Streaming Media Test Suite – Devices Explained

Summary

The WAVE Project has launched its initial “WAVE Streaming Media Test Suite – Devices” to enable automated testing of web-based media playback on various devices like Smart TVs, media sticks, smart phones, laptops and set-top boxes.

This suite aims to streamline testing processes by allowing device implementers to prove compatibility with industry standards, reducing repetitive tests across the industry.

It features unit tests for media using CMAF formats and web streaming technologies like MSE and EME. The suite also includes preliminary tests for popular video and audio codecs, with plans to expand codec testing.

What?

- Tests for AVC video playback using MSE and EME
  - Each test is the combination of an HTML+JavaScript template with CMAF test content
  - Duplicate tests for 25 / 50 Hz, 30 / 60 Hz and fractional frame rates
  - Primarily AVC profile/level 4.0 ('cfhd') but includes level 4.2 ('chdf') for 1080p50/60
- Tests for HEVC video playback using MSE and EME
  - Re-using HTML+JavaScript from AVC video tests but with different media
  - Cover SDR and some coverage of HDR
- Tests for audio
  - Dolby codecs and MPEG codecs
  - Re-using HTML+JavaScript from AVC video tests but with different media
- Test Runner
  - Based on Web Platform Test (WPT) test runner
    - Also used by CTA WAVE 'Web Media API Snapshot' (WMAS) tests
    - Packaged as a docker image
    - Documentation extensively reviewed and improved in last 12 months
- Observation Framework
  - Analyzes recordings made from running tests on devices for compatibility with MSE spec and CTA-5003 Device Playback Capabilities Specification

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

WAVE Product Family – Licenses Used

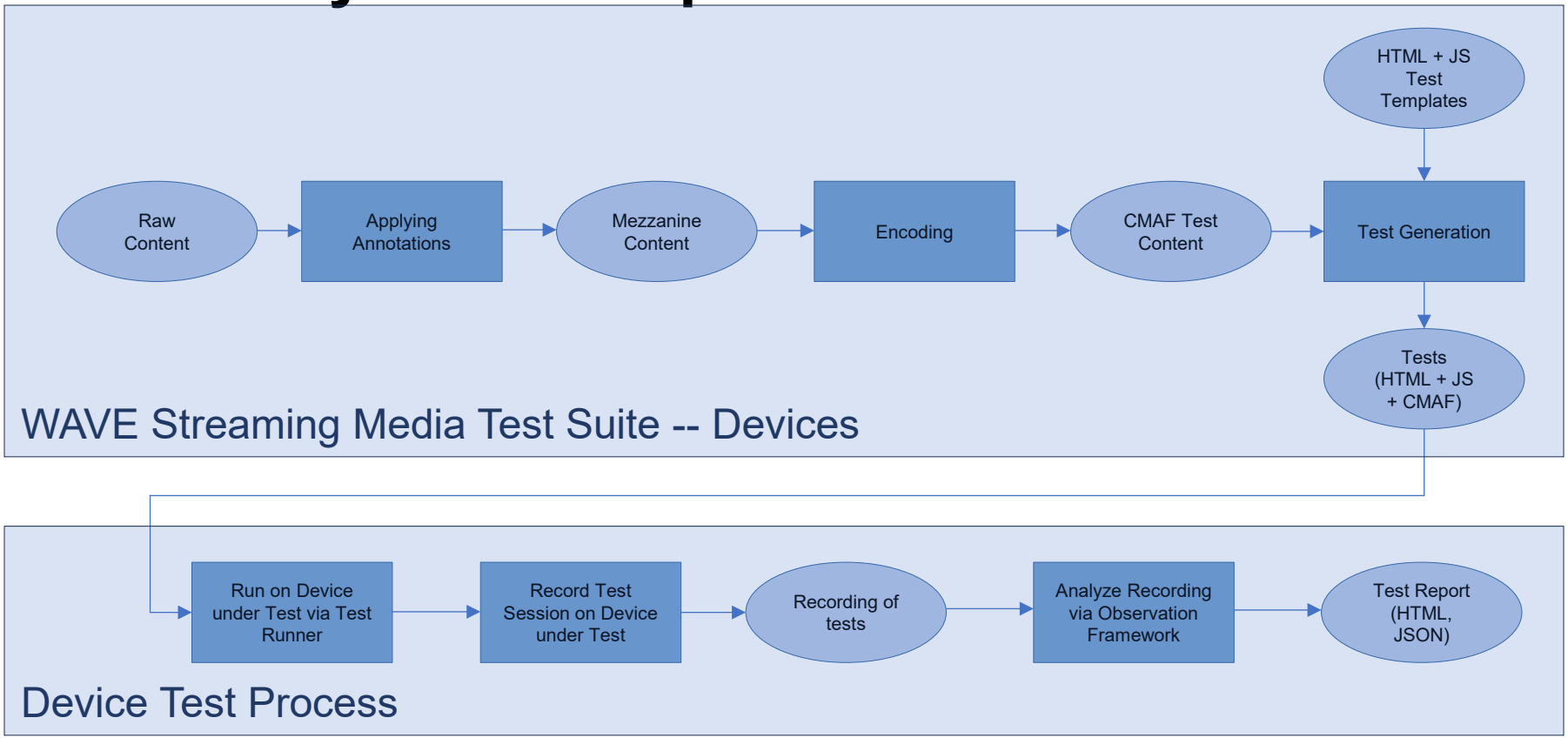
Category	License	Used For
Code	3-Clause BSD	Default for code
	MIT	Code based on W3C Web Platform Tests
	Apache 2.0	Code involving 3rd-party patents
Media	Creative Commons Attribution 3.0	Test streams derived from • *Big Buck Bunny,* or • *Tears of Steel*
	Creative Commons Attribution 4.0	Test streams derived from • *Croatia,* or • WAVE-original content

