Grant Agreement No.: 687645
Research and Innovation action
Call Topic: H2020 ICT-19-2015

ORPHEUS
Object-Based Audio Experience

Object-based broadcasting – for European leadership in next generation audio experiences

D4.1: Requirements for Representation, Archiving and Provision of Object-based Audio

Version: v1.4

Deliverable type	R (Document, report)
Dissemination level	PU (Public)
Due date	01/06/2016
Submission date	17/06/2016
Lead editor	Andreas Silzle (FHG)
Authors	Andrew Mason (BBC), Michael Meier (IRT), Simone Füg (FHG), Andreas Silzle (FHG)
Reviewers	Niels Bogaards (Elephantcandy), Halid Hrasnica (EURES)
Work package, Task	WP 4, T4.1, T4.2, and T4.3
Keywords	File and streaming formats, BW64, meta data, ADM, MPEG-H

Abstract

This deliverable describes the requirements for representation, archiving and provision of object-based audio. Representation includes file and streaming formats for object-based audio. Provision is the distribution to end-user, including IP delivery, unicast streams and file downloads. For both file and streaming formats interoperable metadata have to be used. Among the requirements for the different formats are metadata support, existing standards, file size, compression, and support of advanced audio. The most important requirements for streaming formats are related to synchronization, existing standards, unicast and multicast transmission, latency, bitrate reduction, synchronization of audio signals, quality of service and specific requirements for streaming object-based audio metadata. Constrains and requirements of interoperability between different formats and backward compatibility are explained.

Document revision history

Version	Date	Description of change	List of contributor(s)
v0.1	20/05/2016	Start with BBC input	Andrew, Andreas
v0.2	24/05/2016	Add FHG input	Michael W., Andreas
v0.3	24/05/2016	FHG input	Andreas
v0.5	24/05/2016	FHG input	Simone
v0.6	27/05/2016	FHG input	Simone
v0.62	30/05/2016	FHG input	Andreas
v1.1	03/06/2016	Review	reviewers
v1.3	10/06/2016	Integrate reviews	Andreas
v1.4	17/06/2016	Final editing and corrections	Uwe

Disclaimer

Copyright notice

[1] http://creativecommons.org/licenses/by-nc-nd/3.0/deed.en_US

Executive Summary

This deliverable describes the requirements for representation (a file or stream containing object-based audio), archiving (long-term storage of object-based audio), and provision (the distribution to end-user, including IP delivery, unicast streams and file downloads). These formats have to work for the signal chain blocks of the implementation architecture: recording, pre-production and mixing, contribution (e.g. OB van), radio studio, distribution or emission to the end-user and their reception (usage of the end-user). For both file and streaming formats interoperable metadata have to be used. For this the ITU-R defined Audio Definition Model (ADM) is the most recent and envisioned standard. Other formats are listed. The long list of requirements for file formats is specified; among them are metadata support, existing standards, file size, compression, and support of advanced audio.

Several streaming formats are listed, like DASH, serialized ADM, streaming of MXF, HLS. The requirements are explained; among them are synchronization, existing standards, unicast and multicast transmission, latency, bitrate reduction, synchronization of audio signals, quality of service and specific requirements for streaming object-based audio metadata. Because there is not one format which can fulfil all requirements and which can practically be implemented for all use cases, several formats have to work together. The difficulty and requirements of interoperability between these different formats is explained, two examples (AVB and AES67/Ravenna) are given.

The backward and forward compatibility issue with legacy systems is explained: how to handle new content on legacy emission systems, and how to handle old content on new production systems.

The above listed information and requirements are the basis for the selection and new definitions of file and streaming formats, selected for the reference architecture and the pilot implementations.

Table of Contents

List of Figures

List of Tables

Abbreviations

AC-4	Digital audio compression version 4
ADM	Audio definition model
AES67	Standard for audio-over-IP interoperability
AJAJ	Asynchronous JavaScript and JSON
ASI	MPEG-H Audio Scene Information
AVB	Audio Video Bridging
BBC	British Broadcasting Corporation
BIFS	Binary Format for Scenes
BR	Bayerischer Rundfunk
BW64	Broadcast Wave 64Bit
BWF	Broadcast wave
CAD	Computer aided design
DASH	Dynamic Adaptive Streaming over HTTP
EC	European Commission
FHG	Fraunhofer Gesellschaft
HLS	HTTP Live Streaming
HOA	Higher Order Ambisonics
IP	Internet Protocol
IPF	Instantaneous Playout Frames
ITU-R	International Telecommunication Union, Radiocommunication Sector
MDA	Multi-dimensional audio
MHAS	MPEG-H Audio Stream
MPD	Media Presentation Description
MPEG	Moving Picture Experts Group
MXF	Material Exchange Format
MXF	Material Exchange Format
OB	Outside broadcasting
PCM	Pulse-code modulation
QoS	Quality of Service
RIFF	Resource Interchange File Format
SMIL	Synchronized Multimedia Integration Language
VRML	Virtual reality modelling language
XML	Extensible Markup Language

1 Introduction

An important part of the ORPHEUS project is the selection and definition of appropriate exchange formats among different blocks of the envisioned ORPHEUS architecture presented in Fig. 1; from the production side, via the distribution, until the reception at end-user sides. These formats have to fulfil various requirements for the object-based audio; off-line usage as file formats and real-time usage for streaming formats. Interoperability, back-ward compatibility, and audio over IP are additional requirements, which have to be defined for the different processing blocks and use-cases. The considered use cases for the formats are listed in Table 1.

