**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2015/N15734**

**MPEG 113, October 2015, Geneva, Switzerland**

|  |  |
| --- | --- |
| **Source** | **Requirements, Systems** |
| **Status** | **Final** |
| **Title** | **Context and Objectives for Media Orchestration v.2** |
| **Editor** | **Rob Koenen** |

Context and Objectives for Media Orchestration

# Introduction

This document captures the context, objectives, uses cases and environments for Media Orchestration.

Shortly summarized, and further detailed below, the proliferation of capture and display devices combined with ever-increasing bandwidth, including mobile bandwidth, necessitates better and standardized mechanisms for coordinating such devices, media streams and available resources, like media processing and transmission.

The aim of this document is to start setting the scope of a potential work item on Media Orchestration. This documents does the following:

* Discuss relevant developments in industry;
* Introduce the concept of Media Orchestration;
* Look at use cases and requirements;
* Explore MPEG’s role in Media Orchestration

For the definition of certain terms used in this document, please see the most recent version of the Media Orchestration Context and Objectives [47]*.*

# Environment

Audiovisual equipment is pervasive today; everyone with a smartphone, a tablet or a laptop has both recording and display equipment at their disposal, usually connected to a local or a public network. This equipment is increasingly sophisticated, with higher resolutions and better lenses (for video), often multiple microphones (for audio), coupled with increasingly sophisticated processing capabilities. These devices can not only decode in real time, but often also encode.

Sensors and displays do not only come in the form of personal smart phones. There are smart watches, several types of omnidirectional cameras (e.g., bubl [2]), and a large number of consumer price level Virtual Reality and Augmented Reality glasses and goggles have started to appear (Oculus Rift [3], Microsoft’s HoloLens [4], Samsung Gear VR[5], and Google Glass[6] that is now being completely redesigned). Oculus Rift was bought by Facebook for a value that could reach 2 billion USD [8] because it believes that the technology goes beyond gaming. Quoting Mark Zuckerberg: *“This is really a new communication platform. By feeling truly present, you can share unbounded spaces and experiences with the people in your life”* [9]. Also, screens are getting larger, higher resolutions (4K, maybe even 8K) and sometimes even curved. Several audio technologies can render immersive sound, including MPEG’s 3D audio standard [10].

At the same time, production of media is becoming a normal, everyday consumer activity, aided by many enhancements and sharing apps, often combined in one. Instagram [11] and Vine [12] are just two examples. New consumer devices are essentially always connected, and increasingly consumer devices are not just used for sharing with peers, but also for helping create a professional production. An example can be found in the FFP7 project STEER, which combines professional feeds with user-generated content to create what it calls “Social Telemedia”.

Another relevant trend is the use of multiple devices simultaneously while consuming media. This often means just browsing or chatting while watching TV, but a few innovative second screen concepts combine multiple streams and devices to create a single experience, and the just released HbbTV 2.0 specification includes tools for second screen support and synchronization across devices.

Other interesting examples include Samsung’s Chord [15] and Group Play [16] technologies, that allow devices to be combined for a single, synchronized experience, even without network access (the latter, Group Play, seems to be deprecated now).

# Media Orchestration use cases

With so many capture and display devices, and with applications and services moving towards a more immersive experience, we need the tools to be able to manage multiple, heterogeneous devices over multiple, heterogeneous networks, to create a single experience. We call this process Media Orchestration: orchestrating devices, media streams and resources to create such an experience. Indeed, the output of the Uniform Timeline Alignment activity already mentions the need for *“Orchestrating the synchronization between devices”* [17].

Media orchestration:

* Applies to capture as well as consumption;
* Applies to fully offline use cases as well as network-supported use, with dynamic availability of network resources;
* Applies to real-time use as well as media created for later consumption;
* Applies to entertainment, but also communication, infotainment, education and professional services;
* Concerns temporal (synchronization) as well as spatial orchestration;
* Concerns situations with multiple sensors (“Sources”) as well as multiple rendering devices (“Sinks”), including one-to-many and many-to-one scenarios;
* Concerns situations with a single user as well as with multiple (simultaneous) users, and potentially even cases were the “user” is a machine, although this is not yet represented in the use cases. This may have a relation with the notion of “Media Internet of Things” that is also discussed in MPEG.
* …

A large number of relevant use cases is included in [17], and these are included in this document as well. Additional media orchestration use cases, as described in [22], also have a spatial dimension. The temporal part of media orchestration is required now, and it may be argued that the use cases with a spatial dimension are looking a bit further out in time. Annex A collects the relevant use cases from [17] and [22].

## Analysis of use cases for media orchestration

Table 1 provides a summary of use cases from both [17] and [22], with a high-level labelling.

Table 1 - Summary of use cases and global labelling.

| ***Label*** | **Examples** |
| --- | --- |
| *Accessibility* | **Use cases 7.1.1, 7.2.7**   * ancillary video stream containing sign language; * audio description of broadcasts |
| *Companion / 2nd screen services* | **Use cases 7.2.1, 7.2.2, 7.2.3, 7.2.4, 7.1.1, 7.2.6, 7.2.9, 7.2.11, 7.4.7, 7.7.1, 7.4.6**   * multiple languages; ancillary audio track, e.g. audio commentary; * ancillary subtitle stream, i.e. closed caption information; * additional videos, e.g. multi-angle advertisements; * content lists; * time-limited game-show participation |
| *Social TV / TV as online social event* | **Use case 7.2.12**   * watching apart together, including real-time communication; * presence based games, on-line election events |
| *Scalable coding (UHD, HFR, HDR) enhancements* | **Use cases 7.1.5, 7.4.1, 7.3.6**   * video is enhanced by a temporal, spatial or multi-view scalable layer to achieve either high frame rate, high resolution or 3D experience |
| *Remote and distributed audiovisual events* | **Use cases 7.1.3, 7.1.4, 7.2.10**   * distributed tele-orchestra; * networked real-time multiplayer games; * multiparty multimedia conferencing; * networked quiz shows; * synchronous e-learning; * synchronous groupware; * shared service control; * security and surveillance, using on-site cameras to get better picture of situation and directing locally available sensors; |
| *Multi-Source* | **Use cases 7.3.1, 7.3.2, 7.3.3, 7.3.4, 7.3.5, 7.4.1, 7.4.7**   * live augmented broadcast, combining professional capture with user-generated content; * classroom recording, making coherent and interactive representation by combining several independent recordings; * collaborative storytelling, creating interactive representations of events (holiday, weekend away) by combining individual resources; * shared concert capture, to recreate the concert from your vantage point but with professional quality; * recording an audiovisual scene with multiple cameras for 3D and VR representation; * crowd journalism, with automatic grouping of recordings of the same event, and directing users to make recordings |
| *Multi-Sink* | **Use cases 7.2.5, 7.2.8, 7.2.11, 7.4.5, 7.4.7, 7.5.1, 7.5.2, 7.5.4**   * modular speakers and/or screens, e.g. networked stereo loudspeakers, phased array transducers, video walls; * continue watching on another device, seamless switching among media devices |

Similar to [17], this document considers the dimensions of orchestration and categorizes the use cases along those dimensions. Document [17] distinguishes three dimensions, i.e. the device, stream and communication dimension. For media orchestration, the spatiotemporal dimension and the ingest/rendering dimension are required in addition to the dimensions from [17], further denoted as “Source” and “Sink”.

The device dimension is about how many devices are used by each user. It has two possible values: one device and multiple devices.

* **One device means** that one device (e.g., TV, tablet or smartphone) is used to capture content, or to browse content.
* **Multiple devices** means that e.g. one or more users choose to use multiple devices to play content, or that more users each use their device to capture and ingest content.

The stream dimension addresses about how many streams are generated by a content sharing host and received by Sinks. It has two possible values: one stream and multiple streams.

* **One stream** means that the host shares only one multimedia stream, which may be, for example, a pure video, timed text or a multiplexed one.
* **Multiple streams** mean that the host shares multiple multimedia streams.

The spatiotemporal dimension indicates a focus on temporal, spatial, or spatiotemporal orchestration aspects. It has three possible values: spatial, temporal, and spatiotemporal.

* **Spatial** means that mainly the spatial aspect of orchestration is relevant.
* **Temporal** means that the temporal or synchronization aspect is relevant.
* **Spatiotemporal** means that both aspects are relevant in realizing the use case.

The ingest/rendering dimension indicates the primary role of devices, i.e. capture/ingest device and/or rendering/presentation device. It has three possible values: ingest, rendering and both.

* Ingest and Rendering speak for themselves. They correspond to the notions of **Source** and **Sink**, which are the terms that are used throughout this document.
* **Both** means that both Source and Sink aspects are relevant in realizing the use case.

Examples of capture and ingest are upcoming live mobile video streaming services such as Meerkat [23] and Periscope [24], whereas as rendering involves both established second-screen cases, as well as more advanced multi-device rendering cases such as Chord [15] and Group Play [16].

Use cases with a focus on temporal orchestration and rendering were included from [17], and can be typically realized with media synchronization approaches. Where [17] focuses on the temporal aspects and therefore on the communication aspects, the focus in this document is broader. This document follows a slightly different, more traditional decomposition in Section 4**.** below.

