



# Playout clients and content preparation & distribution



## Deliverable D5.4

ICoSOLE identifier: ICoSOLE-D5.4-BIT-PlayoutContentprepDistribution.doc

Deliverable number: D5.4

Main author of Deliverable: Rik Bauwens (VRT), Maarten Wijnants (iMinds), Mario Graf (BIT), Reinhard Grandl (BIT), Christian Timmerer (BIT)

Internal reviewer: Georg Thallinger (JRS)

Work package: WP5

Document status: Final

Confidentiality: Public

Version	Date	Reason of change
0.1	2016-09-20	Document created
0.2	2016-09-25	Wandercouch / Map of Moments
0.3	2016-10-13	Wall of Moments
0.4	2016-10-17	Efficient Omni-Directional Video Streaming
0.5	2016-10-24	Internal Review
1	2016-11-03	Final version
1.0	2016-11-10	Final version after latest formatting

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.

This document contains material, which is the copyright of certain ICoSOLE consortium parties, and may not be reproduced or copied without permission. All ICoSOLE consortium parties have agreed to full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the ICoSOLE consortium as a whole, nor a certain party of the ICoSOLE consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and does not accept any liability for loss or damage suffered by any person using this information.

## Table of Contents

---

<b>List of Figures</b> .....	<b>2</b>
<b>1 Introduction</b> .....	<b>3</b>
1.1 Purpose of this Document .....	3
1.2 Status of this Document.....	3
1.3 Related Documents .....	3
<b>2 Playout clients</b> .....	<b>4</b>
2.1 WanderCouch - Immersively experiencing music festivals from the living room .....	4
2.2 Map of Moments .....	12
2.3 The Wall.....	15
<b>3 Content preparation and distribution</b> .....	<b>18</b>
3.1 Efficient Omni-Directional Video Streaming .....	18
<b>4 References</b> .....	<b>21</b>
<b>5 Glossary</b> .....	<b>22</b>

## List of Figures

---

Figure 1: Annotated screenshot of the main screen of the WanderCouch application. ....	5
Figure 2: The UGC playback screen (with concert playback continuing in the background). ....	6
Figure 3: The topographic map screen (pinpointing video feeds captured at 5 locations). ....	7
Figure 4: The panoramic map screen; overlays visualize outbound traveling directions. ....	8
Figure 5: Professional camera feed selection during full-screen concert playback.....	9
Figure 6: Tweet visualization, UGC exploration, and UGC PiP playback during concert playback. ....	9
Figure 7: WanderCouch deployment using an Intel NUC as set-top box emulator (on the left) and using an infrared receiver (on the right) to capture RC input. ....	11
Figure 8: Annotated screenshot of the MoM prototype, populated with UGC captured during the Leffingeleuren field trial. ....	12
Figure 9: The MoM marker icons used to denote, from left to right, picture UGC, non-live video UGC and live video UGC. ....	13
Figure 10: Screenshot of the MoM application set in the context of the ICoSOLE Edinburgh field trial. ....	14
<b>Figure 11: Screenshot of the Wall on the event site</b> .....	15
<b>Figure 12: The Wall on a smartphone</b> .....	16
<b>Figure 13: Quick tutorial on how to use the app</b> .....	16
<b>Figure 14: Fast-forward</b> .....	16
<b>Figure 15: Added menu for accessing extra features</b> .....	16
<b>Figure 16: Overview of Moments reporters</b> .....	17
<b>Figure 17: List of posted Moments by a reporter</b> .....	17
<b>Figure 18: Individual view of sent-in Moment</b> .....	17
Figure 19: Tiled 360° Content .....	19
Figure 20: Tile-based Streaming JavaScript Prototype .....	20

# 1 Introduction

---

## 1.1 Purpose of this Document

---

Deliverable D5.4 is the final milestone of WP5. This document describes the work done in context of content preparation, distribution and playout and is successor of previous deliverables in this WP.

## 1.2 Status of this Document

---

This is the final version of D5.4.

## 1.3 Related Documents

---

Before reading this document it is recommended to be familiar with the following documents:

- D2.4.1 Evaluations v1
- D2.4.2 Evaluations v2
- D4.7 Audiovisual playout-engine for interactive playout
- D5.1 First version of content preparation and distribution
- D6.4 Second Demonstration and Field Trial

## 2 Playout clients

---

### 2.1 WanderCouch - Immersively experiencing music festivals from the living room

---

Attending music festivals is a popular pastime. However, there will unfortunately always be a share of fans who, for a wide variety of reasons, are unable to visit a festival of their choice in person. Potential impediments include, but are not limited to, excessive traveling overhead, ticket shortage, accessibility issues due to personal disabilities, or lack of interest among your friends to accompany you. Inspired by this observation, iMinds has conducted qualitative research concerning the expectations end-users have about being able to immersively experience music festivals from the comfort and convenience of their living room<sup>1</sup>. Out of the identified end-user expectations, iMinds has distilled an approach for remote festival engagement that is centred on the concept of blending respectively professional and user-generated content (UGC). This approach has been crystallized into a Smart TV application called *WanderCouch*. The goal of the *WanderCouch* prototype is not to replace a real visit to a music festival, but instead to provide a substitute TV-mediated experience that is able to, to some extent, submerge home users in a music festival, hereby focusing on live (as opposed to on-demand) event disclosure.

Other than the previously mentioned mixing of professional audiovisual footage with UGC, *WanderCouch*'s feature list includes, but is not limited to, the ability to alternate between vantage points while watching a musical performance, offering glimpses of the events that happen in the margin of the music festival, and diverse functions to virtually navigate the spatial area occupied by the festival in the real world. The target audience for the proposed approach consists of music enthusiasts in general and fans of music festivals in particular who, for any reason whatsoever, are unable to personally attend a specific festival.

This section will describe the functional design and the implementation of the *WanderCouch* Smart TV application; the subjective evaluation of the *WanderCouch* application is elaborated on in deliverables [D2.4.1] and [D2.4.2]

#### 2.1.1 Functional design

The *WanderCouch* Smart TV application is composed of a number of discrete screens, each of which will be discussed next in dedicated subsections. Some general remarks related to remote control interaction will also be given.

Before initiating the description of the functional design of the application, it is again explicitly stressed that *WanderCouch* primarily revolves around live conditions. Whereas UGC contributions can be consumed in an on-demand fashion, professional recordings of concerts can only be watched in real-time. Also, performing activities other than concert watching causes the timeline to progress, and spent time can never be recuperated. For example, time allocated to festival site exploration using the panoramic map (see Subsection "Panoramic map") will cause the user to miss out on (parts of) the concerts that are taking place at that same time. That being said, as will become apparent later on, measures were incorporated to mitigate this chance of missing out on interesting things and hence to help users cope with the potential "stress" induced by the imposed temporal constraints.

