Audiovisual Playout-Engine for Interactive Playout

Deliverable D4.7

ICoSOLE identifier:	ICoSOLE-D4.7-BBC-PlayoutEngine_v04.doc
Deliverable number:	D4.7
Main author of Deliverable:	Dave Marston (BBC), Rik Bauwens (VRT), Werner Bailer (JRS)
Internal reviewer:	TaW
Work package / task:	WP4 / T 4.5
Document status:	Final
Confidentiality:	Public

Version	Date	Reason of change
1	2016-09-19	Document created
2	2016-11-03	Added BBC audio content, conclusions & summary
3	2016-11-04	Reviewed by TaW
4	2016-11-10	Final version

The work presented in this document was partially supported by the European Community under the 7th framework programme for R&D.

This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

Table of Contents

List of Figures

Executive Summary

This deliverable describes interactive playout engines for both audio and video. The audio playout engine takes advantage of object-based audio to allow for immersive playback over a variety of playout setups, as well as interaction to allow the user to adjust the sound to their tastes. The playout is presented in a browser using the Web Audio API to process incoming MPEG-DASH audio streams. Browser playout means the user does not need to install any specialist apps, and can use any device.

The video playout engine is an interactive playout client app called “The Wall”, which allows users to view both user-generate content (UGC) and live content, and navigate through the submitted videos. It can display both pre-recorded and live UGC streams, which makes use of MPEG-DASH streaming to receive the videos.

1 Introduction

1.1 Purpose of this Document

This document describes the interactive playout engines for both audio and video playback.

1.2 Scope of this Document

This deliverable covers the interactive audio playout engine and the interactive video playout engine.

1.3 Status of this Document

Final version.

1.4 Related Documents

D4.6 “Audiovisual rendering engine and interface for linear broadcast production”
D5.3 “Broadcast playout engine and audio playout libraries”

2 Interactive Audio Playout

2.1 Object-Based Audio

2.1.1 ADM Metadata for Immersion

One of the features of object-based audio is that it allows for immersive playback. While channel-based and HOA-based audio can also achieve immersion, object-based audio can offer more flexibility for different playout setups.

The Audio Definition Model [ADM] is the basis for the metadata required to define audio objects. The ADM does provide a lot of flexibility and different parameters, but only a small set has been used in the ICoSOLE work. The choice of the parameters is largely driven by the renderer's abilities and the ease of generating the parameters in production.

The parameters for defining object-based immersion appear in the audioBlockFormat element, and the sub-set used in ICoSOLE are:

```
<position coordinate="azimuth"> azimuth(degrees,float) </position>
<position coordinate="elevation"> elevation(degrees,float) </position>
<position coordinate="distance"> distance(degrees,float) </position>
<gain> gain(linear,float) </gain>
```

So the position (in spherical coordinates) and gain of the audio object are used.

2.1.2 ADM Metadata for Interactivity and Personalisation

Object-based audio offers the listener the opportunity to interact with the delivered audio to personalise it to their requirements. This can appear as controls on a webpage to adjust levels of different parts of the sound for example. The ADM is structured and contains parameters to provide different methods of interaction and personalisation. There are flags to allow objects to be interactive or not, and other parameters that restrict the amount of change any interaction can to do the object's gain and position. However, the most useful ADM feature for the sort of personalisation ICoSOLE is using is the ability to group together channels that have some common purpose. For example, if a programme is a panel show with an audience and performers speaking on stage, then it makes sense to group to the channels for the stage sound together and the audience channels together. This would allow the listener to adjust the relative levels of the stage and audience sounds.

The ADM elements that are used for interactivity are audioPackFormat and audioObject. They are both used in grouping together channels and linking them with the audio itself. An example of ADM XML for these two elements is shown below:

```
<audioObject audioObjectID="AO_1001" audioObjectName="Stage" interact="1">
  <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
  <audioTrackUIDRef>ATU_00000002</audioTrackUIDRef>
</audioObject>

<audioPackFormat audioPackFormatID="AP_00031001" audioPackFormatName="Stage" typeDefinition="Objects">
  <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
  <audioChannelFormatIDRef>AC_00031002</audioChannelFormatIDRef>
</audioPackFormat>
```

The audioObject called "Stage" has the 'interact' flag set to 1, and contains two tracks. It refers to the audioPackFormat also called "Stage" which contains two references to audioChannelFormats. So in this example, the "Stage" will be treated as one group of channels, and any interactions on those channels will happen to all those channels in the same way.