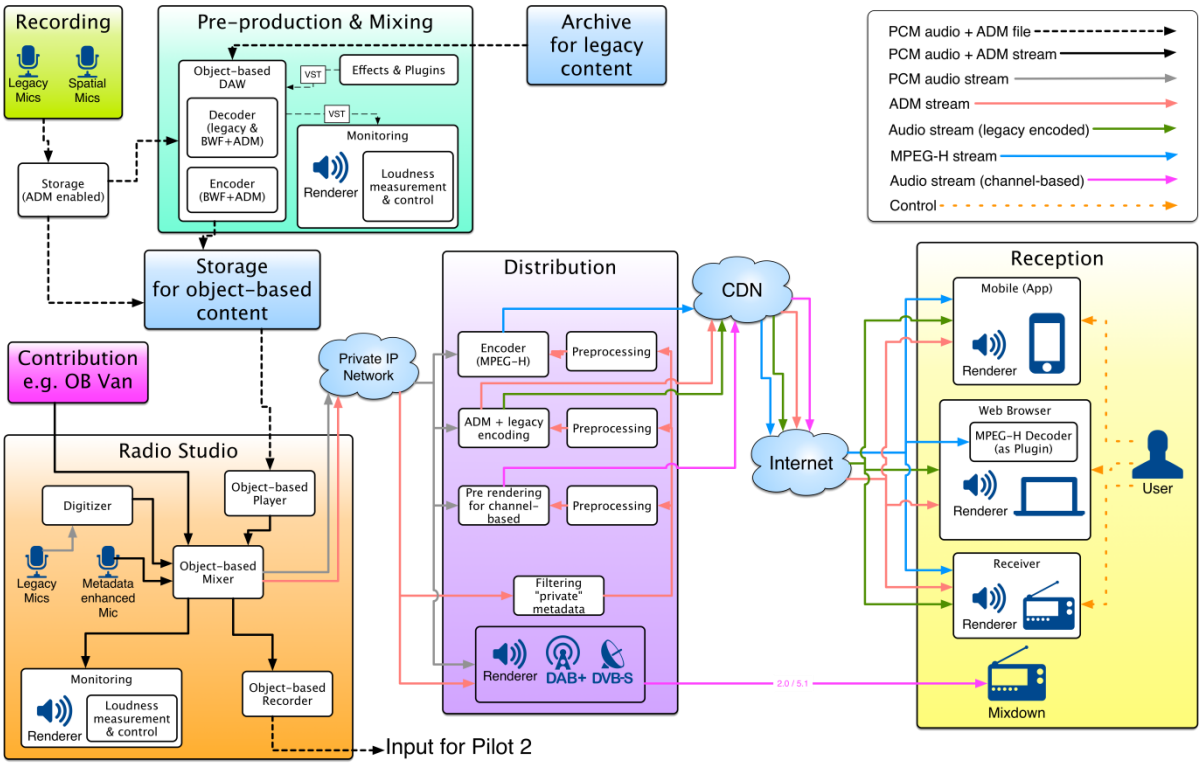

Figure 1: Pilot implementation architecture

#	Use Case	Explanation
1	Representation	File or stream containing object-based audio
2	Archiving	Long-term storage of object-based audio
3	Provision	Distribution to end-user, including IP delivery, unicast streams and file downloads

Table 1: Use cases for formats

The signal chain blocks, which can be identified in Figure 1, are presented in Table 2.

#	Signal Chain Block
1	Recording
2	Pre-production and mixing
3	Contribution (e.g. OB Van)
4	Radio studio
5	Distribution or emission to the end-user
6	Reception, usage of the end-user

Table 2: Signal chain blocks

The deliverable is organised as follows. In Section 2, requirements on metadata in the context of the object-based audio are explained. In the following sections 3 and 4, requirements for file and streaming formats are analysed. Some of them will even have to be conceived. Interoperability issues of different formats in respect to a coherent service provision are discussed in Section 5, whereas backward and forward compatibilities of the formats are analysed in Section 6. The conclusions are presented in Sec. 7.

2 Metadata in the Context of Object-based Audio

Audio for broadcasting is evolving towards a so-called “immersive” (often also referred to as “3D audio” in other context “advanced audio”[2]) and interactive experience of sound that appears to come from any direction around the listener, including above and below. This new experience requires the use of more flexible audio formats, as a fixed channel-based approach is not sufficient, so that new representation principles, such as object-based audio, are needed.

The central requirement for allowing the distribution of 3D audio with an object-based approach, whether by file or by streaming, is that metadata fully describes the audio content, as object-based audio is represented by a number of separate sound events and associated side information.

Therefore, a metadata model is needed to describe the complete format of the object-based audio content. The transport of metadata therefore is crucial for representation, archiving and provision of object-based audio.

The most recently defined and most complete metadata model that supports the description of object-based audio (as well as channel-based and scene-based audio) is the “Audio Definition Model” (ADM), which is described below.

2.1 Audio Definition Model (ADM)

The Audio Definition Model (ADM) is a general description model, giving a complete description of audio configurations in a formalized way. It is an open common metadata model, which is predominantly designed for the use in programme production and international exchange of audio material. The ADM is primarily represented in XML as its specification language, though it could be mapped to other languages such as JSON (JavaScript Object Notation).

The Audio Definition Model (ADM) was first established in EBU Tech 3364 and as part of the EBU Core schema and has since then been extended by the ITU to become Recommendation ITU-R BS.2076 [1], which contains additional features over EBU Tech 3364 version 1.0 [2].

The ADM can be used in combination with “Resource Interchange File Format” (RIFF/WAV)-based file formats. For that purpose, a new wave-based file format called “Broadcast Wave 64Bit” (BW64) was defined by the ITU as a more flexible successor of the "Broadcast Wave File" (BWF) format. BW64 allows the definition of a high number of tracks and file sizes bigger than 4GB by using 64Bit signalling and is therefore also capable to transport object-based audio. The BW64 is defined in Recommendation ITU-R BS.2088 “Long-form file format for the international exchange of audio programme materials with metadata”, [3].

To describe the workflow with ADM metadata in BW64 files, an accompanying report ITU-R BS.2388 “Usage guidelines for the audio definition model and multichannel audio files” [4] was published, describing typical use cases and recommended practices.

The following metadata descriptors are specifically defined for object-based content in ADM. These descriptors are XML attributes that provide a concise explanation of the content:

- **Position:** The object’s position in 3D space can be given in spherical coordinates (azimuth, elevation, distance) or in Cartesian coordinates (X, Y, Z)
- **Gain:** This parameter describes a gain (linear gain value) to be applied to the audio essence of the object when being rendered.

[2] ITU-R

- **Extent/Size:** The extent of an object can be defined by three parameters in the ADM: width, height and depth. When a spherical coordinate system is used, the horizontal and vertical extend are given in degrees and the depth as a distance ratio. For Cartesian coordinates, the extent is given as X-width (width), Y-width (depth) and Z-width (height) in normalized units.
- **Divergence:** If the divergence of an object (objectDivergence parameter) is bigger than 0, two additional objects should be rendered in addition to the original object at defined positions. The gain balance between the three object instances is controlled by the value of the objectDivergence parameter.
- **Diffuseness:** The “diffuse” parameter describes the diffuseness of an object (if it is diffuse or direct sound) as a value between 0.0 (not diffuse) and 1.0 (totally diffuse).
- **channelLock:** The channelLock parameter informs a renderer to playback the object by the nearest speaker, rather than rendering to the given object’s position. A maxDistance parameter can be defined, such that the channelLock is only applied if the object is within a specified distance from a speaker.
- **jumpPosition:** The jumpPosition is used to control the renderer’s temporal interpolation of the object position. If set to 1 the position will be interpolated over a period set by an additional attribute called interpolationLength. An interpolationLength value of zero will mean the object jumps without interpolation.
- **screenEdgeLock:** The screenEdgeLock attribute allows an object to be positioned on the edge of the screen, e.g. locking the object to a corner of the screen.
- **zoneExclusion:** The zoneExclusion parameter indicates which speaker/room zones the object should not be rendered through. Zones can be defined by the use of the zone sub-element.
- **screenRef:** This parameter indicates whether the object is screen-related (flag is equal to 1) or not (flag is equal to 0).
- **Importance:** This parameter describes the importance of an object. It allows a renderer to discard an object below a certain level of importance (0 to 10) if necessary.