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Why?

- Improve inter-operability between apps and devices for media presentation using web APIs
  - Strongest contributor interest is Smart TVs
- Reduce combinatorial explosion of testing between apps and devices
  - Enable device implementers to test once and present results to content providers / platforms
- Improve viability of web as a format for media apps

System Components



WAVE
WEB APPLICATION VIDEO ECOSYSTEM

How It Works: Annotated Mezzanine Content

- Annotated video content in many different resolutions
  - Based on big buck bunny for 60Hz and fractional frame rates & an EBU sequence from Croatia for 50Hz.
  - Green frame on the start and red frame on the end.
- Annotations burnt into the video
  - Rotating QR code for observation framework.
  - Human-readable text for debugging.
    - Same information as in the QR code.
  - Bit pattern for TV manufacturer in-house use.
  - Flashing square with beeps for simple A/V sync testing.
  - Red triangles to check all content is visible.
- See next slide for example.
- Audio content based on pseudo-random noise.
  - Observation framework can reconstruct a timeline from this.


I1
00:00:05.360;0000134;25

CMAF Test Content

- Encoded & validated mezzanine content.
  - CMAF media segments.
  - Metadata describing media segments provided in the form of a DASH MPD.
- 4 groups of streams encoded for each technology
  - Testing combinations of content options.
  - Testing content options individually for debugging.
  - Resolutions for testing CMAF switching sets.
  - Special content for testing splicing, encryption, ...
- For each codec, streams are encoded by a list-driven script.
  - Validation in two phases.
  - Script reads in the list and checks each stream has the correct content options, resolution, cmaf profile and other properties.
  - DASH-IF aka ‘JCCP’ validator checks the content is valid CMAF.
- Examples for AVC video on the following slides.

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

HTML and JavaScript Templates

- Media codec / technology independent
  - Exercise MSE (and EME) playback.
  - Algorithms documented in clauses 8 & 9 of WAVE Device Playback Specification.
  - Common library that reads and plays CMAF content based on metadata in DASH MPD.
- Mixture of templates for either video or audio or for both video+audio.
- Examples
  - Sequential Track Playback
    - Play a CMAF video or audio track from beginning to end.
  - Random Access to Fragment, Random Access to Time
    - Play a CMAF video or audio track from somewhere in the middle.
  - Switching Set Playback
    - Play a CMAF switching set, switching between tracks at various points.
  - Playback over WAVE Baseline Splice Constraints
    - Playback switching from one CMAF track to a second and back again.
  - Buffer Underrun and Recovery
    - Playback terminating.

