This deliverable captures actions planned and done by the consortium to standardise, help finalize emerging standards, and contribute features that the project needed to relevant standardisation groups.
### Basic Information

<table>
<thead>
<tr>
<th>Work package</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Due date</td>
<td>30/04/2020</td>
</tr>
<tr>
<td>Submission date</td>
<td>20/04/2020</td>
</tr>
<tr>
<td>Deliverable lead</td>
<td>W3C</td>
</tr>
<tr>
<td>Version</td>
<td>2.0</td>
</tr>
<tr>
<td>Authors</td>
<td>François Daoust (W3C)</td>
</tr>
<tr>
<td>Reviewers</td>
<td>Alexander Erk (IRT), Leo Andrews (Radioplayer)</td>
</tr>
</tbody>
</table>

### Document Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description of change</th>
<th>List of contributor(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0.1</td>
<td>21/09/2018</td>
<td>Outline</td>
<td>François Daoust (W3C)</td>
</tr>
<tr>
<td>V0.2</td>
<td>16/11/2018</td>
<td>Initial version</td>
<td>François Daoust (W3C)</td>
</tr>
<tr>
<td>V0.3</td>
<td>22/11/2018</td>
<td>Sections on station logos/names filled out, Misc. editorial updates</td>
<td>Leo Andrews (Radioplayer)</td>
</tr>
<tr>
<td>V0.4</td>
<td>27/11/2018</td>
<td>RadioDNS/WorldDAB updates</td>
<td>Alexander Erk (IRT), François Daoust (W3C)</td>
</tr>
<tr>
<td>V1.0</td>
<td>30/11/2018</td>
<td>Final 1st version after coordinator review</td>
<td>Simon Delaere (imec)</td>
</tr>
<tr>
<td>V1.1</td>
<td>11/03/2020</td>
<td>Update status of standardisation activities at the end of the project</td>
<td>François Daoust (W3C)</td>
</tr>
<tr>
<td>V1.2</td>
<td>20/03/2020</td>
<td>Add Other Standardisation Updates, Link actions and contributions</td>
<td>François Daoust (W3C), Dominique Hazaël-Massieux (W3C)</td>
</tr>
<tr>
<td>V1.3</td>
<td>20/04/2020</td>
<td>Complete report with IRT updates, Review</td>
<td>Alexander Erk (IRT), François Daoust (W3C)</td>
</tr>
<tr>
<td>V2.0</td>
<td>20/04/2020</td>
<td>Final 2nd version after coordinator review</td>
<td>Simon Delaere (imec)</td>
</tr>
</tbody>
</table>
Disclaimer

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 731677. This document reflects only the authors’ views and the Commission is not responsible for any use that may be made of the information it contains.

<table>
<thead>
<tr>
<th>Project co-funded by the European Commission in the H2020 Programme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nature of the deliverable:</td>
</tr>
<tr>
<td>Dissemination Level</td>
</tr>
<tr>
<td>PU</td>
</tr>
<tr>
<td>CL</td>
</tr>
<tr>
<td>CO</td>
</tr>
</tbody>
</table>
EXECUTIVE SUMMARY

This is the final version of the standardisation plan and report for the HRADIO project. It captures the standardisation areas that were deemed relevant for the project when it started, along with actions planned and done by the consortium to standardise, help finalize emerging standards, and contribute features that the project needs to relevant standardisation groups. Contributions to standards include the specification of the Open Mobile Radio Interface (OMRI) [1] at WorldDAB and its publication by ETSI, and renewed media standardisation activities in W3C to improve handling of adaptive streaming scenarios within Web applications. This includes the creation of the W3C Media Working Group to develop standards such as Media Capabilities [12], Media Session [13], and a new version of Media Source Extensions that enables codec switching scenarios.
# TABLE OF CONTENTS

1. **INTRODUCTION** ........................................................................................................... 9
   1.1. Background .................................................................................................................... 9
   1.2. Objectives ..................................................................................................................... 9
   1.3. Structure of this document .......................................................................................... 9

2. **STANDARDISATION PLAN** ....................................................................................... 11
   2.1. Relevant standardisation organisations ....................................................................... 11
       2.1.1. RadioDNS / ETSI ................................................................................................. 11
       2.1.2. WorldDAB / ETSI ............................................................................................... 12
       2.1.3. W3C ..................................................................................................................... 12
   2.2. Candidate standardisation areas ............................................................................... 13
       2.2.1. Radio tuner support in native apps ....................................................................... 14
       2.2.2. Radio tuner support on the Web .......................................................................... 15
       2.2.3. Live content support on the Web ........................................................................ 16
       2.2.4. DAB decoding in a Web app .............................................................................. 17
       2.2.5. Distribution and synchronization of Web content with media .............................. 18
       2.2.6. Fragmentation reduction ...................................................................................... 18
       2.2.7. Improve image handling for station logos .............................................................. 19
       2.2.8. Phonetic spelling of station names ........................................................................ 19
       2.2.9. Convergence of metadata vocabularies ............................................................... 20
   2.3. Action plan .................................................................................................................. 20

3. **CONTRIBUTIONS TO STANDARDISATION ACTIVITIES** ...................................... 23
   3.1. Open Mobile Radio Interface (OMRI) ................................................................. 23
   3.2. TV Control API ......................................................................................................... 23
   3.3. Use of EDI for DAB streaming .................................................................................. 25
   3.4. Media Source Extensions ......................................................................................... 26
   3.5. WebRTC / WebTransport / WebCodecs ................................................................. 27
   3.6. Threading support on the Web .................................................................................. 29
   3.7. Media Timed Events ................................................................................................. 30
   3.8. Web Media API ........................................................................................................ 32
3.9. Media Capabilities ................................................................. 33
3.10. Media Session ................................................................. 33
3.11. Link between slideshows and Dynamic Labels .................... 33

4. OTHER STANDARDISATION UPDATES .................................. 35
4.1. Autoplay policy detection .................................................. 35
4.2. Network quality monitoring and prediction .......................... 36
4.3. MediaStreamTrack Content Hints ...................................... 36
4.4. Media Feeds API ............................................................. 37
4.5. Efficient low-level audio hardware device access ................. 37
4.6. Improving spoken presentation on the Web ......................... 38

