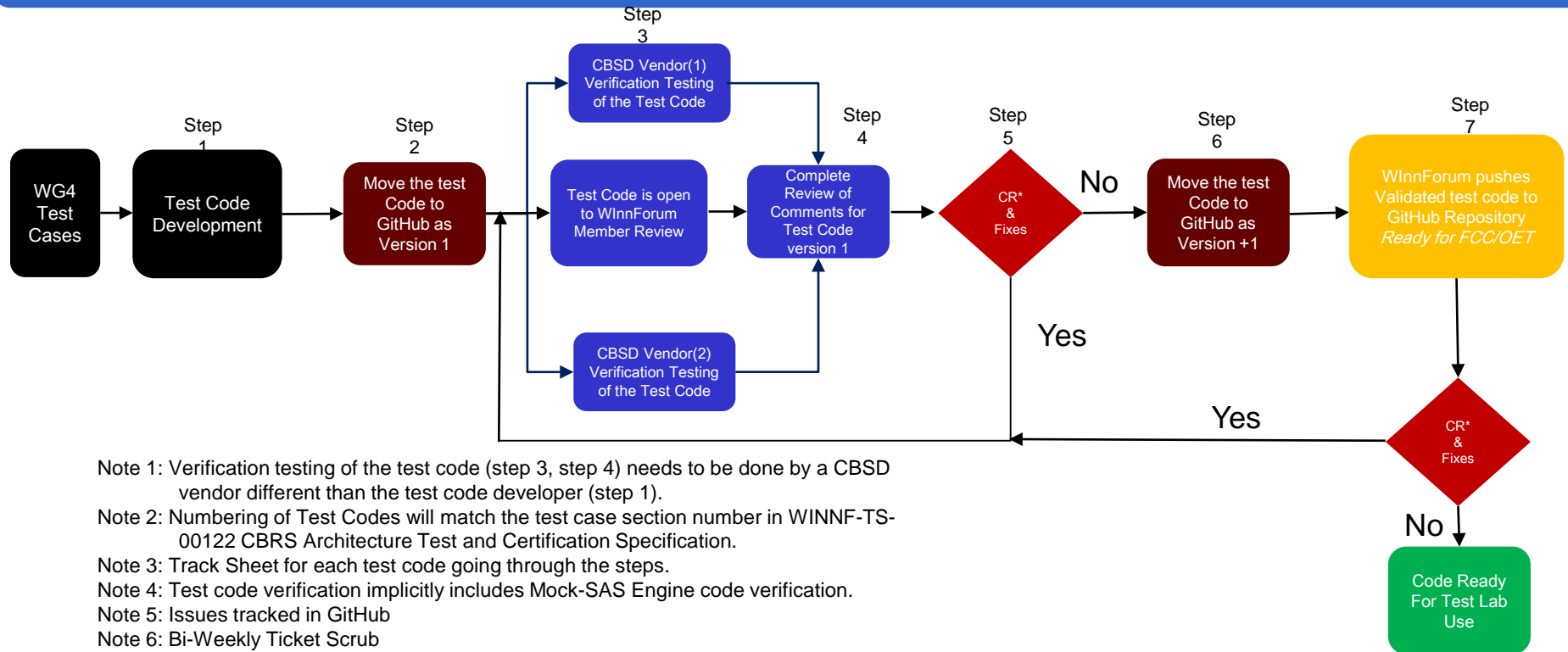
Test Harness Software



Test Code Development and Verification

- **Test code is developed within WinnForum SSC WG4 CBSD Work Group by multiple CBSD vendors**
- **Test code written in Python, code maintained in Wireless Innovation Forum GitHub repository**
- **Test code implements test cases as presented in WINNF-TS-0122 Test Specification**
- **Code for each test case is implemented by one CBSD vendor and verified by at least two other CBSD vendors participating in the Work Group, prior to release**

WinnForum CBSD Test Code Validation Method



Note 1: Verification testing of the test code (step 3, step 4) needs to be done by a CBSD vendor different than the test code developer (step 1).

Note 2: Numbering of Test Codes will match the test case section number in WINNF-TS-00122 CBRS Architecture Test and Certification Specification.

Note 3: Track Sheet for each test code going through the steps.

Note 4: Test code verification implicitly includes Mock-SAS Engine code verification.

Note 5: Issues tracked in GitHub

Note 6: Bi-Weekly Ticket Scrub

Note 7: Final code maintenance to done by jointly with CBRS Alliance Test and Cert Group

* Change Requests

GitHub Repository

Wireless-Innovation-Forum / Citizens-Broadband-Radio-Service-Device

Watch 16 Star 2 Fork 2

Code Issues 22 Pull requests 0 Projects 0 Wiki Settings Insights

Branch: master

Create new file Upload files Find file History

Citizens-Broadband-Radio-Service-Device / CBRS_Mock_SAS_Python_Master /

Idan Raz Initial commit Latest commit a12d414 13 days ago

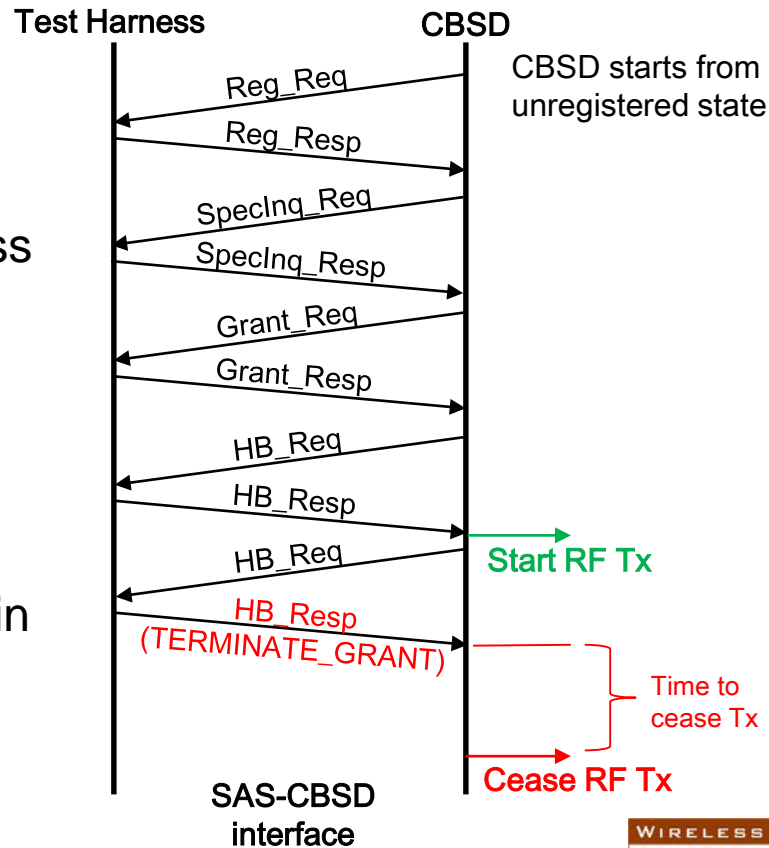
..