#### Main screen

The main screen is the central hub of the application and in a way acts as a "collage" of the application's principal functional assets. As is shown in Figure 1, it provides live video thumbnails of the musical performances that are currently taking place, houses a list of tweets related to the music festival, displays the 3 most recently shared UGC items (in the lower part of the screen), and provides direct access to both the topographic map and the panoramic map screens (which will be elaborated on in subsections "Topographic map" and "Panoramic map", respectively). Both the Twitter feed and the UGC list are updated dynamically and in real-time as respectively new tweets and new UGC content are being published. Interacting with any of the entries in the concerts overview widget causes the selected concert to be rendered in the larger video player on the left. It also causes the application's sound

---

<sup>1</sup> The results of this exploratory study are reported in deliverable [D2.4.1].

output to switch to the audio recording of the indexed concert. Interacting with the large video player itself triggers the full-screen playback of the corresponding concert (see Subsection “Full-screen concert playback”). Selecting a UGC element on the other hand initiates a transition to the UGC playback screen where the highlighted element will subsequently be rendered (see Subsection “UGC playback screen”). Finally, the Twitter feed is a read-only, non-interactive element, conforming to the users’ expectations in this regard, as discussed in deliverable [D4.2.1].

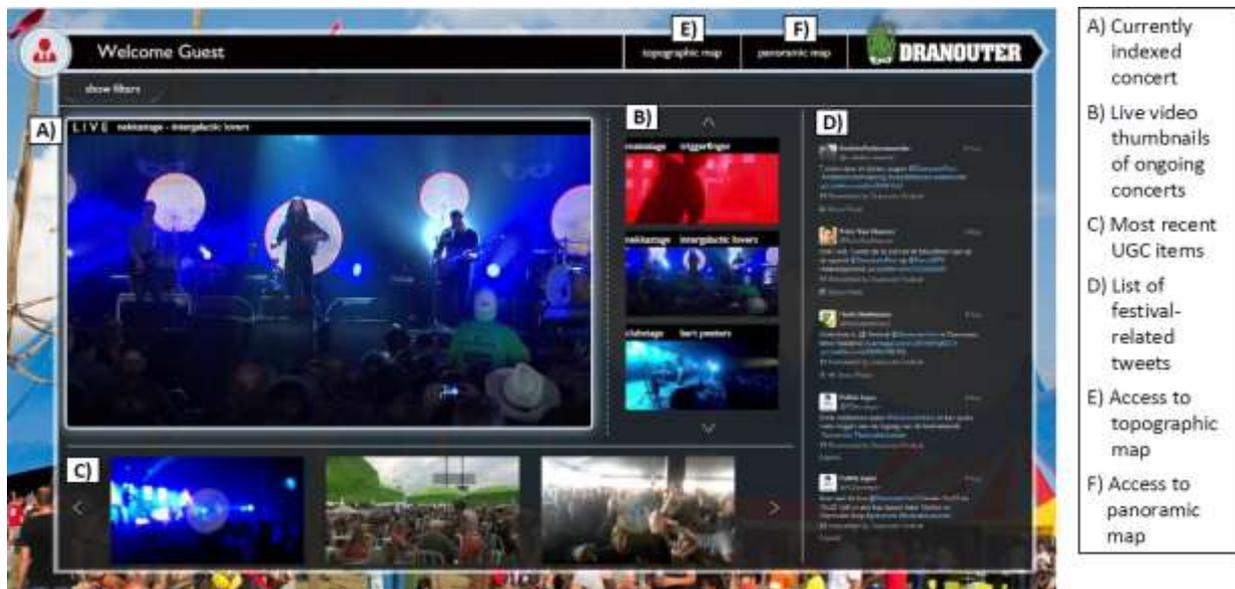


Figure 1: Annotated screenshot of the main screen of the WanderCouch application.

The main screen provides a content filtering feature that enables targeted search. Interacting with the “show filters” widget in the top-left corner of the screen unfolds a hierarchical menu in which filter criteria are divided logically into categories (e.g., location, date, rating, etcetera). Criteria from multiple filter categories can be active at the same time. Only the currently ongoing concerts and 5 latest UGC items that satisfy all selected filters will be visible on the main screen.

### UGC playback screen

As its name suggests, the UGC playback screen is primarily dedicated to the presentation and rendering of UGC items pertaining to the music festival. The layout of the screen is illustrated in Figure 2. The essence (i.e., image or video) and metadata of selected UGC elements are rendered in the top-left and top-right portion of the GUI, respectively. The presented metadata encompasses the author name, a textual description of the location where the content was created (e.g., “Main Stage”), and the creation timestamp. The metadata portion of the GUI also visualizes the community score of the currently selected element (i.e., the average of all the ratings that users of the *WanderCouch* application have assigned to it), and there is the option for users to provide their personal rating. At the bottom of the screen, per UGC type (i.e., pictures versus videos), the 20 most recent items that abide the currently applied filters are presented in a scrollable list, in inverse chronological order. With respect to filtering, the UGC playback screen offers the same options that are also available in the main screen.



**Figure 2: The UGC playback screen (with concert playback continuing in the background).**

In case the application is redirected to the UGC playback screen coming from the full-screen playback of a concert (see later), the video of the referring concert will continue in the background (see again Figure 2). This design decision is motivated by the fact that it allows viewers to keep a link to the musical performance they were watching in full-screen mode before transitioning to the UGC playback screen. In such situations, the audio output of video UGC items will always be given precedence over the audio recording of the concert that is playing in the background. In effect, as soon as the user selects a video UGC element from the list, the background concert will be muted, with its audio output only being re-enabled as soon as the playback of the UGC item has finished.

Please note that, compared to the main screen, the scrollable list of UGC items in this window is considerably more extensive. Indeed, the UGC playback screen supports the exploration of at most 40 user-contributed items, versus only 5 in the main screen. As such, the UGC playback screen affords more opportunities for serendipitous discovery of interesting footage shared by festival visitors.

### **Topographic map**

The topographic map screen offers viewers a two-dimensional, top-down overview of the festival terrain. As can be seen in Figure 3, the live professional video feed of each of the currently ongoing concerts is superimposed on the map and linked visually to (the location of) the stage where that musical performance is taking place. Users can cycle through the stages, with audio output being enabled for the currently highlighted stage only (cf. the approach taken in the main screen). By choosing the currently indexed stage, the viewer is taken to the full-screen playback of the associated concert.



