Weaving the Web(VTT) of Data

Thomas Steiner*
CNRS, Université de Lyon
LIRIS, UMR5205
Université Lyon 1, France
tsteiner@liris.cnrs.fr

Hannes Mühleisen
Database Architectures Group
CWI, Science Park 123
1098 XG Amsterdam, NL
hannes@cwi.nl

Ruben Verborgh
Multimedia Lab
Ghent University – iMinds
B-9050 Gent, Belgium
ruben.verborgh@ugent.be

Pierre-Antoine Champin
CNRS, Université de Lyon
LIRIS, UMR5205
Université Lyon 1, France
pachampin@liris.cnrs.fr

Benoît Encelle
CNRS, Université de Lyon
LIRIS, UMR5205
Université Lyon 1, France
bencelle@liris.cnrs.fr

Yannick Prié
LINA – UMR 6241 CNRS
Université de Nantes
44322 Nantes Cedex 3
yannick.prie@univ-nantes.fr

ABSTRACT

Video has become a first class citizen on the Web with broad support in all common Web browsers. Where with structured mark-up on webpages we have made the vision of the *Web of Data* a reality, in this paper, we propose a new vision that we name the *Web(VTT) of Data*, alongside with concrete steps to realize this vision. It is based on the evolving standards *WebVTT* for adding timed text tracks to videos and *JSON-LD*, a JSON-based format to serialize Linked Data. Just like the *Web of Data* that is based on the relationships among structured data, the *Web(VTT) of Data* is based on relationships among videos based on WebVTT files, which we use as Web-native spatiotemporal Linked Data containers with JSON-LD payloads. In a first step, we provide necessary background information on the technologies we use. In a second step, we perform a large-scale analysis of the 148 terabyte size Common Crawl corpus in order to get a better understanding of the *status quo* of Web video deployment and address the challenge of integrating the detected videos in the Common Crawl corpus into the *Web(VTT) of Data*. In a third step, we open-source an online video annotation creation and consumption tool, targeted at videos not contained in the Common Crawl corpus and for integrating future video creations, allowing for weaving the *Web(VTT) of Data* tighter, video by video.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Video

Keywords

JSON-LD, Linked Data, media fragments, Semantic Web, video annotation, Web of Data, WebVTT, Web(VTT) of Data

*Second affiliation: *Google Germany GmbH, Hamburg, DE*

LDOW2014, April 8, 2014, Seoul, Korea..

1. INTRODUCTION

1.1 From <OBJECT> to <video>

In the "ancient" times of HTML 4.01 [25], the `<OBJECT>` tag[1] was intended for allowing authors to make use of multimedia features like including images, applets (programs that were automatically downloaded and ran on the user's machine), video clips, and other HTML documents in their pages. The tag was seen as a future-proof all-purpose solution to generic object inclusion. In an `<OBJECT>` tag, HTML authors can specify everything required by an object for its presentation by a user agent: source code, initial values, and run-time data. While most user agents have *"built-in mechanisms for rendering common data types such as text, GIF images, colors, fonts, and a handful of graphic elements"*, to render data types they did not support natively—namely videos—user agents generally ran external applications and depended on plugins like Adobe Flash.[2].

While the above paragraph is provocatively written in past tense and while the `<object>` tag is still part of both the current World Wide Web Consortium (W3C) HTML5 specification [2] and the Web Hypertext Application Technology Working Group (WHATWG) "Living Standard",[3] more and more Web video is now powered by the native and well-standardized `<video>` tag that no longer depends on plugins. What currently still hinders the full adoption of `<video>`, besides some licensing challenges around video codecs, is its lack of Digital Rights Management (DRM) support and the fierce debate around it, albeit the Director of the W3C has confirmed[4] that work in form of the Encrypted Media Extensions [8] on "playback of protected content " was in the scope of the HTML Working Group. However, it can well be said that HTML5 video has finally become a first class Web citizen that all modern browsers fully support.

[1]HTML 4.01 `<OBJECT>` tag (uppercased in the spirit of the epoch): `http://www.w3.org/TR/REC-html40/struct/objects.html#edef-OBJECT`
[2]Adobe Flash: `http://get.adobe.com/flashplayer/`
[3]HTML5 `<object>` tag in the "Living Standard" (now lowercased): `http://www.whatwg.org/specs/web-apps/current-work/#the-object-element`
[4]New Charter for the HTML Working Group: `http://lists.w3.org/Archives/Public/public-html-admin/2013Sep/0129.html`

1.2 Contributions and Paper Structure

We are motivated by the vision of a *Web(VTT) of Data*, a global network of videos and connected content that is based on relationships among videos based on WebVTT files, which we use as Web-native spatiotemporal containers of Linked Data with JSON-LD payloads. The paper makes four contributions, including transparent code and data.

i) **Large-Scale Common Crawl study of the state of Web video:** we have examined the 148 terabyte size Common Crawl corpus and determined statistics on the usage of the `<video>`, `<track>`, and `<source>` tags and their implications for Linked Data.

ii) **WebVTT conversion to RDF-based Linked Data:** we propose a general conversion process for "triplifying" existing WebVTT, *i.e.*, for turning WebVTT into a specialized concrete syntax of RDF. This process is implemented in form of an online conversion tool.

iii) **Online video annotation format and editor:** we have created an online video annotation format and an editor prototype implementing it that serves for the creation and consumption of semantic spatiotemporal video annotations turning videos into Linked Data.

iv) **Data and code:** source code and data are available.

The remainder of the paper is structured as follows. Section 2 provides an overview of the enabling technologies that we require for our approach. Section 3 describes a large-scale study of the state of Web video deployment based on the Common Crawl corpus. Section 4 deals with the integration of existing videos into the *Web(VTT) of Data* through a tool called LinkedVTT. Section 5 presents an online video annotation format and an editor that implements this format. We look at related work in Section 6 and close with conclusions and an outlook on future work in Section 7.

2. TECHNOLOGIES OVERVIEW

In this section, we lay the foundations of the set of technologies that enable our vision of the *Web(VTT) of Data*. The `<track>` tag allows authors to specify explicit external timed text tracks for videos. With the `<source>` tag, authors can specify multiple alternative media resources for a video. Both do not represent anything on their own and are only meaningful as direct child nodes of a `<video>` tag.

Web Video Text Tracks format (WebVTT).