cbrsPython	Initial commit	13 days ago
certificates	Initial commit	13 days ago
jsonExpectedFolder	Initial commit	13 days ago
testFiles	Initial commit	13 days ago
.gitignore	Initial commit	13 days ago
.project	Initial commit	13 days ago
.pydevproject	Initial commit	13 days ago

Anatomy of Example Test Case

Example test case:

- CBSD begins in unregistered state
- CBSD triggered to start transmission process
- CBSD follows through protocol steps with SAS to start transmission
- Test harness simulates SAS responses to CBSD, ensuring CBSD adheres to protocol
- Test harness introduces stimulus as called for in test case (e.g. **TERMINATE_GRANT** in Heartbeat Response message)
- RF test equipment measures time to stop transmission from trigger condition



Test Harness Engine Requirements

- **Test Harness Engine uses Python 2.7 code and can run on Windows 7 and Windows 10 Operating Systems running on standard PCs**
- **Test lab wishing to run the Test Harness Engine needs to**
 - Prepare a PC with windows 7 or Windows 10
 - Install Python 2.7 and relevant packages – all available for free download from public internet
 - Download the Test Harness from WInnForum GitHub repository

Running Test Harness in Lab

- **Each test case required for certification in WINNF-TS-0122 has in the Test Harness corresponding CSV and JSON files**
- **Test Lab technician does not need to edit or modify the existing CSV and JSON files – this is prepared by WinnForum and maintained in GitHub**
- **Test lab technician running Test Harness needs to configure a few XML files that define**
 - General parameters of the Test Harness environment (IP address, port number, etc.)
 - CBSD UUT parameters for Registration (CAT A/B, CBSD FCC ID, CBSD Serial Number etc.)

Running Test Harness in Lab

- **Test Lab technician opens a Command Window and is activating the Test Harness Engine**
- **Test Lab technician inputs in the Command Window the relevant test case name from WINNF-TS-0122 which activates the relevant CSV and JSON files in Test Harness Engine**
- **CBSD/Domain Proxy UUT runs the relevant messages with Test Harness Engine**
- **Test Harness Engine provides a PASS/FAIL result for the test and stores relevant logs of SAS<->CBSD protocol messages and test sequence execution**
- **Test Lab technician inputs in the Command Window the next test case for CBSD/Domain Proxy UUT**

Example XML Configuration File for Test Harness General Parameters

```
<?xml version="1.0"?><cbrsParams>  
  <jsonsRepoPath>\jsonExpectedFolder\</jsonsRepoPath>  
  <testRepoPath>\testFiles\</testRepoPath>  
  <hostIp>90.0.0.114</hostIp>  
  <port>5000</port>  
  <heartbeatLimit>30</heartbeatLimit>  
  <pemFilePath>\certificates\flaskcert1.pem</pemFilePath>  
  <keyFilePath>\certificates\flaskcbstdiot.key</keyFilePath>  
  <caCerts>\certificates\cacertidan1.pem</caCerts>  
</cbrsParams>
```

Example XML Configuration File for CBSD Parameters

```
<?xml version="1.0" encoding="UTF-8"?>
<cbrsParams>
  <secondsToAddForGrantExpireTime>3000</secondsToAddForGrantExpireTime>
  <secondsToAddForTransmitExpireTime>1200</secondsToAddForTransmitExpireTime>
  <registrationParams>
    <userId>${maximumLength:128}</userId>
    <fccId>${maximumLength:128}</fccId>
    <cbsdSerialNumber>78DADA12</cbsdSerialNumber>
    <cbsdCategory>"$or":["A","B"]</cbsdCategory>
    <airInterface>
      <radioTechnology>${maximumLength:128}</radioTechnology>
    </airInterface>
  </registrationParams>
</cbrsParams>
```





Running Test Cases

```
c:\Python27>python c:\CBRS_Mock_SAS_16_Aug\cbrsPython-master\cbrsPython\controllers\startOfProject.py
Please enter a csv file path that include the test you request to run (pay attention insert the csv file without the csv suffix)
or quit to close
TERMINATED_GRANT_HB
would you like to add the test to specific folder ? (select yes or no)
no
the selected test from the user is : TERMINATED_GRANT_HB
* Running on https://90.0.0.114:5000/ (Press CTRL+C to quit)
CBSD sent registration Request
validation passed successfully, the engine sent registration Response
90.0.0.27 - - [20/Aug/2017 23:56:32] "POST /v2.0/registration/ HTTP/1.1" 200 -
CBSD sent spectrumInquiry Request
validation passed successfully, the engine sent spectrumInquiry Response
90.0.0.27 - - [20/Aug/2017 23:56:32] "POST /v2.0/spectrumInquiry/ HTTP/1.1" 200 -
CBSD sent grant Request
validation passed successfully, the engine sent grant Response
90.0.0.27 - - [20/Aug/2017 23:56:32] "POST /v2.0/grant/ HTTP/1.1" 200 -
CBSD sent heartbeat Request
validation passed successfully, the engine sent heartbeat Response
90.0.0.27 - - [20/Aug/2017 23:56:33] "POST /v2.0/heartbeat/ HTTP/1.1" 200 -
CBSD sent heartbeat Request
validation passed successfully, the engine sent heartbeat Response
90.0.0.27 - - [20/Aug/2017 23:58:33] "POST /v2.0/heartbeat/ HTTP/1.1" 200 -
CBSD sent heartbeat Request
validation passed successfully, the engine sent heartbeat Response
90.0.0.27 - - [21/Aug/2017 00:00:33] "POST /v2.0/heartbeat/ HTTP/1.1" 200 -
arrived to nstep starting question answer session with the technician
the question is : CBSD 2 stopped transmission and is in Registered state? please choose one of the answers :
y
n
y
for the question : CBSD 2 stopped transmission and is in Registered state? , the user choose y
the question is : CBSD 1 continues to transmit? please choose one of the answers :
y
n
y
for the question : CBSD 1 continues to transmit? , the user choose y
the additional comments for the current test are :
comment for CBSD vendor
The final result of the test : TERMINATED_GRANT_HB is - passed the additional comments for the current test are : comment for
CBSD vendor
Please enter a csv file path that include the test you request to run (pay attention insert the csv file without the csv suffix)
or quit to close
```

