# Introduction

DASH conformance software performs the task of validating at least the DASH MPD and also the segments pointed by that MPD, as shown in Figure 1 [1]. If there is any conformance issue found, it is reported back. This document provides the software architecture, functionality, and basic usage of the conformance software for DASH and extension to HbbTV/DVB [2,3].

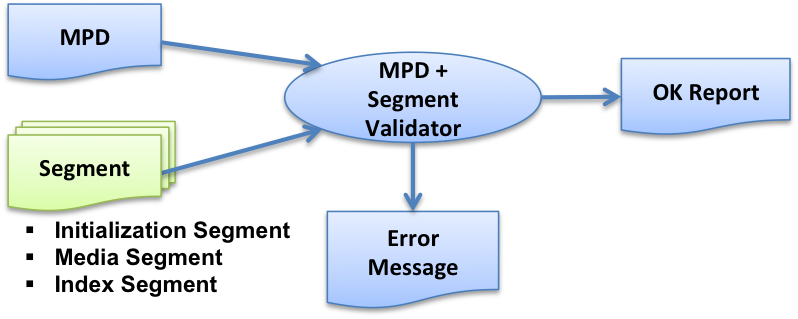


Figure 1: Role of conformance software [1]

Figure 2 shows a relatively detailed process flow of the above-mentioned conformance software. The process flow is described below.

1. Based on the provided location, the MPD is loaded.
2. XML validation and schema validation is done based on the MPD and the MPD schema, the latter pointed to by the MPD.
3. Schematron rules are validated.
4. If segment conformance testing is required, the MPD information is used to access/fetch the segments.
5. ISO BMFF rules are validated.
6. DASH specific rules are validated:
   1. General DASH rules (e.g. for sidx, ssix boxes etc).
   2. MPD information specific rules, e.g.
      1. Profile specific checks,
      2. Encrypted content specific checks etc.
7. Cross-representation checks are made (e.g. segment and subsegment alignment).

HbbTV/DVB conformance can also be verified when the respective mode is selected, which has the following additional process flow as in Figure 2.

1. HbbTV/DVB specific rules are validated: ISO and non-ISO boxes.
2. Manifest specific checks are validated in coordination with boxes.
3. Cross-Track checks are validated (e.g. Identical boxes in all Tracks).

Each of the processing green blocks may generate conformance error messages that are provided to the user. Only if none of the blocks generate any error, the content is conforming to the specific DASH/HbbTV/DVB profile.

DASH Box Checks (General + MPD specific)

Yes

Schema

MPD/Manifest

Segments

Cross-representation checks

End

Start

Fetch MPD, Schema

XML validation+Schema validation

Schematron rule validation

Segment Conformance?

Fetch Segments

ISO BMFF checks

MPD location (HTTP/

File)

Select mode DASH/DVB/ HbbTV

HbbTV/DVB specific Box checks (ISO & non-ISO)

Box + Manifest specific checks

Cross Track checks

No

Yes

No

Mode= HbbTV/DVB?

Figure 2: Detailed flow of DASH and HbbTV/DVB conformance testing

# Realization architecture

Figure 3 shows the functional block diagram of how the DASH conformance software has been realized.

* The conformance software resides on the “Test Server”.
* The “Tester” is a web-client, e.g. Google Chrome browser that access the conformance software on the Test Server.
* The Tester provides the MPD (either as an uploaded file, or points to the MPD location on the web) to the conformance software.
* Based on this information, the conformance software accesses the DASH content.
* Conformance testing is done on the Test Sever.
* The report of the conformance testing is provided back to the Tester.



DASH Content (3)

Test Server (2)

Tester (1)

1. Test Request (MPD)

2. Request content

3. Run Conformance tests

4. Results

Figure 3: Functional block diagram of realization architecture

### Test Server

Linux based Web server

Frontend code: <https://github.com/niteeshbhat/HbbTV_DVB_Conformance_Frontend>

Backend code: https://github.com/niteeshbhat/HbbTV\_DVB\_Conformance\_Backend

# Setup of Test Server

Test server can be installed in Linux. Windows support has been deprecated.

The following packages need to be installed by the specified linux commands.

1. PHP (5.6 or above)

*sudo add-apt-repository ppa:ondrej/php*

*sudo apt-get update*

*sudo apt-get install php5.6 php5.6-dev php php5.6-curl php5.6-xdebug libapache2-mod-php5.6*

1. Install Java

Manually download the newest Oracle (Sun) Java directly from the link <https://www.java.com/en/download/linux_manual.jsp>

choose the *.tar.gz* version, unpack them and place at a desired place.

