

DVB DASH Conformance-Software

Nomor Research GmbH
Munich, Germany
info@nomor.de

12 July 2018

Table of Contents

1	Introduction	4
2	High level overview	4
3	Detailed architecture of DASH conformance	5
3.1	The Components of the Architecture.....	6
3.1.1	Client	6
3.1.2	Conformance Server.....	6
3.1.3	Content Server.....	6
3.2	How it works	7
3.2.1	Client	7
3.2.2	Conformance Server / Validation Software	9
4	Architecture of HbbTV-DVB extension.....	12
4.1	Additional components.....	12
4.1.1	Extension to the Client	13
4.1.2	Extension in the Conformance Server.....	13
4.2	How it works	13
4.2.1	Client	13
4.2.2	Conformance Server / Validation Software	15
5	Live Conformance tool / Dynamic Service Validator	16
5.1	Live Conformance tool for DASH	16
5.1.1	Architecture and flow of Live conformance testing	17
	17
5.1.2	Components and their description.....	18
5.2	Live conformance tool – HbbTV/DVB extension.....	20

Table of Figures

Figure 1 High level overview of Conformance-Software	4
Figure 2 Architecture of DASH Conformance Software	5
Figure 3 Screenshot of conformance client/webpage showing input options	7
Figure 4 Screenshot of webpage showing progress status	8
Figure 5 Screenshot of webpage showing conformance results	9
Figure 6 Architecture of Conformance Software with HbbTV-DVB extension	12
Figure 7 Screenshot of webpage showing profile selection	14
Figure 8 Screenshot of webpage showing HbbTV-DVB specific results	15
Figure 9 Access to live service validation from the front-end	16
Figure 10 Architecture and flow of live DASH conformance testing tool	17
Figure 11 Screenshot of live conformance webpage	19
Figure 12 Architecture and flow of live conformance with HbbTV-DVB extension	21

1 Introduction

This document provides information about the conformance software tool, specifically what its scope is and how it works.

The tool is initially designed for testing the conformity of the provided DASH manifest file against MPEG-DASH specification [1], ISO BMFF specification [2] and DASH IOP guidelines [3]. This version of the tool extends the tool to support the conformance against HbbTV / DVB-DASH specifications, [4] and [5], which build upon DASH specification.

In the following sections, more details on DASH conformance tool and DVB DASH conformance extensions are provided. A section on live conformance is also added.

2 High level overview

DASH conformance software performs the task of validating at least the DASH manifest (also known as MPD-Media Presentation Description) and also the media segments (audio/video/subtitles) pointed to by that MPD, as shown in Figure 1 [6]. If there is any conformance issue found, it is reported back.