 Activating Test Harness Engine in Command Window

 Selecting Test Case to run "TERMINATED_GRANT_HB"

 CBRS/Domain Proxy UUT runs the relevant messages with the Test Harness Engine

 PASS/FAIL Result for Test Case

 Selecting next test case to run

Test Harness Overall PASS/FAIL Summary

The screenshot displays the Allure Test Pack interface. On the left is a dark sidebar with navigation options: Overview, Defects, xUnit, Behaviors, Graph, and Timeline. The main content area is divided into several sections:

- DEFECTS** 1 defect total: A red square icon with the number '1' is followed by the text "the server shot down because : ERROR - An Error occurred while comparing between actual and expected requestin the json : idan5dereg2.json".
- XUNIT** 1 items total: A progress bar for "Idandemo 9aug" shows 1 red segment and 1 green segment.
- TEST RUN** Started 10/08/2017 at 17:26:31: A large number '2' is shown above the text "Cases in 8m 39s 190ms". To the right is a donut chart that is 50% green and 50% red.
- ENVIRONMENT**: A section header with no data below it.
- FEATURES** 1 items total: A progress bar for "Without feature" shows 1 red segment and 1 green segment.
- REPORT STATS** Allure Test Pack: The word "Latest" is prominently displayed. Below it, "Allure's version" is shown. To the right, "Data size: 12.2 KB" and "Generated in: 892ms" are listed.

At the bottom of the sidebar, there is an "En" button and a "Collapse" link.

Test Harness Example of Test Case Message Logging

```
TERMINATED_GRANT_HB_2017_08_20_23_56_18.log x
1  INFO: 08/20/2017 11:56:18 PM the selected test from the user is : TERMINATED_GRANT_HB
2  INFO: 08/20/2017 11:56:32 PM registration request from CBRS : {
3      "registrationRequest": [
4          {
5              "airInterface": {
6                  "radioTechnology": "E_UTRA"
7              },
8              "callSign": "Airspan#1system",
9              "cbstdCategory": "A",
10             "cbstdSerialNumber": "76B1E617472C",
11             "fccId": "Airspan-LTE44",
12             "userId": "Airspan-radio1212"
13         },
14         {
15             "airInterface": {
16                 "radioTechnology": "E_UTRA"
17             },
18             "callSign": "Airspan#1system",
19             "cbstdCategory": "B",
20             "cbstdSerialNumber": "78DADA12",
21             "fccId": "Airspan-LTE44",
22             "userId": "Airspan-radio1212"
23         }
24     ]
25 }
26 INFO: 08/20/2017 11:56:32 PM engine sent successfully, the response to CBRS : {
27     "registrationResponse": [
28         {
29             "cbstdId": "Airspan-LTE44Mock-SAS76B1E617472C",
30             "response": {
31                 "responseCode": 0
32             }
33         },
34     ]
35 }
```

CBSD Test Program



- **WinnForum plans to collaborate with CBRS Alliance to approve Test Labs to test conformance to WinnForum Specification**
 - The program will be built on existing collaboration agreement
 - Agreement will require that testing be open to all applicants, not just CBRS Alliance members
 - Testing will be against WinnForum CBSD Test Specification and maintain the inherent technology neutrality
 - Final agreement will allow CBSD vendors to state that they are “Compliant with WinnForum CBRS Standards” and/or are “WinnForum CBRS Certified”
- **The goal is to have all the Part 96 compliance tests (RF+Protocol) to be completed such that TCB post report review can initiate PAG inquiry to OET for device authorization**
- **The CBRS Alliance managed WinnForum Test Program will include periodic audits for the test labs to assure testing standards are met**
 - Post completion of program definitions it will be made available for FCC/OET review prior to implementation
- **A nominal fee will be charged by CBRS Alliance to maintain this program**



Definitions

CBSD Test Standard

- Standard defining how a Test Lab/CBSD OEMs can be approved to run a CBSD Test program for FCC certification
- Includes a curriculum standard defining the requirements necessary for conducting non-RF Part 96 compliance tests by the test labs

Approving Body

- Entity that approves a Test Administrator to offer Part 96 CBSD compliance test services