5. CONCLUSIONS ....................................................................... 39
LIST OF TABLES

TABLE 1 - STANDARDISATION ACTION PLAN

22
# ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ATSC</td>
<td>Advanced Television Systems Committee</td>
</tr>
<tr>
<td>CDN</td>
<td>Content Distribution Network</td>
</tr>
<tr>
<td>CE</td>
<td>Consumer Electronics</td>
</tr>
<tr>
<td>CTA</td>
<td>Consumer Technology Association</td>
</tr>
<tr>
<td>DAB</td>
<td>Digital Audio Broadcasting</td>
</tr>
<tr>
<td>DASH</td>
<td>Dynamic Adaptive Streaming over HTTP</td>
</tr>
<tr>
<td>EPG</td>
<td>Electronic Program Guide</td>
</tr>
<tr>
<td>ESO</td>
<td>European Standardisation Organization</td>
</tr>
<tr>
<td>FQDN</td>
<td>Fully Qualified Domain Name</td>
</tr>
<tr>
<td>HbbTV</td>
<td>Hybrid broadcast broadband TV</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>HLS</td>
<td>HTTP Live Streaming</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>MSE</td>
<td>Media Source Extensions</td>
</tr>
<tr>
<td>OMRI</td>
<td>Open Mobile Radio Interface</td>
</tr>
<tr>
<td>QUIC</td>
<td>Quick UDP Internet Connections</td>
</tr>
<tr>
<td>SPI</td>
<td>Service and Programme Information</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WAVE</td>
<td>Web Application Video Ecosystem</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

This deliverable reports on the standardisation activities that HRADIO project partners considered and contributed to during the lifetime of the project.

1.1. BACKGROUND

The HRADIO project focuses on radio service innovations enabled by the convergence between cost-effective broadcast distribution and the flexibility and interactivity of internet-based mechanisms, notably to personalise radio services, provide intuitive functionalities like time-shifting and, eventually, to foster and exploit user engagement.

While such features are frequent on TV, through interactive TV standards such as HbbTV in Europe, radios have just started to embrace hybrid technologies. HRADIO paves the way to bring these features not only to broadcasters’ native mobile applications, but also to Web sites, connected radios and into the car, publishing its developments as ready-to-use Android and HTML client implementations of well documented APIs fostering new service developments for the radio sector.

1.2. OBJECTIVES

The main objectives of HRADIO’s standardisation activities were:

- To specify the APIs at the heart of the HRADIO platform and have them adopted as standards and implemented on devices.
- To make sure that the web is a suitable platform for HRADIO, so that HRADIO applications may target a wide range of devices.

Standardisation of a given technology can only be achieved when relevant stakeholders, both within the project and outside of the project, agree to it. The project supported partner efforts needed to drive discussions in standardisation organisations and/or finalize on-going standardisation activities.

1.3. STRUCTURE OF THIS DOCUMENT

Section 2 outlines the standardisation plan created during the first year of the HRADIO project and highlights standardisation topics that have been identified as directly relevant to the project. It also describes the standardisation action plan...
discussed by project partners. This section is essentially the same as the one in the previous version of this deliverable published after the first year of the project.

Section 3 details the contributions towards standardisation activities that project partners made during the project in relation with the HRADIO project. This section has been adjusted since publication of the previous version to account for new partner activities.

Section 4 details evolutions of relevant standards that HRADIO partners tracked during the lifetime of the project but did not directly contribute to, and their potential longer term impact on HRADIO applications. This section did not exist in the previous version.
2. STANDARDISATION PLAN

2.1. RELEVANT STANDARDISATION ORGANISATIONS

2.1.1. RadioDNS / ETSI

RadioDNS is an organisation which develops hybrid radio specifications. RadioDNS provides the link between content that is broadcast over FM, DAB, HD Radio and content provided over an Internet connection. Specifications published by RadioDNS are standardized at ETSI, one of official European Standards Organizations (ESO).

RadioDNS published three standards that are directly relevant to the HRADIO project:

- **TS 103 270 v1.2.1 – RadioDNS Hybrid Radio; Hybrid lookup for radio services** [2]: The standard defines a methodology for discovering an Authoritative FQDN for a radio service, including discovery using DNS queries to radiodns.org, a root domain server operated by RadioDNS.

- **TS 101 499 v3.1.1 – Slideshow** [3]: The standard describes an application that can be delivered using broadcast, IP, or a combination of the two, and that provides a visual and clickable accompaniment to a radio service. Note this standard is now maintained in WorldDAB.

- **TS 102 818 v3.1.1 – Service and Programme Information Error! Reference source not found.**: The standard describes the metadata about a radio station (including logos) and its individual programmes and schedule, as well as an index of on-demand/podcast material, using an XML schema data model. Note this standard is now maintained in WorldDAB.

RadioDNS is working on several other specifications, including:

---

1 https://radiodns.org/
2 https://www.etsi.org/
2.1.2. WorldDAB / ETSI

WorldDAB\(^3\) is the global industry forum for digital radio, facilitating the adoption and implementation of broadcast digital radio based on Digital Audio Broadcasting (DAB/DAB+). WorldDAB develops and promotes DAB specifications, standardized at ETSI, and develops the Open Mobile Radio Interface (OMRI) [1], a technical interface for communication between applications and radio chipsets present on the device, which is at the core of HRADIO’s system architecture.

DAB standards also define features that allow broadcasters to send metadata along the audio signal that the HRADIO project builds on to create new functionalities. These features include:

- **Dynamic Labels (DL)**, short text messages transmitted in the DAB/DAB+ signal and that can be tagged
- **RadioTAG**, a bookmarking protocol for discovery and communication between client radio devices and a broadcaster-provided web service to record the time and radio service being listened to.
- **Slideshows** [3] and **Service and Programme Information** [4] in collaboration with RadioDNS, now fully handled in WorldDAB.
- **Mapping between Transport Protocol Experts Group (TPEG) and DAB transport** to deliver Traffic and Travel Information (TTI) to end-users.

2.1.3. W3C

The World Wide Web Consortium\(^4\) (W3C) develops royalty free Web standards such as HTML, CSS, and various JavaScript APIs that compose the Web platform, implemented in Web browsers on a wide range of devices. Standards published by

---

\(^3\) https://www.worlddab.org/

\(^4\) https://www.w3.org/
W3C are also at the core of interactive layers in most hybrid radio (and TV) specifications.

W3C is organized in three main types of groups:

- **Working groups** develop technical standards
- **Community groups** conduct pre-standardisation work, incubating technical solutions that may lead to standardisation work
- **Interest groups** identify use cases & requirements to steer standardisation priorities.

Main groups identified by the HRADIO project at the beginning of the project:

- The **Media & Entertainment Interest Group**\(^5\), formerly known as the Web and TV Interest Group, steers the standardisation of media-related features on the Web. This group discusses the future of the TV Control API specification (following closure of the TV Control Working Group which drafted the specification), media timed events, new features for Media Source Extensions to improve streaming support on the Web (following closure of the HTML Media Extensions Working Group which developed the standard).

- The **schema.org Community Group**\(^6\) discusses changes, additions and extensions to schema.org, a set of schemas to annotate Web pages with structured data that search engines may ingest.

