RadioDNS Testing Platform

Nick Piggott - 21st August 2019

Overview

RadioDNS is commissioning the development of a testing platform. The purpose of this platform is to test one or more (radio) services to validate that their RadioDNS services are configured and operating correctly according to the relevant technical specifications (TS 103 270, TS 102 818, TS 101 499)

The intended users of this platform are:

- Automotive industry: to test problems they have identified during their own system testing, to establish if the problem is with their implementation of their client, or the service provided from the service provider;
- Broadcasters: to validate that their own system (or the system provided to them by a service provider) is working correctly;
- RadioDNS to: periodically and automatically check that the services provided by each of the registrants in the DNS are operating correctly.

This document describes the required functionality and operating environment. We'll expect to work interactively and iteratively with the developer(s) to get the fine detail correct.

We are inviting tenders to develop this platform and deliver it to us to operate, and provide fix-and-support service.

Environment

The testing platform should operate in a cloud environment, preferably using a scripting / serverless methodology. For example, node.js running on Amazon Lambda.

The source code must be provided, and the code should be installed and made operational in a cloud environment provided by RadioDNS. RadioDNS will cover all costs of operating the platform.

Responses

Please send us a written proposal by 2nd September 2019, which tells us:

- What technology you're proposing
- Your delivery timescale
- Your costs

If you have any questions, please email feedback@radiodns.org

Required Functionality

The interface to the platform should be:

- Programmatic (API) (as part of machine / scheduled / periodic / automated testing)
- Manually, through a browser user interface, with a suitable visual design (which we can provide examples of)

The inputs to be captured in the browser user interface are:

- **Bearer Parameters** - (through context sensitive input starting with choice of bearer - e.g. “fm/dab/hd”) to generate bearer URI
- **Bearer URI for Lookup** as a free text string
- **Authoritative FQDN** as a free text string (with structure validation)
- For each Authoritative FQDN, **zero or one ClientID key** for SPI testing
- For each Authoritative FQDN, **zero or more Bearer URIs for Filter**
- To allow testing of Programme Information, capture number of days to test , 0=don’t test, 1 = today, 2 = today + tomorrow, 3 = today + following 2 days, etc.
- To allow testing of Visuals, capture desired image height, width, dpi values - default to 240 / 320 / 72)
- Allow tests to resolve bearers against “radiodns.org” or “test.radiodns.org”

The API inputs are:

- **0-n Bearer URI for Lookup**
- **0-n Authoritative FQDN**
- For each Authoritative FQDN, **zero or one ClientID key** for SPI testing
- For each Authoritative FQDN, **0-n Bearer URI for Filter**
- To allow testing of PI, capture number of days to test , 0=don’t test, 1 = today, etc
- To allow testing of VIS, capture height, width, dpi values - default to 240 / 320 / 72)
- Specify the testing zone (radiodns.org or test.radiodns.org)

The tests are run on:

- The specified FQDNs and;
- Services within each FQDN which have a bearer which matches a Bearer URI for Filter specified for that FQDN. (If no Bearer URI for Filter are passed with the FQDN, then test all services on that FQDN).

The outputs are:

- A structured report suitable to display in the browser (e.g. HTML/CSS) AND/OR
- A structured report suitable to emailed (to be decided) AND/OR
- A report suitable to distribute as a file. (e.g. PDF)

The user should be offered their choice of which of these outputs they require. If they choose email, it should be emailed to the user’s email address.
All reports should be stored in the system and/or emailed to a preset list of 1-n email addresses which are editable by a system administrator.

The report should show sufficient human-readable information for each service tested for it to be recognisable to a reader (e.g. name, logo, bearers in human format), then for each test on that service:

- **PASS**
- **WARN**, tests continue but display the relevant message for that point along with any future warnings or a failure.
- **FAIL**, tests for this service halt at this point and report the relevant message.

Access to the platform must be restricted to authorised and authenticated users.

- For the **programmatic interface** through a pre-shared authentication key, using a suitable HTTP authentication method
- For the **manual interface** through a white list of email addresses (either full email or domain match), sending a time limited link (2 hours) to the email address entered by the user. The link should be immediately invalidated on use, with a 30 day cookie set to persist the login in their browser. Subsequent use of the link, where a cookie is already set, should still allow access as normal.

This list of authentication keys and white list of email addresses must be editable by a system administrator.