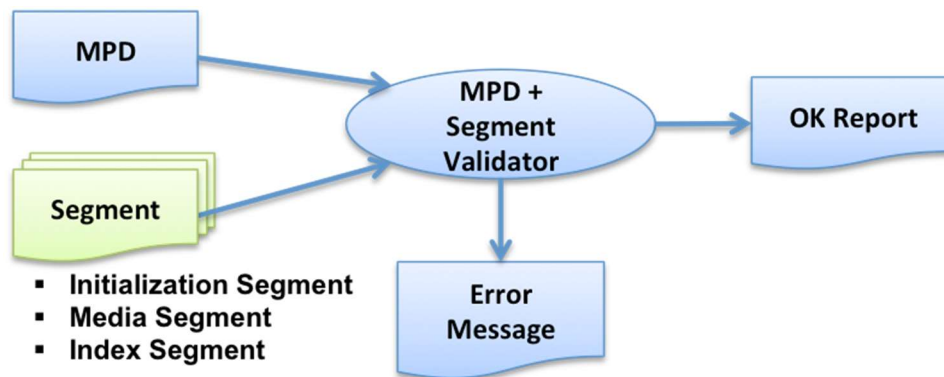


Figure 1 High level overview of Conformance-Software

3 Detailed architecture of DASH conformance

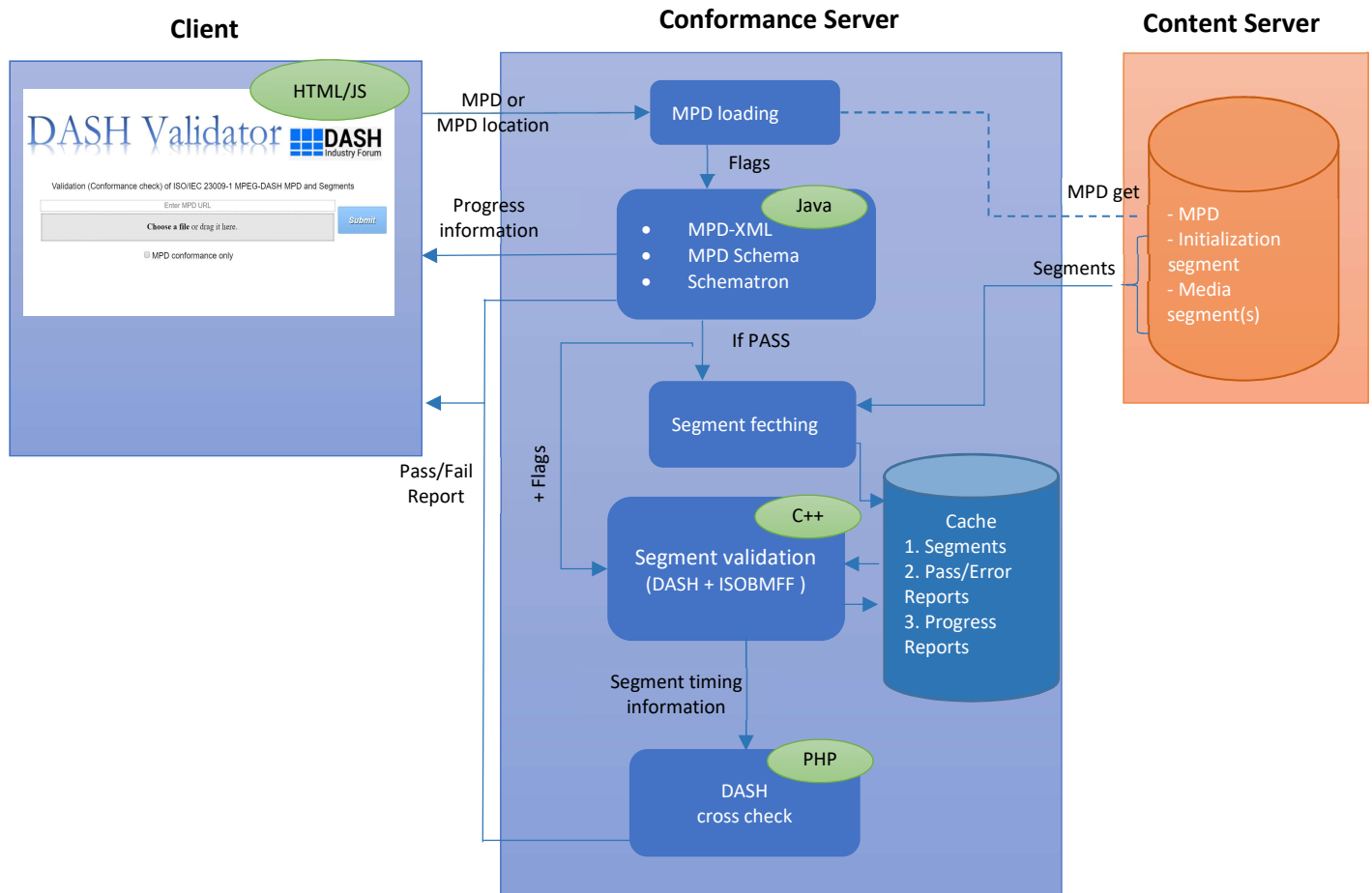


Figure 2 Architecture of DASH Conformance Software

Figure 2 depicts the conformance software architecture with detailed process flow. The main components are

1. Client
2. Conformance Server
3. Content Server

The Client initiates the conformance test by providing either the MPD URL or MPD itself and interacts with the Conformance Server to check the results. The MPD and segments to be

conformed against the abovementioned specifications are accessed from the Content Server. Conformance test of the MPD and segments is carried out by the Conformance Server. The following sections give details of each component and its subcomponents.

3.1 The Components of the Architecture

3.1.1 Client

The Client/Web-Interface consists of the following subcomponents:

- DASH MPD input bar
- Drag & drop area to choose local MPD file
- Selection of MPD-only conformance
- Progress status of validation
- Feature list
- Result of the validation- including error reports in case of non-conformity
- Option to report issues

3.1.2 Conformance Server

The Conformance Server runs the validation software and consists of:

- Temporary folder creation for the session
- DASH MPD loading
- MPD validation (XML, DASH schema and MPD rules validity)
- Media content/segment fetching
- Segment Validation (ISO BMFF and DASH-related ISO BMFF rules validity)
- DASH-related cross-representation rules validity

3.1.3 Content Server

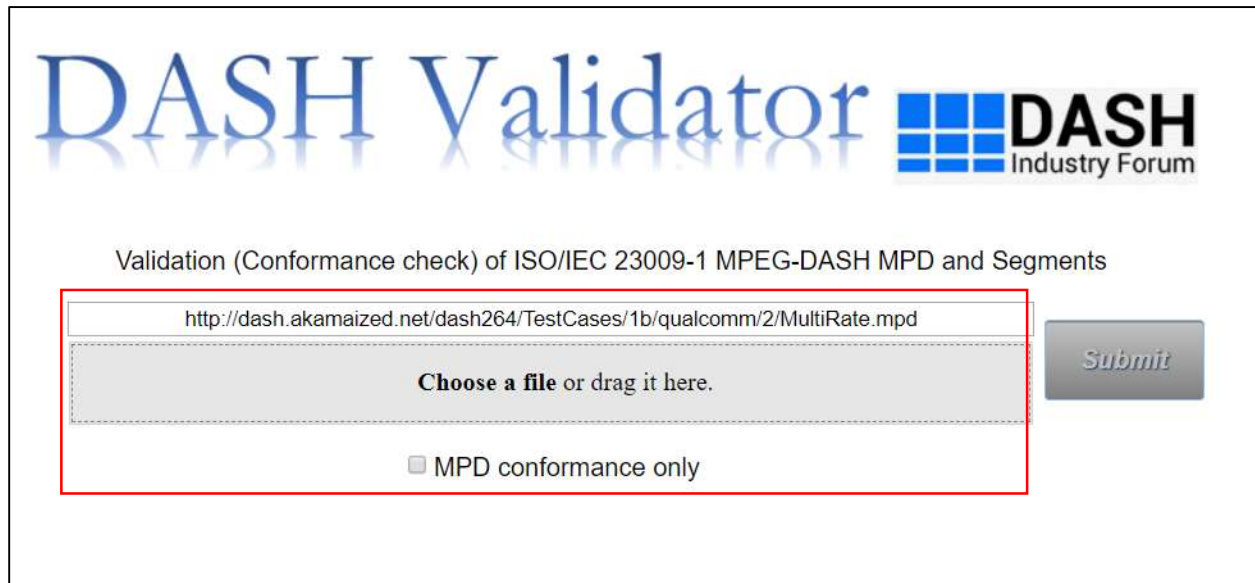
As the name indicates, the Content Server contains the MPD, initialization segments and media segments. The Content Server is a remote location in the Internet as provided by the MPD URL by the user and not a part to be setup for running conformance testing.

3.2 How it works

3.2.1 Client

Conformance software client is the HTML/JavaScript based web user-interface.

A DASH MPD is the input to the UI. MPD can be provided using HTTP URL or 'drag & drop' option for a local MPD file. By default, conformance testing of both MPD and media segments happens when 'Submit' button is pressed. However, the user can select conformance testing of only MPD using the respective checkbox. Figure 3 highlights these components.



The screenshot displays the 'DASH Validator' web interface. At the top, the title 'DASH Validator' is in a large blue font, followed by the 'DASH Industry Forum' logo. Below this, a subtitle reads 'Validation (Conformance check) of ISO/IEC 23009-1 MPEG-DASH MPD and Segments'. The main input area is enclosed in a red rectangular box. It contains a text input field with the URL 'http://dash.akamaized.net/dash264/TestCases/1b/qualcomm/2/MultiRate.mpd'. Below the input field is a dashed border box with the text 'Choose a file or drag it here.' and a checkbox labeled 'MPD conformance only'. To the right of the input area is a grey 'Submit' button.