Afterwards setup the java path with the following commands:

*sudo update-alternatives --install "/usr/bin/java" "java" "path\_to\_java" 1*

*sudo update-alternatives --install "/usr/bin/javaws" "javaws" "path\_to\_javaws" 1*

The paths are located in the bin folder of the unpacked package, which is usually called jre1.8.0\_version, so the path\_to\_java could be e.g. /opt/jre1.8.0\_version/bin/java.

Ensure that you also have a java compiler (JRE). This can be verified by using:

*javac -version*

Also, ensure that javac and java use the same version. This can be verified by comparing the two commands:

*java -version*

*javac -version*If there is no javac present, install the same version as java. For example, the javac 1.8 version could be installed by:

*sudo apt-add-repository ppa:webupd8team/java*

*sudo apt-get update*

*sudo apt-get install oracle-java8-installer*

1. Apache server

Apache 2.4 is recommended, the root folder is /var/www/html/.

*sudo apt-get install apache2 apache2-doc*

1. Configure Apache

Go to the root folder, copy or move the web contents (DASH/DVB frontend code) there, or make a softlink of the projects.

Rename or move the index.html, so that the user defined contents are shown in http://localhost/

If you want to change the root location, then modify the file /etc/apache2/sites-available/000-default.conf

When there is no error with the root location, please check if the user or group have corresponding rights. Check the settings by ' *ls -l ‘*.

Make sure that you are in the group www-data and add write permission to the users in this group by:

*sudo chmod -R 0777 /var/www/*

(Sometimes addition to the sudoers list is needed. Run sudo visudo and add these lines at the end: *www-data ALL=(ALL) NOPASSWD:ALL*)

After any configuration change, a restart of apache service is necessary:

*sudo service apache2 restart*

1. Install python
2. *sudo apt-get install python2.7*
3. *sudo apt-get install python-pip*
4. *sudo apt-get install python-matplotlib*
5. Possible additional installations
6. Install ‘ant’, required to run MPD validator in Conformance-Software.

*sudo apt-get install ant*

1. XML extension of PHP

*sudo apt-get install php5.6-xml*

1. Stdc++ package

*sudo apt-get install libstdc++6:i386*

1. If everything works correctly, you should be able to be open the Frontend conformance software in your browser by navigating to:

*http://localhost/*[*HbbTV\_DVB\_Conformance\_Frontend*](https://github.com/niteeshbhat/HbbTV_DVB_Conformance_Frontend)*/webfe/conformancetest.php*