The Web Video Text Tracks format (WebVTT, [24]) is intended for marking up external text track resources mainly for the purpose of captioning video content. The recommended file extension is `vtt`, the MIME type is `text/vtt`. WebVTT files are encoded in UTF-8 and start with the required string `WEBVTT`. Each file consists of items called *cues* that are separated by an empty line. Each cue has a start time and an end time in `hh:mm:ss.milliseconds` format, separated by a stylized ASCII arrow `-->`. The cue payload follows in the line after the cue timings part and can span multiple lines. Typically, the cue payload contains plain text, but can also contain textual data serialization formats like JSON, which later on in the paper we will show is essential for our proposed approach to semantic video annotation. Cues optionally can have unique WebVTT identifiers. WebVTT-compliant Web browsers [9] support five different kinds of WebVTT tracks: `subtitles`, `captions`, `descriptions`, `chapters`, and `metadata`, detailed in Table 1 and specified in HTML5 [2]. In this paper, we are especially interested in text tracks of kind `metadata` that are meant to be used from a scripting context and that are not displayed by user agents. For scripting purposes, the video element has a property called `textTracks` that returns a `TextTrackList` of `TextTrack` members, each of which correspond to track elements. A `TextTrack` has a `cues` property that returns a `TextTrackCueList` of individual `TextTrackCue` items. Important for us, both `TextTrack` and `TextTrackCue` elements can be dynamically generated. Listing 1 shows a sample WebVTT file.

JSON-LD.

The *JavaScript Object Notation*[5] (JSON) is a (despite the name) language-independent textual syntax for serializing objects, arrays, numbers, strings, booleans, and null. *Linked Data* [4] describes a method of publishing structured data so that it can be interlinked and become more useful, which builds upon standard Web technologies such as HTTP, RDF and URIs. Based on top of JSON, the *JavaScript Object Notation for Linked Data* (JSON-LD, [27]) is a method for transporting Linked Data with a smooth upgrade path from JSON to JSON-LD. JSON-LD properties like `title` can be mapped to taxonomic concepts (like `dc:title` from Dublin Core[6]) via so-called data contexts.

[5]JavaScript Object Notation: `http://json.org/`
[6]Dublin Core: `http://dublincore.org/documents/dces/`

```
WEBVTT

00:01.000 --> 00:04.000
Never drink liquid nitrogen.

00:05.000 --> 00:09.000
It will perforate your stomach.
```

Listing 1: Example WebVTT file with two cues

WebVTT Kind	Description and Default Behavior
`subtitles`	Transcription or translation of speech, suitable for when sound is available but not understood. Overlaid on the video.
`captions`	Transcription or translation of the dialogue, sound effects, and other relevant audio information, suitable for when sound is unavailable or not clearly audible. Overlaid on the video; labeled as appropriate for the hard-of-hearing.
`descriptions`	Textual descriptions of the video component of the media resource, intended for audio synthesis when the visual component is obscured, unavailable, or unusable. Synthesized as audio.
`chapters`	Chapter titles, intended to be used for navigating the media resource. Displayed as an interactive (potentially nested) list in the user agent's interface.
`metadata`	Metadata intended for use from script context. Not displayed by user agent.

Table 1: WebVTT text track kinds in HTML5 [2]

Media Fragments URI.

Media Fragments URI [30] specifies a syntax for constructing URIs of media fragments and explains how to handle them over the HTTP protocol. The syntax is based on the specification of name-value pairs that can be used in URI query strings and URI fragment identifiers to restrict a media resource to a certain fragment. Media Fragments URI supports temporal and spatial media fragments. The *temporal dimension* is denoted by the parameter name `t` and specified as an interval with begin time and end time, with the begin time defaulting to 0 seconds and the end time defaulting to the media item's duration. The *spatial dimension* selects a rectangular area of pixels from media items. Rectangles can be specified as pixel coordinates or percentages. Rectangle selection is denoted by the parameter name `xywh`. The value is either `pixel:` or `percent:` followed by four comma-separated integers. The integers denote x, y, *width*, and *height* respectively, with $x = 0$ and $y = 0$ being the top left corner of the media item. If `percent:` is used, x and *width* are interpreted as a percentage of the width of the original media item, y and *height* of the original height.

Ontology for Media Resources.

The Ontology for Media Resources [17] serves to bridge different description methods of media resources and to provide a core set of descriptive properties. It also defines mappings to common metadata formats. Combined with Media Fragments URI, this allows for making ontologically anchored statements about media items and fragments thereof.

3. LARGE-SCALE COMMON CRAWL STUDY OF THE STATE OF WEB VIDEO

Part of the objectives behind the *Web(VTT) of Data* is to create a truly interconnected global network of and between videos containing Linked Data pointers to related content of all sorts, where diverse views are not filtered by the network bubble, but where serendipitously new views can be discovered by taking untrodden Linked Data paths. In order to get there, we have conducted a large-scale study based on the Common Crawl corpus to get a better understanding of the *status quo* of Web video and timed text track deployment.

3.1 Common Crawl

The Common Crawl Foundation[7] is a non-profit organization founded in 2008 by Gil Elbaz. Its objective is to democratize access to Web information by producing and maintaining an open repository of Web crawl data that is universally accessible and analyzable. All Common Crawl data is stored on Amazon Simple Storage Service (Amazon S3)[8] and accessible to anyone via Amazon Elastic Compute Cloud (Amazon EC2),[9] allowing the data to be downloaded in bulk, as well as directly be accessed for map-reduce processing in EC2. The, at time of writing, latest dataset was collected at the end of 2013, contains approximately 2.3 billion webpages and is 148 terabyte in size [11]. Crawl raw data is stored in the Web ARChive format (WARC, [14]), an evolution of the previously used Archive File Format (ARC, [6]), which was developed at the Internet Archive.[10] Each crawl run is hierarchically organized in segments directories that contain the WARC files with the HTTP requests and responses for each fetch, and individual Web Archive Metadata (WAT, [10]) files, which describe the metadata of each request and response. While the Common Crawl corpus gets bigger with each crawl run, it obviously does not represent the "whole Web", which is an illusive concept anyway, given that a simple calendar Web application can produce an infinite number of pages. Common Crawl decides on the to-be-included pages based on an implementation[11] of the PageRank [23] algorithm, albeit the inclusion strategy is unknown—despite the foundation's focus on transparency.