The ADM metadata support a large set of options to describe interactive and immersive audio. Some of them are described below.

- Additional tracks (e.g. dialogue tracks) can be provided, e.g. a director's commentary or additional sports commentaries.
- Delivery of multiple content versions, e.g. of multiple separate languages. Different language versions can be provided as objects, instead of pre-defined language mixes. One language can then be selected for playback, e.g. by the end user. Different versions of content can be offered, e.g. a sports event with multiple stadium atmospheres, one in favour of the home team and one in favour of the away team.
- The metadata can identify different content types, such as dialogue or ambience. Therefore, the balance between different sounds is adjustable in a convenient way, depending on user preference or the environment for better intelligibility.
- In the case of movies or other audio-visual content, the audio-visual coherence at the end user should remain consistent for different screen-sizes. The ADM metadata therefore contains the production screen size and the identification of screen-related objects, such that a renderer can scale the audio scene according to the reproduction screen.

As an open format, the ADM is easily accessible by the audio community to be used in program production and international exchange of audio material.

It is therefore considered as a good possibility for interoperable delivery of object-based audio, program production and international exchange of audio material.

The transport of metadata either in the ADM format or in an equivalent format compatible with the ADM is therefore considered as one of the main requirements for representation, archiving and provision of object-based audio.

In the next paragraphs further audio and/or meta data streaming formats are described.

2.2 MDA Streaming Format

"The foundation of DTS:X is MDA, DTS' license fee-free, open platform for creation of object-based immersive audio. MDA gives movie studios unprecedented control over the specific placement, movement and volume of sound objects. The platform also enables sound engineers to "mix once" for both immersive and conventional cinemas in a combined object- and channel-based audio format, allowing content to be easily distributed beyond the theatre for streaming, broadcast, optical media and more."[3] [4],

"Open, royalty-free, uncompressed, production format and standard

- Supports immersive audio: Object-Based, Channel-Based, Scene-Based (HOA) content
- Supports unrestricted number of channels and/or objects
- Specification includes: Metadata + Bitstream + Renderer
- DTS Stewardship: Specification, Reference Tools, Reference Renderer"[5]

Other less recent file formats with metadata are described in the following paragraphs.

2.3 Virtual Reality Modelling Language (VRML/X3D)

"VRML (Virtual Reality Modelling Language) is a standard file format for representing 3-dimensional (3D) interactive vector graphics, designed particularly with the World Wide Web in mind. It has been superseded by X3D.[6]

X3D is a royalty-free ISO standard XML-based file format for representing 3D computer graphics. X3D features extensions to VRML (e.g. CAD, Geospatial, Humanoid animation, NURBS etc.), the ability to encode the scene using an XML syntax as well as the Open Inventor-like syntax of VRML97, or binary formatting, and enhanced application programming interfaces (APIs)."[7]

A subset of X3D is XMT-A, a variant of XMT, defined in MPEG-4 Part 11. It was designed to provide a link between X3D and 3D content in MPEG-4 (BIFS).

2.4 MPEG-4 Systems/AudioBIFS

"MPEG-4 Part 11 Scene description and application engine was published as ISO/IEC 14496-11 in 2005. MPEG-4 Part 11 is also known as BIFS, XMT, MPEG-J. It defines:

---

[3] http://www.prnewswire.com/news-releases/welcome-to-dtsx---open-immersive-and-flexible-object-based-audio-coming-to-cinema-and-home-300063437.html

[4] http://dts.com/dtsx

[5] https://www.itu.int/en/ITU-R/study-groups/workshops/2015-TFAB/PublishingImages/Pages/default/Presentation%202-2.pdf

[6] https://en.wikipedia.org/wiki/VRML

[7] https://en.wikipedia.org/wiki/X3D

- the coded representation of the spatio-temporal positioning of audio-visual objects as well as their behaviour in response to interaction (scene description);
- the coded representation of synthetic two-dimensional (2D) or three-dimensional (3D) objects that can be manifested audibly and/or visually;
- the Extensible MPEG-4 Textual (XMT) format - a textual representation of the multimedia content described in MPEG-4 using the Extensible Markup Language (XML);
- and a system level description of an application engine (format, delivery, lifecycle, and behaviour of downloadable Java byte code applications). (The MPEG-J Graphics Framework eXtensions (GFX) is defined in MPEG-4 Part 21 - ISO/IEC 14496-21.)

Binary Format for Scenes (BIFS) is a binary format for two- or three-dimensional audio-visual content. It is based on VRML and part 11 of the MPEG-4 standard.

BIFS is MPEG-4 scene description protocol to compose MPEG-4 objects, describe interaction with MPEG-4 objects and to animate MPEG-4 objects.

MPEG-4 Binary Format for Scene (BIFS) is used in Digital Multimedia Broadcasting (DMB).

The XMT framework accommodates substantial portions of SMIL, W3C Scalable Vector Graphics (SVG) and X3D (the new name of VRML). Such a representation can be directly played back by a SMIL or VRML player, but can also be binarised to become a native MPEG-4 representation that can be played by an MPEG-4 player. Another bridge has been created with BiM (Binary MPEG format for XML).”[8],[9]

2.5 Synchronized Multimedia Integration Language (SMIL)

“Synchronized Multimedia Integration Language (SMIL) is a World Wide Web Consortium recommended Extensible Markup Language (XML) markup language to describe multimedia presentations. It defines markup for timing, layout, animations, visual transitions, and media embedding, among other things. SMIL allows presenting media items such as text, images, video, audio, links to other SMIL presentations, and files from multiple web servers. SMIL markup is written in XML, and has similarities to HTML.”[10]

2.6 Digital Audio Compression AC-4

The digital audio compression AC-4 [5, 6] is not a file format, but audio content, especially movie content, will be produced in this new ETSI standard, because it is the audio format of Dolby ATMOS[11] [7, 8]. It would be of primarily advantage to make this object-based audio format compatible to the broadcast change.

[8] https://en.wikipedia.org/wiki/MPEG-4_Part_11

[9] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38560

[10] https://en.wikipedia.org/wiki/Synchronized_Multimedia_Integration_Language

[11] http://www.dolby.com/us/en/brands/dolby-atmos.html

3 An Investigation into Requirements and possible File Formats for the Representation and Archiving of Object-based Content

3.1 Requirements for Representation

After several discussions among the partners the result was that the format should be able to fulfil the requirement as an exchange format for spatial audio scenes that are listed below:

3.1.1 Metadata support

One main requirement for the representation of object-based audio is the capability of metadata transport: As stated before, the transport of metadata is crucial for representation, archiving and provision of object-based audio. Any considered format should therefore be able to contain the needed metadata, e.g. defined in the Audio Definition Model as described earlier.