Figure 3 Screenshot of conformance client/webpage showing input options

Once the conformance test begins, the client interacts with the Conformance Server. The client has the progress status field where the periodic updates from the Server is printed. The status shows which Representation is currently being processed, how many bytes downloaded and how many bytes processed. Figure 4 highlights the progress status field. In addition, the profiles present in the input MPD are also printed below the progress status.

The other component is the 'Feature list'. When clicked on, a new tab opens displaying all the features of the given MPD. The list of all MPD elements and attributes is displayed in a hierarchical tree format.

The screenshot shows the DASH Validator web interface. At the top, the title 'DASH Validator' is displayed in a large blue font, with the 'DASH Industry Forum' logo to its right. Below the title, the text 'Validation (Conformance check) of ISO/IEC 23009-1 MPEG-DASH MPD and Segments' is centered. A text input field contains the URL 'http://dash.akamaized.net/dash264/TestCases/1b/qualcomm/2/MultiRate.mpd'. Below the input field is a dashed box with the text 'Choose a file or drag it here.' and a 'Submit' button to the right. A checkbox labeled 'MPD conformance only' is positioned below the dashed box. A red rectangular box highlights the progress status: 'Processing Representation 1 in Adaptationset 1, 12% done (25 KB downloaded, 9 MB processed)'. Below this, the text 'Profiles: urn:mpeg:dash:profile:isoff-live:2011' is displayed, followed by a blue underlined link 'Feature list'.

DASH Validator

Validation (Conformance check) of ISO/IEC 23009-1 MPEG-DASH MPD and Segments

Choose a file or drag it here.

☐ MPD conformance only

Processing Representation 1 in Adaptationset 1, 12% done (25 KB downloaded, 9 MB processed)

Profiles: urn:mpeg:dash:profile:isoff-live:2011

[Feature list](#)

Figure 4 Screenshot of webpage showing progress status

At the left side of the webpage, the results of the conformance testing are updated. A column consisting of MPD validation results followed by segment validation results are displayed. Segment validation results are displayed in a tree format displaying results of each Representation of each Adaptation Set and cross representation results. When a non-conformity occurs in any Representation, the link to the error log report is provided below the corresponding Representation. Figure 5 highlights the result display format.

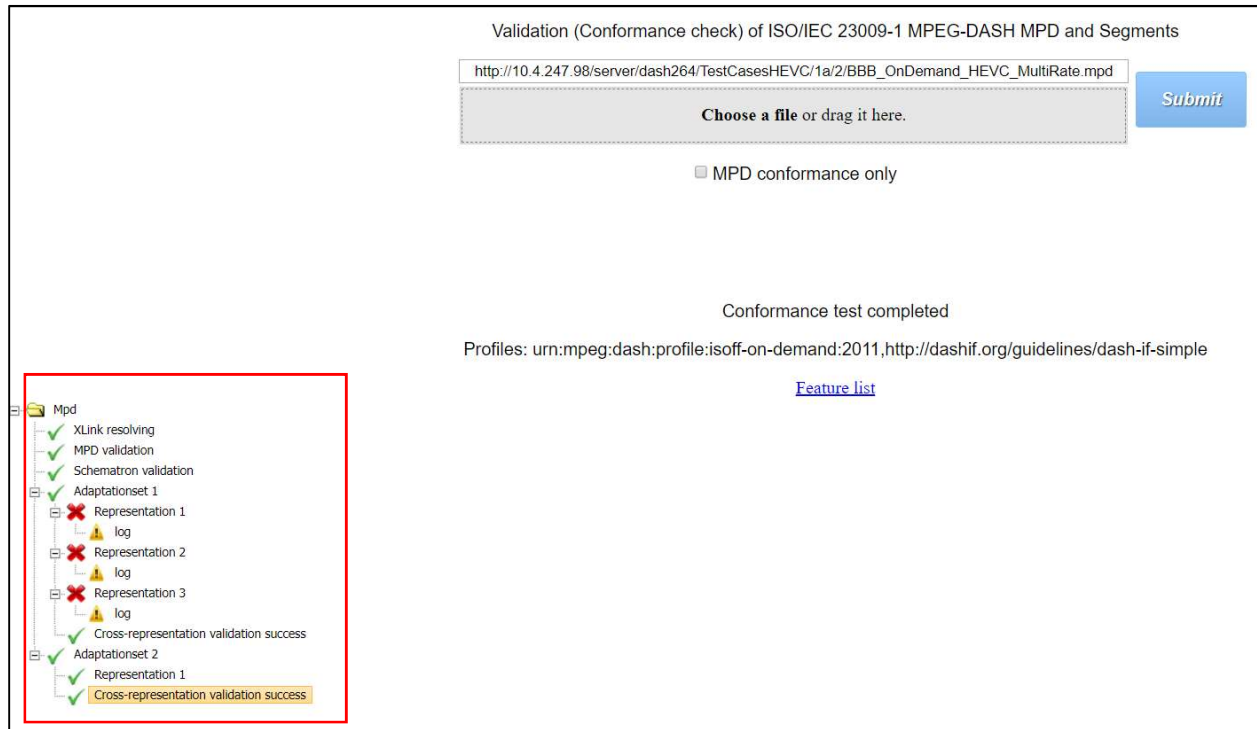


Figure 5 Screenshot of webpage showing conformance results

At the bottom of the webpage, the option to report any software issues is provided. It's a link to the Conformance-Software's Github issue page where new issues can be opened which then will be resolved by the Software team.

3.2.2 Conformance Server / Validation Software

When the conformance test begins, a temporary session folder is created in the server. The intermediate and the final results of the given MPD are stored in this session folder. The rest of the steps are explained in detail in the following sub sections. These steps are according to the flow indicated in the . The only addition is the server module written in PHP which gets the NTP/SNTP server time. This module is added in order to provide UTCTiming element support to the live conformance tool. The address of the NTP/SNTP server is provided in the @value attribute of UTCTiming element. The obtained server time is used to sync local client time and this is followed by the processing steps 3 onwards provided in the section 5.1.2.2.

3.2.2.1 MPD loading

The MPD is fetched from the Content Server using the URL provided by the client. The contents of the MPD is stored in the session folder and will be processed. If a local MPD file is given, then it will be copied to the session folder and will be processed.