### 2.2. CANDIDATE STANDARDISATION AREAS

Hybrid radio builds on the convergence between broadcast and IP technologies. The term **hybrid** covers different dimensions:

\(^5\) [https://www.w3.org/2011/webtv/](https://www.w3.org/2011/webtv/)

\(^6\) [https://www.w3.org/community/schemaorg/](https://www.w3.org/community/schemaorg/)
The HRADIO project has explored these dimensions to identify candidate areas where lack of standards impedes or slows down the realisation of hybrid radio scenarios. These candidate areas are described below along with their status when the HRADIO project starts, and main questions considered by the project.

### 2.2.1. Radio tuner support in native apps

To provide the presentation and interactive layers envisioned in most hybrid scenarios, radio devices need to adopt operating systems originally developed for other embedded devices such as Android. Such systems do not have native or standardized interfaces to expose broadcast signals to applications running on them.

Similarly, to be able to target a wide range of devices, hybrid radio applications need to run on smartphones and other devices, regardless of the underlying operating system (Android, iOS, Windows). Such smartphones may have dedicated radio tuner hardware, or may be coupled with radio tuners, e.g. through USB, but operating systems do not provide radio tuner interfaces.

The existence of a standard interface that can be implemented at the operating system level across devices to access and control the broadcast signal is a key enabler to the creation of native hybrid radio experiences that may leverage signals received over broadcast.

This standard interface would need to be implementable on top of content received...
over IP too, so that hybrid radio applications may reuse the same code regardless of whether the signal comes from broadcast or IP.

When the HRADIO project started in 2017, preliminary discussions for the creation of a common API took place in WorldDAB, leading to the creation of a Technical Committee in February 2017 to develop the specification. Can this API be finalized, standardized and implemented across devices?

### 2.2.2. Radio tuner support on the Web

The Web platform provides a natural fit for the creation of interactive experiences that can run on a wide range of devices. The Web platform provides critical technologies to enable media applications: HTML5 [5] introduced the `<audio>` and `<video>` tags, Media Source Extensions (MSE) [6] gave full control to application over adaptive streaming, and Encrypted Media Extensions (EME) [7] provided APIs to control playback of encrypted content. Taken together, these technologies marked the end of the era of plug-ins for media playback on the Web and led to the explosion of media streaming on the Web in the past few years.

However, while Web applications can easily fetch network resources over HTTP, there is no standard way to access broadcast signals from within Web applications. W3C started work on a TV Control API [8] in 2014, under the scrutiny of AT&T, BBC, Espial, Fraunhofer FOKUS, IRT, LG, Mozilla and Sony. Standardisation efforts stopped in 2017, before the HRADIO project started, due to a combination of factors:
ATSC\textsuperscript{7} was considering referencing the TV Control API in ATSC 3.0\textsuperscript{8} provided the API could be standardized in time for inclusion. They eventually decided to create a custom API to expose the tuner to Web applications as a local network service.

AT&T's acquisition of DirecTV changed their priorities in that space as well.

Mozilla shelved the development of Firefox OS for TV, which was the main driver for them to participate in this work.

Other browser vendors did not view tuner control as a priority and had security concerns, e.g. on ways to associate a broadcast signal with an origin, more so if that signal embeds additional data, including applications.

When the HRADIO project started, there was no ongoing standardisation activity to expose radio tuners to Web applications, and the future of the TV Control API had moved into the hands of the Media & Entertainment Interest Group at W3C.

2.2.3. Live content support on the Web

Media-related Web technologies mentioned in 2.2.2 Radio tuner support on the Web were driven by on-demand use cases. They allowed the creation of smooth media playback experiences for file-based media on the Web. However, live streaming, common in radio, introduces specific requirements that go beyond what these technologies were trying to provide.

Latency of distribution

The main one is the latency of distribution. Hybrid distribution requires the ability to use different distribution mechanisms at once, broadcast and broadband. One user may be listening to radio through the broadcast signal, while another may be listening to radio through broadband. Furthermore, where practical, it should be possible to seamlessly switch from one distribution mechanism to the other. This requires the latency of distribution to be equivalent regardless of the distribution

\textsuperscript{7} https://www.atsc.org/
\textsuperscript{8} https://www.atsc.org/standards/atsc-3-0-standards/
mechanism.

Typical latency of distribution for broadcast from capture to delivery is a few seconds. When the project starts, the typical latency of distribution for broadband using MSE and adaptive streaming technologies such as MPEG DASH [9] is several dozens of seconds. This latency is due to a combination of factors, and includes the creation of media segments before distribution, caching into CDNs, internal buffering in Web browsers and playback latency. Can the latency of distribution of media content on the Web get reduced to a few seconds, to ease cases where multiple distribution mechanisms are used at once?

**Companion data sources**

Other requirements arise from companion data sources. Broadcasting takes an in-band approach, whereby the media streams are combined with additional data sources (events, metadata, images) in the same transport stream. The Web approach is more out-of-band, whereby applications are responsible to fetch additional data as separate resources to complete the media resources being played. Web browsers may support in-band captioning tracks, but practical support is limited and, as of end of 2017, HTML does not have a generic cue mechanism to expose in-band metadata tracks to Web applications. Furthermore, even if it did, the use of timed events to trigger cues may remain problematic if, in order to render a cue, an application first needs to fetch external resources (such as images). HTML does not give a way for applications to learn about new cues before they are triggered.

**Customized continuous media stream**

To produce a continuous media stream or to customize the media stream for the user, broadcasters also need to mix content from different sources. These sources may use different codecs. For instance, ads may be encoded differently from music or content captured in the studio. MSE does not let Web applications splice heterogeneous media streams, forcing broadcasters to either transcode all content to the same codec, or to work around this limitation by switching between multiple media elements, with the risk of creating “air gap”. Could this situation be improved?

**2.2.4. DAB decoding in a Web app**

One approach that the HRADIO project is prototyping is the possibility to deliver the DAB signal over HTTP and let Web applications decode, process and render the content, so that broadcasters do not have to use different tools and processes to distribute content over broadcast and broadband.
Can DAB decoding be done efficiently in Web applications? Efficient binary stream processing is usually done using programming languages that provide facilities for low-level memory manipulation, such as C++. WebAssembly, standardised at W3C, creates a binary instruction format supported across browsers, designed as a portable target for compilation of high-level languages like C++. Is it enough to cover all needs on that front?

2.2.5. Distribution and synchronization of Web content with media

The combination of media content with other types of content to create richer experiences poses several challenges. The lack of a generic data cue mechanism in HTML, and possible timing issues when cues reference external content but are seen too late by the application have been pointed out in 2.2.3 Live content support on the Web. Similarly, security issues that arise when a broadcast signal include interactive content have been raised in section 2.2.2 Radio tuner support on the Web: security on the Web platform relies on the same origin policy, how can an application delivered over broadcast be securely associated with an origin?