3.2 On the Quest for WebVTT

We have analyzed the entire 148 terabytes of crawl data using an Elastic Compute Cloud job whose code was made available as open-source.[12] Rather than parse each document as HTML, we have tested them for the regular expression `<video[^>]*>(.*?)</video>`, an approach that also in previous experiments proved very efficient [3, 22]. We tested exactly 2,247,615,323 webpages that had returned a successful HTTP response to the Common Crawl bot, and had to skip exactly 46,524,336 non-HTML documents. On these webpages, we detected exactly 2,963,766 `<video>` tags, resulting in a 1.37 gigabyte raw text file that we have made available publicly.[13] This means that on average only ≈0.132% of all webpages contain HTML5 video. The whole job took five hours on 80 c1.xlarge machines and costed $555, consisting of $468 for Amazon EC2, plus an additional $87 for Amazon Elastic MapReduce (Amazon EMR).[14]

3.3 Text Track Statistics

From all 2,963,766 `<video>` tags, only 1,456 (≈ 0.049%) had a `<track>` child node. Upon closer examination of the kinds of these 1,456 `<track>` nodes (see Table 1 for an explanation of the various kinds), we saw that the overwhelming majority are unsurprisingly used for `subtitles` or `captions`. Almost no `chapter` usage was detected and neither `metadata` nor `description` usage at all. The full details can be seen in Table 2. Looking at the languages used in the captions and subtitles, these were almost exclusively English and French, as can be seen in Table 3. The track labels listed in Table 4 indeed confirm this observation. In case of multiple tracks for one video, one track can be marked as the default track. This happens through a boolean attribute,[15] whose value either needs to be the empty string or the attribute's name, which is "default" in the concrete case. Table 5 shows that this was used correctly in almost all cases. When we tried to determine the MIME type of the actual text tracks, we relied on the file extension of the values given in the `<track src>` attributes. As a significant amount of text tracks seems to be dynam-

[7] Common Crawl: `http://commoncrawl.org/`
[8] Amazon S3: `http://aws.amazon.com/s3/`
[9] Amazon EC2: `http://aws.amazon.com/ec2/`
[10] Internet Archive: `https://archive.org/`
[11] Common Crawl PageRank code: `https://github.com/commoncrawl/commoncrawl-crawler/tree/master/src/org/commoncrawl/service/pagerank`
[12] EC2 job: `https://github.com/tomayac/postdoc/blob/master/demos/warczenschwein/`
[13] 2,963,766 `<video>` tags: `https://drive.google.com/file/d/0B9LlSNwL2H8YdWVIQmJDaE81UEk`
[14] Amazon EMR: `http://aws.amazon.com/elasticmapreduce/`
[15] HTML boolean attributes: `http://www.whatwg.org/specs/web-apps/current-work/#boolean-attributes`

ically generated on-the-fly—and thus had no file extension but a video identifier in the URL instead—we used an approximation to check if some part of the URL matched the regular expression `/\bvtt\b/gi`. Based on this approximation, a little over half of all text tracks are in WebVTT format with the extension `.vtt` or rarely `.webvtt`. The predecessor SubRip file format[16] can still be encountered in about a quarter of all text tracks. In between SubRip and WebVTT, a format originally called WebSRT (Web Subtitle Resource Tracks) existed that shared the `.srt` file extension. The full distribution details are available in Table 6. Looking at the number of text tracks per video, almost all videos had only exactly one text track rather than multiple, as detailed in Table 7, meaning that the broad majority of all videos are subtitled or captioned in only one language.

[16]SubRip file format: `http://www.matroska.org/technical/specs/subtitles/srt.html`

<track kind>	Count
captions	915
subtitles	525
chapters	2
undefined	10

Table 2: Distribution of values for <track kind>

<track srclang>	Count
en	1,242
fr	117
de	8
Others	7
undefined	78

Table 3: Distribution of values for <track srclang>

<track label>	Count
English	1,069
Français	117
Others	41
undefined	229

Table 4: Distribution of values for <track label>

<track default>	Count
default	650
''	526
true	1
undefined	279

Table 5: Distribution of values for <track default>

File extensions of <track src>	Count
probably .vtt	696
.srt	390
.vtt or .webvtt	66
no extension	304

Table 6: Distribution of values for <track src>

<track> tags	Count
1	1,446
0	9
9	1

Table 7: Number of <track> tags per <video> tag (zero <track> tags means the <video> tag had an unparseable <track>)

<source> tags	Count
1	826
3	404
2	173
0	49
4	4

Table 8: Number of <source> tags per <video> with <track> (zero <source> tags means the video URL was provided via <video src>; 1,405 videos did not have a src attribute, 51 videos had one)

<source> tags	Count
0	7,828,032
1	1,139,240
3	138,540
4	83,121
2	77,853
6	804
5	179
7	137
8	64
10	22
9	9
13	8
11	6

Table 9: Number of <source> tags per <video> with or without <track> (zero <source> tags means the video URL was provided via <video src>)

<source type>	Count
video/mp4	1,285
video/webm	94
video/x-ms-wmv	10
video/ogg	5
Others	6
undefined	58

Table 10: Distribution of values for <source type> of <video> tags with <track>

<source type>	Count
video/mp4	1,204,744
video/webm	163,715
video/mp4; codecs="avc1.42E01E, mp4a.40.2"	10,700
text/json	2,841
video/flv	2,281
video/x-ms-wmv	2,105
video/flash	2,023
video/ogg	1,529
video/youtube	1,528
application/x-mpegURL	1,257

Table 11: Distribution of values for <source type> of <video> tags with or without <track> (with more than 1,000 occurrences)

3.4 Video Statistics

As in Section 5 we will report on ways to make semantic statements about videos on the Web, we have additionally compiled some video statistics. Unlike with images on the Web, where semantic statements in Resource Description Framework (RDF) can be made based on the image's URL [21], with Web video, the situation is another. Due to different Web browsers supporting different video codecs, it is a common practice to provide videos in different encodings. The user's Web browser then dynamically selects a version it can play. This is realized through the <source>

tag. Table 8 shows the observed numbers of <source> tag child nodes per <video> tag *with* <track> tag, with the result that up to four sources are given for essentially the "same" video. Table 9 confirms this observation for the entire collection of all <video> tags *with or without* <track> tag. Table 10 shows the distribution of values for the <source type> attribute of <video> tags *with* <track> tag, the clear leaders being the MP4 format followed by WebM, a trend that again is also reflected in Table 11 within the entire collection of all <video> tags *with or without* <track> tag.