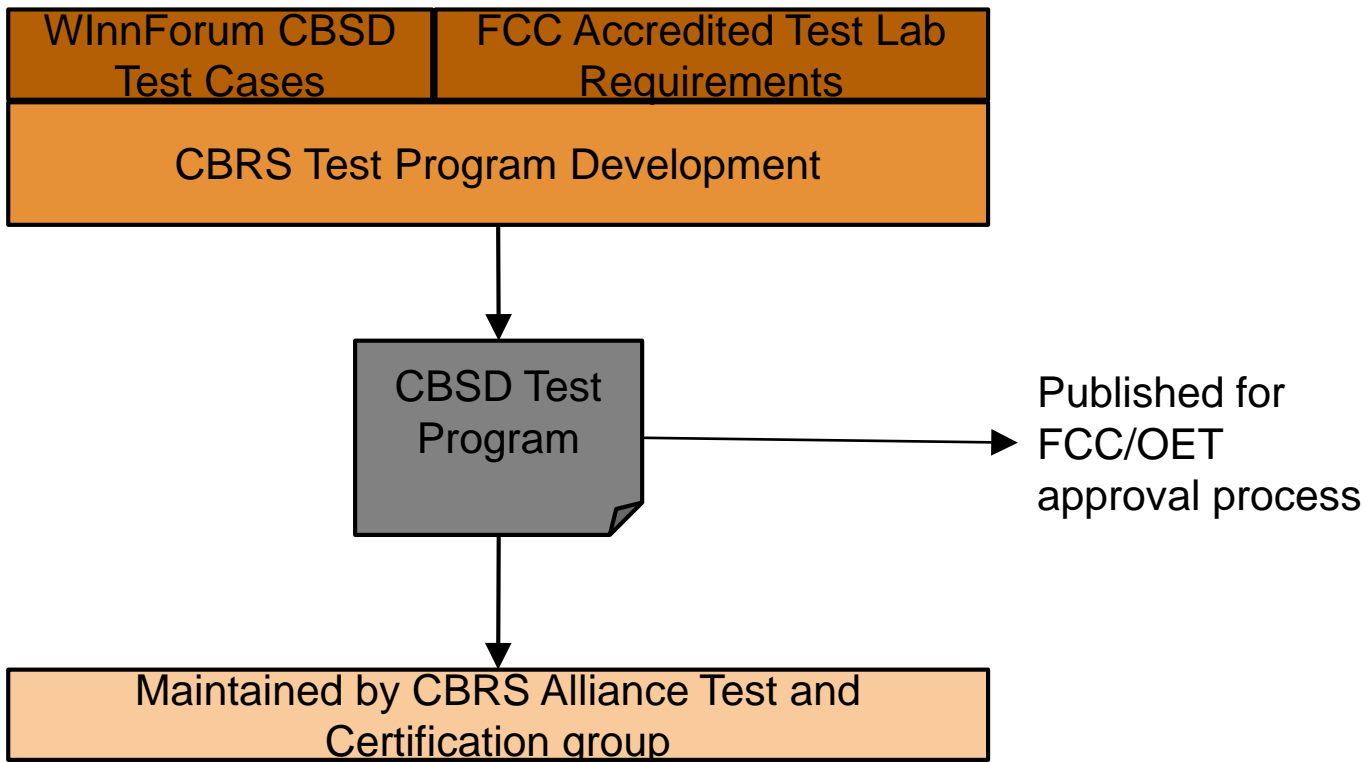
CBSD Test Administrator

- Entity that subscribes to CBRS Alliance Test program to execute tests for e.g. FCC approved Test Labs

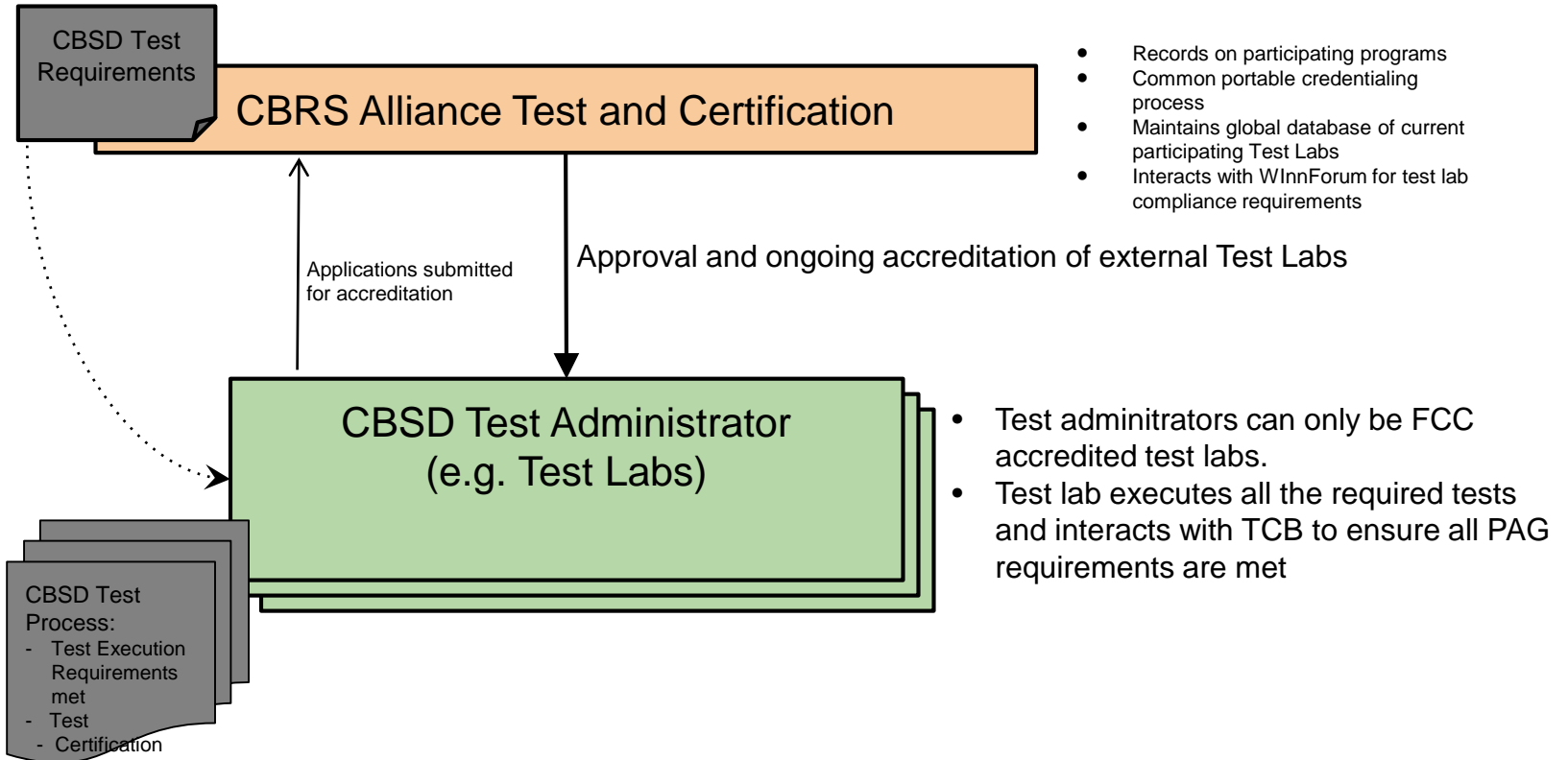
CBSD Test Program

- Maintenance procedures followed by approving body. These are the procedures required for things like record keeping of program subscribers and maintenance of publically available test specifications and test code