Figure 3: The topographic map screen (pinpointing video feeds captured at 5 locations).

The two video annotations on the left-hand side of the map in Figure 3 are non-interactive. These video feeds are originating from festival locations other than stages and are purely meant for ambiance communication by providing viewers a sense of the non-concert-related activities and events that are taking place on the festival grounds.

### Panoramic map

The panoramic map can best be summarized as offering a virtual navigation experience akin to Google Street View, yet hereby relying on 360 degree video instead of static imagery. The map links together a finite number of discrete locations that collectively form a mesh-like topology spanning the most interesting spots on the festival terrain. Live 360 degree video footage is available for each location featured in the panoramic map, with users being able to perform unconstrained PTZ viewport modification at each single location. A 2D minimap, complemented with graphical icons overlaid on top of the 360 degree footage, inform the user of the outbound traveling options. In particular, each location that is reachable from the currently visited one is represented by a digit; by pressing the corresponding button on the remote control, the viewer “teleports” to the selected location. A screenshot of the panoramic map is provided in Figure 4.



**Figure 4: The panoramic map screen; overlays visualize outbound traveling directions.**

The panoramic map supports navigation not only to other panoramic map locations, but also to the full-screen concert playback screen. For example, in Figure 4, the 4th outgoing direction points to a festival stage. Activating such a link transfers the viewer to the full-screen playback of the concert that is taking place at the involved stage.

### **Full-screen concert playback**

This screen aims to maximally immerse the viewer in a musical performance. To this end, and in contrast to all the other screens, it lacks any form of surrounding decoration. Besides full-screen playback of professional concert footage (captured in either a traditional or a 360 degree video format), this scene offers four additional functions. These functions are hinted at by 4 color-coded icons in the top-right corner of the screen. The hues employed in the colour coding scheme match the colours of the four function buttons that are available on typical remote controls (e.g., in Europe, these buttons were originally intended for teletext purposes). The execution of a specific function is initiated by pressing the correspondingly coloured function button on the remote control.



**Figure 5: Professional camera feed selection during full-screen concert playback.**

First of all, there is the camera selection function that allows users to control their perspective of the concert. When activating this function, a mini-map representation of the involved stage is shown (see Figure 5, at the top). The mini-map illustrates both the position and orientation of the professional cameras covering the concert in question so that the viewer can make an informed decision about which vantage point to select. The second function involves a specialized version of the social media integration that was already mentioned as part of the discussion of the main screen. In effect, it is possible to visualize a Twitter feed in a semi-transparent way on top of the video playback (see leftmost image in Figure 6). This Twitter feed is tailored to the consumption context at hand, as it will encapsulate only those tweets that refer to either the current concert or involved stage. Thirdly, while watching the concert, the user is able to explore relevant UGC (central picture in Figure 6). A UGC item is classified as relevant if it is semantically linked to the watched concert and/or the corresponding stage. As is the case in the main screen context, only the 5 most recently published relevant UGC elements are included in the UGC list, with the list being refreshed dynamically at run-time. Selecting an item from the list triggers Picture-in-Picture (PiP) playback of that item (Figure 6 on the right). In case the rendered UGC is a video, its audio output will be disabled. Again selecting the UGC item during its PiP playback will initiate a redirection to the UGC playback screen (see Subsection “UGC playback screen”), where the concert playback will continue in the background (as was discussed previously). Finally, it is possible to access both the topographic and panoramic map from the full-screen concert playback scene.



**Figure 6: Tweet visualization, UGC exploration, and UGC PiP playback during concert playback.**

### Practical considerations concerning remote control interaction

The *WanderCouch* prototype is intended to be interacted with via a standard TV remote control (RC). An effort was made to render the RC interaction as intuitive as possible by logically mapping actions to

RC buttons. The arrow keys are used to move between the composing widgets of a single screen, to navigate through lists, to pan and tilt the viewport in the case of 360 degree video playback, and so on. Pressing the enter button (which is positioned in between the arrow keys on a typical RC) triggers the selection of the currently highlighted element on the screen. The back button takes the user back to the previously visited screen, whereas the home button immediately transfers the user to the main screen of the application. Finally, adjusting the audio volume is achievable via the RC's default volume control buttons.

### 2.1.2 Implementation

*WanderCouch* is implemented as a fully standards-compliant Web application, with each of the application's constituting screens corresponding with a dedicated HTML page. We used CSS to control the page layout and to stylize the application, while all application logic is handled at client side via JavaScript.

For demonstration purposes (e.g., as part of ICoSOLE's presence at the 2016 NAB Show), the application is typically deployed on a low-end Windows 8.1 PC with a small form factor (i.e., an Intel NUC) that is then directly connected to a TV screen via an HDMI cable. Conceptually, the PC fulfils the role of a set-top box, a common type of device in the current TV viewing ecosystem (see Figure 7). The PC runs an Apache HTTP server to host the application and its composing media content. All content (except for the Twitter feed) is currently hence served from local storage, with the live nature of the content being simulated entirely in software. Professional concert footage is streamed from localhost using the bitdash MPEG-DASH player<sup>2</sup>, while UGC footage is streamed progressively (also from localhost). The client-side execution environment for the application consists of a commodity Web browser (i.c., Google Chrome) that is also running on the PC and that is displayed in full-screen mode on the TV. Remote control input is captured using an infrared receiver (see again Figure 7) that is connected via USB to the PC. A combination of the WinLIRC<sup>3</sup> and Mote-Keys<sup>4</sup> open-source tools is exploited to convert RC button presses into JavaScript keyboard events, which are then acted on by the application. The Twitter feed is populated via an Embedded Timeline widget<sup>5</sup>. Finally, the annotations that are overlaid on top of both the topographic and panoramic map are realized using the JavaScript API of the Augmented Video Viewing (AVV) framework [Wijnants, 2016].

---

<sup>2</sup> <https://bitmovin.com/player/>

<sup>3</sup> <http://winlirc.sourceforge.net/>

<sup>4</sup> <http://sourceforge.net/projects/mote-keys/>

<sup>5</sup> <https://dev.twitter.com/web/embedded-timelines>



**Figure 7: WanderCouch deployment using an Intel NUC as set-top box emulator (on the left) and using an infrared receiver (on the right) to capture RC input.**

The PC-based deployment context was chosen due to the comfort with which Web applications can be developed and debugged on such a setup. The application could however readily be converted into a native Smart TV application by leveraging any of the proprietary SDKs provided by Smart TV manufacturers (e.g., <https://www.samsungdforum.com/Tizendevtools/Sdkdownload>). Furthermore, Smart TVs are increasingly being shipped with built-in HTML5-compliant Web browsers that could directly render the client-side part of our implementation to the TV screen.