3.2.2.2 MPD Validation

The MPD is validated in three stages.

- Xlink resolving
- MPD Schema validation
- Schematron validation

As the name suggests, in the first step, all the xlinks in the MPD (XML) – if any – are resolved. In the next step, the MPD is validated against DASH schema as specified in the DASH specification [1]. The final step is the schematron validation where the rules specified in DASH specification [1] for MPD elements and attributes are checked. The complete MPD validation is built over Java.

In each step of the MPD validation, pass/fail result is produced along with the mention of the error; and these results are conveyed to the client as shown in the . The only addition is the server module written in PHP which gets the NTP/SNTP server time. This module is added in order to provide UTCTiming element support to the live conformance tool. The address of the NTP/SNTP server is provided in the @value attribute of UTCTiming element. The obtained server time is used to sync local client time and this is followed by the processing steps 3 onwards provided in the section 5.1.2.2.

After the validation, the elements and attributes of MPD are stored locally in the program which would later be communicated with Segment validator for cross checks. They are given a general term as 'flags'.

3.2.2.3 Media content/segment fetching

When the MPD validation is successful, i.e., all the steps in section 3.2.2.2 are passed, and if the segment validation is also desired, the media content/segments pointed to by the MPD are downloaded one by one from the Content Server. The segments of all Representations provided in the MPD are downloaded sequentially. During this download, the tool skips the 'mdat' boxes as it checks only the metadata. The downloaded segments are stored in the session folder in the local cache server shown as 'Cache' in the architecture in . The only addition is the server module

written in PHP which gets the NTP/SNTP server time. This module is added in order to provide UTCTiming element support to the live conformance tool. The address of the NTP/SNTP server is provided in the @value attribute of UTCTiming element. The obtained server time is used to sync local client time and this is followed by the processing steps 3 onwards provided in the section 5.1.2.2.

3.2.2.4 Segment Validation

Segment validation is performed for segments of each Representation separately by invoking the C++ validator program (also referred as 'backend'). The inputs for the segment validation are the segments and the flags provided from MPD processing of section 3.2.2.2. In this step, the ISOBMFF specific checks and DASH-specific rules are validated [1][2]. The errors in the case of non-conformity (failing one or more checks) are collected in a log file and stored in the session folder. The error reports are conveyed to the client for the display on the webpage. Only if all the checks are passed, a 'success' is displayed on the webpage for that Representation.

3.2.2.5 Cross-representation checks

After individual Representation validation of an Adaptation Set, the cross-representation validation is performed as required by the DASH specification [1], using PHP in the Conformance Server. This step involves checking the segments' timing and alignment across Representations. An error report is generated in case of non-conformity and conveyed to the client for display.

4 Architecture of HbbTV-DVB extension

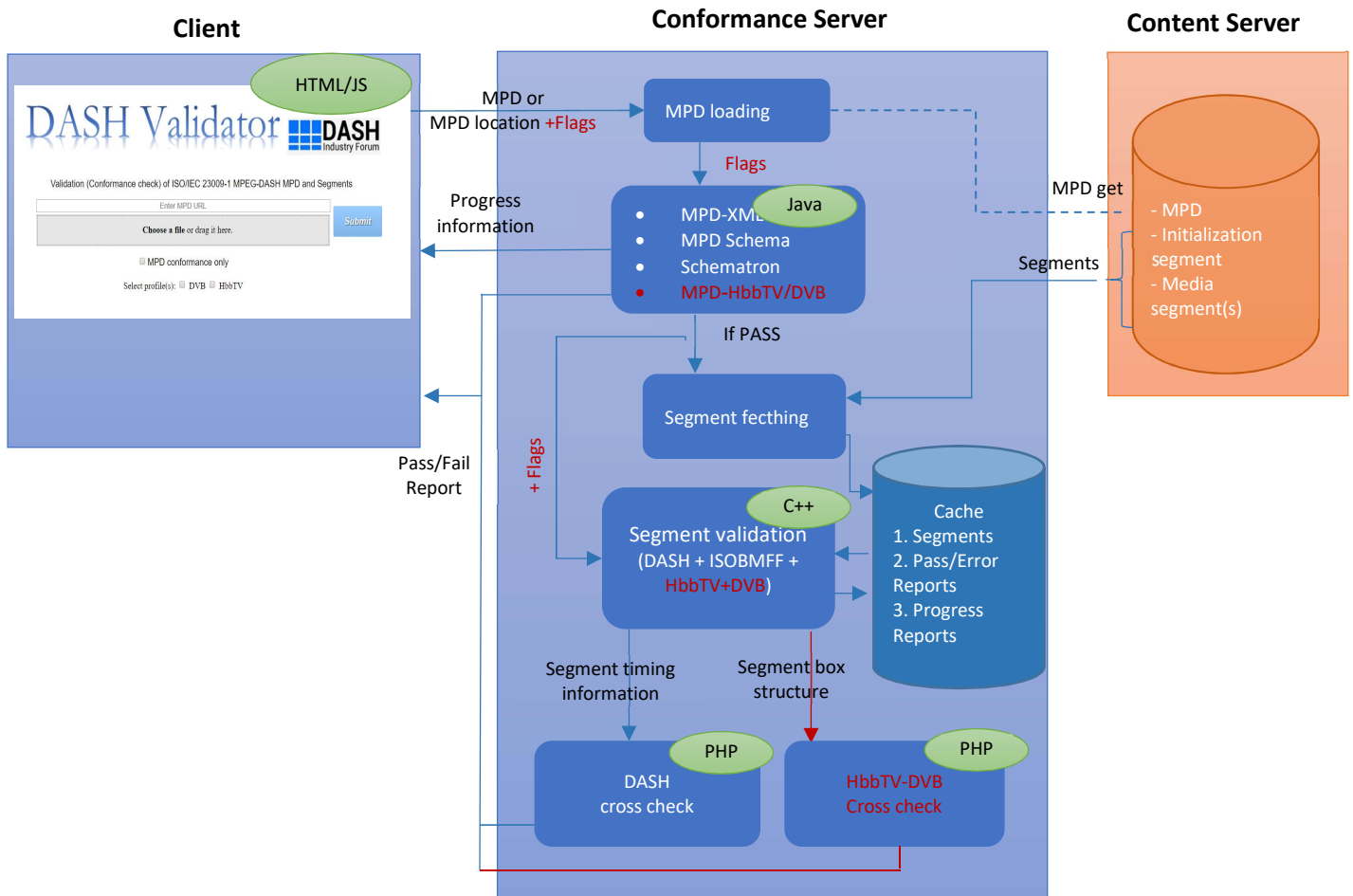


Figure 6 Architecture of Conformance Software with HbbTV-DVB extension

Figure 6 shows the DASH conformance architecture extended for HbbTV-DVB specifications [4][5]. All the components of DASH conformance exist and new additions are applied to the Client and the Conformance Server (highlighted in red).

