Traction
Opera co-creation for a social transformation

D2.1: Technical Requirements, Architecture and Integration

DISSEMINATION LEVEL	PUBLIC (PU)
WORK PACKAGE	WP2 TOOLSET DESIGN AND DEVELOPMENT
DELIVERABLE NUMBER	D2.1
VERSION	V3.0
SUBMISSION DATE	01/05/2020
DUE DATE	30/04/2020

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 870610

Authors

Authors in alphabetical order		
Name	**Organisation**	**Email**
Pablo Cesar	CWI	p.s.cesar@cwi.nl
Jie Li	CWI	Jie.Li@cwi.nl
Thomas Röggla	CWI	T.Roggla@cwi.nl
Alina Striner	CWI	Alina.Striner@cwi.nl
Anderson Simiscuka	DCU	anderson.simiscuka2@mail.dcu.ie
François Matarasso	FM	matarasso@me.com
Ana Dominguez	VICOM	adominguez@vicomtech.org
Stefano Masneri	VICOM	smasneri@vicomtech.org
Iñigo Tamayo	VICOM	itamayo@vicomtech.org
Mikel Zorrilla	VICOM	mzorrilla@vicomtech.org
Camille Donegan	VRI	camille@vr.ie
Terry Madigan	VRI	terry@vr.ie

Control sheet

Version history			
Version	**Date**	**Modified by**	**Summary of changes**
0.1	17/02/2020	Pablo Cesar	First ToC
0.2	28/02/2020	Contributors	Initial set of inputs
0.5	30/03/2020	Pablo Cesar	Consolidated version with initial inputs
0.6	06/04/2020	Contributors	Second round of inputs
1.0	11/04/2020	Pablo Cesar	Consolidated draft version with all the inputs on related work and requirements
1.5	24/04/2020	Contributors	Architecture contributions
2.0	27/04/2020	Pablo Cesar	Final version with all the inputs
2.5	30/04/2020	Pablo Cesar	Final version ready for review
3.0	01/05/2020	Pablo Cesar	Final version after review

Peer review		
	Reviewer name	**Date**
Reviewer 1	Dorleta García Melero	30/04/2020
Reviewer 2	Esther Novo	30/04/2020

Legal disclaimer

Table of content

List of figures

Abbreviations

Abbreviation	Definition
DoA	Description of Action
EC	European Commission
GA	General Assembly
WP	Work Package
WPL	Work Package Leader
VPaaS	Video Platform as a Service
REST	Representational State Transfer
API	Application Programming Interface
UX	User Experience
UI	User Interface
FG	Focus Group
DoF	Degrees of freedom

Executive summary

This deliverable provides the reader an overview of the activities of WP2, toolset design and development, focusing on the technical requirements, architecture and integration. It is an iterative deliverable, which will be updated as the project progresses and the toolset is further developed and integrated. In particular, the objective of the deliverable is to detail the process of gathering requirements and to justify design and implementation decisions, for later on deployment of the toolset for evaluation. WP2, the technology, works in close collaboration with WP3, the trials, and WP4, the user experience evaluation.

The deliverable is divided into three main sections.

Section 2 overviews the related technologies and infrastructure. It provides a comprehensive report on technologies and infrastructural components that have been considered by the consortium as alternatives for the development and deployment of the toolset. These technologies are divided into the three main components that have been identified during the requirements gathering phase: media vault, performance engine, immersive media production and rendering tools.

Section 3 provides the initial list of requirements. Based on a number of meetings and discussions, the consortium has identified three main tools as part of the toolset. The media vault is a software component that allows for asynchronous communication around media objects. It is like a vault that stores media assets, where end-users can upload content, comment about it, and make use of it for media productions, addressing in particular co-creation and co-design processes. The performance engine is a real-time communication infrastructure, deployed at the theatre, that enriches the live performance by orchestrating the stage and allows for online participants to remotely enjoy the show and contribute to it. Finally, the immersive media tool is a set of production and rendering tools that enable the deployment of capsules and other interactive and immersive experiences.

Finally, Section 4 details an initial architecture that satisfies the requirements. In order to decide about the architecture, the alternatives reported in Section 2 have been assessed, and a number of experiments have been conducted. The final decisions have been to use Amazon Web Services (AWS) as the backbone for the media vault, and to re-utilise results from previous EU projects for the other two toolsets: FlexControl[1] for the performance engine and the ImAc player[2] as the basis for the immersive media rendering tool.

The core contributions of this deliverable, after four months into the project are:

- A comprehensive report, comparison, and evaluation, on technologies and infrastructure components that may help on the development and deployment of the TRACTION toolset
- An initial set of requirements, based on initial discussions and conversations with all the consortium
- An initial architectural design decision based on the requirements and experimentation with competing technologies and potential solutions
- A detailed work plan for the formal gathering of requirements in the next months

The next period will continue with the identification of more specific requirements and development work.

[1] An evolved solution of the outcome of FP7 MediaScape project (https://cordis.europa.eu/project/id/610404)
[2] ImAc Player, outcome of H2020 ImAc project (https://cordis.europa.eu/project/id/761974)

1 Introduction

1.1 TRACTION concept and approach

Opera uses all the visual and performing arts to create extraordinary worlds of passion and sensibility. It is rightly recognised as a great achievement of European culture. And yet a form that once inspired social and artistic revolutions is often seen as the staid preserve of the elite. With rising inequality and social exclusion, many see opera—if they think of it at all—as symbolic of what is wrong in Europe today. TRACTION aims to change that using opera as a path for social and cultural inclusion, making it once again a force for radical transformation.

We do not want to make opera palatable to those who don't attend. We want to define new forms of artistic creation through which the most marginalised groups (migrants, the rural poor, young offenders and others) can work with artists to tell the stories that matter now. By combining best practice in participatory art with digital technology's innovations of language, form and process, we will define new approaches to co-creation and innovate in three fields: a) Opera creation and production; b) Immersive and interactive digital media; and c) Social integration and community development.

Experimental projects in inner-city Barcelona (ES), a youth prison in Leiria (PT) and diverse communities in Ireland will test and share new ideas. Bridging the social and cultural divides involved will challenge many existing beliefs, structures and habits. The exceptional resources of the TRACTION partnership will help us meet that challenge through mutual support. The immediate outcomes will be new routes for social and economic integration for the people involved, better relationships between opera producers and society, and cutting-edge technological development. But the long-term prize is the definition of new processes that renew the art's potential to build cohesive societies and imagine a revitalised, common culture in which everyone can feel that they belong

1.2 Purpose of the deliverable

This article provides a snapshot (after four months) of the technology that will be developed in the TRACTION project. In particular, it details the requirements, architecture and development plans of the project.

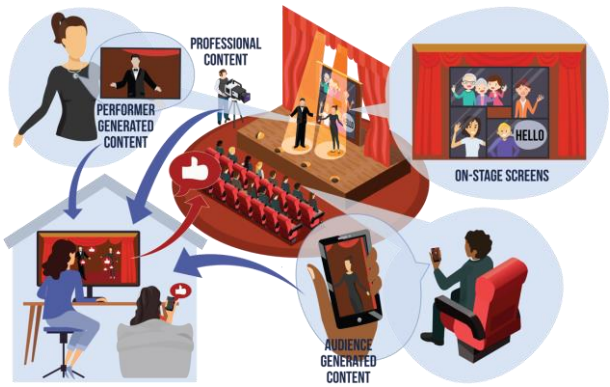
The deliverable includes three main contributions:

- A comprehensive and detailed survey of software components and solutions that could potentially influence the technology to be developed in TRACTION
- An initial description of the toolsets that will be developed (shown in **Figure 1**), including
    - a media vault for storage and asynchronous communication during a co-creation and co-design process;
    - a presentation engine for enriching Opera performances and synchronous real-time communication; and

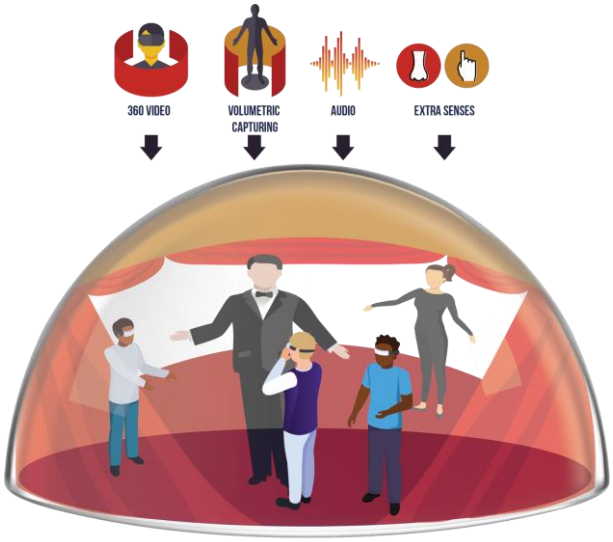
- an immersive environment for the deployment of interactive and immersive experiences.

- A detailed plan of the next steps in terms of requirements gathering using focus groups, and development and integration of components

a) Media Vault



b) Performance Engine



c) Immersive Media Environment

Figure 1: Sketches of the toolsets

1.3 Intended audience

This deliverable is public, so its audience is anyone interested to learn more about the TRACTION project, and its aim of using Opera as a vehicle for facilitating social inclusion. In particular, this deliverable focuses on the technological innovations of the project.

1.4 Structure

This deliverable is divided into three main sections. Section 2 overviews the related technologies and infrastructure. Section 3 provides the initial list of requirements, based on a number of meetings and discussions. Finally, Section 4 details an initial architecture that satisfies the requirements.

2 Related Technologies and Infrastructure

This section details the related work regarding technologies an existing infrastructure, which can be potentially used and integrated in the project. It is divided into the three main tools that are expected to form the overall toolset: Media Vault, Performance Engine, and Immersive Media Environment (production and rendering). It includes as well a section on Communication Tools and Delivery of Media, which it is cross-toolset, as it provides the mechanisms to delivery and adapt the content to different contexts and environments.

2.1 Media Vault

This section describes the different tools considered to serve as media vault for TRACTION: database and media asset management platform. The motivation behind this is that TRACTION needs a tool for asynchronous video-based communication, something that can store multimedia content (not only video, but also audio, immersive and 360º video, etc.) and that such content can be easily accessed, commented, modified and annotated by all stakeholders. Such tool should satisfy the typical requirements of every Video Platform as a Service (VPaaS):

- Extensibility and configurability, to adapt to the specific needs of TRACTION as well as to give the consortium the possibility to add plugins for automatic annotation and metadata extraction;
- Provide support for video and audio editing and transcoding, for example to create videos at different resolutions and bitrates, that can be consumed by any devices and with different bandwidth conditions;
- Include chat and commentary system, including annotations in specific locations and at specific time intervals of the videos, to let the stakeholders ask questions and provide insights about the content uploaded;
- Be open source;
- Provide different user groups, such as administrator, moderator, standard user etc.

Additionally, the chosen VPaaS system should be easy to use or at least provide a short learning curve.

While looking for such tools we first checked the state of the art in the literature and examined many of the tools presented in surveys such as (Watts, 2016) or (Ljubojević, 2017) as well as recent papers like (Wu, 2019), (Carneiro, 2019), (Rossi, 2019) and (Ma, 2017). Unfortunately, all these platforms are highly experimental and are not suitable to be used in the context of TRACTION, as they lack the required stability, support, documentation or even the source code.

For these reasons we turned our attention to existing products. Some of the commercial platforms like Muvi[3], Vimeo OTT[4], Dacast[5] or OvercastHQ[6] were considered but

[3] https://www.muvi.com/

ultimately discarded, as they either did not provide the required flexibility, or do not offer an adequate license model. Furthermore, the expensive price of such solutions would make it difficult to maintain the VPaaS platform beyond the end of TRACTION.

We have then given a closer look to existing open source solutions like Kaltura[7], Plumi[8] and OpenCast[9]. These tree platforms are open source and allow to create a video database and management platform without additional costs beyond hosting and bandwidth ones. Kaltura. Table 1 summarizes the main advantages and disadvantages of these platforms.

Table 1: Overview of open source video database solutions.

Name	Features	Pros	Cons
Kaltura	Mature VPaaS Widely used Admin & user section	OSS & commercial license Video editor Mobile support	PHP Steep Learning curve Too many features
Plumi	Video MGMT & Viz	Open source	No editor
OpenCast	Video capture, MGMT and distribution (for academic institutions)	Open source Video editor (very basic)	Java No mobile video capture

2.1.1 Kaltura

Given the initial comparison, we decided to closely focus on Kaltura, as it offers a set of characteristics suitable for TRACTION:

- It offers a very mature API for video management
- The plugin systems allow adding new functionalities, such as
  - Batch processing of all new or existing videos
  - Processing of user-defined sets of videos
  - Provide manual or automatically extracted metadata
- The API is available in several programming languages (PHP, NodeJS, Python among the others)
- The Admin console allows quick and easy testing of the API
- It has a strong user community
- If needed, Kaltura offers commercial cloud-based service, as well as paid support

[4] https://vimeo.com/ott/home
[5] https://www.dacast.com/video-hosting-manager/
[6] https://www.overcasthq.com/about/
[7] https://corp.kaltura.com/
[8] https://plumi.org/
[9] https://opencast.org/

Figure 2 shows the front-end of a VPaaS solution using Kaltura. In the centre of the screen there is the dashboard, showing a list with latest videos added to the database. The menu on the left of the screen allows switching between different views, such as the Dashboard, the Admin Console, the User groups etc.

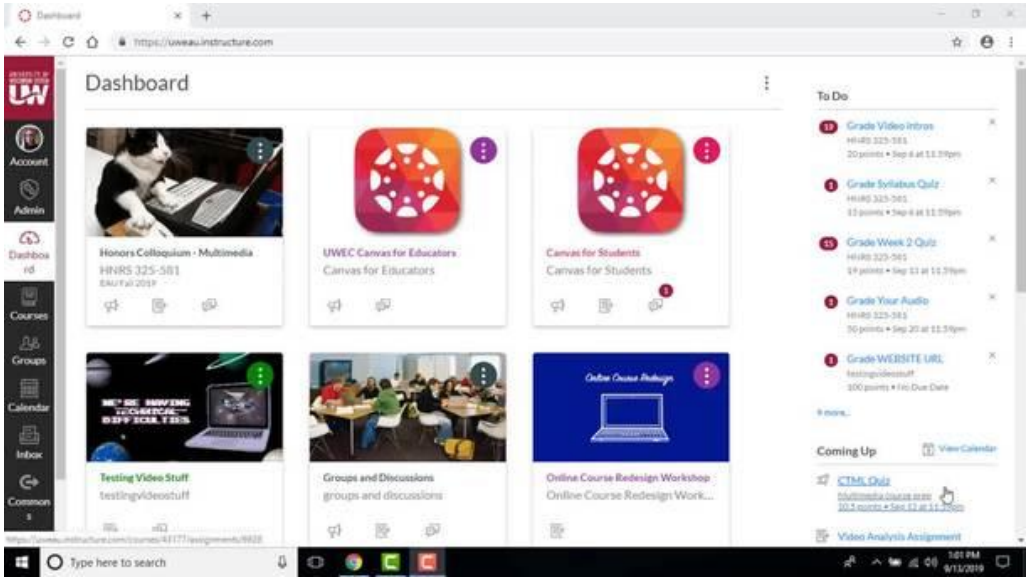


Figure 2: An example of a video database using Kaltura.

The architecture of Kaltura is quite complex and it is organized in layers, in order to enable flexible usage of its technologies and features. An overview of Kaltura's architecture is depicted in Figure 3. The backend part is the most significant part of Kaltura, since the most complex functionalities are implemented in it: user control, multimedia flow control, coding, Rest API services, etc. The frontend basically consumes the Rest API resources of the backend. It provides interesting features, such as the video editor or automatic preview capture. Unfortunately, some of the features of the front-end (such as the video editor) are only available in the commercial version of Kaltura.

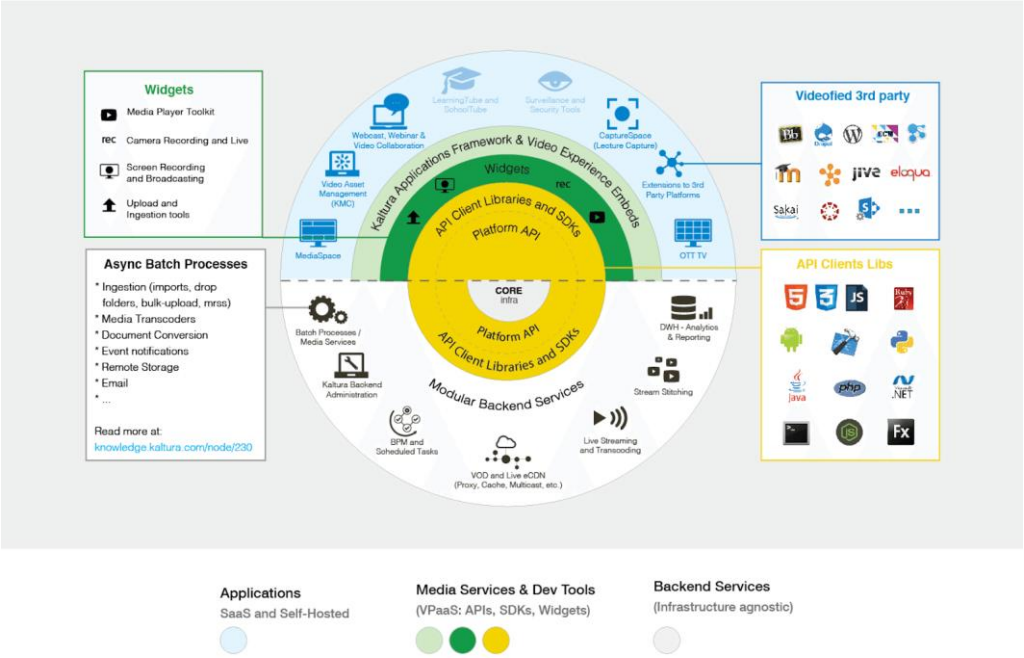


Figure 3: Overview of Kaltura architecture (credits: corp.kaltura.com).

Kaltura's platform is organized in five different layers:

1. Core technologies
2. Web services (Kaltura API)
3. API Client libraries
4. Widgets
5. Applications

The core technologies are the main features of the Kaltura server-side implementation, and they include the following services:

- Storage and hosting;
- Content delivery and streaming;
- Media ingestion and transcoding;
- Media manipulation and management;
- Account and user management;
- Advertising and monetisation;
- Media syndication and distribution;
- Styling and branding.

Kaltura's web services layer provides web access to Kaltura's server Application Programming Interfaces (APIs), through a standard HTTP POST/GET URL-encoded requests structure. Kaltura server APIs, also named Kaltura Partner Services, are designed to follow REST principles. The API consists of several service actions for querying, setting, updating and listing entities, as well as for activating processes within the Kaltura Platform. Service actions are grouped according to the entity type they apply to.

Layer 3 provides several API client libraries written in all common programming languages. These client libraries implement the actual calls to the Kaltura server APIs. They handle HTTP request creation, queuing, and response processing while providing specific classes corresponding to entities and data types used by Kaltura server APIs.

Layer 4 implements specific APIs in Flash (now deprecated) or HTML to provide widgets. Kaltura widgets rely on advanced web technologies, encapsulating graphics, media functionalities, and workflow events control along with direct access to Kaltura core technologies (via the Kaltura web services layer). Kaltura's most commonly used widgets include: Kaltura Dynamic Player (KDP), Kaltura Contributor Wizard (KCW) and the Kaltura Video-PowerPoint Widget.

Finally, layer 5 provides applications, that is a tailored combination of Kaltura widgets and server APIs. Applications are tailored to support specific media workflow or are extensions to a 3rd party content management platform. Examples of basic Kaltura applications include the Kaltura Management Console and the Kaltura Admin Console.

2.1.2 Amazon Web Services (AWS)

One other alternative is Amazon Web Services (AWS), which provides a series of services for hosting, distribution, cloud computing and processing of media. Services, that might be of interest to the project include:

S3 (Simple Storage Service): Storage of arbitrary files, used to store web assets as well as uploaded video files. The service organizes files as objects inside so-called buckets. Each object within a bucket has a unique key and prefixes can be used to give the bucket a hierarchical structure, much like a common file system.

EC2 (Elastic Compute Cloud): VPS hosting, used for hosting backend code which processes user inputs, host the API or website assets. Is also responsible for transferring uploaded content to S3, storing metadata in a database or starting the transcoding process. Note that EC2 is not the only option here. Based on specific requirements, this can be replaced by Elastic Container Service (equivalent but uses containers), Lambda (breaks functionality down into functions which are run in response to specific events) or a whole Kubernetes cluster

Elastic Transcoder: Video/audio processing, used to transcode files uploaded by users. This service takes files from S3 and puts them through a transcoding pipeline which can convert videos/audio to different resolutions or file formats and places the resulting files back into S3.

DocumentDB: Structured data storage, can be used to store file meta- data, user data or and type of structured data in a document-oriented format which is compatible to MongoDB. Alternatively, this service can be replaced by RDS, which instead of document-oriented data storage stores the data in a relational manner.

Cloudfront: CDN, used for distribution and caching of data in geographically advantageous locations relative to the client. This is largely optional, but will improve response times on the end-user side.

2.2 Performance Engine

This section describes the different available technologies in order to provide an engine to manage the rendering of heterogeneous content sources across distributed displays. First, orchestration technologies are introduced, followed by relevant communication platforms, and ending with production alternatives.

The motivation behind these features is that TRACTION needs a tool for the real-time distribution and communication of rich media, understanding media as a broad concept that includes videos and images, real-time video and audio streams, immersive media formats (such as 360 videos), textual information (content from social media), etc. These features are particularly relevant for the performance stage of the trials, where the technology could contribute to enhance traditional opera performances.

2.2.1 Orchestration Engines

2-IMMERSE was a EU-funded project which aimed at creating a system for immersive object-based immersive multi-screen broadcasts over the internet. Viewers could enjoy live broadcasts using a 2-IMMERSE set-top box form their living room. The main broadcast would run on their television set, whereas additional content would be displayed on a connected tablet. This additional content included textual information about the content being shown on the television, the ability to interact with other people watching the same broadcast or the switching of camera angles. This interactivity was enabled through cutting-edge web technologies and object-based broadcasting. While in traditional broadcasting, the content is assembled at the studio and sent to peoples' homes as a single broadcast, in object-based broadcasting, the studio generates the assets as distinct objects, which as all sent as separate streams the end-users, where they are assembled. This gives content producers the ability to create broadcasts which adapt themselves to their consumers' needs. The content can adapt itself to different screen sizes, different device types and gives the user more agency in the type of content they actually want to see. This new broadcasting paradigm also necessitated a shift in the way that the content was produced. For this purpose, a whole new pipeline for temporal and spatial synchronisation of broadcast objects was implemented. This also gave producers more agency through a new editing system, which aimed to extend the paradigm of non-linear video editing by instead of arranging clips of a timeline, gave the producer the ability to lay out their broadcast in terms of relationships between media objects, i.e. parallel and sequential ordering of media objects. Finally, in order to extend this system for the use in live sports broadcasting, the system was extended with a live-editing system, which allowed a team of media professionals to insert broadcast objects into a live stream through a simple button-oriented interface.