According to different value combinations in those four dimensions of orchestration, there are a total of 36 categories in theory. As a next step in the use case analysis, below is a categorization of the use cases in Annex A according to the four dimensions listed above

| **Sink** | **Orchestration** | | **# Use Case** |
| --- | --- | --- | --- |
| **Across Devices** | **Across Streams** |
| One device, One stream, Spatial, Source |  |  |  |
| One device, One stream, Spatial, Sink |  |  | **7.6.1** |
| One device, One stream, Spatial, Source and Sink |  |  |  |
| One device, One stream, Temporal, Source |  |  |  |
| One device, One stream, Temporal, Sink |  |  | **7.7.2** |
| One device, One stream, Temporal, Source and Sink |  |  |  |
| One device, One stream, Spatio-temporal, Source |  |  |  |
| One device, One stream, Spatio-temporal, Sink |  |  |  |
| One device, One stream, Spatio-temporal, Source and Sink |  |  | **7.6.2** |
| One device, Multiple Streams, Spatial, Source |  |  |  |
| One device, Multiple Streams, Spatial, Sink |  |  |  |
| One device, Multiple Streams, Spatial, Source and Sink |  |  |  |
| One device, Multiple Streams, Temporal, Source |  |  |  |
| One device, Multiple Streams, Temporal, Sink |  | X | **7.1.1, 7.1.2, 7.1.3, 7.1.4, 7.1.5, 7.7.1** |
| One device, Multiple Streams, Temporal, Source and Sink |  |  |  |
| One device, Multiple Streams, Spatio-temporal, Source |  |  |  |
| One device, Multiple Streams, Spatio-temporal, Sink |  |  |  |
| One device, Multiple Streams, Spatio-temporal, Source and Sink |  |  |  |
| Multiple devices, One stream, Spatial, Source |  |  |  |
| Multiple devices, One stream, Spatial, Sink |  |  |  |
| Multiple devices, One stream, Spatial, Source and Sink |  |  |  |
| Multiple devices, One stream, Temporal, Source |  |  |  |
| Multiple devices, One stream, Temporal, Sink | X |  | **7.2.5** |
| Multiple devices, One stream, Temporal, Source and Sink |  |  |  |
| Multiple devices, One stream, Spatio-temporal, Source |  |  |  |
| Multiple devices, One stream, Spatio-temporal, Sink | X |  | **7.2.8, 7.5.1, 7.5.2,7.5.3, 7.5.4** |
| Multiple devices, One stream, Spatio-temporal, Source and Sink |  |  |  |
| Multiple devices, Multiple Streams, Spatial, Source |  |  |  |
| Multiple devices, Multiple Streams, Spatial, Sink |  |  |  |
| Multiple devices, Multiple Streams, Spatial, Source and Sink |  |  |  |
| Multiple devices, Multiple Streams, Temporal, Source |  |  |  |
| Multiple devices, Multiple Streams, Temporal, Sink | X | X | **7.2.1, 7.2.2, 7.2.3, 7.2.4, 7.2.5, 7.2.6, 7.2.7, 7.2.9, 7.2.11, 7.2.12,** |
| Multiple devices, Multiple Streams, Temporal, Source and Sink |  |  |  |
| Multiple devices, Multiple Streams, Spatio-temporal, Source | X | X | **7.3.1, 7.3.2, 7.3.3, 7.3.4, 7.3.5, 7.3.6, 7.3.7, 7.4.1, 7.4.2, 7.4.3, 7.4.4,** |
| Multiple devices, Multiple Streams, Spatio-temporal, Sink |  |  |  |
| Multiple devices, Multiple Streams, Spatio-temporal, Source and Sink | X | X | **7.2.10, 7.4.5, 7.4.6, 7.4.7, 7.4.67.8.1** |

**Table 2 - Categorization of the use cases.**

# Technical Framework

The technical framework distinguishes three independent layers:

1. The Functional Architecture,
2. Data Model and Protocols
3. Data representation and encapsulation

These layers somewhat resemble the functional level, carriage level and implementation level as used in [17]; this analysis uses the decomposition above instead as it is closer aligned with MPEG’s work and expertise.

A clean separation allows developing separate tools for each layer, which provides a clean design. The functional architecture defines the different media components that needs to be orchestrated against each other and how. The framework considers concepts such as functional roles, Timed Data and processing functions. Media processing in the network may optionally be used in media orchestration. Examples of media processing are transcoding, adding or changing timelines, multiplexing of Timed Data, selection of Timed Data (orchestrator with an editor role) and tiling (MPEG DASH SRD).

The current document only addresses the Functional Architecture. The Data Model and Protocols as well as Data representation/encapsulation are subject to an upcoming Call for Proposals

##### Timed Data

The concept of Timed Data was introduced by DVB CSS, see TS 103 286-1 [25], clause 5.2.1. (Note that DVB uses the term “timed content” for this concept.) Timed Data has an intrinsic timeline. It may have a start and/or end (e.g. Content on-Demand), or it may be continuous (broadcast). It may be atomic (video-only) or it may be composite (A/V, 3D-audio, multiplex). Classic examples of Timed Data are video, audio and timed text. In the context of media orchestration, also streams of location and orientation, as well as other sensor outputs are Timed Data, see Figure 1.

Note that the notion of time may have various applications in Media Orchestration. It may pertain to the *delivery* or on the media itself, or the *presentation.* It may also pertain to the *capture* of media.

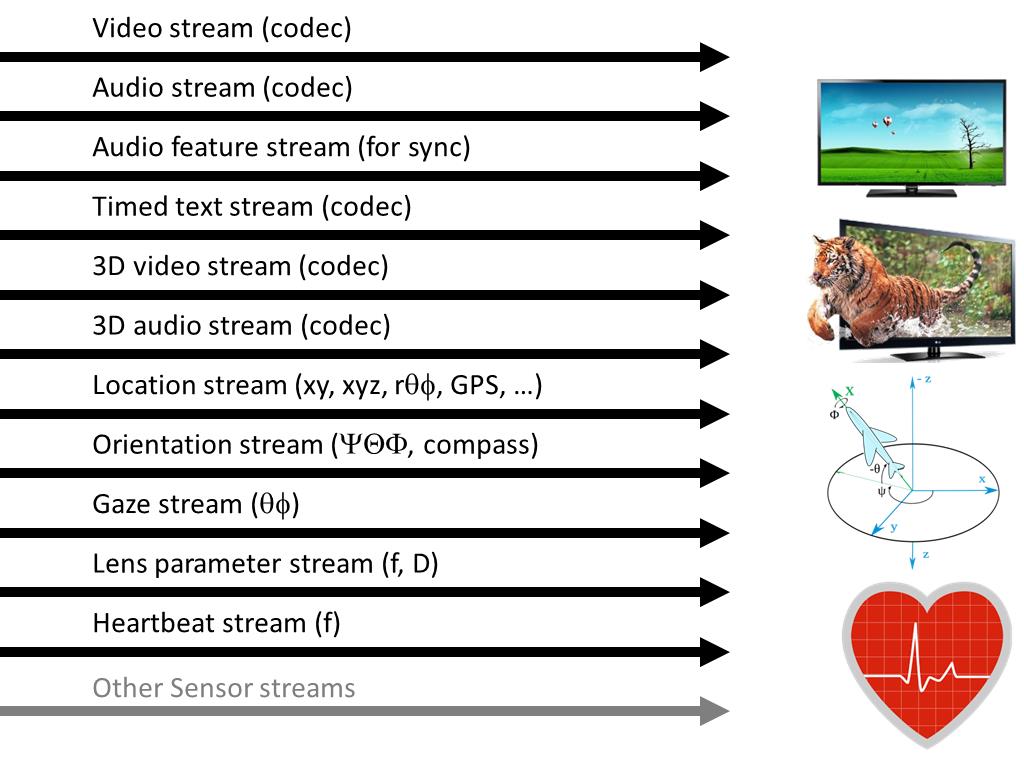


Figure 1 - Examples of Timed Data in the context of media orchestration

## Functional architecture

We distinguish a number of functional entities for Media Orchestration, all derived from the same abstract entity. This document first describes the entities and then gives an example of how they can be configured for an actual use case.

The arrows on the pictures below have the following meaning:

|  |  |
| --- | --- |
|  | Media Data (timed or non-timed, unidirectional) |
|  | Metadata (timed or non-timed, unidirectional) |
|  | Orchestration Data (timed or non-timed, unidirectional) |
|  | Messaging and control (non-timed, possibly bi-directional) |

Table 3 - Types of Data

**Source**

A Source is an element that outputs Media Data and/or Metadata, and may receive control information

Source

**Sink**

A Sink is an element that receives Media Data, and possibly Orchestration Data and/or Metadata, and that presents the Media Data

Sink

**Orchestrator**

An Orchestrator is an element that receives Media Data, Metadata and/or Orchestration Data, and outputs Orchestration Data

Orchestrator

**M-Processor**

An M-Processor is an element that processes Media Data and/or Metadata. It receives Media Data and/or Metadata, and may also receive Orchestration Data, and outputs Media Data or Metadata, or both.