4.1 Additional components

In this section, only the additional components added to the DASH conformance architecture are mentioned.

4.1.1 Extension to the Client

The additional components in the client are:

- User selection of profiles for validation
- MPD validation results specific to HbbTV-DVB
- Cross representation results specific to HbbTV-DVB

4.1.2 Extension in the Conformance Server

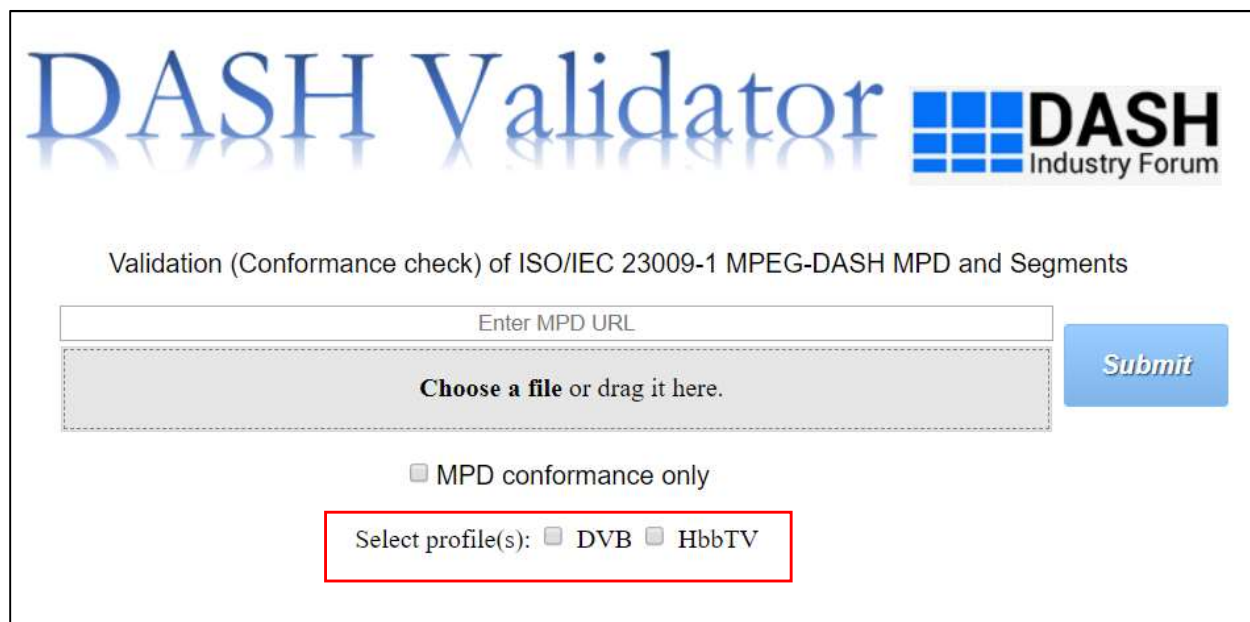
- Flags for signaling HbbTV/DVB specific validation
- MPD validation specific to HbbTV-DVB
- Segment validation specific to HbbTV-DVB
- Cross-representation validation specific to HbbTV-DVB

4.2 How it works

In this section, the working of the additional components onto the DASH conformance are explained.

4.2.1 Client

Once the MPD input is provided, the user has an option (checkbox as in Figure 7) to choose profiles (HbbTV or DVB or both) to validate the given MPD against. This is an additional validation where user can validate an MPD with respect to profile other than the one present in the MPD's @profiles attribute.



The screenshot shows the DASH Validator web interface. At the top, the title "DASH Validator" is displayed in a large blue serif font, with the DASH Industry Forum logo to its right. Below the title, the text "Validation (Conformance check) of ISO/IEC 23009-1 MPEG-DASH MPD and Segments" is centered. A form area contains a text input field labeled "Enter MPD URL", a large dashed box labeled "Choose a file or drag it here.", and a blue "Submit" button. Below these, there is a checkbox labeled "MPD conformance only". At the bottom, a red rectangular box highlights the "Select profile(s):" section, which includes two checkboxes: "DVB" and "HbbTV".

Figure 7 Screenshot of webpage showing profile selection

In the results section of the webpage, the pass/fail status of MPD validation results specific to HbbTV and DVB specifications is displayed separately after DASH specific MPD results. In addition, link to the MPD log report is also provided. Another addition in the result section, is the cross-representation validation results specific to HbbTV-DVB for each of the Adaptation Set as provided in Figure 8.