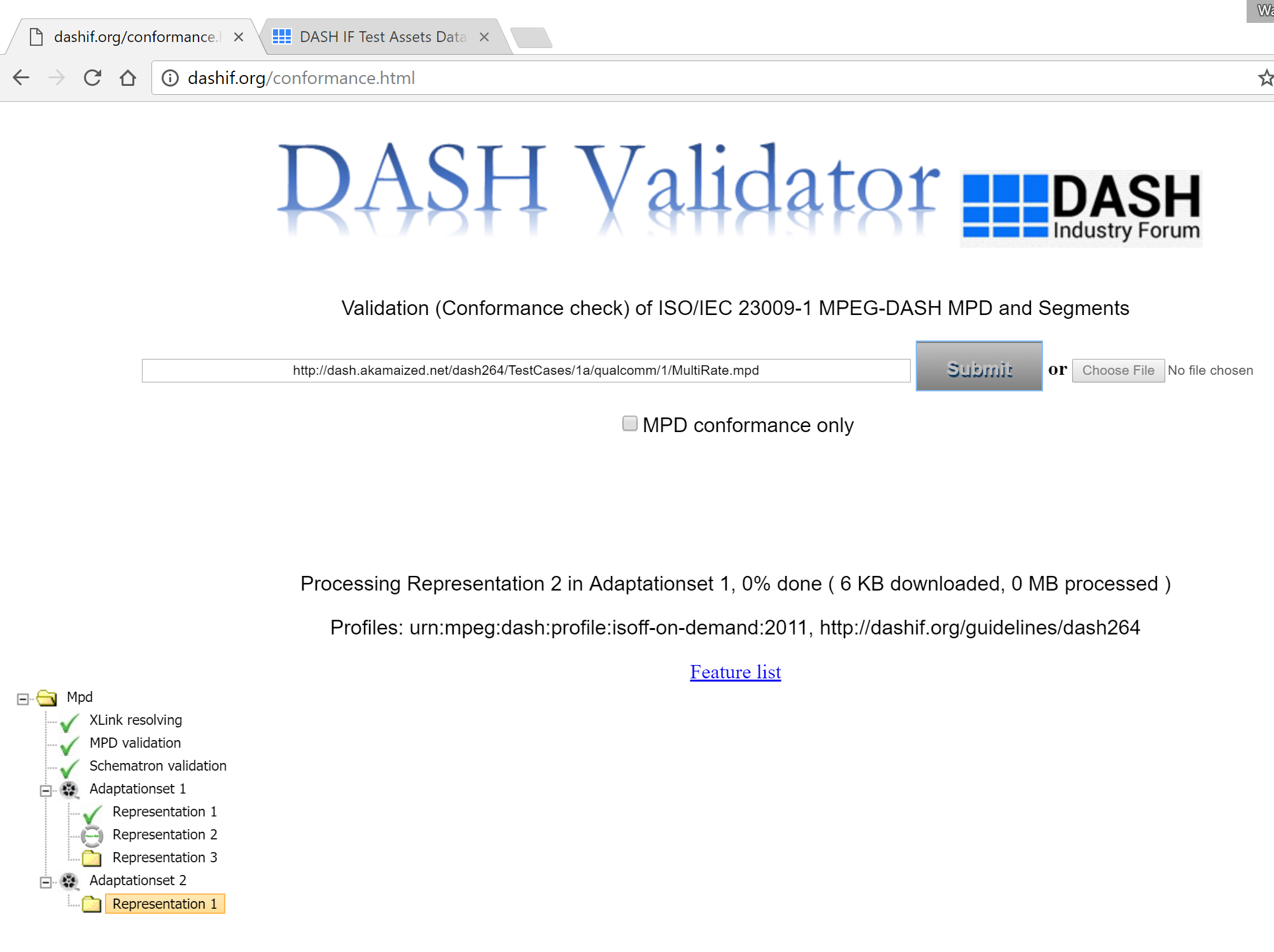
# Web-user interface

### 4.1 DASH Frontend

The web-based user interface for interaction with the conformance software is shown in Figure 4.