M-processor

**Controller**

A Controller is an element that controls on or more other elements. It has no defined input interface, and outputs Control Data.

*Note: There is an ongoing discussion on whether this entity should be merged with the Orchestrator into a single functional element.*

Controller

### Real-world application of abstract functional roles

The definitions as listed in [53] apply here as well. Note that these are abstract functional roles and functional elements, that may be mixed and matched in real-world applications. For instance:

* A user device may (and very often will) have both Source and Sink functions;
* A user device may also include Orchestrator functions;
* An application server may combine Controller, Orchestrator and M-processing functions;
* M-processing function may include transcoding, adding or changing timelines, multiplexing, demultiplexing, selection (editor), stitching, tiling (e.g., using MPEG DASH Spatial Relationship Description), translation, extracting metadata streams (e.g. CDVS metadata), and more.

### Example instantiation of functional model

An example instantiation of this model is as follows:

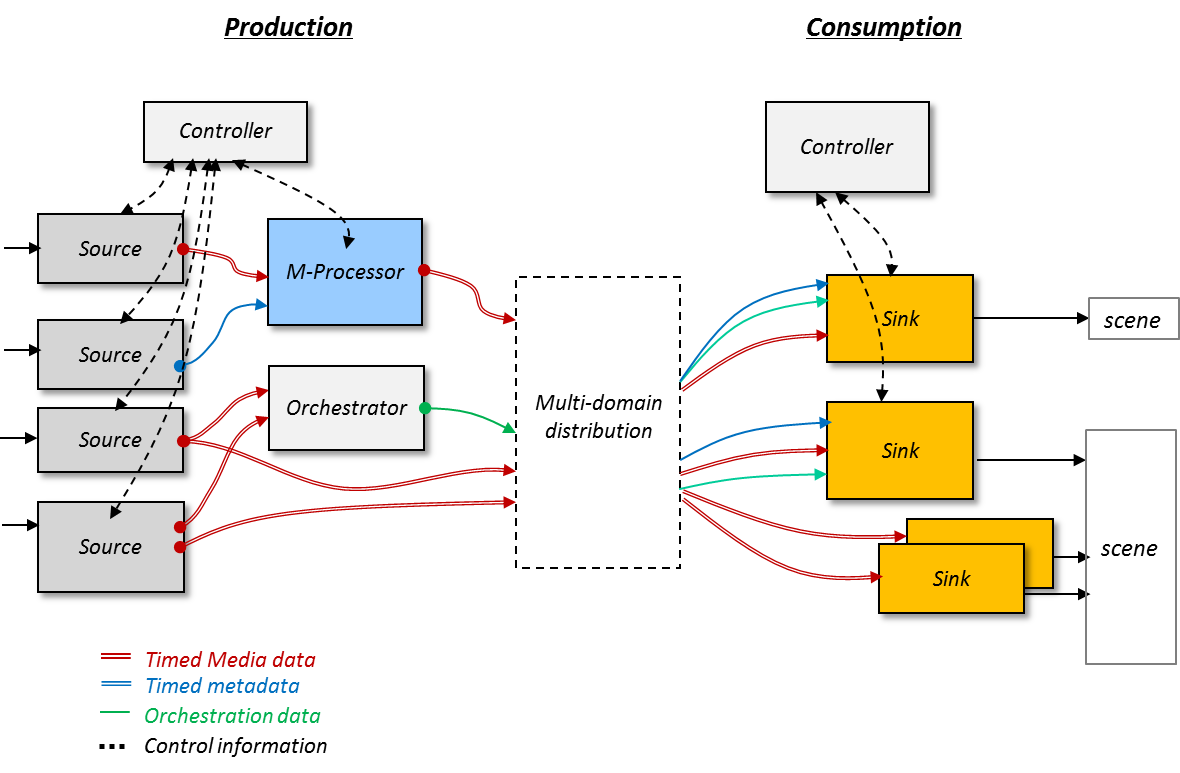


Figure 2 - Examples of Timed Data in the context of media orchestration

This model shows a complex scenario where many sources contribute to an experience with many, orchestrated sinks. (Note that the coding of the lines is not fully consistent with the pictures in above)

##### Orchestration Data

Several approaches can be considered for the arrangement, coordination and management of media and devices using the functional elements – be it by a user or in a (semi) automated fashion. For example, a signalling framework might be created between the functional components. However, because this document includes use cases that are (very) dynamic and variable, we believe a more real-time and fine-grain orchestration approach is required. That’s why we propose the concept of Orchestration Data, to orchestrate devices and media streams at the production side and the consumption side – and often on both sides, especially in real-time use cases where media is consumed as it is being produced.

Orchestration Data can be Timed Data. Orchestration Data are used to orchestrate Media Data and other forms of Timed Data. Like Media Data, an Orchestration Data has an intrinsic timeline. An Orchestration Data may have a start and/or end, possibly controlled by signalling. Orchestration Data may be continuous. Orchestration Data may be unidirectional (e.g. carried in broadcast) or bidirectional. Orchestration Data may need to be multiplexed with other (Timed) Data streams. Orchestration Data may also be stored, as any other form of (Timed) Data.

In summary, the following applies to these streams:

* Media, Metadata are a form of Timed Data. Orchestration Data may also be Timed Data
* Orchestration Data can orchestrate any number of Timed Data streams

##### Examples of Orchestration Data

Several of the timeline-alignment use cases in this document are enabled through the use of media Orchestration Data. The following are examples of Orchestration Data to illustrate the concept in the context of such use cases.

#### Correlation timestamp stream for synchronizing content on multiple devices

The DVB-CSS standard [51] provides a set of protocols to achieve media synchronisation between a television broadcast at a TV device and a companion stream at a companion screen (often called a second screen). That document specifies correlation timestamps which are used to correlate the timelines of the two media streams and so enable their mutual synchronisation. When the clocks of the two media streams are not perfectly genlocked, the correlation timestamps will vary over time, see clause 5.5.5 of [51]. This issue is resolved in DVB-CSS for television broadcast by providing an HTTP-based long-poll mechanism, see clause 7.6.4 of the same document.

DVB-CSS does not provide a solution to provide correlation timestamps, if the two to-be-synchronised media streams are recorded or provided by a content delivery network (CDN). In those cases, the content provider does not know where the user is in the play-out of the media streams and hence it cannot provide the relevant correlation timestamp.

A correlation timestamp *stream* may provide a solution to this problem. This Orchestration Data is a timestamped stream that provides correlation timestamps. Other protocols provided by DVB-CSS can then be used to synchronise the two media streams at the two devices, specifically the DVB-CSS wall-clock protocol (clause 8 of [51]), and the DVB-CSS timeline-synchronisation protocol, see clause 9.

#### Material resolution stream for synchronizing event-based content on multiple devices

The DVB-CSS standard also provides a Material Resolution Service (MRS). The playback device (e.g. a TV or STB) provides a content identifier associated with the current TV program and a URL pointing to the MRS service, see clause 7. The companion screen application queries the MRS with the content identifier and receives material information in response, representing e.g. an interval of a broadcast service during which a given DVB event is signalled; an individual advert, trailer or continuity announcement, or a collection of these; an entire programme or a segment of it; or a segment of content that could be presented by a companion screen application. DVB-CSS assumes that there is a broadcast stream, that content identifiers are provide with the broadcast stream, and that there is an MRS service that resolves the provided content identifiers.

DVB-CSS does not provide a solution to provide material information for unicast or stored streams without content identifiers, or if there is no active MRS service.

A material resolution stream may provide a solution to this problem. This is a type of Orchestration Data that is a timestamped stream of material information. Once synchronised with the original media stream (from broadcast, unicast or storage), the material resolution stream provides the right material information at the right time.

#### Coordinated trick-play stream for smart conferencing and remote learning

W3C is in the process of specifying an HTML timing object [52]. This timing object enables synchronised playout and coordinated trick-play of multiple Timed Data streams. A timing object may connect to an online timing service via the internet, if Timed Data is to be synchronised between multiple devices. The timing object acts as a stopwatch. Users can play, pause, fast-forward and rewind the time at this hypothetical stopwatch, and all sources (media, timed text, timed data, applications) connected to the timing object will follow the stopwatch, and e.g. pause all at the exact same point on the timing object timeline. W3C specifies the API to create, connect and control a timing object.

W3C does not specify any protocols to coordinate timing objects at different devices with a timing service. It mentions a proprietary timing provider object acting as proxy to an online timing resource to enable cross-device synchronization.

A coordinated trick-play stream may provide a solution to this lack. This Orchestration Data is a timestamped stream of positions on the timing object timeline as well as timed instructions to start, pause, accelerate or reverse the timing object timeline.

### Orchestration Rules

[To be provided]

# Required technology support and standardisation

## Required Technologies

A number of technology elements need to be in place for orchestration to work. Some of these benefit from standardization, and some have indeed been standardized already – in MPEG and elsewhere. We first list what may be required, regardless of whether it is already available. The list below is a high-level overview; the document *Requirements for Media Orchestration* [47] contains a longer and more elaborate list of requirements for Media Orchestration.