The audio scene should be described in a way that headphone or loudspeaker signals can be generated for an arbitrary reproduction system, such that the reproduced scene sounds as close to the initial audio scene as the chosen reproduction system allows. The reproduction system itself should not be specified in the scene description.

3.1.2 Use of existing standards

The use of open standards is preferred. A standardized format provides an established format with a full definition, which has been tested extensively during standardization. A standardized format also has the advantage of being rather stable.

3.1.3 File size requirements / Compression

Storage space / file size is considered as important for storage of object-based audio, as the number of objects can be very high.

Object-based audio scenes may contain audio objects for several personalized mixes. Several language versions can be provided to the user to choose of, as well as e.g. different stadium atmospheres of a sports recording. The number of contained objects may therefore be higher than the number of played back objects. Therefore compressed audio formats, such as MPEG-H 3D Audio, should be considered for the representation of object-based audio content.

3.1.4 Support of advanced audio

The format needs to be capable of transporting a high number of audio tracks to enable the separate storage of sound events as objects. Large file sizes should be supported, as the audio data are also accompanied by metadata that may rapidly change over time.

For legacy reasons, the format should also be capable of transporting channel-based audio. It would also be beneficial if also scene-based content can be carried for flexibility reasons.

3.2 Requirements for Archiving

After several discussions among the partners the requirements for archiving are the following.

3.2.1 Format-related requirements

When choosing a file format for archiving of object-based audio content, the following requirements should be satisfied:

- The format should be publicly and openly documented.
- The format should be non-proprietary and the format should not depend on proprietary equipment.
- The format should be widely used.
- The format should be self-documenting.
- The format should be accessed using readily-available tools (opened, read).
- The format should not depend on a specific operating system or type of equipment.
- The format should be as stable as possible, future versions need to be backwards compatible.

3.2.2 File-related requirements

Beside the above-mentioned general requirements, the following requirements on single archive files should be considered:

- An archive file should be self-documenting and self-containing
- An archive file should include technical metadata (e.g. EBUCore metadata), such as
    - settings of and information on production/recording equipment
    - technical adjustments applied to the audio
    - ownership
    - description of any signal processing
    - integrity of data (on content level, record access etc.)

3.2.3 Audio-related and object-based requirements

Specific requirements for the archiving of audio content can be formulated. It has to be ensured that the archive contains a high quality version of the content (archiving quality). Therefore, the following requirements are defined:

- The format should be an uncompressed format.
- The format should support a bit-depth of at least 24bit.
- The format should support sampling rates up to at least 48kHz, better up to 96kHz

The means of the archive being specifically used for object-based audio also implies a requirement with respect to the combination of audio essence and object-based metadata:

- Audio essence and object-based metadata should be part of a larger preservation file/package

3.2.4 System-related requirements

System-related requirements may be out of scope of ORPHEUS, however, they should for completeness at least be briefly considered for the choice of a format for archiving.

An archiving system should…

- be able to protect confidential media assets
- provide seamless access when needed
- provide a mechanism for version control of files (e.g. combining multiple media files in a packet with version control)
- provide a mechanism to check data integrity on a file level (e.g. check-sum calculation)
- provide the means for unique persistent identifiers for the archival master files

3.3 Possible Formats for Representation and Archiving

3.3.1 Broadcast Wave (BWF) and Broadcast Wave 64Bit (BW64)

The Broadcast Wave Format (BWF) is the standard audio data file format used today in broadcasting. For the EBU – TECH 3285 specification see [9]:

> "The Broadcast Wave Format (BWF) is a file format for audio data. It can be used for the seamless exchange of audio material between different broadcast environments and between equipment based on different computer platforms. As well as the audio data, a BWF file contains the minimum information – or metadata – which is considered necessary for all broadcast applications. The Broadcast Wave Format is based on the Microsoft WAVE audio file format, to which the EBU has added a 'Broadcast Audio Extension' chunk."

A more recent (10/2015) specification by the ITU in Recommendation ITU-R BS.2088 "Long-form file format for the international exchange of audio programme materials with metadata" [3] provides a new RIFF/WAVE format that allows the carriage of metadata. It is a successor of the "Broadcast Wave File" (BWF), extending the functionality to be able to contain more data in a more flexible way.

This so-called BW64 (Broadcast Wave 64Bit) format allows for maximum file sizes larger than 4GB by using 64Bit signalling and is therefore also capable of transporting immersive audio. Two specific chunks are defined: The <chna> chunk (Channel Allocation Chunk) provides the references from each track in a BW64 file to the identifiers in the ADM metadata.

The <chna> chunk is specifically defined for the use with ADM as defined in Recommendation ITU-R BS.2076. It consists of a header followed by the number of tracks and track identifiers. This is followed by an array of ID structures that each contains IDs corresponding to ADM element IDs.

The ADM metadata itself is stored in the <axml> chunk, which is defined for storing and transferring metadata as XML. The combined usage of ADM and BW64 enables file-based workflows where a mixed library of legacy WAVE-file based content and immersive content will exist and for the usage in WAVE-file based environments, where WAVE-file based broadcast applications wish to upgrade to handle immersive content, while maintaining forward compatibility.

For the use within ORPHEUS, especially the BW64 format in combination with ADM metadata is considered as relevant, because they represent the latest status regarding the given requirements. Partners like BBC, IRT and FHG were heavily involved in the standardization process in ITU-R.

3.3.2 MXF

The MXF (Material Exchange Format) is defined in SMPTE ST 385 [10], developed by the Society of Motion Picture & Television Engineers (SMPTE). MXF is a container format for professional digital video and audio media.

MXF supports a number of different streams of coded "essence", encoded in any of a variety of video and audio compression formats. It can hold all relevant media formats including metadata (different from object-based metadata; describes the material contained within the MXF file) and time code.

Different so-called "Generic Containers" are defined, e.g. also for storage of Broadcast Wave (BWF) in MXF (SMPTE 382M: GC-AESBWF).

In general, MXF is also compatible with BW64/ADM and may therefore also be considered as relevant within the scope of ORPHEUS.

An overview about documents about MXF published by SMPTE is given by IRT[12].

3.3.3 MPEG-H

MPEG-H 3D Audio, specified as ISO/IEC 23008-3 (MPEG-H Part 3) [11], is an audio coding standard developed by the ISO/IEC Moving Picture Experts Group (MPEG) to support coding audio as audio channels, audio objects, or Higher Order Ambisonics (HOA).