A significant benefit of a purely Web-based implementation is its platform independence. In effect, the application could be ported with minimal effort to deployment contexts other than the considered PC and Smart TV setup (e.g., tablets or low-cost, non-PC-based set-top boxes). At the same time, the Web-driven approach allows for “branding” of the application, simply by changing some HTML mark-up and CSS instructions. As such, designers and artists need to possess only generic Web development skills to be able to thematically tailor the appearance of the application to the music festival at hand.

### **2.1.3 Featured content**

Content-wise, the *WanderCouch* prototype is populated with footage that was accumulated during the ICoSOLE Y2 field trial at the 2015 Dranouter music festival (<http://www.festivaldranouter.be/>) in Belgium. The content corpus encompasses directed feeds of 3 integral concerts, extended with 5 professional recordings depicting 2 of these concerts from different perspectives (respectively 3 and 2 static cameras per concert), amounting to more than 10 hours of professional video footage in total. Exactly one of these video recordings is in the 360 degree format, covering the performing artist, the integral stage, and (the front of) the attending audience. Besides professional material, we also integrated 103 UGC contributions produced by actual festival visitors (76 videos and 27 pictures). These contributions were manually curated and selected, in the sense that a human editor scanned through the individual items and discarded the ones that contained inappropriate content or that did not satisfy predefined audiovisual quality standards (e.g., video too blurry or shaky).

## 2.2 Map of Moments

Analogous to the *WanderCouch* prototype described in Section 2.1, the *Map of Moments* (MoM) is a tool that affords an alternative way to discover, explore and participate in real-world large-scale events. However, whereas *WanderCouch* aims to provide users a rather holistic event-related experience, MoM focusses primarily on (the geospatial characteristics of) UGC footage shared by event visitors. Another dissimilarity between *WanderCouch* and MoM concerns their deployment and consumption options: whereas *WanderCouch* is chiefly intended to be deployed in a residential context, the potential consumption space of MoM is more broad in that it could be exploited both on- and off-site to the event. Furthermore, the MoM application is not necessarily tied to domestic off-site consumption contexts but could, for example, also be leveraged by mobile users.

### 2.2.1 Motivation and rationale

The MoM application utilizes a topography-inspired approach towards real-world event disclosure. As a result, in the MoM prototype, all interaction with the real-world event occurs on top of a topographic map visualizing the event's geospatial layout. In this context, the prime feature of the MoM application consists of pinpointing UGC contributions from event attendees on a map based on their physical capture location (see Figure 8 and Figure 10). Leveraging a location-driven visualization scheme for event-related UGC contributions offers a number of opportunities, not only for end-users aiming to get a virtual impression of the event at hand, but also for analytical purposes. The following is a non-exhaustive list of such opportunities:

- Easily discover “hotspots” (i.e., locations with above average activity) in the geographical area covered by the event.
- Analyse the migration behaviour of on-site visitors of the event.
- Identify and experience the prevailing atmosphere at a certain geospatial sub-area of the real-world event.
- Provide remote look-in at distant sub-areas of the real-world event for physical visitors.

### 2.2.2 Functional design

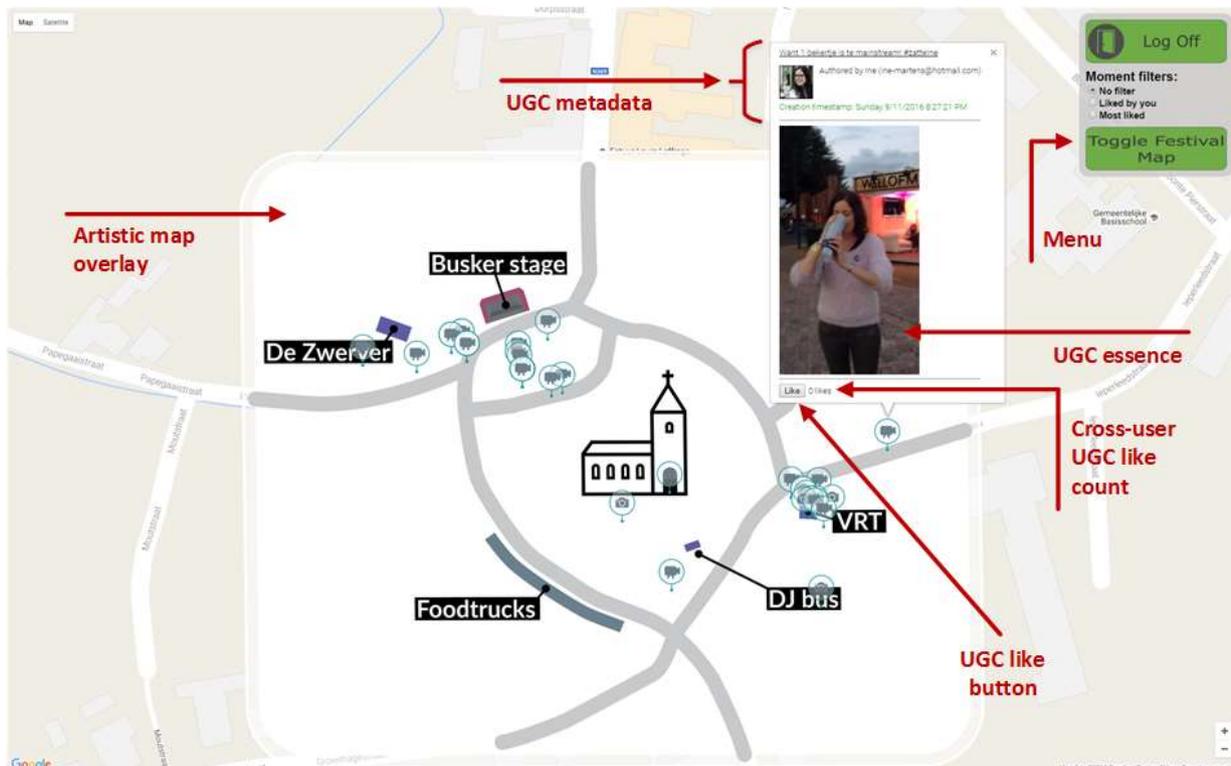
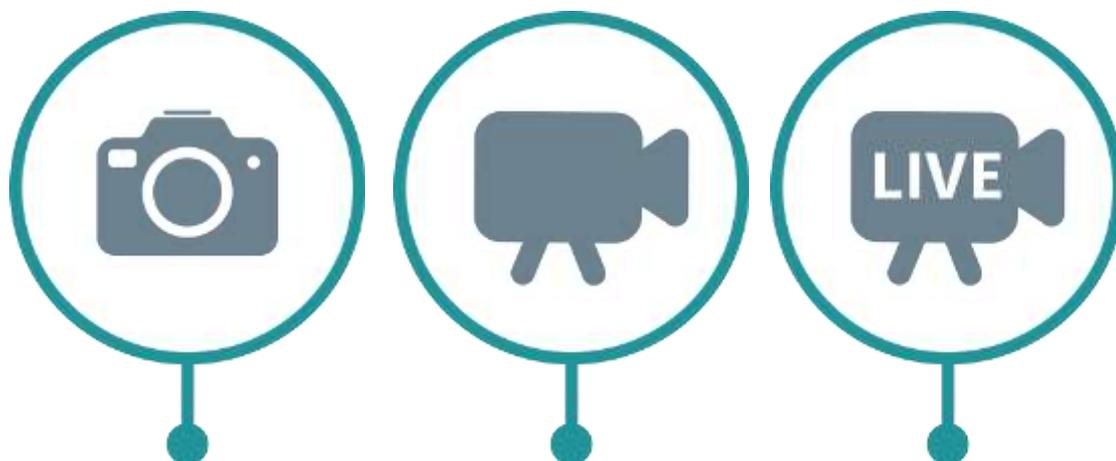