CBSD Test Program Development Process



WinnForum approved CBSD Test Program



Appendix

Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

Example Test Case from WINNF-TS-0122

Conformance and Performance Test Technical Specification; CBSD/DP as Unit Under Test (UUT):

- section 6.4.4.2.2
- Test Case identifier: WINNF.FT.C.HBT.43
- Test Case name: Heartbeat responseCode=500
(TERMINATED_GRANT) WINNF testing with Domain Proxy

Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

Step #1:

Ensure the following conditions are met for test entry:

DP has two CBSD registered successfully with test harness

Each CBSD {1,2} has a valid single grant as follows:

- valid *cbsdId* = C_i , $i=\{1,2\}$
- valid *grantId* = G_i , $i=\{1,2\}$
- grant is for frequency range F_i , power P_i

Both CBSD are in **AUTHORIZED** state and transmitting within their granted bandwidth on RF interface

Step #2:

DP sends a Heartbeat Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of size 2.

Ensure Heartbeat Request message is formatted correctly for each CBSD, including, for CBSDi $i=\{1,2\}$:

cbsdId = C_i , $i = \{1,2\}$

grantId = G_i , $i = \{1,2\}$

operationState = "AUTHORIZED"

Result for Step #2: PASS or FAIL



Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

Step #3:

If separate Heartbeat Request message was sent for each CBSD by the DP, the test harness shall respond to each Heartbeat Request message with a separate Heartbeat Response message.

If a single Heartbeat Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Heartbeat Response message containing a 2-object array.

Parameters for each CBSD within the Heartbeat Response message should be as follows, for CBSDi:

cbsdId = Ci

grantId = Gi

For CBSD1:

- *transmitExpireTime* = current UTC time + 200 seconds
- *responseCode* = 0

For CBSD2:

- *transmitExpireTime* = T = current UTC time

responseCode = 500 (TERMINATED_GRANT)



Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

Step #4:

After completion of step 3, test harness shall not allow any further grants to the UUT.

If CBSD sends further Heartbeat Request messages for CBSD1, test harness shall respond with a Heartbeat Response message with parameters:

cbsdId = C1

grantId = G1

transmitExpireTime = current UTC time + 200 seconds

responseCode = 0

Step #5:

Monitor the RF output of CBSD2. Verify:

CBSD2 shall stop transmission within bandwidth F2 within (T + 60 seconds) of completion of step 3

Result for Step #5: PASS or FAIL

Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

Configuring Test Harness (Mock-SAS) :

Mock-SAS needs to be configured with the following files:

- XML configuration file with the CBSD parameters required for Registration Request
- JSON files with desired Request/Response messages according to the test steps
- CSV file listing the JSON files in the order of the messages between Domain Proxy/CBSD UUT and Mock-SAS

Preparing JSON files in Mock-SAS:

Step #1 of the test case specifies that Both CBSD are in AUTHORIZED state and transmitting within their granted bandwidth on RF interface.

For this the following JSON files are prepared in Mock-SAS:

- Registration Request/Response JSON file with response code = 0 and relevant parameters
- Spectrum Inquiry Request/Response JSON file with response code = 0 and relevant parameters
- Grant Request/Response JSON file with response code = 0 and relevant parameters
- Heartbeat Request/Response JSON file with response code = 0 and relevant parameters

Example Test Case with Test Harness (Mock-SAS)

Step #2 of the test case specify Domain Proxy UUT sends Heartbeat Request.

[BACK](#)

Step #3 of the test case specify that Heratbeat Response will have response code = 500 (TERMINATED_GRANT)

For this the following JSON files is prepared in Mock-SAS:

- **Heartbeat Request/Response JSON file with response code = 500 and relevant parameters**
- **According to the steps this is the last required message between Domain Proxy UUT and Mock-SAS, meaning no more JSON file are required.**
- **The JSON files are located at directory cbrsPython-master\jsonExpectedFolder**
- **Each JSON file is a single Request/Response message for a single CBSD**
- **The JSON file which is for the last message between Domain Proxy UUT and Mock-SAS will have the name of the test case identifier.**
- **The JSON file which is for the last message between Domain Proxy UUT and Mock-SAS will have the relevant PASS/FAIL question according to the last step of the test (step 5)**

Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

Name	Date modified	Type	Size
OptionalParams	08/08/2017 05:42	File folder	
grant1.json	15/06/2017 21:48	JSON File	1 KB
grant2.json	15/06/2017 21:48	JSON File	1 KB
Heartbeat1.json	15/06/2017 21:48	JSON File	1 KB
Heartbeat2.json	15/06/2017 21:48	JSON File	1 KB
registration1.json	15/06/2017 21:48	JSON File	1 KB
registration2.json	15/06/2017 21:48	JSON File	1 KB
spectrumInquiry1.json	15/06/2017 21:48	JSON File	1 KB
spectrumInquiry2.json	15/06/2017 21:48	JSON File	1 KB
WINNF_FT_C_HBT_43_1.json	15/06/2017 21:48	JSON File	1 KB
WINNF_FT_C_HBT_43_2.json	08/08/2017 05:52	JSON File	1 KB

Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

```
{
  "name": "WINNF.FT.C.HBT.43 Heartbeat responseCode=500 TERMINATED_GRANT",
  "questions":
  [
    {
      "question" : "CBSD2 stopped transmission within bandwidth F2 within (T + 60 seconds) of completion of step 3?",
      "answers" : ["y", "n"],
      "expectedAnswer" : "y"
    }
  ],
  "heartbeatRequest":
  [
    {
      "cbsdId": "abc456cbsdSerialNumbersn1",
      "grantId": "8385094448610395450",
      "grantRenew": false,
      "operationState": "AUTHORIZED"
    }
  ],
  "response":
  {
    "heartbeatResponse":
    [
      {
        "cbsdId": "abc456cbsdSerialNumbersn1",
        "grantId": "8385094448610395450",
        "transmitExpireTime":
        "response": {"responseCode": 500}
      }
    ]
  }
}
```

Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

Preparing CSV file in Mock-SAS:

- The CSV file is located at directory cbrsPython-master\testFiles
- Each column of the CSV file relates to a single CBSD. So for Domain Proxy UUT with 2 CBSDs, 2 columns are required in the CSV file.
- Mock-SAS handles separately each CBSD and expects the Request messages in the order appearing in the lines of the CSV file.
- If Request message from Domain Proxy UUT is combined JSON array for several CBSDs, Mock-SAS will reply with combined JSON Array Response .