MPEG-H 3D Audio has been designed to meet requirements for delivery of next generation audio content to the user, ranging from highest-quality cable and satellite TV down to streaming to mobile devices.

The main features that make MPEG-H 3D Audio applicable for delivery of next generation audio ranging from highest-quality cable and satellite TV down to streaming to mobile devices are its flexibility with regard to input formats and its flexibility with regard to reproduction formats.

MPEG-H 3D Audio supports channel-based audio, object-based audio and Higher Order Ambisonics (HOA). The MPEG-H 3D Audio codec allows for any combination of channel, object and HOA audio content within one MPEG-H audio bitstream. Thus, the most appropriate representations of different elements of a sound scene can be chosen.

In contrast to audio production and monitoring, where the setup of loudspeakers is well defined, the setup of loudspeakers in consumers' homes often includes non-ideal placement and differs regarding the number of speakers. Within MPEG-H 3D Audio, a format converter adapts the content format to the actual real-world speaker setup available on the playback side to provide flexible rendering to different speaker layouts.

The rendering of audio objects on arbitrary trajectories is supported by an object renderer based on Vector Base Amplitude Panning (VBAP) and providing an automatic triangulation algorithm for arbitrary target configurations. The triangulation makes use of imaginary loudspeakers to provide complete 3D triangle meshes for any setup to the VBAP algorithm.

As media consumption is moving further towards mobile devices as smartphones or tables with headphones, a binaural rendering module was included in the MPEG-H 3D audio decoder. This module aims to convey the spatial impression of immersive audio productions on headphones.

The definition of audio metadata in MPEG-H 3D audio allows for personalized playback options, such as increasing or decreasing the level of dialog relative to the other audio content. With the metadata definition, MPEG-H 3D Audio also supports several use-cases for audio interactivity and object-based audio, such as changing the position of sound events, changing the language of a program, enabling of additional dialog tracks, choosing between content versions and automatic screen-related audio scene scaling. An overview of MPEG-H audio metadata is provided in [12].

[12] http://ftp.irt.de/IRT/mxf/information/specification/index.php

4 An Analysis into Requirements and Possible Formats for the Provision of Object-based Content via Streaming

ADM based streaming, and JSON, MDA, ETSI TS 103 223, synchronization

4.1 Possible Formats for Streaming

Exchanging audio material using complete files is convenient for non-real-time applications, but in broadcasting audio often needs to be streamed in real-time. Streaming is required in playout systems and for live capture and delivery. It involves either slicing an existing audio file into frames, or generating frames on-the-fly and delivering these frames in real-time over the delivery medium (such as an IP network).

The design of a streaming format depends upon the delivery medium. For emission stages of the broadcast chain bit-rate restrictions mean that lossy coding is required for audio.

In distribution and contribution stages, bit-rates are usually far less restrictive so either higher bit-rate codecs can be used, or PCM audio can be delivered.

With the move to media distribution over IP, bit-rates are potentially high enough to cope with uncompressed audio and associated metadata. Streaming over IP requires the data to be carried in packets, so any streaming format requires the audio is segmented into frames. With the move towards next generation audio (i.e. object, channel and scene-based), metadata is required to be delivered alongside the audio signal. Generating an audio-plus-metadata stream requires segmentation of the audio and metadata into frames of a size that is suitable for the delivery medium.

The following formats are considered by the partners as relevant for streaming applications in the scope of the ORPHEUS project.

4.1.1 DASH

“Dynamic Adaptive Streaming over HTTP (DASH), also known as MPEG-DASH, is an adaptive bitrate streaming technique that enables high quality streaming of media content over the Internet delivered from conventional HTTP web servers. Similar to Apple's HTTP Live Streaming (HLS) solution, MPEG-DASH works by breaking the content into a sequence of small HTTP-based file segments, each segment containing a short interval of playback time of content that is potentially many hours in duration, such as a movie or the live broadcast of a sports event. The content is made available at a variety of different bit rates, i.e., alternative segments encoded at different bit rates covering aligned short intervals of play back time are made available. While the content is being played back by an MPEG-DASH client, the client automatically selects from the alternatives the next segment to download and play back based on current network conditions. The client selects the segment with the highest bit rate possible that can be downloaded in time for play back without causing stalls or re-buffering events in the playback. Thus, an MPEG-DASH client can seamlessly adapt to changing network conditions, and provide high quality play back with fewer stalls or re-buffering events.

MPEG-DASH is the first adaptive bit-rate HTTP-based streaming solution that is an international standard. MPEG-DASH should not be confused with a transport protocol — the transport protocol that MPEG-DASH uses is TCP.

MPEG-DASH uses existing HTTP web server infrastructure that is used for delivery of essentially all World Wide Web content. It allows devices like Internet-connected televisions, TV set-top boxes, desktop computers, smartphones, tablets, etc. to consume multimedia content (video, TV, radio, etc.) delivered via the Internet, coping with variable Internet receiving conditions. Standardizing an adaptive streaming solution is meant to provide confidence to the market that the solution can be adopted for universal deployment, compared to similar but more proprietary solutions like Smooth

Streaming by Microsoft, or HDS by Adobe.

The technology is codec-agnostic which means it can use content encoded with any codec like H.265, H.264, VP9 etc."[13]

4.1.2 Serialized ADM

A serialized version of the Audio Definition Model (ADM) is currently under development in the ITU. While the audio file format as described in Recommendation ITU-R BS.2088 combined with the Audio Definition Model described in Recommendation ITU-R BS.2076 provide the ability to exchange object, channel and scene-based audio files, they are not readily suitable for streaming, particularly of live productions. Therefore a serialized form is required.

With a serialized representation of the Audio Definition Model with segmentation of audio and metadata, it will be possible to use the ADM for streaming of immersive audio content. For the serialized ADM, JSON (JavaScript Object Notation) will probably be used as a representation format. The main difference from the ADM XML representation is that no referencing between elements is used; instead, it is a complete hierarchical structure. This ensures each frame is self-contained and JSON is more compact and also easier to parse.

JSON is an open-standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs.

It is the most common data format used for asynchronous browser/server communication: asynchronous JavaScript and JSON (AJAJ)[14]. JSON is derived from JavaScript, but code to generate and parse JSON data is available in many programming languages. For more detail see also 'json.org'.

Basically, any XML-based metadata scheme can be represented in JSON as well.

Depending on the timeline of the ITU work, the envisioned serialized ADM representation may be considered for the use within ORPHEUS.

4.1.3 Streaming of MPEG-H

Because MPEG-H is primarily a compression format with the addition of 3D-rendering capability, it can be used like any other compression format together with codec-agnostic streaming formats such as DASH and HLS. As such, MPEG-H is mostly relevant for the distribution to end devices ("emission") and is already defined as an optional audio codec in version 3.2 of the DASH-IF[15] Interoperability guidelines.