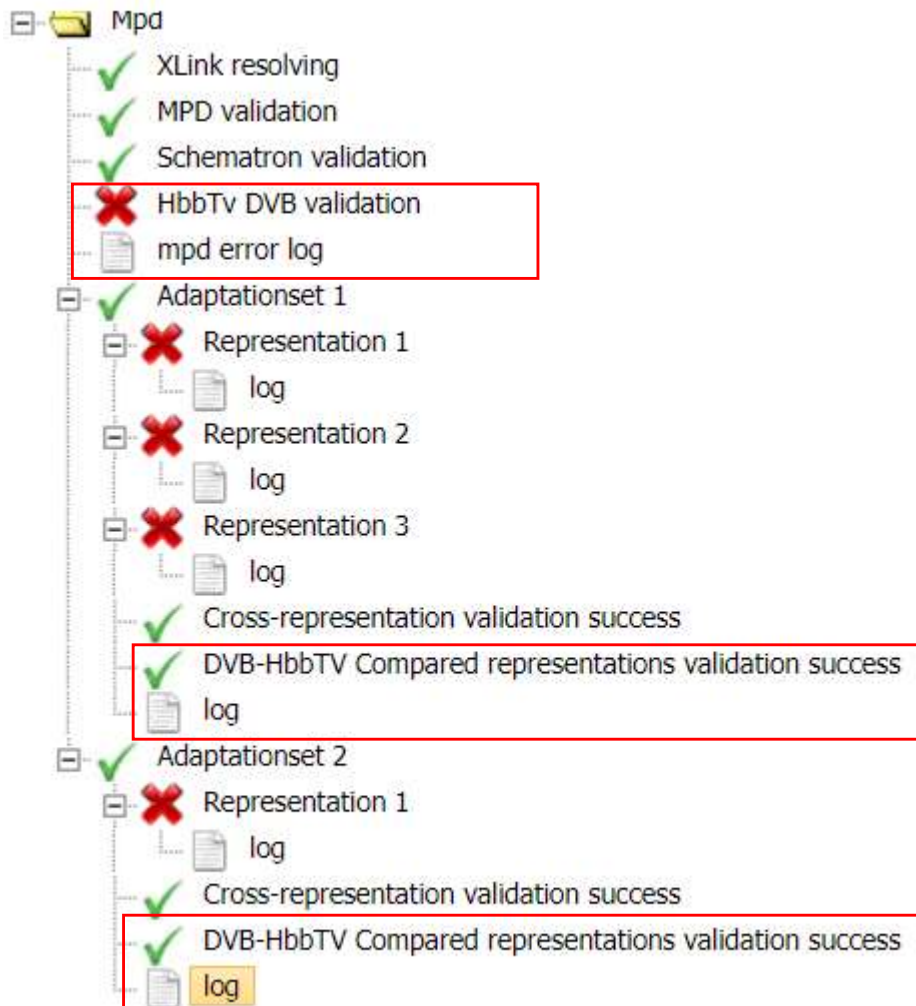


Figure 8 Screenshot of webpage showing HbbTV-DVB specific results

4.2.2 Conformance Server / Validation Software

Separate flags (highlighted in red in Figure 6) are used to signal HbbTV or DVB validation in the conformance software. Validation for the HbbTV or DVB is by default considered from the @profiles attribute of the given MPD. Additionally, the user selected profiles from the client is also used for validation process.

In the MPD validation phase of the software, after DASH-related validation (xlink, schema and schematron validation) is completed, HbbTV- and DVB-specific MPD rules are checked. The results including information, warnings and errors in case of non-conformity are stored in the log report file in the session folder. This report is conveyed to the client for display.

Similar to MPD validation, segment validation (C++) also has additional checks specific to new extensions and they are executed according to the flags signaled to the validator. The errors/warnings are added to the Representation-specific log report as described in section 3.2.2.4. In addition, the segment validator stores the complete ISO BMFF box structure from the provided segments of each Representation separately in XML format in the session folder.

The cross-representation checks specific to HbbTV-DVB are carried out in PHP using the segment box structures stored in the previous step. The results are stored and then displayed on the webpage as in Figure 8.

5 Live Conformance tool / Dynamic Service Validator

5.1 Live Conformance tool for DASH

The Conformance Software described in sections 3 and 4 invokes live conformance tool when the provided MPD is of type 'dynamic'. Figure 9 highlights the link to open the live conformance tool.

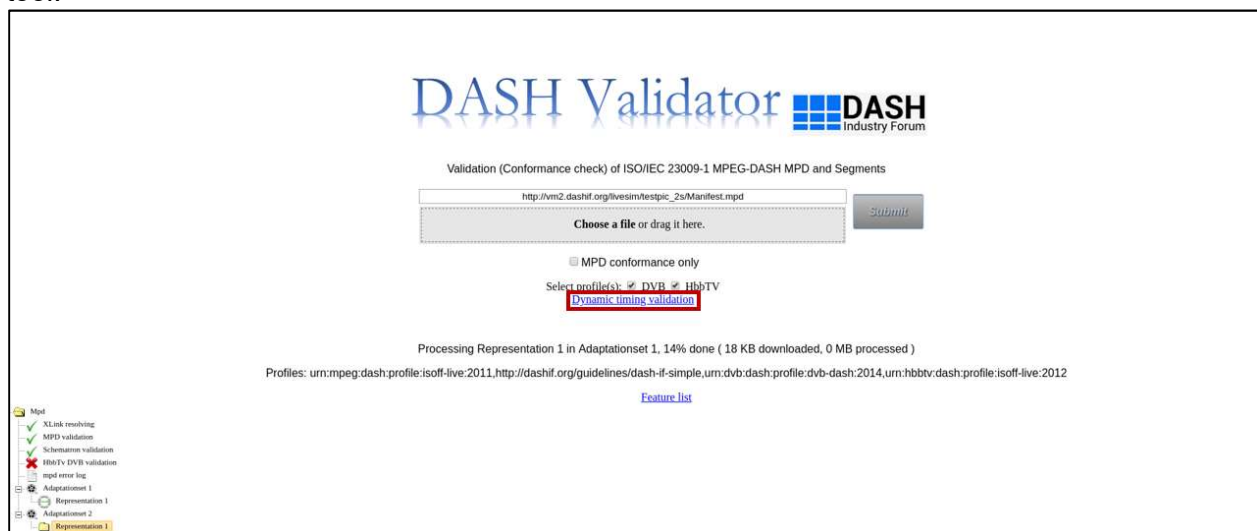


Figure 9 Access to live service validation from the front-end

5.1.1 Architecture and flow of Live conformance testing

Dynamic service conformance tool is a JavaScript-based tool intended to run at or close to the server providing the live service. The main components are User Interface (UI) and processing part as in Figure 10. The components and their working are described in detail in the next section.