Additional content may be distributed out-of-band. One question that arises in such cases is: How can out-of-band content be synchronized with media playback? There are no strong guarantees on the Web that media playback follows any particular clock, no explicit mechanism that allows an application to detect the exact position on the media timeline at a specific time, and no mechanism to trigger an action at a precise position on the media timeline. Is the available (lack of) precision enough to implement hybrid radio scenarios on the Web?

2.2.6. Fragmentation reduction

CE devices dedicated to media playback remain constrained devices that follow specific production processes. The convergence between devices in terms of capabilities and software is ongoing, but media devices have not turned into all-purpose computers yet. One notable remaining difference between most media devices and laptops or recent smartphones is that the firmware and operating system installed on media devices is very rarely updated once the device has shipped (typically because no one pays for these updates).

9 https://webassembly.org/
At the other end of the spectrum, Web technologies and Web browsers are updated almost daily.

As media devices embrace Web technologies to provide the interactive layer, one question that arises is: What is the Web platform? In other words, what are the Web technologies that media devices need to support so that they can run media Web applications?

This is one the main goals of the CTA WAVE Project\(^\text{10}\), started in 2016, and run in collaboration with W3C: to define a common baseline of Web technologies, updated annually to keep pace with the evolving Web platform, that should be supported by manufacturers of media devices. That baseline is based on technologies supported across the main browser codebases used to browse the Web on all-purpose devices. Can this baseline be used for hybrid radio devices as well? Does it need to be adjusted?

2.2.7. Improve image handling for station logos

The RadioDNS specification provides five logo sizes by way of a standard. These sizes were determined some time ago, when devices did not have high pixel density screens. Although the specification does not preclude the use of other sizes, these five mandatory sizes are felt by automotive manufacturers to be of adequate resolution.

One approach would be to define an update to the standard to include the addition of a high-resolution square (1400 square is the minimum size Apple accepts for podcast cover art) and a high resolution 16:9 image such as a full HD 1920x1080 image.

Adequate station logos are vital for hybrid interfaces which are often competing with other visually rich audio services (Spotify, Pandora) and so the requirement is to make these two images mandatory alongside the existing ‘Project Logo’ sizes.

2.2.8. Phonetic spelling of station names

As voice control is introduced into hybrid radio, there is a need to ensure voice control systems are able to both correctly recognise station names and correctly

\(^{10}\) https://cta.tech/Research-Standards/Standards-Documents/WAVE-Project/WAVE-Project.aspx
pronounce them. For example, a radio station called *Hot Sounds Liverpool FM* may also be known by locals as *Hot Radio* or *Hot Liverpool* or *Hot FM*. In other words, the public uses a local synonym for a station and not the ‘official’ station name. A voice control system needs to know these synonyms.

Likewise, when announcing the name of a radio station, a voice system must pronounce this correctly. For example, *BBC Radio Five Live*. The word “live” can be pronounced in two ways. When the word is used as a verb – to live – the I vowel is a short vowel. However, in this context, the word “live” is used as an adjective where the pronunciation is more like “lyve”. The mispronunciation can sound peculiar to the user.

To avoid this, phonetics can be provided for station names, as part of the metadata schema. There are a number of approaches which can be adopted as standards already exist. The main standards are: International Phonetic Alphabet, or a phonetic respelling system such as those established by the BBC, Wikimedia, and the major dictionary publishers – or pidgin English. Further, there is also a standard phonetic markup language – Speech Synthesis Markup Language (SSML) [11] which adequately caters for the above.

### 2.2.9. Convergence of metadata vocabularies

The HRADIO platform provides a way to discover radio programmes. Traditional radios will follow vocabularies described in the RadioDNS SPI standard [4] to describe their programmes, so that they can be indexed by radio content platforms. Pure Web radios will rather target usual search engines and follow vocabularies that these search engines understand, notably schema.org vocabularies.

A few years ago, BBC and EBU worked with the schema.org community to extend support for TV and radio programs in schema.org. These extensions are now part of the core schema.org vocabulary. Can SPI content be mapped to schema.org metadata automatically? Can schema.org metadata that describes a radio programme be mapped to SPI content automatically? Does either of these vocabularies need to be completed?

### 2.3. ACTION PLAN

Table 1 below describes the standardisation plan that project partners discussed and started to enact after the first year. No significant change was made to the plan afterwards. Resulting contributions – or rationale not to pursue a given topic – are summarized in section 3 Contributions to standardisation activities.
<table>
<thead>
<tr>
<th>Goal</th>
<th>Action</th>
<th>Who</th>
<th>Contribution / Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio tuner in native apps</td>
<td>Draft initial OMRI spec</td>
<td>IRT</td>
<td>3.1 Open Mobile Radio Interface (OMRI)</td>
</tr>
<tr>
<td></td>
<td>Finalize OMRI spec and standardise at ETSI</td>
<td>IRT</td>
<td></td>
</tr>
<tr>
<td>Radio tuner on the Web</td>
<td>Assess support for resuming work on W3C TV Control API in Media &amp; Entertainment Interest Group</td>
<td>W3C</td>
<td>3.2 TV Control API</td>
</tr>
<tr>
<td></td>
<td>Draft TV Control spec</td>
<td>W3C</td>
<td></td>
</tr>
<tr>
<td>Live content on the Web</td>
<td>Evaluate requirements on MSE</td>
<td>W3C</td>
<td>3.3 Use of EDI for DAB streaming</td>
</tr>
<tr>
<td></td>
<td>Contribute to MSE v2 incubation</td>
<td>W3C</td>
<td>3.4 Media Source Extensions</td>
</tr>
<tr>
<td></td>
<td>Explore usage and possible extensions of WebRTC technologies to reduce the distribution latency</td>
<td>W3C</td>
<td>3.5 WebRTC / WebTransport / WebCodecs</td>
</tr>
<tr>
<td>DAB decoding in a Web app</td>
<td>Identify possible technical gaps (WebUSB, WebAssembly)</td>
<td>IRT</td>
<td>3.6 Threading support on the Web</td>
</tr>
<tr>
<td>Distribution and synchronization of Web content with media</td>
<td>Gather inputs from HRADIO, organize W3C discussions on the topic</td>
<td>W3C</td>
<td>3.7 Media Timed Events</td>
</tr>
<tr>
<td>Fragmentation reduction</td>
<td>Help transition the Web Media API Community Group to a W3C Working Group</td>
<td>W3C</td>
<td>3.8 Web Media API 3.9 Media Capabilities</td>
</tr>
<tr>
<td>Improve image handling for station logos</td>
<td>Research solutions that help downstream systems and standards (e.g. RadioDNS) benefit</td>
<td>RP</td>
<td>3.10 Media Session</td>
</tr>
<tr>
<td>Phonetic spelling of station names</td>
<td>Research solutions</td>
<td>RP</td>
<td>None</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------</td>
<td>----</td>
<td>------</td>
</tr>
<tr>
<td>Convergence of metadata vocabularies</td>
<td>Explore mapping between SPI (used in RadioDNS) and schema.org (used by pure Web radios)</td>
<td>W3C</td>
<td>None</td>
</tr>
<tr>
<td>Link SPI and on-demand downloadable content for timeshifting use cases</td>
<td>Review SPI and design complementary information if needed</td>
<td>IRT</td>
<td>None</td>
</tr>
<tr>
<td>Authenticity verification</td>
<td>Integrate authenticity verification to RadioWEB (as done in DVB for HbbTV)</td>
<td>IRT</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 1 - Standardisation action plan
3. CONTRIBUTIONS TO STANDARDISATION ACTIVITIES