So for playout this grouping of channels can be used for generating suitable interactive controls.

2.2 DASH Playout

2.2.1 Summary

This section describes the approach taken to capture, encode and stream multi object audio feeds to client web browsers.

2.2.2 Client Play out Library

Current browsers do not consistently handle channel counts greater than 7.2, which poses a problem for object-based broadcasting. We have experimented with a novel approach to object-based streaming making use of the MPEG-DASH[DASH] standard and Web Audio API[WA].

The goal of this approach was to:

- Allow streaming of multiple audio channels and objects to the browser - regardless of browser channel decode limits.
- Synchronisation across audio streams and with metadata.
- Support for both on demand (pre-recorded) and live media.
- Enable personalisation.
- Enable interaction.
- Support for user selectable rendering (plug-in renderers).

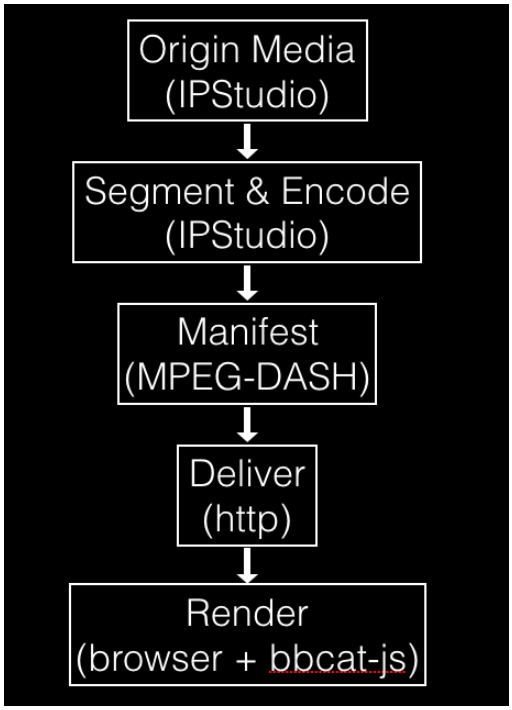

Figure 1: Live workflow

As shown in Figure 1 the workflow can be split into 5 stages:

1. Origin Media is captured using the IPStudio platform.
2. IPStudio is then used to encode the audio and video into the required formats, and segment the data into MPEG-DASH compatible segments.
3. IPStudio makes a manifest (.mpd) file available on an http server which points to the media segments.
4. The DASH segments and manifest are then delivered over http.
5. The bbcat-js toolkit and dash.js[JS] are then used to synchronise and playback the audio and video.

In order to circumnavigate the channel handling limitations in current browsers an approach was taken which splits audio objects in to sub-streams which can be decoded successfully using the Web Audio API method *decodeAudioData()*.

This approach separates the audio object stream into n streams of 5 channels or fewer giving a media structure as shown in Figure 2. Typical segment sizes for the audio range from 1 to 5 seconds long, depending upon required delays, robustness and other frames sizes used in the chain, including the

audio. The audio channels are generally AAC encoded, where 64kbit/s per channel is an acceptable quality bit-rate.

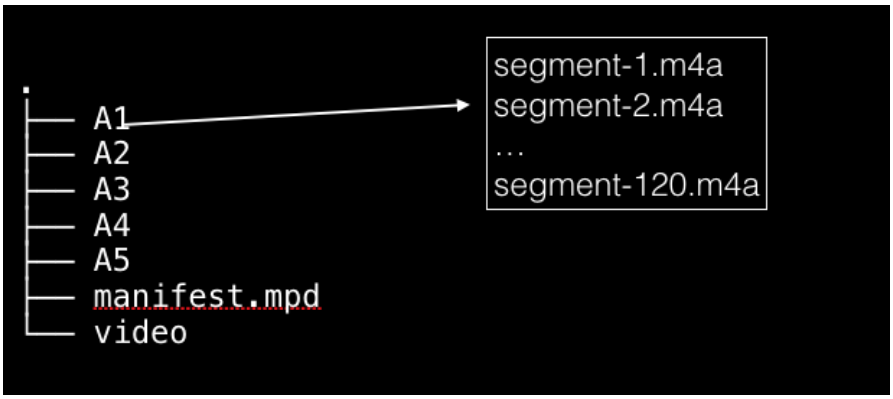

Figure 2: Media folder layout

The BBC DASH player is specifically designed to read multiple audio segment streams simultaneously and schedule the playback of concurrent segments with sample accuracy. Once scheduled samples are routed to a rendering engine (as described in [D4.6]) and finally to the audio destination (speakers) as shown in Figure 3. A range of different renderers is available, as these can either be chosen by the user, or selected for the particular device that is being used for playout (e.g. it would not make sense to offer multi-speaker VBAP rendering for a smartphone).