VICOM coordinated an FP7 European Project called MediaScape[10] (2013-2016), to provide interoperable technologies for the creation and distribution of HTML-based media services that can be delivered seamlessly and in a simultaneous way across any type of connected devices, fostering the convergence of Television and Internet. VICOM,

[10] MediaScape Website in Cordis: https://cordis.europa.eu/project/id/610404

as an outcome of the project, created MESH[11], a library to enable services that define multi-user and multi-devices media-viewing experiences for end-users in a standard-based approach.

Figure 4: Screenshot of the MESH library being used for a TV programme[12]

This technology has evolved from a broadcast-oriented solution towards a flexible media control system for crisis management in Public Safety. The technology, now called FlexControl, is being used by Police Bodies to monitor crisis event and to make decisions. It connected agents in mobility with a crisis room and remote experts. The following image shows how Ertzaintza[13] uses FlexControl during a crisis.

Figure 5: Screenshot of the FlexControl library being used by a police body during a crisis[14]

[11] VICOM's libraries: https://www.vicomtech.org/en/rdi-tangible/software-libraries
[12] Demo Video of MESH: https://www.youtube.com/watch?v=NwixgA_M144
[13] Ertzaintza is the Basque Police Body
[14] Video of FlexControl (Ertzaintza): https://www.youtube.com/watch?v=OeHYqYjG1PA

FlexControl allows to define a Web application with multiple and heterogeneous sources, such as video files, images, real-time media, HLS, WebRTC, text, etc. The library enables the association of multiple devices, having an overview of how many devices are connected, the features and capabilities of each one of the devices, the role of each one, etc. Moreover, FlexControl provides adaptation rules and algorithms to automatically decide what kind of content must be shown where, and which User Interface is the best. Finally, the library provides synchronisation mechanisms across different devices.

Through a Web browser, FlexControl enables different roles to participate in the application:

- Visualisation of the information: With this profile, the display will be used to show part of the content sources.
- Administrator: With this profile, this interface will decide what to see where. The adaptation rules can provide an automatic outcome, but then the administrator can manage and change everything, moving components from one device to another, changing the layout, interacting with the content, etc.
- Operator: With this profile, an operator can be using a specific application, for instance Google Earth, and the information being showed in the screen can be distributed in real-time as one content source more for the others.

At the moment, FlexControl is adapted to cover a crisis management use case in a very flexible way, since it is possible to create a videowall with N devices (TVs, laptops, etc.) without having an specific hardware or software, and moreover, mobile devices can be used (as far as they have a Web browser).

A typical use case for emergencies is to create a videowall with 6 or 9 displays (typically TVs) in a crisis room, where there is an administrator deciding what to see where in the videowall, some operators providing inputs (Google Earth, Social Media, etc.), and some police officers in mobility consuming information in their devices (sent by the administrator) and providing also information (e.g. the camera of the mobile device). The following image provides a diagram of the use case.

There is a worldwide company based in USA, called Carbyne[15], that provides a proprietary cloud native solution that unifies the flow of emergency life-saving information (Voice & Data) into one unified platform[16].

An engine which shares some functionalities with FlexControl is Bosch Video Management System (BVMS), developed by the Security System branch of Bosch. BVMS is an integrated solution which combines both hardware and software components. BVMS is oriented to security and surveillance applications and, unlike FlexControl, it focuses specifically on video streaming, without offering broadcast integration or message passing. The most important software component is Bosch Video Client, which is installed on every IP camera in the CCTV system and allows managing them all from a single PC. The software is able to manage and configure up to 128 cameras and show on

[15] Carbyne Company: https://carbyne911.com/
[16] Carbyne solution video: https://www.youtube.com/watch?v=SyU6pLZMRxw

screen up to 40 camera channels. It is possible to apply different configuration settings for different cameras, as well as to control the pan, tilt and zoom settings of each camera. More advanced settings include the possibility of recording videos, either locally, directly on camera or on cloud storage, and the automatic localization of important events based on triggers such as specific sounds, or unexpected movements.

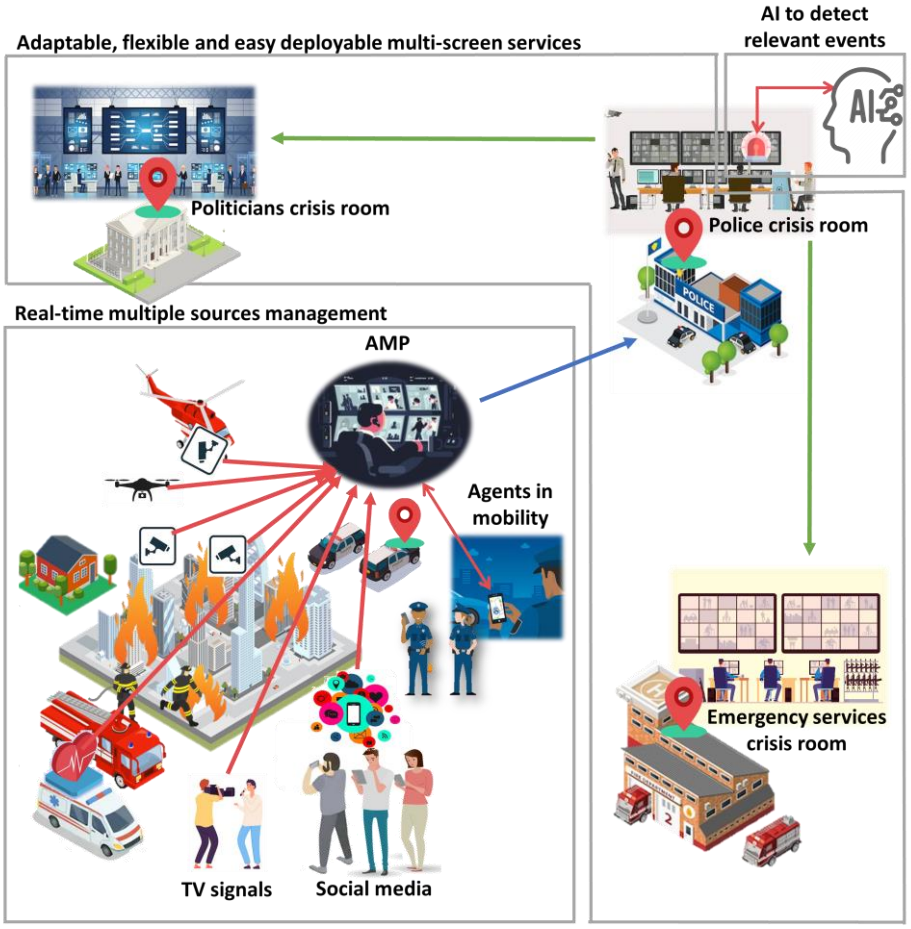


Figure 6: Diagram of the use case that FlexControl addresses in a crisis management

2.2.2 Communication Platforms

Besides the engines presented above, there is the possibility to develop tools for the real-time distribution and communication of audio-visual content by using existing communication platforms. The two most popular such platforms are probably Firebase and Parse. Firebase, acquired by Google in 2014, is a comprehensive platform for the development of web and mobile applications. The first product Firebase offered was a real-time database API which allowed synchronization of application data between web pages and Android or iOS application, hosting data on Firebase cloud. Since then, many other products have been added, like for example a tight integration with Google services, such as storage, authentication, a machine learning platform, tools for app monitoring and remote configuration. Firebase REST API allows for passing of notifications to single or multiple clients, while video streaming can be implemented in two ways:


- Using a combination of Firebase Storage and Firebase Real Time Database: storing short video chunks and then syncing them through the database. This solution is easy to implement but performs bad in terms of latency, as it incurs in multiple seconds delay. Even though such delays can be acceptable depending on the applications, this is far from an ideal solution.
- Use WebRTC for video streaming and use Firebase Real Time Database only for signalling. Implementation this solution requires more effort and is harder to maintain (as Firebase does not offer a WebRTC API at the moment) but it solves the latency problem of the method described above.

Parse, similarly to Firebase, is a platform for the creation of web and mobile applications which provides persistent object and file storage, user authentication, dashboard and push notifications. Parse, an open source project initially developed at Facebook, is composed by two main parts: the Parse Server and the Parse Dashboard.

Parse Server is an open source backend that can be deployed to any infrastructure that can run Node.js. It uses MongoDB as a database and can be deployed on any cloud provider (e.g. Heroku or AWS). Its core functionalities include caching, OAuth, live queries, push notifications and class level permissions. The open source community is creating modules to more and more functionalities, including adapters to GCloud, authentication with Firebase accounts, integration of image and video modules to the backend. Parse Dashboard is a standalone dashboard that allows web-based management of Parse Server applications. All the apps parameters can be managed through JSON configuration files, and through the dashboard the administrator is able to manage multiple Parse Server apps as well as manage the UI, security and authentication options.

2.2.3 Production Tools

In terms of production, MAX MSP is a visual programming language for music and multimedia that allows users to build complex, interactive programs without any prior experience writing code. (Ghassaei, 2017). MaxMSP has been described as a bridge for developing interactive music performance software (Place, 2006). It is especially useful for building audio, MIDI, video, and graphics applications where user interaction is needed, has been used by composers, performers, software designers, researchers, and artists to create a variety of performances and installations. Also known as Max/MSP/Jitter, the tool splits into several operations. “Max” handles discrete operations and MIDI, “MSP” deals with signal processing and audio, and “Jitter” is for graphics rendering and video manipulation. The Max tool is modular, supporting most routines through shared libraries, supporting a range of integration with external software, hardware controllers, and computer vision. The tool also has an API that allows third-party developers to develop external objectives and routines (Ghassaei, 2017). Max has a large user base of independent programmers who enhance the software with commercial or non-commercial extensions using C++, NodeJS, Java, or JavaScript. For instance, a popular extension is the Max integration with Ableton Live, a digital audio workstation for live performance, composition and mastering.

LICEU has been using several commercial tools for real-time composition and manipulation of multimedia streams. Although these tools are more oriented towards the creation of audio-visual material, they also provide functionalities of multimedia management. Among these products, the ones that come closer to providing the functionalities required by the performance engine are Millumin[17], QLab[18] and Catalyst[19]. Millumin is a solution for the creation of audiovisual shows through the combination of several workflows. The software, currently available only on Mac platforms, provides plugins for several software like AfterEffect, Cinema4D, Unity, Syphon) and provides a list of features such as sequences, videomapping, multiscreen support, object tracking. QLab offers the possibility to synchronize audio, video and light cues using a graphical user interface. Among its features, it offers matrix-based audio routing, video mapping, multi-screen playback, multi-device support. Every component of the audiovisual product can be rearranged on a timeline view by drag-and-drop, making it easy to adapt and change shows to different kind of scenarios. Catalyst is an image processing system that provides the facilities to control and manipulate high resolution (HD & 4K) movies, live camera, or images for playback through any number of video projection or LED screen devices. Catalyst offers many image processing tools, such as 20x Composite or 8x SD/HD-SDI live video inputs (using multiple capture cards). Controlled via DMX / Artnet or standalone, Catalyst provides instant access playback high resolution (HD & 4K) content and the ability to manipulate images with a range of colour and visual effects. Precise keystone or 3D geometry control can be achieved within each layer or globally on a mixed output.

Another commercial product which offers some of the characteristics required of a performance engine is Jamkazam[20]. Unlike the products described above, Jamkazam focuses on allowing multiple musicians to perform at the same time without sharing the same physical space, over a standard internet connection (provided the latency is low enough). Jamkazam combines characteristics of a live music platform with those of social networks. The basic version, which is free to use, offers two core functionalities:

- Playing music live, remotely and synchronized, with friends using the system
- Connect through the platform to other musicians, with possibilities to search for similar tastes, instrument played and so on

Music session can be recorded, shared between bandmates and even livestreamed using the platform or Facebook. Jamkazam has some limitations related to the bandwidth and latency requirements. As lag can severely impact the quality of the user experience as well as that of the live recording, the system simply refuses to start, when it detects that the quality of the internet connection is not good enough.

[17] Millumin.com/v3/index.php
[18] Qlab.app
[19] Catalyst-v5.com
[20] Jamkazam.com

2.3 Immersive Media Environment

2.3.1 Immersive Media formats

Currently, several different immersive media formats exist. One differentiator is the degrees of freedom (DoF): 3 DoF v 6 DoF where DoF stands for 'Degrees of Freedom'.

- 3 DoF means orientation tracking. This means the 3 axes in which an object can be rotated about are tracked. This exists in mobile VR headsets and standalone VR headsets like the Oculus Go. If you turn your head while wearing a 3 DoF headset, it can track the angle change of this axes and allows you to look around in the environment.
- 6 DoF VR headsets allow for the position of the headset to be tracked, as well as the orientation of the headset. Movement can be tracked in 3 axes x, y and z and any combination of these three axes can express movement, this is called a vector.

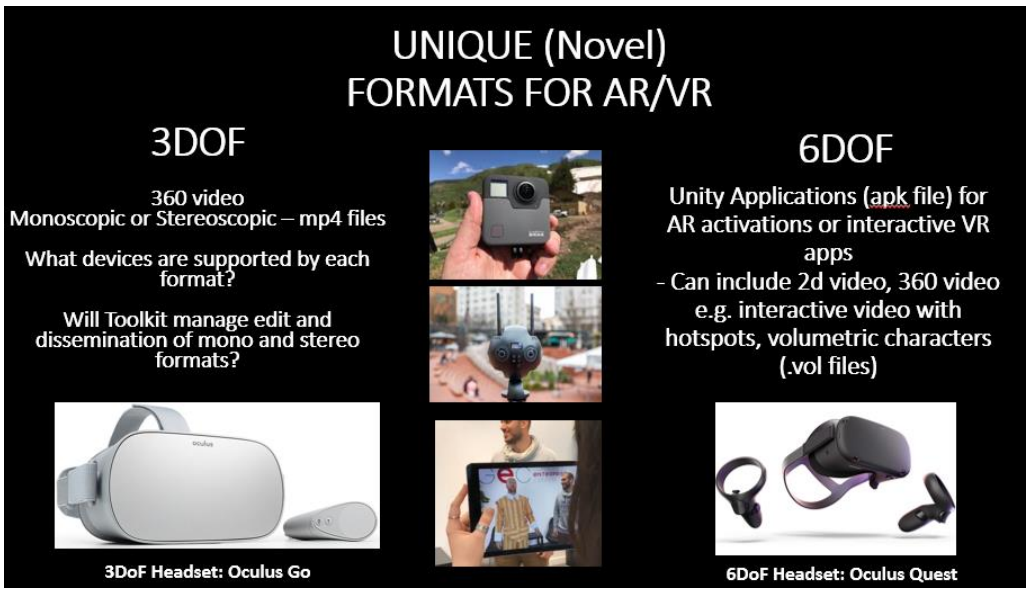


Figure 7: 3DoF and 6DoF formats and devices

For 360 Videos, we can differentiate between monoscopic and stereoscopic. Monoscopic 360 consists of a single flat image, projected on a sphere around the viewer. Monoscopic VR is great for 360 videos for which a headset is not required, or if the shots contain lots of dynamic movement. Stereoscopic imagery is often displayed in the Top/Bottom 3D format. The top image is the Left Eye view, the bottom image is the Right Eye view. Stereoscopic productions require more careful planning and execution, to make sure the viewer is comfortable to focus on the story. In most cases, the 360-degree VR video is formatted with MP4 or MKV container formats, which can be seen from YouTube VR videos and Facebook 360 videos. You can also watch 4K 360 video online with flash video format like FLV, WebM, MPEG, etc.

Volumetric video is a technique that captures a three-dimensional space, such as a location or performance. This type of volumography acquires data that can be viewed on flat screens as well as using 3D displays and VR headsets. Consumer-facing formats are

numerous and the required motion capture techniques lean on computer graphics, photogrammetry, and other computation-based methods. The viewer generally experiences the result in a real-time engine and has direct input in exploring the generated volume.

2.3.2 Capture and authoring: 360 videos

This section describes the different workflows for 360 videos. We describe the typical workflow of Virtual Ireland as an example of professional usage of existing tool: Go Pro Fusion camera for Monoscopic content, and the Insta 360 Pro 2 camera for Stereoscopic 360 video content. The content from either camera needs to be stitched, then edited, then rendered for mp4 file creation.

Monoscopic 360 Video: Go Pro Fusion

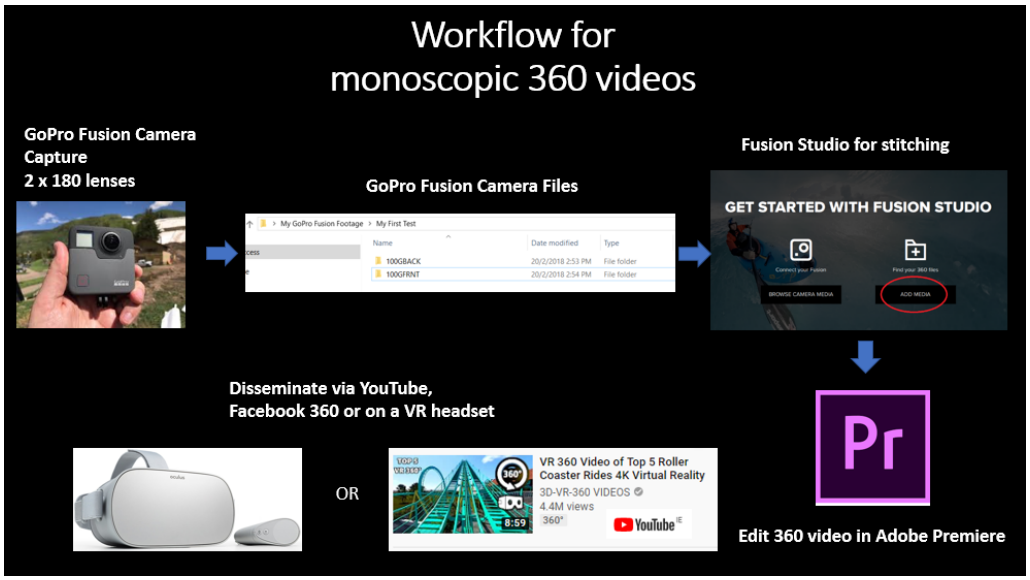


Figure 8: Workflow for the GoPro Fusion camera (monoscopic footage)

Video settings for the GoPro Fusion camera:

- Frames per second (FPS) refers to the number of video frames that are captured in each second. When selecting a resolution and FPS combination, consider the activity that you want to capture.
- Fusion video resolutions are captured with a 16:9 aspect ratio, which is the standard for televisions and editing programs.
- The field of view (FOV) refers to how much of the scene (measured in degrees) can be captured through the camera lens. The FOV for all Fusion video is Spherical, which captures a full 360 view.
- Fusion video resolutions:
    - Video
    - Resolution
    - (RES)
    - FPS (NTSC/PAL)* FOV
    - Screen

- Resolution
- 5.2K
- (default)
- 30/25 Spherical 4992X2496
- 3K 60/50 Spherical 3000X1504

*NTSC and PAL refer to the video format, which depends on the region that you are in.

Capture footage with the GoPro Fusion camera:

- Press the Mode button to power on the camera.
- Select a mode and settings. For details, see Changing Modes and Settings.
- Press the Shutter button. The camera beeps and the camera.
- status lights flash while the camera is capturing.
- To stop capturing video or time lapse, press the Shutter button.
- The camera beeps and the camera status lights flash quickly.

Stitching footage from the GoPro Fusion camera:

- Open the Fusion Studio software.
- In Fusion Studio they can stitch, render, stabilize and colour both spherical and reframed OverCapture content.
- Steps:
  - Open Fusion Studio with your camera connected, and select “Browse Camera Content.”
  - With your content organized at left of Fusion Studio window, select clips you want to render, choose your in- and out-points with the slider just below preview window.
  - Select “360” or “OverCapture” with the toggle below player at right of workspace. **Choose 360 if you plan to edit in Adobe Premiere CC with GoPro VR Plugins. If you want to punch out multiple angles out of a single clip with OverCapture, select the “shot” on the left, and click “Create A Copy” at bottom right of Studio window. Make as many copies as you want clips.
  - Use the Yaw, Pitch, Roll and FOV (OverCapture) sliders to adjust your composition.
  - Once you have composed your shots, click “Add to Render Queue.”
  - Click “render”.

Adobe Premiere Pro CC:

- After you have completed stitching your footage, you can now “edit” them in traditional video editing software like Adobe Premiere Pro CC. Stitching is a common term that involves merging the separate camera inputs into single viewable format. On the other hand, editing is a broad umbrella term that we will use to refer to the post-production that occurs after stitching. This will include trimming the clips, adding multimedia elements, and more.

- Once the footage is edited in Adobe Premier it is output with export settings that are dependent on the distribution device or platform. For example, for Oculus Go distribution, the recommended settings are:

Container and general information

MPEG-4 (Base Media / Version 2): 765 MiB, 14 min 20 s Encoded date: UTC 2019-02-04 13:02:57
1 video stream: HEVC Tagged date: UTC 2019-02-04 13:03:21
1 audio stream: AAC LC TIM: 00:00:00:00
TSC: 25
TSZ: 1

First video stream

English, 7 137 kb/s, 3840*1920 (2.000), at 25.000 FPS, HEVC (Main@L5@Main)

First audio stream

English, 317 kb/s, 48.0 kHz, 2 channels, AAC LC

Note: Export format is an mp4.

Insta 360 Pro 2 / Stereoscopic 360 Video

For Stereoscopic workflow, the Insta Stitcher software for stitching the Insta 360 Pro 2 footage can be used.

Figure 9: Workflow for the Inst 360 Pro 2 camera (stereoscopic footage)

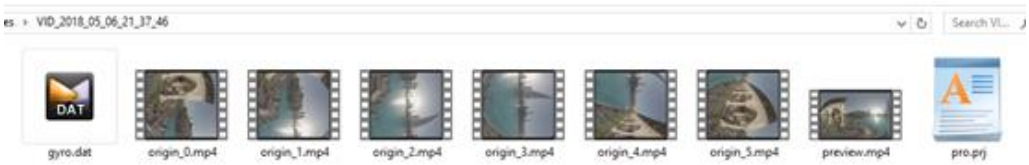
Insta 360 Pro 2 Capture:

- Select the video mode from the camera menu, press "POWER" key to confirm entering such mode. After the camera completes the preparation, the video recording READY state will be displayed.
- When the video recording mode is in video recording READY state, press UP and DOWN keys to switch over the size levels for recording.

- When the video recording mode is in video recording READY state, press POWER key and you can start the recording as per the current size. (Note: If the storage device is used for the first time, the speed testing is required. The video recording proceed smoothly only after the speed is ensured to meet the standard.)
- To stop recording video, please press POWER key again. If the real-time stitching level is needed after the video is shot, the camera will enter the processing state. After processing is done, the storage will be carried out, and then, the camera will be switched to video recording READY state. The work indicating lamp will be flashing until video shooting is done. If the flashing lamp affects the shooting, it can be turned off in the camera settings. Once the video is shot and saved, a sound will be made for indication.