This section summarizes contributions to standardisation activities made throughout the project to address the candidate standardisation topics identified in section 2.2. The project did not make substantive contributions on all of these topics but rather focused on higher priorities for the project: the ability to create a common API to handle DAB streams and improvements to streaming mechanisms. Notably, no significant work was done on the phonetic spelling of station names and on the convergence of metadata vocabularies.

3.1. OPEN MOBILE RADIO INTERFACE (OMRI)

Linked to 2.2.1 Radio tuner support in native apps.

As reported in 2.2.1, preliminary discussions for the creation of a common API to access and control radio tuners from native applications had taken place in WorldDAB when the HRADIO project started. These discussions led to the creation of a task force under the umbrella of the Technical Committee in February 2017 to develop the Open Mobile Radio Interface (OMRI).

Through the HRADIO project, IRT developed the OMRI specification in WorldDAB, and submitted it to ETSI for standardisation. The OMRI specification was published by ETSI in October 2018 under the name TS 103 632 – Digital Audio Broadcasting (DAB); Open Mobile Radio Interface (OMRI); Application Programming Interface (API) [1].

IRT has also developed an implementation for Android within WP3 to validate the design of the API and promote its adoption. First implementations for iOS and Web applications have also been developed.

3.2. TV CONTROL API

Linked to 2.2.2 Radio tuner support on the Web.

The TV Control API [8], developed at W3C, aims to enable web applications to present audio and video media from broadcast TV and radio, using the media elements defined in HTML. The API provides access to programme and service information. The API had initially been identified as key for the project. However, work on the API was discontinued in 2017 (as explained in 2.2.2).

W3C conducted a series of discussions with stakeholders involved in the development of the API to better understand needs and assess interest for
resuming the work. AT&T, BBC, Comcast, Espial, Google, LG, Sony and others indicated interest for such an API in theory, but the outcome of both private and public discussions within the Media & Entertainment Interest Group\(^\text{11}\) so far is that there is not enough traction to resume the work. Most media companies are heavily investing on adaptive streaming over HTTP, and while broadcast is certainly here to stay, adding support for broadcast signals on the Web is not seen as a priority compared to improvements to IP-based streaming, especially because not all devices embed tuners.

The approach in hybrid standards such as ATSC 3.0 is rather to expose tuners to Web applications as a local network service that Web application can discover and interact with. In other words, in ATSC 3.0, Web applications are rendered on top of a video overlay that they can control through commands sent to a local WebSocket server. ATSC 3.0 defines the discovery API and application protocol. This approach is consistent with the approach taken in HbbTV, where a specific API is used to control a custom object element that renders the broadcast signal.

The HRADIO project acknowledged the lack of support for standardizing the TV Control API at the beginning of the project. This situation did not change significantly throughout the project. Convergence between broadcast and broadband (streaming, VOD and catch-up, and offline distribution) technologies remains one of the main hurdles raised by content providers though, especially when they create interactive experiences that mix pure media content and non-media content such as slideshows, overlays, and user interaction. W3C has started to explore the possibility to organize a workshop on Bundling Interactive Media Content on the Web\(^\text{12}\) through initial exchanges with representatives from key media companies and organizations such as ATSC, CTA WAVE, HbbTV, 3GPP, and MPEG. A program committee is being assembled as the project ends for a workshop that was to take place at the end of June 2020, hosted by IRT, but which got postponed due to the ongoing pandemic.

The workshop would bring clarity on possible convergence paths across distribution channels. Convergence may favour Web technologies (where the notion of

\(^{11}\) See draft call for participation for a possible (but unconfirmed) workshop on Bundling Interactive Media Content on the Web at: [https://www.w3.org/2020/02/media-bundle-workshop/](https://www.w3.org/2020/02/media-bundle-workshop/)
D6.2: Standardisation plan and report

package is more ingrained at the HTTP level), media streaming technologies (where the media container is the natural package but needs to be generated on-the-fly) or broadcast technologies, or neither as there may not exist one practical solution for all scenarios. For instance, in the radio context, it seems useful to leverage the fact that the size of radio packets remains quite reasonable to simplify the production workflow. This is exactly what the HRADIO project did: instead of adopting complex adaptive streaming algorithms such as the ones used to stream video content on the Web, which would have required re-packaging of radio content for broadband delivery, the project rather chose to use for streaming the exact same DAB packets prepared for broadcast. Practically speaking, IRT developed an implementation of the OMRI API in WebAssembly for use in regular Web applications. Such applications simply fetch DAB packages over HTTP, WebSocket or a WebRTC data channel, then decode the DAB package in real-time for playback. This approach allows broadcasters to reuse their broadcasting production pipeline and to simply attach a streaming-over-IP module at the end of it to stream the resulting packets.

3.3. USE OF EDI FOR DAB STREAMING

Linked to 2.2.3 Live content support on the Web.

The decision to use the DAB system as the technical basis for HRADIO application development means that a DAB+ signal must be available in all usage scenarios. In general, the OMRI implementation supports the usual DAB-USB receiver sticks. On mobile devices, however, the external receivers are a foreign body and are quite unwieldy for the user, especially with the antenna required for reception. But even in places where no DAB+ signal is available, it should be possible to switch to IP reception - in the hybrid idea of the project. The HRADIO project has therefore decided to rely on DAB and to transmit the multiplex in IP streams where the radio services have to be transmitted via IP networks. In the DAB standard family there is the so-called EDI standard for this. EDI was originally designed to feed DAB ensemble data from the multiplexer to the transmitter sites. An EDI ensemble contains several individual radio services and usually has a bandwidth of approx. 2.5 Mbit/s. For the unicast transport of a radio service to the client it is of course not necessary to transmit the whole ensemble. For this reason, a server software was implemented, which uses the broadcasters standard DAB multiplex and generates single service EDI streams to be delivered over HTTP/HTTPS to clients.