	A	B	C	D
1	jsonFileName	jsonFileName2		
2	registration1.json	registration2.json		
3	spectrumInquiry1.json	spectrumInquiry2.json		
4	grant1.json	grant2.json		
5	Heartbeat1.json	Heartbeat2.json		
6	WINNF_FT_C_HBT_43_1	WINNF_FT_C_HBT_43_2		

Slide 70

Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

Preparing XML configuration files in Mock-SAS:

- The XML files are located at directory cbrsPython-master\cbrsPython\model\CBRSConf
- The XML file has relevant CBSD parameters: CBSD serial number, FCC ID, etc.
- Each XML file is for a single CBSD parameters
- The JSON files refer to the relevant XML files for CBSD parameters
- Mock-SAS automatically creates cbsdID and GrantID based on the information in the XML files

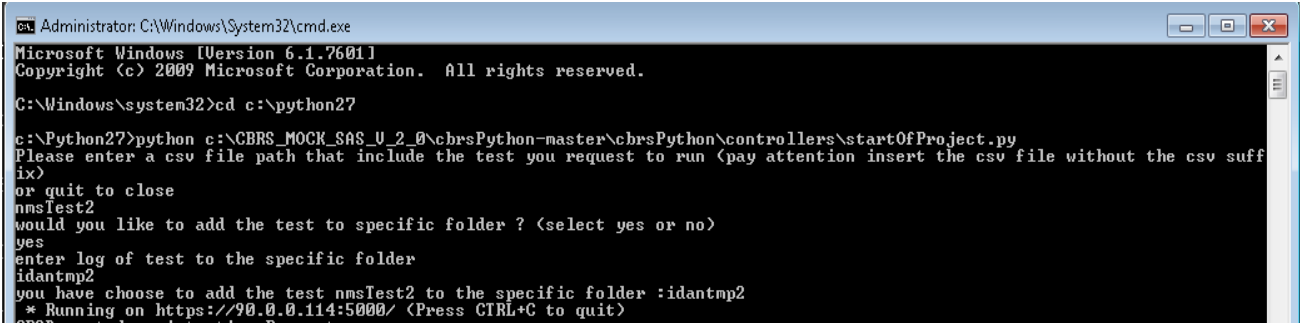
```
<?xml version="1.0" encoding="UTF-8"?>
<cbrsParams>
  <fccId>1234567890123456789</fccId>
  <userId>user123</userId>
  <grantExpireTime>2017-07-28T23:20:08Z</grantExpireTime>
  <secondsToAddForExpireTime>3777</secondsToAddForExpireTime>
</cbrsParams>
```

Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

Activating Mock-SAS (from Command Prompt):

- Mock-SAS will ask to input the CSV file name
- Test starts -> Mock SAS is waiting for CBSD/Domain Proxy UUT to send the first Request message according to the order of the lines in the CSV file.
- The Request/Response messages between CBSD/Domain Proxy UUT and Mock-SAS are seen in the Command Prompt as the test sequence progress.
- Mock-SAS validates the CBSD/Domain Proxy UUT Request messages according to the respective JSON files. If Mock-SAS fails the Request message validation, the test is reported as “FAIL”.