Insta Stitcher:

- Format of video files:



- Video shot on Pro 2 is stored in MP4 format, encoded by H.264.
- Each shoot creates a folder in SD card, which contains 6 low-resolution proxy videos, preview file (Preview.mp4), project file (pro.prj) and some necessary data files (gyro.mp4) as well as video files. The other 6 MicroSD cards are used to store original high-resolution videos shot on 6 lenses respectively.
- origin_*.mp4 sequences are original files captured by each lens for post stitching. Videos in 3840 * 2160 resolution can stitch 8K 2D panoramic video at the maximum and those in 3840 * 2880 resolution can stitch 8K 3D panoramic video at the maximum.
- Preview.mp4 is a preview file of 1920 * 960 with frame rate of 30 fps, which can be used to quickly review the video effects like exposure, composition and so on. Please note that FlowState stabilisation doesn't take effect in the preview file.

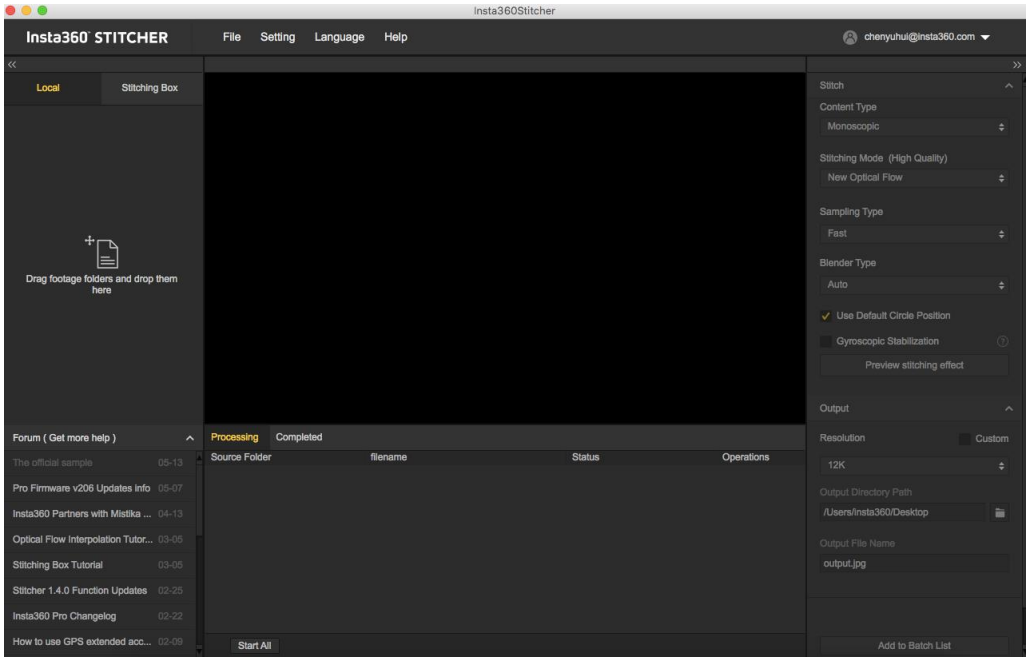


Figure 10: Stitcher interface

The interface of Stitcher includes on the top is the menu bar: File, Settings, Language, Help. You can find functions of file import, google street view upload, log display, preference settings (hardware decoding or software decoding), hardware performance test, language settings, log upload, etc. On the left is a list of files. You can drag folders directly to import files. Pro's official forum is shown at the bottom left, which provides latest software information, tutorials, technical discussions, feedback and suggestions to Insta360. In the middle is real time monitor window, supporting the playback of the file in any lens. Below is task status bar, where you can see the stitching process and check what has been done. In the upper right is stitching setting area, where you can set up stitching Content Type (Monoscopic and Stereo) and Stitching Mode (optical flow and template stitching). Sampling type and Blender Type generally have default settings. Default Circle Position is used to optimize stitching at top of the frame and under dark conditions.

2.3.3 Capture and authoring: volumetric videos

Volumetric video is regarded worldwide as the next important development step in media production. Especially in the context of rapidly evolving Virtual and Augmented Reality markets, volumetric video is becoming a key technology. A number of professional solutions are reviewed, based on the know-how of VR Ireland. One example is Vologrames, a start-up in Dublin who have a volumetric capture studio and have created bespoke algorithms for mesh. Other solutions include Volucap and Mimic Productions in Germany.

Vologrames offer multi-view capture, with different camera configurations. Capture in Vologrames green-screen studio, but also in your own studio, or arrange a custom capture space and they will bring their cameras.

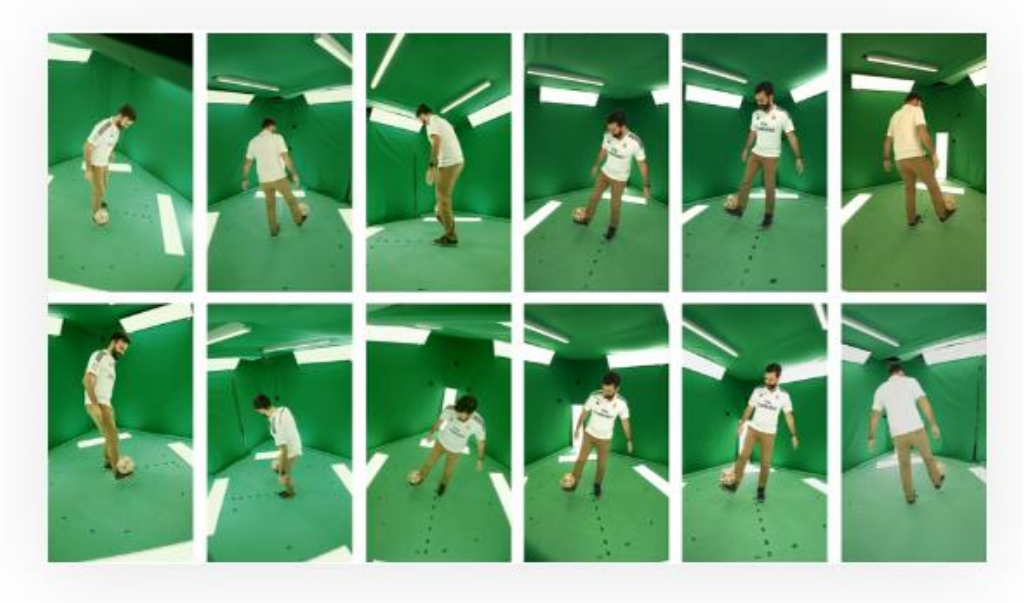


Figure 11: Volumetric video multi-view capturing.

It as well offers reconstruction technology to create high-quality volumetric video sequences. One can choose the right polygon count and texture resolution to guarantee the best performance. It as well provides easy integration with all kind of vendors such as Unity, Magic Leap, Hololens, AR Kit / AR Core, Snapchat lenses, and Spark AR.

As indicated by the CEO at Vologrames, Rafael Pages, “There is no standard volumetric file or open-source player across the various volumetric studios.” A volumetric video sequence is basically a collection of the 3D meshes, and all the players are simply loading them sequentially. Each company has its own way of compressing the sequences, and a different way to represent the models- Microsoft, 4Dviews, Vologrames, and most of others use 3D meshes and image textures: Vologrames track a mesh throughout a sequence of meshes to be able to re-use the same topology through a series of frames. This helps with compression (by eliminating redundancies) and also with the textures, as you can store them as a video, instead of a sequence of images. 8i and Samsung use point cloud rendering (point billboards or voxels), and they have their own ways of storing the point positions in a video frame. The closest thing to an open-source volumetric video player is a plugin in Blender that allows you to load a sequence of OBJs or PLYs with or without texture and play them as a stop-motion player.

Regarding the format for the Vologrames files, the characters are housed in a .vol file and build into a unity package and related asset library. The example below is volumetric character created for VR Ireland by Vologrames called ‘Axe Guy’

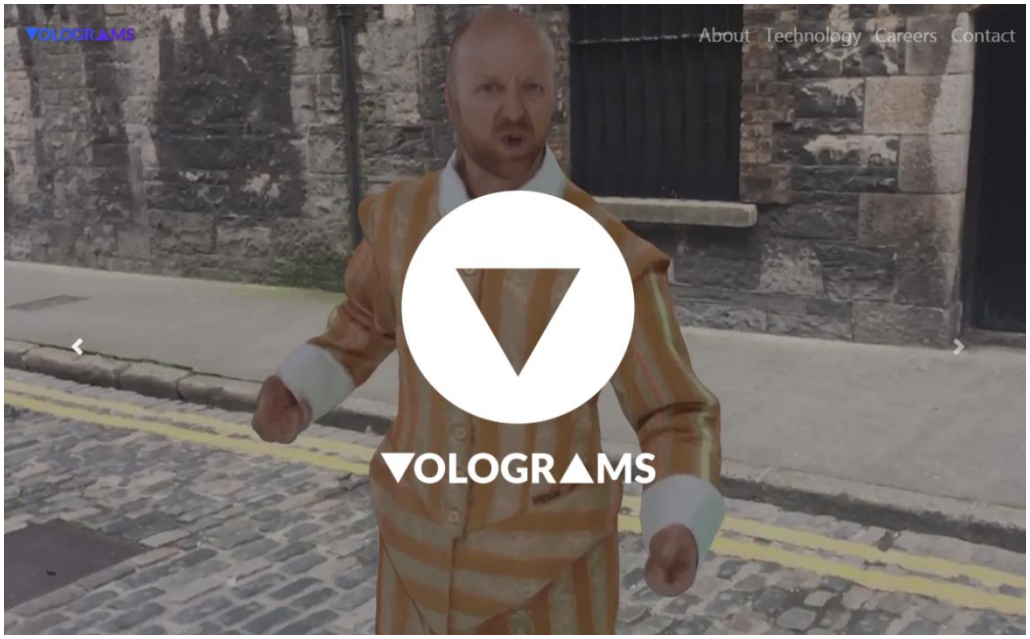


Figure 12: Axe Guy volumetric character from Vologrames.

To ensure that there are no errors in the Vologrames scripts ensure that the "Scripting Runtime Version" is set to ".NET 4.x Equivalent". This can be found through Edit > Project Settings > Player and then within the Inspector window under Other Settings > Configuration. Where to find Scripting Runtime Version in Player Settings

To set up the vologram in Unity, import the Vologrames package and follow these steps:

- Create an Empty Object in your scene and name it Vologram.
- Create another Empty Object and make it the child of Vologram. Name this object Player.
- Add the VOL Play Frames component to Player.
- This will automatically add the VOL Asset Buffer and VOL Asset Loader components, and a Video Player.
- In the Assets folder in Unity, add a new folder and name it StreamingAssets (spelling is important).
- Drag the AxeGuy 1K Mobile folder and drop it into the StreamingAssets folder.
- Within Unity, drag the same folder into the VOLS file field in the VOL Play Frames component.
- Drag and drop the AxeGuyAudio.mp3 file into Unity's Asset folder.
- Add an Audio Source component to the Player object.
- Drag the AxeGuyAudio.mp3 file into the AudioClip field in the Audio Source and disable.
- Play on Awake.
- Add the Audio Source component to the Audio source field in the VOL Play Frames component.
- Ensure the tick boxes for Play on start and Loop are enabled.

Below there are some screenshots of how the set up should look like in Unity:

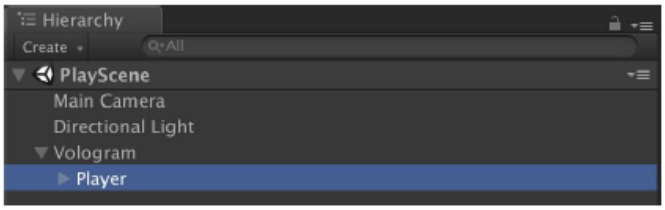


Figure 13: Unity: Hierarchy in the scene for importing Vologrames

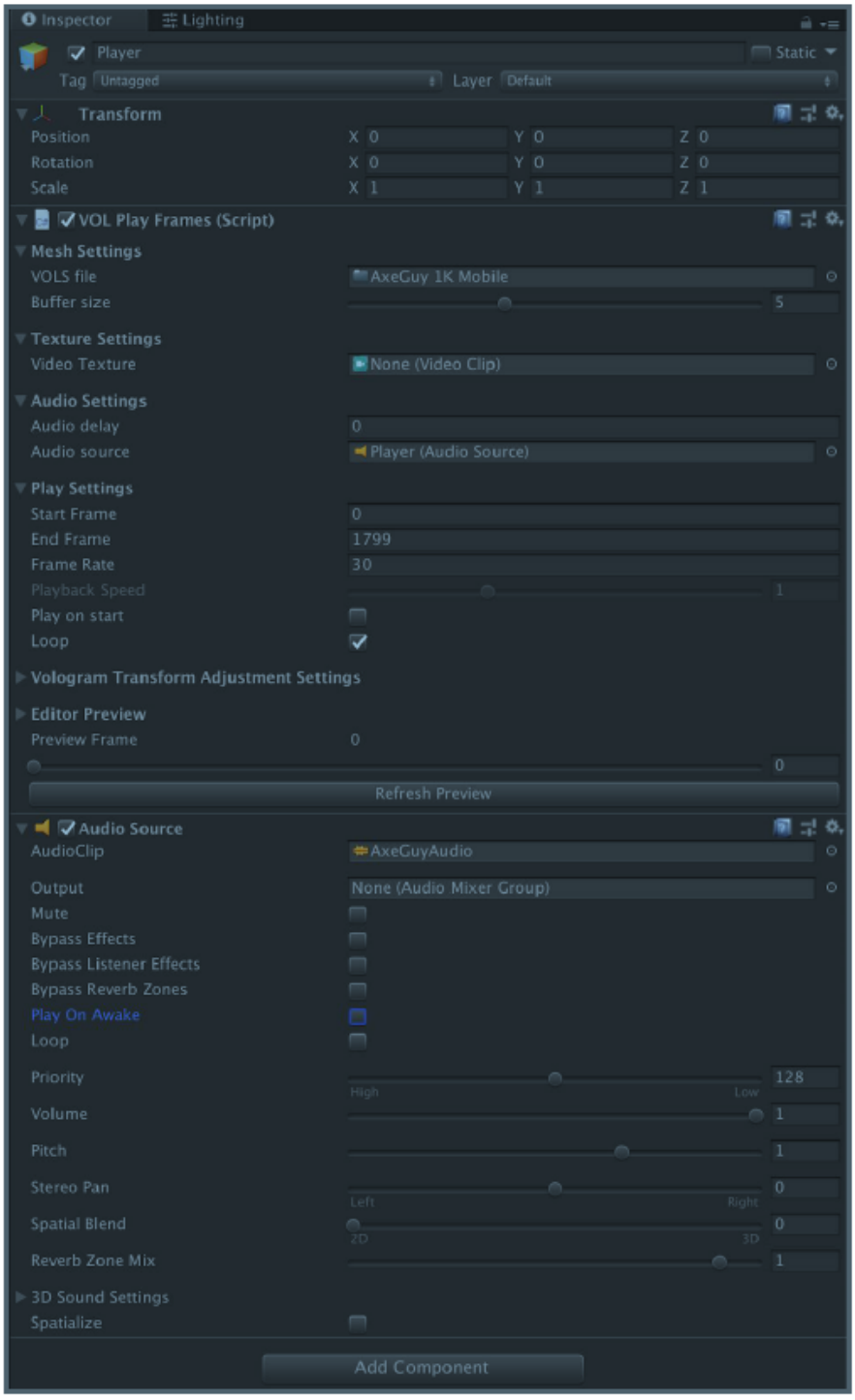


Figure 14: Unity: Player Configuration for importing Vologrames

VOLPlayFrames Reference

To Play the Vologram from a Unity script you can use these functions to control playback:

`void VOLPlayFrames.Play()`	Begins or continues the vologram playback.
`void VOLPlayFrames.PlayDelayed(float time)`	Begins or continues the vologram playback after a specified time (in seconds) has passed.
`void VOLPlayFrames.Pause()`	Pauses the vologram playback.
`void VOLPlayFrames.Stop()`	Stops the vologram playback and returns the video and audio players to the start.
`void VOLPlayFrames.Restart()`	Stops the vologram playback and starts playback from the beginning.
`bool VOLPlayFrames.IsPlaying()`	Returns true if the vologram is playing, false otherwise.
`bool VOLPlayFrames.IsReady()`	Returns true if the vologram is prepared to start playback, false otherwise.

Figure 15: Unity: Scripts for importing Vologrames

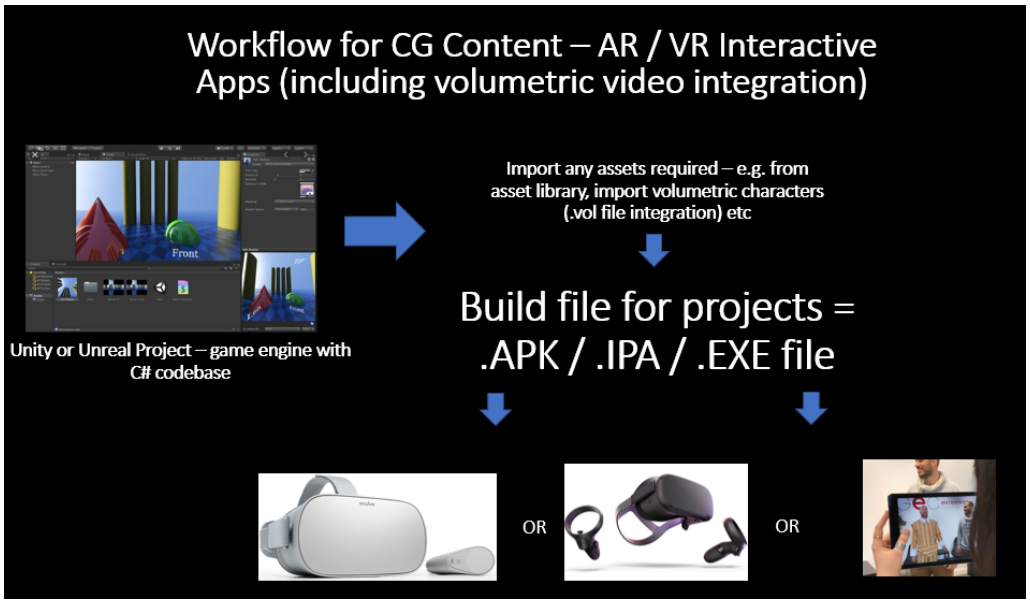


Figure 16: Workflow for Unity Applications

Device	AR/VR	Unity File Format
HTC Vive	VR	Exe
Oculus Rift	VR	Exe
Pico	VR	Apk
Oculus Go	VR	Apk
Oculus Quest	VR	Apk
iPad	AR	IPA
Android mobile	AR	APK

Figure 17: File Formats Unity Applications

2.3.4 Capture and authoring: immersive audio

Ambisonics is a sound format that is different from your usual stereo/surround paradigm because its channels are not attached to speakers. Instead, an Ambisonics recording actually represents the whole spherical soundfield around a point. In practice, it means that you can represent sound coming from all directions around a listening position and, using an appropriate decoder, you can playback the same recording in any set of speakers with any number of channels arranged around the listener horizontally or vertically. That is exactly why it is so interesting to us when we are working with spatial sound for VR.

The biggest challenge of VR audio is that you cannot predict which direction the viewer will be looking at in any given time. By using Ambisonics, we can design the whole sound sphere and the VR player decodes the sound to match the direction of the video in real time, decoding it into binaural for accurate headphone playback. The best part is that the decoding process is relatively light on processing power, which makes this a suitable option for mediums with limited resources such as smartphones.

In order to work with Ambisonics, we have two options: to record the sound on location with an Ambisonics microphone, which gives us a very realistic representation of the sound in the location and is very well suited to ambiance recordings, for example; or we can encode other sound formats such as mono and stereo into Ambisonics and then manipulate the sound in the sphere from there, which gives us great flexibility in post-production to use sound libraries and create interesting effects by carefully adjusting the positioning and width of a sound in the sphere.

There are many tools for integrating Ambisonic audio, one of them is Facebook spatial workstation, which it is the one used by VRI.

The Facebook 360 Spatial Workstation is a software suite for designing spatial audio for 360 video and cinematic VR. It includes plugins for popular audio workstations, a time synchronized 360 video player and utilities to help design and publish spatial audio in a variety of formats. There are quite a few techniques to record spatial audio such as using Ambisonic microphones. The Facebook 360 Spatial Workstation enables even normally recorded tracks and existing mono or stereo content to be panned in space in sync with the 360 video during the authoring process. It is completely fine even if you don't have an Ambisonic microphone to record spatial audio in location, as most of the sound editing process happens in the post-production phase, very similar to creating and editing sound

mixes for conventional films. The Facebook 360 tools leverage existing software and platforms to enable authoring for 360 videos and Cinematic VR.

The Facebook 360 Spatial Workstation is designed for use by professional sound designers. It is a set of tools that allows you to position sound tracks with what you see on a 360 video. The tools include a 360 video player that is synchronised to the DAW, which means you can instantly preview mixes as you go along. Imagine you have a dialogue track that is recorded with lavalier microphones for a character who is walking around the camera. The tools will allow you to place the sound, in time, with the movement of the character on screen. The magic happens when the final video is played back. The 360 Audio Engine takes care of head-tracking in real-time on the playback device — if the character is on your left, and you choose to look towards the character, the dialogue track corresponding to the character will automatically move to the front of the listener in real-time. The final audio is rendered with no noticeable latency across platforms, keeping the performance and quality uniform across platforms.

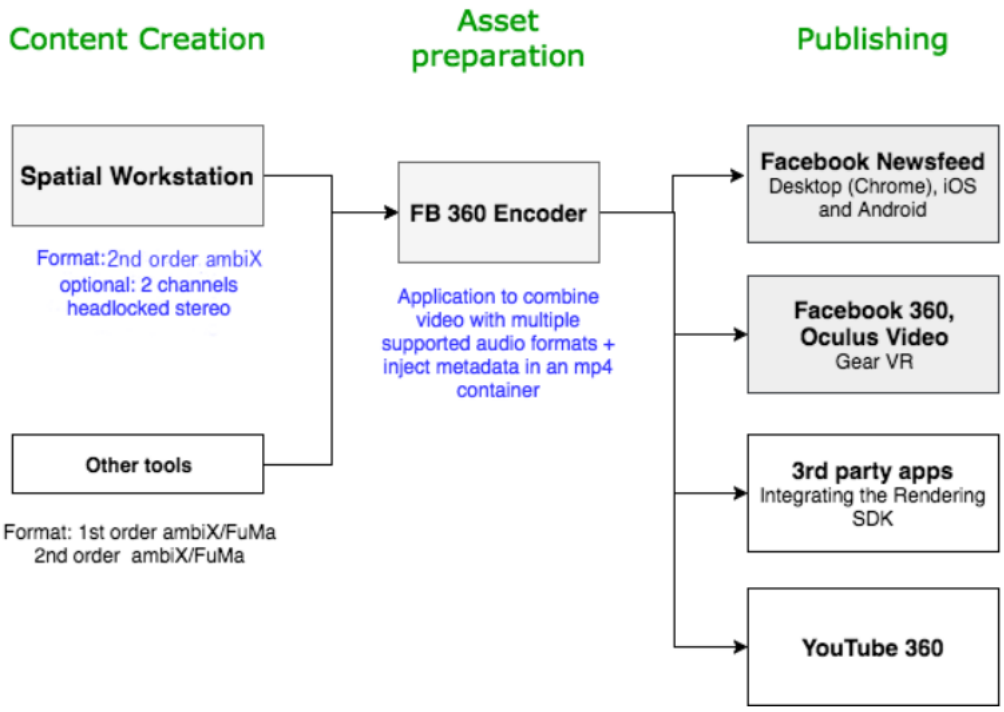


Figure 18: Workflow for Audio Creation, Preparation, and Publishing

The diagram above illustrates a typical end-to-end workflow focusing on sound design, asset preparation, multiplexing, a.k.a 'muxing' (combining audio and video) with final video and publishing to Facebook, Oculus or other supported apps.