:

1. Extended synchronization support for temporal orchestration, as listed in [1]:
   1. Multi source content capture: orchestrating a single media experience from multiple independent sources
      1. Discovery and coordination of diverse and dynamic sources
      2. Support for “self-synchronisation”, without a common master clock. (See e.g. [43] for an example of how this can be done.)
   2. Multi-domain content distribution: controlling and harmonizing play-out across different delivery methods on a single device and on multiple devices
   3. Accurately and dynamically controlling play out across multiple devices
      1. Taking into account delay between decoding and presentation through processing delays in notably high-end screens
   4. Synchronization in real-time and also post-hoc.
      1. May rely on good time stamps added by the source; may need to do some work after the fact.
   5. Global Network sync – example is doing encoding at multiple sites for redundancy, while keeping the encoders in sync.
2. Spatial orchestration: dynamically orchestrating media coming from, and played across, multiple devices
   1. Discovery of sensors and their relative locations and direction (gaze) in a 3D environment,
      1. The same, but with dynamic tracking of these coordinates
   2. Discovery of Sinks (play-back devices) and their relative locations in a 3D environment, again including orientation
      1. And again with dynamic tracking
      2. Noting that sources and Sinks could be the same device, with the same position and orientation, but that their “gaze” may differ
   3. Metadata may be added by sources, or it may be required to infer parameters through processing. Such metadata is already present from professional equipment
3. Protocols to enable device calibration
   1. Describing, exchanging and controlling characteristics of sources and Sinks, including
      1. fixed parameters like (non-exhaustive list), resolution, size (for displays), pixels per inch, colour gamut, brightness, contrast, volume, loudness, focus distance, focal length, aperture, etc. Some of these parameters are dynamic .
4. Device network interface characteristics
   1. Network type, in use and available
   2. bandwidth, delay, network location

… catering for

* 1. dynamically changing conditions
  2. Networks that are already available, and ad-hoc networking

1. We may also need to capture, and be able to exchange, user position and orientation
2. Content analysis and its result, content description (the latter of which is the part that needs to be standardized).
   1. Support for efficient search in media, e.g. many takes of the same event
3. Support for dynamically sharing of computational resources for distributed media processing; support for network-based media processing

## Available tools

Several tools for media orchestration (data models, protocols, frameworks, object models, APIs) already available from MPEG and other SDOs:

* MPEG has designed a number of media composition technologies (MPEG-4 BIFS, LASeR, MMT-CI). Such technologies are perfectly suited for complex compositions of media streams
* MPEG transport technologies usually allow for the declaration of associated resources through hyperlinking, addressing part of the media announcement issues. However, all these technologies are used to communicate a list of known locations and will probably not be adequate for media discovery.
* MPEG 3D Audio, encoding and decoding/rendering 3D audio objects independent of loudspeaker configuration [10]
* Video coding (including AVC and HEVC)
  + But make sure we can be codec independent, at least for the contribution part of this, and probably also as much as possible for distribution
  + New visual coding tools may be required to support orchestration
* Support for tiles HEVC
* Spatial Relationship Description in DASH [20]
* MPEG-V [19], elements of which may apply
* MPEG-7, for content description
* MPEG Visual Search
* MPEG TEMI: Delivery of Timeline for External Data [26]
* Device discovery and zero-config networking (DLNA [34], Bonjour [36], mDNS [35]) – but only if on the same local network and only “availability”
* DVB Companion Screens and Streams (CSS) protocols [25]
  + DVB-CSS-DA: discovery of TV devices by Companion Screen Applications (CSA) based on UPnP [27].
  + DVB-CSS-MRS: material resolution, identification and localisation of ancillary content associated to a TV channel, correlation timestamps to correlate different timelines of Timed Data [28].
  + DVB-CSS-CII: identification of the channel to which the TV Device is tuned, bootstrapping of the other DVB-CSS protocols (URLs of the CSS-WC, CSS-TS and CSS-MRS server end points) [28].
  + DVB-CSS-WC: wallclock synchronisation between TV Device and CSA (a quickly-converging NTP variant) [28].
  + DVB-CSS-TS: timeline synchronisation, timestamping of video frames at the TV Device with the wallclock [28].
* HbbTV 2.0 features for coordinating hybrid streams (broadcast+broadband) on an HbbTV Terminal, coordinating applications between HbbTV Terminal and Companion Screen (mutual launching of applications, app-to-app communication), the javascript MediaSynchroniser object and APIs for media synchronisation [29].
* Wallclock synchronisation protocols
  + NTP: network time protocol, application-layer clock synchronisation protocol [30].
  + PTP: precision time protocol, sub-microsecond clock synchronisation with minimal network and local clock computing resources [31].
  + GPS: Global Positioning System, a space-based satellite navigation system that provides location and time information [32].
* W3C
  + WebRTC: direct streaming of RTP stream (audio, video, files) between two browsers [21]
  + W3C Multi-device Timing Community Group. [44]
  + HTML Timing Object, based on the system clock, used for timed coordination of multiple HTML5 objects (e.g. coordinated pause, seek, …) on multiple devices [33].
  + Glass to Glass Internet Ecosystem task force (GGIE), see [45]
* IETF:
  + RTP: Real-time Transport Protocol, streaming of real-time data, such as audio, video or simulation data, over multicast or unicast network services [37].
  + RTCP: RTP Control Protocol, monitoring of the data delivery in a manner scalable to large multicast networks [37].
  + IDMS: Inter-Destination Media Synchronization using RTCP [38]. IDMS Report Block for RTCP [40]. IDMS XR Block SPST Registry [41].
  + RTP clock source signalling: using SDP signalling to identify timestamp reference clock sources and media clock sources in a multimedia session [39].
* EBU runs the project FIMS (Framework for interoperable media services): interfaces/APIs for service oriented architectures for audiovisual production, including capture/ingest, transformation, transfer of content, distributed repositories, quality analysis. Automatic metadata extraction will be addressed next. See [46]
* 3GPP
  + Location and gaze of cameras, as defined in TS 26.244, release 13

## Standardisable aspects and the role of MPEG

MPEG traditionally focusses on encoding/decoding and multiplexing. Specification of the encoding/decoding of new types of Timed Data (location, heartbeat, …) would typically fall within the scope of MPEG standardisation. Similarly, specification of the multiplexing of existing and new types of Timed Data in transport containers (MPEG TS, MPEG DASH, ISOBMFF) is within MPEG scope. Whether Application Programming Interfaces (APIs) for Media Orchestration in devices is in scope, is subject to further discussion. The same holds true for network-network interfaces and APIs for media orchestration, e.g. the control of network-based media processing.