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd c:\python27

c:\Python27>python c:\CBRS MOCK SAS_U_2_0\cbrsPython-master\cbrsPython\controllers\startOfProject.py
Please enter a csv file path that include the test you request to run (pay attention insert the csv file without the csv suffix)
or quit to close
nmsTest2
would you like to add the test to specific folder ? (select yes or no)
yes
enter log of test to the specific folder
idantmp2
you have choose to add the test nmsTest2 to the specific folder :idantmp2
* Running on https://98.0.0.114:5000/ (Press CTRL+C to quit)
```


Example Test Case with Test Harness (Mock-SAS)

[BACK](#)

Activating Mock-SAS (from Command Prompt):

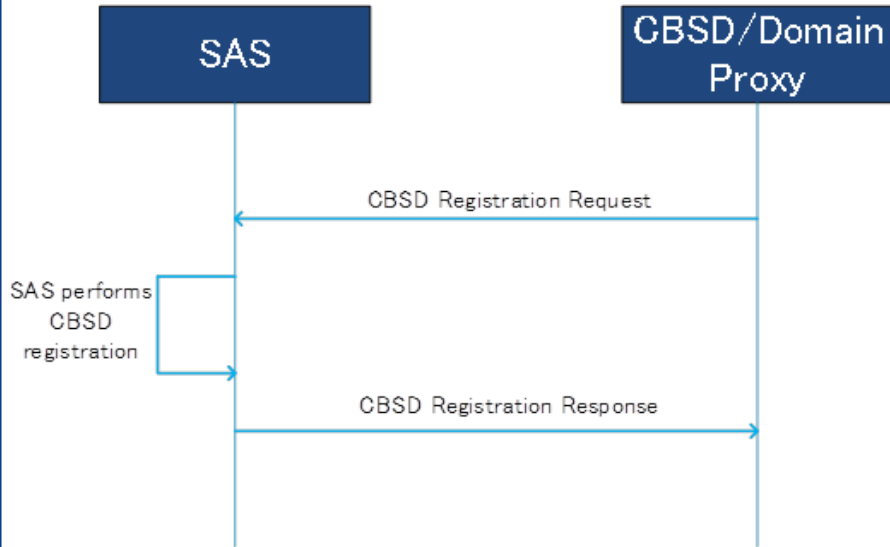
- After the last SAS-CBSD Request/Response step in the CSV file is performed, the questions are invoked.
- Answers to the questions are manually input to the Command Prompt based on the RF test equipment results
- Mock-SAS presents in Command Prompt the final PASS/FAIL result of the test.
- Mock-SAS ignores any additional incoming Request messages from CBSB/Domain Proxy UUT.
- Mock-SAS asks for the next CSV file name to run the next test.

Mock-SAS Logs:

- Logs are saved by Mock-SAS both for the Command Prompt and for the entire SAS<->CBSB JSON messages during test execution.
- The logs have timestamps based on local time from the PC running Mock-SAS
- The log files are located at directory cbrsPython-master\logs

CBSD Registration Procedure

[BACK](#)



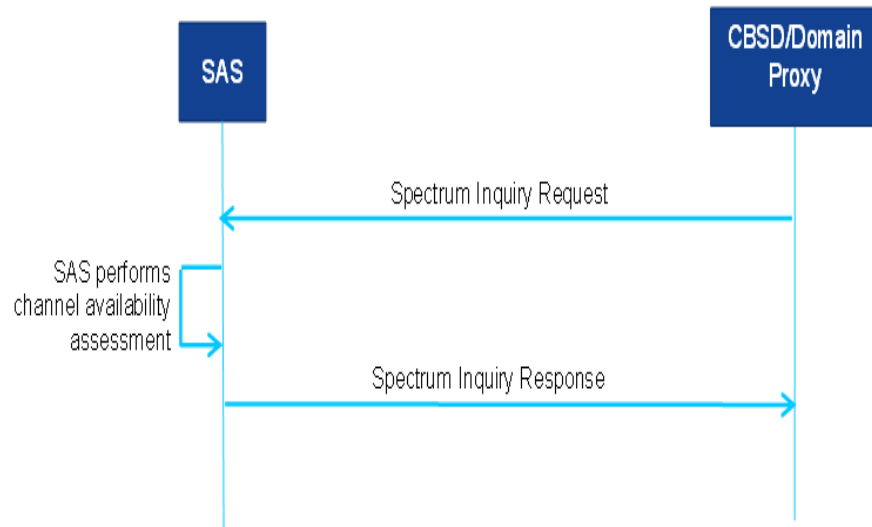
RegistrationRequest object (*userId*, *fcclId*, *cbsdSerialNumber*, *callSign*, *cbsdCategory*, *cbsdInfo*, *airInterface*, *installationParam*, *measCapability*, *groupingParam*)

RegistrationResponse object (*cbsdId*, *measReportConfig*, *response*)

- Procedure is initiated after the CBSD has successfully discovered the SAS and performed Authentication
- Some Registration parameters will be different whether the device is a Category A or B CBSD
- If the *measReportConfig* parameter is included in the *RegistrationResponse* object, the CBSD shall send the requested measurement report in the first *SpectrumInquiryRequest* object and in the first *GrantRequest* object to the SAS
- The *response* parameter in the *RegistrationResponse* indicates whether the registration succeeded or failed.
 - If the registration succeeded, the *RegistrationResponse* object contains the *cbsdId*. The CBSD uses the *cbsdId* parameter value for all subsequent procedures with the SAS.
 - If the SAS determines the registration is incomplete, the SAS returns a *REG_PENDING* value in the *response* parameter.
- Domain Proxy can perform bulk CBSD registration

CBSD Spectrum Inquiry Procedure

[BACK](#)



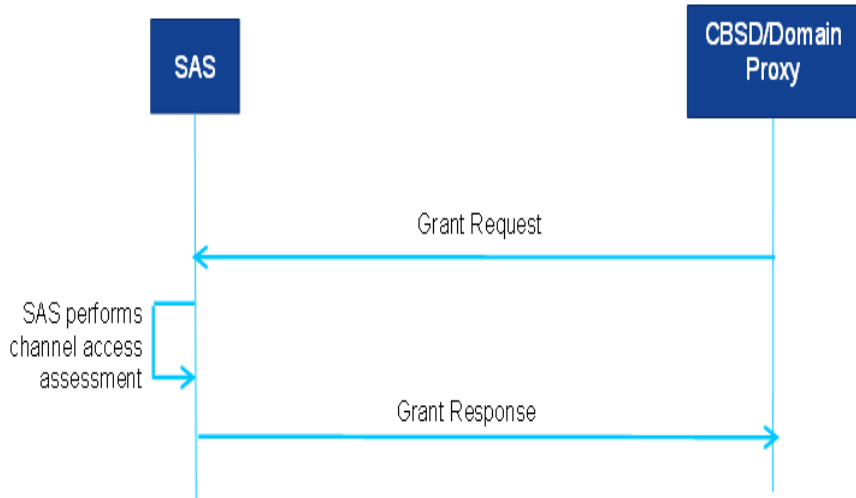
- This is an optional procedure that the CBSD may initiate any time after receiving a *cbstdId* value from the SAS (successful registration)
- Spectrum Inquiry allows Registered CBSDs to request information on available channels
- The *inquiredSpectrum* parameter is an array of *FrequencyRange* objects indicating the frequency range(s) for which the CBSD seeks information
- The CBSD should consider the information in the *availableChannel* parameter (results of the potential channel availability for the inquired spectrum) in the response from the SAS as an indication of the channels available to the CBSD.
- If there is a Domain Proxy, it can send/receive bulk messages

SpectrumInquiryRequest object (*cbstdId*, *inquiredSpectrum*, *measReport*)

SpectrumInquiryResponse object (*cbstdId*, *availableChannel*, *response*)

CBSD Grant Procedure

[BACK](#)



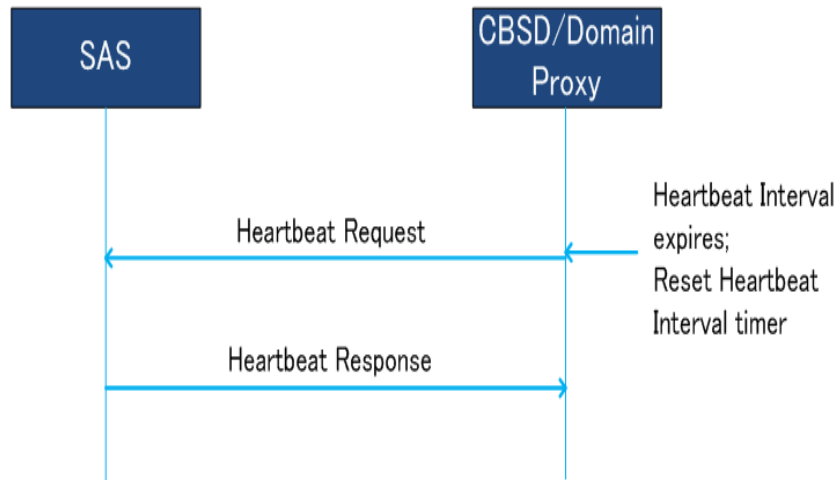