If the application uses the Audio 360 Rendering SDK, the sound designer prepares a .tbe file, which is delivered separately. The Rendering SDK allows for the .tbe file to be played in sync with a video file. On Facebook and YouTube, the Encoder application creates an upload-ready video file.

The Spatial Workstation is a collection of DAW plugins that allow the sound designer to create an interactive spatial mix for 360 videos:

- Spatialiser plugin: This allows the sound designer to place a sound source in space. The source itself could be a mono source, an Ambisonics recording, or a multi-channel source such as a surround reverb. Non-mono sources act as a 'bed' while diegetic mono sources, such as dialogue and sound effects, are usually placed in a scene. Non-diegetic audio such as narration or background music is usually routed to the head-locked stereo bus. This makes it part of the final mix but not relative to head orientation.
- Control plugin: This plugin acts as the command centre, controlling how all audio is routed for real-time binaural playback over headphones. This plugin also manages global settings of features such as early reflections and mix focus.
- Video player: Spatial Workstation includes a 360 video player that is 'slaved' to the DAW timeline, and allows the sound designer to preview the mix with the 360 video in real time, either in VR or on the desktop. Desktop mode allows rotating the video with the keyboard or mouse, which will rotate the sound field instantly, providing direct feedback during the authoring stage.
- Converter plugin: Utility plugin offering the option to rotate a mix after it has been created, or output to other formats such as 4-channel ambiX.
- Loudness meter: Provides an overview of the loudness of the entire mix. Loudness for spatial mixes is considerably different than what is offered inside DAWs, which is usually for static content. Spatial audio for 360 videos is considerably more complex and this meter gives useful data that will prevent the final uploaded content from distorting when played back on the target device.

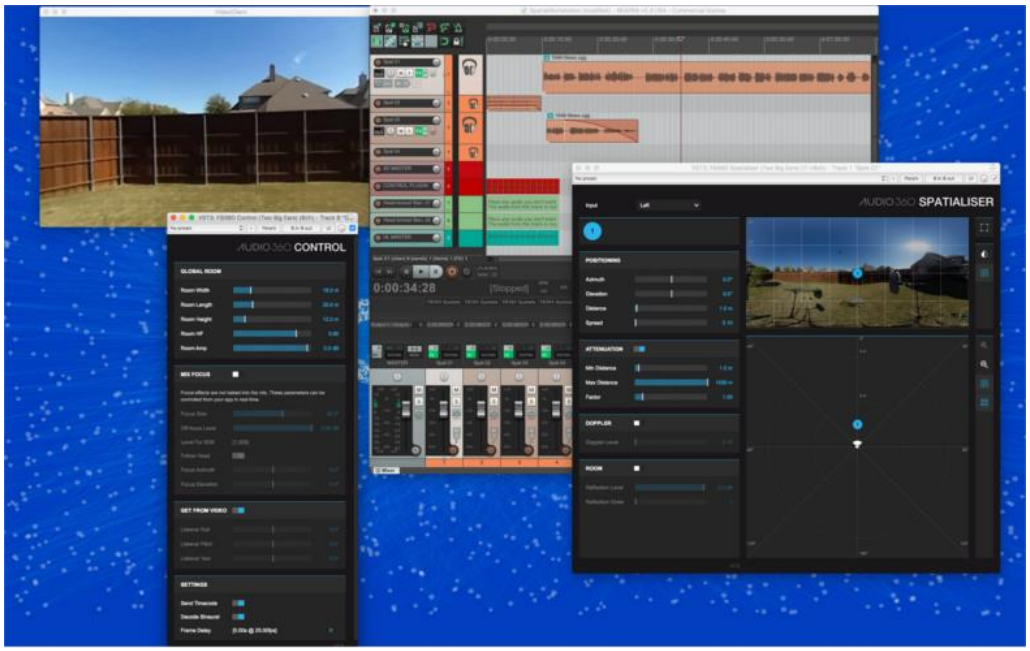


Figure 19: Facebook 360 Spatial Workstation

The FB360 Encoder application, takes a video file and combines the audio files into the video container, suitable for playback on Facebook and other supported platforms. Additionally, it also allows adding metadata to the file describing values for the Focus feature.

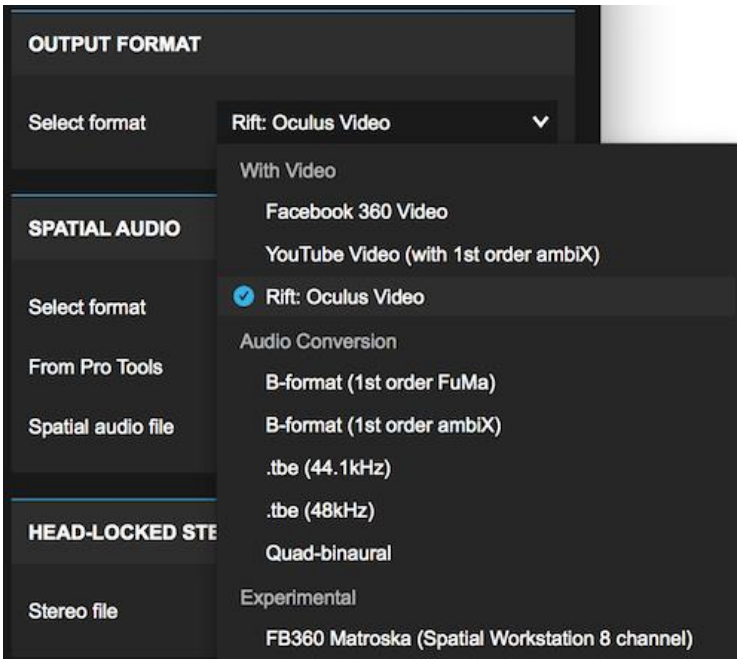


Figure 20: Facebook 360 Encoding and Asset Preparation

This process also injects relevant metadata into the tool, making the final asset ready for upload to supported platforms like:

- Facebook 360 Video format (8 or 10 channel audio): Facebook newsfeed, Oculus video on Gear VR
- .tbe format: Apps with Rendering SDK
- YouTube 360
- Other platforms with support for ambisonics or quad-binaural format. Note that these platforms have specific instructions for preparing assets

2.3.5 Rendering

A number of video players and technologies that support VR and 360° content rendering have been developed in the recent years.

WebVR and the newest version of it, WebXR, are open specifications for VR experiences in desktop and mobile web browsers, allow the use of simple VR headsets, such as the Google Cardboard, with smartphones that support a compatible web browser. The main goals of WebVR and WebXR are to render content into two screens (one for each eye) and capture motion control from phones' accelerometers and gyroscopes. They can also be used with advanced headsets (e.g. Oculus Rift) and most recent desktop browsers, such as Chrome, Chromium, Firefox, Microsoft Edge and Safari. WebVR and WebXR have been important technologies for bringing VR to the general public, as they only require a web browser, normally found in every computer and smartphone.

WebGL (Web Graphics Library) is an API written in JavaScript for rendering interactive 2D and 3D graphics in web browsers. WebGL is integrated with other web standards, therefore it is possible to execute GPU-based physics and image processing in web pages, as well as integrate other HTML elements into the applications.

Plenty of web-based players and platforms support 360° content, such as YouTube, Facebook and Vimeo. YouTube also supports 180° content, which is more affordable to produce. Other websites usually create their own customised players, using libraries such as Three.js and Video.js. Three.js is a JavaScript 3D library with cross-browser compatibility that contains an API for the creation and display of animated 3D computer graphics in web browsers. Video.js is an open source library for video support on the web and it supports HTML5, Flash video, YouTube and Vimeo content, VR and 360° videos, and other features, through its extensive plugin library. Developers can decide which features can be supported in their custom players through the addition of available plugins. Video.js supports video playback on desktops and mobile devices.

Many other 360° video players that can be installed on operating systems such as Windows, MacOS, Linux and Android are also available. Some examples of desktop players include the VLC player, which supports all the aforementioned operating systems, and other players for specific vendors, such as the Movies & TV app for Windows.

The flagship VR headsets Oculus and HTC Vive have several players available in their application stores, and they also offer bundled 360° video players with their platforms, such as the Oculus Video and the Viveport Video.

Polygonal 3D applications built on Unity, a popular cross-platform game engine used in VR development, can be exported as standalone players. For instance, when exporting a Unity executable application for Windows, the generated file is an .exe, while for Android it is an .apk. These applications can be executed in a wide range of devices, including desktops, smartphones and VR headsets. Unity applications can also contain diverse files generated on other tools, such as 3D assets from Google Poly (created on Google Tilt Brush and Google Blocks), 360° videos and volumetric content.

UAB, one of the TRACTION partners, is part of a H2020 consortium that built the open-source Immersive Accessibility (ImAc) player, described in (Montagud, 2018) and (Montagud, 2019), which employed many of the immersive features required by TRACTION.

Currently, the ImAc player supports 360° videos with accessibility content, including subtitling, (spatial) audio description and sign language interpreting. The player has been developed on standard compliant web-based technologies and components. Its architecture consists of the following layers:

- Immersive Layer: responsible for handling the presentation of both traditional and immersive content.
- Accessibility Layer: which manages the presentation of accessibility content.
- Assistive Layer: which assists in a more effective usage of the player, by enabling features such as voice control, preview and zooming.

- Synchronization Layer: a layer that ensures a synchronized consumption of content, both within each device and across devices in multi-screen scenarios.

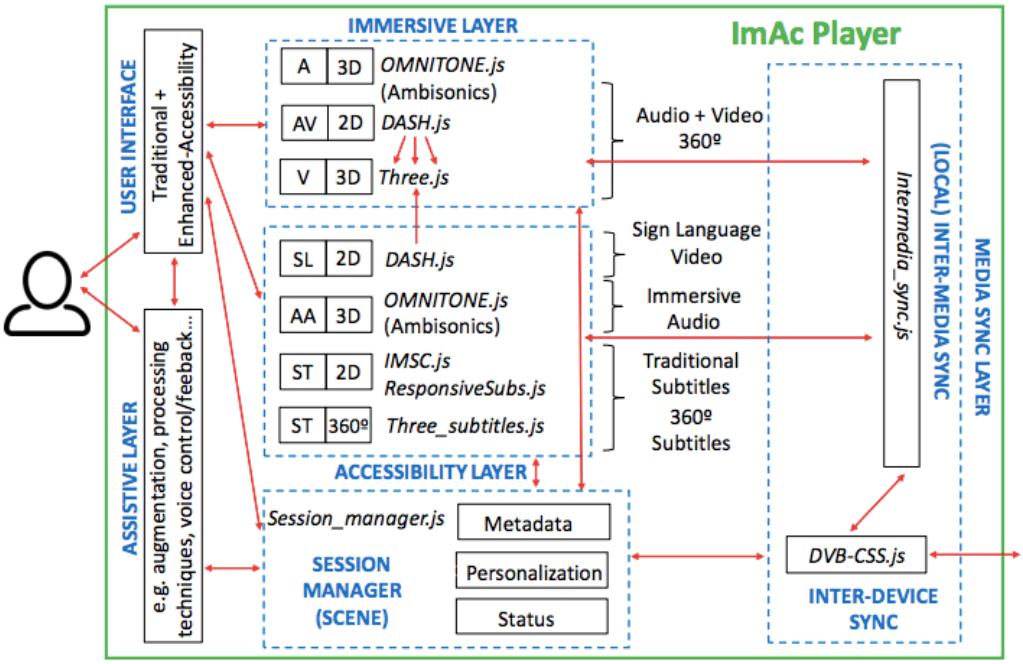


Figure 21: ImAc player layers (Montagud, 2019)

Other features for accessibility include:

- Voice Control: voice recognition and spoken feedback for/when executing commands.
- Menu types: two menu types are available, a traditional user interface (UI) and an enhanced-accessibility UI, which has a much larger size. Both UIs provide visual feedback to users after the execution of commands.
- Safe Area: size of the field of view in which visual elements are presented on screen.
- Indicator: a graphical element such as an arrow or radar that indicates where the target speaker is located in the 360° area. An auto-positioning mode is also available.
- Subtitling: It includes several settings, such as language, multiple fonts, sizes and colours, position (top, bottom), subtitle type (traditional, easy-to-read), and background (outlined or semi-transparent box). Emojis or text are available for representing sound effects. Multiple rendering modes are also being tested, using the video sphere, the field of view or the target speaker/object as the reference.
- Audio Description: Multiple languages available, three different spatial audio presentation modes and narratives (i.e. the direction of the sound can be modified), and three gain levels of the audio description track compared to the main audio track.
- Sign Language: Support to multiple languages, customizable position and size of the window, and speaker's identification methods.

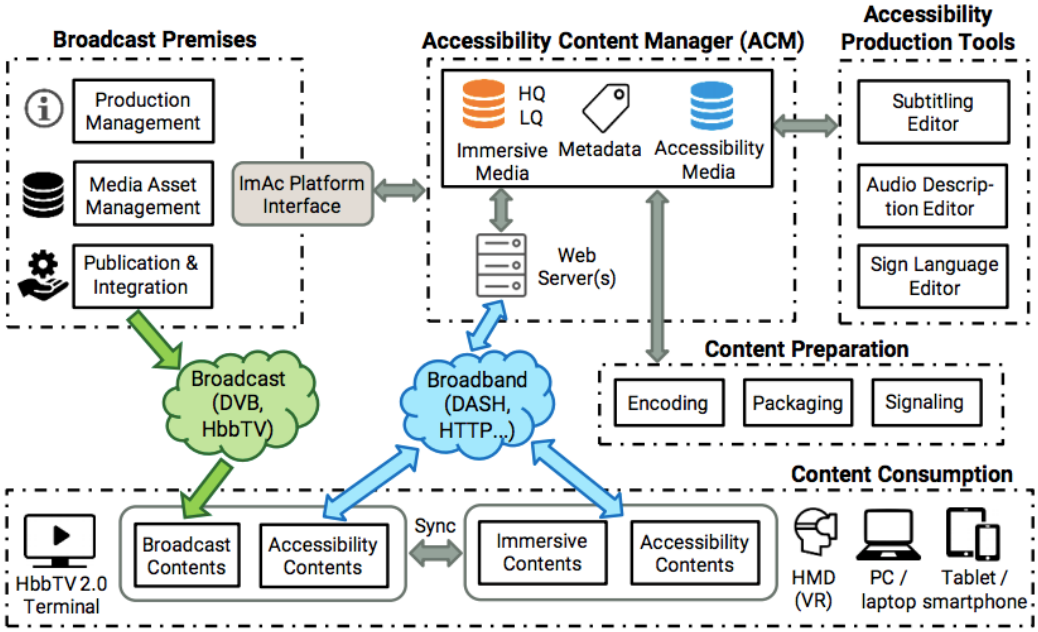


Figure 22: Diagram of the ImAc player architecture (Montagud, 2019)

Regarding media delivery and signalling solutions employed in the ImAc player, Dynamic Adaptive Streaming over HTTP (DASH) and the Hybrid Broadcast Broadband TV (HbbTV) were considered. DASH is used for the delivery of the immersive and accessibility content and to signalize their availability and features via standard-compliant extensions to the Media Presentation Description (MPD). HbbTV was used to support hybrid broadcast broadband multi-screen scenarios. In these scenarios, DASH is used for the delivery of the immersive and accessibility contents.

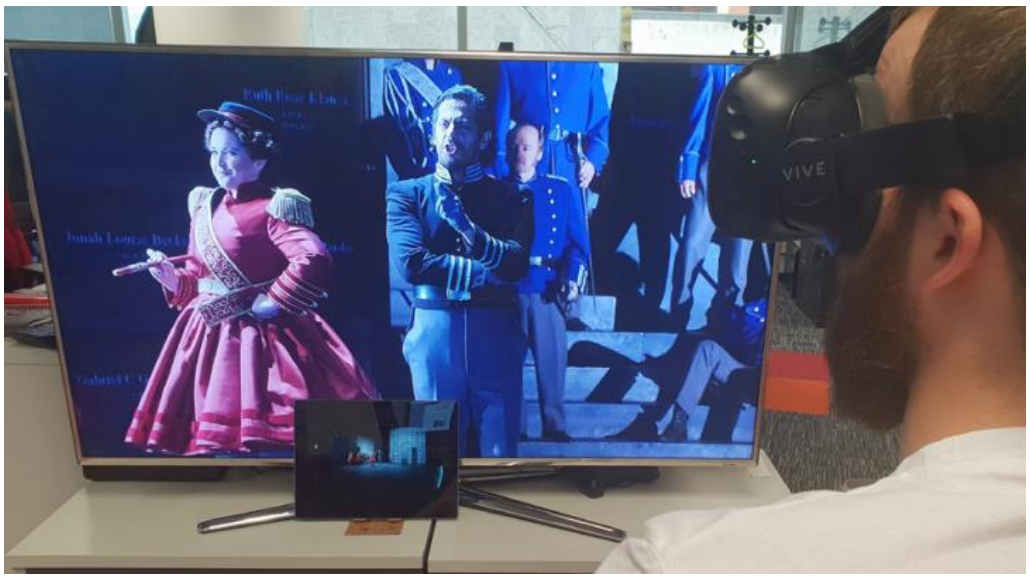


Figure 23: Multi-screen scenario with the ImAc player (Montagud, 2019)


Two key software components are required for the installation of the ImAc player: a web server (such as Apache Tomcat); and a web browser (such as Chrome). MongoDB and a Node.js server are also necessary for features such as the voice control and KPI metrics monitoring.

2.4 Communication Tools and Delivery of Media

This section describes communication and media delivery (and adaptation) technologies, as will be needed in the TRACTION project.

2.4.1 Communication Technologies

Real-time peer-to-peer communication on the today is still a somewhat difficult topic when it comes to browser support. There are however, several technologies which see some use for different areas of application. Chief among which is WebRTC which initially came out of Google which used it to implement their Hangouts videoconferencing product in 2010. Since then, the technologies and protocols behind WebRTC have been open-sourced and IETF and W3c have been tasked with maintaining the standard. As of today, WebRTC capabilities are included in every major browser.

While WebRTC was initially meant for peer-to-peer video and audio streaming, it also supports direct exchange or arbitrary data between users through a low latency channel similar to the way WebSockets work.

One issue with WebRTC is that even though it establishes direct peer-to-peer connections, a third-party server is still required for coordinating the connection and initial exchange of metadata between clients. This process is called signalling and is implemented through a selection of different protocol and thus not part of the WebRTC itself. Commonly, the signalling protocol can be implemented once two clients have exchanged the necessary data through the signalling channel, the third-party server is not needed anymore.

In order to maximise compatibility and applicability of WebRTC and allow for integration with other devices such as IP cameras or VoIP telephones one can employ a so-called gateway to mediate between the different types of clients. The gateway acts as a bridge between incompatible clients. More concretely, the latest web standards mandate that WebRTC connections are only to be established over secure links. However, some VoIP telephones might not support secure communication, therefore a gateway between them and browser clients is needed to establish a connection. Another use-case for a gateway would be to access a regular RTSP-enabled IP camera from a browser via WebRTC. The gateway would take the RTSP stream and relay it to browsers in a format they can interpret.

Another technology for real-time communication, albeit with a different use-case, are WebSockets. WebSockets provide a lightweight wrapper around standard TCP sockets that is accessible from the browser. Browser support for WebSockets is likely even better than support for WebRTC, but that is probably also because it is a much simpler technology. Another thing to be kept in mind is that while WebSockets allow for real time

communication, between server and clients, there is no way to establish a direct peer-to-peer connection between clients. There is always a third party in between clients to relay the messages. Also, WebSockets are not really suited for transferring large amounts of binary data. It is more suited for smaller text-based payloads and thusly the technology has been used to implement things such as chat functionality on websites or simple collaborative multiplayer browser games.

Several companies have integrated the use of WebSockets into their business model. One such company is Firebase, which was acquired by Google in 2014. Firebase offers online data storage and low-latency exchange of messages between server and browser clients on top of WebSockets. In that sense, it is more of a database and can be used for analytics and building entire interactive web experiences. Finally, Google has also opted to replace their Google cloud messaging platform for communicating with Android devices via push messaging with a Firebase-backed system. The service can also be used to send push messages to iOS devices, thereby creating a unified system for push messaging between browsers and mobile operating systems.

2.4.2 Multimedia delivery

Multimedia delivery, including movies, video clips, and live content, can be processed in networks in real or non-real time. The main methods used for multimedia content delivery are downloading and streaming. The Hyper Text Transport Protocol (HTTP) and the File Transfer Protocol (FTP), which are based on the Transmission Control Protocol (TCP), can be used for content downloading, allowing users to watch content during download.

Traditional content streaming requires a multimedia streaming server, which delivers the exact amount of data for playback and the video file is not downloaded. The Real Time Streaming Protocol (RTSP) sends content from the server to the client at a fixed real-time rate. Adaptive streaming techniques support users with good or bad connections, as content is stored in small chunks with different encoding rates/resolutions, delivered according to clients' connection quality and reducing the need for buffering.

Content quality in streaming needs to be constantly assessed by Quality of Service (QoS) and Quality of Experience (QoE) metrics, so content can be delivered at higher quality given network constraints. QoS metrics are related to data transport and network parameters, such as packet loss, delay, jitter, round trip time, etc. QoE metrics focus on the quality perceived by users. Another important metric for video quality assessment is the Peak Signal to Noise Ratio (PSNR), which is the ratio between the maximum power of a signal and the power of the signal's noise, and is used to measure the quality of video reconstruction during video compression.

Several standards and solutions support adaptive streaming with QoS provisioning, aiming to deliver content with efficient use of network resources and at high quality. The Moving Picture Experts Group (MPEG) developed the Dynamic Adaptive Streaming over HTTP (MPEG DASH) based on previous standards, such as the Adaptive HTTP Streaming (AHS) and the 3GP-DASH. Move Networks developed a HTTP-based streaming technology that does not need a dedicated streaming server, and different quality streams are

delivered according to bandwidth availability. Microsoft also has an adaptive streaming solution called IIS Smother Streaming, while Adobe developed the HTTP Dynamic Flash Streaming, Apple created the HTTP Live Streaming, and the online video providers Hulu, Netflix and YouTube integrated adaptive streaming to their video players.

With the introduction of demanding new high-definition video formats such as 4K, 8K and 360°, multipath solutions aim to split the traffic over available networks, as many devices support multiple interfaces (e.g. cellular, WiFi, Bluetooth, 5G, etc.). Multipath TCP (MPTCP), a standard by the Internet Engineering Task Force (IETF), extends the TCP protocol to use multiple paths with a single transport connection.

Traditional digital television broadcast technologies such as DVB-T, ATSC, ISDB-T and DTMB are also evolving to integrate broadband content for a synchronised experience. Integrated Broadcast-Broadband (IBB) systems, such as HbbTV, Hybridcast, ATSC 3.0 and Ginga, allow users to receive on-line broadband information related and synchronised with the broadcast content. IBB systems can easily add the provision of accessibility (e.g. sign language), social networking, multi-device, multi-view and tailored advertising to traditional broadcast.