3.5 Implications on Linked Data for Videos

The biggest issue with this practice of putting multiple sources is that rather than having one unique identifier (URL) per video, there can be multiple identifiers. Listing 2 shows a minimal example. Unless one repeats all statements for each source, there will always remain unclear sources without structured data. We note that a video in encoding A and the "same" video in encoding B may not be marked as <owl:sameAs>, because statements about the encoding format of one video do not apply to the other, the identity symmetry condition would thus be violated. In practice, a solution similar to specifying canonical URLs in Web search [15] seems feasible. Another approach is to require a unique identifier in the <video id> attribute, which allows for addressing the video with fragment identifiers. More advanced approaches to the problem stemming from the bibliographic universe like FRBR [29] are possible, but for the concrete use case seem quite complex.

```
<div about="kitten.jpg">
  <img src="kitten.jpg" alt="Cute␣kitten" />
  <a rel="license" href="http://creativecommons.
      org/licenses/by-sa/3.0/">
    Creative Commons Attribution Share-Alike 3.0
  </a>
</div>

<div about="kitten.mp4">
  <video>
    <source src="kitten.mp4"/>
    <source src="kitten.webm"/>
  </video>
  <a rel="license" href="http://creativecommons.
      org/licenses/by-sa/3.0/">
    Creative Commons Attribution Share-Alike 3.0
  </a>
</div>
```

Listing 2: Specifying a license for an image and attempt to do the same for a video with two sources (the license of kitten.webm stays unclear)

4. WEBVTT CONVERSION TO RDF-BASED LINKED DATA

The WebVTT specification [24] defines a syntax for conveying timed video text tracks, and a semantics for this syntax in terms of how Web browsers should process such tracks. It achieves this by specifying an underlying data model for those tracks. The aim of this section is to show how this data model can easily be mapped to RDF-based Linked Data, and thus allowing for many other usage scenarios for this data. For this purpose, we propose an RDF-Schema ontology[17] conveying the WebVTT data model. In the rest of the paper, terms from this ontology will be preceded by the vtt: prefix. An online implementation of this interpretation process that we have titled LinkedVTT is likewise available online.[18] It takes the URL of any WebVTT file, the contents of a raw WebVTT file, or a YouTube URL of any video with closed captions as an input, and applies the conversion from WebVTT to Linked Data on-the-fly.

4.1 Basic Interpretation

A WebVTT file defines a set of cues, which are described by a pair of timestamps and a payload. In other words, each cue is an annotation of the video, associating a temporal video fragment to the payload, delimited by the two timestamps. As there is a standard way of identifying *temporal* and *spatial* video fragments with a URI [30] it is straightforward to represent this annotation as an RDF triple. We therefore propose a property vtt:annotatedBy to serve as predicate for those triples. To keep the context of each annotation, we use the notion of RDF dataset [7]. Each vtt:annotatedBy triple is enclosed in a named graph, whose name is either a URI, based on the cue identifier if it has one, or a blank node if the cue has no identifier. The default graph of the dataset describes its overall structure, linking the dataset URI to all the URIs and blank nodes identifying its cues with the vtt:hasCue property. In the default graph, each cue is also linked to the Media Fragments URI it describes, with the vtt:describesFragment property. As the notion of dataset is a recent addition to the RDF core concepts (previously, it was specific to the SPARQL query language), we envision that some consumers will not be able to deal with it. Hence, we propose an alternate interpretation of WebVTT as RDF. In this *flat* interpretation, the contents of all named graphs is merged into the default graph, at the expense of contextual information.

4.2 Advanced Interpretation

WebVTT is not limited to textual timed text tracks. As Table 1 details, the HTML5 <track> tag supports different kinds of tracks, one of them being *metadata*, a track designed for machine rather than human consumption. Although it was shown in Subsection 3.3 that there is no measurable evidence of use for this kind of track yet—which is understandable given that the technology is still under development—we propose that JSON data is a good candidate for cues of such tracks. JSON has a textual syntax that is easy to author and easy to process in a Web browser and elsewhere. Furthermore, JSON-LD [27] provides a standard way to interpret JSON data as Linked Data, which fits nicely with our approach. More precisely, whenever the payload of a cue successfully parses as a JSON object, we consider that this object is meant to represent the annotated media fragment itself, and interpret it as JSON-LD. In consequence, all properties of the JSON object are applied directly to the fragment, and embedded structures can be used to describe other resources related to that fragment, *e.g.*, depicted persons, locations, topics, related videos or video fragments, or

[17] RDF-Schema ontology: http://champin.net/2014/linkedvtt/onto#

[18] LinkedVTT: http://champin.net/2014/linkedvtt/

```
WEBVTT

cue1
00:00:00.000 --> 00:00:12.000
{
  "@context": "http://champin.net/2014/linkedvtt/
      demonstrator-context.json",
  "tags": ["wind scene", "opening credits"],
  "contributors": ["http://ex.org/sintel"]
}
```

Listing 3: Sample WebVTT metadata file with JSON-LD payload in a cue identified as "cue1"

spatiotemporal video tags. In this case, all the triples generated from parsing the payload as JSON-LD *replace* the `vtt:annotatedBy` triple in the cue's named graph. Listing 3 gives an example of such JSON-LD payload. We note that it includes the JSON-LD specific *@context* key, to allow its interpretation as Linked Data. This context can be specified in each cue, but below we also provide an alternative way to declare it once for the entire WebVTT file.

4.3 Linked Data Related Metadata

In addition to the cues, WebVTT files can contain metadata headers described as key-value pairs. While the WebVTT specification defines a number of metadata headers, it leaves it open for extensions. We propose three extended metadata headers listed below. Most WebVTT currently does not contain these metadata headers, but we argue that they allow for an easy transition from plain WebVTT to Linked Data WebVTT, just like JSON-LD makes it easy to turn plain JSON into Linked Data by adding a `@context` property. Further more, other metadata headers will be evaluated against the JSON-LD context, and can produce additional triples with the WebVTT file as its subject.

@base Sets the base URI used for resolving relative URIs. This applies to any relative URIs that would be found in the JSON-LD descriptions, but also to generate URIs for cues based on their identifiers. It defaults to the URI of the WebVTT file.