Figure 8: Annotated screenshot of the MoM prototype, populated with UGC captured during the Leffingeleuren field trial.

The annotated screenshot shown in Figure 8 illustrates the design of the MoM demonstrator. As can be told from this screenshot, a topographic map forms the (dynamic) backdrop of the application. Users can pan and rotate the map as well as perform zoom in/out operations as they see fit. Different types of UGC contributions (i.e., photos versus offline videos versus live videos) are pinpointed on this map using dedicated icons (see Figure 9). As such, the MoM application readily visualizes the capture locations of UGC material shared by event visitors.



**Figure 9: The MoM marker icons used to denote, from left to right, picture UGC, non-live video UGC and live video UGC.**

When a user selects a UGC icon (i.e., by clicking or tapping it), a call-out widget is drawn on top of the topographic map; the widget holds both some metadata and the media essence pertaining to the UGC item. In the case of video UGC, its essence will autoplay as soon as the call-out widget is opened. The UGC metadata encompasses the title assigned to the content by the author, the name and e-mail address of the author, and the UGC creation timestamp; if the author selected an avatar representation in the UGC capture app, the avatar picture will also be visualized as part of the UGC metadata in the call-out window. Besides UGC metadata and media essence, the call-out widget also holds a “like” button (cf. Facebook) to allow users to endorse specific UGC items.

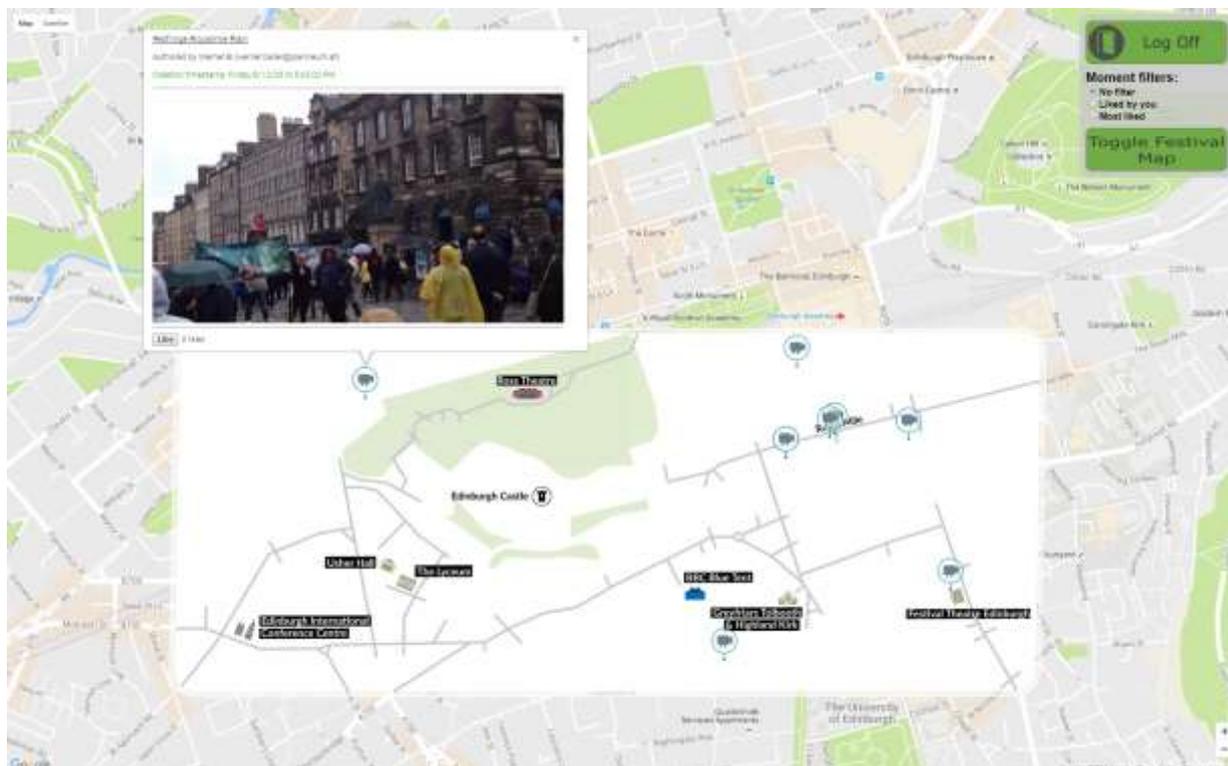
In the top-right corner of the screen, a menu is drawn on top of the map (in a semi-transparent manner) that allows users to control the MoM application. In particular, this menu holds two buttons, respectively to log off and to toggle the visualization of an artistic map overlay, as well as UGC filtering provisions implemented in the form of a radio button group. These features will be described separately in the 3 subsequent paragraphs.

Users need to register themselves (and be logged in) before they are able to exploit the MoM demonstrator. Other than for access restriction purposes, the registration requirement is motivated by the fact that the MoM application leverages user-specific state information that needs to be persisted across usage sessions. For the time being, this user-specific state is limited to the UGC likes that (s)he has issued, yet it could readily be extended in the future.