MPD URL or file selected

Results



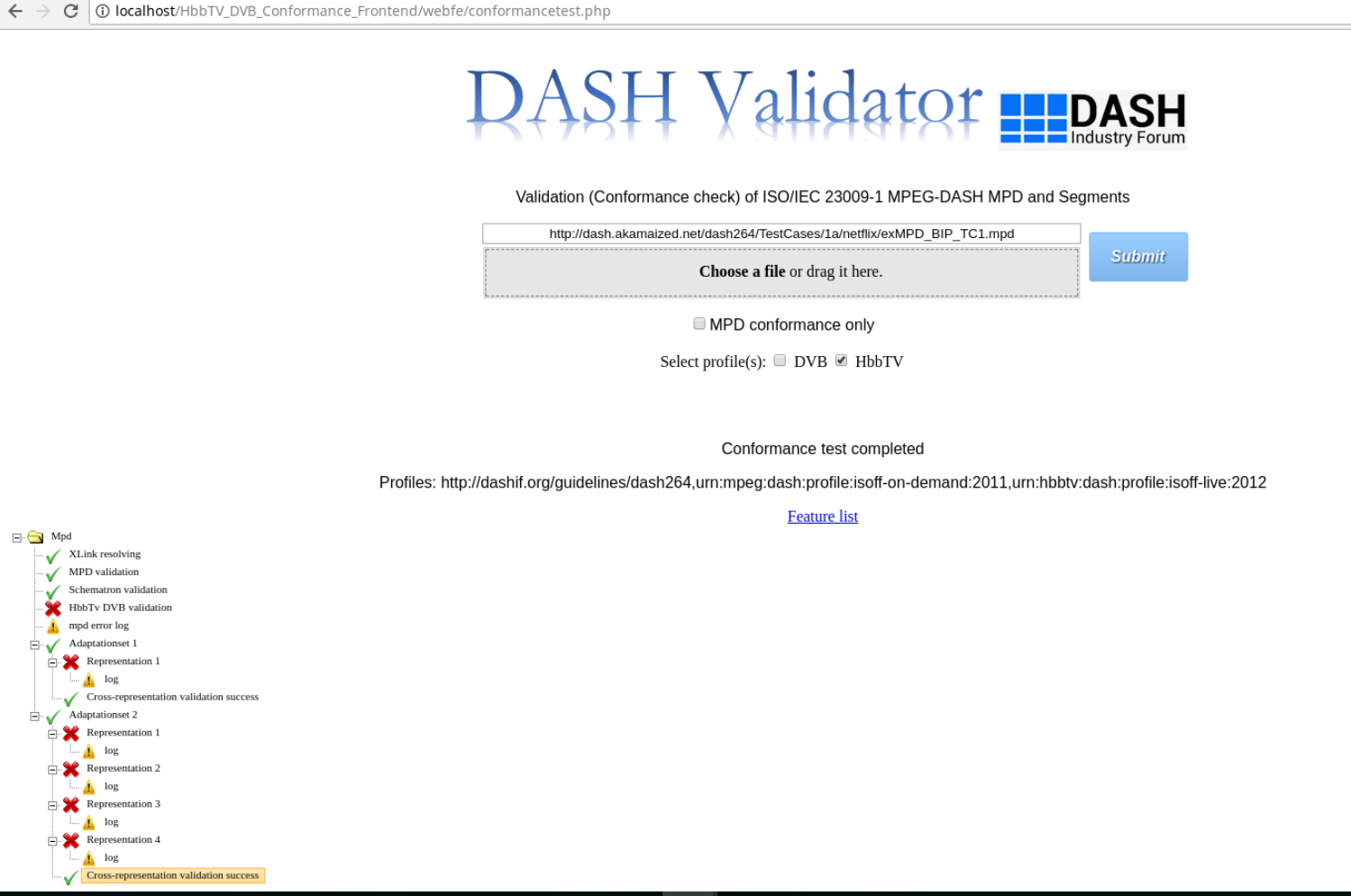
MPD URL or file selected

Results

Figure 4: Web user interface of the conformance software.

### 4.2 HbbTV/DVB Frontend

A checkbox is used to initiate conformance validation against HbbTV profile or DVB profile or both as shown in the Figure 5.



Results as per selected profile

Figure 5: Web user interface of the HbbTV/DVB conformance software.

### 4.3 HbbTV/DVB Live Conformance

HbbTV/DVB validation for live services can be reached via the front-end. Whenever a dynamic live MPD is detected, an option for the live validation will be displayed on the front-end, named as “Dynamic timing validation” as shown in Figure 6.

When clicked on, you will be directed to the live validation tool in a new tab, where the provided live MPD is taken as input. This part of the process is shown in Figure 7.

When clicked on “Start” button on this tool, the tool will compute the available segments from the MPD along with their availability end times and start sending “HEAD” requests to check if the computed segment URLs are actually available on the server and if they are actually not available on the computed end time. If yes, a “STATUS OK” response will be displayed on the screen for the specific segment along with the segment availability start time information. If not available, “STATUS Not Found” response will be displayed for the specific segment along with the segment availability start time information. The segment status information is provided on the web interface under “Response information for the segment requests”. Under “Overall progress of segment requests”, information on how many availability start and end checks are performed and how many of them are successful are provided. Only if all the responses are “STATUS OK”, the live service is said to be conformant. This process is shown in Figure 8.