@context This key can be used multiple times; each value is the URI of a JSON-LD context that should be used to interpret the JSON payloads in the WebVTT file.

@video Sets the URI for the video for generating media fragment URIs. If not present, the video URI must be provided externally, *e.g.*, the `<video src>` attribute of the video containing the WebVTT track. This metadata header is a direct response to an issue that we have outlined in Subsection 3.5.

4.4 Integrating Existing Videos Into the Web(VTT) of Data

Given the currently rather manageable amount of videos with captions or subtitles as outlined in Subsection 3.3, approaches for the automated semantic lifting based on timed text track data are feasible. These approaches extract the transcribed text snippets from cues and either convert them into one consistent block of text or treat each text snippet in isolation before applying named entity extraction on them. Representative examples based on this idea are [18, 19, 20] by Li *et al.* or also [28] by us. In combination with Media Fragments URI, spatiotemporal annotations can be created with good precision and reasonable time effort both on-the-fly or in bulk for static storage in a triple store.

5. ONLINE VIDEO ANNOTATION FORMAT AND EDITOR

Complementary to the conversion process presented in Section 4, in this section we focus on facilitating the online creation and consumption of `metadata` tracks for future video creations and videos not contained in the Common Crawl corpus. We begin with the annotation model.

5.1 Annotation Model

Our annotation model is the same as the one produced by the interpretation process presented above. Annotations take the form of RDF statements (subject-predicate-object), where the subject is any temporal or spatiotemporal fragment of the video, identified by the corresponding Media Fragments URI. They are encoded as `TextTrackCues` with JSON-LD payloads such as the one shown in Listing 3. A dedicated data context defines their semantics.

5.2 WebVTT Editor

We have implemented this annotation model in form of an online demonstrator prototype. The demonstrator interprets the existing metadata track for a video and reacts on annotations when the `currentTime` of the media resource matches the `startTime` or `endTime` of a cue. We call existing annotations *Read* annotations. Users can add *Write* annotations by creating new `TextTrackCues` at the desired start and end times and by providing their JSON-LD payloads. The editor facilitates this task through a graphical user interface, abstracting the underlying details. Figure 1 shows a screenshot of the WebVTT editor. Newly generated annotations get directly interpreted and can be persistently stored locally or in the future remotely for collaborative editing. We have developed a WebVTT to JSON-LD converter, capable of transforming WebVTT metadata tracks following our annotation model into JSON-LD for the Web of Data. This allows for straight-forward local annotation creation with Semantic Web compliance upon global publication.

5.2.1 Semantic Annotation Types

Our JSON-LD context eases common annotation tasks by defining the semantics of a few useful JSON properties described below. According to this context, Listing 3 is interpreted as in Listing 4 (RDF in JSON-LD syntax) and Listing 5 (RDF in N-Triples syntax). More advanced annotation tasks can be supported by extending the data context.

Plain Text Tags Annotations of type `tags` allow for adding plain text tags to a media fragment. They are interpreted as Common Tag [13] format `ctag:label`.

Semantic Tags Annotations of type `semanticTags` allow for adding semantic tags to a media fragment. Unlike plain text tags, semantic tags are references to well-defined concepts complete with their own URIs. They are interpreted as Common Tag [13] format `ctag:means`. Spatiotemporal semantic tags allow for interesting Linked Data experiences if the tags point to well-connected concepts.

```
{
  "@context": "http://champin.net/2014/linkedvtt/
      context.json",
  "@id": "http://ex.org/metadata.vtt",
  "@type": "VideoMetadataDataset",
  "video": "http://ex.org/video",
  "cues": [{
    "@id": "#id=cue1",
    "fragment": {
      "@context": "http://champin.net/2014/
          linkedvtt/demonstrator-context.json",
      "@id": "http://ex.org/video#t
          =0:0.0,0:12.0",
      "tags": ["wind scene", "opening credits"],
      "contributors": ["http://ex.org/sintel"]
    }
  }]
}
```

Listing 4: Generated JSON-LD file based on the WebVTT file shown in Listing 3 (flat interpretation)

Contributors The `contributors` annotation type allows for denoting the contributors in a media fragment, like its actors. They are interpreted as Ontology for Media Resources [17] format `ma:hasContributor`.

Summary The `summary` annotation type allows for summarizing a media fragment (note, not the whole video like kind *description* tracks) with plain text. They are interpreted as `ma:description` [17].

5.2.2 Presentation-Oriented Annotation Types

Presentation-oriented annotations—similar to temporal style sheets—do not generate RDF data, but only impact the way videos get presented.

Visual Effect Annotations of type `visualEffect` allow for applying visual effects in the syntax of Cascading Style Sheets[19] (CSS) to a media fragment, *e.g.*, filters, zoom, transparency, and 2D/3D transformations and animations.

Audial Effect The `audialEffect` annotation type allows for applying audial effects to a media fragment. Currently, we support modifying the volume from 0 to 1.

Playback Rate The `playbackRate` annotation type allows for specifying the effective playback rate of a media fragment. The playback rate is expressed as a floating point multiple or fraction of the intrinsic video speed.

HTML Overlay Via the `htmlOverlay` annotation type, overlays in freeform HTML code can be added to a media fragment. Examples are graphical, textual, or combined overlays that can contain links to (temporal fragments of) other videos or within the current video.

[19]Cascading Style Sheets: `http://www.w3.org/Style/CSS/`

```
<http://ex.org/metadata.vtt> <http://www.w3.org
    /1999/02/22-rdf-syntax-ns#type> <http://ex.
    org/VideoMetadataDataset> .
<http://ex.org/metadata.vtt> <http://champin.net
    /2014/linkedvtt/onto#hasCue> <http://ex.org/
    metadata.vtt#id=cue1> .
<http://ex.org/metadata.vtt#id=cue1> <http://
    champin.net/2014/linkedvtt/onto#
    describesFragment> <http://ex.org/video#t
    =0:0.0,0:12.0> .
<http://ex.org/video#t=0:0.0,0:12.0> <http://
    commontag.org/ns#label> "wind scene" .
<http://ex.org/video#t=0:0.0,0:12.0> <http://
    commontag.org/ns#label> "opening credits" .
<http://ex.org/video#t=0:0.0,0:12.0> <http://www.
    w3.org/ns/ma-ont#hasContributor> <http://ex.
    org/sintel> .
```

Listing 5: RDF triples based on the JSON-LD code from Listing 4

5.3 Interpretation Layer

In our WebVTT editor, we propose an interpretation layer capable of dealing with the herein defined annotation types. We thus make an open world assumption by supporting a set of pre-defined values for predicate and object listed below, and ignoring unknown ones. This permits others to extend—or even completely replace—our interpretation layer. If a `TextTrackCue` has a WebVTT identifier, we use it to address its annotations via the metadata track's URI and corresponding cue fragment identifier, allowing for meta annotations of annotations, *e.g.*, to attach provenance or license information to them.