GrantRequest object (*cbsdId*, *operationParam*, *measReport*)

GrantResponse object (*cbsdId*, *grantId*, *grantExpireTime*, *heartbeatInterval*, *measReportConfig*, *operationParam*, *channelType*, *response*)

- The Grant procedure describes how a CBSD requests spectrum from the SAS, following a successful registration
- The CBSD shall specify in the *operationParam* of the Grant request the frequency range and maximum EIRP parameters it wants to use for operation
- If the SAS approves the Grant request, the SAS allocates spectrum according to the parameters in the *operationParam* parameter
- SAS shall determine the eligibility of the CBSD to use a PAL reserved channel
- If the *GrantResponse* object includes an *operationParam* parameter, the CBSD may elect to use that information to issue a new *GrantRequest* object.
- The CBSD shall not use the spectrum (i.e., activate its radio transmitter) until successfully completing the Heartbeat procedure
- Domain Proxy can aggregate grant requests for multiple CBSDs

CBSD Heartbeat Procedure

[BACK](#)



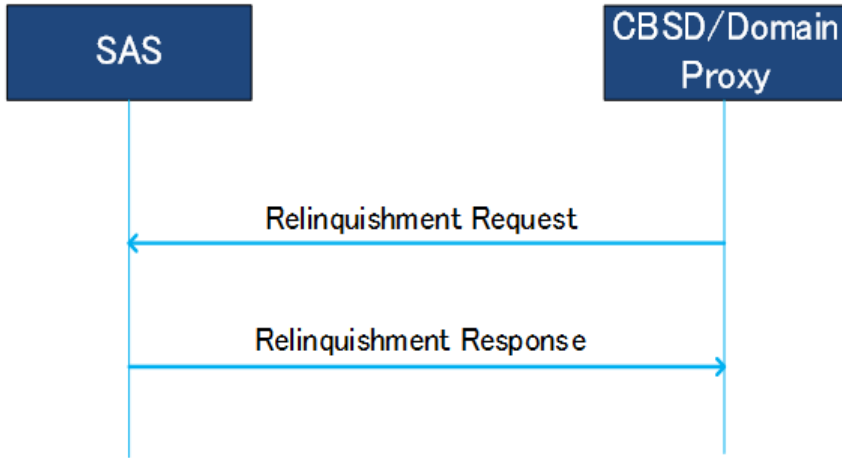
HeartbeatRequest object (*cbsdId*, *grantId*, *grantRenew*, *operationState*, *measReport*)

HeartbeatResponse object (*cbsdId*, *grantId*, *transmitExpireTime*, *grantExpireTime*, *heartbeatInterval*, *operationParam*, *measReportConfig*, *response*)

- The *HeartbeatRequest* informs the SAS that the CBSD needs access to the allocated spectrum for the specified *grantId*
 - CBSD initiates this procedure any time prior to the expiration of the Heartbeat Interval timer.
- The procedure allows the SAS to suspend or terminate the Grant.
 - The SAS has the option of suggesting the CBSD request alternative spectrum based on the *operationParam* parameter in its response
- the CBSD shall discontinue transmission for the Grant within 60 seconds after the time specified in the *transmitExpireTime* parameter
 - If the *transmitExpireTime* parameter expires prior to reception of a *HeartbeatResponse* object
 - If the *response* parameter indicates TERMINATED_GRANT
- SAS to CBSD connectivity is considered to be lost when, during a seven-day period, there is no successful Heartbeat procedure between the SAS and the CBSD

CBSD Grant Relinquishment Procedure

[BACK](#)



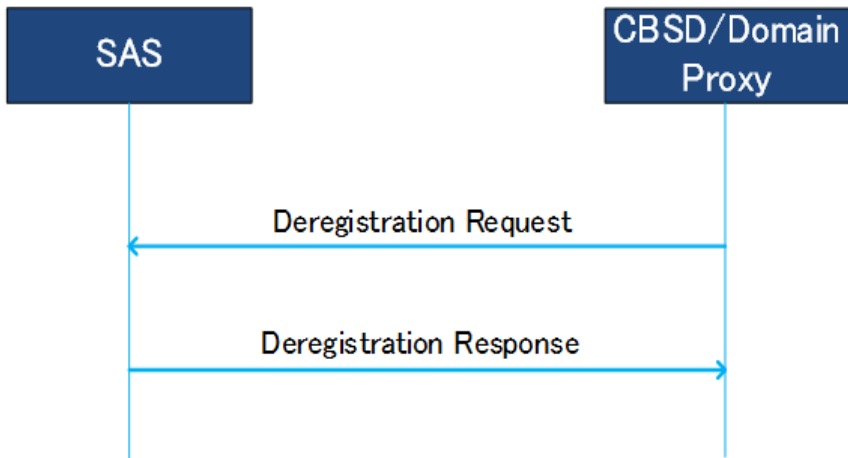
RelinquishmentRequest object (*cbsdId*, *grantId*)

RelinquishmentResponse object (*cbsdId*, *grantId*, *response*).

- The CBSD may initiate this procedure for an existing Grant. The CBSD shall terminate radio operation associated with this Grant before initiating this procedure.
- Upon reception of the *RelinquishmentRequest*, the SAS relinquishes the spectrum assigned to the CBSD.
 - If the request succeeded, the CBSD no longer has authorization to use the spectrum associated with the Grant.
 - In case of failure, regardless of the reason for the failure of *RelinquishmentRequest*, the CBSD no longer has authorization to use the spectrum associated with the Grant.
- Domain Proxy can aggregate relinquishment information for multiple CBSDs and send a bulk *RelinquishmentRequest*

CBSD Deregistration Procedure

[BACK](#)



- When the CBSD determines that it should deregister from the SAS, it shall cease transmission associated with any Grants and then send a *DeregistrationRequest*
- The SAS marks the CBSD as Unregistered, removes any existing Grants, and responds with a *DeregistrationResponse*
- The CBSD should send a *RelinquishmentRequest* for each Grant prior to sending the *DeregistrationRequest*
- Domain Proxy can aggregate deregistration requests for multiple CBSDs

DeregistrationRequest object (*cbsdId*)

DeregistrationResponse object (*cbsdId*, *response*)