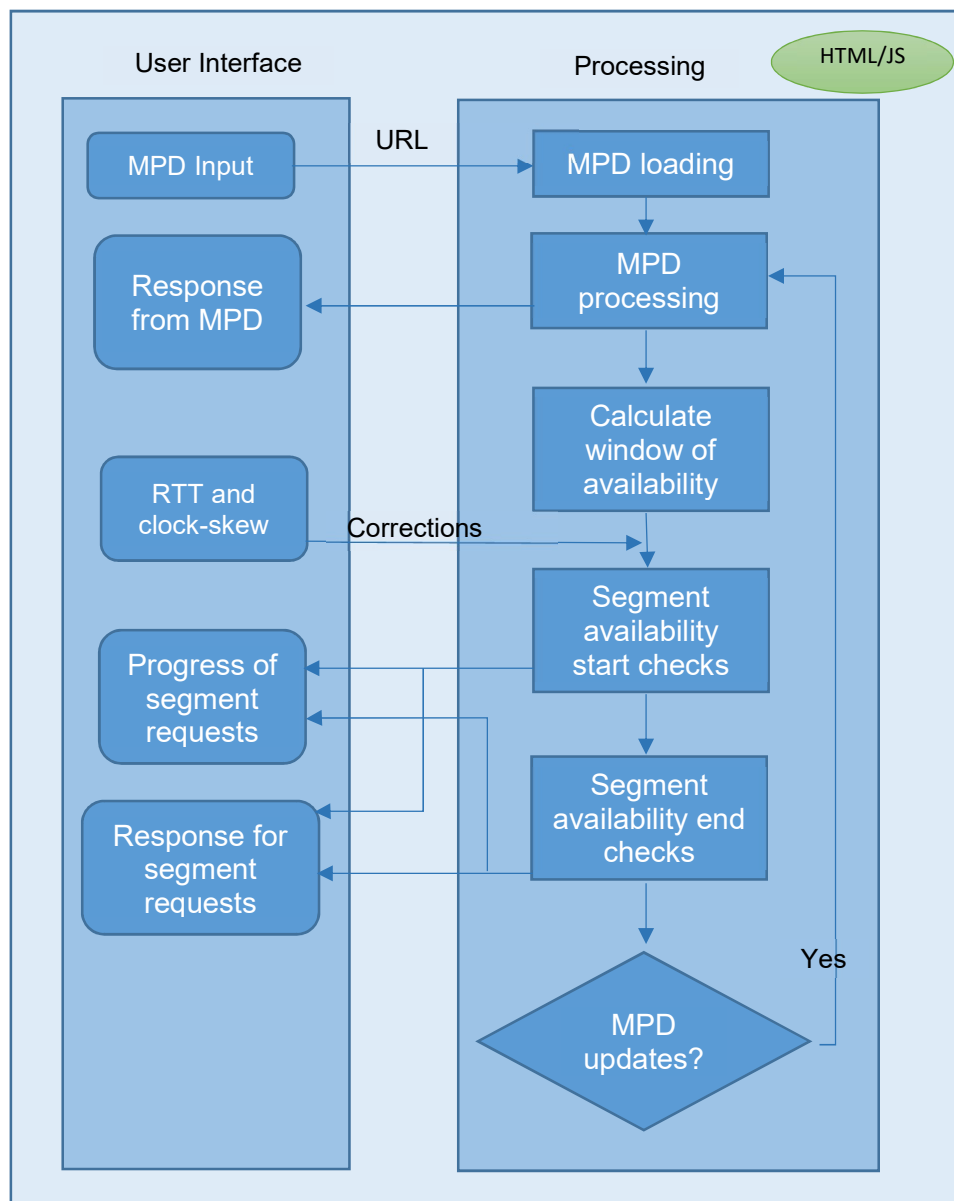


Figure 10 Architecture and flow of live DASH conformance testing tool

5.1.2 Components and their description

As the complete tool runs in HTML/JavaScript, the components are shown logically as 'User-interface' and 'Processing'.

5.1.2.1 User Interface

The UI is divided as follows

1. MPD URL input, RTT and clock skew corrections.
2. Response display for MPD request
3. Progress display for segment requests
4. Response display for segment requests

The following screenshot (Figure 11) shows all the above-mentioned components.

As a first step, MPD URL is provided in the input bar. Round Trip Time (RTT) corrections can be set in msec. Dynamic clock skew correction can also be applied.

When clicked on 'Start' button, the processing of MPD starts and the progress results are displayed in various sections on the webpage. The MPD fetch time, publish time and current window/number of available segments are displayed in area 2 of Figure 11. Information on number of successful segment checks are displayed in area 3 of Figure 11 along with the calculated mean RTT and clock skew. The response for the segment requests (availability start time and availability end time checks) is displayed in area 4.

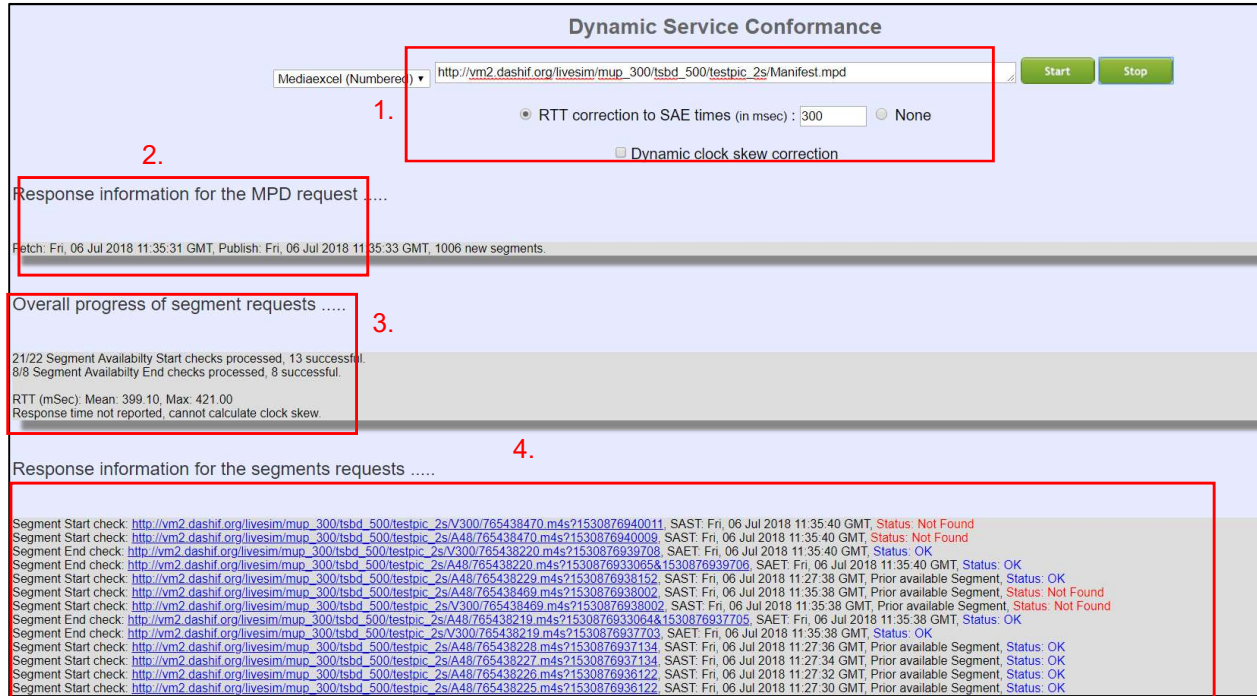


Figure 11 Screenshot of live conformance webpage

5.1.2.2 Processing

Once the MPD is provided, the following processing happens as depicted in Figure 10.