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Test Generation

- A script that encodes test content and also generates combinations of HTML+JS templates to use with test content.
- Script copies templates and inserts URLs for MPDs.

Test Status

- "Validated" tests have met our quality criteria
  - These are the tests that would be referenced by a content / service provider or platform when requiring devices to pass the CTA WAVE test suite.
- "Beta tests"
  - Ones that pass on at least one implementation but do not meet the criteria to become "validated"
- "Secondary tests"
  - For debugging when an implementation fails certain validated tests
- Other tests that do not pass on any implementation at the time

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Test Statistics for V4.0.0

CMAF 4CC	Validated	Beta	Secondary	Other	Total
cfhd,chdf	32	53	6	36	127 (48 25/50Hz, 40 30/60Hz, 39 for fractional frame rates)
ceac, ca4s	15	0	0	7	22
caaa, caac, camc	15	0	0	5	20
chd1, chh1, clg1, cud1	27	0	0	217	244 (54 chd1, 84 chh1, 54 clg1, 52 cud1)
Total	89	53	0	271	413

Numbers are based on "local" variant of tests only, "online" variant is not included.
HEVC tests are new and have not yet had the in-depth analysis of the AVC tests.

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

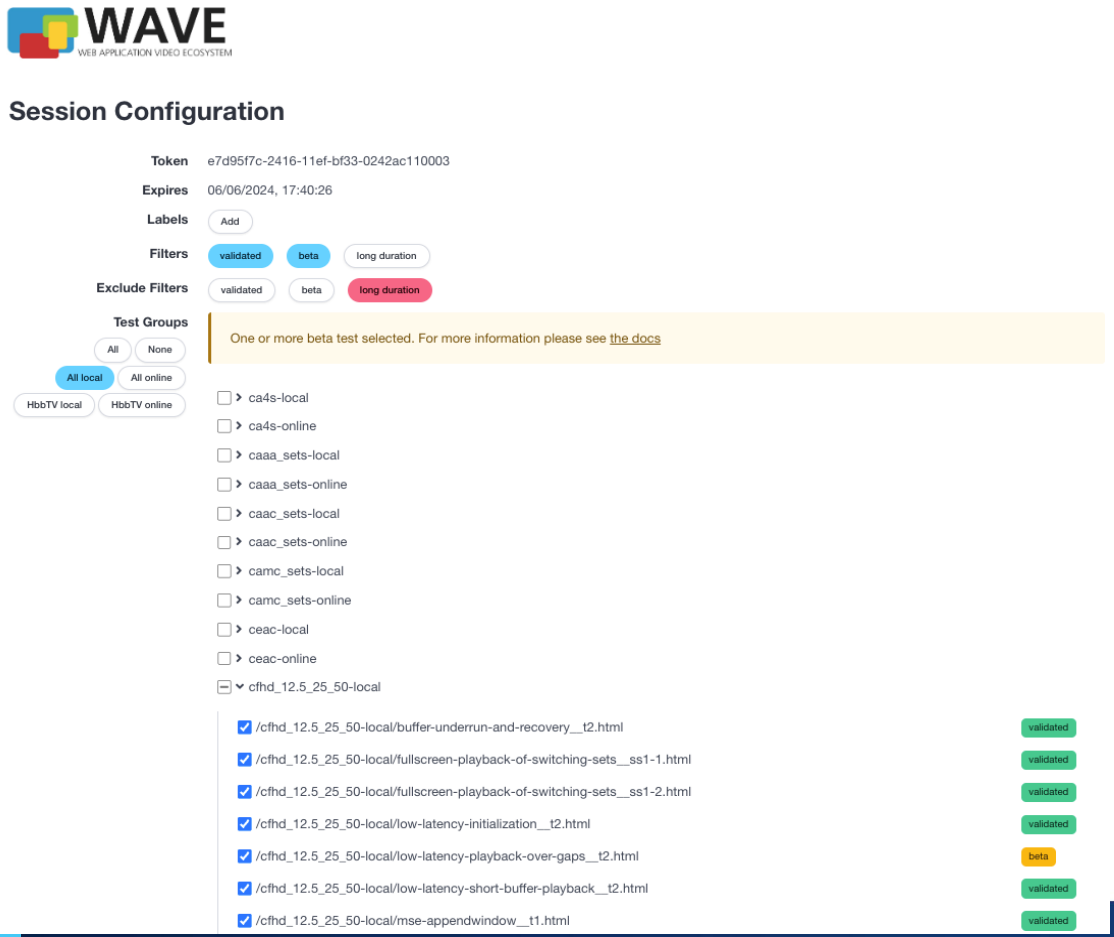
WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Test Runner