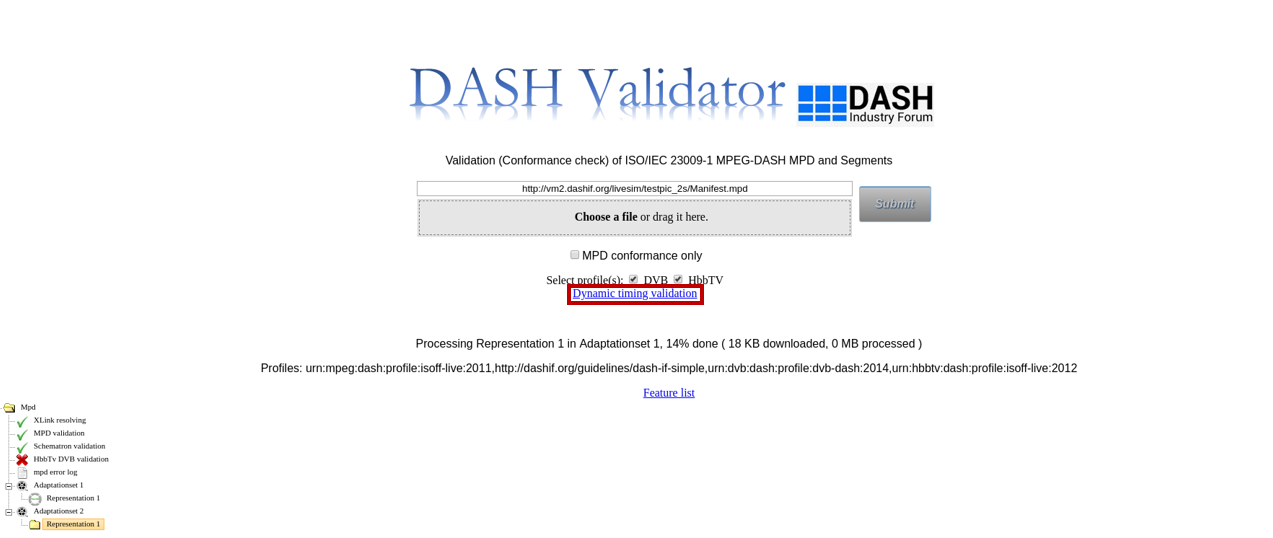


Figure 6: Access to live service validation from the front-end



Figure7: Live service validation tool

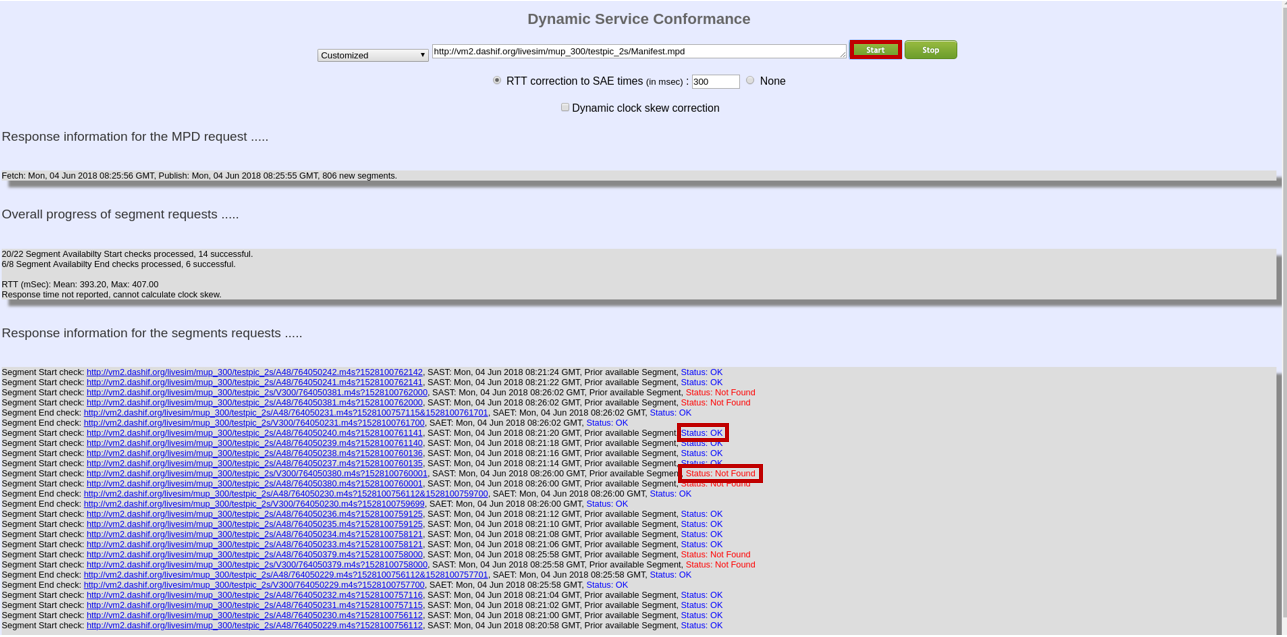


Figure 8: Live service validation process

References

[1] ISO/IEC 23009-2, “Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 2: Conformance and reference software”

|  |
| --- |
| [2] HbbTV, “HbbTV 1.5 including Errata,” 07 06 2016. |
| [3] DVB, “ETSI TS 103 285 V1.1.1 (2015-05): "Digital Video Broadcasting (DVB); MPEG-DASH Profile for Transport of ISO BMFF Based DVB Services over IP Based Networks",” 04 05 2015. |