# References

1. Rob Koenen, TNO, Media Orchestration from Capture to Consumption, included in ISO/IEC JTC1 SC29 WG 11 N14881, MPEG110, Strasbourg, FR, October 2014
2. Bubl, http://www.bublcam.com/
3. Oculus Rift, http://www.microsoft.com/microsoft-hololens/en-us
4. Microsoft HoloLens, http://www.microsoft.com/microsoft-hololens/en-us
5. Samsung Gear VR, http://www.samsung.com/global/microsite/gearvr/
6. Google Glass, www.google.com/glass/start/, also see [7]
7. Nick Bilton in New York Times, Why Google Glass broke, http://www.nytimes.com/2015/02/05/style/why-google-glass-broke.html?\_r=0 (also see [6])
8. Stuart Dredge, Facebook closes its $2bn Oculus Rift acquisition. What next?, July 2014 http://www.theguardian.com/technology/2014/jul/22/facebook-oculus-rift-acquisition-virtual-reality
9. Mark Zuckerberg, blog, 25 March 2014, (https://www.facebook.com/zuck/posts/10101319050523971)
10. MPEG, ISO/IEC JTC1/SC29/WG11 N1464423008-3, MPEG 3D Audio, (MPEG-H part 3), July 2014
11. Vine, http://www.vine.co
12. Instagram, http://instagram.com
13. FP7 STEER (Social Telemedia Environment for Experimental Research), http://fp7-steer.eu/
14. HbbTV 2.0, http://hbbtv.org/pages/about\_hbbtv/specification-2.php
15. Samsung Chord, http://developer.samsung.com/galaxy#chord
16. Samsung Group Play, http://developer.samsung.com/group-play
17. MPEG, ISO/IEC JTC1/SC29/WG11 N14644, Exploration of “Uniform Signaling for Timeline Alignment”, MPEG 109, July 2014
18. MPEG, ISO/IEC JTC1/SC29/WG11/N14507, Compact Descriptors for Video Analysis: Objectives, Applications and Use Cases, MPEG 108, April 2014
19. MPEG, Study of CD of ISO/IEC 23005-2, ISO/IEC JTC1/SC29/WG11/N14774
20. MPEG, ISO/IEC 23009-1:201x/FPDAM 2ISO/IEC JTC 1/SC 29 N 14861, Information Technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats / Amd 2, October 2014
21. Web Real-Time Communication http://www.webrtc.org/
22. MPEG, ISO/IEC JTC1/SC29/WG11/N15091, Context and Objectives for Media Orchestration, MPEG 111, May 2015
23. Meerkat, https://meerkatapp.co/
24. Periscope, https://www.periscope.tv/
25. DVB, “Companion Screens and Streams; Part 1: Concepts, roles and overall architecture”, [TS 103 286-1](http://www.etsi.org/deliver/etsi_ts/103200_103299/10328601/01.01.01_60/ts_10328601v010101p.pdf), 2014, http://www.etsi.org/deliver/etsi\_ts/103200\_103299/10328601/01.01.01\_60/ts\_10328601v010101p.pdf
26. MPEG TEMI, “Delivery of Timeline for External Data”, ISO/IEC 13818-1:2014 AMD 2 Delivery of Timeline for External Data, October 2014.
27. DVB, “Companion Screens and Streams; Part 3: Discovery”, TS 103 286-3, 2014, http://www.etsi.org/deliver/etsi\_ts/103200\_103299/10328603/01.01.01\_60/ts\_10328603v010101p.pdf
28. DVB, “Companion Screens and Streams; Part 2: Content Identification and Media Synchronization”, TS 102 286-2, 2014, <http://www.etsi.org/deliver/etsi_ts/103200_103299/10328602/01.01.01_60/ts_10328602v010101p.pdf>
29. HbbTV “HbbTV 2.0 Specification”, 2015, https://www.hbbtv.org/pages/about\_hbbtv/HbbTV\_specification\_2\_0.pdf
30. IETF, “Network Time Protocol Version 4: Protocol and Algorithms Specification”, RFC 5905, 2010, http://tools.ietf.org/html/rfc5905
31. IEEE, “Precision Time Protocol - Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems”, IEEE 1588-2008, 2008, https://standards.ieee.org/findstds/standard/1588-2008.html
32. GPS, “Global Positioning System”,http://en.wikipedia.org/wiki/Global\_Positioning\_System
33. W3C, “HTMLTimingObject”, draft specification, 2015, http://webtiming.github.io/timingobject/.
34. DLNA: Digital Living Network Alliance, http://en.wikipedia.org/wiki/Digital\_Living\_Network\_Alliance
35. mDNS: Multicast Domain Name System, http://en.wikipedia.org/wiki/Multicast\_DNS
36. Bonjour: Apple's implementation of Zero-configuration networking, http://en.wikipedia.org/wiki/Bonjour\_(software)
37. IETF: “RTP: A Transport Protocol for Real-Time Applications”, RFC 3550, 2003, https://www.ietf.org/rfc/rfc3550.txt
38. IETF: “Inter-Destination Media Synchronization (IDMS) Using the RTP Control Protocol (RTCP)”, RFC 7272, 2014, https://tools.ietf.org/html/rfc7272
39. IETF: “RTP Clock Source Signalling”, RFC 7273, 2014, https://tools.ietf.org/html/rfc7273
40. IANA: “Block Type (BT) 12, IDMS Report Block”, IANA XR Block Type Register, http://www.iana.org/assignments/rtcp-xr-block-types/rtcp-xr-block-types.xhtml#rtcp-xr-block-types-1
41. IANA: “IDMS XR Block SPST Registry”, http://www.iana.org/assignments/rtcp-xr-block-types/rtcp-xr-block-types.xhtml#idms-spst
42. Hbb-NEXT project, <http://www.hbb-next.eu/>
43. JRS at Event Synchronisation Task, poster at MediaEval2014, <http://icosole.eu/wp-content/uploads/2014/11/digpos312014MediaEvalBailerWeb.pdf>
44. W3C Device Timing Community Group, <https://lists.w3.org/Archives/Public/public-webtiming/2015May/0000.html>
45. W3C Glass to Glass Internet Ecosystem task force (GGIE), <https://www.w3.org/2011/webtv/wiki/GGIE_TF>
46. EBU project FIMS (Framework for interoperable media services), <http://fims.tv/>
47. MPEG, ISO/IEC JTC1/SC29/WG11 N15343, Requirements for Media Orchestration, MPEG 112, June 2015
48. 3GPP, 3GPP TS 26.244 V13.1.0 (2015-06), 3GPP file format (3GP), Release 13, www.3gpp.org/ftp/tsg\_sa/WG4\_CODEC/Specs\_update\_after\_SA68/26244-d10.zip, June 2015
49. FP7 ICoSOLE project, [http://icosole.eu/](http://icosole.eu/public-deliverables/).
50. Cameraad, http://www.persinnovatie.nl/17283/nl/cameraad-breaking-video (In Dutch)
51. DVB, “Companion Screens and Streams; Part 2: Content Identification and Media Synchronisation”, DVB A167-2, ETSI TS 10328602, <http://www.etsi.org/deliver/etsi_ts/103200_103299/10328602/>, July 2014
52. W3C, “Timing Object”, work in progress, see <http://webtiming.github.io/timingobject/>, downloaded September 2015.
53. MPEG, ISO/IEC JTC1/SC29/WG11 N15735, Requirements for Media Orchestration, MPEG 113, October 2015

# Annex A : Media Orchestration Use Cases

There are several use cases in which it may be interesting for a service provider to deliver part of its content on one transport channel, for example free-to-air terrestrial or satellite broadcast, and part of it on another such as broadband IP connection. In an attempt to catalogue a wide range of scenarios, the sections below provide an exhaustive list of use cases. Each use case is described in terms of number of sources, format of the content, the delivery network used and the accuracy of the synchronization.

## Synchronized content on the same playback device

### Ancillary audio and subtitle stream content that requires synching with main program

An ancillary audio track is being played on another device which needs to synchronize with a broadcast stream. An example is sports commentary in obscure languages needs to be synched with actual game being played. In addition, an ancillary subtitle stream delivered over IP need to be displayed and synched on the main TV set.

### Ancillary video stream containing sign language

In the context of Hybrid Broadcast Broadband TV (HbbTV 2.0), TV viewers can experience regular broadcast programs enhanced by ancillary content. For instance as depicted in the picture below, the TV displays a sign language video content in a PIP fashion that enhances the main audio/video content. In this scenario the broadcast is a DVB stream while the ancillary stream is MPEG-DASH content. Here the ancillary video has to be synchronized with the primary audio/video stream. The synchronization requires being frame-accurate in order to provide the expected experience. Indeed TV viewers will be very sensitive to out-of-sync effect just like in case of lip sync issues. Synchronicity requirements for sign language interpretation are less severe.

HbbTV incorporated MPEG-DASH in HbbTV specification 1.5 and sync features in version 2.0.



Figure 3 - FP7 Hbb-next project’s use case [42]

In both cases, measurement and buffering in the TV, or close to it, are needed to achieve synchronous playout.

### Multiple DASH streams that need synchronization

A wedding is happening in Timbuktu. Some guests (grandmas and grandpas, cousins, and etc.) are unable to attend in person. They should all be able to watch the wedding in real time with properly synched audio. This use case assumes that audio and video come from different sources/devices

### Multiple DASH streams + SAP that need synchronization

Same as above, plus: the younger generation lives in America and can no longer speak Swahili so a professional interpreter was hired to provide live translation.

### Scalable/multiview enhancements

In this use case, the broadcast video is enhanced by a temporal, spatial or multiview scalable layer on a different delivery channel to achieve either high frame rate, high resolution or 3D experience. The broadcast content may be temporarily paused to buffer the enhancement layer. The synchronization has to be frame accurate with no error tolerance to allow for proper decoding of the enhancements.

## Synchronized content on multiple devices

### Second screen

The audio synchronization technology allows “second screen” applications where the 2nd screen content is synchronized to the 1st screen content. In this scenario, no common clock covering the 1st and 2nd screen devices is required, nor a way to exchange time-stamps between the devices. Synchronization of the contents between the devices is carried out by using audio feature (fingerprint) stream extracted from the 1st screen content. In this example, 1st screen content is distributed over existing broadcast system, and 2nd screen content is distributed over IP network, where 1st screen content is a regular TV program of car racing, and 2nd screen content is a driver’s view of the car racing.