All required metadata for decoding and rendering MPEG-H bit-streams is defined in the MPEG-H specification and can be embedded in the MPEG-H Audio Stream (MHAS) in binary form. Therefore, the definition and encoding of all relevant metadata as well as its decoding and rendering behaviour is well defined when using MPEG-H as a codec format for streaming. However, the mapping between the metadata in production (e.g. ADM) and MPEG-H has to be assured.

In addition, MPEG-H offers the following features that are relevant for streaming: First, it supports seamless bit-rate switching through Instantaneous Playout Frames (IPF) which simplifies tune-in and adaptive streaming in DASH or HLS. MPEG-H also supports configuration changes and splicing on the bit-stream level, which can be important for use cases such as insertion of advertisements ("ad-insertion"). In addition, objects can be transmitted as independent streams and merged at the client,

[13] https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP

[14] https://en.wikipedia.org/wiki/AJAJ

[15] DASH Industry Forum

e.g. for hybrid delivery of an alternate language. Finally, the flexible rendering and format conversion of MPEG-H allows decoding the same bit-stream for e.g. a 5.1 surround speaker setup, stereo speaker setup, or binaural headphones. Hence, a single bit-stream is stored on the server covering multiple playback scenarios and therefore saving storage space and signalling complexity.

When initializing the streaming session it has to be considered that the overall audio scene may contain multiple channel beds and objects which can only be combined in certain ways. For example, an ambience atmosphere with 5.1 channels has to be combined with the dialog in either German or English language, and in addition an optional sound effect can be added. Though all those dependencies are well described in the MPEG-H Audio Scene Information (ASI) data structure, some of this information is also needed on the session initialization level, i.e. in the Media Presentation Description (MPD) of DASH. This allows, for example, that a DASH-client only fetches the English dialog track and therefore saves transmission bit rate. The mapping of audio scene information and dependencies onto the MPD is being defined in MPEG and DASH-IF.

4.1.4 Streaming of MXF

The Material Exchange Format (MXF), which is described in a previous section of this document in some detail, is widely used for production and contribution in the broadcasting community. In its common form, MXF is suitable for file-based-only workflows, however it can be used for streaming applications as well with the High Bitrate Media Transport Protocol (HBRMT), which is standardized in SMPTE 2022-6, as described in a paper by the IRT in [13]

4.1.5 HTTP Live Streaming (HLS)

“HTTP Live Streaming is an HTTP-based media streaming communications protocol implemented by Apple Inc. as part of its QuickTime, Safari, OS X, and iOS software. It is similar to MPEG-DASH in that it works by breaking the overall stream into a sequence of small HTTP-based file downloads, each download loading one short chunk of an overall potentially unbounded transport stream. As the stream is played, the client may select from a number of different alternate streams containing the same material encoded at a variety of data rates, allowing the streaming session to adapt to the available data rate. At the start of the streaming session, it downloads an extended M3U playlist containing the metadata for the various sub-streams which are available.”[16], [17]

4.2 Requirements for Streaming

4.2.1 Streaming of Object-based Content

Considering the streaming of object-based content, a distinction should be made between the streaming of audio signals and the streaming of object-based metadata related to these audio signals. This is necessary because those two types of content, even though they might be transmitted within a single (network) stream, they have different requirements associated with them.

In addition, it is useful to further distinguish between the format of the payload[18] data transmitted and the actual transport protocol used to transmit said payload.

[16] https://en.wikipedia.org/wiki/HTTP_Live_Streaming

[17] https://developer.apple.com/streaming/

[18] Payload is the part of transmitted data that carries that actual information or message, excluding any overhead data solely required for payload delivery as part of the transmission protocol

There are a variety of usage scenarios for streaming object-based content in production, ranging from studios with dedicated and isolated networks to wide-area networks.

While the main target applications are related to controlled studio environments, proposed solutions should be applicable to other usage scenarios as well. Additionally, it is understood that while the focus is on IP-based operation/streaming, data formats that are not limited to IP-based operation per se are preferred. Therefore, consideration should be given to the different layers in the OSI model when defining data formats and the carriers that might be used to transport that data, such that compatibility is maintained where necessary to support different physical layers or transport layers.

4.2.2 General Requirements

4.2.2.1 Synchronization

It is required that synchronisation of audio signals and object-based metadata related to those audio signals can be maintained, to a precision of one audio sample period or better.

4.2.2.2 Use of Existing Standards

The use of open and/or free (zero-cost) standards is preferred. Furthermore, the use of existing standards is preferred over creating new ones as long as they are available under fair, reasonable and non-discriminatory terms.

4.2.2.3 Unicast and Multicast Transmission

The format of the payload data for both audio signals and metadata related to that audio signals must be structured in such a way to support both unicast (point-to-point) and multicast (point-to-multipoint) transmission.

4.2.3 Requirements for Streaming Audio Signals

4.2.3.1 Latency

In a production environment, streaming formats must operate with a constant, predictable latency. Furthermore, low-latency may be required for some applications to support efficient monitoring and editing of object-based audio streams.

4.2.3.2 Bitrate Reduction

Streaming of uncompressed (PCM) audio signals at typical production sample rates and bit depths must be supported. It can be assumed that in a studio or studio centre the available network capacity is such that a bit rate reduction is not required and therefore preservation of audio quality is the priority.

Nevertheless, streaming of compressed audio signals can be supported in addition, as it extends the applicability of the streaming format to usage scenarios where network bandwidth might be limited. This is particularly the case for streaming to the end user (“emission”), see also Quality of Service (QoS) below.

4.2.3.3 Synchronization of Audio Signals

The streaming format and/or protocol must provide means to synchronize all audio signals, preferably in relation to a common master clock.

There are two use cases that must be supported:

1. All audio signals are contained within one stream.

2. Audio signals are split across multiple streams.

The latter case is especially required when audio signals origin from different sources must be merged or switched.

4.2.3.4 Quality of Service (QoS)

Especially in application scenarios where network resources might be shared with other applications, it is required that different priorities can be assigned to data packets in order to guarantee a certain level of performance and to avoid the loss of packets carrying audio signals.

For distributing object-based audio to the end user over the open Internet ("emission"), the only viable QoS-level is often "Best Effort". In this situation adaptive streaming protocols such as DASH and HLS can significantly reduce the frequency of playback interruptions and are therefore required. Especially for reliable playback of audio streams in mobile environments, fall-back bit rates below 256 kbps shall be supported.