5.4 Evaluation

We evaluate or annotation model and related technology stack based on a state-of-the-art hypervideo model by Sadallah *et al.* [26] that builds on a careful study of prior art.

The CHM Hypervideo Model.

Sadallah *et al.* define *hypervideo* as *"interactive video-centric hypermedia document built upon audiovisual content"*. The authors identify three common hypervideo characteristics, namely *(i) interactivity*, which, *e.g.*, can enable richer navigational possibilities, *(ii) non-linearity*, which allows for features like video montages, and finally *(iii) enrichments* that include all sorts of supplementary material besides and on top of hypervideos. The authors have examined hypervideo systems of recent years and found recurring patterns, summarized and compared to our approach in the following.

Video player and controls Hypervideo systems by definition provide one or multiple video players, however, the corresponding video controls are not necessarily exposed.

✓ Our approach uses the (optionally customizable) default HTML5 player that includes hidable controls (Figure 1).

Timeline A timeline is the spatial representation of temporally situated metadata in a video. The most common timeline pattern shows the time along the x-axis and corresponding metadata along the y-axis.

✓ Our approach supports temporal metadata. Customizable timeline visualizations exist[20] and can be added.

Textual or graphical overlay Additional textual or graphical information can be displayed in form of overlays on the video. Overlays can also serve as external or video-internal hyperlinks, referred to as *hotspots*.

✓ We realize overlays and links with `htmlOverlay` types. Figure 1 shows both a graphical (yellow box) and two textual overlays (red and green texts).

Textual or graphical table of contents If a video is logically separated into different parts, a table of contents lists these in textual or graphical form, makes them navigable, or visually summarizes them, referred to as *video map*.

✓ Textual tables of contents are directly supported via WebVTT text tracks of type *chapters*. Graphical tables of contents can be created based thereon.

Transcript The textual document of the transcribed audiovisual content of a video allows for following along the video by reading and also serves for in-video navigation.

✓ Subtitles and captions are natively supported by WebVTT tracks of the types *subtitles* and *captions*. Figure 1 shows active subtitles (white text).

6. RELATED WORK

With our annotation approach, we leverage WebVTT metadata tracks as a means for tying semantic JSON-LD annotations to temporal or spatiotemporal video fragments. As each `<track>` tag by pure definition is bound to exactly one `<video>` tag, and as modern search engines parse and interpret JSON-LD annotations, a unique relation of annotations to video content is made. In consequence, related work can be regarded under the angles of online annotation creation and large-scale Linked Data efforts for video. Many have combined Linked Data and video, typical examples are [16] by Lambert *et al.* and [12] by Hausenblas *et al.* We have already described the text track enriching approaches [18, 19, 20, 28] in Subsection 4.4, [20] being closest to our idea of a *Web(VTT) of Data*, albeit their approach is centered around their application Synote. The online video hosting platform YouTube lets video publishers add video annotations in a closed proprietary format. From 2009 to 2010, YouTube had a feature called Collaborative Annotations [1] that allowed video consumers to collaboratively create video annotations. Unlike the format of YouTube, our format is open and standards-based. In [31], Van Deursen *et al.* present a system that combines Media Fragments URI and the Ontology for Media Resources in an HTML5 Web application to convert rich media fragment annotations into a WebVTT file that can be used by HTML5-enabled players to show the annotations in a synchronized way. Building on their work, we additionally allow for writing annotations by letting annotators create WebVTT cues with an editor. The Component-based Hypervideo Model Popcorn.js[21] is an HTML5 JavaScript media framework for the creation of media mixes by adding interactivity and context to online video by letting users link social media, feeds, visualizations, and other content directly to moving images. PopcornMaker[22] is an interactive Web authoring environment that allows for videos to be annotated on a video timeline. While Popcorn media annotations are essentially JavaScript programs, our approach is based on directly indexable WebVTT files.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced our vision of the *Web(VTT) of Data*, a global network of videos and connected content that is based on relationships among videos based on WebVTT files, which we use as Web-native spatiotemporal containers of Linked Data with JSON-LD payloads. With the recent graduation of the JSON-LD syntax as an official W3C Recommendation and a major search engine company[23] supporting embedded JSON-LD documents in HTML documents,[24] JSON-LD definitely is here to stay. Likewise for WebVTT, which in the more recent past has been natively implemented by all major Web browser vendors, the future is bright. We combine both technologies in a fruitful way that is focused both at common Web search engines as well as at the entire Linked Data stack of technologies. Using WebVTT as a container for JSON-LD is both innovative and natural. Making commonly understood semantic statements about video fragments on the Web has become feasible thanks to Media Fragments URI, a standard that allows for applying Linked Data approaches to moving images on a temporal and spatiotemporal axis. We have organized this paper in three major steps. *(i)* in order to get a better understanding of the *status quo* of Web video deployment, we have performed a large-scale analysis of the 148 terabyte size Common Crawl corpus, *(ii)* we have addressed the challenge of integrating existing videos in the Common Crawl corpus into the *Web(VTT) of Data* by proposing a WebVTT conversion to RDF-based Linked Data, and *(iii)* we have open-sourced an online video annotation creation and consumption tool, targeted at videos not contained in the Common Crawl corpus and for integrating future video creations. In this paper, we have combined Big Data and Small Data. On the Big Data side, we have learned from the Common Crawl corpus which kind of timed text tracks are out there, which allowed us to propose a realistic approach to integrating it into the *Web(VTT) of Data*. On the Small Data side, we have implemented an online editor for the creation of semantic video annotations that can be applied video by video, so that the *Web(VTT) of Data* gets woven tighter and tighter with each new addition.