The interactive object control can allow groups of channels (treated as object) have their gain or positions adjusted by the user. Usually the level of interaction would be given limits to prevent unpleasant mixes being played out, and to simplify the controls for the user. Examples of interaction would be to change the relative levels of stage and crowd sounds, or to be able to zoom in on an area of interest (such as a single musician in a group).

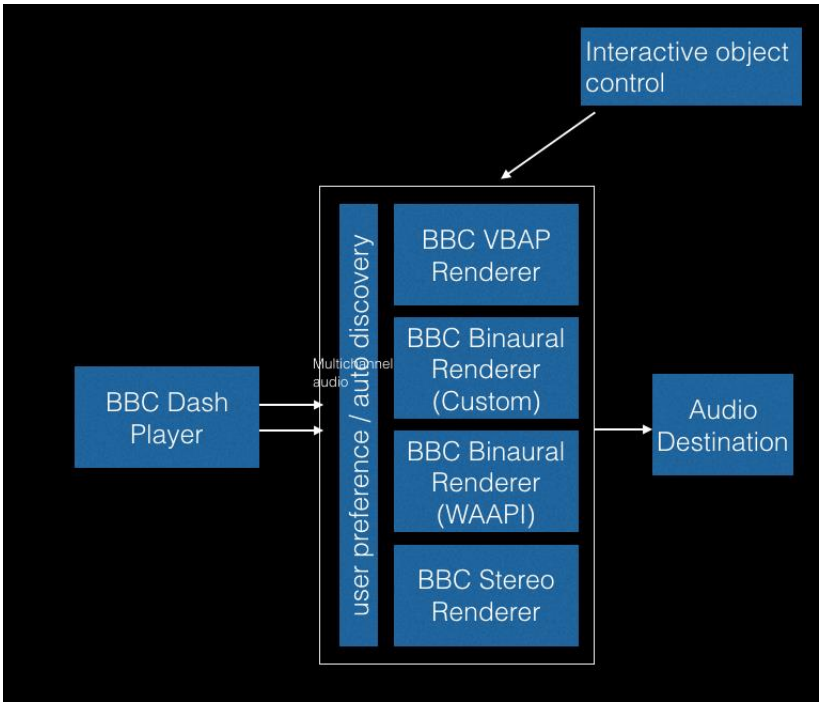

Figure 3: Playout engine structure

Figure 4 shows an example of the playout engine. In addition to metadata which provides information on the spatial position of objects in the scene, an interaction control is also provided to the user allowing them to select the level of audience and stage sound and the preferred rendering method.

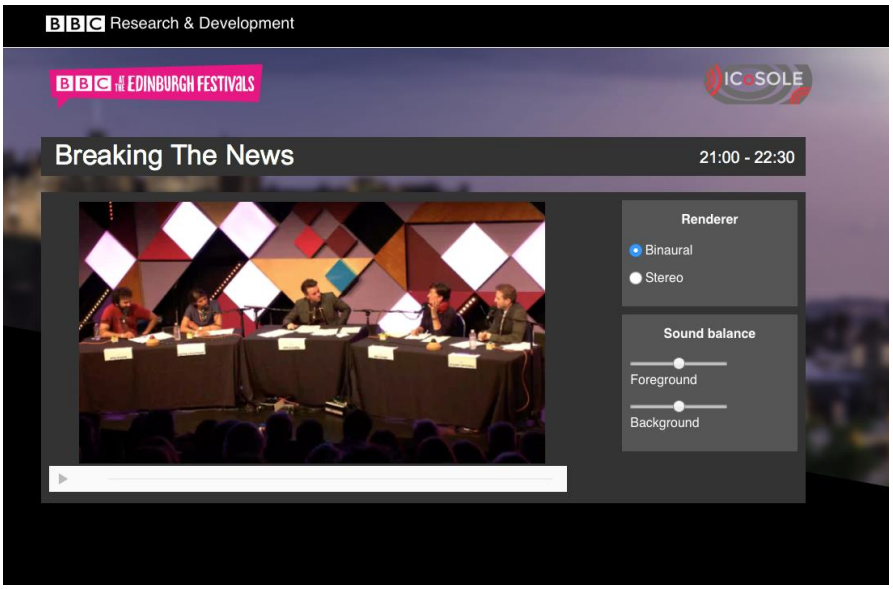

Figure 4: Example playout page

3 Interactive Video Playout

As part of the Wall of Moments/Map of Moments ecosystem, VRT developed “The Wall” as a standalone app for web and mobile. The Wall is an interactive playout client for user-generated content as well as professional video streams.