This technical solution has great advantages for all parties involved. The application developers always work in the same DAB world. The services that are used (besides the audio signal) are “in band”, synchronous and their format and behaviour are
standardized. A transmitter that already generates DAB services for terrestrial transmission can send them into the EDI splitter without additional effort and use them as IP stream.

Currently, the EDI format adds a significant amount of overhead when used for single service point to point IP distribution. However, first calculations show a great potential to reduce overhead. In the connection-oriented TCP distribution of EDI services it is sufficient to transfer the FIC only in the first packets until all, most static, data is delivered.

The tag, length, value approach used by EDI also opens up for proprietary extensions without losing compatibility to other standard receivers. As per definition, all unknown tags shall be ignored by the receiver. Thus, enabling richer and personalized user experiences within the same, standard compliant stream. To make these changes happen, work needs to be done in relevant working groups with WorldDAB.

3.4. MEDIA SOURCE EXTENSIONS

Linked to 2.2.3 Live content support on the Web.

Different approaches can be envisioned to lower the latency of distribution of media content on the Web. One approach is to improve the efficiency of adaptive streaming mechanisms. This is ongoing e.g. with the progressive adoption of chunked encoding in MPEG DASH and HLS\(^\text{13}\), whereby media segments are subdivided into a series of smaller chunks that can be passed through the chain immediately as they become available. This approach is promising and could bring the latency of distribution on the Web on par with the broadcasting world. The HRADIO project is monitoring progress but does not plan to contribute to these activities.

Another approach to reducing the latency of distribution is to look at the latency introduced on the client device during media playback. MSE was developed for on-demand scenarios and does not specify any requirements in terms of playback

---

latency and does not contain any hook that a Web application could use to control the amount of buffering that a Web browser uses before it starts rendering frames.

Similarly, the first version of MSE did not support the ability to switch between media chunks encoded using different codecs.

W3C discussed the standardisation of a new version of MSE with media companies and browser vendors, pondering interest to standardize and implement the codec switching feature for which a technical solution\(^{14}\) had been proposed, and a low-latency mode\(^{15}\). Discussions on MSE and on other media-related topics showed some momentum to resume standardisation efforts around media within W3C, which eventually led to the creation of the W3C Media Working Group\(^ {16}\) in May 2019. By charter\(^ {17}\), the Media Working Group is set to develop a revision of MSE that notably adds the codec switching feature mentioned above. The group is also adding new coded frame eviction policies\(^ {18}\) to MSE to give web applications greater control over how the implementation manages buffered media. This both allows web applications to have greater control over the playback latency, but also to have greater control over how the user agent should react to gaps in live streams (whether to stall playback or skip them). The project supported the exploration phase prior to the creation of the working group, and the position of the W3C staff contact in the working group after its creation.

### 3.5. WEBRTC / WEBTRANSPORT / WEBCODECS

*Linked to 2.2.3 Live content support on the Web.*

Another interesting approach to reduce the latency of distribution of media content on the Web is to look at whether other distribution mechanisms could be used, borrowing from real-time technologies used in WebRTC to enable real-time communications between peers. W3C raised the use of WebRTC for media content


\(^{15}\) [https://github.com/w3c/media-and-entertainment/issues/6](https://github.com/w3c/media-and-entertainment/issues/6)

\(^{16}\) [https://www.w3.org/media-wg/](https://www.w3.org/media-wg/)

\(^{17}\) [https://www.w3.org/2019/05/media-wg-charter.html](https://www.w3.org/2019/05/media-wg-charter.html)

distribution as an issue\(^{19}\) in the Media & Entertainment Interest Group and reached out to media companies in that field to organize a discussion call within that group. Google presented shorter-term and longer-term options to the group\(^ {20}\), notably using QUIC for transport.

The possibility to use WebRTC technologies to distribute content on the Web would allow broadcasters to get down to lower than one second distribution latency. For radio, this could ease or enable real-time interactions with radio listeners.

In practice, discussions showed that having to deploy a complete WebRTC set of technologies is somewhat complex and overkill for media streaming scenarios. WebRTC is geared at bi-directional peer-to-peer exchanges whereas streaming follows a more classical client/server architecture with a one-to-many distribution model. The existing coupling between transport and media encoding/decoding in WebRTC works quite well for one-to-one media channels. It does not scale very easily when a server needs to distribute the same media content to multiple peers at once.

Google proposed to incubate solutions to decouple transport from media encoding/decoding in order to overcome these restrictions. This includes WebTransport \(^{14}\), which allows data to be sent and received between a browser and server, implementing pluggable protocols underneath with common APIs on top, based on QUIC. The API is similar to WebSocket in that it exposes bidirectional connections between a client and a server, but allows to further reduce the latency of network communications between a client and a server, and also supports multiple streams, unidirectional streams, out-of-order delivery, and unreliable transport. This would ease low-latency streaming of media chunks from a server to a client.

A WebCodecs API\(^{21}\) has been proposed as well to expose media encoders/decoders to Web applications, to give them more flexibility and control over how the media gets decoded, in live streaming scenarios for instance. Combined with WebTransport, these two APIs could open new ways to stream very low-latency media.

\(^{19}\) https://github.com/w3c/media-and-entertainment/issues/1

\(^{20}\) https://www.w3.org/2018/10/me-minutes.html

\(^{21}\) https://github.com/wicg/web-codecs
media on the Web.

The project supported W3C’s efforts to gather feedback on the WebTransport and WebCodecs ideas among the media industry. These proposals received warm support during the Workshop on Web Games\textsuperscript{22} that W3C organized in June 2019 and were heavily discussed in breakout sessions\textsuperscript{23} during W3C’s Technical Plenary and Advisory Committee (TPAC) in September 2019 in Fukuoka.

At the end of the HRADIO project, the WebTransport and the WebCodecs API are being incubated in the Web Platform Incubator Community Group within W3C. The IETF created a dedicated WebTransport Working Group\textsuperscript{24} in March 2020 to define protocol extensions that support the development of the API. Discussions regarding the possible creation of a WebTransport Working Group within W3C to standardise the API have started. The WebCodecs API was added to the charter of the W3C Media Working Group, mentioned above, as a potential normative deliverable of the group, which means the group may adopt and standardise the API once incubation has sufficiently advanced.

### 3.6. THREADING SUPPORT ON THE WEB

Linked to 2.2.4 DAB decoding in a Web app.

Workers provide support for threads on the Web. However, communication between workers on the Web is only possible through message passing, which only allows to pass strings, or through unstructured shared memory buffers (using SharedArrayBuffer). Support for threading in WebAssembly follows the same design. Message passing is impractical in streaming scenarios where chunks of bytes need to be passed multiple times per second across threads (and which would have to be serialized/deserialized as strings each time). SharedArrayBuffer is much more suitable for streaming scenarios.