Future work has several dimensions. Beginning from video annotation, a first concrete research task is to work on our editor prototype. While a lot of efforts can be put in the editor itself, far more added value is created by proposing an extension to the most well-known online video annotation stack, the Popcorn.js and PopcornMaker projects. A minimal Popcorn.js example annotation can be seen in Listing 6. Rather than storing the annotations as steps of a JavaScript program that "artificially" need to be aligned to the corresponding parts of the video, an extension to

[20] D3 timeline implementation: `https://github.com/jiahuang/d3-timeline`

[21] Popcorn.js: `http://popcornjs.org/`

[22] PopcornMaker: `https://popcorn.webmaker.org/`

[23] JSON-LD in Gmail: `https://developers.google.com/gmail/actions/reference/formats/json-ld`

[24] Embedding JSON-LD in HTML Documents: `http://www.w3.org/TR/json-ld/#embedding-json-ld-in-html-documents`

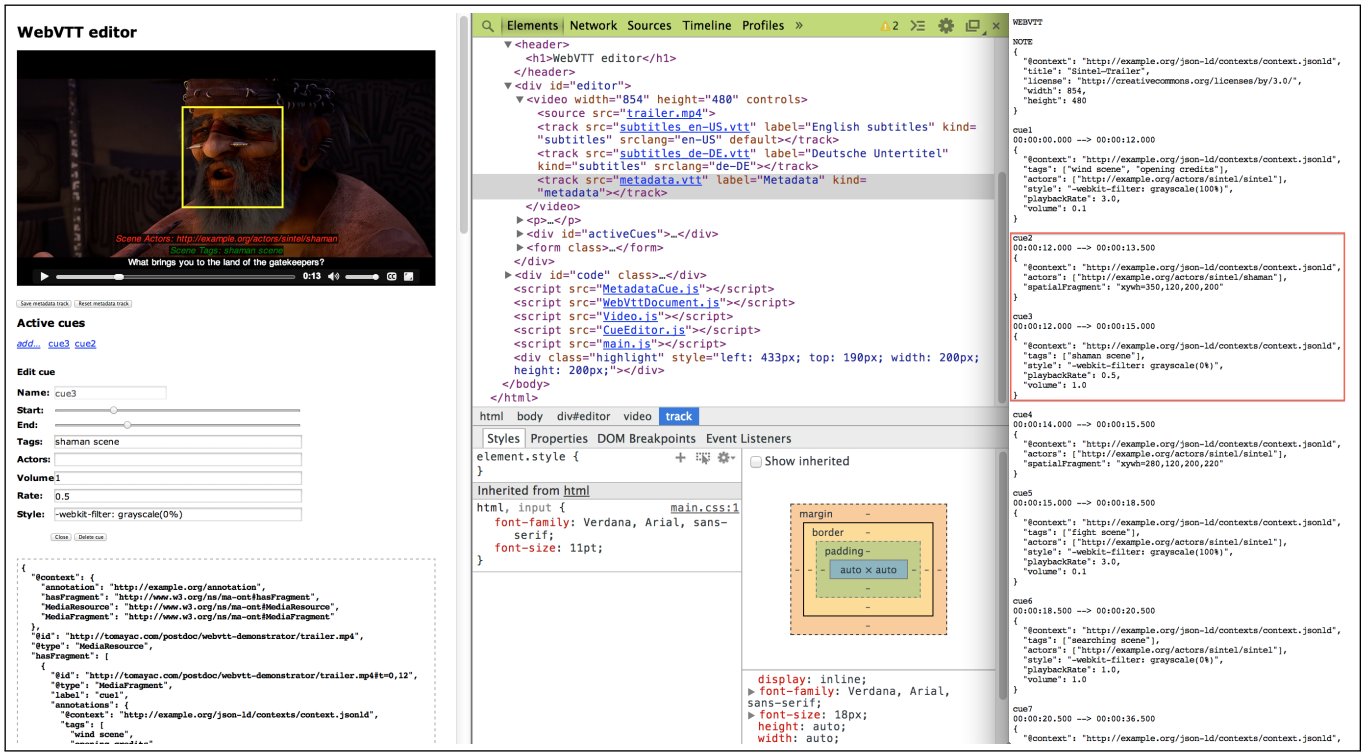

Figure 1: WebVTT editor interpreting the spatiotemporal annotation "cue2" that identifies the highlighted spatial fragment as `ex:actors/sintel/shaman`; while in parallel modifying "cue3" with tag, volume, playback rate, and style (① left: Graphical User Interface with JSON-LD debug view, ② center: Chrome Developer Tools with highlighted `<track src="metadata.vtt" kind="metadata">` tag, ③ right: raw WebVTT file `metadata.vtt` with highlighted "cue2" and "cue3")

Popcorn.js could use our approach of leveraging naturally temporally aligned WebVTT cues with JSON-LD payloads for the annotations. We have been able to play video in Web browsers plugin-free for a couple of years now, the next step is adding resources to videos to make them more accessible and provide more options to the viewer. Straightforward things to do are to profit from recent advances in machine-translation and speech recognition to evaluate the usefulness of automatically transcribed and translated captions combined with language-independent `metadata` annotations based on named entity extraction for providing Linked Data paths between videos no matter their original language. We have learned that with regard to HTML5 video with timed text track support, it is still early days, so at least in the short-term it will be inevitable to deal with legacy plugin-dependent content and ways to integrate it in the *Web(VTT) of Data* through adaptors or converters *etc.*

Concluding, the vision of the *Web(VTT) of Data* is a realistic one and all building blocks are in place. We are optimistic that by leveraging the popularity of existing tools like Popcorn.js, we can push the state of Web video forward toward an interconnected, semantic, and wall-free experience.

Acknowledgments

The research presented in this paper was partially supported by the French National Agency for Research project *Spectacle En Ligne(s)*, project reference ANR-12-CORP-0015.

```
<video id="video" src="http://ex.org/video.mp4">
</video>
<div id="footnote-container"></div>
<div id="wikipedia-container"></div>
<script>
  // get a reference to the video
  var pop = Popcorn("#video");
  // add a footnote from second 2 to second 6
  pop.footnote({
      start: 2,
      end: 6,
      text: "In this scene: George Clooney",
      target: "footnote-container"
    });
  // add a reference to a Wikipedia article
  // from second 2 to 6
  pop.wikipedia({
      start: 2,
      end: 6,
      src: "http://en.wikipedia.org/wiki/George
          _Clooney",
      title: "George Clooney",
      target: "wikipedia-container"
    });
</script>
```

Listing 6: Popcorn.js example

8. REFERENCES

[1] S. Bar et al. YouTube's Collaborative Annotations. In *Webcentives '09, 1st International Workshop on Motivation and Incentives*, pages 18–19, 2009.