2.4.3 Multimedia Content Adaptation

Several multimedia content adaptation techniques have been proposed over the years, reducing buffering times and adapting content to devices with various screen sizes and network capabilities. These techniques can be incorporated into the technologies used in TRACTION, in order to increase perceived user quality even in constrained devices and network conditions.

The region of interest-based adaptive scheme (ROIAS) performs adaptation at the level of regions within clip frames, based on user interest obtained from eye-tracking monitoring (Muntean, 2008). ROIAS adjusts the quality of those regions from the multimedia frames the viewer is the least interested in, if necessary due to network conditions. Regions in which the viewers are the most interested in, either do not change or involve little adjustment, resulting in high overall end-user perceived quality.

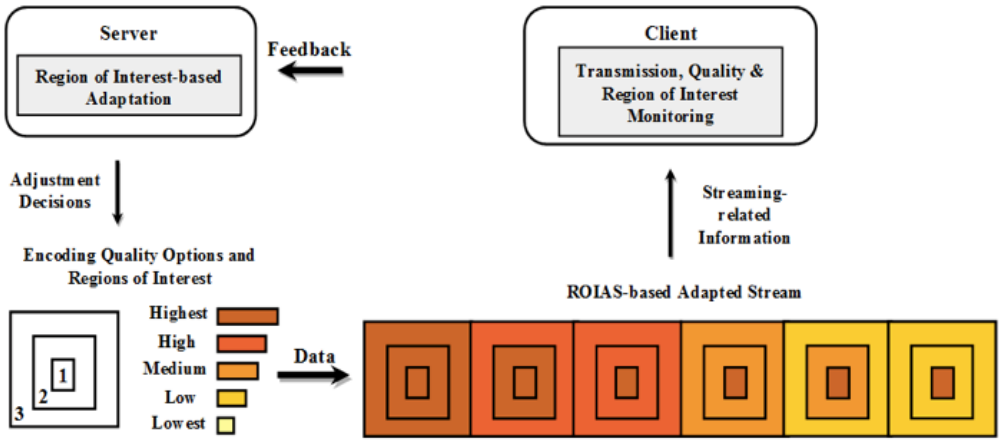


Figure 24: The region of interest-based adaptive scheme (ROIAS) (Muntean, 2008)

The Dynamic Adaptive Streaming over HTTP (MPEG-DASH) is a well-known adaptive bitrate streaming technique, standardized in 2011. MPEG-DASH has two main components: The Media Presentation and the Media Presentation Description (MPD). Media Presentation is a sequence of one or more segments, incorporating periods, adaptation sets, and representations, which break up the media from start to finish. The MPD is a document defined using the eXtensible Markup Language (XML), and it identifies the various content components and the location of all alternative segments, providing the relationship between them.

Several algorithms have been based on MPEG-DASH since its conception. In the Fair, Efficient, and Stable adapTIVE algorithm (FESTIVE), proposed by (Jiang, 2014), clients trying to maintain a stable buffer have to wait when the buffer size is over a desired value, and thus get an inaccurate picture of the network bandwidth. A random jitter is added to the waiting time in FESTIVE, avoiding synchronization effects among clients and improving the bandwidth estimation. A stateful bitrate selection is used to aggressively probe the channel at lower bitrates and remain conservative when the bitrate is already high. Quality switches are limited to one quality level at a time, so the algorithm penalizes further upward switches if the client's quality has increased in the recent past. The bandwidth estimate is performed with a long-term harmonic mean, using the measured throughputs of the last 20 segments.

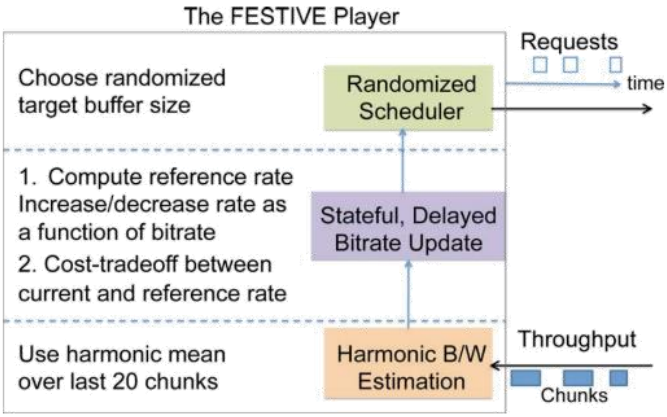


Figure 25: FESTIVE algorithm (Jiang, 2014)

The Probe and Adapt approach (PANDA), introduced by (Li, 2014), aims to not underestimate the available bandwidth, probing the network bandwidth in a TCP-like fashion, increasing the rate slowly and significantly decreasing it in case of sudden bandwidth drops (and a consequent rebuffering risk). Panda initially estimates the capacity bandwidth, which it then smooths with an Exponentially Weighted Moving Average (EWMA) filter. The smoothed value is then associated with the adaptation having the closest achievable bitrate (i.e., the highest bitrate, which is lower than the bandwidth estimate minus a safety threshold). Finally, the algorithm schedules the new download request, waiting for some time before requesting the next segment, so as to regularise the on-off behaviour of clients and improve fairness.

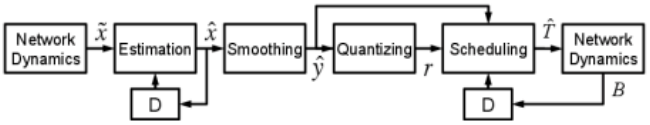


Figure 26: PANDA algorithm (Li, 2014)

The Model Predictive Control (MPC), suggested by (Yin, 2015), employs the control theory to improve video streaming quality. The QoE function proposed (which is the objective of the maximisation) linearly combines: picture quality (using the bitrate of the segment), its variations, the frequency and length of rebuffering events, and the initial startup delay. The algorithm performs the optimisation for a given throughput prediction, which can be more or less pessimistic. Using the prediction for the following N steps, MPC iterates over all possible policies on the given time and chooses the one that maximises its reward function. Complexity can be reduced by tabulating values or approximating the prediction.

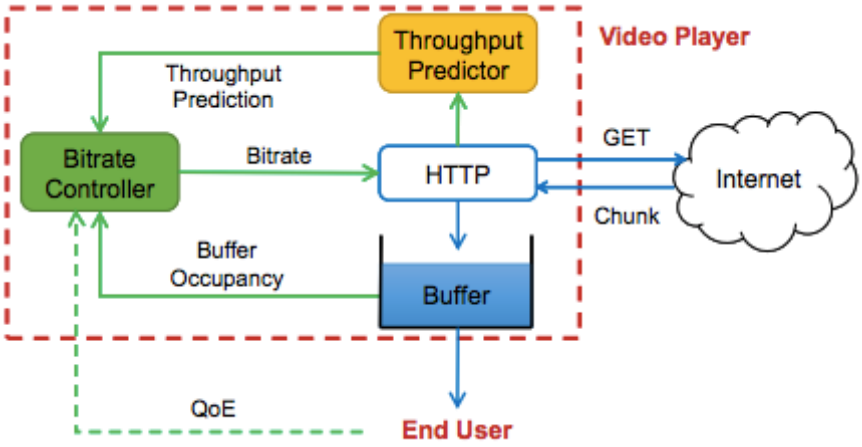


Figure 27: MPC model (Yin, 2015)

The Deep Q-Learning Framework for DASH (D-DASH) is a learning-based adaptation algorithm proposed by (Gadaleta, 2017) that models the video streaming problem as a Markov Decision Process (MDP) and uses reinforcement learning to gradually learn the optimal rate adaptation policy by trial and error. It does not estimate the future capacity explicitly, but learns the long-term reward associated with each action in the current state. D-DASH employs reinforcement learning for deep Q-Learning, a technique that mitigates dimensionality, which affects reinforcement learning agents, by exploiting a neural network to approximate the long-term reward.

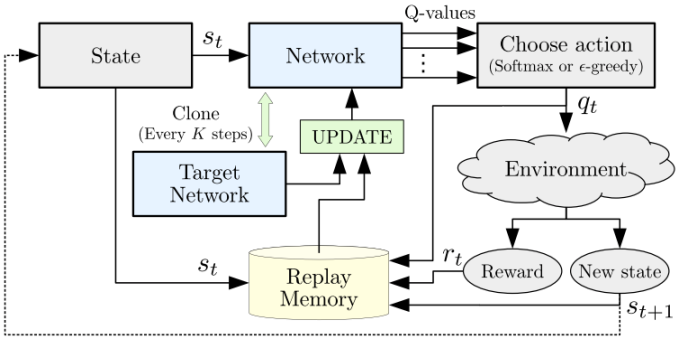


Figure 28: D-DASH diagram (Gadaleta, 2017)

A bandwidth-efficient Field of View (FoV)-aware streaming adaptation scheme was proposed by (Hosseini, 2016) aiming to address the high bandwidth demands of 360° VR videos. The authors exploit the semantic link of the MPEG-DASH Spatial Relationship Description (SRD) with a user's viewport to provide dynamic view awareness in the context of VR videos. 360° VR videos and the underlying 3D geometry are divided into spatially partitioned segments/tiles in the 3D space, and are adapted with more or less priority, according to the regions the user are more likely to look. Initial results indicate up to 72% less consumption of bandwidth on 360° VR video streaming without major impact on quality.

2.5 Other Research and Innovation activities linked with the project

Each partner in TRACTION has a proven experience and has previously executed research and innovation activities that will directly contribute to this research and innovation action. The research centres and universities involved in the project have an extensive expertise in their respective technical fields. A sample of the previous projects that have a direct impact in TRACTION at various levels (existing software, knowledge, methodologies, user labs, etc.) are described in the DoA and include:

- **Cloud-LSVA**: Cloud Large Scale Video Analysis. H2020 Project ID: 688099.
- **VI-DAS**: Vision Inspired Driver Assistance Systems. H2020 Project ID: 690772.
- **ImAc**: Immersive Accessibility. H2020 Project ID: 761974.
- **EasyTV**: Easing the access of Europeans with disabilities to converging media and content. H2020 Project ID: 761999.
- **VRTogether**: An end-to-end system for the production and delivery of photorealistic social immersive virtual reality experiences. H2020 Project ID: 762111.
- **2-IMMERSE**: Creating and Delivering Shared and Personalised Multi-Screen Broadcast and Broadband Experiences. H2020 Project ID: 687655.
- **NEWTON**: Networked Labs for Training in Sciences and Technologies for Information and Communication. H2020 Project ID: 688503.

Moreover, other significant European projects will be considered during the lifetime of TRACTION. Relationships will be built with different ongoing activities and results from these projects will be leveraged when possible. Some of the most relevant ongoing and recently finalised H2020 projects include, but are not limited to:

- **BEATIK**: BEATIK (Collaborative Digital Scores Platform for Classical Music) will provide an integrated platform and eco-system to help all musicians and the institutions they belong to, such as orchestras, conservatoires and music schools, from performance preparation to sharing annotations and score distribution. Grant agreement ID: 822897 (SME Instrument), more information: http://www.beatik.com/
- **BEYOND4.0**: Inclusive Futures for Europe BEYOND the impacts of Industrie 4.0 and Digital Disruption aims to help deliver an inclusive European future by examining the impact of the new technologies on the future of jobs, business models and welfare. Grant agreement ID: 822296, more details: http://beyond4-0.eu/
- **CICERONE**: Creative Industries Cultural Economy pROduction NEtwork provides policymakers with a unique and innovative perspective from which to understand the cultural and creative industries (CCIs). Previous analyses have mapped the location and distribution of the CCIs; CICERONE innovates by exploring the flows of products and ideas that generate the economic and cultural values in and of places, and which also account for the disparities between them. Moreover, CICERONE explores the evolving relationships between cultural and the economy. GAID: 822778, further details: cordis.europa.eu/project/rcn/218760/factsheet/en
- **CO3**: CO3 (Digital Disruptive Technologies to Co-create, Co-produce and Co-manage Open Public Services along with Citizens) aims at assessing the benefits and risks of disruptive technologies, namely: blockchain, augmented reality, geolocated social network, liquid democracy tools and gamification, in the co-creation, co-production and co-management of public services with citizens as PAs partners. Grant agreement ID: 822615, further information: https://www.projectco3.eu/
- **DETECt**: DETECt -Detecting Transcultural Identity in European Popular Crime Narratives addresses the formation of European cultural identity as a continuing process of transformation fostered by the mobility of people, products and representations across the continent. Grant agreement ID: 770151, more information can be found here: http://www.detect-project.eu/
- **DiCrEd**: DiCrEd is a multidisciplinary project aiming to address these limitations through the integration of the innovative instruments offered by digital humanities into “The Works of Giuseppe Verdi”. DiCrEd employs the digital tools designed by the Edirom project to develop an applied model of digital critical edition with an interactive system of fruition of the score, and an applied model of digital critical edition of preparatory materials. Grant agreement ID: 800280 (MSCA-IF), more information: cordis.europa.eu/project/rcn/214512/factsheet/en
- **DIDONE**: The Sources of Absolute Music: Mapping Emotions in Eighteenth-Century Italian Opera. The results will be applicable to three main fields: (i) opera performance; (ii) analysis and interpretation of other types of music; and (iii) composition in several scenarios, from film soundtracks to creation by Artificial Intelligence. Grant agreement ID: 788986 (ERC Advanced Grant), more details: http://www.didone.eu/

- **DISCE**: The DISCE (Developing Inclusive & Sustainable Creative Economies) project is set to improve and enhance the growth, inclusivity and sustainability of the cultural and creative industries (CCIs) in the EU. DISCE will bring out recommendations for actors how to react, function and decide in specific situations to promote inclusive growth and progress on the sustainable development in the field of CCIs. Grant agreement ID: 822314, further information: https://disce.eu/
- **EMOTIVE**: The principal objective of the EMOTIVE project is to research, design, develop and evaluate methods and tools that can support the cultural and creative industries in creating Virtual Museums which draw on the power of 'emotive storytelling'. Grant agreement ID: 727188, more details: http://www.emotiveproject.eu/
- **ERIN**: ERIN offers a network analysis, investigating the cultural articulation of national identity in 19th-century Europe as found in the musical works of Irish poet-songwriter Thomas Moore. Grant agreement ID: 658376 (MSCA-IF), further details: http://blogs.qub.ac.uk/erin/
- **gE.CO Living Lab**: Generative European Commons Living Lab aims at creating a platform for bringing together and supporting formal groups or informal communities of citizens who manage fab-lab, hubs, incubators, co-creation spaces, social centres created in regenerated urban voids. Grant agreement ID: 822766, further information: https://generative-commons.eu/
- **GIFT**: The GIFT project suggests creating meaningful personalization through digital gifting and emotional appropriation: Designs for allowing visitors to create their own museum tours as digital "mixtapes", and to play with technologies that measure emotional responses to artwork as a playful re-appropriation of museum spaces. Grant agreement ID: 727040, further detail can be found here: http://gift.itu.dk/
- **GLOBALINTO**: GLOBALINTO will provide new measures of intangible assets at the firm level, filling an important gap in measurement which has restricted statistical production, micro-based analysis and evidence-based policymaking, building on earlier work by the EU FP7 Innodrive project to create refined and validated intangible capital data and indicators that can be implemented in official statistics production. Grant agreement ID: 822259, further details: ps.au.dk/forskning/forskningscentre/dansk-center-for-forskningsanalyse/globalinto/
- **GoldOpera**: The goal of this project is the recovery and analysis of operatic renditions created from the innovative drammi giocosi by Goldoni, today nearly forgotten, to finally supply a comprehensive picture of the interaction between text, music, and theatre that fuelled the radical evolution of Classical opera in the later 18th century. Grant agreement ID: 701269 (MSCA-IF), more information: http://www.unive.it/data/16033/
- **GROWINPRO**: The project is meant to investigate the causes of the socio-economic slowdown in Europe and to propose an integrated policy package able to sustain an inclusive and welfare-enhancing process of growth resilient to climate change and population aging, building on the results from the H2020 project: ISIGrowth http://www.isigrowth.eu/. Grant agreement ID: 822781, more information: http://www.growinpro.eu/
- **iMARECULTURE**: iMARECULTURE (Advanced VR, iMmersive serious games and Augmented REality as tools to raise awareness and access to European underwater CULTURal heritagE) is focusing in raising European identity awareness using maritime and underwater cultural interaction and exchange in Mediterranean Sea. GA ID: 727153, more details: https://imareculture.eu/

- **MICROPROD**: MICROPROD (Raising EU Productivity: Lessons from Improved Micro Data) will examine the empirical observation that productivity growth in the developed world has slowed down in the past decade despite both technological innovation continuing as well as greater openness to trade. Grant agreement ID: 822390, more information: http://www.microprod.eu/
- **Mingei**: Mingei (Representation and Preservation of Heritage Crafts) will explore the possibilities of representing and making accessible both tangible and intangible aspects of craft as cultural heritage. Heritage Crafts involve craft artefacts, materials, and tools and encompass craftsmanship as a form of Intangible Cultural Heritage. Grant agreement ID: 822336, more details: http://www.mingei-project.eu/
- **MIP-Frontiers**: The project will create a multidisciplinary, transnational and cross-sectoral European Training Network for Music Information researchers, in order to contribute to Europe's leading role in this field of scientific innovation and accelerate the impact of innovation on European products and industry. Grant agreement ID: 765068 (MSCA-ITN), more information: https://mip-frontiers.eu/
- **NETCHER**: NETCHER (NETwork and digital platform for Cultural Heritage Enhancing and Rebuilding) seeks to address the complex challenge of harmonising and bringing together worthy, but often disconnected initiatives by using a participative approach that will result in the establishment of a structured network drawing together a broad range of players such as international bodies, umbrella organizations, national governments, researchers, public policy makers, NGOs, as well as public and private foundations. GA ID: 822585, further details: https://netcher.hypotheses.org/
- **NewsEye**: The NewsEye project addresses a number of challenges, which will result in significant scientific advances, in several directions: in text recognition, text analysis, natural language processing, computational creativity and natural language generation, with regard to historical newspapers but also more universally; in digital newspaper research, addressing a number of editorial issues like OCR and article separation; in digital humanities, in respect to huge amounts of text material, availability of useful tools and possibilities of searching and browsing; in history, in terms of analyzing historical assets with new methods across different language corpora. Grant agreement ID: 770299, more information: https://www.newseye.eu/
- **NoVaMigra**: NOVAMIGRA – Norms and Values in the European Migration and Refugee Crisis will enhance the European knowledge base on these issues with a unique combination of social scientific and historical analysis, as well as legal and philosophical normative reconstruction and theory. Grant agreement ID: 770330, further information: https://novamigra.eu/
- **OpenDrama**: *The Digital Heritage of Opera in the Open Network Environment* aimed at the definition, development and integration of a novel platform to author and to deliver rich cross-media digital objects of lyric opera and other vocal dramatic music, opening this heritage to a dimension of learning, exploring and entertainment. Grant agreement ID: IST-2000-28197
- **PERICLES**: PERICLES (PrEseRvIng and sustainably governing Cultural heritage and Landscapes in European coastal and maritime regionS) promotes sustainable governance of cultural heritage in European coastal and maritime regions through the development of a theoretically grounded, multi-actor participatory framework. Grant agreement ID: 770504, more details: www.pericles-heritage.eu/

- **PLUGGY**: Pluggable Social Platform for Heritage Awareness and Participation (PLUGGY) will support citizens in shaping cultural heritage and being shaped by it. PLUGGY will enable them to share their local knowledge and everyday experience with others. GA ID: 726765, more information: www.pluggy-project.eu/
- **PLUS**: PLUS aims to address the main features of the platform economy's impact on work, welfare and social protection through a ground-breaking trans-urban approach. Grant agreement ID: 822638, further information can be found here: https://project-plus.eu/
- **QualiChain**: QualiChain targets the creation, piloting and evaluation of a decentralised platform for storing, sharing and verifying education and employment qualifications and focuses on the assessment of the potential of blockchain technology, algorithmic techniques and computational intelligence for disrupting the domain of public education, as well as its interfaces with private education, the labour market, public sector administrative procedures and the wider socio-economic developments. Grant agreement ID: 822404, further information can be found here: https://qualichain-project.eu/
- **RePAST**: RePAST aims at investigating how European societies deal with their troubled pasts today through the analysis of conflict discourses rooted in those pasts, with a view on the impact of those discourses on European integration. It will implement actions and propose strategies, both at the levels of policy-making and civil society, for reflecting upon these discourses to strengthen European integration. Grant agreement ID: 769252, more details: https://www.repast.eu/
- **SILKNOW**: SILKNOW aims to produce an intelligent computational system that goes beyond current technologies in order to improve our understanding of European silk heritage. This legacy will be studied, showcased and preserved through the digital modelling of its weaving techniques. Grant agreement ID: 769504, further details: http://silknow.eu/
- **TECHNEQUALITY**: TECHNEQUALITY (Technological inequality – understanding the relation between recent technological innovations and social inequalities) will provide forecasting of labour market consequences of technological innovations, explore new ways of measuring automation rates in European countries, explain how technological innovations are most likely to shape societal inequalities, study the role of various forms of education and innovative forms of social welfare in maximizing growth and reducing inequality, and assess the consequences of automation for public finances. TECHNEQUALITY will serve as a foundation for a better understanding of technologically driven social inequalities and a catalyst for new research and also set the agenda for policy debates on societal consequences of technological developments. Grant agreement ID: 822330, further details: http://www.technequality-project.eu/
- **TRACES**: Transmitting Contentious Cultural Heritages with the Arts: From Intervention to Co-Production (TRACES) aims to provide new directions for cultural heritage institutions to contribute productively to evolving European identity and reflexive Europeanization. To do so, it deploys an innovative ethnographic/artistic approach, focused on a wide range of types of contentious heritage. Grant agreement ID: 693857, further details: http://www.tracesproject.eu/
- **TROMPA**: TROMPA will enrich and democratise European publicly available musical heritage through a user-centred co-creation setup. For analysing and linking music data at scale, the project will employ and improve state-of-the-art technology. Music-loving citizens will cooperate with the technology, giving

feedback on algorithmic results, and annotating the data according to their personal expertise. Following an open innovation philosophy, all knowledge derived will be released back to the community in reusable ways. This enables many uses in applications which directly benefit crowd contributors and further audiences. Grant agreement ID: 770376, more information: https://trompamusic.eu/

- **TRIPLE**: TRIPLE, Transforming Research through Innovative Practices for Linked interdisciplinary Exploration, enables researchers to discover and reuse SSH data, but also other researchers and projects across disciplinary and language boundaries. It provides all necessary means to build interdisciplinary projects and to develop large-scale scientific missions. Grant agreement ID: 863420, further details can be found in the following webpage: https://cordis.europa.eu/project/rcn/224734/factsheet/en
- **TrueTalent**: TrueLinked is a Danish start-up with an innovative business model to disrupt the classical music and opera industry through the use of new technologies to bring more transparency and efficiency to the planning and programming of live performances. Grant agreement ID: 807821 (SME Instrument), more information: http://truelinked.com/
- **URBANA**: UrbanA takes up the challenge of synthesizing and brokering the knowledge and experience generated in EU-funded projects, many of which have identified interventions that address grand societal challenges, of which urban inequalities and social exclusion across different contexts. Grant agreement ID: 822357, more information: https://urban-arena.eu/
- **VHH:** VHH (Visual History of the Holocaust: Rethinking Curation in the Digital Age) is an innovation action that focuses on the digital curation and preservation of film records relating to the discovery of Nazi concentration camps and other atrocity sites. We combine state-of-the-art concepts and practices from information science, museum pedagogy and digital storytelling to design a new approach for the engagement with a significant aspect of European audio-visual heritage. Grant agreement ID: 822670, more information: www.vhh-project.eu
- **ViMM**: Virtual Multimodal Museum (ViMM) proposes a major, high-impact CSA across the field of Virtual Museums, within the overall context of European policy and practice on Virtual Cultural Heritage, to define and support high quality policies, strategic and day-to day decision making, the utilisation of breakthrough technological developments such as VR/AR and to nurture an evidence-based view of growth and development impacted by VM. Grant agreement ID: 727107, more details: http://www.vi-mm.eu/

3 Technical Requirements

This section summarises the requirements gathered during the initial period of the project, which will be extended in subsequent versions of the deliverable. First, the methodology that has been followed (and that will be followed) is described. Then, the trials are introduced. And finally, a set of requirements are proposed, for each of the toolsets that have been identified.