Testing Process

1. **FQDN Resolution**
   - 1.1. **For each Bearer URI for Lookup provided:**
     Validate bearer, FAIL if invalid.
     GCC only (no ISO country code), parameters in required format. Attempt to capture common gotchas.
   - 1.2. **Does Bearer resolve to CNAME record (the fqdn)? Allow for radiodns.org or test.radiodns.org suffix.**
     If no, FAIL.
     If yes, add to list of FQDNs and add Bearer as a Bearer URI for Filter for that FQDN.

2. **Locate Services and Hosts**

   For each FQDN
   - 2.1. **Resolve hostnames (servers) for EPG, SPI, VIS and VIS-HTTP:**
   - 2.2. **Does _radioepg.<fqdn> SRV record resolve? Output all records, FAIL if host is not an A record, WARN on any duplicate host:port entries**
   - 2.3. **Does _radiospi.<fqdn> SRV record resolve? Output all records, FAIL if host is not an A record, WARN on any duplicate host:port entries**
   - 2.4. **Does _radiovis SRV record resolve? Output all records, FAIL if host is not an A record, WARN on any duplicate host:port entries**
   - 2.5. **Does _radiovis-http SRV record resolve? Output all records, FAIL if host is not an A record, WARN on any duplicate host:port entries**
   - 2.6. **Did no SRV records resolve against CNAME? If yes, FAIL.**
     You should advertise at least one server for at least one application for your bearer registration.

3. **Service and Programme Information Application Support**

   For each SPI hostname (server):
   - 3.1. **Request /radiodns/spi/3.1/SI.xml over HTTPS (using the hostname:port number) without providing ClientID:**
     If the response is not 401 and a ClientID was provided for this FQDN, **WARN** that a ClientID was provided but the server is responding anonymously.
     If the response is not 401, jump to 3.4.
   - 3.2. **If no Client ID was provided for this FQDN**
     FAIL that this server requires a Client ID and non was provided, so testing of SPI on this server will not continue.

3.3. Request /radiodns/spi/3.1/SI.xml over HTTPS with the ClientID for this FQDN
If the response is 401 **FAIL** with the warning that the SPI server refused the supplied client identifier.

3.4. Store the retrieved SI.xml document, HTTP status and HTTP Headers.
At the end of processing all SPI hostnames, compare the SI.xml documents. If they are not all identical, **FAIL** that different version of SI.xml have been received and end SPI testing by removing all SPI/SI documents retrieved. Otherwise retain one version of the EPG/SI document.

For each EPG hostname (server):

3.5. Request /radiodns/spi/3.1/SI.xml over HTTP (using the hostname:port number)

3.6. Store the retrieved SI.xml document, HTTP status and HTTP Headers
At the end of processing all EPG hostnames, compare the SI.xml documents. If they are not all identical, **FAIL** that different version of SI.xml have been received and end EPG testing by removing all EPG/SI documents retrieved. Otherwise retain one version of the EPG/SI document.

3.7. If there are no SI.xml documents (either SPI or EPG)
FAIL - no requests for SI.xml have succeeded.

For each retrieved (SP/EPG) SI.xml document

3.8. If the document was received via a 3xx status, WARN.
The document was returned through a redirect, and provide the re-directed URL.

3.9. If the MIME type is not prefixed application/xml (allowing for +SI etc.), FAIL.
The MIME type is not valid for an SI document.

3.10. Check no HTTPS URLs exist in an EPG/SI document - WARN if they do, listing the element and attribute name and URL

3.11. Validate the SI file against the schema. If not valid, FAIL.
The SI document does not validate against the schema.

3.12. Validate each image in the service provider and ensure they match their definitions.

3.13. Check for terms attribute, if none WARN, if defined INFO with contents.

3.14. Is each bearer unique to one service only, if not FAIL.
A bearer cannot be assigned to more than one service.

3.15. For each Bearer URI for Filter associated with this FQDN:
Perform tests up to 2.25 against only against services matching this bearer uri, else perform test on all services.

3.16. Does each bearer validate and lookup to match the same server being currently tested? If not, FAIL
A bearer on the service in the SI is being claimed by another registration within RadioDNS. Detail which bearer and where it points.

3.17. If no <radiodns> element is included for a service, WARN and SKIP to 2.19
All services should include a <radiodns> element to assist with verification.

3.18. Do the <radiodns> element parameters match those originally encountered during bearer resolution OR directly supplied, if not FAIL.
The service element refers to an alternative authoritative FQDN to that discovered whilst performing bearer lookup.