[2] R. Berjon, S. Faulkner, T. Leithead, et al. HTML5, A Vocabulary and Associated APIs for HTML and XHTML. Candidate Recommendation, W3C, 2013. http://www.w3.org/TR/html5/.

[3] C. Bizer, K. Eckert, R. Meusel, H. Mühleisen, M. Schuhmacher, and J. Völker. Deployment of RDFa, Microdata, and Microformats on the Web – A Quantitative Analysis. In H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. Parreira, L. Aroyo, N. Noy, C. Welty, and K. Janowicz, editors, *The Semantic Web – ISWC 2013*, volume 8219 of *Lecture Notes in Computer Science*, pages 17–32. Springer Berlin Heidelberg, 2013.

[4] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data—The Story So Far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[5] C. Bizer, T. Heath, T. Berners-Lee, and M. Hausenblas, editors. *WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012*, volume 937 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[6] M. Burner and B. Kahle. Arc File Format. Technical report, Jan. 1996. http://archive.org/web/researcher/ArcFileFormat.php.

[7] R. Cyganiak, D. Wood, and M. Lanthaler. RDF 1.1 Concepts and Abstract Syntax. Proposed Recommendation, W3C, Jan. 2014. http://www.w3.org/TR/rdf11-concepts/.

[8] D. Dorwin, A. Bateman, and M. Watson. Encrypted Media Extensions. Working Draft, W3C, Oct. 2013. http://www.w3.org/TR/encrypted-media/.

[9] S. Dutton. Getting Started With the Track Element, Feb. 2012. http://www.html5rocks.com/en/tutorials/track/basics/.

[10] V. Goel. Web Archive Metadata File Specification. Technical report, Apr. 2011. https://webarchive.jira.com/wiki/display/Iresearch/Web+Archive+Metadata+File+Specification.

[11] L. Green. Winter 2013 Crawl Data Now Available, Jan. 2014. http://commoncrawl.org/winter-2013-crawl-data-now-available/.

[12] M. Hausenblas, R. Troncy, Y. Raimond, and T. Bürger. Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments. In *Linked Data on the Web Workshop (LDOW 09), in conjunction with the 18th International World Wide Web Conference (WWW 09)*, 2009.

[13] A. Iskold, A. Passant, V. Miličić, et al. Common Tag Specification, June 2009. http://commontag.org/Specification.

[14] ISO 28500. Information and documentation – The WARC File Format. International Standard, 2008. http://bibnum.bnf.fr/WARC/WARC_ISO_28500_version1_latestdraft.pdf.

[15] J. Kupke and M. Ohye. Specify your canonical, Feb. 2009. http://googlewebmastercentral.blogspot.de/2009/02/specify-your-canonical.html.

[16] D. Lambert and H. Q. Yu. Linked Data based Video Annotation and Browsing for Distance Learning. In *SemHE '10: The Second International Workshop on Semantic Web Applications in Higher Education*, 2010.

[17] W. Lee, W. Bailer, T. Bürger, et al. Ontology for Media Resources 1.0. Recommendation, W3C, Feb. 2012. http://www.w3.org/TR/mediaont-10/.

[18] Y. Li, G. Rizzo, J. L. Redondo García, R. Troncy, M. Wald, and G. Wills. Enriching Media Fragments with Named Entities for Video Classification. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 469–476, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[19] Y. Li, G. Rizzo, R. Troncy, M. Wald, and G. Wills. Creating Enriched YouTube Media Fragments with NERD Using Timed-Text. In *11th International Semantic Web Conference (ISWC2012)*, November 2012.

[20] Y. Li, M. Wald, T. Omitola, N. Shadbolt, and G. Wills. Synote: Weaving Media Fragments and Linked Data. In Bizer et al. [5].

[21] P. Linsley. Specifying an image's license using RDFa, Aug. 2009. http://googlewebmastercentral.blogspot.com/2009/08/specifying-images-license-using-rdfa.html.

[22] H. Mühleisen and C. Bizer. Web Data Commons – Extracting Structured Data from Two Large Web Corpora. In Bizer et al. [5].

[23] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford InfoLab, Nov. 1999.

[24] S. Pfeiffer and I. Hickson. WebVTT: The Web Video Text Tracks Format. Draft Community Group Specification, W3C, Nov. 2013. http://dev.w3.org/html5/webvtt/.

[25] D. Raggett, A. Le Hors, and I. Jacobs. HTML 4.01 Specification. Recommendation, W3C, Dec. 1999. http://www.w3.org/TR/html401.

[26] M. Sadallah et al. CHM: An Annotation- and Component-based Hypervideo Model for the Web. *Multimedia Tools and Applications*, pages 1–35, 2012.

[27] M. Sporny, D. Longley, G. Kellogg, et al. JSON-LD 1.0, A JSON-based Serialization for Linked Data. Proposed Recommendation, W3C, Nov. 2013. http://www.w3.org/TR/json-ld/.

[28] T. Steiner. SemWebVid – Making Video a First Class Semantic Web Citizen and a First Class Web Bourgeois. In A. Polleres and H. Chen, editors, *Proceedings of the ISWC 2010 Posters & Demonstrations Track: Collected Abstracts, Shanghai, China, November 9, 2010*, volume 658 of *CEUR Workshop Proceedings ISSN 1613-0073*, pages 97–100, Nov. 2010.

[29] B. Tillett. FRBR: A Conceptual Model for the Bibliographic Universe. Technical report, 2004. http://www.loc.gov/cds/downloads/FRBR.PDF.

[30] R. Troncy, E. Mannens, S. Pfeiffer, et al. Media Fragments URI 1.0 (basic). Recommendation, W3C, Sept. 2012. http://www.w3.org/TR/media-frags/.

[31] D. Van Deursen, W. Van Lancker, E. Mannens, et al. Experiencing Standardized Media Fragment Annotations Within HTML5. *Multimedia Tools and Applications*, pages 1–20, 2012.