3.1 Methodology

The project follows a user-centric approach to gather requirements, with the goal of informing the design of the TRACTION technology for the three trials. In the first year, the approach is to build a common understanding of the project between the team members, identify potential users, and identify and refine a set of user requirements for toolset design. First, some initial conversations, moderated by François Matarasso (FM), took place between the project partners. In parallel, the technical team defined a number of potential solutions that were discussed in the first General Assembly. These will be followed by more structured focus groups taking place in May 2020. This section details each of the phases.

The first phase was to clarify the aim of each trial in order to specify the objectives and activities in more details. To this end, a simplified logical framework model was used, followed by a number of telco meetings with each trial team to develop a fuller understanding of their situation and plans. The results of such discussions are the better definition of each use case (see next section) and the identification of the users of the system and the contact point to continue the more structured conversations. The following tables report initial results that define some key concepts and categories of participants (by which is meant people who are involved in the TRACTION trials, not audiences). The categories are divided into three groups, with subdivisions:

Professionals artists	Creative and artistic team
	Production team
	Co-creation team
Non-professional artists	People involved in the co-creation of the community opera
	Participating organisations (e.g. school, prison, NGOs)
Audiences	Live audiences
	Online audiences
	Event audiences

A further categorisation is proposed:

Professional users	Someone who uses the technology as part of an artistic process or creation to enhance the experience of the audience (e.g. a sound engineer, a director, etc.). audiences
Non-professional users	Someone who uses the technology to access and enjoy an artistic output.

Each of these groups will be involved in the research and evaluation at some stage, but most cannot be involved at the outset – and specifically – defining needs and expectations of the UX, because they are either not identified or do not have sufficient knowledge or both. Therefore, this part of the work will need to begin with the trial leaders and then, depending on what we learn from experience, will reach out further as follows.

Project partners	Individuals already working in INO, LICEU and SAMP and involved in TRACTION
Project partners' associates	A wider range of people (e.g. conductor, artistic director, stage manager etc.) with whom the partner has previously worked, but who is not necessarily involved in the trial, and who may draw on wider experience of opera creation and new technology to contribute to the focus group
Project partners' audiences	People drawn from the trail leader's existing audiences (i.e. people who already attend and enjoy opera) to find out the extent of their experience with new technology and their interest/expectations of it in future

During this first phase (between January and March of 2020), a number of other activities took place with the goal of getting to know the project members, their objectives, and their background. These activities included the writeup of a guide on methodologies for user experience research, the conduction of an expertise survey with TRACTION trial members to understand their expertise and familiarity with user studies methodologies, and the in-person discussion with the project members around specific technological innovations that the project can bring. At the end of this phase, we consulted with trial leaders to identify potential user groups for the two sets of studies, such as Opera producers, professionals involved in the trials, and community members.

The second phase is to follow a more structured approach that helps the technology team (WP2) to identify the requirements for the trails (WP3), proxied by the user experience experts (WP4). This will include two rounds two rounds of focus groups and expert interviews (see timeline in **Figure 29**). The first set (FG1) will be conducted with the TRACTION project trial leaders, and the second set (FG2) will be conducted with users that include opera producers and community members.

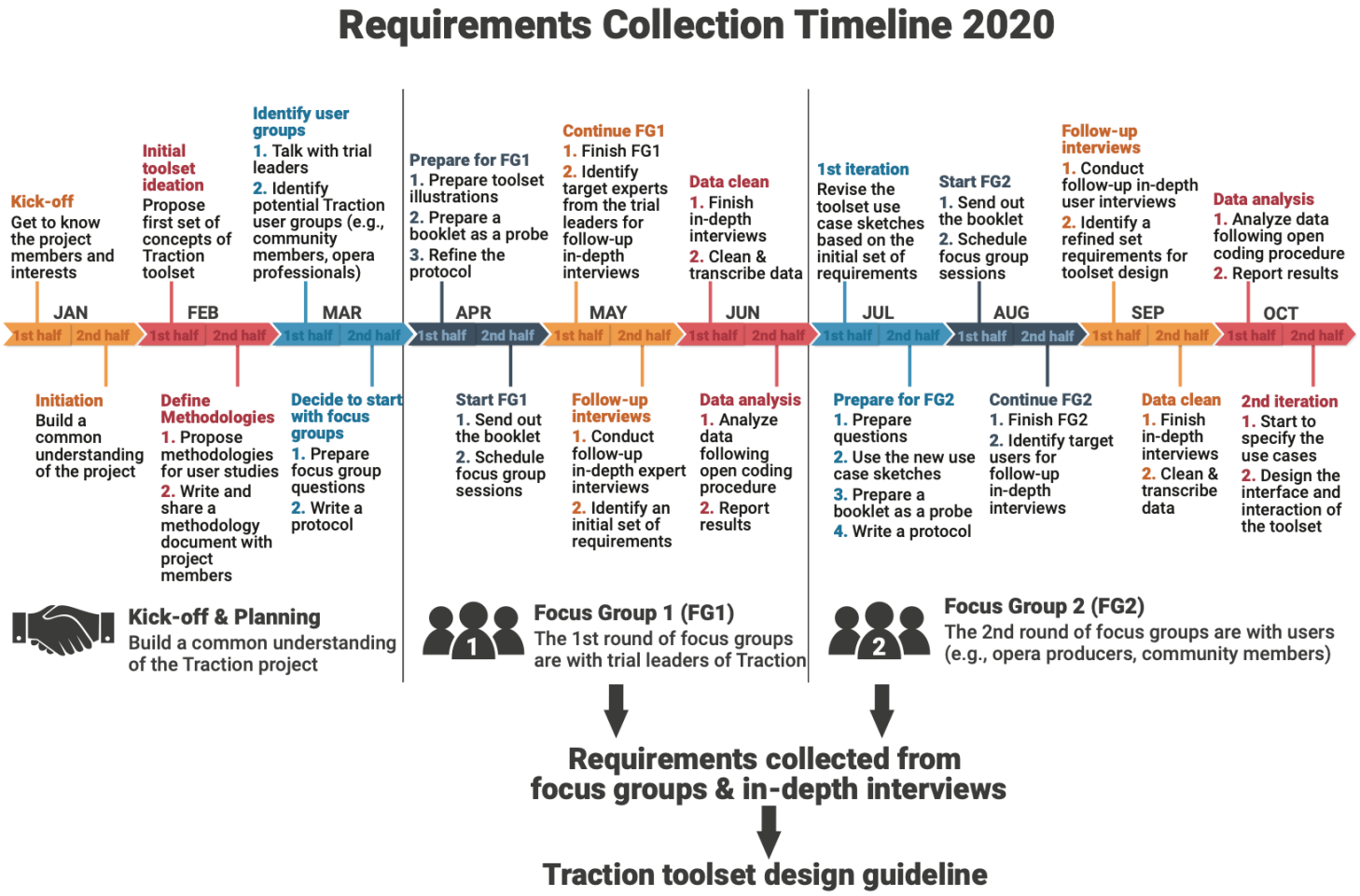


Figure 29: Timeline for gathering technical requirements

The FG1 stage is taking place between March and July of 2020. The goal of FG1 is to understand the current practices in Opera production of the trial leaders, consider trial leaders' goals and requirements for technology, and receive feedback on technology design concepts. In March, we began the preparation for the focus groups by writing questions, and integrating these questions into a formal protocol. In April, we refined the focus group protocol, and developed a booklet probe for the focus groups. In addition, we prepared a set of toolset illustrations to receive feedback from participants. At the end of April, we will send out the booklet and schedule the focus group sessions. We plan to conduct the focus groups between the end of April and beginning of May. After conducting the initial set of focus groups, we will identify target experts from the trial leaders for follow-up in-depth interviews. We will conduct these interviews at the end of May and beginning of June, and use a combination of the focus group and expert interview findings to identify an initial set of user requirements for the technology design. In June, we plan to also conduct a formal analysis of the FG1 data. We will clean and transcribe the focus group and expert interview data, analyse it using open coding procedures, and report the results. In July, we will use this formal analysis to revise the toolset use cases sketches based on the initial set of requirements.

The FG2 stage will take place between July and October of 2020. The goal of FG2 is to design the interface and interaction of the toolsets based on focus groups with opera producers and community members. The FG2 stage will mirror the process of FG1. In July we will prepare a focus group protocol, focus group protocol questions, and prepare a booklet probe. In August, we will send out the booklet, schedule FG2 sessions, and conduct the focus groups, using these to identify experts for follow up in-depth interviews. In September we will conduct the follow up expert interviews, and identify a refined set of requirements for toolset design. At the end of September, we will clean and transcribe the FG2 data, and in October, we will analyse the data using open-coding procedures and report the results. At the end of October, we will begin to specify the use cases and design the interface and interaction of the toolset based on our data.

3.2 Use Cases

This section sets out the aim and objectives defined with each trial partner, based on the initial phase of requirements gathering. **Figure 30** shows an initial simplistic categorisation of the trials based on the more predominant characteristic: technological, artistic or social. The following sections details the current common understanding of the trials that have informed the technological decisions.

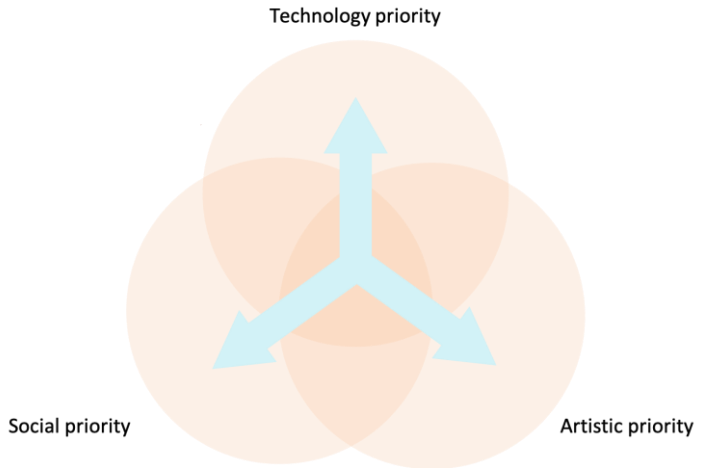


Figure 30: Categorisation of trials based on the more predominant objective

3.2.1 LICEU trial

The aim of the LICEU trial is:

- To increase the participation of residents of the Raval neighbourhood, with special attention to people with disabilities and migrants in Liceu's work.

It will achieve this by:

1. Working with community groups in Raval to build trust and common purpose
2. Using technology to facilitate their creative inclusion in access, design and marketing requirements of Opera Prima Raval
3. Working with them to identify how technology can enhance the audience experience
4. Co-creating digital capsules with community groups to extend the life and reach of the community opera
5. Research the process to identify learning of wider value to the opera sector

At Liceu, the TRACTION trial is only part of a much larger community opera project called Opera Prima, which was begun in 2018 and will continue until 2022, in its first phase. The trial therefore focuses on smaller areas with specific community groups. With regard to technology has been divided into three time frames.

In the first one, the trial will develop new resources to make the opera more accessible to people with sensory and cognitive disabilities, working with community partners. One question is whether technology can help the process of working with disabled end-users in the co-creation of new accessibility resources that can meet their needs.

In the second one, the objective is to include interactive digital technology on the foyer and public spaces around the time when the presentation of Opera Prima Raval at the theatre in November 2021. There is also a wish to be able to present the opera live in Raval, for people who cannot or do not want to come to the opera house. That might include a giant screen, links to screens in schools, bars or retirement homes, or through a mobile phone application.

In the third phase, the life of Opera Prima Raval is intended to extend beyond the two performances in the Gran Teatre del Liceu in autumn 2021. Technological capsules – ideally developed in co-creation with the community groups involved in the lead up to the opera – will be required to carry that experience forward, but it is not yet clear in what form or what content might be involved.

3.2.2 INO trial aim and objectives

The aim of the INO trial is:

- To make contemporary opera a route for social cohesion in Ireland

It will achieve this by:

1. Providing outreach activities that engage several communities in exploring opera's potential in expressing their own cultures and lives
2. Facilitating an opera co-creation project with communities and professional artists
3. Testing how digital technology can enrich opera creation and production processes
4. Producing and presenting a VR community opera in different parts of Ireland
5. Building understanding between different communities through the opera
6. Sharing the learning from the TRACTION experience with other opera companies and arts organisations

Two things are clear about the INO trial at this stage. The first is that it will involve three separate communities (one in suburban Dublin, one in the midlands and one in the Irish speaking west of the country). The second is that it will focus on virtual reality (VR). The idea is to co-create a work from the stories and experiences of people in the three communities, but it is not yet known whether this will result in one opera or three. In either event the work will be short and conceived from the start to be experienced in VR: it is unlikely to have a live performance, although it is possible that it might be presented in events with elements of live music and or performance to enhance the experience. There is an ambition to tour the work extensively, so that it is seen not only in the three communities, but throughout Ireland. The technology therefore needs to be robust and reliable, because it may be presented in places without technical support (or reliable Internet access). The trial will also test the value of virtual reality as a co-creation tool via co-creation workshops in year one and two.

3.2.3 SAMP trial aim and objectives

The aim of the SAMP trial is:

- To make opera a viable route to rehabilitation of offenders.

It will achieve this by:

1. Building the skills and enthusiasm of inmates and staff at Leiria Prison through multidimensional opera workshops;
2. Developing a professional music and theatre education course for use in prisons;

3. Testing ways of using digital technology to involve families and community in the creative process;
4. Co-creating and performing small opera elements in and outside the prison;
5. Producing a new large-scale opera involving the prisoners and the community;
6. Promoting understanding of and support for the work in wider society.

For SAMP the technology presents both difficulties and opportunities. The opportunity is to use the technology to cross the barrier that is the prison walls. Inside the prison there are 250 young offenders (18-25 years) who are separated from their families (mothers, wives, girlfriends, children) and from the community of Leiria. The community opera hopes to connect these three groups (prisoners, relatives and citizens) in a process of co-creation (and also involve other parties such as the ministries of justice and education). That can happen in two principal ways. During the co-creation of the opera (which will be a new work) there can be an exchange of small pieces of music and art on video or sound recordings, distributed to families via mobile phones. During the performance, the idea is emerging of a performance happening simultaneously inside the prison and in Leiria, with an audience for each, and communication between them being assured by technology.

The difficulties are principally about the security needs of the prison service, which will require very controlled access to technology by the inmates, where it is permitted at all. SAMP's proposed solution is to limit onsite technology to the Mozart Pavilion (a dedicated space that has been set up within the prison for SAMP's work) but this is not yet agreed. The other difficulty is the limited access to internet or smartphone services that some relatives may have, as well as the many social disadvantages that they are living with (for instance as a single parent of a child with a father in prison).

During the period of co-creation, it is anticipated that small performances will be presented in the city to test ideas, with the final production happening in year three. One idea envisaged to make all this feasible is a web resource about the project that has two elements. The first is a public-facing part that advocates for the value of this work in the rehabilitation of offenders through a rich presentation of material from the project. The second is a secure site where relatives can access material created by prisoners and contribute to the project from a distance. Again, the security obstacles are formidable.

3.3 Technical Requirements

Based on the identified initial objectives, the technical team has identified three main toolsets, with some basic requirements.

Media Vault (Liceu, SAMP and INO): the media vault is a software component that allows for the storage of a heterogeneous group of media objects (2D and 360 videos, volumetric media). It enables asynchronous communication between users around the uploaded content, for conversation and co-creation, and includes functionality for generating stories based on the stored content. The media vault has a strong requirement on accessibility at the user interface, for supporting its use by people with cognitive and sensory disabilities.


Performance engine (Liceu and SAMP): the performance engine is a communication infrastructure, deployed at the theatre or in the rehearsal rooms. It enriches the live performance by orchestrating in real-time the stage. It allows for remote participants to see the show and to contribute to it with them on media. This communication infrastructure does not only enrich the show, but as well enables synchronous communication between actors and spectators.

Immersive media environment (Liceu and INO): the immersive media environment includes both the authoring tools, for creating, and the rendering engines, for deploying, immersive and interactive experiences in the form of capsules or radically new Opera productions. This environment will enable static (dome-based) and moveable (HMD based) installations, pushing the boundaries of immersive media consumption. It will as well enable browser-based remote experiences.

The following sections details the requirements for each of the toolsets.

3.3.1 Media Vault

For the media vault a few key requirements need to be met. This section gives a brief overview of the technical underpinnings which are required to support these requirements in a service-oriented manner.

First and foremost, some form of online storage is required to store all assets which are required for the application to function. These assets include, but are not limited to graphics, icons, markup, script and style files as well content which is uploaded by users, such as video. For these videos, we may want to store originals as well as transcoded versions in different resolutions and bitrates. This online storage should function and be organised like a file system, with the ability to group assets into a hierarchical structure. While the end-user (by way of a web interface) should be able to read from this storage, they should not directly be allowed to write or modify it. Write access should be strictly limited to components which take user input and process/validate the given inputs before writing it to the storage. At the same time, access to this storage system should be shared among all components/processes with write access.

In order to facilitate fast and convenient access to static assets located in the online storage system, a content distribution system of some description would be advantageous. Such a system enables caching and distribution of files over a larger geographical area at edge locations to ensure data is always available in closest-possible proximity to end-users. Caching of the data also minimises direct reads at the file storage system, thereby reducing cost.

Further, as a point of interaction between end-users and the system, some form of compute engine, such as a cloud-based VPS (virtual private server) is required. This machine renders the web user interface with which the user interacts and performs the necessary processing and database queries. It will also submit long-running jobs to other processes in the system, such as video transcoding, through a common messaging bus. This compute engine should be set up in such a way that it can be seamlessly and flexibly duplicated in times of high load. This can either be achieved by multiplexing several instances behind a load balancer or spawning additional threads or processes within a

single engine. Besides running a user interface, this engine can also be employed to serve an API (application programming interface), which makes the data accessible computationally. This approach could facilitate the development of mobile and/or desktop applications which interact with the infrastructure.

As mentioned in the previous paragraph, long-running operations such as video transcoding should be offloaded to secondary services in order to guarantee maximum responsiveness of the web interface. Such a service will pick user-uploaded video files up from the file storage, process them and store the results back into the storage. It should be flexible enough to handle different input and output formats transparently and signal errors in a standardised way.

For storing user-generated content, structured data and asset metadata a hierarchical file storage is certainly not optimal. In this case, a structured database should be employed. This database can either come in the shape of a relational or document-oriented database. Both solutions allow for storage and efficient retrieval of data in a structured way. The database can be queried to transform and aggregate the data and retrieve analytics and insights. Furthermore, the database should store user profiles and access control rights. This database can be accessible through all services running on the system although it is not required. At the very least, though, it should be accessible to the compute engine running the web interface

Finally, to tie all services together and give them a convenient way to communicate, a message bus is required. This is yet another service running within the system. Processes can subscribe to message channels for events they are interested in, as well as publish messages to channels for instruct other processes to perform work. So for instance, if a user uploads a video through the web interface, the compute process will submit a message to a dedicated channel for transcoding operations. A transcoding process listening on that channel will pick up the metadata and perform the transcoding task. Once complete, it will publish another message on a channel informing service of success or failure. In this example, a process that might be interested in such a message might be the web interface, which would then update a database entry or a machine learning module, which could perform further analysis on the converted video.

As a summary, a working media vault should provide the following core components:

1. **Web services module**: Apache server and Kaltura layer 2, as a single access point for client-server applicative communication. This module should be deployed on front-end servers, with traffic distributed by load balancing equipment;
2. **Batch jobs module**: Scalable middleware entities deployed on back-end server/s. Central orchestration of atomic batch services such as media import, media info extraction, transcoding, server notification and others. This module should be deployed on a backend server.
3. **Transcoding module**: This module manages all media transcoding tasks, by utilizing open source and/or commercial transcoders. This is a CPU intensive module and could either be deployed on a backend server at a local deployment or can be distributed using independent transcoding servers deployed in a cloud solution.

4. **Shared storage**: A dedicated disk space that is shared and accessible by all of Kaltura's servers within a specific deployment. The Shared storage holds all content and application files, including media assets, Kaltura widgets or applications, skins, thumbnails, players/playlist configuration files (UI conf) etc. The shared storage can be deployed as part of a local deployment or using independent storage within a cloud solution.
5. **Operational database**: This is the applicative database, used for storing and managing both content related data (metadata, identifiers, URLs etc.) as well as application and business logic supporting data. The operational database should be deployed as part of a local deployment, preferably on dedicated servers utilizing a master/slave topology.
6. **Site admin console**: This module is responsible for operating Kaltura's Admin Console, enabling site administrators to monitor and operate their own deployment of Kaltura's online video platform.
7. **Video analytics module**: This module is responsible for operating Kaltura's Admin Console, enabling site administrators to monitor and operate their own deployment of Kaltura's online video platform.

3.3.2 Performance Engine

For orchestrating the performance in real-time, several roles are needed. There must be at least an *administrator*, responsible for the whole performance, using one or more devices depending on the complexity of the application and the number of services to be managed. Another important role is that of the *operators*, who are responsible for managing which content is visualized on different displays. Each operator is typically in charge of multiple devices at the same time. The final role is the one represented by *end-users*. This all-encompassing role represents everything that is neither an administrator nor an operator: all the displays in the stage (or the fully digital representation), the audience providing additional information through their mobile phones, producers who want or plan to add content during the performance or even remote audience who, while watching the performance through a TV, laptop or mobile device can send feedback and content through their smart devices.

In terms, of functionality, the engine is required to support several multimedia formats: first and foremost, it should be able to fetch and play the videos and images uploaded by the users to the media vault, as it will also act as a database for User Generated Content (UGC). Apart from this data, the engine must handle real time data, as it will be in charge of streaming real time videos from professional cameras in the stage as well as videos created by on-site and remote audience. The engine should support static and animated images, textual information, illustrations, presentations as well as UGC like social media posts and reactions. Finally, the engine should ideally handle advance multimedia information such as 360 or volumetric videos and manage how this type of content is handled on devices with reduced processing capabilities.

Furthermore, the engine should be developed with the possibility to easily support other current or future data types: different scenarios would have different requirements, and

new standards for audiovisual content appear every few years. That is why extensibility and flexibility are parts of the Performance Engine requirements.