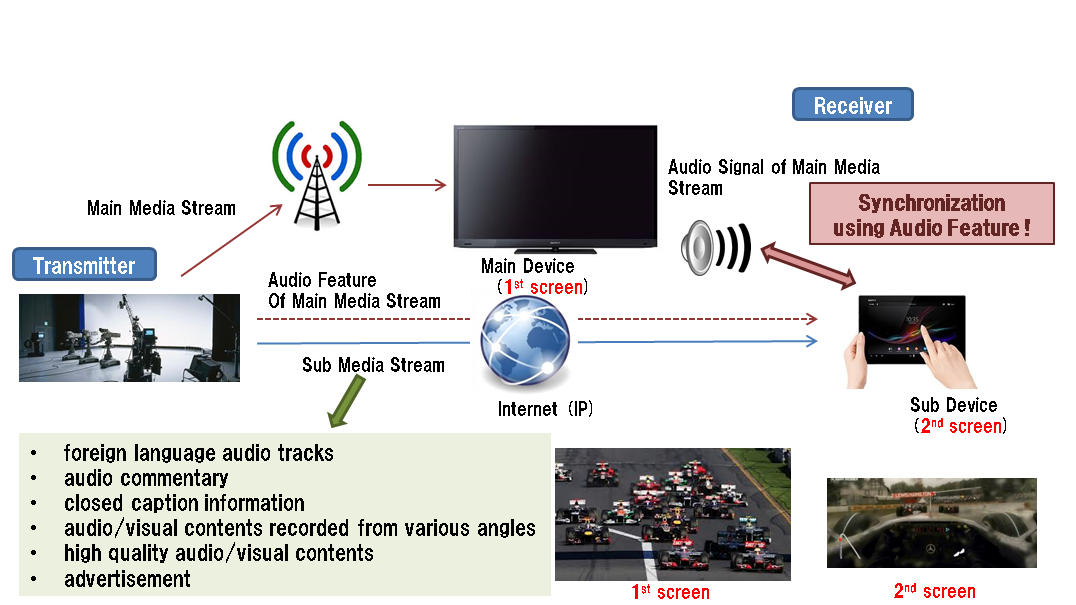


Figure 4 - Overview of a fingerprinting-based sync mechanism for 2nd screen application

### Second Screen Content List Synchronized to a TV Program

This use case has been investigated in the FP7 BRIDGET project[[1]](#footnote-1). The user is watching a broadcast program on TV and, at the same time, a second screen application running on his smartphone or tablet, presents a list of additional program-related on-demand content (informative content, director’s cut clips, advertisements, etc…). The list of contents available is dynamically updated taking into account the current position in the timeline of the broadcast program. The user can browse the list, select a content and play it on his second screen. Content playback on the second screen is independent from the timeline of the TV program (i.e. does not require synchronization of the timelines). In this use case the additional content is delivered using MPEG-DASH or HTTP streaming of ISOBMFF.

### Ancillary audio stream containing audio that requires synching

An ancillary audio track is being played on another device which needs to be synchronized with a broadcast stream. An example is sports commentary in obscure languages needs to be synched with actual game being played.

### Ancillary audio and subtitle stream content that requires synching with main program at different location

A video broadcast from one content provider is played on a main TV. DASH stream of same program is played on a tablet somewhere else has to be synchronized with main program. In addition, an audio stream delivered via IP needs to be synched and played back on an IP connected audio player at the gym.

### Continue watching on another device

It is envisioned that TV viewers can watch the same content on a variety of devices. For instance people watch a certain program on their mobile phone, and when coming home, they can continue watching it on their home television. Or, you are watching something on your television, and you need to leave your couch (for whatever reason), without wanting to miss out on the action. So, you continue watching on your tablet. The use case is quite similar when multiple devices are presenting the same content in the same physical location, e.g. multiple TV sets tuned to the same program in the same house. This requires highly accurate synchronization, as the audio will only remain good if perfectly aligned between various presentations.

### Mobile audio listening of TV program

In the context of Broadcast TV, viewers may want to follow the audio on their smartphone headset, giving the possibility to freely move in the house. In this scenario the broadcast is, e.g., a DVB stream while the ancillary stream is RTP or DASH-based audio content. Here the ancillary audio has to be synchronized with the audio visual stream from the main TV content. The synchronization requires lip-sync accuracy in order to provide the expected experience. Audio will be very sensitive to out-of-sync effect just like in case of lip sync issues.

### Audio description for the hard-of-sight

Audio description of on-screen events is a service already provided by some digital television broadcasters. (E.g. BBC[[2]](#footnote-2)). It provides an additional audio track that it played concurrently with the original audio, although it is mostly like premixed and then just offered as a completely separate, alternate audio track.

A group is watching a standard TV broadcast. John requires audio description but the others do not want to hear it. Using, e.g., audio fingerprinting, the extra/alternative audio may be streamed via John’s mobile phone / tablet and he may listen on headphones.

*Editor’s Note: Audio fingerprinting is a solution – this context should focus on the use case functionality.*

### Modular screen systems

A company offers extremely large screens to be formed by combining several smaller ones. A single video stream can be split over all screens, possibly even utilizing HEVC tiling.

### Subtitling or alternate language based on audio fingerprint

A company provides a database of subtitles and alternate language audio for published content that the original content producers did not provide themselves. Timing information is provided (e.g., by an audio fingerprint), and subtitles/alternate language audio is fetched over IP and presented on the same device.

### Smart conferencing (capture and rendering)

Six users in two groups of three are having a video conference session together. They all use their tablets for both capturing and rendering. They put their tablets side-by-side to build a kind of video wall, and this also creates proper audio output. Each user is shown in the proper position, the user on the left on the left screen, the user in the middle user on the middle screen, etc. The audio will appear to be coming from the screen showing the person speaking. There is also some tracking involved, as people don’t sit still, so the audio and video capture follow the person speaking. Alternatively, smart conferencing systems might help in increasing the interaction in e.g. unidirectional internal broadcasts.

### Additional information with Synchronization

A user is watching content on her TV set. She uses her tablet device to interact with the content, e.g. to receive more information on an ad that is currently being shown on the TV. The user watches the content on multiple devices simultaneously and enjoys the accurately synchronized audio in a multi-channel setup.

### SocialTV use case

Social networks have become pervasive and platforms like Facebook, Twitter, and Google+ provide new ways of social interaction. The idea of experiencing multimedia content together while being geographically distributed and, additionally, to communicate with each other via text, voice, or video telephony is a step towards making, e.g., watching television (TV) an online social event. For example, two (or more) geographically distributed users are watching the same live event (e.g., football) in different contexts (i.e., networks, devices, platforms) while at the same time having a real-time communication (e.g., text, voice, video) among the participating users. In such situations, the playout of each Sink could be out-of-sync and one user may report a vital event (e.g., goal, touchdown) while other users are still waiting for this event to happen. Hence, the media playout needs to be synchronized among the participating Sinks.

## Combining many sources of user-generated content into a single experience

### Classroom recording (capture)

Multiple users each record a teaching session in a classroom, and the various recordings are used to enhance the result. A classroom can be a noisy environment, and various audio recordings are used to clean up the audio of the teacher speaking, and of students asking questions. Various video recordings are used together to recreate the classroom, showing both the teacher, the presentation and classroom with students asking questions. This can be done manually or human-assisted. Note that the devices record independently, without first communicating with one another.

### Collaborative storytelling

A group of users are enjoying their holiday together. They all use their smartphone and camera to capture parts of it, and all recordings are automatically uploaded and combined. The result is an interactive, audiovisual story of their holiday, to be shared with friends and family. Viewers can browse through space and time to view and listen to the various happenings during the holiday, and multiple views of the same event are offered whenever available.

### Crowd journalism

This use case is described in [18]. It involves end-users contributing to news broadcasts, using their smartphones. Similar to upcoming live mobile video streaming applications such as Periscope [24], Meerkat [23]. Note that from a service provider and/or broadcaster perspective, it may be important to recognize, kill and prevent unsolicited and illegitimate live streams from high-profile events.

Editors can be given the possibility to communicate with their journalists using text and audio, to give them directions on where to go and what to capture. Editors will also have the possibility to understand where their “reporters” are, and optionally which way they are facing. An example of such a system is Cameraad [50].

Lastly, editors may be given the option to control source settings, such as zoom, aperture (to control exposure and depth of field) and framerate (to control frame rate/quality trade-offs).

### Synchronized user generated content

During live events, such as big concerts or big sports events, many people make their own recordings of what they find interesting. Nowadays, users share this online with others to be watched separately. But, such recordings can also be synchronized with a main broadcast, e.g. to be viewed picture-in-picture on the TV or separately on a secondary screen. This can be done live during the actual broadcast, which requires the user to stream its recording directly onto the network. It can also be done for later on-demand viewing, which requires the user to upload the recording. High frame synchronization accuracy may not be required as the recording is displayed separately. Typically synchronization accuracy should be within a few hundred milliseconds.

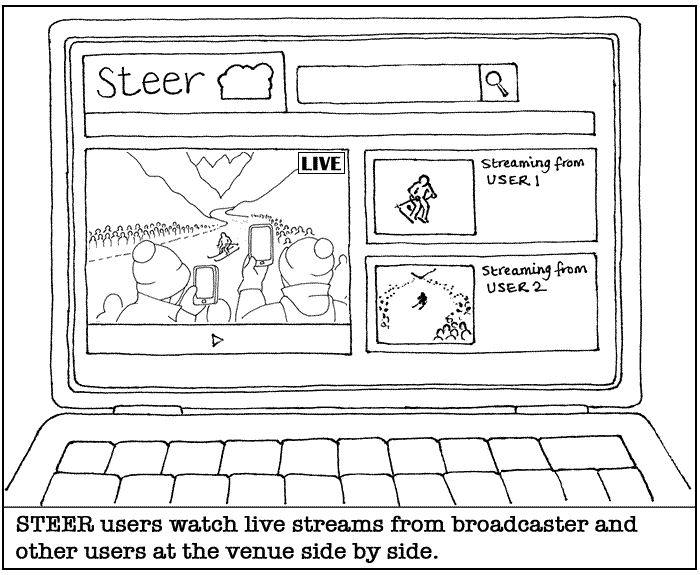


Figure 5 - Use case from FP7 STEER project [13]

### Surveillance (capture and rendering)

In case of an incident, emergency services use available street cameras to get a view of the situation. But there may be people present with smartphone that have a better viewpoint. Such bystanders, who have opted-in to help out in emergency situations, receive an invite to start recording to enhance the view captured by the fixed cameras. When they start recording, they receive instructions from the central control room for where to point their camera to fill in gaps in the video image. Such instructions can be through voice, or through some protocol that directs the recording device/person. First responders can get a complete picture of the situation, and they are able to zoom in on their tablets to the parts they need to analyze.

