Live video timing validator

In technical terms, a live video is the same as any other video, except the provider keeps publishing more data as time goes on, at least until the stream ends. For successful video playback on all types of end-user devices, the timing aspects of this publishing process are critical (players tend to fail if they cannot get the data they expect).

While some tools exist to validate the data itself, there is currently a lack of good tooling in the industry to validate the timing aspects of how/when data is published. This project is aimed at filling this gap.

High level goal

Create a website where the user can enter a video URL, press a submit button and receive validation results. The validation will be "instantaneous", checking the video in the form that currently exists - the app will not monitor/observe the video over a time span.

At minimum, the results should be a pass/fail judgement with a list of errors. Ideally, the validator will also offer diagnostic information that may be useful to the user in understanding timing behavior (e.g. "the video stream uses a globally synchronized clock and started at 12:30 today").

All validation scenarios need to be covered with a good test suite to make it as obvious as possible that the validation logic is operating according to the specification.

Output

The output shall be published on GitHub as free software. The students will retain ownership of the software.

Nature of the expected work

The execution of the software project will involve:

- Creating the validator software.
- Creating automated tests to verify functionality.
- Creating a test data generator to enable tests to be created.
Skills exercised

- Small amount of web development (to make the GUI)
- Small amount of HTTP client scenarios (to download video data)
- Significant amount of specification analysis (to comprehend what is what)
- Significant amount of data processing (to perform the validation)
- Significant amount of test and test data design (to make a good test suite)

Validation rules

The stream validation rules will be defined by Axinom and will largely (though not exclusively) be based on DASH Industry Forum guidelines (https://dashif.org/guidelines/). These rules will range from basic statements like "no referenced URL can return a 404 not found error" to "the tfdt structure in each video segment must contain an accurate and sequentially increasing timestamp according to (some algorithm defined in ISO standard)".

The two video delivery technologies

There are two modern video delivery technologies - HLS is used on Apple devices and DASH is used everywhere else. It is normal to have the same video published concurrently using both technologies. Therefore, the app should support both technologies.

These two technologies use rather different timing models, so they need to be processed based on different principles and validated against different rules. They also need their test data created based on different principles. This will allow for the possibility of relatively easy scope changes during the execution of the project, targeting either one or two of the technologies depending on the level of the students' ambition and available time. DASH is the first preference if one needs to be cut.

Live video streams

A valid DASH and HLS example stream can be found at https://github.com/Axinom/public-test-vectors#live-clear

Invalid streams must be created as part of this project. The DASH live source simulator can generate simulated DASH live streams. Extending this simulator with fault injection capability is likely to be a productive approach leading to a good test suite with a variety of invalid test streams.