4.2.4 Requirements for Streaming Object-based Audio Metadata

4.2.4.1 Independence of Transport Mechanism Used to Stream Audio Signals

In order to support a variety of audio signal streaming solutions, preserve flexibility and compatibility with future protocols, the format describing the object-based metadata must not introduce any direct dependency on the protocol used to stream audio signals.

This is especially required in order to incorporate streaming of object-based content into existing production environments that already feature IP-based streaming of audio signals.

4.2.4.2 Embedded and Unenclosed Transmission

It is required that the object-based metadata can be embedded in /multiplexed with the audio signals to form a single stream.

Furthermore, it is recommended that object-based metadata and audio signals can be transmitted in separate streams.

The single stream variant has the advantage of being simpler in terms of operation and monitoring, while the latter enables combination and integration of object-based audio streaming with devices that support streaming of audio signals only.

4.2.4.3 Metadata Set

It is required that ITU-R BS.2076 (ADM) metadata set can be carried.

Furthermore, it is required that production metadata (as in the EBU Core, for example) can be carried.

For the streaming to end users the ADM metadata may be converted into formats that are aligned to the requirements of the emission codec, e.g. MPEG-H. Furthermore, some of the production metadata shall be removed for reasons of confidentiality and bit rate efficiency.

4.2.4.4 Monitoring and Inspection

The streaming format must support easy monitoring and inspection of the metadata. This is essential for fast error detection and failure recovery.

4.2.4.5 Delta Frame Transmission

In general, the full object-based metadata must be available for a device to interpret, modify or render the object-based audio scene. But continuous transmission of the complete metadata set can unnecessarily increase the required bandwidth and metadata parsing complexity.

It is therefore required that delta metadata frames can be transmitted in-between full metadata frames, carrying only changed metadata entities. This concept is similar to “intra-frames” and “predicted” frames as used in video encoding applications.

This introduces an additional “tune-in” time, which a device must wait until a first full frame is received. In order to suit different applications with different requirements in terms of bandwidth, complexity and tune-in time, it is required that the full frame repetition rate is not limited. This includes the case that no delta frames are transmitted at all.

4.2.4.6 Efficient Filtering of Metadata and Stream Splicing

Metadata streams within a production environment might include sensitive metadata for internal use only. Furthermore, object-based metadata streams from multiple sources might be used to form a single production.

Therefore, it is required that object-based metadata is structured in a way that can be modified efficiently, with a special focus on the removal of sensitive production metadata and merging of multiple metadata streams.

For streaming to the end users (“emission”) it shall be possible to access individual elements of an object-based audio scene based on the requirements of the client. In other words, the client shall be able to pick relevant objects and shall not be enforced to download a single aggregated audio stream with all objects (including those he will never decode/render – e.g. an alternate language track). For the example of DASH, this means that some metadata has to be represented in the Media Presentation Description (MPD) which is used for session initialization.

5 Interoperability

As stated above, some of the requirements for storage/representation, archiving and provision are identical or very similar while others are application-dependent (e.g. usage of data rate reduction (compressed audio) vs. uncompressed audio).

However, all the formats have to contain/carry different representations of the same object-based audio content.

Therefore interoperability of the formats is an important factor when choosing formats for storage/representation, archiving and provision (file-based provision and provision via streaming) of object-based audio.

“Interoperability” in the context of the ORPHEUS broadcasting chain can be understood as the following high-level aspects:

- The formats should provide coherent services for users, even when individual components of the formats are technically different.
- The different modules/formats should be able to intercommunicate and share or exchange data via a defined set of exchange interfaces.
- The different modules of the broadcasting chain should work together in implementation, even if using different formats.

In more detail, two main parts of object-based audio need to be considered in the context of interoperability: The transport of audio data and the transport of metadata (object-based/audio-related metadata as well as production/technical metadata). It should be possible to exchange the needed information (audio data plus metadata) adequately between the formats and/or their users.

As is addressed several times in the document, the support of the (audio-related) metadata set of the Audio Definition Model (ADM) is desirable. Therefore any format used in the broadcasting chain should either directly support the carriage of ADM metadata (e.g. using BW64 files) or provide a metadata model, which supports comparable concepts, i.e. having a different syntax, but providing descriptors for the same object-based characteristics, even if named differently. In that case, a translation of ADM metadata to the other model (and vice versa) would be possible.

With respect to the audio content, it needs to be ensured that the chosen formats can all carry the needed information in terms of a defined target quality. If there are differences between the formats with e.g. respect of the usage of compressed vs. uncompressed audio, they should all be able to provide the defined minimum target quality if translation/transcoding is needed (e.g. by means of scalable compression, support of defined sampling rates).

Besides, it needs to be considered that some formats may only carry a subset of the complete audio data. The archiving file can be seen as a kind of master format, containing the overall superset of audio data, however in use-cases where the data rate is limited, it might be reasonable to provide only a subset of the audio data (e.g. reduce the choice of languages, provide a smaller set of objects or a smaller set of presets/mixes for the user to choose from). If this is needed, mechanisms to choose a reasonable subset of audio data are also an aspect to consider in the context of interoperability.

As one of the main interoperability requirements is the way how modules in a system can exchange data and information, the interoperability is also an aspect of the entire system architecture. In ORPHEUS, the choice of the needed formats for storage/representation, archiving and provision of object-based content cannot be made without keeping in mind the complete processing chain and how the different formats need to interact. Besides the formats themselves, also interfaces and (if needed) translation rules have to be considered in the system definition process as well.

Achieving the necessary degree of interoperability is in general a compromise between a pragmatic system definition (i.e. striving for a minimum variability between components/formats, which is

beneficial in terms of implementation effort and simplicity of the overall system) and the choice of highly performant single components (i.e. choosing formats, which are very specific to their use-case, but satisfy their defined requirements in the best way possible, even if this makes the translations/interfaces more difficult).

The following paragraphs describe two formats suitable for transmission of audio over IP.

5.1 Audio Video Bridging (AVB)

“An Audio Video Bridging (AVB) network implements a set of protocols being developed by the IEEE 802.1 Audio/Video Bridging Task Group. AVB works by reserving a fraction of the available Ethernet bandwidth for AVB traffic. There are four primary differences between the proposed architecture and existing 802 architectures:

- Precise synchronization,
- Traffic shaping for media streams,
- admission controls, and
- Identification of non-participating devices.