Figure 5: Screenshot of the Wall on the event site

3.1 User-generated content

3.1.1 Recorded UGC

User-generated content is collected using the Moments and Focus apps. This content can be either a photo or a video, and is tagged with an author, timestamp and location (depending on the origin). Once this content has been reviewed and approved, it can be collected by the Wall, which shows it in a moving carousel (Figure 5). The order in which Moments are displayed depends on different conditions. For the Walls at the event site, the basic principle is to feature recently added Moments prominently, and to intertwine this with some randomly chosen older Moments from time to time. When proximity sensors are attached to The Wall, Moments from people nearby are displayed approximately for a third of the time as well, in order to create a more tailored and personal user experience. People that use The Wall app on a smartphone have a slightly modified experience, as depicted in Figure 6. Moments are ordered by time, the most recent ones first. They have the ability, however, to quickly skip a Moment or a collection of Moments, and rewind to the most recent one (Figure 8).

Figure 6: The Wall on a smartphone

Figure 7: Quick tutorial on how to use the app

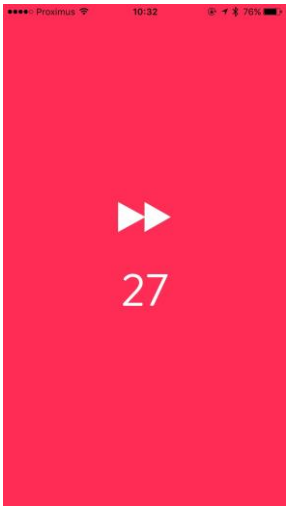

Figure 8: Fast-forward

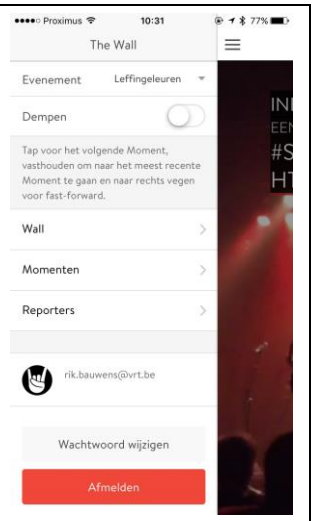

Figure 9: Added menu for accessing extra features

3.1.2 Live UGC

Live UGC is captured with the dedicated capture app developed in the project, which also handles synchronisation. The streams from the mobile phone (video, audio and metadata are streamed separately) are captured on a server in the cloud, where also processing (content and quality analysis) takes place. This server also (re-encodes, if necessary,) multiplexes (video and audio), packages and streams the live content to a Bitdash encoder instance in the cloud using RTMP. In order to avoid the startup time of the encoder instance (which would either require buffering the stream or losing some seconds at the beginning), the capture server ensures that a running instance is available at any time, and immediately assigns an available instance. This instance is configured to write to a cloud storage area, which can be accessed by the Wall/Map of Moments clients. The encoder creates MPEG DASH and HLS representations according to the configured profile. The URLs to access the content are inserted into the live database. The streams are thus immediately available to the clients, and are also shown in Trademark, so that an operator can make them visible or invisible to the viewers.

Once the live stream has ended, the DASH and HLS manifest files are converted from live to on-demand files, so that adaptive streaming is also available later for on-demand access. Note that this does not require touching the data, but is only a metadata operation. In addition, a concatenated MPEG-4 file is created from the highest quality segments, so that the content is also available for progressive download. This enables consumption of the content on the widest possible range of web and mobile clients.

3.2 Live content

For the screen on the event site, besides user generated content, we enabled the Wall to include live streams as well. These can be live UGC streams as well as raw or edited streams from the professional capture pipeline. To achieve this, we integrated the Bitdash player from Bitmovin. In Trademark, and were able to set the URI of the Walls (with live streams enabled). This URI could, for example, point to the live stream of the currently performing band.

The DASH streams for UGC are created as described above, the DASH streams for professional content are either created by sending the mixing engine output to the Bitdash encoder in the cloud, or by using the BBC IP Studio infrastructure, which provides DASH streams for the content it ingests.