Web browsers unfortunately had to drop support for SharedArrayBuffer due to

\[\text{\footnotesize \cite{webtransport, webtransport-wp, webcodecs}}\]

\[\text{\footnotesize \cite{ietf-webtrans}}\]
Spectre and Meltdown security vulnerabilities. Good news is that browser vendors reached consensus on a mechanism that would allow pages to opt-in to `SharedArrayBuffer`, based on new HTTP headers, and are slowly re-introducing support for `SharedArrayBuffer` across devices... except on Android devices as of March 2020. As a result, the WebAssembly implementation of the OMRI API requires running Chromium with a dedicated flag to enable `SharedArrayBuffer` in the meantime.

### 3.7. MEDIA TIMED EVENTS

*Linked to 2.2.5 Distribution and synchronization of Web content with media.*

Through the project, W3C organized and contributed to a series of discussions on the combination of media content with other types of content. Contributions include:
Review of the “Carriage of Web Resource in ISOBMFF” proposal developed at MPEG, and organization of relevant discussions with the Technical Architecture Group (TAG) at W3C. The proposal describes a mechanism to store and deliver Web data in ISOBMFF containers to enable enriching audio/video content, as well as audio-only content with synchronized, animated, interactive Web data, including overlays. The review identified several issues with the proposal, summarized in a letter sent to MPEG on behalf of the Media & Entertainment Interest Group. Issues raised led to the removal of JavaScript tracks from the proposal before publication as an ISO standard [15].

Participation in the Media Timed Events Task Force within the Media & Entertainment Interest Group to specify use cases and requirements for a generic cue mechanism for the Web [10], that could e.g. be used to expose MPEG DASH emsg box events. This work led to the incubation of the DataCue proposal [25] in the Web Platform Incubator Group, led by the BBC, to resume work on a generic concept of cue, that had been previously dropped from HTML5 for lack of browser support. It also helped identify two issues related to the handling of cues within HTML, one to clarify timing accuracy expectations for cue events [26], the other to suggest to relax cues definitions to allow cues whose end time is the end of the stream (typically useful in live streaming scenarios). These issues are under discussion at the end of the HRADIO project.

Discussions on synchronization needs within the Media & Entertainment Interest Group, including scenarios that require frame accuracy precision [27], and on mechanisms best suited to meet these needs. W3C summarized options [28] during the face-to-face of the Media & Entertainment Interest Group in September 2019, which led to starting more generic work on use cases and requirements for Media Production on the Web [29] (frame accuracy being one of the requirements in that space) within the Interest Group. This work is still at early stage when the project ends.

---

25 https://github.com/WICG/datacue/blob/master/explainer.md
26 https://github.com/whatwg/html/issues/5306
27 https://github.com/w3c/media-and-entertainment/issues/4
28 https://www.w3.org/2019/Talks/TPAC/frame-accurate-sync/
29 https://github.com/w3c/me-media-production/
3.8. WEB MEDIA API

Linked to 2.2.6 Fragmentation reduction.

The CTA WAVE Project started an effort in 2017 to define a common baseline of Web technologies that should be supported by all media devices. The Web Media API specification is the result of these discussions. It is targeted at device manufacturers so that they can have a reference document at their disposal when they integrate support for Web technologies in their devices. It is also targeted at content providers and developers in general so that they know what technologies they can rely on being supported across devices.

This baseline is relevant for hybrid radio devices. The Web Media API is developed in a Community Group\(^\text{30}\). W3C investigated the possibility to standardise the specification within a Working Group. The standardisation process is not entirely adapted to the effort at hand, though: standardisation efforts at W3C are often loosely bounded in time, and the goal of this effort is to produce annual specifications that follow the evolution of the Web platform. CTA WAVE participants decided against transitioning to a Working Group for now, and in favour of pursuing the work in the Web Media API Community Group.

One interesting aspect of this effort is testing, needed to certify devices. The CTA WAVE project invested time and resources (through Fraunhofer FOKUS) into producing a test suite to assess conformance to the Web Media API specification. Tests are extracted from tests in the Web Platform Tests project\(^\text{31}\), a community-based project that contains official test suites for Web standards. W3C reviewed the testing tools produced as part of this effort, and worked with Comcast, CTA and Fraunhofer FOKUS to integrate the test runner that Fraunhofer FOKUS developed for use on constrained CE devices back into the Web Platform Tests project. Fraunhofer FOKUS sent a pull request against the Web Platform Tests GitHub repository to integrate the Web Media API Snapshot test runner to Web Platform Tests in January 2020\(^\text{32}\). This Pull Request is still being refined, in particular to add tests, when the

\(^{30}\) https://www.w3.org/community/webmediaapi/

\(^{31}\) https://github.com/web-platform-tests/wpt

\(^{32}\) https://github.com/web-platform-tests/wpt/pull/21323
3.9. MEDIA CAPABILITIES

Linked to 2.2.6 Fragmentation reduction.

While not strictly speaking a mechanism that helps reduce fragmentation, the ability to access precise media capabilities of the underlying platform is key to addressing the differences between those platforms. The W3C Media Working Group standardises the Media Capabilities specification [12] which defines an API to expose that information, with a view to replacing the more basic and vague canPlayType() and isTypeSupported() functions defined in HTML and in Media Source Extensions.

3.10. MEDIA SESSION

Linked to 2.2.7 Improve image handling for station logos.

To compete with rich audio streaming applications such as Spotify, radio applications need to integrate with the media interface provided by the underlying system, for instance to render information about the current track (title, album, cover art), along with playback actions in the notification bar on smartphones. While native applications can directly integrate with the notification bar across systems, this has remained out of reach for Web applications. The Media Session [13] specification, standardised by the W3C Media Working Group, aims to bring Web applications on a par with native applications in that domain.

3.11. LINK BETWEEN SLIDESHOWS AND DYNAMIC LABELS

Linked to 2.2.5 Distribution and synchronization of Web content with media

Visual components are increasingly being added to Radio as well. On high-resolution screens of smartphones, tablets or car navigation systems, the listener also expects a visual representation of what he or she has heard. Pictures showing presenters, guests or cover art are considered standard. The local and server based timeshift systems developed for HRADIO use trigger points in the DAB multiplex (Dynamic Label+) to segment the timeshift buffer according to the radio contributions. Unfortunately, there is no way in the DAB specifications to uniquely assign a slideshow image to a DynamicLabel tag. For example, if a new music title is signaled via DynamicLabel, the HRADIO application can currently only assign a slideshow picture to this segment on a best effort basis.
During the WorldDAB Technical committee meeting on 28th Dec. 2019 at IRT in Munich, IRT suggested an amendment to the relevant specifications, which would enable such an assignment, on behalf of the HRADIO project. The TC planned to have an extensive discussion of such a proposal during the upcoming meeting in February 2020 in Geneva. Unfortunately, this meeting was cancelled in February, and next meeting has not been planned yet (due to Covid-19 pandemic). IRT plans to propose and discuss these extensions at next WorldDAB-TC.
4. OTHER STANDARDISATION UPDATES

Some of the areas identified in section 2.2 Candidate standardisation areas have also progressed on the standardisation front during the lifetime of the project without direct contributions from the HRADIO project.