These are implemented using relatively small extensions to standard layer-2 MACs and bridges. This "minimal change" philosophy allows non-AVB and AVB devices to communicate using standard 802 frames. However only AVB devices are able to: a) reserve a portion of network resources through the use of admission control and traffic shaping and b) send and receive the new timing-based frames. AVB packets are sent regularly in the allocated slots. As the bandwidth is reserved, there will be no collisions.”[19]

5.2 AES67 and Ravenna

“AES67 is a standard for audio-over-IP interoperability. The standard was developed by the Audio Engineering Society and published in September 2013. It is a layer 3 protocol suite based on existing standards and is designed to allow interoperability between various IP-based audio networking systems such as RAVENNA, Livewire, Q-LAN and Dante. It also identifies commonalities with Audio Video Bridging (AVB) and documents AVB interoperability scenarios."[20]

“High-performance media networks support professional quality audio (16 bit, 44,1 kHz and higher) with low latencies (less than 10 milliseconds) compatible with live sound reinforcement. The level of network performance required to meet these requirements is available on local-area networks and is achievable on enterprise-scale networks. A number of networked audio systems have been developed to support high-performance media networking but until now there were no recommendations for operating these systems in an interoperable manner. This standard provides comprehensive interoperability recommendations in the areas of synchronization, media clock identification, network transport, encoding and streaming, session description and connection management” [14].

“RAVENNA is an open technology for real-time distribution of audio and other media content in IP-based network environments. Utilizing standardized network protocols and technologies, RAVENNA can integrate and operate on existing network infrastructures. Performance and capacity are scaling

[19] https://en.wikipedia.org/wiki/Audio_Video_Bridging

[20] https://en.wikipedia.org/wiki/AES67

with the capabilities of the underlying network architecture. Emphasize is put on data transparency, tight synchronization, low latency and reliability. It aims at applications in professional environments, where networks are planned and managed, and where performance has to surpass the requirements of consumer applications.

As an open technology, the functional principles are publically available and RAVENNA technology can be freely implemented and used without any proprietary licensing policy. Numerous industry partners are already supporting the RAVENNA technology.

In September 2013, the AES has published the AES67 "Standard on High-performance Streaming Audio-over-IP Interoperability", which defines guidelines and mechanisms to achieve interoperability between different IP-based real-time streaming solutions. Since RAVENNA's fundamental operational principles, protocols and formats are in-line with what has been defined in AES67, RAVENNA is already compatible with AES67" [15].[21]

[21] http://www.ravenna-network.com/about-ravenna/resources/

6 Backward and Forward Compatibility with Legacy Systems

6.1 How to Handle New Content on Legacy Emission System

The main focus of broadcasters is traditionally on the simultaneous emission of content for a large audience via a terrestrial, satellite or cable radio medium. The reception devices (TVs, Set-top boxes) for those transmission ways are very well established and can be found in almost every household all over Europe. These devices are and mostly will be not capable to deal with object-based content as the built-in decoders can be hardly updated to new standards such as the coming DVB UHD Phase 2[22] which will enable object-based delivery via DVB.

Even though the focus of ORPHEUS is on IP delivery where new inventions and technologies such as object-based audio can be easily and quickly introduced due to software based systems, the backward compatibility with existing end user devices needs to be ensured. Therefore, ORPHEUS has to make sure that both the reference architecture as well as the pilot implementation architecture will be able to transcode and transform the object-based content for legacy playout and emission infrastructures. Figure 1 illustrates the envisioned solution for legacy systems. The object-based content will be rendered to Stereo and / or 5.1 surround signals in the distribution block and can be afterwards emitted via legacy broadcasting mediums.

6.2 How to Handle Old Content on New Production Systems

As the majority of archived and pre-produced content is not object-based but channel-based, the ORPHEUS partner need to make sure that he reference architecture and the pilot implementation architecture will be able to handle legacy content during the production. As per Figure 1, the planned workflow has two steps where legacy channel-based content (Mono, Stereo, 4.0, 5.1) could be used for the further production. Investigations are to be conducted in order to find out how the architecture and the implemented system can and should handle such channel-based content.

[22] http://www.digitaltveurope.net/462862/dvb-embraces-hdr-with-next-phase-ultra-hd-tv-requirements/

7 Conclusions

The above listed information and requirements are the basis for the selection and new definitions of file and streaming formats, selected for the reference architecture and the pilot implementations.

References

[1] ITU-R, BS.2076-0, Audio Definition Model. 2015, Intern. Telecom Union, Geneva, Switzerland.

[2] EBU, Tech 3364, Audio Definition Model - Metadata Specification. 2014, European Broadcasting Union, Geneva, Switzerland.

[3] ITU-R, BS.2088-0, Long-form File Format for the International Exchange of Audio Programme Materials with Metadata. 2015, Intern. Telecom Union, Geneva, Switzerland.

[4] ITU-R, BS.2388-0, Usage Guidelines for the Audio Definition Model and Multichannel Audio Files. 2015, Intern. Telecom Union, Geneva, Switzerland.

[5] ETSI, TS 103 190, Digital Audio Compression (AC-4) Standard. 2014, EBU, Sophia Antipolis Cedex - FRANCE. http://www.etsi.org/news-events/news/783-2014-04-etsi-releases-ac-4-the-new-generation-audio-codec-standard.

[6] ETSI, TS 103 190-2, Digital Audio Compression (AC-4) Standard, Part 2: Immersive and personalized audio. 2015, EBU, Sophia Antipolis Cedex - FRANCE. http://www.etsi.org/news-events/news/783-2014-04-etsi-releases-ac-4-the-new-generation-audio-codec-standard.

[7] Dolby, Dolby Atmos - Cinema Technical Guidelines (White Paper). 2013. http://www.dolby.com/uploadedFiles/Assets/US/Doc/Professional/Dolby-Atmos-Cinema-Technical-Guidelines.pdf.

[8] Dolby, Dolby Atmos - Nex Generation Audio for Cinema (White Paper). 2013. http://www.dolby.com/uploadedFiles/Assets/US/Doc/Professional/Dolby-Atmos-Next-Generation-Audio-for-Cinema.pdf.

[9] EBU, Tech 3285, Specification of the Broadcast Wave Format (BWF) - A Format for Audio Data Files in Broadcasting. 2011, European Broadcasting Union, Geneva, Switzerland.

[10] SMPTE, ST 385, Material Exchange Format (MXF) — Mapping SDTI-CP Essence and Metadata into the MXF Generic Container. 2012, The Society of Motion Picture and Television Engineers, New York, USA.

[11] ISO/IEC, CD23008-3, Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 3: 3D audio. 2014.

[12] Füg, S., et al., Design, Coding and Processing of Metadata for Object-Based Interactive Audio, in 137th AES Convention. 2014: Los Angeles, USA.

[13] Laabs, M. and C. Nufer. MXF Streaming over IP Networks. 14th ITG Conference on Electronic Media Technology (CEMT). 2011. Dortmund, Germany.

[14] AES67, AES Standard for Audio Applications of Networks - High-performance Streaming Audio-over-IP Interoperability. 2015, Audio Engineering Society, Inc., New York, USA.

[15] Ravenna, White Paper V1.0. 2014. www.ravenna-network.com.

[end of document]