- Extended version of test runner used for ‘Web Platform Tests’
  - Device Under Test loads HTML page from the test runner.
  - Test operator connects to test runner with desktop or mobile browser.
- Test operator configures a session by selecting one or more groups of tests.
  - Based on CMAF media profile and/or use by HbbTV.
  - Filtered by validated or beta (see later) or long duration.
- Device runs the session controlled by the test runner.
  - Results and debug output saved to the test runner.
- See next slide for screen shot.

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Test Runner UI for Session Configuration



WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Installer

- Creates two docker images, one with test runner and one with observation framework.
- Can be run on Unix-like systems
  - Linux, Mac, Windows with WSL2 (without Docker Desktop)
- Installation needs someone who knows what they’re doing with IP routing, TLS server certificates, ...
  - Either someone from IT or a local power user.
  - Ensuring the test runner can talk to the Device Under Test and to the Observation Framework is not straightforward in some configurations.

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Camera

- Recording & analyzing 50/60Hz video playback needs camera recording at 120Hz
  - Validation done with top of the range mobile phone, e.g. Samsung S23+ or equivalent, or GoPro 11+
    - Care needed as phones may have issues with adequately obtaining audio from TV.
  - Experiments can limit video <= 30Hz and use lower-spec cameras.
- Avoiding reflections and glare when setting up camera is very important
  - Otherwise, Observation Framework will fail to extract QR codes.
- Recording made to local memory card.
  - Downloaded to PC by moving the memory card across.
- Investigations in progress on streaming from camera to test runner.
  - Cameras likely to be more expensive professional equipment.
  - Mobile phone video streaming typically has limitations compared to recording on memory cards.

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Observation Framework

- Processes recordings from the camera and reports results to the test runner.
  - Results are merged into the results from when the test was actually run.
- Two steps
  - Extraction of QR codes from the recording.
  - Processing of data from QR codes against test procedures in the WAVE Device Playback Specification (CTA-5003).
- OF can be run on the same device as runs the test runner or a different (faster) device.
  - OF can be installed either as a docker image or directly.
- OF generates log files and debug output for developers to analyze when a device is reported as failing.

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Example of Failing Observations

cfhd_12.5_25_50-local: All Results

Test files: 1; **Total subtests**: 6

Test Files

1. /cfhd_12.5_25_50-local/low-latency-playback-over-gaps__t2.html

Test Show/Hide Messages	Xx01
/cfhd_12.5_25_50-local/low-latency-playback-over-gaps__t2.html	
Test workflow	PASS
[OF] Every video frame S[k,s] shall be rendered and the video frames shall be rendered in increasing presentation time order.	FAIL
Xx01: First frame found is 8, expected to start from 1. First frame number tolerance is 0. Last frame found is 251, expected to end at 750. Last frame number tolerance is 0. Mid frame number tolerance is 10. Total of missing frame count is 506. Last frame detected before gap 115 exceeded 'stall_tolerance_margin'=7.5 frames of expected frame 125.	
[OF] Video: The playback duration shall match the duration of the CMAF Track	FAIL
Xx01: Playback duration 10089.88ms does not match expected duration 9760.0ms +/- tolerance of 50ms. Detected duration is different by 329.88ms. Allowed tolerance is 50ms and duration frame tolerance is 0. Starting missing frame number is 7. Ending missing frame number is 499.	
[OF] Video: The presented sample shall match the one reported by the currentTime value within the tolerance.	PASS
Xx01: Total failure count is 0. Tolerances: +/-(1 frame(s) + 150ms.)	
video ended event fired	PASS
video remains in waiting state until skipping over the gap	PASS