*Discussion:*

Bystanders are often directed out of the way in emergency situations. This may make this use case improbable and impractical. On the other hand, emergency workers also have devices with sensors, sometimes even body cams.

### Recording for 3D and VR

A number of users make a recording together, with the purpose of displaying this later in 3D on a virtual reality headset. They have a capture cardboard in which they place their smartphones. With 6 smartphones, they are able to capture a rough, 360 degree video. All smartphones have their recording app connect with the master app running on a tablet. One user uses the tablet to start and stop the capture, and the application on the tablet automatically calibrates the various smartphones for a proper VR capture.

*Discussion*

VR implies 3D; with the proposed set-up it will not be possible to capture 3D images with depth information.

### Video Street View for an Event

A lot of people visiting an event, e.g. a carnival/festival, use their smartphones to create recordings. Furthermore, the event has own cameras to create other recordings, on the stage, on the event objects, performer, on drones, etc. Their recorded video are uploaded to the cloud, and they are used to create/playback video street view. When a certain time and a certain position in the event are specified, video corresponding to the time and position starts to be played back. The specified positions are changed through the playback period, and the videos are changed. In order to playback videos continuously, i.e. with no-gap, during the playback period, videos near the specified positions can be selected to be played back as well as the ones on the strict positions. The videos near the positions can be also processed, e.g. zooming, panning, synthesis, in order to be more seamlessly.



Brazil07 by Venturist

Figure 6 - Video Street View for an Event

---

Figure 7 below illustrates the difference of the playback of VSVfE between when video playback near the specified position is permitted and when it is not permitted. In the former case, the selected videos are played back continuously (2-(b)). On the other hand, in the latter case, discrete videos are selected and some gaps appear in the playback period (2-(c)).

**A**

**B**

**C**

**D**

shooting position

of video A

shooting position

of video B

shooting position

of video C

shooting position

of video D

designation (track)

(a) designated position track and videos

**A**

**C**

**D**

**B**

(b) video near (non-strict) the position is permitted

**A**

**C**

(gap)

(gap)

Figure 7 - Playback Sequence examples of VSV

## Combining Professional and User –generated Content

### Concert recording (capture)

A user at a concert makes a recording of the concert for personal use, to be able to show that ‘they were there’. By determining the location of the user and the direction of the camera, it is possible to create footage for the user based on the professional recordings made at the concert. Users receive a personal recording from their own viewpoint, but with the quality of the professional recording.

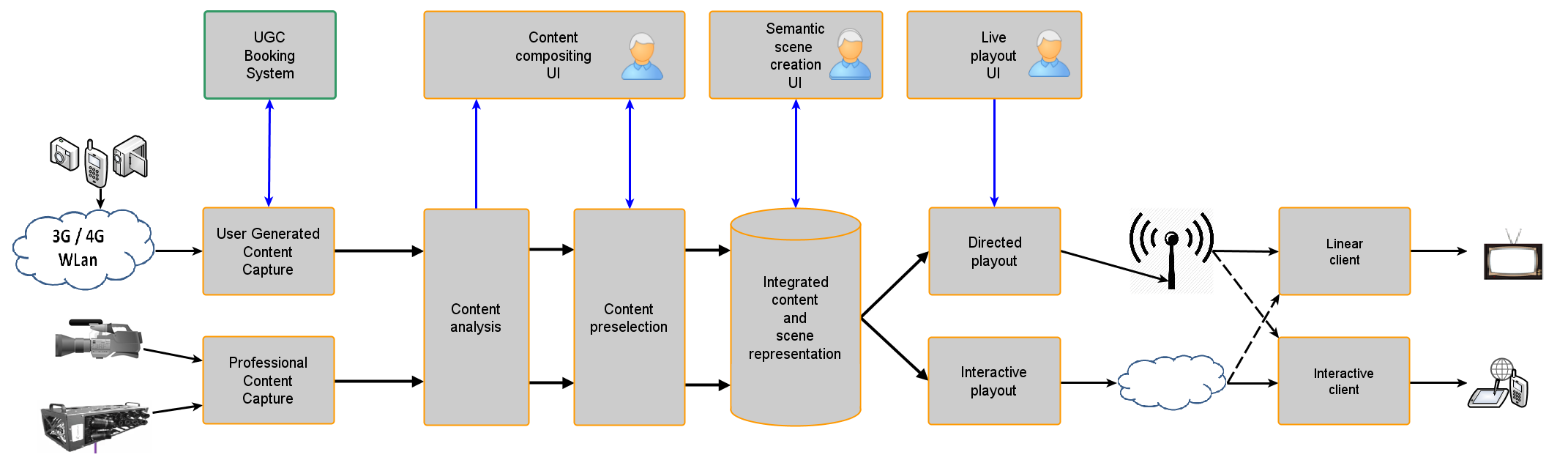
### Media Stitching

On the location of a concert/festival, there are a number of video feeds, both pro (good quality, reliable, continuous) and amateur (any quality, unreliable, on and off randomly), as well as picture contributions (of all qualities) to the capture scene. A (e.g. distant) user has a means to browse the capture scene and select a point of view. The system provides a best effort media with a point of view as close as possible to the selected point of view. The chosen point of view may:

* correspond to an existing point of view, so the result is directly the corresponding video feed
* correspond to a synthetic point of view that can be constructed by stitching available video feeds
* correspond to a synthetic point of view only partially covered by video feeds, in which case a combination of photos and videos are used to construct the synthetic point of view; over time, the selection of feeds used to construct the current choice of point of view may vary.

### Immersive Coverage of Spatially Outspread Live Events (ICoSOLE)

ICoSOLE aims at supporting use cases that enables users to experience live events which are spatially spread out, such as festivals (e.g. Gentse feesten in Belgium, Glastonbury in the UK), parades, marathons or bike races, in an immersive way by combining high-quality spatial video and audio and user generated content. Therefore, a platform shall provide support for a context-adapted hybrid broadcast-Internet service, providing efficient tools for capture, production and distribution of audiovisual content captured by a heterogeneous set of devices (professional and user-generate content) spread over the event site. An overview of such a generic example system is shown in the figure below where content is coming both from professional content capture devices and user-generated content capture devices. After the content is being analyzed and preselected – possibly supported by an editor through an appropriate user interface (or done automatically) – the content is integrated using an appropriate scene representation and prepared for dissemination (live and on-demand enabling interactivity) to a plethora of heterogeneous devices.



In particular, the capturing app (incl. Moments app) allows for audio-visual content capturing including various metadata (e.g., location, orientation, A/V recording parameters, sensor information, manual annotations by the user, user profile data incl. psychophysiological data, obviously subject to privacy precautions) and live streaming or (later) uploading to a central storage and processing unit on site (typically in an outside broadcast van). These user-generated media assets may be referred to as ‘Moments’.

On the consumption side, a feature called ‘The Wall of Moments’ allows a remote user to have a unique festival experience by using ‘Moments’ captured by peers attending the live event. The remote user pitches in to the live event via a Moment and can enjoy a high quality A/V stream from there on, as Moments are precisely synchronized with the professional production feed. In this way, the user still has an immersive experience by having direct social contact with the crowd (more importantly, his/her friends/peers). The Wall of Moments could be realized as a mosaic representing the most popular moments users shared (e.g., via Facebook, Twitter), moments the production highlighted, live production feeds and a virtual map. It acts as an online interactive portal to the live event for users not attending the event, yet providing an immersive experience bringing them closer to the event. Various interaction possibilities for professional editors and users of the system can be envisioned for this use case.

An alternative mode of presentation is the interactive exploration of a ultra-high resolution or panoramic video. In this case spatial audio for the selected viewpoint will be rendered based on multiple input sources, requiring temporal synchronization precision well beyond the video frame rate.

More details available in D2.1 Usage Scenarios (December 2013) and D2.2 Use Cases & Requirements (July 2014) of the ICoSOLE project which are available here: <http://icosole.eu/public-deliverables/>.

### Drone camera recording for multiple view and angles

A group of users are enjoying same event (e.g., skiing) together. Some of the users can make a video recording in flight view by using their drone cameras. Each drone camera can track its user and capture cinematic footage of them. As the user moves, the drone camera follows the user while recording. Some of users can make a video recording by using their smart phones in their own view. All recordings are combined to recreate a single video that can provide multiple views or different angles of the same event.