The artistic map overlay feature enables a schematic representation of (the geographic traits of) the real-world event to be superimposed on the topographic map. One example advantage of such an artistic representation is that it allows topographic simplification of the covered geospatial area while at the same time granting opportunities with respect to highlighting the event’s most important landmarks.

Especially for real-world events whose geographic extent is rather limited, situations might arise where the topographic map is cluttered with (clusters of) UGC contribution markers. The native zooming functionality of the leveraged topographic map library (see Section 2.2.3) already mitigates this issue to some extent. However, for situations where the user would like to have a high-level (i.e., not zoomed in), non-cluttered overview of the event, the MoM demonstrator includes filtering functionality. For the time being, two specific UGC filters (“Liked by you” and “Most liked”) have been implemented. Enabling the “Liked by you” filter removes all markers from the map that pertain to UGC contributions that have not (yet) been liked by the local user. On the other hand, the “Most liked” filter visualizes only those UGC items that have received the largest amount of likes across the complete MoM user base. Extra types of UGC filters (e.g., temporal ones) could easily be integrated in the MoM application.

The MoM demonstrator has been showcased in the context of both the Edinburgh and Leffingeleuren ICoSOLE Y3 field trials (please see deliverable [D6.4] for more information). A screenshot of the MoM application populated with Leffingeleuren content has already been given in Figure 8; a (non-annotated) MoM screenshot featuring Edinburgh content can be found in Figure 10.



**Figure 10: Screenshot of the MoM application set in the context of the ICoSOLE Edinburgh field trial.**

### 2.2.3 Implementation

The MoM prototype is implemented as a Web application and is hence intended to be executed in a Web browser environment. The outcome is a highly portable and platform independent solution for real-world event disclosure. In effect, the MoM demonstrator to date has successfully been deployed on the Windows Operating System (OS) on desktop/laptop, on iOS-based mobile devices, and on mobile devices running the Android OS; deployment on other contexts have not yet been investigated, yet little problems are anticipated in this regard.

To render and interact with the topographic map, the MoM implementation leverages version 3 of the Google Maps JavaScript API (<https://developers.google.com/maps/documentation/javascript/>). To pinpoint UGC contributions from event visitors on the topographic map, GPS metadata (registered by the UGC capture application) is exploited.

All relevant UGC metadata is gathered from a Firebase back-end that implements the database schema employed by VRT's Wall of Moments demonstrator [D4.7]. User authentication is based on username/password combinations and is also managed via Firebase. Finally, the Firebase back-end is exploited to store the user-specific state (i.e., UGC like metadata). To interact with the Firebase back-end, the MoM application exploits the Firebase JavaScript API (<https://firebase.google.com/docs/reference/js/>).

Contrary to UGC metadata, UGC essence is not stored in the Firebase database; instead, the Firebase entry of a specific UGC item encompasses a URL to the external location where the UGC media essence is stored. Currently, Amazon S3 storage is exploited to store the UGC media essence. The MoM demonstrator gains access to the UGC essence by leveraging version 2.4.5 of the Amazon Web Services (AWS) SDK for JavaScript (<http://docs.aws.amazon.com/AWSJavaScriptSDK/latest/index.html>).

For implementing the playback of video UGC, a discrimination is made between respectively offline/uploaded and live/streamed video clips. The former category is transported from Amazon S3 to the client device using HTML5 progressive streaming; the latter category on the other hand is streamed using MPEG-DASH. To implement the MPEG-DASH streaming (and resulting video essence playback), the bitdash HTML5 player from consortium partner bitmovin is used (<https://bitmovin.com/player/>).

Google Maps applies a tile-based approach towards map visualization (e.g., to enable efficient distribution of map content). As such, overlaying a custom graphic on top of a Google Maps instance requires the spatial segmentation of the visual overlay into disjoint tiles. In addition, this tiling operation needs to be performed at different resolutions for different zoom levels (i.e., increased resolution per additional zoom level). To produce the tile imagery for the artistic map overlays, a license for the commercial MapTiler software (<http://www.maptiler.com/>) has been procured.

Finally, the maximum number of UGC items that at any time will be visualized simultaneously in the MoM demonstrator is configurable in software. Limiting the amount of concurrent UGC markers to the 50 most recent contributions has empirically been found to strike a suitable balance between marker clutter and comprehensive UGC-based coverage of the event at hand.

## 2.3 The Wall

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



Figure 11: Screenshot of the Wall on the event site

### 2.3.1 User-generated content

User-generated content is collected using the Moments and Focus app. This content can be either a photo or video, and is tagged with an author, timestamp and location (depending on origin). Once this content has been reviewed and approved, it can be picked up by the Wall, which shows it in a carousel (**Figure 11**). The order of displayed Moments depends on different factors. For the Walls at the event site, the base principle is to feature recently added Moments prominently, and to intertwine this with some randomly chosen older Moments from time to time. When sensors are attached to The Wall, Moments from people nearby are displayed approximately a third of the time as well, to make the experience more tailored and personal. People who install The Wall app on a smartphone have a slightly modified experience (**Figure 12**). Moments are ordered by time, recent first. They have the

ability, however, to quickly skip a Moment or a collection of Moment, and rewind to the most recent one (Figure 13).