From https://github.com/cta-wave/dpctf-tests/issues/152

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Debugging a Failing Test

- Start your favorite video editor, go to the frame number from the log files and step through.
- Record what happens at what frame number (e.g.)
- Has the OF correctly failed the recording according to the test procedure?
- Is the test procedure wrong?
- Is the device wrong?

	A	B	C	D	E	F
1	Frame	Event				
2	73557	"Next test is about to start"				
3	74210	s: waiting; a: initialize				
4	74253	s: waiting; a: initialize; with app drawn QR code				
5	74271	s: ready; a: initiialize' ct: 0;				
6	74337	s: buffering; a: initialize; ct: 0;				
7	74648	s: playing; a: play; ct: 0;				
8	74692	s: playing; a: play; ct: 0; - QR code burnt into video detectable				
9	74696	1st frame of croatia detectable				
10	75315	last frame before seek 1st detectable – 00:00:05.240; 0000131;25				
11	76667	1st frame after seek may be detectable from QR code				
12	76669	1st frame after seek fully visible – 00:00:15.040;0000376;25				
13	78462	last frame of croatia first detectable – 00:00:29.960;0000749;25				
14	78463	1st frame of red first detectable – 00:00:30;0000750;25				

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

System Validation

- Tests run on Smart TVs at HbbTV interop event / plugfests.
  - Three times per year since 2023.

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Example Errors Found by this Test Suite

- Dropped frames in the middle of content, either
  - Media timeline is preserved, one frame persists during the period of the dropped frames.
  - Media timeline is not preserved, period of the dropped frames is cut out of the media timeline, frames persist for their expected time.
- Dropped frames at the start and end of the content.
  - Either no video is shown, or broadcast video is shown in the case of a Smart TV.
- Media timeline stalls.
  - Media timeline just stops advancing for some period.
- Errors with JavaScript currentTime property relative to displayed video.
  - Either drift or currentTime continuing to increment after video playback has stopped.
  - Likely results in loss of sync between subtitles / captions and video.
- Seeking starts playback from the wrong location.
- Failure to restart after buffer underflow.

Give It a Try

- Instructions at https://github.com/cta-wave/dpctf-deploy/releases/
- Successfully used with four host configurations
  - Linux (primary supported host environment)
  - Windows using WSL2 (almost the same as Linux)
  - Mac (almost the same as Linux)
- Three phases
  - Deployment (one-time action, to be performed with support from IT personnel or power users)
    - Note it is possible to get yourself tied in knots between virtualization systems unless done by someone who knows about Docker and IP routing between LAN, host and docker instances
    - Note doing this 'for real' requires a TLS server certificate which may be a challenge in a company network. It needs to be done by someone who knows what they're doing.
  - Test execution and recording
  - Observation

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

Acknowledgements

- Romain Bouqueau (Motion Spell) with support from Qualcomm and CTA
- Nicholas Frame, Jon Piesing (TP Vision)
- Louay Bassbouss, Fritz Heiden, Martin Lasak, Ilja Gavrylov (Fraunhofer FOKUS) with support from CTA
- Yan Jiang, Richard Cottingham, Peter Shorrock (Resillion) with support from Dolby, Fraunhofer IIS, CTA
- Thomas Stockhammer (Qualcomm)
- Zachary Cava (Walt Disney Co.)
- Mike Bergman, Bill Rose, Alexandra Blasgen (CTA)

Thank you