The project tracked progress on media-related standardisation activities to inform partners about upcoming features and new technologies. W3C integrated those relevant to the project, along with their current status, in the Overview of Media Technologies for the Web [33] roadmap document (with a major update early 2020). This document serves as a basis for discussion within the W3C Media and Entertainment Interest Group to track standardisation progress and remaining gaps.

This section presents main updates since the beginning of the project, and as of March 2020, which may impact the HRADIO platform on the long term.

4.1. AUTOPLAY POLICY DETECTION

*Linked to 2.2.3 Live content support on the Web.*

To preserve bandwidth, memory and battery on mobile, and prevent possibly unwanted media playback, browsers have put autoplay policies in place and may deny automated playback of media content. The Autoplay Policy Detection [34] project will let applications know whether autoplay will succeed for a given media element. The project is being standardized in the W3C Media Working Group.

Support for this specification in Web browsers would allow HRADIO applications to adjust their user interface to cope with the different autoplay policies.

---

[33] https://w3c.github.io/web-roadmaps/media/

[34] https://github.com/w3c/autoplay
4.2. NETWORK QUALITY MONITORING AND PREDICTION

Linked to 2.2.3 Live content support on the Web.

The Web and Networks Interest Group was created in W3C in 2019 to explore solutions for web applications to leverage network capabilities in order to achieve better performance and resources allocation, both on the device and network. The group has started to investigate use cases and requirements for exposing network quality parameters to Web applications. Adaptive streaming of media content is a typical scenario where the ability to predict fluctuations of the available bandwidth in real-time would allow to optimize playback and improve the quality of experience.

Although distribution of radio content requires less network bandwidth than video, improvements to adaptive streaming algorithms will benefit HRADIO applications over time, e.g. because immersive radio experiences may embed video feeds.

4.3. MEDIASTREAMTRACK CONTENT HINTS

Linked to 2.2.3 Live content support on the Web.

The W3C WebRTC Working Group, which standardizes peer-to-peer communication technologies, has published a new specification called MediaStreamTrack Content Hints [16]. This specification exposes a media-content hint about the type of content that is being consumed. Such a hint allows Web browsers to make more informed decision of the processing algorithm to use on the consumed content.

For radio content, the hint notably allows to specify whether the audio track contains mostly speech data – on which noise suppression and other algorithms to boost intelligibility of the incoming signal could be applied – or music – on which such algorithms should not apply.

This specification will help improve the quality of the playback experience in live streaming scenarios when WebRTC technologies are used to distribute the content.

35 https://www.w3.org/wiki/Network_Quality_Monitoring_and_Prediction
4.4. MEDIA FEEDS

Linked to 2.2.9 Convergence of metadata vocabularies.

The Media Feeds [17] specification is a recent (March 2020) proposal from Google to allow Web applications to advertise a feed of personalized media recommendations for the user that could be aggregated by the Web browser in some native user interface. The proposal re-uses terms of the schema.org vocabulary and defines a mechanism by which the Web browser can discover and fetch personalized feeds.

The Media Feeds API is under incubation in the W3C Web Platform Incubator Community Group.

Longer term, this specification could impact the way HRADIO applications convey and advertise the electronic program guide for the underlying channel.

4.5. EFFICIENT LOW-LEVEL AUDIO HARDWARE DEVICE ACCESS

Linked to 2.2.4 DAB decoding in a Web app.

The W3C Audio Working Group recently re-chartered, partly to work on a new specification for efficient low-level audio hardware device access. This specification will aid the porting of existing audio code bases, free developers from the constraints of a fixed audio quantum size, and give access to properties of the audio system such as I/O device selection, multi-channel I/O support, and configurable sample rates. The Audio Device Client[36] proposal was notably presented during the W3C Workshop on Web Games[37], organized in June 2019, and could serve as basis for these discussions.

The specification could be useful in a radio context to ease cross-compilation of existing audio processing code present in the broadcasting pipeline to handle DAB streams, for use in Web applications.

[37] https://www.w3.org/2018/12/games-workshop/report.html#audio
4.6. IMPROVING SPOKEN PRESENTATION ON THE WEB

Linked to 2.2.8 Phonetic spelling of station names.

The W3C Accessible Platform Architectures Working Group created a Pronunciation Task Force in 2019 to develop normative specifications and best practices guidance to provide for proper pronunciation in HTML content when using text to speech (TTS) synthesis.

The task force published a first draft document titled “Explainer: Improving Spoken Presentation on the Web” [18] that studies mechanisms to allow content authors to include spoken presentation guidance in HTML content, by embedding SSML [11] characteristics.

Such a mechanism would allow HRADIO applications on the web to clarify pronunciation of station names.
5. CONCLUSIONS

The HRADIO project identified several candidate areas where lack of standards impedes or slows down the realization of hybrid radio scenarios, starting with the lack of a common interface to access and control radio tuners from applications, and including mechanisms to optimize live streaming on the Web.

Project partners investigated candidate areas and made contributions to progress relevant standardisation efforts. Main contribution is the development of the Open Mobile Radio Interface (OMRI) at WorldDAB and its standardisation at ETSI. The project has also allowed standardisation work on media streaming technologies to resume at W3C through the creation of the Media Working Group. This working group develops technologies that will directly benefit or simplify the HRADIO platform, such as a revision of the Media Source Extensions standard to add support for codec switching, and work on Media Session to integrate a playback experience with the media interface provided by the underlying system.

Further areas (see 3.3 and 3.11) for coordination and standardisation became relevant for discussions in WorldDAB-TC and will be followed up by IRT.

The project also helped investigate and clarify the status of the TV Control API, raise requirements, create issues and propose solutions to improve media timed events support, and discuss low-latency distribution mechanisms for media content on the Web.

The HRADIO project helped seed standardisation discussions and actual standardisation efforts that will outlive the project. This includes explorations on bundling of interactive media experiences on the Web, on media production use cases and on low-latency streaming technologies. Project partners will remain engaged in relevant standardisation activities after the end of the project.
REFERENCES

[1] ETSI TS 103 632 v.11.1, Digital Audio Broadcasting (DAB); Open Mobile Radio Interface (OMRI); Application Programming Interface (API), October 2018, URL: https://www.etsi.org/deliver/etsi_ts/103600_103699/103632/01.01.01_60/ts_103632v010101p.pdf