Figure 12: The Wall on a smartphone



Figure 13: Quick tutorial on how to use the app



Figure 14: Fast-forward

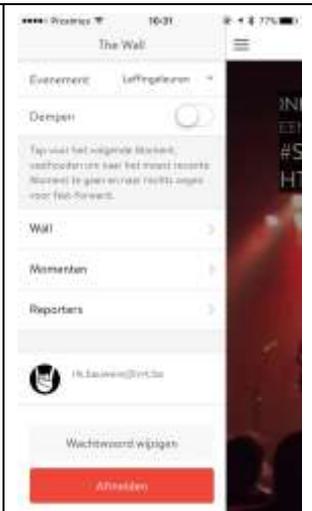


Figure 15: Added menu for accessing extra features

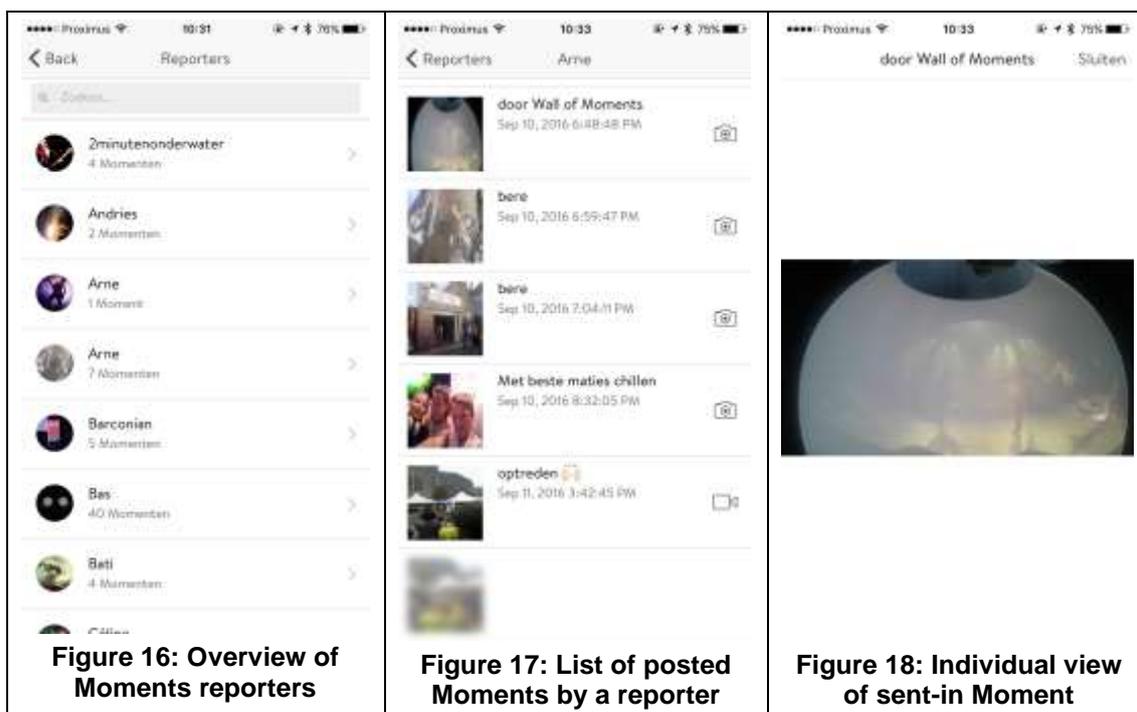
### 2.3.2 Professional content

For the screen on the event site, we enabled the Wall to pick up live streams as well. To achieve this, we integrated the Bitdash player from Bitmovin. In Trademark, we 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.

### 2.3.3 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 14**). Moreover, a menu is available (

**Figure 15**), from where a list of sent-in Moments is available, as well as a list of Moments reporters (**Figure 16**). A user can search for a specific reporter, and request a list of (approved) Moments for that reporter (**Figure 17**). From that list, the Moment can be displayed, picked and played individually (**Figure 18**).



**Figure 16: Overview of Moments reporters**

**Figure 17: List of posted Moments by a reporter**

**Figure 18: Individual view of sent-in Moment**

## 3 Content preparation and distribution

---

### 3.1 Efficient Omni-Directional Video Streaming

---

Omni-directional video (ODV) content allows the user to change the viewing orientation while watching a video. Such video content can be consumed using different devices ranging from mobile phones to special head-mounted displays (HMD). Since a full 360°-panorama has to be stored for ODV content, the file sizes can be huge compared to traditional videos, where only a predefined and fixed viewing angle is provided. These huge file sizes can be a challenge when it comes to streaming of such content. By carefully choosing the content representation and by utilizing new delivery approaches, the efficiency of streaming ODV content can be increased while preserving the image quality presented to the user.

#### 3.1.1 Motivation

When it comes to streaming, it's important to understand some characteristics of ODV content. If we have a device which is able to display a horizontal field of view (FoV) of 120°, only a third of the full 360° panorama is displayed at once in the horizontal direction. Some head-mounted displays (HMDs) only offer a FoV of 90° which would result in only a fourth of the full panorama displayed at once. When we further want to render the content of the visible FoV in 2K resolution, our full 360° content would need a resolution of 6K to 8K depending on the horizontal FoV angle. Assuming a 4K FoV we would need a 12K to 16K resolution video to encode the full panoramic video.

Since even today it is a real challenge to transmit such huge amounts of data over the Internet, there is a strong need for efficient ODV streaming technologies. Treating ODV content just like traditional video content and applying the same streaming technologies won't work well in the end, given today's network conditions.

#### 3.1.2 Maps, Projections and Coding

For storage, the captured ODV is encoded with a video codec one would also use for traditional video content. These codecs like H.264/AVC and H.265/HEVC need a rectangular two-dimensional video representation as input in order to work well. To get such a two-dimensional representation of the viewable sphere we need map projections. Each map projection introduces distortion in the shape, area, distance, or direction of the content. Which properties of the projected content are distorted and which are preserved is defined by the specific projection used [Kennedy, et al. 1994]. Map projections can be also categorized by the type of projection surface they use. Projection surfaces can be usually easily unwrapped into a two-dimensional space. Commonly used projection surfaces are conus, cylinder, cube, and plane [Kennedy, et al. 1994]. For the purpose of ODV, most commonly used are the cylinder and the cube as projection surfaces.

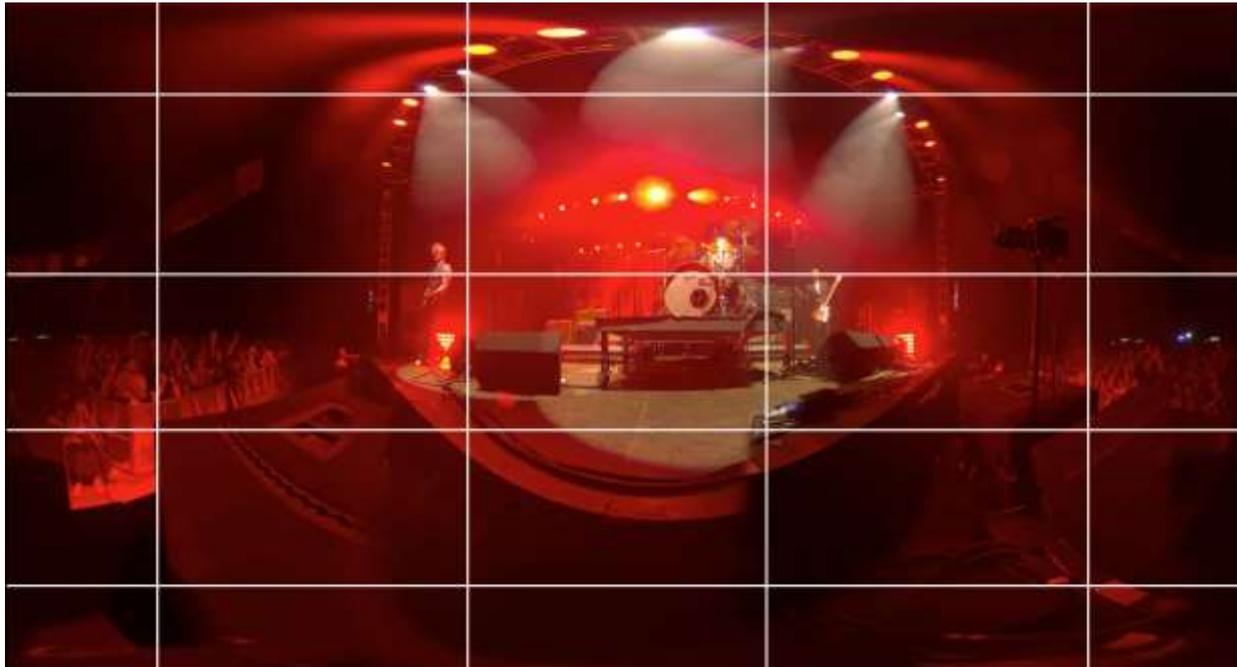
#### 3.1.3 Storage/Delivery and Playback