### Enjoying an event at home

A user is enjoying a large sports event broadcast in a home environment. The big screen shows the main coverage by the broadcaster, and wireless speakers throughout the house provide a great audio experience. As the user goes to fetch drinks in the kitchen, he takes his tablet with him to continue watching the event. Meanwhile, the audio is adjusted so the user remains immersed while moving about. When the audio levels of the public increase, and a smartwatch starts to vibrate to indicate an upcoming highlight, the user rushes back. When back on his couch, the devices (tablet, phone) lying around go back into “ambi-mode”, providing a visual “surround-image” with the content back on the main screen.

### Audio/ Video selection on User Demand

The user watches a Premier League game on his large screen TV. He can watch the common video which is edited from BBC channel. Also he can subscribe the YouTube channel which is provided by the separate provider such as an UGC or another camera in the stadium. He is a big fan of Chelsea team. So he wants to watch the other view of the content related with his team. The content would be shown in second screen with audio, video synchronization.

### Enriched broadcast (capture and rendering)

People visiting an event use their smartphones to create recordings. When the event is broadcasted, these user recordings are used to augment the “official” broadcast. For viewers watching the event at home, some of the user recordings are depicted on the main screen, as they are related directly to something shown in the main broadcast. Users at home can also use their tablet to “look around” at the event venue, where the content they see on their tablet depends on the position of their tablet. In other words, they use their tablet to browse the available sources. They can receive alerts in which area event visitors are making recordings, and can view/hear these recordings by moving their tablet around (i.e. “surround vision”). The viewers will also hear the audio that goes with the area viewed on their tablet.

*Discussion:* the directing and editing part can (partially) move to end-users, so they can decide what to see.

## Audiovisual orchestration at the Sink

### Modular speaker systems

Consider the case of network-attached speakers. Currently, there are several proprietary solutions (Bose, Sonos, etc.) offering synchronized audio streaming from multiple speaker units which are each attached to home network. With the current focus on 3D audio it is not unreasonable to expect this to be used by content creators and delivered over a streaming service (e.g. Netflix) in the near future. Modular speaker systems could be used in the home to create an arbitrary and extensible speaker setup in order to for the user to consume this new brand of media. An open standard would allow speakers from different manufacturers to interact seamlessly, meaning that upgrades or replacements are possible at any time. The timing of each module would have to be sample accurate in order to avoid phase-cancelation artifacts.

**Case 1**: John buys an internet TV set which contains 2 speakers, on which he streams movies. Next month he buys 3 more speakers which also connect to his home network, and arranges them to create a 5.1 setup. The next month he buys 4 more speakers and installs them to create a 9.1 3D sound stage. Since each speaker is independent, he may also rearrange them as he wishes, for example, a party requiring a stereo pair in several different rooms.

**Case 2:** A company hires out professional audio equipment for events and hence desires the flexibility to create a custom speaker setup in any space. Wirelessly connected speakers eliminate the need to attach and then secure huge lengths of audio cables.

### Joining multiple speakers in an ad-hoc fashion to create a single speaker system

A group of users are together, and wish to play some music. They all brought their mobile devices, which they take out and place throughout the room. One user selects and start the music from a tablet, using the speakers of the various devices to create a rich sound, filling the room. The playout is controlled in such a way that the proper stereo or surround sound is reproduced, as intended by the artist. This includes adapting to specific device locations and orientation and controlling volume and equalizer settings. This is essentially the same use case as use case 15: Modular Speaker Systems in [17].

Another example is a user watching a program on his large screen TV, and using his smart phone as an additional speaker as an additional source of sound, for instance to make the sound more spatial. More devices can be added to add to the experience.

### Dynamic sound Orchestration

A variation on the previous use case is Kate, who is listening to music through her wireless headset whose audio source is fed from her smart phone during walking. She gets in her car and starts to drive the car. As she takes off her headset and the car speaker system recognizes that her smart phone is now playing the music through the smart phone’s speaker. The car speaker system automatically recognize the position and the orientation of her smart phone’s speaker and provides newly added sound effects to the playing music. Thus, the surrounded sound can be reproduced by smart phone speakers and car speakers. When she leaves the car, she keeps listening to the music through her headset with her smart phone in only stereo. The added dimension of this use case is that speakers can be dynamically added or taken away from the experience.

### Joining multiple screens in an ad-hoc fashion to create a single large screen

A group of users are together, and one user wishes to share some photos and videos with other users. The users put their devices on the table, side by side, thereby forming a larger screen. The photos and videos are shown on this larger screen, taking into account the specifics of the various devices. To make the content look good, the screen settings are harmonized across the devices and the images are adjusted for differences in screen size between the devices and for bezel thickness.

## Augmented scenes

### Augmented Reality over Digital Signage

Kate wearing Augmented Reality(AR) glasses visits New York City’s Times Square. While walking, she sees a large digital signage screen displaying animated advertisements. When she looks at the digital billboard displaying the advertisement for the musical “Wicked”, she can see the Wicked trailer automatically superimposed over the digital signage through the AR glasses. Moreover, she can hear one of the original sound tracks for this musical personally via earphones. While others without any AR glasses only see the advertisement displayed on the digital signage, she can enjoy the enriched experience with additional advertisement content through the AR glasses.

### Augmented Broadcast Service

Consider the case of smart phones that have one or more augmented reality apps. Even though these augmented reality apps working on the smart phone are not widely deployed yet, they are expected to flourish in near future. A user can have AR experience in daily life when the user has the smart phone installed with augmented reality apps and with one or more cameras built into it. The user also can have similar AR experiences with AR glasses. For example, the user can enjoy additional content over broadcast program in watching TV at home as described in examples below. The temporal and spatial synchronization between the main TV content and augmented smart phone content are required in order to provide the enhanced experience.

*Case 1*: Kate is watching a cooking show on TV. The TV screen shows the scene that a cook introduces food ingredients. Kate wants to know further details of one of the food ingredients and looks at the video displayed on the TV screen through the smart phone (using its built-in camera). The smart phone captures the TV scene and the detailed description of this ingredient (e.g., name, kind) is automatically superimposed nearby the ingredient on the main TV video. Kate can see both of the main TV content and the augmented content (e.g., the description of the ingredients) simultaneously. During watching the main TV content with the smart phone, the position of the augmented description superimposed over the ingredient is updated according to the changed position of the ingredient on the TV scene.

*Case 2*: Kate is watching a televised car race that is augmented with driver’s view recordings. She discovers that an interactive feature is available and activates an augmented reality app in her smart phone. She chooses the “Favorite Driver View” mode that provides the selected driver’s view. She sees the TV screen through the smart phone and clicks one of race cars in the scene. She can view the selected driver’s view recording augmented over the car in the TV scene (not necessarily covering the entire TV screen area). She will also view the selected driver’s current speed, RPM, and position in the race. The selected driver’s view disappears whenever the car is not displayed in the main TV scene.

*Case 3:* consider the case of the smart phone that can additionally provide tactile or haptic feedbacks. In the above case 2, when a selected driver’s car is placed at risk to bump against other race cars, a smart phone starts to vibrate to indicate this risk. As another case, Kate is watching a TV shopping channel demonstrating a skirt. She discovers an interactive feature is available and activates an augmented reality app in her smart phone. When she views the main TV video through her smart phone, the notification that tactile effects are available is displayed on the smart phone. Then, she can feel tactile texture effects of the skirt through the touch display of the smart phone.

## Timeline Use Cases

### Interactive TV

A TV broadcast offers user participation by streaming an application to a phone/tablet. It might be beneficial to only make some functionality in the app available for a limited time, e.g. voting for a contestant on a live talent show. The orchestration aspect is to make align programmatic behaviour with timelines in Media Streams

### Interactive navigation in live broadcast

In this use case, the user wants to browse the past hour(s) of a broadcast football game. The broadcast was not locally recorded, and the user wants to fast-rewind the action he just missed. The terminal has to reconstruct a frame-accurate timeline and request the content from an OTT source.

Note: It is not clear which elements in this use case are not currently supported in MPEG technologies, and why this constitutes an example of media Orchestration, or Timeline Alignment. Unless this is made clear, the use case will be deleted.

## Other Use Cases

### Use Case: Teleconferencing over Different Divisions

A teleconferencing event involves a number of divisions of users.  Users join and leave the event based on whether their division leader joins and leaves. The relationships that need to be orchestrated are:

* 1. the membership between audio/visual/chat components of users and their divisions,
  2. the subset relationship between audio/visual/chat components of the each division and the entire pool of users, and
  3. the hierarchical relationship between audio/visual/chat components of users and their division leaders.

Note that content components associated with collection of divisions of employees can be recast in different contexts, such as collection of families, social network groups, buildings with cameras, mosaics of channels and groups of window objects.

1. <http://ict-bridget.eu/> [↑](#footnote-ref-1)
2. <http://www.bbc.co.uk/iplayer/tv/categories/audiodescribed> [↑](#footnote-ref-2)