1. MPD is loaded by using HTTP GET request.
2. The loaded MPD is processed, i.e., Period, Adaptation Set and Representation, Segment Template and or Segment Timeline information are extracted.
3. Window of availability is calculated from the timing information present in the MPD. The segment list is constructed for all the segments within the availability window, using the extracted information from step 2.
4. Segment availability start time checks are performed by using HTTP HEAD requests for each of the segments from the constructed list at the start of the availability window. RTT and clock skews are considered while sending segment requests.
5. Similar to step 4, segment availability end time checks are performed.
6. When the MPD is updated according to MPD update interval, the steps from 2 to 5 are repeated.

5.2 Live conformance tool – HbbTV/DVB extension

The flow of live conformance testing for HbbTV-DVB conformance is shown in Figure 12. The only addition is the server module written in PHP which gets the NTP/SNTP server time. This module is added in order to provide UTCTiming element support to the live conformance tool. The address of the NTP/SNTP server is provided in the @value attribute of UTCTiming element. The obtained server time is used to sync local client time and this is followed by the processing steps 3 onwards provided in the section 5.1.2.2.

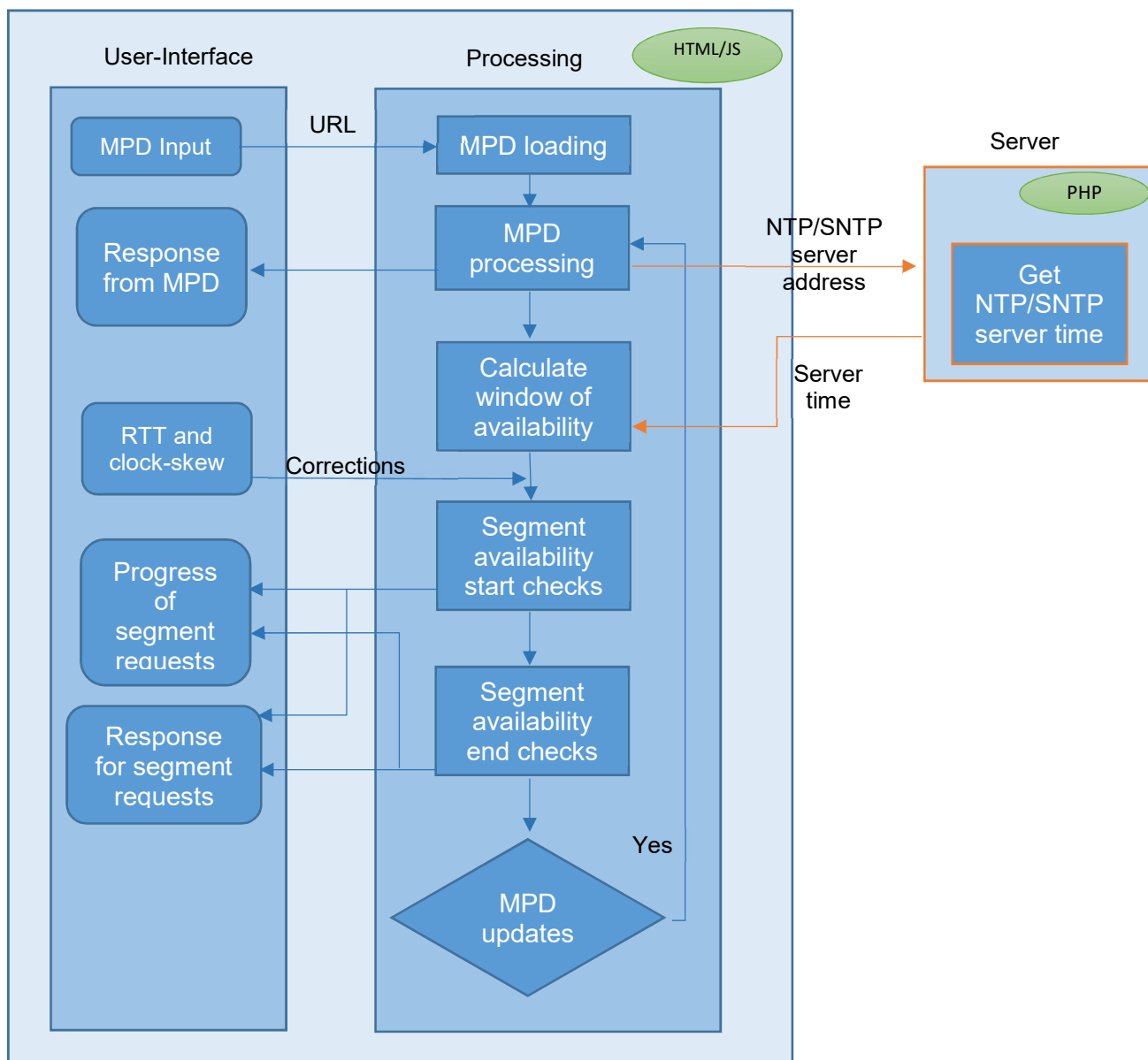


Figure 12 Architecture and flow of live DASH conformance with HbbTV-DVB extensions

References

1. ISO/IEC 23009-1, "Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats"
2. ISO/IEC 14496-12, "Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format"
3. Guidelines for implementation: DASH Industry Forum- Interoperability points v4.1- 'https://dashif.org/wp-content/uploads/2017/09/DASH-IF-IOP-v4.1-clean.pdf'
4. HbbTV, "HbbTV 1.5 including Errata," 07 06 2016.
5. 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.
6. ISO/IEC 23009-2, "Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 2: Conformance and reference software"