3.19. Request each logo and verify the sizes and type match those defined and what the specification defines, if not FAIL.
Ensure all logos in the SI are defined correctly in terms of size and accurately reflect the images themselves.

3.20. Ensure no logos are missing that are required in the specification, else FAIL.
Check that all five logos are specified, that they exist on HTTP URLs (INFO any redirects), WARN if the logo_square and logo_rectangle file types are not PNGs, check they are the dimensions they say they are, and warn if the filesize is bonkers

3.21. If image sizes (in bytes) are unusually large, WARN.

3.22. Request each IP stream, if 404 FAIL
Stream is unavailable. Perhaps this is due to a geoblock, please consider where suitable to include in the SI file.

3.23. Verify the codec and bitrate matches those defined, if playlist encountered verify streams within playlist, if not FAIL.
Ensure all IP stream definitions accurately reflect the codec and bitrate of the resource.

3.24. If a service defines itself in a service group, ensure the service group exists.

3.25. If no bearer was originally supplied or that bearer is found in the document SKIP to 2.29.

3.26. If DAB bearer, is the same SId found on another mux? If yes, FAIL
The multiplex for this bearer appears to be missing.

3.27. If DAB bearer and the SId and EId are found but reversed, FAIL
The DAB bearer is present but incorrectly configured.

3.28. If DAB or FM bearer and a match is found with incorrect GCC, FAIL
Possible match with invalid GCC found.

3.29. If FM matches on PI but different frequency, FAIL
Possible match with invalid frequency.

3.30. If a service group exists with no services associated, FAIL.
A service group should have at least one member.

3.31. Request n days PI documents based on the path to the SI file. If not found, WARN and end Section 2.

3.32. Validate the PI file against the schema, if invalid FAIL.
The PI file does not validate against the schema.

3.33. Ensure all programmes in the PI document fall within the date requested and sequential, FAIL if not.
PI documents should contain sequential programmes that do not overlap (start + duration must not be greater than another start time) and occur within the date requested (pay close attention to what the EPG spec says about local time).

3.34. Validate images, FAIL if any don't match their definitions.

4. Slideshow Application Support

Combine bearers from the filtered services in SI file with Lookup Bearers to create a list of unique topics

4.1. Stomp

4.1.1. Does the server successfully handshake on supplied port within 5 seconds? If no, FAIL.
Server took too long to respond to a connection attempt.

4.1.2. Does the server successful respond to a subscription to the supplied topic (either derived from a bearer lookup or supplied as a parameter for the service?) If no, FAIL.
Unable to subscribe to topic.

4.1.3. Skip to 3.3

4.2. HTTP

4.2.1. Does the server respond to an HTTP topic request with a 200? If no, FAIL.

HTTP Slideshow request failed.

4.2.2. Does the response include a Message ID? If no, WARN

A message ID should be included and used in subsequent requests to minimise polling.

4.2.3. If a Message ID was received, make a new request with that message ID as last_id and enforce a response timeout of 10 seconds. Did the request timeout or return a new message? If not, FAIL.

The same message should not be repeated when a message ID is supplied.

4.2.4. Attempt a JSONP request, does the server return an appropriately formatted callback? If no, FAIL.

Slideshow HTTP server does not implement JSONP callback format.

4.3. Common

4.3.1. Does the server successfully deliver a frame within 5 seconds? If no, FAIL.

Server should have retroactive enabled to deliver a frame on subscription, else had regular delivery to ensure fast state acquisition.

4.3.2. For an image frame, attempt a standard frame request. Is the image a PNG or JPEG and 320x240 (should dwell no longer than 10 seconds to try and capture an image)? If no, FAIL.

Invalid image format.

4.3.3. Attempt to request an alternative (larger than standard) image size using custom headers. Is the image the same size or smaller than the dimensions requested? If bigger, FAIL. If requested size, INFO custom headers supported, if smaller WARN custom headers may not be supported.

Invalid support for custom image sizes. The image must either match the request, or be smaller than requested if unable to support.

4.3.4. For a text frame, is the text longer than 128 characters (should dwell no longer than 10 seconds to try and capture a text frame)? If yes, FAIL.

Text messages must be no longer than 128 characters.

4.3.5. For either frame, if there's a link URL, is the URL valid? If not, FAIL.

URLs included as links on a frame should be valid.

4.3.6. If no text or image frame received within dwell time, WARN.

No (text/image) frame received. It is recommended to implement both frame types to ensure coverage for all device classes.

Indicative Process Flow