To ensure multimedia synchronization, requirements include support for WebSocket to provide clients communication. A server providing a master clock (based on *TimingService* or similar libraries), calibrated to the multimedia playback rate, is also required. Furthermore, the engine should also be able to operate, when requested, in an unsynchronized scenario, where all video and audio content is sent and played on a best-effort mode with the aim to keep latency to a minimum.

3.3.3 Immersive Media Environment

The authoring and rendering of immersive media require capabilities that allow import of multi-media assets, spatial and temporal arrangement and interconnection of these assets into a completed immersive experience, interactions with the experience through interfaces and peripheral controls (UX/UI) and finally exporting the entire experience to an enabling platform for consumption.

The engine for authoring and rendering immersive content must support the importation and manipulation of multiple files and assets, such as 3D models and textures, volumetric videos and photos, photogrammetric and mesh-based depth information of objects and locations, audio in various formats (mono, stereo, ambisonic, etc.), 360-degree video (monoscopic and stereoscopic), 360-degree photos and standard “2D” videos and photos.

Spatial and temporal arrangement/interconnections must allow creators to arrange all assets in a non-linear manner within a spatial virtual environment. Assets may be passive or interactive and may need to be controlled or guided by the user through the use of digital or physical interfaces, such as gaze control, motion control, computer peripherals, voice and hand recognition.

Gaze control affects computer actions by changing the direction of one's gaze. This involves determining the angle or position of a user's visual attention, usually with cameras, and choosing from a set of available instructions that are mapped to those positions

Motion controller is a type of game controller that uses accelerometers or other sensors to track motion and provide input, this can generally be 3DOF or 6DOF. Other standard peripherals or HID (human interface devices) such as keyboard, mouse, joystick and touch screens should also be supported, as they are the most traditional input methods.

Voice recognition, in particular for simple commands, has advanced rapidly in recent years, as well as hand recognition and tracking features, which enable the use of hands as an input method on devices. It delivers a sense of presence, enhances social engagement, and delivers more natural interactions with fully tracked hands and articulated fingers.

Immersive experiences must be exportable from authoring tools to enable unified VR experiences to take place. We can extend or use the capabilities of Unity Cloud services to enable this for TRACTION.

The main technologies suitable to author/build projects for immersive platforms include Unity, Unreal Engine, WebXR and Mozilla Hubs. The applications created in these engines must support the inclusion of assets generated in other platforms, such as 3D models and audio. That will result in the final immersive experiences to be delivered to users.

Therefore, the immersive media engine must have important features for immersive content authoring, such as build, compilation and versioning capabilities, support to numerous file types to be imported into the immersive application, support to multiple input devices and hardware, and export features for multiple operating systems and devices.

Assets and applications must be stored on a repository with all authors' creations, including 3D models, videos and audio. The repository must also contain the compiled applications for download.

The immersive engine server must also contain file management capabilities and a control centre for servers monitoring, video and audio transcoding tools, and user management features, as multiple authors from different trials must have access to specific files/folders.

User devices such as desktops, smartphones, tablets and VR headsets must be able to support the immersive content created in TRACTION. These devices must be able to install the applications, which have minimum requirements on operating systems, storage, memory, graphics APIs and CPU. Audio support and compatible web browsers are also required.

Web-based players can be used for streaming of 2D and 360° content. The player must have accessible features, such as subtitles, audio description, sign language videos, as well as well as support to adaptive standards (e.g. MPEG-DASH) for content adaptation. Adaptation also requires that transcoding technologies must be implemented into the storage and servers, for automatic encoding of video into multiple resolutions.

4 Architecture and Integration

As discussed in the previous section, the TRACTION consortium has identified three main tools to be developed:

- Media vault (see Figure 31) for content storage and asynchronous communication;
- Performance engine (see Figure 32) for stage orchestration and synchronous communication;
- Immersive media environment (see Figure 33)for the creation and rendering (and distribution) of interactive and immersive experiences.

Figure 31: Media Vault (sketch)

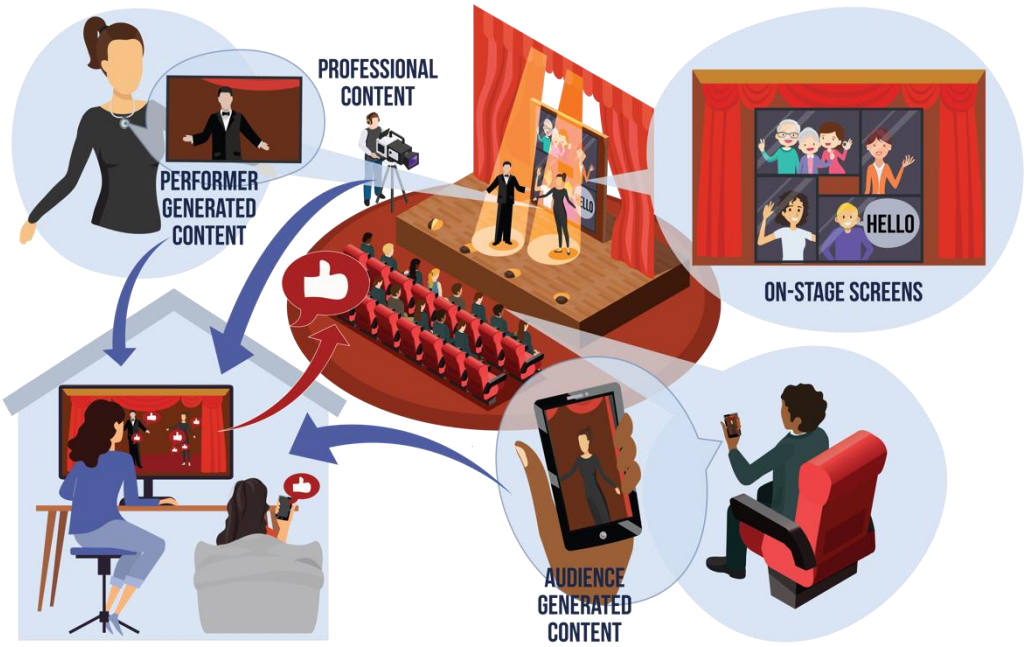


Figure 32: Performance Engine (sketch)

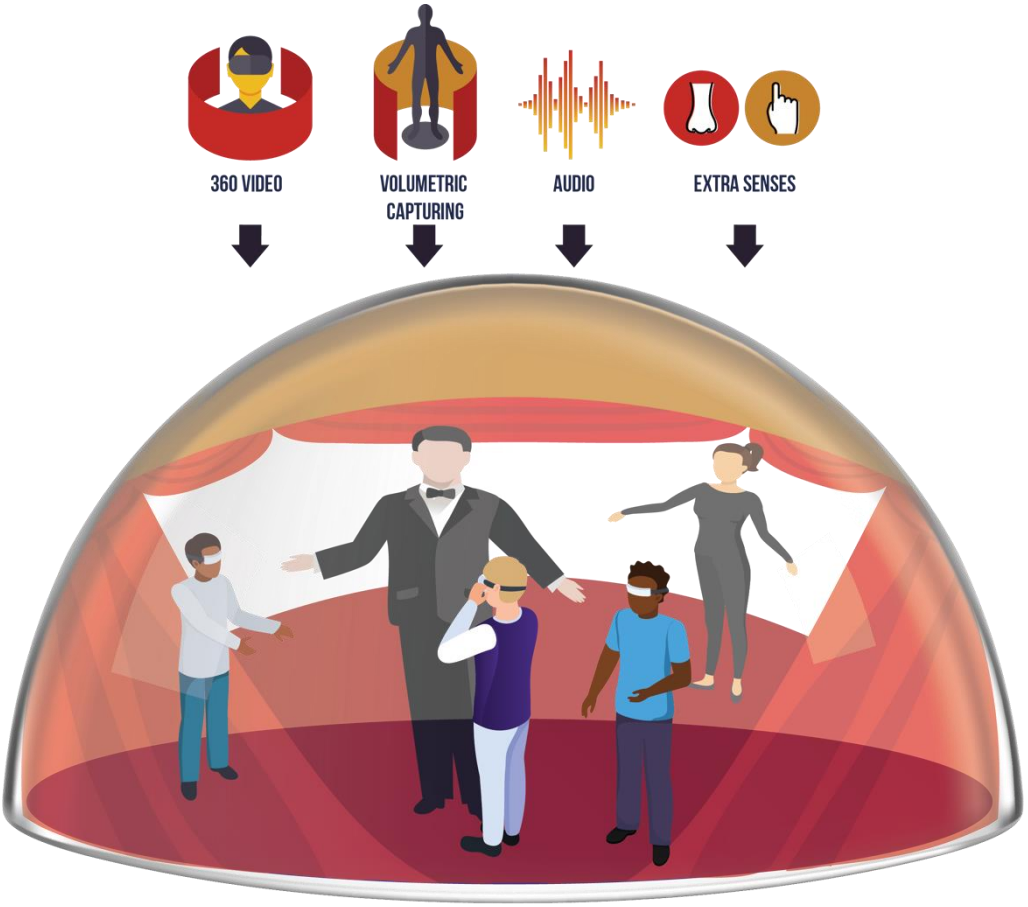


Figure 33: Immersive Media Environment (sketch)

Based on the comprehensive survey of technologies (see Section 2) and the requirements (see Section 3), the technological team has explored different possibilities for satisfying the requirements, and run a number of tests, to assess the benefits of each of them. The final decision, as justified in the sections below, are:

- To build the media vault on top of AWS.
- To build the media engine on top of FlexControl.
- To build the immersive media environment including the ImAc player for accessible 360 content visualisation.

4.1 Media Vault for asynchronous communication

Two options were considered for the creation of a media vault for asynchronous communication. The first option involved using Kaltura and extending it to fulfill TRACTION requirements, while the other options revolves around using the AWS infrastructure and develop our services on top of that.

4.1.1 Option 1: Kaltura and extensions

Due to the limitations of the free edition of Kaltura, and to the specific requirements of TRACTION, the idea is to provide an additional platform to pair together with Kaltura, in order to:

- Provide webservices to end-users.
- Include application logic in the web application.
- Consume resources through TRACTION RESTful API.

Designing this additional platform would guarantee the possibility to provide an easier to use and visual appealing user interface, where different roles and permission could be specified for different stakeholders. Figure 34 shows the proposed architecture, where the Kaltura platform on the left connect with the TRACTION platform, through several scalable instances. The proposed technological stack for the TRACTION platform is based on NodeJS for the backend, MongoDB as a database and Angular for the front-end.

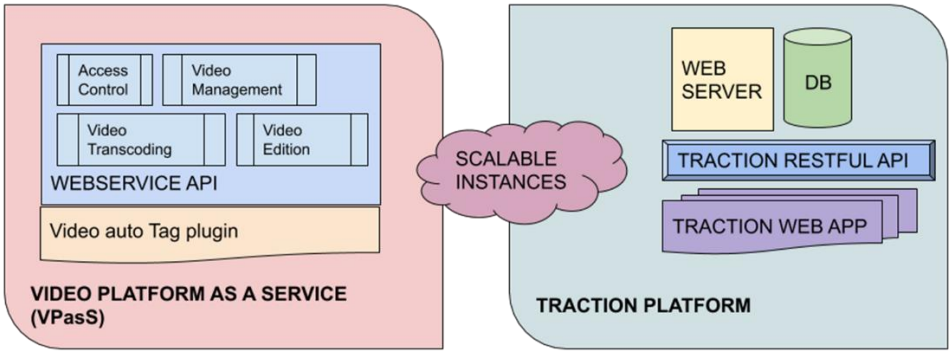


Figure 34: Proposed architecture combining Kaltura and TRACTION platforms

The Kaltura backend should also be extended, to provide additional services for accessibility, backup, redundancy etc. As an example, a new batch service for the addition of subtitles was developed. Such service runs each time a new video is uploaded, and, after the completion of the transcoding services, it adds subtitles to the video if a subtitles file (in *srt* format) is provided. Figure 35 shows the Kaltura Admin Console displaying a graph of the services running for a video who was recently uploaded. The service adding subtitles is called each time a video has been transcoded after it has been uploaded to the database.

Developing new batch services for Kaltura is not an easy task: the platform documentation is lacking on many aspects, while the addition of a new service requires modifying several files in different parts of the repository. Ideally, adding a new batch service should require minimal changes to the existing source code and only require the addition of the classes which implement the service.

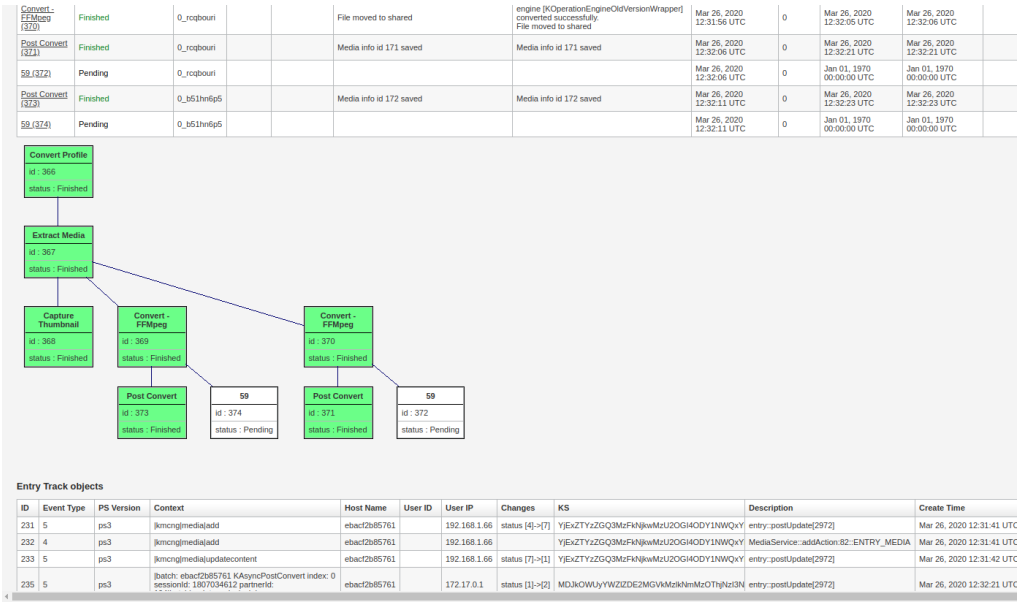
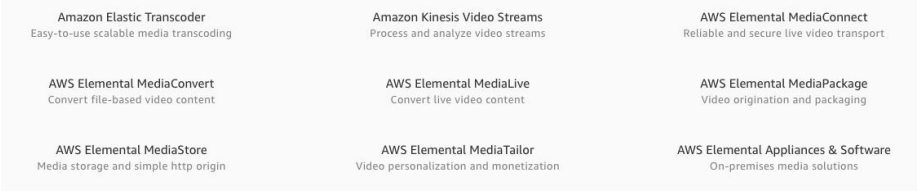


Figure 35: Kaltura Admin Console showing the subtitle batch service running.

4.1.2 Option 2: AWS infrastructure

For the purpose of evaluating the usability of AWS for the project, we investigated which services would fulfil the requirements for the project.

Amazon Web Services offers a multitude of services for storage, distribution, processing and analysis of data. The following figure shows AWS' selection of services specifically related to media processing, which could be of potential use to the project:



Services

Of special interest for this project is the *Elastic Transcoder*. This document outlines the possible use this service in combination with a selection of other services to upload and store user-generated video files:

- **S3 (Simple Storage Service):** Storage of arbitrary files, used to store web assets as well as uploaded video files. The service organises files as objects inside so-called buckets. Each object within a bucket has a unique key and prefixes can be used to give the bucket a hierarchical structure, much like a common file system.
- **EC2 (Elastic Compute Cloud):** VPS hosting, used for hosting backend code which processes user inputs, host the API or website assets. Is also responsible for transferring uploaded content to S3, storing metadata in a database or starting the transcoding process. Note that EC2 is not the only option here. Based on specific requirements, this can be replaced by *Elastic Container Service* (equivalent but uses containers), *Lambda* (breaks functionality down into functions which are run in response to specific events) or a whole Kubernetes cluster.
- **Elastic Transcoder:** Video/audio processing, used to transcode files uploaded by users. This service takes files from S3 and puts them through a transcoding pipeline which can convert videos/audio to different resolutions or file formats and places the resulting files back into S3.
- **DocumentDB:** Structured data storage, can be used to store file meta- data, user data or and type of structured data in a document-oriented format which is compatible to MongoDB. Alternatively, this service can be replaced by *RDS*, which instead of document-oriented data storage stores the data in a relational manner.
- **Cloudfront:** CDN, used for distribution and caching of data in geographically advantageous locations relative to the client. This is largely optional, but will improve response times on the end-user side.

Figure 36 outlines a possible architecture which serves to realise a prototype for the project using the services described in the previous paragraphs:

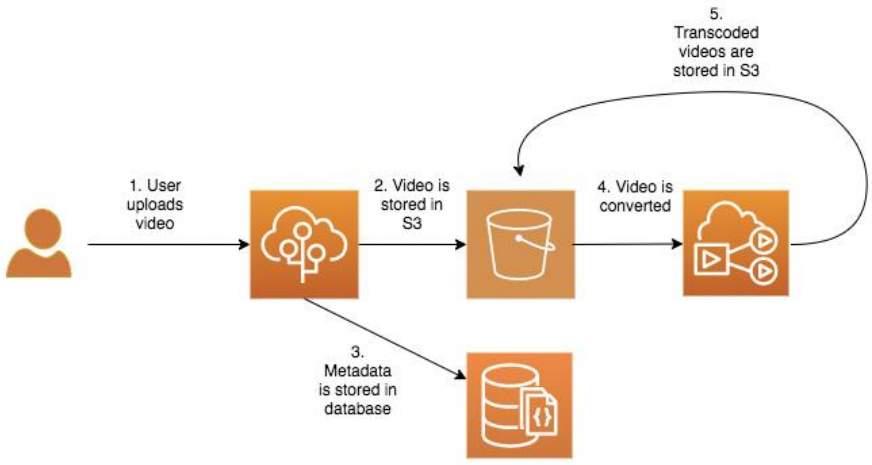
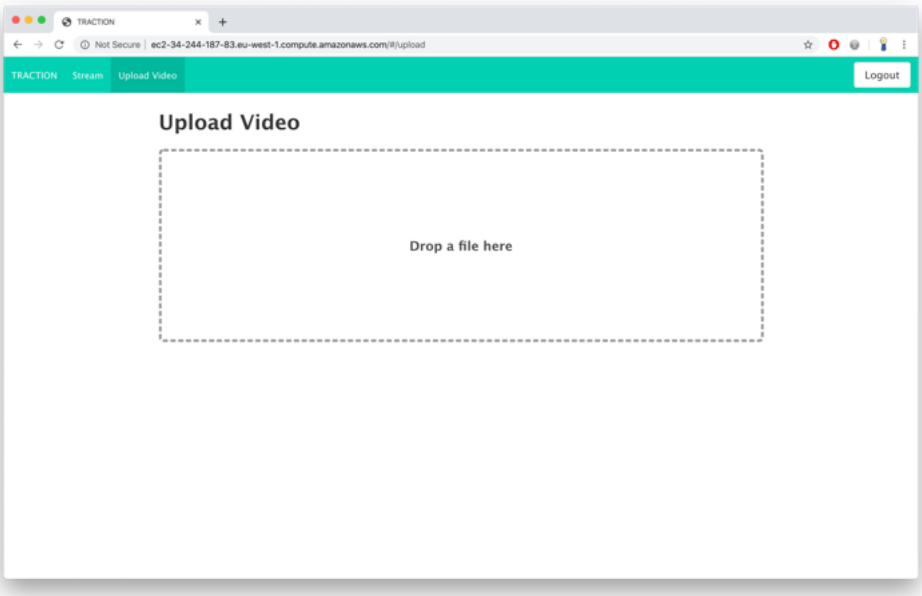


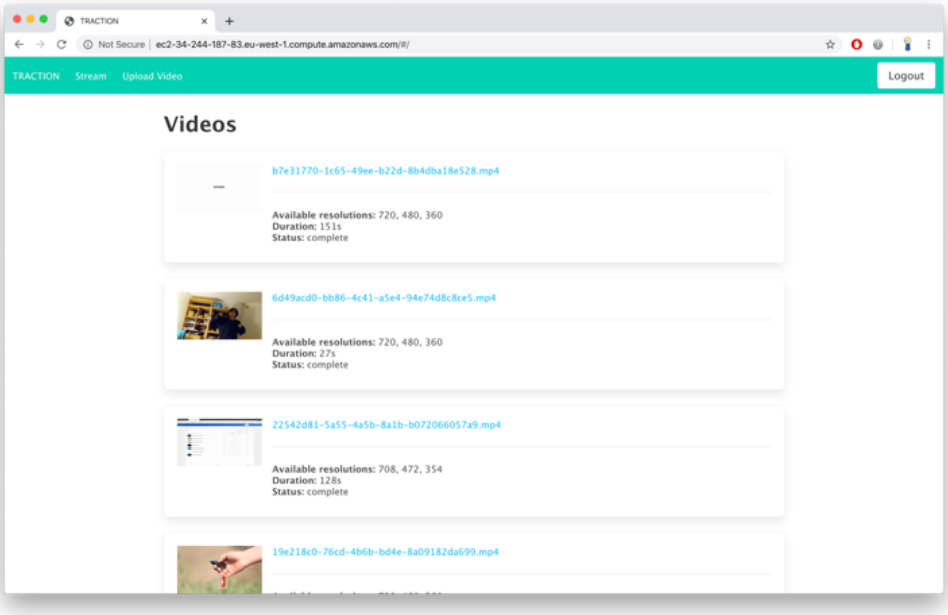
Figure 36: Proposed Architecture for the Media Vault

The advantage of this deployment is the fact that the project can offload management and maintenance of the services to AWS while the data also remains easily accessible and runs in a highly performant infrastructure. So for instance, if one wanted to automatically generate subtitles or perform computer vision tasks/machine learning on an uploaded video, this could be performed by an entirely separate service, which is triggered once a video is uploaded to an S3 bucket. Once completed, the results (e.g. the generated subtitles) are simply placed alongside the relevant entry in the database and are easily searchable through a standard query language.