For simple ODV streaming the omnidirectional content is treated like a common video stream. The same packaging and storing technologies can be used. Also the same technologies (DASH, HLS, RTP) can be used as for common video streams. For playback a special video player is needed. The video player must know which projection was used to create the omnidirectional video. Based on this information the player can properly render the ODV based on the current viewing angle chosen by the user.

#### 3.1.4 Tiled Streaming

Since modern camera rigs can often capture content with resolutions of 8K and above, there is a high need to use bandwidth efficient content representations for ODV. Given such high video resolutions, the delivery of ODV over a network can easily become a critical bottleneck.

The idea of tiled streaming is to divide the video into a grid of rectangular tiles, as it can be seen in Figure 19. Each of these tiles can then be encoded as separate video streams which are independent of each other from a coding perspective. Therefore, these tiles can be independently streamed, decoded and rendered to the user. Each tile can generally be encoded in multiple different bitrates and resolutions, so that for playback each tile can be adapted with respect to the quality.



**Figure 19: Tiled 360° Content**

The chosen size of the tiles strongly influences the bandwidth efficiency of tiled-streaming. Larger tiles lead to a better coding efficiency but also a lot of unused data is transmitted when only part of a tile is within the currently visible area. Smaller tile sizes decrease coding efficiency but the wasted bandwidth also decreases in this case [Heymann, et al. 2005].

Some newer video coding standards like H.265/HEVC provide a feature called tiles [Misra, et al. 2013]. Tiles in HEVC were introduced to enable the independent and parallel decoding process of rectangular parts of the video to utilize the availability of multiple processor cores on the client device. This feature can be used to realize a compressed domain mixing of tiles without the need of full or partial decoding of the different tiles. This compressed domain mixing could be realized as a network service but can be also implemented on the client side.

### **3.1.5 DASH Prototype Implementation**

For our prototype implementation, we decided to work towards a client-based approach. Furthermore, we also concluded to make use of a single decoder for all tiles. For this approach the different tiles which should be displayed need to be mixed into a single video stream before being passed to the decoder. This solves two issues at once: Only one decoder is required and there is no need for synchronization between the tiles. For example, the number of parallel decoding pipelines on client devices is limited, specifically on mobile devices, and requiring only one decoder also contributes to better utilization of power/battery consumption.

Our prototype implementation is fully written in JavaScript and uses the Media Source Extensions (MSE) browser interface for decoding and rendering. As it can be seen in Figure 20, switching the quality of each tile is done manually, using the controls beneath the video. For our prototype implementation, we used three different quality levels for each tile and a 6x4 tiling scheme. Where a higher number of tiles in general might lead to an increased user experience, but decreases the efficiency of the codec. Research about the influence of the tile scheme on coding efficiency and QoE is currently ongoing.



Tiling scheme selector

0	0	0	0	0	0
0	1	2	2	1	0
0	1	2	2	1	0
0	0	0	0	0	0

Figure 20: Tile-based Streaming JavaScript Prototype

Figure 20 shows a scenario, where the high quality tiles (quality level 2), which corresponds to the FoV of the user, are located in the center of the entire equirectangular image. The tiles to the near left and right of the FoV are streamed in a medium quality (quality level 1) and the regions with highest distance to the FoV, in lowest quality (quality level 0).

For such tiled streaming systems based on DASH, there is a need to describe the position and size of the single tiles, so that the player can download and render the content properly. For this purpose, the Spatial Representation Description (SRD) feature of the second amendment of MPEG DASH standard part 1 (media presentation description and segment formats) [SRD, 2015] can be used. SRD was introduced to describe how content is spatially organized.

## 4 References

---

[D2.4.1] Evaluations v1

[D2.4.2] Evaluations v2

[D4.7] Audiovisual playout-engine for interactive playout

[D5.1] First version of content preparation and distribution

[D6.4] Second Demonstration and Field Trial

[Wijnants, 2016] Maarten Wijnants, Kris Van Erum, Peter Quax and Wim Lamotte, "Augmented ODV: Web-Driven Annotation and Interactivity Enhancement of 360 Degree Video in Both 2D and 3D", Lecture Notes in Business Information Processing (LNBIP) volume 246, 2016

[Kennedy, et al. 1994] M. Kennedy and S. Koop. Understanding map projections. GIS by ESRI, Environmental System Research Institute, 1994.

[Heymann, et al. 2005] S. Heymann, A. Smolic, K. Mueller, Y. Guo, J. Rurainsky, P. Eisert, and T. Wiegand. Representation, coding and interactive rendering of high-resolution panoramic images and video using mpeg-4. In Proc. Panoramic Photogrammetry Workshop (PPW), 2005.

[Misra, et al. 2013] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou. An overview of tiles in hev. IEEE Journal of Selected Topics in Signal Processing, 7(6):969–977, Dec 2013.

[SRD, 2015] Spatial relationship description, generalized URL parameters and other extensions, ISO/IEC 23009-1:2014/amd 2:2015.

## 5 Glossary

---

AVV	Augmented Video Viewing
CSS	Cascading Style Sheets
FoV	Field of View
HMD	Head-Mounted Display
HTML	HyperText Markup Language
MLV	Multi-depth Layered Video
MoM	Map of Moments
ODV	Omni-Directional Video
PiP	Picture-in-Picture
RC	Remote Control
SRD	Spatial Representation Description
UGC	User-Generated Content

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.