3.3 Interactivity

When people use the mobile version of the Wall (iOS/Android), basic interactivity is added. Users can navigate through the (ordered) list of Moments by using swiping gestures (Figure 7). Moreover, a menu is available (Figure 9), where a list of contributed Moments is shown, as well as a list of Moments reporters (Figure 10). A user can search for a specific reporter, and request a list of (approved) Moments for that reporter (Figure 11). From that list, the Moment can be displayed picked and played individually (Figure 12).

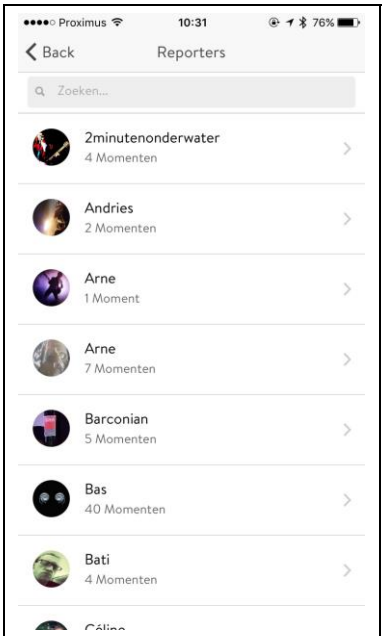

Figure 10: Overview of Moments reporters

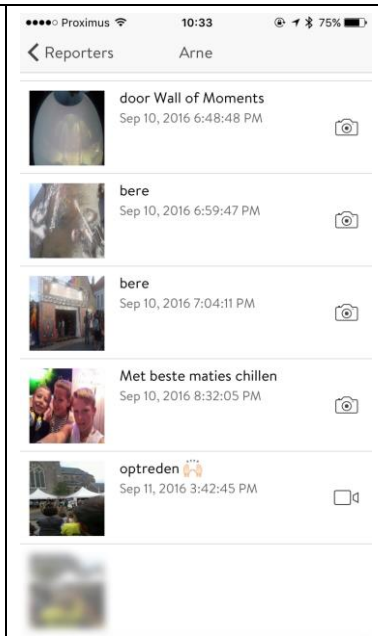

Figure 11: List of posted Moments by reporter

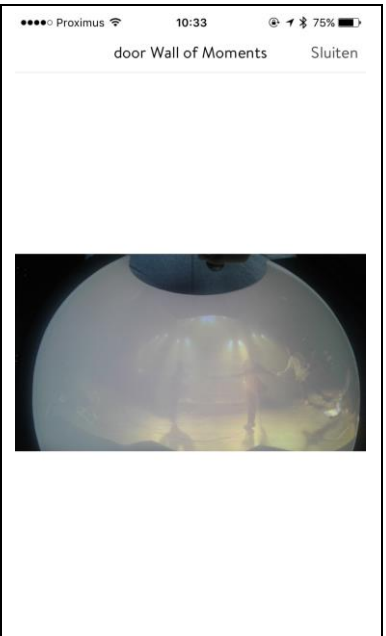

Figure 12: Individual view of contributed Moment

4 Conclusions

The interactive audio playout provides users with the ability to control their listening to live streams, such as balancing levels to their taste. It also provides the flexibility to playout over a range of different outputs, whether it is binaural playout over headphones, conventional stereo speakers or a fully immersive multi-speaker setup. By using the browser for playout, no specialist software is required to be installed, and is also platform agnostic. It can also be used across a range of different devices.

The interactive video playout allows users to navigate a range of UGC and professional video feeds, both live and recorded. The Wall app can also provide the viewer with a random selection of video feeds to help highlight different performances and moments (or indeed Moments – the short UGC video clips) at an event to help show them what is going on.

Both these systems have been put through their paces at field trials and have provided an enhanced experience for the users who have used them.

5 References

[ADM] ITU-R Recommendation BS.2076, “Audio Definition Model”
[D4.6] Audiovisual rendering engine and interface for linear broadcast production
[DASH] http://dashif.org/
[WA] https://webaudio.github.io/web-audio-api/
[JS] https://github.com/Dash-Industry-Forum/dash.js

6 Glossary

Partner Acronyms

BBC	British Broadcasting Corporation, UK
BIT	Bitmovin GmbH, AT
DTO	Technicolor, DE
iMinds	iMinds Vzw, BE
JRS	JOANNEUM RESEARCH Forschungsgesellschaft mbH, AT
TaW	Tools at Work Hard+Soft Vertriebsgmbh, AT
VRT	De Vlaamse Radio en Televisieomroeporganisatie NV, BE

Acknowledgement: The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 610370.