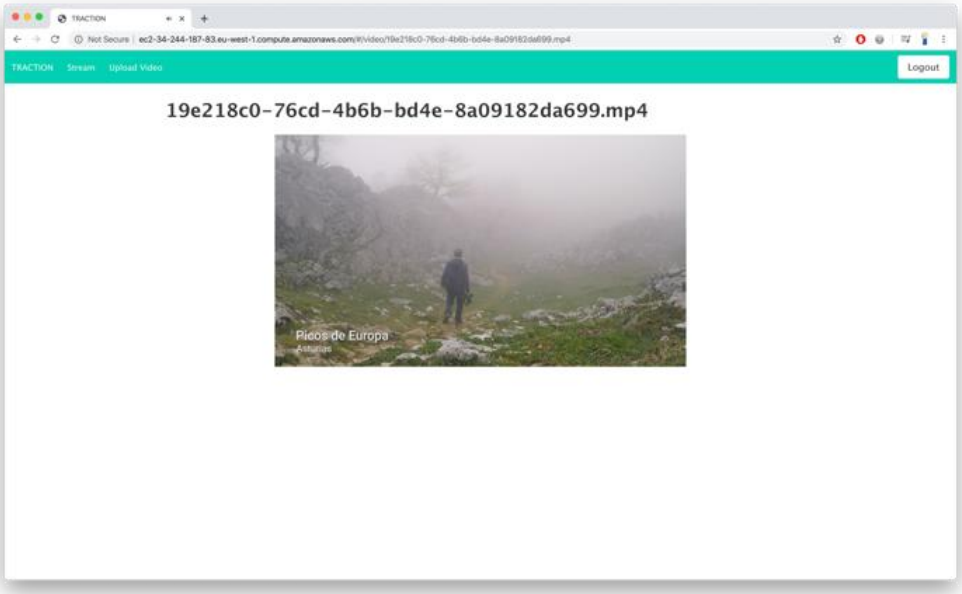
Based on this selection of services and the architecture diagram a small prototype has been developed and deployed. It makes use of S3 for data storage, Cloudfront for distribution, EC2 for providing a virtual machine and Elastic Transcoder for conversion of media. The prototype consists of a Express.js web application implemented in Typescript running on an EC2 virtual machine. The web application itself runs inside a Docker container and uses a separate container running MongoDB for data and session storage. It consists of a simple login and registration system, which, after logging in, allows the user to upload videos by means of drag and drop as shown in the figure below.



A video file dropped onto the marked zone is uploaded to the web server where it is analysed to ensure that it is a valid video file and to determine which streams it contains. If the file is valid, it is the stored in an S3 bucket and the encoding process is started. The encoding is carried out by Elastic Transcoder, which will encode the video stream as fragmented MP4 files in three different resolutions. The audio stream, if present, is also encoded in a fragmented format. After that, thumbnails for the videos are extracted and a MPEG-DASH playlist is generated. All these files are then stored back onto S3and a notification is sent on a dedicated publish/subscribe channel using SNS (Simple Notification Service). This alerts the web application that the transcoding process has completed successfully, and a corresponding entry is made in the database. After this, the video is accessible to the user.



The figure above shows a list of available videos, listing their duration, name, thumbnail and their current status. All videos start out with the status ‘processing’ after they’ve been uploaded. Once a notification is received from the transcoding service, their duration is updated and their status is updated to ‘complete’, or ‘failed’ in case the transcoding process could not be completed. Clicking on a video leads the user to another page, where the video can be watched.



The video on this page is streamed using the open-source player video.js from a MPEG-DASH stream, which automatically switches to the most appropriate resolution and bitrate based on end-user device and network conditions. The player also supports other streaming formats such as HLS and can play 360/VR videos using a plugin system.


Budgetary Considerations

During the development of the prototype, we also took budgetary constraints into consideration. The following

- **ETS:** The most costly aspect of the estimate is video transcoding. The cost of this is highly variable, as it is billed by number of outputs and length of the submitted videos. Through their acquisition of Elemental, Amazon has launched another transcoding service dubbed Elemental MediaConvert, which is claimed to be cheaper and more appropriate for most applications. Pending further investigation, this may provide an alternative to ETS. Another possible alternative to ETS would be to wrap FFmpeg and other command line tools into a container and provide an API similar to ETS to convert videos locally inside a docker container. This solution will be outlined in a later paragraph. An example of how this could be achieved in principle can be found here: https://rybakov.com/blog/mpeg- dash/
- **S3:** Another big share of the costs is taken up by storage on S3. This service is the most difficult to feasibly replace in a local setup as the project working with content that needs to be shared over the internet. S3 provides virtually unlimited storage and makes the data accessible worldwide through a simple API. There is, however, an open-source project called localstack (https://github.com/localstack/localstack), which includes a drop-in replacement for S3 that can be run in a local environment. This of course does not make the data accessible over the internet by default.
- **EC2:** The developed prototype was deployed on a medium-sized EC2 instance with a single CPU, 8GB of memory and 30GB of block storage. This should be sufficient, as its only task is running the Docker container infrastructure for the production environment. During development, the containers can be run locally. An alternative to this could be other cloud providers like DigitalOcean. Their pricing, however, seems to be in line with EC2 on most accounts, though offering more straightforward deployment.

Potential Alternative to Elastic Transcoder

As mentioned in a previous paragraph, it is possible to replace the use of Amazon's ETS or MediaConvert for transcoding video through a localised service. Sources for this can be found on the Internet, as well as in previous EU-projects, such as ImAc, which took a very similar approach. They made use of the open-source tool FFmpeg and its media transcoding abilities.

In essence, an option for this project would be to wrap FFmpeg and another open-source tool by the name of MP4Box inside a Docker container and make it accessible through a web-based API, where other services can submit transcoding tasks to the service via this API. FFmpeg would convert the video into a common format, such as MP4, into different resolutions and bitrates. Once complete, MP4Box would be responsible for generating a DASH manifest, store the resulting files back to a known location and inform interested parties of completion or potential errors.

While this approach will incur additional time spent on development and cause uncertainty when it comes to performance compared to a cloud-based solution, it provides the advantage of having full control of the inner workings of the solution and can be easily adapted to the requirements of the project.

Hosting Considerations

An important concern for EU-project is hosting the data within the European Union. For this, AWS offers to allocate the services/data within any of its regions.

In the European Union, these regions are Ireland, France, Sweden and Germany, with regions in Italy and Spain to launch soon. Note that not every service is available in every single region, but every service should be available in at least one region per continent. So for instance, all of the services discussed in this document are available in the Ireland region.

Moreover, apart from these service zones, AWS also maintains Edge Locations in most European countries. These edge locations are where servers for Cloudfront are located to guarantees lowest possible latencies for data delivery.

4.1.3 Decision

After careful consideration of the options that were evaluated in the experiments by comparing their strengths and weaknesses, the project settled on developing their own infrastructure based on AWS.

While the Kaltura platform offers video transcoding out of the box, we found it more positioned by its creators as a white label media platform to add video functionality to existing websites. Thus, the functionality should serve most general use-cases, but extending it would require more effort. While the platform indeed provides facilities for adding new functionality, we found it to be somewhat inconvenient. This is also not helped by the lack of detailed developer documentation. Furthermore, the platform is based on PHP and backed by a MySQL database. Developing our own infrastructure provides us with making more informed decisions about using not only the best-suited technology for each task, but also the technology we are most comfortable with. Another issue which made Kaltura not well-suited for our use-case was the fact that some aspects of the infrastructure would have to be run by secondary services (such as machine learning) tasks, which would be needlessly complicated without direct access to the file storage used for the video assets and the database which stores the metadata.

Developing our own infrastructure gives us the greatest amount of flexibility and structuring the platform as a series of independent services communicating through a


message bus ensures that the work can be parallelized as much as possible. We will use off-the-shelf open-source components and services where appropriate and wrap them into an interface in order to make them communicate with the rest of the system. Running the system will be facilitated by a container-based infrastructure which allows for the easy deployment of a distributed service-oriented infrastructure locally as well as in the cloud.

Although most of the services will either be developed by the project and run inside containers, for some, such as storage, content distribution, compute and possibly transcoding we will resort to services provided in the cloud by Amazon. Moreover, we evaluated several solutions for handling authentication and user profiles and pending a final decision we might employ a third-party solution for it. The approach that will finally be employed for this purpose should transparently plug into the existing infrastructure.

The following list details the tools and technologies that shall be employed for the development of the video vault:

Services:

- Amazon S3: Cloud-based file storage for assets and user-uploaded content
- Amazon EC2: Compute engines running the container infrastructure
- Optional: Amazon Elastic Transcoder or MediaConvert for video transcoding as well as Amazon Simple Message Service for communication between services

Technology:

- JavaScript/Typescript: Backend server code and interactivity in the frontend
- MongoDB: Storage of structured data and user profiles as well aggregation of analytics
- Alternatives to cloud solutions: Container running FFmpeg for transcoding and a Redis-based publish/subscribe solution for asynchronous communication between services

4.2 Real-Time Performance Engine

FlexControl (see Section 2.2) can be transformed towards a rich media engine in TRACTION, in order to manage live media, on-demand content, and multiple sources and devices, to provide an orchestrated and synchronised experience. This toolset can help to improve and enhance a traditional opera show, or to provide a completely digital novel opera format. FlexControl is a software library registered by VICOM, which is built on top of open source licenses that allow the distribution and commercialisation of the software citing the authorship of the libraries:

- Motion module: LGPL
- Shared Data module and MappingService: Apache License, Version 2.0
- Polymer: BSD
- Janus Gateway: GPL v3
- Gstreamer 1.14 LGPL
- Libnice LGPL

VICOM, as a non-for-profit research institute and the coordinator of TRACTION, is open to evolve FlexControl towards part of the toolset of TRACTION, with the vision to follow an open approach (open source licenses, etc.). The following figure depicts the potential contribution of such a tool to TRACTION.

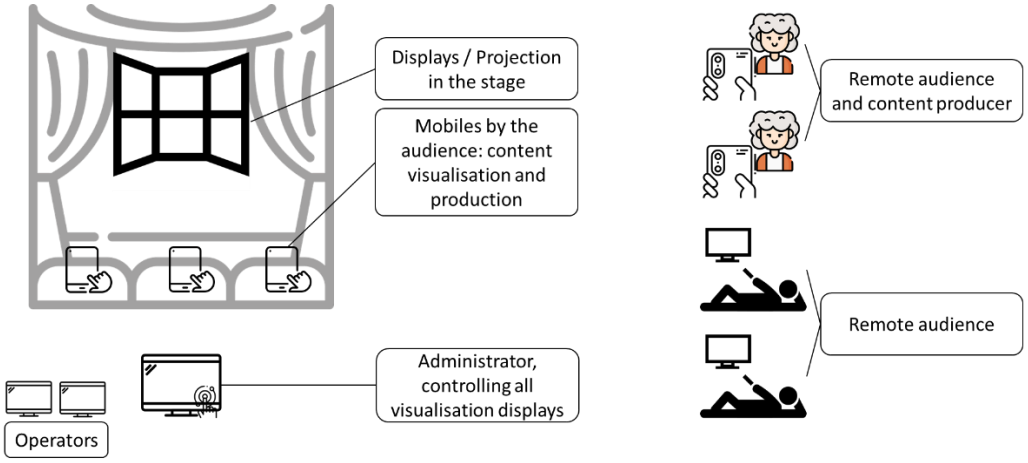


Figure 37: Proposed Architecture for the Performance Engine

This Rich Media Engine will be able to manage live streaming show, where the administrator could create a real-time production to be consumed remotely and customise and enhance the production to the audience. However, it could also be used to manage an on-demand show, where all the content could be synchronized. The FlexControl architecture, illustrated in Figure 38, shows the different modules that allow creating a multimedia multi-device application. Such applications require two servers:

- A *FlexControl* server, for managing and controlling sessions in a cryptographic secure way;
- A *Stream* server, for managing and controlling multimedia streams in a secure and adaptive way.

The FlexControl server hosts three modules:

- Authentication, responsible for user verification;
- Mesh, responsible for device synchronization among the different sessions;
- Control, for the platform administration tasks as well as resource assignment.

The Stream server is composed of the modules required to manage the multimedia streams of any data source. The most common modules added to the server include a Janus Gateway, a Nvidia Jetson service, an IP camera stream panel.

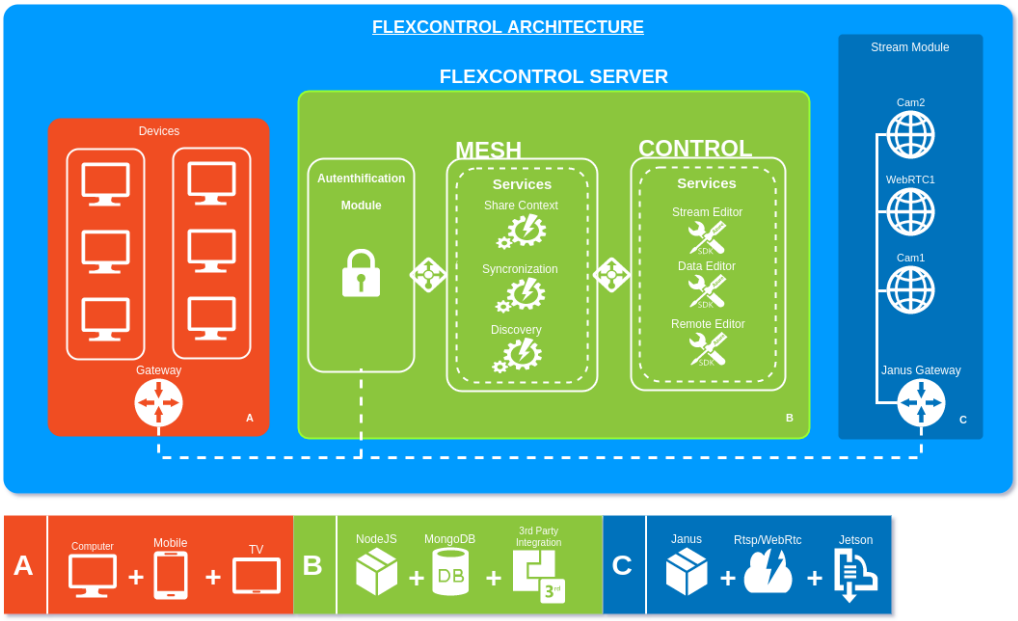


Figure 38: FlexControl architecture

4.3 Immersive Media Environment

Several tools are used in the process of immersive media authoring. This toolset can be used for the creation of complex and engaging experiences, as illustrated in the architecture for immersive media, in Figure 39. These tools allow creators to produce several types of assets, such as 360° videos in monoscopic and stereoscopic formats, volumetric captures using specific tools and scripts, 3D models in tools such as Google Tilt Brush and Blocks from the Google Poly platform, audio/ambisonics created with Facebook's 360 spatial workstation, etc.

A variety of creation tools is being reviewed for the authoring process and some of them are going to be used by experienced art directors while other tools will be used by community members. The process for selecting these tools include an analysis of their features and ease of use, and also workshops with users, which will indicate the preferred tools to become part of toolset.

The creation process aims to generate several files and assets that can be combined in the final applications compiled for multiple platforms, and installed in user devices (e.g. smartphones, desktops and VR headsets). Unity can be used in the creation of these immersive applications, as it provides features for importing various audio and visual assets, creation of 3D environments, and compilation into executable applications.

After the creation process, the created assets and applications must be stored in the TRACTION Media Vault. The TRACTION Media Vault integrates and stores the content (in several file types) developed by authors in the various authoring platforms and must provide tools for the management of the content and its metadata as well as for download/upload of immersive applications and assets, and streaming of 360° and 2D video (HTTP and FTP). Other important features of the media vault include storing accessible content (e.g. subtitles, audio description, sign language videos) and the

transcoding of content into different resolutions and into MPEG-DASH, for adaptation and compatibility with the content player and web browsers.

The polygonal 3D applications available in the Media Vault are usually installed on desktops, mobile and VR devices, while 360° and 2D videos can be streamed over the Internet. Some 3D applications and assets, available in a web server, can also be delivered through web browsers that support WebXR.

The web-based TRACTION player is going to be used for viewing content that does not need to be installed on user devices, such as 360° and 2D videos, and simpler 3D applications. The player runs on devices' web browsers and it expands the ImAc player, described in section 2.3.5. The ImAc player requires a webserver (e.g. Tomcat), a HTTP server (e.g. Apache), and it currently has the capabilities of playing 2D and 360° content encoded in MPEG-DASH. In addition, the ImAc player contains accessible features, such as voice control, large menus, subtitling, audio description and sign language support.

The TRACTION player needs new features that are not available in the ImAc player. The first expansion is the ability to connect to the Media Vault, through APIs. Authentication can also be considered for content delivery based on the user.

WebXR can potentially be added to the player, allowing the delivery of polygonal immersive applications through web browsers, and support for WebXR can be added with JavaScript APIs. WebXR is a novel and evolving technology, so we need to continuously monitor changes in the API and browser requirements.

Another important feature to be added to the player is the introduction of novel adaptive algorithms based on device, user and network requirements. The development of research in these algorithms will allow multiple concurrent users located in areas with limited Internet bandwidth and with a variety of devices to access and produce content at higher quality, even in constrained environments. The player must generate KPIs for the analysis of the network status, and the algorithms must adapt content in order to maintain higher quality of service and quality of experience.

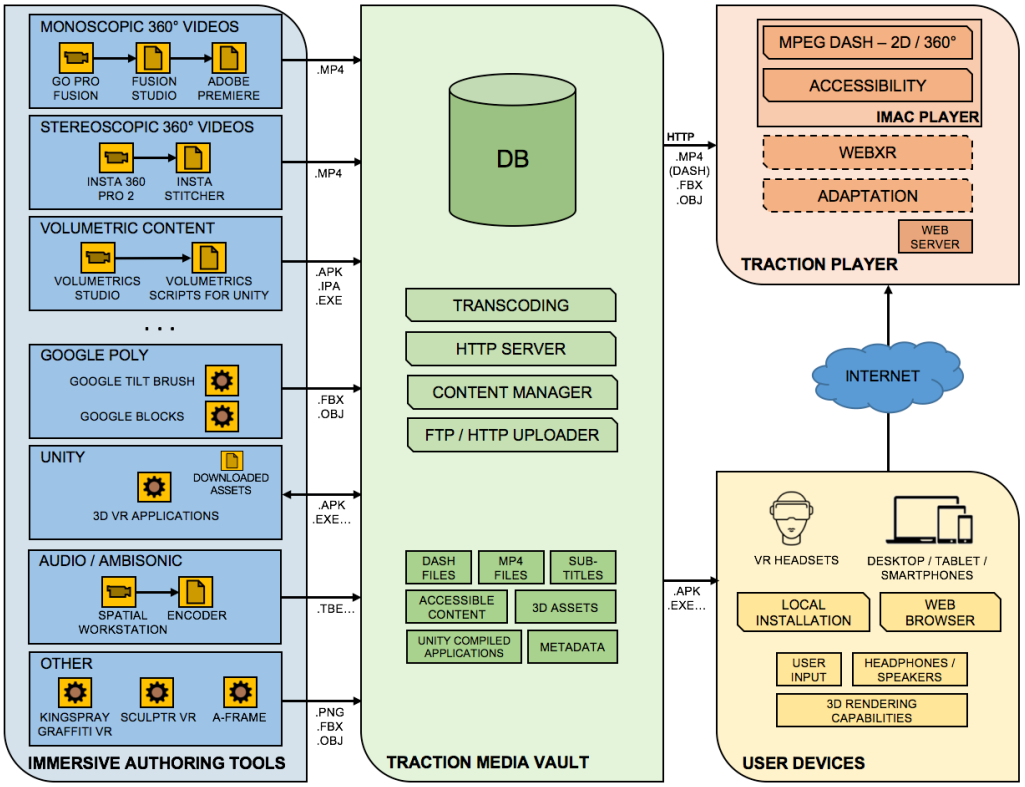


Figure 39: Proposed Architecture for the Immersive Media Environment

5 Conclusion

This deliverable is the first iteration (as of month 4) of the technical requirements, architecture and integration of the TRACTION toolset. Subsequent versions of the deliverable will provide more details and information, towards a fully functional and deployable solution for the trials that will take place during the project.

In particular, this deliverable has three main contributions:

- A comprehensive survey on the technology and infrastructure that can be used for realising the technology envisioned by the project.
- An initial set of requirements, based on a user-centric methodology, and proposal of a timeline for a more structured approach to continue gathering requirements.
- An initial architectural and integration decisions, based on the survey and the requirements, identifying three main software components that will be developed by the project.

The consortium is satisfied with the initial results included in this deliverable and will continue to work on two main activities.

As detailed in the work plan, a more structured gathering of requirements will continue based on two sets of focus groups with representatives of the trials. While the first round of requirements has allowed the technical team to initiate the development of the tools, the second round of will provide more detailed requirements in terms of actual functionality, user groups, and expected interfaces. The objective is to ensure that the ongoing alignment between technology (WP2), trials (WP3) and user experience expertise (WP4), allows the development of the adequate tools and interfaces.

In parallel to the gathering of more concrete requirements, the technology development will continue based on the initial architecture detailed in this deliverable. The intention is to develop and test the basic infrastructure, that will later on, provide the basis for the toolset. In particular, the media vault will evolve by including a number of basic services, the performance engine will be extended for supporting extra timing and layout functionality, and it will be explored how to include WebXR support for the immersive media environment.

References

Carneiro, G. e. (2019). Deb8: A Tool for Collaborative Analysis of Video. ACM International Conference on Interactive Experiences for TV and Online Video. ACM.

Gadaleta, M. et al. (2017). D-DASH: A Deep Q-Learning Framework for DASH Video Streaming, IEEE Transactions on Cognitive Communications and Networking, 3(4) 703–718.

Ghassaei, A. (2017). Intro to MaxMSP. Retrieved from https://www.instructables.com/id/Intro-to-MaxMSP/

Hosseini, M. et al. (2016). Adaptive 360 VR Video Streaming: Divide and Conquer! IEEE International Symposium on Multimedia (ISM). 107–110.

Jiang, J. et. al. (2014). Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive, IEEE/ACM Trans. Netw., 22(1), 326–340.

Li, Z. et al. (2014). Probe and adapt: Rate adaptation for HTTP video streaming at scale, IEEE J. Sel. Areas Commun., 32(4), 719–733.

Ljubojević, M. e. (2017). "A COMPARATIVE ANALYSIS OF THE ONLINE VIDEO PLATFORMS FOR INTERACTIVE MULTIMEDIA DELIVERY. Aktuelnosty.

Ma, X. a. (2017). Video-based evanescent, anonymous, asynchronous social interaction: Motivation and adaption to medium. ACM Conference on Computer Supported Cooperative Work and Social Computing. ACM.

Montagud, M. et al. (2018). ImAc: Enabling immersive, accessible and personalized media experiences. TVX 2018 - Proceedings of the 2018 ACM International Conference on Interactive Experiences for TV and Online Video. 245–250.

Montagud, M. et al. (2019). ImAc Player: Enabling a Personalized Consumption of Accessible Immersive Contents. Adjunct Proceedings of the ACM International Conference on Interactive Experiences for Television and Online Video (TVX 2019).

Montagud, M. (2019). ImAc Project - Deliverable 3.5. Player. Available at: https://imac-project.eu/wp-content/uploads/2019/09/D3.5.pdf (Accessed: 27 February 2020)

Muntean, G-M. et al. (2008) Region of Interest-Based Adaptive Multimedia Streaming Scheme, IEEE Transactions on Broadcasting, 54(2).

Place, T.; Lossius, T. (2006). A modular standard for structuring patches in Max. Proc. of the International Computer Music Conference 2006. pp. 143–146.

Rossi, G. a.-J. (2019). Libreflix: A Peer-to-Peer On-demand Video Platform for Free Streaming. do XXV Simpósio Brasileiro de Sistemas Multimídia e Web.

Watts, L. (2016). Synchronous and asynchronous communication in distance learning: A review of the literature. Quarterly Review of Distance Education.

Wu, Q. Y. (2019). Danmaku: A New Paradigm of Social Interaction via Online Videos. ACM Transactions on Social Computing, 1-24.

Yin, X. et al. (2015). A Control-theoretic approach for dynamic adaptive video streaming over HTTP, SIGCOMM 2015 - Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, 325–338.